

Chapter 8: Vehicles, Animals and Furniture

§8.1. *Bicycles, Cars and Boats*; §8.2. *Ships, Trains and Elevators*; §8.3. *Animals*;
§8.4. *Furniture*; §8.5. *Kitchen and Bathroom*

-  Contents of *The Inform Recipe Book*
-  Chapter 7: Other Characters
-  Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
-  Indexes of the examples

§8.1. Bicycles, Cars and Boats

The vehicle kind in Inform refers to an object which can carry at least one person, but is small enough to fit into a single location:

In the Garden is a vehicle called the motor mower.

We can then apply different rules to a player going somewhere on foot or in the vehicle. **Peugeot** (a bicycle) is an easy example; **No Relation** (a car) adds an ignition switch to the vehicle; **Straw Boater** (a motorboat) gets around areas of lake where travel on foot is not just slower but impossible.

Hover (a sci-fi "hover-bubble") changes the appearance of the landscape when it is seen from inside the vehicle.

★ See **Ships, Trains and Elevators** for larger conveyances

-
-  Start of Chapter 8: Vehicles, Animals and Furniture
 -  Back to Chapter 7: Other Characters: §7.16. Social Groups
 -  Onward to §8.2. Ships, Trains and Elevators
 -  Example 27: ★ **Peugeot** A journey from one room to another that requires the player to be on a vehicle.
 -  Example 104: ★ **No Relation** A car which must be turned on before it can be driven, and can only go to roads.
 -  Example 49: ★★ **Straw Boater** Using text properties that apply only to some things and are not defined for others.
 -  Example 29: ★★★ **Hover** Letting the player see a modified room description when he's viewing the place from inside a vehicle.
-

§8.2. Ships, Trains and Elevators

This section covers vehicles whose interior consists of at least one entire room. Moving into this room constitutes boarding: there is then some pause while it travels: moving out again disembarks at a different location. The only complication arising is how the player controls the journey (by talking to someone? by pressing buttons? by steering?).

The Unbuttoned Elevator Affair provides the simplest possible whole-room vehicle, which ferries between two fixed points. If you are at one of these, it must be the other one you want to get to, so there is no need for controls.

Dubai is a much more elaborate elevator, with many possible destinations, chosen using buttons inside the elevator.

Empire simulates a train journey. Here there are no controls as such, but the train passes through a sequence of stops spaced apart in time, so the player chooses an exit by getting out at the right moment.

On a very large, slowish craft such as a cruise liner, we are not so much travelling in a vehicle: it's more as if we are visiting a whole building, which becomes our world for the (probably long) duration of the journey. The liner steers around in long, slow curves, changing its orientation in the water, so that (if we think of "north" as a strictly magnetic matter, anyway) north is constantly rotating: something we don't notice on board because our own reference points, provided by the ship itself, stay fixed relative to ourselves. Because of this, some ships in IF are navigated using FORE, AFT, PORT and STARBOARD directions rather than NORTH, EAST, SOUTH and WEST: see **Fore**.

★ See **Bicycles, Cars and Boats** for smaller conveyances

-
-  Start of Chapter 8: Vehicles, Animals and Furniture
 -  Back to §8.1. Bicycles, Cars and Boats
 -  Onward to §8.3. Animals
 -  Example 9: ★ **The Unbuttoned Elevator Affair** A simple elevator connecting two floors which is operated simply by walking in and out, and has no buttons or fancy doors.
 -  Example 145: ★★ **Empire** A train which follows a schedule, stopping at a number of different locations.
 -  Example 42: ★★★ **Fore** Understand "fore", "aft", "port", and "starboard", but only when the player is on a vessel.
 -  Example 271: ★★★★ **Dubai** An elevator which connects any of 27 floors in a luxury hotel.
-

§8.3. Animals

Animals exhibit a wide range of behaviour: much of the chapter on Other Characters applies just as well to animals as to human beings, with the exception of the material on conversation. But two examples here, both fairly simple, show how a fairly convincing domestic pet can be achieved simply by reacting to certain events going on nearby: **Feline Behavior** (a cat) and **Today Tomorrow** (a dog).

Fido provides a dog which the player can re-name at will.

For animals that we can sit on and ride - a camel or a horse, say - we may want to use the Rideable Vehicles extension by Graham Nelson, which also provides a rideable animal kind.

★ See **Liveliness** for pets that change what they're doing every time the player looks

★ See **Bags, Bottles, Boxes and Safes** for a cat that eats food put in its container

-  Start of Chapter 8: Vehicles, Animals and Furniture
 -  Back to §8.2. Ships, Trains and Elevators
 -  Onward to §8.4. Furniture
 -  Example 406: **Feline Behavior** A cat which reacts to whatever items it has handy, returning the result of a rulebook for further processing.
 -  Example 420: **Fido** A dog the player can name and un-name at will.
 -  Example 99: **Today Tomorrow** A few notes on "In the presence of" and how it interacts with concealed objects.
-

§8.4. Furniture

Most domestic furniture consists of supporters and containers of one size or another. This means that the simplest furniture needs no elaborate instructions:

The candlestick is on the dining table. The dining table is fixed in place.

The silver salt cellar is on the serving trolley. The serving trolley is pushable between rooms.

The pillow is on the bed. The bed is enterable and fixed in place.

The examples below are therefore mostly ways to get around the usual restrictions on containers (that they only have one interior) and supporters (that they cannot simultaneously be containers as well).

Yolk of Gold provides a set of drawers, that is, a container with multiple interiors.

U-Stor-It provides a way to have containers with a lid which is also a supporter.

Swigmore provides a supporter which holds up the player, but has no top surface as such, and cannot hold up anything else. **Kiwi** demonstrates a kind of high shelf, whose objects cannot be seen or used unless the player stands on a ladder.

Princess and the Pea shows how a pile of supporters, each on top of the last, could be managed.

Tamed demonstrates furniture large enough to get inside, or on top of.

Circle of Misery demonstrates a conveyor belt, which can hold multiple items but only brings one of them within the player's reach at a time.

★ See **Position Within Rooms** for a box that can be positioned and used as a stepping stool

★ See **The Human Body** for letting the player take different postures on furniture or on the floor

★ See **Room Descriptions** for tables and other furniture whose content listing is suppressed or modified in a room description

★ See **Entering and Exiting, Sitting and Standing** for making the player automatically rise from a seat before leaving the room

★ See **Clocks and Scientific Instruments** for a grandfather clock

★ See **Kitchen and Bathroom** for a mirror the player can look into

↑ Start of Chapter 8: Vehicles, Animals and Furniture

← Back to §8.3. Animals

→ Onward to §8.5. Kitchen and Bathroom

↓ Example 13: ★ **Tamed** Examples of a container and a supporter that can be entered, as well as nested rooms.

↓ Example 173: ★ **Princess and the Pea** The player is unable to sleep on a mattress (or stack of mattresses) because the bottom one has something uncomfortable under it.

↓ Example 361: ★ **Kiwi** Creating a raised supporter kind whose contents the player can't see or take from the ground.

↓ Example 432: ★ **Circle of Misery** Retrieving items from an airport luggage carousel is such fun, how can we resist simulating it, using a list as a ring buffer?

↓ Example 83: ★★ **Yolk of Gold** Set of drawers where the item the player seeks is always in the last drawer he opens, regardless of the order of opening.

↓ Example 397: ★★ **Swigmore U.** Adding a new kind of supporter called a perch, where everything dropped lands on the floor.

↓ Example 58: ★★★ **U-Stor-It** A "chest" kind which consists of a container which has a lid as a supporter.

§8.5. Kitchen and Bathroom

Before implementing elaborate mechanisms to handle plumbing, we should pause to ask ourselves: how much of this do we need? Is it really necessary to simulate the complete set of fixtures and fittings?

This turns out to be a little tricky to do, and also rather dull to set out. The example **Modern Conveniences** was actually written as a demonstration of how an extension to Inform might be written to provide a general "kitchens and bathrooms service" for writers, but it contains a nice implementation well worth borrowing. The idea is to provide a "kitchen" kind of room and a "bathroom" kind of room. All kitchens created automatically contain standard kitchen appliances: fridge, freezer, sink with taps, counters, cabinets, and a stovetop with built-in oven. Similarly, all bathrooms will have sinks, baths, cabinets, and toilets, and respond to some standard interactions.

Another common feature of bathrooms is a mirror: **Versailles** demonstrates how to create a simple one.

- ↑ Start of Chapter 8: Vehicles, Animals and Furniture
 - ← Back to §8.4. Furniture
 - Onward to Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics: §9.1. Food
 - ↓ Example 75: **☆☆ Versailles** A mirror which will reflect some random object in the room.
 - ↓ Example 449: **☆☆ Modern Conveniences** Exemplifying the kind of source we might use in writing extensions for kitchen and bathroom appliances.
-

Examples from Chapter 8: Vehicles, Animals and Furniture

- ↑ Start of this chapter
- Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
- ↘ Indexes of the examples

27

☆☆ Example Peugeot

WI

A journey from one room to another that requires the player to be on a vehicle.

Let's say that our protagonist is about to flee . Obviously, he can't make the journey on foot; he needs transportation.

"Peugeot"

Include Rideable Vehicles by Graham Nelson.

The Lot is a room. The ten-speed bike is a rideable vehicle in the Lot.

We make the ten-speed bike a rideable vehicle because we want to say that the player is on it rather than in it. Then our other room:

Cambridge is east of the Lot.

And now we borrow from the Actions chapter to prevent travel without the proper equipment:

Instead of going to Cambridge when the player is not on the ten-speed bike:
say "It's a long journey to Cambridge: you'll never make it on foot."

After going to Cambridge:
say "You begin pedalling determinedly.";
continue the action.

Test me with "e / get on ten-speed bike / e".

★ Example No Relation

A car which must be turned on before it can be driven, and can only go to roads.

We need to designate certain rooms as roads. Since the status of being a road will not change during play, we do this with a kind:

"No Relation"

A road is a kind of room. Definition: a room is offroad if it is not a road.

Instead of going by a vehicle (called the auto) to somewhere offroad:
say "You can't drive [the auto] off-road."

Trafalgar Square is a road. "The Square is overlooked by a pillared statue of Admiral Lord Horatio Nelson (no relation), naval hero and convenience to pigeons since 1812."

The National Gallery is north of Trafalgar Square. The Strand is east of Trafalgar Square. The Strand is a road.

The car is a vehicle in Trafalgar Square. The ignition is a device. The ignition is part of the car. Instead of going by the car when the ignition is switched off: say "The ignition is off at the moment." Instead of switching on the car, try switching on the ignition. Instead of switching off the car, try switching off the ignition.

Test me with "get in car / n / e / turn on car / n / e / get out / w / n / s / e / get in car / turn off car / w / turn on ignition / w".

(In the course of the writing of Inform 7, much of Trafalgar Square was pedestrianised, making this example already out of date.)

A further technical note: notice "going by a vehicle" in the above rule, rather than "going by something". A rule such as "Instead of going by something..." will be matched whenever the player tries to go some direction while in an enterable object, whether or not that object is actually capable of movement. This is sometimes useful, but in this case we want the warning to apply only when the player is in a vehicle; if we added Trafalgar Square's statue bases to the scenario, we would not want

You can't drive the pedestal off-road.

So we restrict the rule to "Instead of going by a vehicle..."

★★ Example Straw Boater

Using text properties that apply only to some things and are not defined for others.

Sometimes we like to give properties to kinds of thing, but not fill them in in all cases. For instance, we might have vehicles that optionally make noise, and those might have a "movement sound".

All properties have a default value, which we can find by looking in the Kinds tab of the index. This is what the property will be set to automatically, if we do not change it ourselves. In the case of a text property, that is ""; so for instance we might use our movement sound thus:

"Straw Boater"

Boathouse is a room. "A boathouse circa 1915, which -- though in poor repair -- still suggests Sunday afternoon jaunts taken by women in white gowns and men in straw hats."

North of the Boathouse is the Shallow Water. The description of Shallow Water is "Just south is the boathouse, and beyond it are trees and the marble terrace of the house above. The water deepens to the north."

North of Shallow Water is Deep Water. The description of Deep Water is "From here the boathouse has dwindled invisibly to the south, and you have a broad panorama of the shoreline, all the way down to the Skeleton Point Lighthouse in the southeast."

A vehicle has some text called the movement sound. The sailboat and the motorboat are vehicles in the Boathouse. The movement sound of the motorboat is "VRRRROOOMMM..." Understand "boat" as the sailboat. Understand "boat" as the motorboat.

Note that we haven't given the sailboat any movement sound at all.

After going somewhere by a vehicle (called cart):
if the movement sound of the cart is not "", say "[the movement sound of the cart][paragraph break]";
continue the action.

Instead of exiting when the player is in a vehicle and the location is not the Boathouse:
say "You're not dressed for a swim."

Instead of going somewhere when the player is not in a vehicle:
say "You'd rather not try to make this journey by swimming alone."

Test me with "n / get in sailboat / n / get out / s / get in motorboat / n / n".



Suppose we want the player to see a modified room description when he's viewing the place from inside a vehicle. There are several conceivable ways of doing this; the example here shows a rather advanced way, but is very flexible and will let us write all sorts of special cases.

"Hover"

Use full-length room descriptions.

Emerald City is a room. "All the buildings are spires and none of them have doors." The Vast Desert is west of Emerald City. "[if the player is in a vehicle]Outside, a[otherwise]A[end if] trackless waste stretches as far as the eye can see in every direction."

The hover-bubble is a vehicle in the Emerald City. "Your hover-bubble awaits." The description is "The hover-bubble is a clear globe-shaped vehicle capable of transporting you anywhere you could walk, but faster." Understand "bubble" as the hover-bubble. The hover-bubble contains a chocolate wrapper and a parking ticket.

Here's the tricky part, which relies on material from the chapters on Activities and Rulebooks:

The container interior rule is listed before the room description body text rule in the carry out looking rules.

This is the container interior rule:

if the actor is the player and the player is in an enterable thing (called current cage), carry out the describing the interior activity with the current cage.

Describing the interior of something is an activity.

Now we've done that, we can write a "rule for describing the interior" of something, which will print whatever we like:

Rule for describing the interior of the hover-bubble:

say "The hover-bubble is transparent, but tints everything outside very faintly lavender."

In fact, as a special refinement, we could even say:

Rule for describing the interior of the hover-bubble when the hover-bubble contains more than one thing:

say "The hover-bubble is transparent, but tints everything outside very faintly lavender. Beside you you can see [a list of other things in the hover-bubble]."

Definition: a thing is other if it is not the player.

Rule for listing nondescript items of the hover-bubble when the player is in the hover-bubble: do nothing.

Test me with "get in bubble / look / west / take all / look / get out / east".

And now anything that's beside us in the vehicle will be described during that first paragraph, rather than later on.

9 **★ Example The Unbuttoned Elevator Affair** WI

A simple elevator connecting two floors which is operated simply by walking in and out, and has no buttons or fancy doors.

This is very simple. The interior of the elevator is a single room, but which is mapped east of both of its termini. The reverse map connection, west from the elevator, can only go to a single room, and that's what determines which floor the elevator is on.

"The Unbuttoned Elevator Affair"

UNCLE Headquarters is a room. "The steel nerve-center of the free world's battle against the Technological Hierarchy for the Removal of Undesirables and the Subjugation of Humanity. Being against technology, we have only a very simple elevator to the east."

Del Floria's Tailor Shop is a room. "Only trained anti-THRUSH agents recognise the booth in the east wall as a secret elevator."

The Secret Elevator is east of UNCLE Headquarters. The Secret Elevator is east of Del Floria's Tailor Shop.

After going to the Secret Elevator:

say "The doors automatically close, there is a rush of motion, and they open again.";

if UNCLE Headquarters is mapped west of the Secret Elevator, now Del Floria's Tailor Shop is mapped west of the Secret Elevator;

otherwise now UNCLE Headquarters is mapped west of the Secret Elevator;
continue the action.

Test me with "east / west / east / west".

145

★★ Example Empire WI

A train which follows a schedule, stopping at a number of different locations.

Suppose we want to have a train which, at fixed times, arrives at and leaves stations. It should be possible for the player to get on and off the train when it is stopped, but not while the train is in motion.

"Empire"

The Empire Builder Train is a room. The Train has a room called the station. The station of the Train is Seattle.

The description of the Empire Builder is "One of the (relatively) plush long-distance Amtrak trains. You're in a two-story car with toilets and a cafe at one end, not having sprung for a sleeper.

[if the station of the Train is the train]Outside the window there is rapidly-passing countryside.[otherwise]Through the windows you can see the [station of the Train] train station.[end if]"

Instead of exiting when the player is in the Train:
if the station of the Train is the Train:
say "The train is not stopped at a station." instead;
otherwise:
move the player to the station of the train instead.

Before going outside when the player is in the Train:
try exiting instead.

Before going inside when the player is in the station of the Train:
move the player to the Train instead.

Seattle, Edmonds, Everett, Wenatchee, and Spokane are rooms. The description of a room is usually "The scenic train station of [the location][if the location is the station of the train].

The pompously-titled Empire Builder train is pulled up here, soon to continue its journey towards Chicago[end if]."

And now our schedule for the train -- somewhat truncated, admittedly, since the full three-day journey from Seattle to Chicago is a bit long even for an ambitious example.

At 4:45 PM:
if the player is in the train or the player is in the station of the train, say "The train pulls out of [the station of the Train]!";
now the station of the Train is the Train.

At 5:10 PM:
now the station of the Train is Edmonds;
if the player is in the train or the player is in the station of the train, say "The train pulls into Edmonds and comes to a stop."

At 5:17 PM:
if the player is in the train or the player is in the station of the train, say "The train pulls out of [the station of the Train], running north along the shore towards Everett.";
now the station of the Train is the Train.

At 5:39 PM:
now the station of the Train is Everett;
if the player is in the train or the player is in the station of the train, say "The

The port is a direction. The port has opposite starboard. Understand "p" as port when the location is nautical.

The fore is a direction. The fore has opposite aft. Understand "f" as fore when the location is nautical.

The aft is a direction. The aft has opposite fore. Understand "a" as aft when the location is nautical.

Does the player mean going a nautical direction when the location is nautical: it is very likely.

Index map with fore mapped as north. Index map with aft mapped as south.
Index map with port mapped as west. Index map with starboard mapped as east.

And we can even add new ways to talk about the ways things are mapped, borrowing from the Relations chapter. The following will allow us to use "is abaft of" as well as "is aft of":

[The verb to be abaft of means the mapping aft relation.]

Now, to prevent the player from using NORTH onboard ship, or AFT on land:

A room can be nautical or earthbound. A room is usually not nautical. A direction can be nautical or earthbound. A direction is usually not nautical. Starboard, port, fore, aft, up, down, the inside and the outside are nautical.

Before going a nautical direction when the location is not nautical, say "Nautical directions can only be used on board ship."

Before going an earthbound direction when the location is nautical, say "Compass directions make no sense on board ship, but you can use [list of nautical directions] instead." instead.

Section 2 - Scenario

The Fish Room is aft of the Spirit Room. Starboard of the Fish Room is the After Powder Magazine. The Bread Room is aft of the After Powder Magazine.

The Fish Room, the Spirit Room, the Bread Room, and the After Powder Magazine are nautical.

The description of the Fish Room is "Absurd quantities of salt fish are kept here, and periodically visited by the cook or someone serving him. It is otherwise an unexceptional little chamber, so far below the waterline that there are no portholes and no external light of any kind. [paragraph break]A narrow doorway leads forward into the Spirit Room, and the After Powder Magazine is starboard."

The description of the Spirit Room is "Despite its ghostly name, this is little more than a closet down at the very navel of the ship, in which alcohol is kept: both for purifying wounds and for drinking. Under normal circumstances there is a guard posted here at every hour, lest anyone take to raiding the larder. The current

absence of the guard marine strikes you as a very bad sign indeed. [paragraph break]The only way out is aft."

The description of the Bread Room is "The Bread Room is not only tiny from side to side and front to back: it is also about half the height of a proper room, and the floor slopes up very steeply with the curve of the hull. [paragraph break]What is kept here would not, on land, be dignified by the name of bread: it's hard tack, punishing to the teeth, dry on the tongue, and usually a home to weevils before half the journey is done. [paragraph break]More headroom, and access to the rest of the ship, lies fore through the After Powder Magazine."

The description of the After Powder Magazine is "Kept in near darkness because no one with any sense would bring a naked flame down here: when necessary, it can be lit with a single small lantern made of very thick glass and sealed to keep the sparks within. Sacks of powder are passed up into the higher levels of the ship by the scrubby little boys called 'powder monkeys' -- but none such are here now."

Test me with "north / aft / fore".

271



Example Dubai

WI

An elevator which connects any of 27 floors in a luxury hotel.

The problem of implementing an elevator that opens onto a large number of floors often challenges novice interactive fiction authors. It also raises a fundamental design problem -- how to implement a large building in an interesting way. It is tempting to write a hotel with an elevator and innumerable tedious and identical floors just for the sake of realism; in many cases it is a better idea simply to omit any locations that contribute nothing to the story or the play of the game.

In charity, though, let us assume that the author has a legitimate reason for wanting to implement an elevator that opens onto some generic floors. We will go whole-hog, and set this in the world's tallest hotel: Burj al-Arab, Dubai.

"Dubai"

The Burj al-Arab Lobby is a room. "The 202-suite Burj al-Arab - or Tower of the Arabs - stands 321 metres (1,060 feet) high, and floats on its own man-made island. It is shaped like the sail of a boat; just crossing the private bridge to reach this place set you back \$55."

(Since our budget did not run to visiting Burj al-Arab, the descriptions place implicit faith in the hotel's website.)

The Assawan Spa is a room. "Treatment rooms, hydrotherapy baths, oriental massage, stand-up solarium, sauna, steam rooms and jacuzzi, two swimming pools, squash court, two fully equipped fitness studios and an aerobics floor. To the south is a shopping area, for those who do not find exercise adequately therapeutic."

The Shopping Area is south of Assawan Spa. "In a setting that would make a poet sigh, you may enjoy the services of (among others) Bulgari, Black Pearl Caviar, Albarajeel Carpet Shop, Abdul Samed Al Qurashi (amber and Arabic perfumes), Dianoor (jewellery), and the Commercial Bank of Dubai."

The Al Falak Ballroom is a room. "A palatial, two-tiered, domed ballroom crowned with a unique crystal chandelier."

Sahn Eddar is a room. "At the base of the world's tallest atrium, the Sahn Eddar restaurant offers light fine fare and Afternoon Tea. At the center, a 32-meter water column leaps toward the roof of the atrium."

Al Mahara is a room. "After the elevator, you must take a three-minute virtual submarine voyage to reach this seafood restaurant. A magnificent oval aquarium, full of sharks, is visible from every table."

Al Iwan is a room. "Middle-eastern food in an environment of dramatic black, red, and gold."

Al Muntaha is a room. "A top-floor restaurant with a magnificent view: the name means the ultimate or the highest, suitable for a place that stands at 200 metres above the Arabian Gulf. It offers modern European cuisine; and just off to the south is the Skyview Bar."

The Skyview Bar is south of Al Muntaha. "'A wonderful location for pre- and post dinner drinks,' claims the hotel's brochure, and certainly you can't fault the view."

The Juna Lounge is a room. "Two humidors that offer one of the largest selections of the finest Havana cigars in Dubai."

The Lift is a room. "This is not a mere elevator: it is an express panoramic lift, traveling six meters a second, and capable of taking you from lobby to the rooftop restaurant in an astonishingly short time. The illuminated number above the door says [current level of the Lift] - though you can make it move by pressing a numbered button."

The Presidential Suite is a room. "Astonishingly, this is not the most elaborate or expensive of suites; there is another, the Royal, upstairs of here."

The Royal Suite is a room. "The brochure described this as 'the last word in luxury,' and you have to admit that it is certainly the last word in [italic type]something[roman type]. A vast carpet of patterned red and gold stretches from you to the sofa; beyond which, in the distance, you make out several bedrooms and bathrooms outfitted in Carrera marble. There is also, of course, a private cinema."

Table of Floors

level	floor
0	Al Mahara
1	Burj al-Arab Lobby
2	Al Iwan
3	Juna Lounge
4	Sahn Eddar
15	Al Falak Ballroom
18	Assawan Spa
24	Presidential Suite

The elevator exterior is a backdrop. It is not scenery. The initial appearance of the elevator exterior is "You can enter the elevator here." It is in Generic Floor, Al Mahara, the Lobby, Al Iwan, Juna Lounge, Sahn Eddar, Al Falak, Assawan Spa, Presidential Suite, Royal Suite, and Al Muntaha.

Before entering the elevator exterior, try going inside instead.

Instead of going inside in the presence of the elevator exterior:

- if there is a level corresponding to a floor of the location in the Table of Floors:
 - let the present level be the level corresponding to a floor of the location in the Table of Floors;
 - now the current level of the Lift is the present level;
- otherwise:
 - now the current level of the Lift is the current level of Generic Floor;
 - move the player to the Lift.

The Lift has a number called current level. The current level of the Lift is 1. Instead of going up in the Lift: say "You'll have to select a specific floor; your options range from 0 to 27." Instead of going down in the Lift: try going up instead. The Lobby is outside from the Lift.

Before going outside in the Lift:

- if there is a floor corresponding to a level of the current level of the Lift in the Table of Floors:
 - let the other place be the floor corresponding to a level of the current level of the Lift in the Table of Floors;
 - move the player to the other place instead;
- otherwise:
 - now the current level of the Generic Floor is the current level of the Lift;
 - move the player to the Generic Floor instead.

The Generic Floor is a room. The Generic Floor has a number called current level. The printed name of the Generic Floor is "Floor [current level of the Generic Floor]". "A long hallway between suites, some of which run up to \$15,000 a night."

Understand "push [number]" as pressing button. Understand "push [number] button" as pressing button. Understand "push button [number]" as pressing button. Pressing button is an action applying to one number.

Check pressing button:

- if the player is not in the Lift, say "You cannot control the express panoramic lift unless you are yourself inside." instead;
- if the number understood is the current level of the Lift, say "The lift pings politely and reopens its doors, since you are already on floor [number understood]." instead;
- if the number understood is greater than 27, say "There are only 27 floors." instead;
- if the number understood is less than 0, say "You cannot go below the ground floor in this elevator." instead.

Carry out pressing button:

- now the current level of the Lift is the number understood;

say "You press button [the number understood]. The lift whirs into action and moves to the correct level."

Test me with "press 3 / in / press button 3 / look / out / in / press 27 / out / s / in / n / in / press 15 / out / in / press 18 / out / s / in / n / in / press 4 button / out".

This will all work very well, unless the player has portable objects; in that case, anything he drops on the Generic Floor will be there every time he goes back -- whether it's masquerading as Floor 6 or Floor 23. There are there are two ways round this -- (i) the cheeky way. When we drop something, the unobtrusive yet ever-vigilant maids pick it up and take it down to the Foyer's lost property office; and (ii) the super-duper way, in which things are moved out of play but with their floor numbers remembered, so that the scenario can be reconstructed each time. (i) is probably in fact the more true-to-life, considering the hotel's boasts about its service, but we will demonstrate both methods.

Here is the version with vigilant maids:

The player carries a shopping bag. In the bag are a diamond necklace, a small rug, and a jar of caviar.

Carry out pressing button:

now every portable thing enclosed by the Generic Floor is in the Office.

The Office is south of the Lobby. "Here the maids collect everything abandoned by careless guests." The printed name of the Office is "Lost and Found Office".

Test maids with "in / press 6 / out / drop all / in / out / in / press 23 / out / in / press 1 / out / s / get all".

Notice that we tie the maid service to the pressing of the lift button, so that if the player just goes into the lift and comes out again, the maids will not have had a chance to clear his possessions.

Alternatively:

The player carries a shopping bag. In the bag are an evening gown, a bolero jacket, and tickets to the Wild Wadi Animal Park.

Carry out pressing button when something is in the Generic Floor:

repeat with item running through portable things in the Generic Floor:

clear the item;

repeat with item running through portable things enclosed by the Generic

Floor:

clear the item.

The "enclosed by" line clears even things left on, say, small un-portable side-tables or whatever; but because we do "in" first, we make sure to move any containers or supporters undisturbed. The next bit could be more tidily incorporated into our previous "before going outside" rule, but since we are writing this code to be optionally pasted onto the end of the first bit, we'll express the rule separately:

Before going outside in the Lift when something is in Limbo:
 unless there is a floor corresponding to a level of the current level of the Lift in
 the Table of Floors:
 repeat with item running through things in Limbo:
 if the current level of the item is the current level of the Lift, move the
 item to the Generic Floor.

A thing has a number called current level.

To clear (item - a thing):
 now the current level of the item is the current level of the Lift;
 move item to Limbo.

Limbo is a room.

Test management with "get all from bag / in / press 22 / out / drop tickets / in /
press 23 / out / drop gown / in / press 22 / out / get tickets / in / press 23 / out /
get gown".

And now we have a situation in which the player's valuables are left untouched
wherever in the hotel he happens to abandon them.

Incidentally, this example was almost set in an entirely different location: the largest
hotel in the world may some day be the Ryugyong Hotel in Pyongyang, North Korea,
with its 105 floors -- but for some years construction halted at the creation of the
building's huge concrete shell.

406

★ Example Feline Behavior

WI

A cat which reacts to whatever items it has handy, returning the result
of a rulebook for further processing.

Suppose we have a cat which is supposed to react to (and destroy) the most
interesting thing in its environment. There are several ways we could approach this
problem, but for the sake of demonstration, let's have it follow a rulebook to figure
out which thing it most wants to interact with. We will then return the chosen object
as "the object produced by the cat behavior rules".

"Feline Behavior"

The Kitchen is a room. The cat is an animal in the Kitchen. In the Kitchen is a
bowl, a ball of wool, a newspaper. The bowl contains a quantity of cream.

The cat is wearing a silver collar. The description of the cat is "It is wearing [a list
of things worn by the cat]."

The player carries a closed openable container called a bag. The bag contains
catnip.

The cat behavior rules is a rulebook producing an object.

A cat behavior rule when the cat can touch the catnip:
say "The cat frolics with the catnip until nothing remains of it.";
rule succeeds with result catnip.

A cat behavior rule when the cat can touch the cream:
say "The cat laps up the cream.";
rule succeeds with result cream.

A cat behavior rule when the cat can touch the ball of wool:
say "The cat makes the ball of wool into a useless tangle which must be
discarded.";
rule succeeds with result ball.

A cat behavior rule when the cat can touch the newspaper:
say "The cat bats playfully at the newspaper until all the nasty boring articles
are destroyed.";
rule succeeds with result newspaper.

A cat behavior rule:
say "The cat looks miffed at the lack of ready entertainment, and glares at you
with yellow eyes as though wondering whether your pants leg is good for claw-
sharpening.";
rule fails.

Every turn:
let the destroyed object be the object produced by the cat behavior rules;
if the destroyed object is not nothing:
now the destroyed object is nowhere;
say "[line break]Good thing you have no use for [the destroyed object]
yourself.[paragraph break]".

Test me with "z / z / open bag / z / z".

We include the if rule succeeded... condition here because nothing will be returned if
the cat's search failed (as for instance in the result of the final rule).

Naturally, if we wanted we could equally well ask "if rule failed...".

420

★ Example Fido

WI

A dog the player can name and un-name at will.

Suppose we'd like to have a dog which the player is allowed to name himself. We'd
like to deal correctly with both

>name the dog fido

and

>name the dog "fido"

so we'll also need to strip quotation marks out of the command. We can do this as follows:

"Fido"

The Back Yard is a room.

A dog is an animal in Back Yard. The dog has some text called the nickname. The nickname of the dog is "nothing". Understand the nickname property as describing the dog.

Rule for printing the name of the dog when the nickname of the dog is not "nothing":

say "[nickname of the dog]"

Naming it with is an action applying to one thing and one topic. Understand "name [something] [text]" as naming it with. Check naming it with: say "You can't name that."

Instead of naming the dog with "nothing":

now the nickname of the dog is "nothing";
now the dog is improper-named;
say "You revoke your choice of dog-name."

Instead of naming the dog with something:

let N be "[the topic understood]";
replace the text "" in N with "";
now the nickname of the dog is "[N]";
now the dog is proper-named;
say "The dog is now known as [nickname of the dog]."

Test me with "name the dog Fido / name the dog Lawrence / look / x lawrence / name Lawrence nothing / look / x lawrence".

99



Example Today Tomorrow

WI

A few notes on "In the presence of" and how it interacts with concealed objects.

Rules about concealment will affect "in the presence of", too. For instance, suppose we have a man with a pocket pet:

"Today Tomorrow"

The Temporary Employment Office is a room. "Tomorrow's Temporary Workers - Today!" proclaims the logo over the door. The office is divided into two areas, the inner sanctum where you take calls and fiddle with the computer, and the outer area where workers take skill exams and watch inspirational videos ('Earn your way to partial benefits!', 'Vacation days and you!', 'Temping the Tomorrow Way', etc.)."

Maya is a woman in the Office. "Your coworker Maya sits at her own computer, diligently modifying all the [one of]pay rates in the database down from \$9.00 an hour to \$8.96[or]billing rates in the database up from \$25.00 an hour to \$25.04[purely at random]." She wears a trenchcoat. She carries a chihuahua. The description of Maya is "[if Maya is wearing the trenchcoat]She's wearing a trenchcoat, in a bizarre bid to keep your boss unaware of the chihuahua in her pocket. 'Because petsitters are really expensive!' she explained to you in an urgent hiss this morning over the coffeemaker. [otherwise]She looks cold. [end if]To all appearances, she is carrying [a list of unconcealed things carried by Maya]."

Maya's computer is scenery in the Employment Office. The description is "You can't see the screen from here, but she was perfectly happy to tell you what she was doing."

Rule for deciding the concealed possessions of someone (called carrier):
if the particular possession is the chihuahua and the carrier wears the trenchcoat, yes;
otherwise no.

Instead of eating something in the presence of the chihuahua:
say "[The chihuahua] yips at you! Maya looks despairingly at [the noun], which is obviously inciting it."

The desk is scenery in the Office. On the desk are a multi-line telephone and a printer. The description of the printer is "Every morning, this instrument of torture spits out a list of the assignments you have to fill - professional, attractive receptionist with level three Excel certification, at \$7.05 an hour; that sort of thing. You spend the ensuing three hours trying to meet its tyrannical demands." The description of the multi-line telephone is "Line three blinks urgently at you. You'll pick it up again as soon as you can remember who it was you put on hold."

The player carries a double bacon cheeseburger and a chocolate shake. Understand "milkshake" or "beverage" as the shake. The description of the cheeseburger is "A minor self-indulgence to make up for the fact that you have to work through lunch." The cheeseburger is edible. The shake is edible. The description of the shake is "It has the consistency of spackle and is no longer cold, but there is some chocolatey goodness in it still." Understand "burger" or "hamburger" as the cheeseburger. Instead of drinking the chocolate shake: try eating the shake instead.

Test me with "x maya / x cheeseburger / drink shake / eat cheeseburger".

Unless we somehow get the trenchcoat away from Maya, the chihuahua will not be in view, and will not intervene in our lunch. All very well for the player character, but not so interesting to the story... To this end, we might add an unfortunate event, courtesy of later chapters:

The time of day is 11:45 AM.

At 11:47 AM: say "Your boss pokes his head in, temporarily free of the round of conference calls that occupy all his days. 'Maya,' he says. 'Your coat?' He shakes his head, clucking sadly. 'It doesn't say professional!' But mercifully Maya manages to take it off so slowly that he doesn't glimpse her pet before her phone

rings again.";
now Maya carries the trenchcoat.

13

Example Tamed

WI

Examples of a container and a supporter that can be entered, as well as nested rooms.

Within a room, we might have containers and supporters that a player can enter. A chair, stool, table, dais, or pedestal would be an enterable supporter (anything we would describe a person as being "on"); a cage, hammock, or booth would be an enterable container (because we would describe the person as being "inside").

When the player is in or on something, he is able to see the rest of the contents of the room, but a note such as "(in the hammock)" or "(on the poster bed)" is added to the room title when he looks around.

Here is an example to show off the possibilities:

"Tamed"

The Center Ring is a room. The cage is in the Center Ring. A lion is an animal in the cage. The cage is enterable, openable, transparent, and closed.

Notice that we made the cage transparent. Strictly speaking it is not made of transparent materials, but we can see into (or out of) a closed cage due to the gaps between the bars, so that from Inform's point of view a cage behaves much like a large sturdy glass box. (If we really wanted to make a distinction between, say, an airtight container and one with perforations, we could do so, but Inform does not model such nuances by default.) If a container is not transparent, we can see into and out of it only when it is open.

Supporters are a bit more straightforward because there is no circumstance in which they separate the player from the rest of the world:

The pedestal is in the Center Ring. It is enterable.

And in fact we can tell Inform that the player starts on the pedestal with this line:

The player is on a pedestal.

Now the player will begin there rather than just in the Center Ring.

This last bit is an entirely unnecessary bit of local color, but if we're going to keep getting into and out of the lion's cage, we ought to expect him to take notice:

Every turn when the player is in the cage:
if a random chance of 1 in 2 succeeds, say "The lion eyes you with obvious discontent.";

otherwise say "Though the lion does not move, you are aware that it is watching you closely."

Randomness is explained more completely in the chapter on Change, and every turn rules in the chapter on Time.

Finally, we might want a container whose interior is modeled as its own separate room: say, a magician's booth in which volunteers are made to disappear.

The magician's booth is a container in Center Ring. "Off to one side is a magician's booth, used in disappearing acts. The exterior is covered with painted gilt stars." The booth is enterable, open, not openable, and fixed in place.

Now we create our other location:

Inside from the Center Ring is the Starry Vastness.

...which handles the case of the player typing >IN. (We will not assume by default that he wants to get into the cage with the lion, this being obviously perilous.) But we also want to make sure that the player who types >ENTER BOOTH winds up in the same place, so we should add:

Instead of entering the magician's booth: try going inside.

Test me with "get in cage / open cage / get in cage / z / close cage / out / open cage / get on pedestal / get off / look / enter booth / out".

173

★ Example Princess and the Pea

WI

The player is unable to sleep on a mattress (or stack of mattresses) because the bottom one has something uncomfortable under it.

The main point here is that we need to figure out where the stack meets the floor:

"Princess and the Pea"

The Topmost Turret is a room. A mattress is a kind of supporter. A mattress is always enterable. A mattress is portable.

A large mattress is a mattress in the Turret. A medium mattress is a mattress in the Turret. A small mattress is a mattress in the Turret.

Instead of sleeping when the player is on a mattress (called the bed):

let the item be the bed;

while the holder of the item is not a room:

let the item be the holder of the item;

say "You can still feel something very uncomfortable under [the item]."

Instead of sleeping:

say "You can't sleep standing up!"

Instead of looking under a mattress, say "You scout around, but are unable to determine what's causing you this discomfort. If only your maid Winnie were here. She's very good at this."

Test me with "sleep / enter small / sleep / get up / get small / put small on medium / get on small / sleep / get up / g / get medium / put medium on large / get on small / look / sleep".

361

★ Example Kiwi

WI

Creating a raised supporter kind whose contents the player can't see or take from the ground.

Suppose we want there to be some high shelves in our game, which the player can't get at unless he's standing on a prop of some kind. (This is a pretty hoary and over-used puzzle, but there may still be occasions when it becomes useful again.)

In order to resolve this, we want to set up a raised supporter kind. When something is on a raised supporter, it should be mentioned to the player only if the player is in the right position (i.e., standing on something) and otherwise omitted from the description entirely.

"Kiwi"

Section 1 - Procedure

A raised supporter is a kind of supporter.

For printing a locale paragraph about a raised supporter (called the high place):
if the player is on a supporter (called the riser):
 say "Up on [the high place] (and only visible because you're on [the riser])
[is-are a list of things on the high place].";
otherwise:
 say "The [high place] is above you."

Note that here we don't continue the activity because we want to completely replace the normal behavior of describing what is on supporters.

Definition: a thing (called target item) is out of reach:
if the player is on a supporter, no;
if the target item is on a raised supporter, yes;
no.

Now we also need to prevent the player from interacting with things that are out of reach:

Before doing something:
if the noun is out of reach or the second noun is out of reach:
 say "You can't reach from down here." instead.

...or restoring things to the shelves while the player is in the wrong position...

Instead of putting something on a raised supporter when the player is not on a supporter:

say "You can't reach from down here."

And raised supporters shouldn't be searchable from the ground either:

Instead of searching or examining a raised supporter when the player is not on a supporter:

say "You can't see from down here."

Finally, we need to tackle the case where the player types GET ALL FROM SHELF, because we don't want to list the objects up there if the player can't even see them. We use a rule for deciding whether all includes in order to tell Inform not to consider items that can't be reached, and then we adjust the parser error so that it's a little more instructive than "There are none at all available!", which is what the response would otherwise be:

Disallowed-all is a truth state that varies. Disallowed-all is false.

Rule for deciding whether all includes an out of reach thing:

now disallowed-all is true;
it does not.

Rule for printing a parser error when the latest parser error is the nothing to do error and the player is not on a supporter:

if disallowed-all is true:
say "Whatever might be up there, you can't see or reach from down here.";
otherwise:
make no decision.

A first action-processing rule:

now disallowed-all is false.

Section 2 - Scenario

The Bottom of the Nursery is a room. "Ever since you ate that mysterious cake, you've been even shorter than usual."

The high shelf is a raised supporter in the Nursery. It is scenery. On the high shelf are a kiwi-green ball and a stuffed dodo.

The step-ladder is an enterable supporter in the Nursery. Understand "ladder" as the step-ladder.

Test me with "x shelf / search shelf / get dodo / get all from shelf / stand on ladder / get all from shelf / search shelf / get off / put all on shelf / get all from shelf / stand on ladder / put all on shelf".

The only point we need to be careful about is that the carousel is simulated twice over, in the following text: once in the built-in way that objects are inside other objects, so that the luggage items are objects contained in the carousel object; but then again by the "circle of misery" list, a ring buffer keeping track of what order things are in. We need to be careful that these two records do not fall out of synchrony: anything put into the carousel must be added to the list, anything taken out must be removed. (In fact we forbid things to be added, for simplicity's sake.)

"Circle Of Misery"

Luggage item is a kind of thing. The blue golf bag, the leopardskin suitcase, the green rucksack and the Lufthansa shoulder-bag are luggage items.

Heathrow Baggage Claim is a room. The carousel is a container in Heathrow. "The luggage carousel, a scaly rubbered ring, does for the roundabout what Heathrow Airport does for the dream of flight: that is, turns the purest magic into the essence of boredom, only with extra stress. [if the number of entries in the circle of misery is 0]For once it stands idle. Perhaps it's broken.[otherwise]The baggage approaching you now: [the circle of misery with indefinite articles]."

The circle of misery is a list of objects that varies.

When play begins:

now all the luggage items are in the carousel;
add the list of luggage items to the circle of misery.

The list "circle of misery" is our ring, in which entry 1 is considered to be the position of whichever bag is currently frontmost. And here it goes, round and round:

Every turn when the number of entries in the circle of misery is not 0:
rotate the circle of misery;
let the bag be entry 1 of the circle of misery;
say "The carousel trundles on, bringing [a bag] to within reach."

After taking a luggage item (called the bag):
remove the bag from the circle of misery, if present;
say "Taken."

Before doing something with a luggage item (called the bag) which is in the carousel:
if the bag is not entry 1 of the circle of misery, say "[The bag] is maddeningly out of range. You'll have to wait for it to come round." instead.

Instead of inserting something into the carousel, say "In recent years, the authorities have tended to frown upon depositing bags in random places at airports."

Test me with "get suitcase / get suitcase / get suitcase / get suitcase / look / get golf bag / look / get golf bag".



Example **Yolk of Gold**

Set of drawers where the item the player seeks is always in the last drawer he opens, regardless of the order of opening.

Suppose that for dramatic effect we would like the player to find the thing he is looking for always in the last drawer he opens...

"Yolk of Gold"

The Turret is a room. "A cramped little room distinguished chiefly by the spiral staircase that descends from it. The windows look out over the rooftop."

The Rooftop is outside from the Turret.

The staircase is an open unopenable door. It is scenery. It is below the Turret and above the Library. The description is "A winding stair carved out of the single trunk of a massive tree, all in a dark wood; the outside of the stairs left unfinished with the bark still on, but the treads worn smooth by long and constant use."

The description of the Library is "Here, it seems, you have found your mark: books line both walls, a dark carpet lies on the floor, and a strange dress hangs up in a presentation case. And the thing you were told to look for, a desk with three drawers.

A spiral staircase leads up and out."

The cherry desk is scenery in the library. The description is "A deep, satin-lustrous cherry, with scrollwork legs and gilt touches. The years have not been kind, and it has cracked and split in several places; the finish is damaged, and where there is inlaid mother of pearl, it is beginning to come up from its bed. But it is still a sound piece, and features three drawers."

After examining the desk for the first time, say "(Your employers were able to tell you to look for it, but not which drawer to look in. Typical.)"

A drawer is a kind of container. A drawer is always openable and closed. The description of a drawer is "The usual drawer of heavy wood, inadequately smoothed for ease of use."

The top drawer is a drawer. The middle drawer is a drawer. The bottom drawer is a drawer. The top drawer, the middle drawer, and the bottom drawer are part of the desk. A drawer can be explored or unexplored. A drawer is usually unexplored. Instead of searching a closed drawer, try opening the noun.

After opening a drawer when no drawers are explored:

now the noun is explored;
say "There is a tremendous screech, but nothing whatsoever inside, not even dust."

After opening an unexplored drawer when exactly one drawer is explored:
now the noun is explored;
say "You struggle to open this one a bit more quietly, conscious all the time of noise... but no, it's empty. Just stands to reason."

After closing a drawer:
say "There is a racket of wood protesting against wood as you do so, which makes you wonder if you hadn't better just leave them open from now on. The unhappy owner is probably going to catch on soon enough anyway."

Before opening an unexplored drawer when exactly two drawers are explored:
move the mysterious thing to the noun;
now the noun is explored.

There is a mysterious thing. The description is "A very familiar-looking hemispherical weight of metal, goldish in tone though perhaps not made of gold. This one has a slight stickiness about the bottom surface."

Instead of opening the desk when at least one drawer is unexplored:
let the next drawer be a random unexplored drawer;
say "(starting with [the next drawer])[line break]";
try opening the next drawer.

Instead of looking under the desk when no drawers are explored:
say "You carefully survey the ground around the desk. There don't seem to be any hidden tripwires or traps to prevent you from having a look in the drawers."

Instead of looking under the desk when at least one drawer is unexplored:
say "Nothing there either. Thoroughness is a virtue with tedium as a side effect, as your mother used to say -- but they always counted her a trifle slapdash."

Instead of looking under the desk when all drawers are explored:
say "There's nothing down there, but this doesn't come as a great surprise."

Instead of searching the desk when at least one drawer is unexplored:
move the mysterious thing to the player;
say "You perform a hasty, squeaky search of ";
if no drawer is explored, say "all three drawers, discovering nothing and nothing in rapid succession. But on the third drawer you ";
if exactly two drawers are unexplored, say "the remaining two drawers. There's nothing in the second, but in the third you ";
if exactly one drawer is unexplored, say "the last drawer. In it, you ";
say "turn up a promising hemispherical object.";
now all the drawers are explored.

Perhaps, just for fun, we have all the other scenery draw the player's attention back to the main point, as well.

The carpet is scenery in the Library. The description is "It is too dark for you to make out details of the antique design, which seems dimly to represent an early voyage to the moon, with ships and the travelling stars." Understand "rug" as carpet.

Instead of looking under the carpet:

if none of the drawers are explored, say "You peel up a corner of the rug gingerly; nothing results. A perfectly ordinary rug, then." instead;

if all of the drawers are explored, say "No time for that kind of nonsense. You'd better get out and away while you can." instead;

if some of the drawers are explored, say "No time for that nonsense. The desk's what you want now; what you came for won't be embedded in the flooring." instead.

The glass presentation case is transparent scenery in the Library. The description is "The case is taller than you are, framed in wood, with large panels of glass, the better to display the remarkable contents."

Instead of attacking the glass presentation case:

say "The glass pane of the presentation case shatters, throwing fine glass everywhere, including over the delicate museum-piece inside. Nor does the noise pass unnoted: only a second passes before there are footsteps in the hall, and though you make for the concealing darkness and escape of the turret, you are not swift enough. The servants are soon on you, and you are made to regret, quite painfully, this casual act of vandalism.";

end the story saying "You have lost your opportunity."

The strange dress is a wearable thing in the glass case. The description is "Not the sort of dress that anyone wears now: such elaboration would be ludicrous. It drips gold -- gilt lace, ruffles of trimmed gold, shimmering golden tracery -- dulled here and there by the sinister black of faceted jet."

Test me with "d / x case / x dress / x carpet / look under carpet / look under desk / x desk / open top drawer / close top drawer / look under desk / open bottom drawer / close bottom drawer / open middle drawer / get thing / look under carpet / look under desk / up".

397



Example Swignore U.

WI

Adding a new kind of supporter called a perch, where everything dropped lands on the floor.

Inform's default assumption is that if a player on an enterable object drops something, the dropped article winds up beside him on the same supporter or in the same container. This makes lots of sense for a dais, say, or a king-sized bed. It's a little less sensible if the enterable supporter in question is a bar stool or the like. So suppose we want to add a new kind of supporter called a perch, where everything dropped lands on the floor.

There are actually several ways of implementing this, but one of them is to reach right into the drop action and replace the "standard dropping rule" with a different one of our own invention -- like this:

"Swignore U."

Moe's Tavern is a room. The bar is an enterable supporter in Moe's. A drink is a kind of thing. On the bar is a drink called a flaming Homer.

A perch is a kind of supporter. A perch is always enterable. The stool is a perch in Moe's.

The player carries a dead field mouse and a tomacco fruit.

The sophisticated dropping rule is listed instead of the standard dropping rule in the carry out dropping rulebook.

This is the sophisticated dropping rule:
if the player is on a perch (called the awkward position):
 let place be the holder of the awkward position;
 move the noun to the place;
otherwise:
 move the noun to the holder of the player.

Test me with "sit on stool / drop mouse / look / get up / look".

Now the carry-out behavior of the dropping action has been changed, but we haven't had to interfere in the checks or reporting at all. The rest of the action works just as it always did.

Of course, maybe we do want to change the way the action is reported, to make it clearer to the player where the dropped article wound up:

The sophisticated report dropping rule is listed instead of the standard report dropping rule in the report dropping rulebook.

This is the sophisticated report dropping rule:
say "You drop [the noun] on [if the holder of the noun is a room]the ground[otherwise][the holder of the noun][end if]."

58



Example U-Stor-It

WI

A "chest" kind which consists of a container which has a lid as a supporter.

Suppose we want to write a game in which there are a number of chests. Each of these chests will be a container, but have a lid which is a supporter.

"U-Stor-It"

Section 1 - Assemblies and Supporters

A chest is a kind of container. A chest is always openable. A chest is usually fixed in place. A chest is usually closed. The specification of a chest is "Represents a container with a separately implemented lid; the lid is itself a supporter."

A lid is a kind of supporter. A lid is part of every chest. The specification of a lid is "A supporter attached to a chest, which can only support things when the chest is closed."

(The "specification" of a kind is not really a property, and is used instead to describe the kind in the Index. So the text of these specifications is never found in the game.) Of course, this doesn't get us very far. We will also want the game to correctly interpret variations on "open the chest" and "close the lid", redirecting actions appropriately.

Section 2 - Opening and closing

Before opening a lid which is part of a chest (called the item):
try opening the item instead.

Before closing a lid which is part of a chest (called the item):
try closing the item instead.

Before opening a chest (called the box) when something is on a lid (called the obstruction) which is part of the box:

repeat with item running through things on the obstruction:

say "(first removing [the item])";

try taking the item.

Instead of opening a chest when something is on a lid (called the item) which is part of the noun:

say "You'd have to remove [the list of things on the item] from the lid first."
instead.

Instead of looking under a lid which is part of a chest (called the item):
try opening the item.

We may also want to be able to deal with "put in" and "put on" appropriately, even if the player names the wrong part of the object:

Section 3 - Insertion and Support

Before inserting something into a lid which is part of a chest (called the item):
try inserting the noun into the item instead.

Before putting something on a chest when a lid (called the item) is part of the second noun:

try putting the noun on the item instead.

Furthermore, we don't want the player to be able to put things on the lid while the chest is open:

Before putting something on a lid which is part of an open chest (called the item):

say "(first closing [the item])";

try closing the item.

Instead of putting something on a lid which is part of an open chest (called the item):

say "[The item] would need to be closed first."

And then we may also want a couple of rules for describing our assembled object nicely:

Section 4 - Description in Rooms

Instead of examining a closed chest when something is on a lid (called the top) which is part of the noun:

say "[The noun] is closed, and there [is-are a list of things on the top] on top."

After printing the name of a chest (called the item) while listing contents of a room:

if a lid (called the second item) which supports something is part of the item:
say "(on which [is-are a list of things on the second item])";
omit contents in listing.

Now we are free to create entire treasure rooms at a single blow:

Section 5 - U-Stor-It Facility

The U-Stor-It Facility is a room. The sea trunk, the shipping crate, and a metal box are chests in the U-Stor-It Facility. The metal box contains a sapphire, a gold coin, and a signed photograph of Babe Ruth.

Even though we have never explicitly defined it, the metal box has a "metal box's lid", which we can use at need.

The metal box's lid supports a small card. The description of the small card is "It reads, 'Back in 5 mins - Pandora.'"

Test me with "open trunk / x card / open metal box / put all in metal box / get card / put card on box".

75

★★ Example Versailles

WI

A mirror which will reflect some random object in the room.

One of the advantages of descriptions is that we can use them to pick an item randomly from a specified category. (For more on this possibility, see the Change chapter sections on randomness.)

For instance, suppose we wanted to create a mirror in which the player would see some item from the room reflected. We might write

Instead of searching the mirror:

say "You see [a random thing in the location] reflected back at you."

This is the same as "a random thing which is in the location": phrase "in..." can be used briefly in Inform as it can in English.

But, on a little more thought, we might want to expand on this: the mirror perhaps should reflect not only things that are in the room, but anything that the player can see (even if it's on a supporter or carried by someone). So then we might instead write

Instead of searching the mirror:

say "You see [a random visible thing] reflected back at you."

There's still a risk, though, that this will produce the response

You see the mirror reflected back at you.

because, of course, the mirror is itself visible. So instead we might write

Instead of searching the mirror:

say "You see [a random visible thing which is not the mirror] reflected back at you."

"Versailles"

The Hall of Plywood Boards is a room. "The Hall of Mirrors is under reconstruction: it is currently a dank tunnel enlivened only by short placards about the history of the room.

As though to mock tourists such as yourself who bought their tickets without knowing this, the officials have left uncovered a single panel of mirror."

The mirror is scenery in the Hall of Plywood Boards. Understand "panel" or "panel of mirror" as the mirror. The description is "Lovingly restored to shimmering brilliance, it suggests how marvelous this room would be if you had had the good sense to arrive after the renovations were complete."

Some tourists are a person in the Hall of Plywood Boards. The tourists are scenery.

Instead of searching the mirror:

say "You see [a random visible thing which is not the mirror] reflected back at you."

Test me with "x mirror / look in mirror / g".

A final note: we use "searching" here because Inform understands both SEARCH THING and LOOK IN THING as the searching action, and the player is most likely to type LOOK IN MIRROR in order to see the reflection there. In the absence of an example, we can discover the relationship between actions and their command vocabulary in one of two ways. A complete list of actions and the vocabulary associated with them is available in the Actions index. Alternatively, we can type ACTIONS at a prompt, followed by LOOK IN MIRROR, and get the response

[searching the mirror]

You find nothing of interest.

[searching the mirror - failed the can't search unless container or supporter rule]

...which tells us that Inform is understanding the action as "searching the mirror".

449



Example Modern Conveniences

WI

Exemplifying the kind of source we might use in writing extensions for kitchen and bathroom appliances.

Suppose we want to write an extension or other portable code which defines a "kitchen" kind of room and a "bathroom" kind of room. All kitchen rooms we create in the future will automatically contain standard kitchen appliances: fridge, freezer, sink with taps, counters, cabinets, and a stovetop with built-in oven. Similarly, all bathrooms will have sinks, baths, cabinets, and toilets, and respond to some standard interactions.

We would do this with a standard assembly:

"Modern Conveniences"

Section 1 - Kitchens

A kitchen is a kind of room.

A refrigerator is a kind of container. A refrigerator is usually closed and openable. A refrigerator is usually fixed in place. A refrigerator is usually scenery. Understand "fridge" as a refrigerator.

A freezer compartment is a kind of container. A freezer compartment is usually closed and openable. A freezer compartment is part of every refrigerator.

Now: we're going to want many of the items in our kitchen to have switches, and to handle input sensibly regardless of whether the player types TURN ON STOVE or TURN ON STOVE SWITCH. (This is apparently a stove with only one burner.) For convenience, we'll define an "includes" relation:

Inclusion relates a thing (called X) to a thing (called Y) when Y is part of X. The verb to include means the inclusion relation.

A stove is a kind of supporter. It is usually scenery.

An oven is a kind of container. An oven is usually openable and closed. One oven is a part of every stove.

A switch is a kind of device. A switch is part of every stove. A switch is part of every oven.

Understand "[something related by reversed incorporation] switch" as a switch.

What follows is fairly straightforward, but notice that we are somewhat obsessively naming every rule. This is much more important in extensions (where someone else may need to manipulate our code from within their own source) than it is when we are simply composing source text for ourselves.

Setting action variables for opening a stove (this is the stove-opening rule):
let relevant oven be a random oven which is part of the noun;
now the noun is the relevant oven.

Setting action variables for switching on something which includes a switch (this is the redirecting switches for switching on rule):
let relevant switch be a random switch which is part of the noun;
now the noun is the relevant switch.

Setting action variables for switching off something which includes a switch (this is the redirecting switches for switching off rule):
let relevant switch be a random switch which is part of the noun;
now the noun is the relevant switch.

Before printing the name of a switch (called target) (this is the switch identification rule):
say "[random thing which includes the target]".

A sink is a kind of container. A sink is usually fixed in place and scenery. A tap is a kind of switch. A tap is part of every sink. Understand "faucet" or "taps" as a tap. Understand "[something related by reversed incorporation] tap/faucet/taps" as a tap.

Instead of opening a tap, try switching on the noun. Instead of closing a tap, try switching off the noun.

Report switching on a tap (this is the standard report switching taps on rule):
say "You turn on [the noun]." instead. [since "switch on" sounds weird in this context.]

Report switching off a tap (this is the standard report switching taps off rule):
say "You turn off [the noun]." instead.

After examining something which includes a switched on tap (called relevant tap) (this is the report flowing water rule):
say "The water is flowing from [the relevant tap]."

A drain is a kind of container. A drain is part of every sink. Understand "plughole" as the drain. Understand "[something related by reversed incorporation] drain/plughole" as a drain.

Instead of inserting something into a drain (this is the no clogging drains rule),
say "Pointless."

This is probably about as far as we want to go in a generic simulation: it is tempting to code up water, drains down which the player can lose objects, sinks that get clogged and overflow, and so on; but the more we embellish in these ways, the more likely the end result would be disruptive to individual games. For right now what we're aiming for is something simple which provides the basic interactions a player might expect in this kind of room, but which does not have any significant implications for the surrounding world model.

A particularly conservative author might even want to make it turn out that the water has been shut off and nothing flows from the taps: in the extension documentation,

we might want to include a line or two of example showing how this might be done by changing or removing the relevant rules of our extension.

A cabinet is a kind of container. A cabinet is usually openable and closed. It is scenery.

Understand "cupboard" or "cupboards" or "cabinets" as a cabinet.

A counter is a kind of supporter. It is scenery.

Understand "countertop" as a counter.

A cabinet is in every kitchen.

A counter is in every kitchen.

A refrigerator is in every kitchen.

A sink is in every kitchen.

A stove is in every kitchen.

Section 2 - Bathrooms

A bathroom is a kind of room.

A toilet is a kind of supporter. Understand "john" as a toilet. A toilet is usually fixed in place and enterable and scenery.

A bath is a kind of container. A bath is usually a fixed in place and enterable and scenery. A tap is part of every bath. A drain is part of every bath. Understand "bathtub" or "shower" as a bath.

A sink is in every bathroom.

A toilet is in every bathroom.

A bath is in every bathroom.

A cabinet is in every bathroom.

If we were feeling particularly ambitious and inclined toward interior decoration, we could add bath mats, mirrors, plungers, toilet brushes, overhead lighting, towel racks, scented candles, boxes of facial tissue, shampoo bottles, scrubbing loofahs, etc. ad nauseam; but we'll keep it relatively simple for now. Of course, if we have a toilet, we pretty much have to accept that the player will try to make use of it:

Understand "flush [toilet]" or "use [toilet]" as a mistake ("You have no need at the moment.").

Understand "take shower" or "take bath" or "bathe" or "wash" as bathing.

Bathing is an action applying to nothing.

Check bathing (this is the restrict baths to bathrooms rule):

if the location is not a bathroom, say "This isn't the place." instead.

Check bathing (this is the block bathing rule):

say "You haven't time for a bath." instead.

Washing is an action applying to one thing. Understand "clean [something]" or "wet [something]" or "wash [something]" as washing.

Instead of washing the player, try bathing.

Check washing (this is the restrict washing to the proximity of sinks rule):
unless the player can touch a sink, say "This isn't the place." instead.

Check washing (this is the block washing rule):
say "It doesn't seem worth the bother." instead.

Now we might put this to work in a short example.

One slight challenge lies in giving these assembled pieces separate descriptions. When we have an assembly that adds parts to objects, we can then talk about (for instance) "the stove's switch" elsewhere in the code. But items that have been assigned rooms are not named in the same way, so we cannot talk about "the Industrial Kitchen's stove" in our code as a way to assign it a description or special behavior. In quite a simple example, we could make the descriptions of the kind simply be the descriptions we want for the individual items:

Section 3 - An Example We Might Offer

Our Household Kitchen is a kitchen.

The Tiny Bathroom is a bathroom. It is west of Our Household Kitchen.

The description of a stove is "Scrupulously polished."

The description of a refrigerator is "It is baby blue and has the contours of a 50[']s chevy. One of these days it really will break down, but it's been serving your family faithfully since your grandmother's honeymoon."

Test me with "x refrigerator / open fridge / x freezer / look in freezer / open freezer / turn on stove / turn on oven / x oven switch / turn off oven switch / turn off stove switch / turn on taps / x sink / w / x sink / turn on sink / take bath / use toilet".

In a game that featured multiple bathrooms and kitchens, this wouldn't be enough; our author might give the stove kind (say) a description that checked its location, as in

The description of a stove is "[if in Industrial Kitchen]A massive stainless steel stove-top with six burners[otherwise]Your standard four-burner item[end if]."

or create an

Instead of examining the stove in the Industrial Kitchen: ...

sort of rule for those objects he wanted to describe specially; or he might use a when play begins rule to initialize a few things:

When play begins:
let N be a random stove in the Industrial Kitchen;

move the boiling pot to N;
change the description of N to...

Or we might even (if we anticipate lots of these kinds of amendments) want to rig up something more complex that finds the descriptions of appliances in a table, rather than relying on their individual description properties. This can all be done, but it is only interesting as long as it remains genuinely labor-saving: that is, as long as the convenience of having the assembly is greater than the annoyance of writing special rules to cover for the automation. In the end, the "kitchen" and "bathroom" room types are likely to be most useful for authors who want to include the standard props but not actually make them a critical part of the game; if stoves and sinks have more of a starring role in the production, authors may be better off coding them or at the very least placing them by hand, as in

The Industrial Kitchen is a room.

Thor is a stove in the Industrial Kitchen. It supports a boiling pot.

All these quirks are things that we (as the extension author) want to think out in advance: we should ideally warn authors about possible pitfalls in using our extension (if we can think of them) and point out ways of customizing the behavior (if there are interesting ways).

Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics

*§9.1. Food; §9.2. Bags, Bottles, Boxes and Safes; §9.3. Clothing; §9.4. Money;
§9.5. Dice and Playing Cards; §9.6. Reading Matter;
§9.7. Painting and Labeling Devices; §9.8. Simple Machines;
§9.9. Televisions and Radios; §9.10. Telephones; §9.11. Clocks and Scientific Instruments;
§9.12. Cameras and Recording Devices*

-  Contents of *The Inform Recipe Book*
-  Chapter 8: Vehicles, Animals and Furniture
-  Chapter 10: Physics: Substances, Ropes, Energy and Weight
-  Indexes of the examples

§9.1. Food

Inform provides an either/or property called "edible" and an action, "eating", for consuming edible things:

The lardy cake is edible. After eating the lardy cake, say "Sticky but delicious."

For eating something not immediately to hand, see **Lollipop Guild. Delicious, Delicious Rocks**, conversely, adds a sanity check which prevents the player from automatically taking inedible things only to be told they can't be eaten.

Inform does not normally simulate taste or digestion, but to provide foods with a range of flavours, see **Would you...?**; to make eating different foods affect the player differently, see **Stone**, or for the extreme case of poisoning foods, **Candy**. In **MRE**, hunger causes the player problems unless he regularly finds and eats food.

★ See **Liquids** for things to drink

★ See **Dispensers and Supplies of Small Objects** for a pizza buffet table from which the player may take all the slices he wants

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to Chapter 8: Vehicles, Animals and Furniture: §8.5. Kitchen and Bathroom
 - ➡ Onward to §9.2. Bags, Bottles, Boxes and Safes
 - ⬇ Example 48: **★ Would you...?** Adding new properties to objects, and checking for their presence.
 - ⬇ Example 133: **★ Candy** One of several identical candies chosen at the start of play to be poisonous.
 - ⬇ Example 143: **★ MRE** Hunger that eventually kills the player, and foodstuffs that can delay the inevitable by different amounts of time.
 - ⬇ Example 391: **★ Stone** A soup to which the player can add ingredients, which will have different effects when the player eats.
 - ⬇ Example 196: **★★★ Delicious, Delicious Rocks** Adding a "sanity-check" stage to decide whether an action makes any sense, which occurs before any before rules, implicit taking, or check rules.
-

§9.2. Bags, Bottles, Boxes and Safes

The kind "container" allows one thing to contain others. Things are sometimes containers automatically, sometimes by instruction:

The match is in the matchbox. The bucket is a container.

The matchbox, like the bucket, is a container. Containers come in all sizes and have a variety of behaviours, mainly controlled by the properties we give them: they can be "open" or "closed", "opaque" or "transparent" (when closed), "openable" or not, "lockable" or not, "enterable" or not. The basic ideas of containment are to do with carrying and sometimes hiding the contents, and Inform makes this easy. Allowing for locking and unlocking is again straightforward:

The strongbox is a locked container. The little steel key unlocks the strongbox.

Two built-in extensions enhance and modify the behavior of locks and keys: Locksmith automates a number of steps, automatically unlocking doors when trying to open them (for instance). Skeleton Keys allows us to define multiple keys that unlock the same object, rather than being restricted to one matching key per item.

For a container with a combination lock, rather than a key, see **Safety**; for a more sophisticated safe requiring digits dialed over multiple turns, see **Eyes, Fingers, Toes**.

Trachypachidae Maturin 1803 provides a bottle that is stoppered with a cork: when it is closed, the cork is part of the bottle, but otherwise the cork becomes a separate object we can carry around.

The normal assumption is that there is no problem with any two portable items being carried together, but in reality they may affect each other. (For effects like magnetism, or getting each other wet, or setting each other on fire, see the Physics chapter.) Here is a cat which, if boxed up with one or more items of food, will eat something each turn until all is gone:

The player carries a wicker basket and a scarlet fish. The cat is an animal in the wicker basket. The fish is edible.

Every turn when the cat is in a container (called the bag) and something edible (called the foodstuff) is in the bag:

remove the foodstuff from play;
say "With mingled sounds of mewling and chomping, the cat nibbles up [the foodstuff]."

The examples below provide subtler effects, adapting text to the current situation. In **Cinco**, the container's name changes depending on what it contains: putting beef in a taco allows the player to call it a SHREDDED BEEF TACO. In **Unpeeled** and **Shipping Trunk**, the description of something inside a container changes according to other things are alongside it. This is taken further in **Hudsucker Industries**, which describes the contents of a container as a group.

Finally, any action that destroys a container has to consider what to do with the things inside. **Fallout Enclosure** demonstrates a zapping action that destroys cash registers and shelves but leaves their contents tidily behind.

★ See **Liquids** for a *SHAKE* command that makes containers rattle when there are contents

★ See **Glass and Other Damage-Prone Substances** for opening containers by cutting into them

★ See **Fire** for fire damage that spreads between containers and their contents, leaving fireproof objects intact

★ See **Volume, Height, Weight** for containers breaking under the weight of their contents

★ See **Heat** for keeping things warm in insulated containers

★ See **Furniture** for chests with lids that can support other objects

★ See **Modifying Existing Commands** for ways to allow the player to unlock with a key he isn't currently holding

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.1. Food
 - ➡ Onward to §9.3. Clothing
 - ⬇ Example 298: ★ **Safety** A safe whose dial can be turned with SPIN SAFE TO 1131, and which will open only with the correct combination.
 - ⬇ Example 319: ★ **Cinco** A taco shell that can be referred to (when it contains things) in terms of its contents.
 - ⬇ Example 339: ★ **Shipping Trunk** A box of baking soda whose name changes to "completely ineffective baking soda" when it is in a container with something that smells funny.
 - ⬇ Example 344: ★ **Unpeeled** Calling an onion "a single yellow onion" when (and only when) it is being listed as the sole content of a room or container.
 - ⬇ Example 433: ★ **Eyes, Fingers, Toes** A safe with a multi-number combination, meant to be dialed over multiple turns, is implemented using a log of the last three numbers dialed. The log can then be compared to the safe's correct combination.
 - ⬇ Example 340: ★★ **Trachypachidae Maturin 1803** Bottles with removable stoppers: when the stopper is in the bottle, the bottle is functionally closed, but the stopper can also be removed and used elsewhere. Descriptions of the bottle reflect its state intelligently.
 - ⬇ Example 342: ★★★ **Hudsucker Industries** Letters which are described differently as a group, depending on whether the player has read none, some, or all of them, and on whether they are alike or unlike.
 - ⬇ Example 35: ★★★ **Fallout Enclosure** Adding an enclosure kind that includes both containers and supporters in order to simplify text that would apply to both.
-

§9.3. Clothing

A person can wear any (portable) thing which has the "wearable" property. (This property seldom needs to be quoted because it is deduced automatically from sentences like "Trevor wears a red hat.")

In most traditional IF, clothing is only used when it is exceptional in some way. That is, we ignore the three to eight different garments most people are wearing at any given time - the everyday clothes which people wear without thinking about them - and only simulate the unexpected extras: a borrowed jaunty red hat, a radiation-proof space suit, and so on.

These unusual garments turn up only occasionally in play and usually one at a time, so Inform does not normally provide rules to restrict how much or little is worn, or in what unlikely combinations. **Get Me to the Church on Time** categorises clothing by body area (trousers for lower body, shirts for upper); **Bogart** by layer, distinguishing underwear from outer garments. **What Not To Wear** combines both into a general-purpose system adequate for most kinds of clothing situations.

★ See **Kitchen and Bathroom** for a simple mirror implementation, which could be adapted to reflect what the player is currently wearing

Hays Code is a somewhat stripped down version.

Clothes are normally single things which have no function other than display and concealment, but **Being Prepared** gives them pockets which act as containers, and **Some**

Assembly Required allows clothes to be stitched together from pieces of cloth.

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.2. Bags, Bottles, Boxes and Safes
 - ➡ Onward to §9.4. Money
 - ⬇ Example 56: ★ **Being Prepared** A kind for jackets, which always includes a container called a pocket.
 - ⬇ Example 332: ★ **Some Assembly Required** Building different styles of shirt from component sleeves and collars.
 - ⬇ Example 338: ★ **Hays Code** Clark Gable in a pin-striped suit and a pink thong.
 - ⬇ Example 242: ★★ **What Not To Wear** A general-purpose clothing system that handles a variety of different clothing items layered in different combinations over different areas of the body.
 - ⬇ Example 46: ★★★ **Get Me to the Church on Time** Using kinds of clothing to prevent the player from wearing several pairs of trousers at the same time.
 - ⬇ Example 234: ★★★ **Bogart** Clothing for the player that layers, so that items cannot be taken off in the wrong order, and the player's inventory lists only the clothing that is currently visible.
-

§9.4. Money

Money could be anything which the two people in a bargain both agree is valuable. Here, the player and an ogre agree on a copper coin as money:

The player carries a copper coin. The ogre carries a rock cake. The cake is edible.

Instead of giving the coin to the ogre:
now the ogre carries the coin;
now the player carries the cake;
say "The ogre grunts and hands you a rock cake."

Now Inform does provide an action, "buying", and a command for it, BUY, but they ordinarily respond simply "Nothing is on sale." This is no longer true, so we should make BUY CAKE work. The difficulty here is that a command like BUY CAKE does not specify what should be handed over in exchange. Here we just check that the player has the coin, but in principle we could check for any of a range of monetary tokens - coins, notes, cheque book, debit card, and so on.

Instead of buying the cake:
if the player has the coin, try giving the coin to the ogre;
otherwise say "You have no money."

In more advanced economies, where shopping replaces barter, the seller will stock a wide range of differently priced goods. For a tabulated catalogue of wares, see **Introduction to Juggling**: to allow the player to negotiate prices, see **Money for Nothing**. In both of those examples, the player's current financial worth is simulated only as a current total amount of money carried - say, \$2.50. This is typical, because in most situations what matters is how much money is in the pocket, not how it is made up. Money behaves more like a liquid than

a set of items: hence terms like "liquidity", "cash flow" or **Frozen Assets** - the name of the simplest example demonstrating this. If we really need a comprehensive simulation down to pieces of currency - where it makes a difference carrying four quarters rather than a dollar bill, because the quarters can be fed into a vending machine - see **Nickel and Dimed**.

Fabrication takes the problem in a different direction, making calculations about the cost of a new garment based on the price of the pattern, the quantity of fabric required, and the value of the fabric type chosen -- showing off what we can do with unit multiplication in Inform.

Widget Enterprises explores the challenge of pricing widgets for maximum profit, given certain necessary costs and customers with varying willingness to pay.

★ See **Actions on Multiple Objects** for an implementation of giving that allows the player to offer multiple objects at once, where their combined value determines whether they are accepted

-
- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.3. Clothing
 - ➡ Onward to §9.5. Dice and Playing Cards
 - ⬇ Example 263: ★ **Widget Enterprises** Allowing the player to set a price for a widget on sale, then determining the resulting sales based on consumer demand, and the resulting profit and loss.
 - ⬇ Example 264: ★ **Frozen Assets** A treatment of money which keeps track of how much the player has on him, and a BUY command which lets him go shopping.
 - ⬇ Example 265: ★★ **Money for Nothing** An OFFER price FOR command, allowing the player to bargain with a flexible seller.
 - ⬇ Example 269: ★★★ **Fabrication** A system of assembling clothing from a pattern and materials; both the pattern and the different fabrics have associated prices.
 - ⬇ Example 262: ★★★★ **Nickel and Dimed** A more intricate system of money, this time keeping track of the individual denominations of coins and bills, specifying what gets spent at each transaction, and calculating appropriate change.
 - ⬇ Example 283: ★★★★★ **Introduction to Juggling** Assortment of equipment defined with price and description, in a table.
-

§9.5. Dice and Playing Cards

Most toys are single things, and no harder to create than any other small items, but games often require a multitude of tokens to be combined, and this can be logistically tricky.

The classic example is a pack of playing cards, where the player must individually control 52 items but without fussy commands or verbose text being printed back. **Jokers Wild** provides a simple "one card at a time" approach; **Tilt 1** is more sophisticated, with 52 independently accessible cards; **Tilt 2** can further judge the value of a selection of cards - the ranking of a poker hand.

Drawing cards from a shuffled pack is only one source of randomness. Games of chance also involve items drawn from a bag: **Wonka's Revenge** provides just such a lottery. More often, dice are thrown. A single die is easy enough:

The die is carried by the player. After dropping the die: say "It lands with [a random number from 1 to 6] uppermost." Understand "roll [something]" as dropping.

Quick, but not very good. Most dice games involve rolling more than one die at a time, to get a more interesting distribution of outcomes: they may also involve special rules applying to doubles, for instance. See **Do Pass Go**.

★ See **Typography** for on-screen notations for chess and card games

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.4. Money
 - ➡ Onward to §9.6. Reading Matter
 - ⬇ Example 129: **Do Pass Go** A pair of dice which can be rolled, and are described with their current total when not carried, and have individual scores when examined.
 - ⬇ Example 175: **Wonka's Revenge** A lottery drum which redistributes the tickets inside whenever the player spins it.
 - ⬇ Example 275: **Jokers Wild** A deck of cards which can be shuffled and dealt from.
 - ⬇ Example 318: **Tilt 1** A deck of cards with fully implemented individual cards, which can be separately drawn and discarded, and referred to by name.
 - ⬇ Example 407: **Tilt 2** A deck of cards with fully implemented individual cards; when the player has a full poker hand, the inventory listing describes the resulting hand accordingly.
-

§9.6. Reading Matter

Many things can be read, from warning notices to encyclopaedias, and a range of techniques is needed to provide them because the quantity of text, and how it presents itself, can vary so much. With a small amount of very large type, the player should not need any command at all to read the text:

The road sign is in the Junction. The road sign is fixed in place. "A road sign points north: 'Weston on the Green - 6'."

If the print is smaller, or the object portable, the player will expect to use the EXAMINE command:

The business card is in the Junction. The description is "'Peter de Sèvres: consultant mnemonicist.'"

But if the object is a leaflet, say, EXAMINE should only describe the cover: READ would be the command a player would expect to use to get at the text itself. Inform normally defines READ to be the same command as EXAMINE, which is good for things like the business card, but counter-productive here. **The Trouble with Printing** shows how to separate these two commands, allowing any thing to have a property called its "printing" for text produced by READ, which will be different from its "description", the text produced by EXAMINE.

If the object is a lengthy diary, say, nobody would read it from cover to cover in a single IF turn. We might then want to allow the player to turn the pages one by one, with commands like READ PAGE 4 IN DIARY or READ THE NEXT PAGE: see **Pages**.

If the object is an encyclopaedic reference work, the player would consult named entries: see **Costa Rican Ornithology**, which allows commands like LOOK UP QUETZAL IN GUIDE.

Still larger sources of text often occur in IF: libraries or bookshelves, where many books are found together, and it is clumsy to write them as many individual items. One approach is to simulate an entire bookshelf with a single thing: see **Bibliophilia**. (This is much like looking up topics in a single book, except that each topic is a book in itself.) Another is to provide each book as an individual item, but have them automatically join together into a single portable collection: see **AARP-Gnosis**.

Signs, leaflets and encyclopaedias, being printed, have a wording which will never change during play. But sometimes the player reads something which acts of its own accord. Text substitutions are usually all that is needed to achieve this:

The computer display is on the desk. The description is "Giant green digits read: [the time of day]."

This is easy because we know all the variations we want. But what if we want the player to write his own text, for instance, adding to a diary? This is trickier, because it means storing text as the player typed it, and replaying it later. (And suppose the player types reams and reams of text, not just a few words as we might have hoped?) **The Fourth Body** and **The Fifth Body** show how to use an external file - a multimedia trick requiring features only available if the project is set to the Glulx story file format - to handle even the most difficult cases.

Should we want a computer that responds to vocal commands, as in ASK COMPUTER ABOUT KLINGONS, the built-in extension Inanimate Listeners will allow the player to talk to inanimate objects as well as people.

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.5. Dice and Playing Cards
 - ➡ Onward to §9.7. Painting and Labeling Devices
 - ⬇ Example 292: ★ **The Trouble with Printing** Making a READ command, distinct from EXAMINE, for legible objects.
 - ⬇ Example 303: ★ **Pages** A book with pages that can be read by number (as in "read page 3 in...") and which accepts relative page references as well (such as "read the last page of...", "read the next page", and so on).
 - ⬇ Example 443: ★ **The Fourth Body** Notebooks in which the player can record assorted notes throughout play.
 - ⬇ Example 324: ★★ **Bibliophilia** A bookshelf with a number of books, where the player's command to examine something will be interpreted as an attempt to look up titles if the bookshelf is present, but otherwise given the usual response.
 - ⬇ Example 335: ★★★ **AARP-Gnosis** An Encyclopedia set which treats volumes in the same place as a single object, but can also be split up.
 - ⬇ Example 444: ★★★ **The Fifth Body** An expansion on the notebook, allowing the player somewhat more room in which to type his recorded remark.
 - ⬇ Example 94: ★★★★ **Costa Rican Ornithology** A fully-implemented book, answering questions from a table of data, and responding to failed consultation with a custom message such as "You flip through the Guide to Central American Birds, but find no reference to penguins."
-

§9.7. Painting and Labeling Devices

Writing on something is only one way a player can change its visual appearance. IF authors have long been wary of paint brushes, because a sufficiently motivated player could go through a whole landscape like a graffiti artist with a railway bridge. We want to give the player the illusion of freedom of action, while avoiding a situation where unlimited numbers of different decorations might be needed - that would need a table of potentially unlimited size.

One approach is to limit the number of items which can be decorated. In **Palette**, only the canvas can be painted, and each image overlays the last. **Early Childhood** increases the range to allow a whole kind ("block") to be painted, and also shows how the changing colours can be used to distinguish between otherwise identical objects.

Brown finds a different way to limit the number of simultaneous decorations: almost anything can have a red sticky label attached, but there is only one red sticky label. (So to decorate a new item, the player must first un-decorate an old one.)

★ *See **Electricity and Magnetism** for another form of stickiness*

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.6. Reading Matter
 - ➡ Onward to §9.8. Simple Machines
 - ⬇ Example 308:  **Palette** An artist's workshop in which the canvas can be painted in any colour, and where painterly names for pigments ("cerulean") are accepted alongside everyday ones ("blue").
 - ⬇ Example 36:  **Brown** A red sticky label which can be attached to anything in the game, or removed again.
 - ⬇ Example 55:  **Early Childhood** A child's set of building blocks, which come in three different colours - red, green and blue - but which can be repainted during play.
-

§9.8. Simple Machines

The "device" kind provides for the simplest form of machine: one which is, at any given moment, switched on or switched off. Inform looks after this state, but leaves it to us to make the machine actually do something:

The air-conditioning unit is a device in the Florist's Shop. The air-conditioning is fixed in place and switched on.

Every turn when the location is the Florist's Shop:
 if the air-conditioning is switched off, say "You worry about the cut flowers in this jungle-hot air."
 otherwise say "There is an low susurrantion from the air-conditioning unit."

One primary dictionary definition for a machine is "an apparatus using or applying mechanical power and having several parts", and we often use the "part of" relationship to build machinery. **Control Center** provides a neat way to display the component parts of a machine to the player who examines it.

One component almost always part of an electrical machine is the (literal) switch, lever or button to control whether it is switched on or off. In **Model Shop** just such an on/off button is automatically made part of every device.

While an electrical device has only two states, a mechanical machine might have many, and for these the best approach is to define a kind of value naming the possibilities: see **Signs and Portents**, where the states are the possible destinations pointed towards.

Perhaps stretching the definition of "machine", **What Makes You Tick** demonstrates a fishing pole which the player can put together from several pieces.

★ See **Bags, Bottles, Boxes and Safes** for a safe that can be dialed to different combinations

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.7. Painting and Labeling Devices
 - ➡ Onward to §9.9. Televisions and Radios
 - ⬇ Example 61: ★ **Control Center** Objects which automatically include a description of their component parts whenever they are examined.
 - ⬇ Example 427: ★ **What Makes You Tick** Building a fishing pole from several component parts that the player might put together in any order.
 - ⬇ Example 57: ★★ **Model Shop** An "on/off button" which controls whatever device it is part of.
 - ⬇ Example 53: ★★★ **Signs and Portents** Signpost that points to various destinations, depending on how the player has turned it.
-

§9.9. Televisions and Radios

IF authors often provide clues or background information to the player by means of radio broadcasts, TV shows or video tapes because they can talk to the player without needing to listen back, or to react to what the player does. The simplest radio set, like the one in **Aftershock**, really only has one thing to say: which is serendipitously being broadcast just at the moment the player tunes in (regardless of when that is). The next simplest approach is to spool a broadcast on an endless loop taking several turns to play through, as in **Radio Daze**.

Televisions come in all shapes and sizes, and **Aspect** allows their shape ("aspect ratio") to be described by the player.

In **Channel 1**, we can also refer to the television by what it is currently showing: thus WATCH CHANNEL 4 will work if the set is indeed tuned to 4. In **Channel 2**, numbered channel changing is taken further: we can now TUNE TV TO CHANNEL 3, as well. **Channel 2** is a reasonable base implementation of a television for many purposes.

-
- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.8. Simple Machines
 - ➡ Onward to §9.10. Telephones
 - ⬇ Example 66: ★ **Radio Daze** A radio that produces a cycle of output using varying text.
 - ⬇ Example 311: ★ **Aspect** Understanding aspect ratios (a unit) in the names of televisions.
 - ⬇ Example 313: ★★ **Channel 1** Understanding channels (a number) in the names of televisions.
 - ⬇ Example 316: ★★★ **Channel 2** Understanding channels (a number) in the names of televisions, with more sophisticated parsing of the change channel action.
 - ⬇ Example 336: ★★★ **Aftershock** Modifying the rules for examining a device so that all devices have some specific behavior when switched on, which is described at various times.
-

§9.10. Telephones

Telephones are much harder to achieve than televisions and in some ways as difficult to make convincing as a human character is - though of course there are corners which can be cut: we could have the reception drop off, or the other party hang up in a misunderstanding, and so on.

A single telephone line is tricky enough to provide that one might just as well have a general solution providing a whole network. **Four Cheeses** demonstrates a system where we can dial either people or numbers: CALL JANET ON TELEPHONE, or CALL 7103, for instance.

While **Four Cheeses** provides only four-digit phone numbers, like internal company extensions, **Alias** shows how to manage US-style seven digit numbers.

Finally, we might occasionally want the player to be able to address a microphone or telephone receiver directly when the identity of the person on the other end is unknown, in the form TELL MICROPHONE ABOUT CRIME. Ordinarily Inform will disallow this because we're not allowed to talk to inanimate objects, but the extension Inanimate Listeners provides more options.

★ See **Saying Complicated Things** for more approaches to conversation

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.9. Televisions and Radios
 - ➡ Onward to §9.11. Clocks and Scientific Instruments
 - ⬇ Example 226:  **Four Cheeses** A system of telephones on which the player can call distant persons and have conversations.
 - ⬇ Example 253:  **Alias** A telephone with phone numbers of the standard American seven-digit length.
-

§9.11. Clocks and Scientific Instruments

The simplest form of clock is a wrist watch. Here is a choice of analogue or digital:

The player wears a wrist watch. The description of the wrist watch is "It is [the time of day in words]."

The player wears a digital watch. The description of the digital watch is "It is [the time of day]."

Better clocks would allow us also to set the time, and to stop and start them: see **Tom's Midnight Garden**.

Scientific instruments provide sharper versions of our own senses. In the case of vision, they allow us to see closer up, or further away. It's a convention of IF that people can normally see only the current location, that is, they cannot see from one location into another. The boundary of the current room is like a horizon, even out of doors (though it's true that there are ways to disguise that with a continuous outdoor landscape). **Ginger Beer** provides a telescope able to see into other rooms.

Witnessed 2 provides a meter which measures how close a ghost is to the player.

★ See **Continuous Spaces and The Outdoors** for more on seeing into adjacent locations

★ See **Heat** for infrared goggles

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.10. Telephones
 - ➡ Onward to §9.12. Cameras and Recording Devices
 - ⬇ Example 180: ★ **Witnessed 2** A piece of ghost-hunting equipment that responds depending on whether or not the meter is on and a ghost is visible or touchable from the current location.
 - ⬇ Example 299: ★ **Tom's Midnight Garden** A clock kind that can be set to any time using "the time understood"; may be turned on and off; and will advance itself only when running. Time on the face is also reported differently depending on whether the clock is analog or digital.
 - ⬇ Example 365: ★★ **Ginger Beer** A portable magic telescope which allows the player to view items in another room of his choice.
-

§9.12. Cameras and Recording Devices

Recording what is going on, for later playing back or examination, is difficult because the range of situations is very complex. Exactly how much information should we store when we make a recording, and will this require problematically large tables? Will it be difficult even to do at all?

The usual approach is to record only basic details of events or situations. In **If It Hadn't Been For...** the tape recorder preserves only a few different sounds - footsteps, creaking, rustling - rather than capturing exactly the sound of every action taking place in earshot. In **Claims Adjustment**, we can take up to 36 Polaroid-style photographs, but each is described only by saying what it is a photo of. Thus we can have a photograph of a vase, or even a photograph of a photograph of a vase (because that too is a thing), but not a photograph of a still life in which several items have been gathered together by the player. That would ordinarily require too much storage.

A similar trick, though involving impromptu sculpture rather than photography, can be found in **Originals**. (The artist magically "manifests" these models rather than sculpting the conventional way in order to avoid the nuisance of carrying around raw materials - wax maquettes and so forth - which would clutter up the example.)

Text, of course, can store arbitrary descriptions. **Mirror, Mirror** provides a perfect visual recorder: it remembers a room description exactly as the player saw it at the time.

Actor's Studio provides a video camera that records and time stamps all actions performed in its presence while it is set to record.

★ See **Telephones** for ways to speak to inanimate objects, which might be appropriate when, say, tape-recording a confession

- ⬆ Start of Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
 - ⬅ Back to §9.11. Clocks and Scientific Instruments
 - ➡ Onward to Chapter 10: Physics: Substances, Ropes, Energy and Weight: §10.1. Gases
 - ⬇ Example 273: **★ If It Hadn't Been For...** A sound recording device that records the noises made by player and non-player actions, then plays them back on demand.
 - ⬇ Example 369: **★ Originals** Allowing the player to create models of anything in the game world; parsing the name "model [thing]" or even just "[thing]" to refer to these newly-created models; asking "which do you mean, the model [thing] or the actual [thing]" when there is ambiguity.
 - ⬇ Example 417: **★ Mirror, Mirror** The sorcerer's mirror can, when held up high, form an impression of its surroundings which it then preserves.
 - ⬇ Example 221: **★★ Actor's Studio** A video camera that records actions performed in its presence, and plays them back with time-stamps.
 - ⬇ Example 322: **★★ Claims Adjustment** An instant camera that spits out photographs of anything the player chooses to take a picture of.
-

Examples from Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics

- ⬆ Start of this chapter
- ➡ Chapter 10: Physics: Substances, Ropes, Energy and Weight
- ⬇ Indexes of the examples

48

★ Example **Would you...?**

WI

Adding new properties to objects, and checking for their presence.

For instance, if we want to give some objects a flavor:

"Would you...?"

The House is a room. The mouse is an animal in the House.

The player carries some green eggs and a ham.

A food is a kind of thing that is edible. Food has some text called flavor. The flavor of food is usually "Tolerable."

Things are, in general, not edible by default, so we have to make them edible specifically in order to allow them to be eaten by the player. Here we've defined food to be edible by default, and we have given it a standard piece of flavor text.

The ham and the green eggs are food. The flavor of the green eggs is "Delicious!"

After eating something:

if the noun provides the property flavor, say "[the flavor of the noun][paragraph

break]";
otherwise say "It's [noun]-flavored."

Note that we use "if the noun provides a flavor..." to make sure that the property exists before attempting to use it. Otherwise, there is the risk that we will try to print a property that does not exist, resulting in errors in the game.

We will only get the "It's [noun]-flavored." response if we successfully eat something that is not a food and does not have flavor text. To test this feature, let's suppose something that isn't exactly food but can theoretically be chewed on:

The player carries some paper. The paper is edible.

Test me with "eat ham / eat green eggs / eat paper".

133

Example Candy

WI

One of several identical candies chosen at the start of play to be poisonous.

Suppose we want to give the player a bag of candies, of which a random one is poisonous. We can pick which one should be poisoned at the start of play, like this:

"Candy"

The plural of piece of candy is pieces of candy. A piece of candy is a kind of thing. A piece of candy is always edible. Four pieces of candy are in the Halloween bag.

Toxicity is a kind of value. The toxicities are safe and poisonous. A piece of candy has a toxicity. A piece of candy is usually safe.

The Porch is a room. The player carries the Halloween bag.

After eating a poisonous piece of candy:
say "Oh, that didn't taste right at all. Oh well!"

When play begins:
now a random piece of candy is poisonous.

Test me with "eat candy / g / g / g".

143

Example MRE

WI

Hunger that eventually kills the player, and foodstuffs that can delay the inevitable by different amounts of time.

Many older interactive fiction games required the player to find food and eat on a regular basis in order to avoid death. This effect was often unrealistic (since most people can survive much longer than a few hours without eating) and is often seen as an annoyance. However, for the sake of argument, suppose that we do want to construct a hunger-and-death system.

To make things a little more interesting, we will postulate that different foods are differently filling, so that if the player manages to find something really caloric, he is off the hook on his hunger search for a while.

We will also implement the system so that the player gets messages when he is hungry, then dies a short time later. (The times involved are ludicrously short, but this allows us to see the effects within a simple example. In a real game we would want to allow a considerably longer timer for the hunger to play out.)

First, a little scene-setting:

"MRE"

When play begins:

now the right hand status line is "[time of day]";
say "The procedure was painless at first: increased strength was the first sign, followed by a sensation of delayed time, as though everyone around you moved more slowly. Your ability to dodge and perform feats of agility doubled, then trebled. You were heralded as a triumph of medicine. They told you there would be no side effects worth speaking of.

They were wrong."

The Base Camp Larder is a room. "This room has been reinforced after each incident -- and there have been dozens in the last two months -- so that it now rivals Fort Knox. Only your new skill and speed enabled you to dodge the motion sensors, knock out the computerized security system, fool the retinal scanner, and punch a hole in the steel containment grating. But you're inside now."

Now we define our food, and add some special instructions for what happens to our hunger counters when the food is eaten:

Food is a kind of thing. Food is usually edible. Food has a time called the satisfaction period. The satisfaction period of a food is usually 5 minutes.

A person can be hungry or replete. The player is hungry.

The Larder contains an apple, a candy bar, and a large plate of pasta. The apple, the candy bar, and the pasta are food. The satisfaction period of the apple is 2 minutes. The satisfaction period of the pasta is 125 minutes.

Check eating something which is not food:

say "[The noun] might be edible, but it's not what you'd consider really food."

Check eating something when the player is not hungry:

say "You're not hungry right now."

Carry out eating something:
now the player is replete;
hunger resumes in the satisfaction period of the noun from now.

The first of those two phrases, "now the player is replete", causes the player to cease to be hungry; the second one sets up a future event in which the hunger sets in again. The length of time until that event depends on how satisfying the specific food is. Now we define that event:

At the time when hunger resumes:
starvation occurs in three minutes from now;
now the player is hungry.

At the time when starvation occurs:
if the player is hungry, end the story saying "You have starved".

Note "if the player is hungry": it is possible that the starvation event will be set up but the player will eat before it occurs; in that case, we want it not to take effect.

And now, since we really ought to give the player some warning of what is happening to him:

Every turn when the player is hungry:
choose a random row in the Table of Hunger Complaints;
say "[hunger entry][paragraph break]".

Table of Hunger Complaints

hunger

"Gosh, you're starving."

"It feels as though you haven't eaten in days. Weeks, almost."

"The world seems to slow down and everything becomes sharper and brighter.

You are a hunter, a hunter of foodstuffs."

"You find yourself staring at [the random visible thing that is not the player] and wondering how it would taste."

Test me with "eat apple / z / z / z / eat candy bar / z / z / z / z / z / z / z / z / z / z".

391

★ Example Stone

WI

A soup to which the player can add ingredients, which will have different effects when the player eats.

A thing can have a rule as a property, if we like. Here we are going to allow the player to make a soup whose effects will depend on its ingredients. Each ingredient will have its own "food effect" rule, to be followed when the food is eaten.

Note that there are other, slightly less cumbersome ways to do the same thing -- we will see in the chapter on Rulebooks that we could make a "food effects rulebook" and then write a number of rules such as "food effects rule for carrots" or "food effects rule for the stone". Nonetheless, we demonstrate rules-as-properties here for the sake of thoroughness.

So:

"Stone"

A food is a kind of thing that is edible. A food has a rule called the food effect.

Carry out eating a food:

if a food is part of the noun:

repeat with item running through things which are part of the noun:

if item is a food, follow the food effect of the item;

follow the food effect of the noun.

Report eating a food:

say "You eat [the noun]. [diagnosis of the player]";

stop the action.

To say diagnosis of (victim - a person):

if the victim is ill:

say "You are ill.";

rule succeeds;

otherwise:

say "You are healthy. ";

if the victim is awake, say "You are wide awake. ";

otherwise say "You are sleepy. ";

if the victim is bright-eyed, say "Your eyesight is clear. ";

otherwise say "Your eyesight is dim. ";

if the victim is weak, say "You are weak. ";

otherwise say "You are strong. ";

if the victim is hungry, say "You are hungry.";

otherwise say "You are well-fed."

And now to provide some particular foods:

Some carrots are a food. The food effect of carrots is the bright-eye rule. This is the bright-eye rule: now the player is bright-eyed.

Some potatoes are a food. The food effect of the potatoes is the sleepiness rule. This is the sleepiness rule: now the player is sleepy.

The broth is a food. The indefinite article of the broth is "some". The food effect of broth is the filling rule. This is the filling rule: now the player is full.

The hambone is a food. The food effect of the hambone is the heartiness rule. This is the heartiness rule: now the player is strong. Instead of eating the hambone: say "You cannot just eat a bone!"

The poison ivy is a food. "Poison ivy grows nearby." The food effect of poison is the illness rule. This is the illness rule: now the player is ill.

A person can be bright-eyed or blind. The player is blind.

A person can be well or ill. The player is well.

A person can be hungry or full. The player is full.

A person can be strong or weak. The player is weak.

A person can be awake or sleepy. The player is sleepy.

The broth is in the kettle. The kettle is on the fire. The fire is in the Clearing. The Clearing is a room.

The player carries the hambone, the potatoes, and the carrots. The ivy is in the clearing.

Instead of examining the broth:

if something is part of the broth, say "In the broth, [a list of things that are part of the broth] float[if exactly one thing is part of the broth]s[end if].";

otherwise say "It is just a thin broth with no other ingredients."

Instead of inserting something into the broth: try inserting the noun into the holder of the broth.

Instead of taking the broth: say "You cannot take the broth in your bare hands."

And the following is a relatively unimportant nicety:

To sink is a verb.

After inserting a food which is not the broth into a container which contains the broth:

now the noun is part of the broth;

say "[The noun] [sink] into [the second noun], making the broth richer."

Test me with "x broth / eat hambone / put hambone in kettle / x broth / put potatoes in broth / x broth / eat carrots / eat broth / put ivy in kettle / eat ivy".

196



Example Delicious, Delicious Rocks

WI

Adding a "sanity-check" stage to decide whether an action makes any sense, which occurs before any before rules, implicit taking, or check rules.

In some cases, we may want to add new stages to action processing. One possibility is a stage where we check the sanity of what the player is trying to do before executing any of the other commands; so that we avoid, for instance

```
>EAT ROCK
(first taking the rock)
That's plainly inedible.
```

Here is how we might insert such a stage in our action processing, using rulebook manipulation.

"Delicious, Delicious Rocks"

Section 1 - Procedure

The sanity-check rules are a rulebook.

This is the sanity-check stage rule:
abide by the sanity-check rules.

The sanity-check stage rule is listed after the before stage rule in the action-processing rules.

Section 2 - Scenario

Candyland is a room. The lollipop tree is an edible thing in Candyland. The genuine rock is a thing in Candyland.

Sanity-check eating an inedible thing:
say "Your digestion is so delicate -- you're sure [the noun] wouldn't agree with you." instead.

Test me with "eat lollipop / eat rock".

Notice that now Inform does not try taking the rock before rejecting the player's attempt to eat it.

It is of course possible to get the same effect with

Before eating an inedible thing:
say "Your digestion is so delicate -- you're sure [the noun] wouldn't agree with you." instead.

...and in a small game with few rules, there's not much reason to add an extra stage. The ability to modify the stages of action processing becomes useful when we have a fairly large game with sophisticated modeling and want to be sure that some kinds of message (such as the sanity-check) are always handled before other things that we might be doing at the before stage (such as generating implicit actions like opening doors before going through them).

298

★ Example Safety

A safe whose dial can be turned with SPIN SAFE TO 1131, and which will open only with the correct combination.

WI

"Safety"

The Vault is a room. "Snug yet paranoid, this represents the state of the art in cheerless security." The Safe is here. "A mammoth safe, with a dial which can spin to any number, has pride of place. It must weigh about the same as a small car, so don't get any ideas." Instead of opening the safe, say "The safe opens only when turned to the correct combination."

In the Safe is a silver florin. The Safe is closed and fixed in place. Understand "dial" as the Safe.

Spinning it to is an action applying to one thing and one number. Check spinning it to: if the noun is not the Safe, say "[The noun] does not spin." instead. Report spinning it to: say "Click! and nothing else happens."

Understand "spin [something] to [a number]" as spinning it to.

After spinning the closed Safe to 1384: now the Safe is open; say "Clonk! and the safe door swings slowly open, revealing [a list of things in the Safe]."

Test me with "open safe / spin safe to 1131 / open safe / spin safe to 1384 / x safe / get florin".

319

Example Cinco

WI

A taco shell that can be referred to (when it contains things) in terms of its contents.

It's fairly common that we want to be able to refer to a container in terms of what it has in it: a bottle of wine, a salt shaker, a chicken sandwich. The player is free to remove the contents again, and the object will go back to using its usual name:

"Cinco"

Cinco de Mayo Fundraiser is a room.

The taco shell is an edible thing in the Fundraiser. It is a portable container. It has carrying capacity 1.

Understand "[something related by containment] taco" as the taco.

Rule for printing the name of the taco shell while not inserting or removing:
if the taco contains something (called filling), say "[filling] taco";
otherwise say "taco shell";
omit contents in listing.

The player carries shredded beef. It is edible.

The taking action has an object called source (matched as "from").

Setting action variables for taking:
now source is the holder of the noun.

Report taking something from the taco shell:
say "You gingerly pick [the noun] out of the taco shell." instead.

Test me with "x taco / put shredded beef in taco / get taco / i / x shredded beef taco / get shredded beef / x shredded beef taco".

★ Example Shipping Trunk

A box of baking soda whose name changes to "completely ineffective baking soda" when it is in a container with something that smells funny.

First to lay some groundwork:

"Shipping Trunk"

A chest is a kind of container. A chest is always openable. A lid is a kind of supporter. A lid is part of every chest.

Before opening a chest when something (called the obstruction) is on a lid which is part of the noun:

say "Better remove [the obstruction]." instead.

A thing can be innocent or smelly.

The Storage Unit is a room. The shipping trunk is a closed chest in the Storage Unit. The trunk contains some garlic, a loaf of moldy sourdough, a mildewy bath towel, a pair of unwashed socks, two dead trout, and a box of baking powder. The garlic, trout, sourdough, bath towel, and socks are smelly. The baking powder is innocent.

The shipping trunk's lid supports a small card. The description of the small card is "Please, please do not open this trunk."

After opening the trunk:

if the trunk had been open:

say "You steel yourself...";

continue the action;

otherwise:

say "There roils up from inside an indescribable funk, which, when you can see straight, you have no trouble attributing to the presence of [a list of smelly things in the trunk]. You also note [a list of innocent things in the trunk] in the corner."

And now, with that preparation:

Before printing the name of the baking powder when the powder is in a container which contains a smelly thing: say "completely ineffective ".

Test me with "open trunk / examine card / get card / open trunk / get powder / inventory".

★ Example Unpeeled

Calling an onion "a single yellow onion" when (and only when) it is being listed as the sole content of a room or container.

"Unpeeled"

Scullery is a room. A sack is carried by the player. The sack contains a yellow onion. The player carries a cork.

Before printing the name of the onion while listing contents:
if the holder of the onion contains exactly 1 thing, say "single ".

Test me with "i / put cork in sack / i".

433

★ Example Eyes, Fingers, Toes

WI

A safe with a multi-number combination, meant to be dialed over multiple turns, is implemented using a log of the last three numbers dialed. The log can then be compared to the safe's correct combination.

It is not difficult to implement a safe which can be set to a single number to open; but a more common scenario in the real world is for the safe to open on a sequence of numbers when they have been dialed in the right order.

For IF, this means that we have to keep running track of the last N digits the player has dialed, dropping the first digit and adding a new one to the end each time the player re-dials the safe. This is a perfect occasion for lists:

"Eyes, Fingers, Toes"

The Addams Wine Cellar is a room. It contains a closed lockable locked container called a safe.

The safe has a list of numbers called the current combination.

The safe has a list of numbers called the true combination. The true combination of the safe is {2, 10, 11}.

Understand "set [something] to [a number]" as setting it numerically to. Setting it numerically to is an action applying to one thing and one number.

Instead of examining the safe:

if the number of entries in the current combination of the safe is 0,
say "You haven't dialed the safe to any combination yet."
otherwise say "You have dialed the safe to [the current combination of the safe]."

Check setting something numerically to (this is the block setting numerically rule):

say "[The noun] cannot be set."

Instead of setting the safe numerically to the number understood:

truncate the current combination of the safe to the last 2 entries;
add the number understood to the current combination of the safe;
if the safe is locked and the current combination of the safe is the true

combination of the safe:

say "You dial [the number understood], and [the safe] gives a joyous CLICK.";

now the safe is unlocked;

otherwise if safe is unlocked and the safe is closed and the current combination of the safe is not the true combination of the safe:

say "You spin the dial, and [the safe] snicks locked.";

now the safe is locked;

otherwise:

say "You dial [the number understood] on the safe."

Test me with "x safe / set safe to 10 / x safe / set safe to 29 / x safe / set safe to 2 / x safe / set safe to 10 / x safe / set safe to 11 / open safe / set safe to 14 / close safe / set safe to 15 / open safe".

340



Example Trachypachidae Maturin 1803

WI

Bottles with removable stoppers: when the stopper is in the bottle, the bottle is functionally closed, but the stopper can also be removed and used elsewhere. Descriptions of the bottle reflect its state intelligently.

"Trachypachidae Maturin 1803"

A bottle is a kind of container. Bottles are usually openable, transparent, and closed. A cork is a kind of thing. A cork is in every bottle.

Understand "cork [something]" as corking.

Understand the command "stopper" as "cork".

Understand "uncork [something]" as uncorking.

Corking is an action applying to one thing.

Check corking:

if the noun is not a bottle, say "[The noun] cannot be corked." instead.

Carry out corking:

try closing the noun.

Uncorking is an action applying to one thing.

Check uncorking:

if the noun is not a bottle, say "[The noun] cannot be uncorked." instead.

Carry out uncorking:

try opening the noun.

Understand "close [something] with [something preferably held]" as corking it with.

Understand "cork [something] with [something preferably held]" as corking it with.

Corking it with is an action applying to one thing and one carried thing.

Check corking it with:

- if the noun is not a bottle, say "[The noun] cannot be corked." instead;
- if the second noun is not a cork, say "[The second noun] will not fit in [the noun]." instead.

Carry out corking it with:

- try inserting the second noun into the noun instead.

Instead of closing a bottle:

- if a cork (called the item) is carried by the player, try inserting the item into the noun instead;
- otherwise say "You need a stopper of some kind."

Instead of opening a bottle:

- if a cork (called the item) is in the noun, try taking the item instead;
- otherwise say "[The noun] has no stopper."

Carry out inserting a cork into a bottle:

- now the second noun is closed.

After inserting a cork into a bottle:

- say "You stopper [the second noun] with [the noun]."

Before taking a cork when the noun is in a closed bottle (called the item):

- now the item is open.

Instead of taking a cork when the noun is in a bottle (called the item):

- move the noun to the player;
- say "You pull [the noun] from [the item]." instead.

Before printing the name of a bottle (called target) while not inserting, taking, searching, or removing:

- if the target is closed, say "sealed ";
- otherwise say "now open ".

After printing the name of a bottle (called target) while not inserting, searching, examining, or removing:

- if the target contains a noncork thing, say " containing [a list of noncork things in the target]";
- omit contents in listing.

Instead of examining a bottle:

- say "[The noun] contains [a list of noncork things in the noun]."

Definition: a thing is noncork if it is not a cork.

The Doctor's Cabin is a room. "A dark, cramped triangle, like a slice of cake, except that its sharp end has been cut off: and so low that a moderately tall man would strike his head on the deck above if he were to stand upright. Every free

surface is covered with sheets of best Venetian looking-glass, to increase the light filtering in. Long use and the carpenter's ingenuity have packed in a folding cot and table, and lockers are built into unlikely places: lockers filled with specimens, skeletons, sketches, drafts and serial letters." The jug is a bottle in the Doctor's Cabin. The jug contains a beetle. The description of the beetle is "The doctor assures you that it is a nondescript."

Test me with "get jug / x jug / open jug / x jug / i / x cork / cork jug / i / uncork jug / i / x jug / get beetle / i / close jug / i / x jug".

342



Example Hudsucker Industries

WI

Letters which are described differently as a group, depending on whether the player has read none, some, or all of them, and on whether they are alike or unlike.

In this scenario, the player starts with a bag full of unsorted letters. These can be polite or rude, but he won't know which until he has examined them. What's more, he is allowed to sort the letters, in which case a group of letters will be shown as (for instance) "two polite letters"; but a group of mixed letters, even if they have all been read, will be called "unsorted letters".

Further, the player should be allowed to refer to sorted letters by tone, but not unsorted letters.

To do this, we'll need printing the name... and printing the plural name..., as well as some special understanding rules.

"Hudsucker Industries"

Tone is a kind of value. The tones are effusive, affectionate, polite, curt, and flamingly rude.

A letter is a kind of thing. The description of a letter is usually "On inspection, it turns out to be quite [tone]." A letter has a tone. The tone of a letter is usually polite.

A letter can be read or unread. A letter is usually unread. Carry out examining a letter: now the noun is read.

Before printing the name of a read letter: say "[tone]".

Before printing the name of an ungrouped letter: say "random".

Before printing the plural name of a letter (called the subject):

if the subject is grouped:

say "[tone]";

otherwise if the number of unread letters which are next to the subject is 0:

say "unsorted".

After printing the plural name of a letter (called the subject):

- if the number of read letters which are next to the subject is 0, say " (all unread, at the moment)" instead;

- if the number of unread letters which are next to the subject is greater than 0, say " (some as yet unread)" instead.

Proximity relates a thing (called X) to a thing (called Y) when the holder of X is the holder of Y. The verb to be next to means the proximity relation.

Definition: a letter is grouped:

- if it is unread, no;

- if the number of unread letters next to it is greater than 0, no;

- repeat with item running through letters which are next to it:

 - if the tone of item is not the tone of it, no;

 - yes.

Definition: a letter is ungrouped if it is not grouped.

The Mailroom is a room. "Usually a thrumming hive of bee-like workers, but you got in early to get a jump on the day's work."

The satchel is carried by the player. Two flamingly rude letters are in the satchel. Five polite letters are in the satchel.

The mail wall are fixed in place in the mailroom. "Before you is a wall of mailboxes, including [a list of mailboxes which are part of the mail wall]."

The plural of mailbox is mailboxes. A mailbox is a kind of container. The CEO box is a mailbox. The Hold box is a mailbox. The Trash box is a mailbox. Understand "mailbox" as a mailbox.

Now, there's a good bit of interaction to streamline. We intend that the player will be taking letters from the satchel, reading them, and putting them (perhaps grouped) into boxes. Our interaction rules should assist in this process as much as possible. To start with, the player will be most likely to examine letters he hasn't read yet:

Does the player mean examining a letter (called the subject):

- if we have examined the subject, it is very unlikely;

- it is very likely.

The rules about taking are more subtle: the player is more likely to want to take an ungrouped letter than a grouped one; he is more likely to want one from the satchel than not; and he is most unlikely to want to take a letter (grouped or ungrouped) that he is already holding.

Does the player mean taking a letter (called subject) which is grouped:

- if the player carries the subject, it is very unlikely;

- if the subject is in the satchel, it is possible;

- it is unlikely.

Does the player mean taking a letter (called subject) which is ungrouped:

- if the player carries the subject, it is very unlikely;

- if the subject is in the satchel, it is very likely;

- it is possible.

And finally, we will assume by default that anything other than examining or taking is most likely to apply to a letter he's already identified:

Does the player mean doing something other than examining or taking with a letter (called the subject):
if we have examined the subject, it is likely;
it is unlikely.

And we would also like to understand properties under the same circumstances as printing -- a letter will be identifiable as "polite" if it's already been read and it is either by itself or in a sorted stack of polite letters, but otherwise not. What's more, to make it possible to disambiguate commands in the other direction, we'll call any unsorted letter "random", to represent that the player doesn't know what it is.

Understand the tone property as referring to a letter when the item described is grouped. Understand "random" as a letter when the item described is ungrouped.

When play begins:
now every mailbox is part of the mail wall;
repeat with switch count running from 1 to 5:
move a random letter to the satchel.

Test me with "inventory / examine letter / get letter / i / put letter in ceo box /
inventory / get letter / x letter / g / g / i / x letter / g / g / i / put letter in hold box /
get letter / g / g / i".

That last "repeat" is merely a device to shuffle the order of items in the satchel so that the player will not always encounter the letters in a neatly presorted order, despite our defining them that way. (Of course, that means that the test produced by TEST ME cannot be very exciting...)

35



Example Fallout Enclosure

WI

Adding an enclosure kind that includes both containers and supporters in order to simplify text that would apply to both.

It may not be immediately obvious why we might want to create new intermediate categories of the kinds hierarchy. But there may be times, for instance, where we would like to make an action that applies in the same way to both containers and supporters, but to nothing else in the game. To avoid creating two nearly-identical rules, we would instead roll the two categories together into one, on the principle that duplicating source text is usually a sign of bad design.

So for instance let's say the player is able to zap objects to make them go away, but any contents -- things inside a container or on top of a supporter -- should always be left as residue. Here's one way we might do this:

"Fallout Enclosure"

Section 1 - Procedure

An enclosure is a kind of thing. A container is a kind of enclosure. A supporter is a kind of enclosure.

Understand "zap [something]" as zapping. Zapping is an action applying to one thing. The Zapping action has a list of things called the remnants.

Carry out zapping an enclosure:
if the noun holds something:
now the remnants is the list of things held by the noun;
repeat with N running through the remnants:
move N to the holder of the noun.

Carry out zapping:
now the noun is nowhere.

Report zapping:
say "You zap [the noun], destroying [them][if the remnants is not empty] and leaving [the remnants with indefinite articles] behind[end if]."

Section 2 - Scenario

SuperDuperMart is a room. SuperDuperMart contains some shelves and a cash register.

The shelves support a bottle of Buffout and a container of Jet.

The cash register contains some prewar money, a coin purse, and a bottle cap. The coin purse contains a prewar nickel. It is closed.

The cash register is closed and locked.

Test me with "zap shelves / zap buffout / zap register / zap purse".

56

★ Example Being Prepared

WI

A kind for jackets, which always includes a container called a pocket.

"Being Prepared"

A jacket is a kind of thing. A jacket is always wearable.

A pocket is a kind of container. A pocket is part of every jacket. The carrying capacity of a pocket is always 2.

After examining a jacket:
let target be a random pocket which is part of the noun;
say "[The target] contains [a list of things in the target]."

Now we've created the rules that will govern any specific jackets we might happen to put in our game: each one will always have one pocket, which will be able to contain no more than two things. The description of "a list of things" is text with a list, which we will learn about further in a few sections.

Next we might want to create the environment and an actual example of the jacket kind:

Tent is a room. "A dome made of two flexible rods and a lot of bright green ripstop nylon. It bills itself as a one-man tent, but you'd call it a two-dwarf tent: there is no way to arrange yourself on its square floor so that you can stretch out completely."

The hoodie is a jacket. "Your hoodie is balled up in the corner." The description of the hoodie is "Both elbows are stained from yesterday's entrenching project."

The hoodie's pocket contains a Swiss army knife and a folded map. The hoodie is in the Tent.

Notice that, since Inform has created a pocket for the hoodie, we can now refer to it by name in our source, giving it any additional properties we need to define. Here we simply put a few items into it.

The player wears a whistle. The description of the whistle is "To frighten bears."

Test me with "x hoodie / get hoodie / get knife / get map / i / put hoodie in pocket / put whistle in pocket / put map in pocket / put knife in pocket / i".

Notice that Inform automatically refuses to put the hoodie into its own pocket: as a default, a container cannot contain something of which it is itself a part.

332

★ Example Some Assembly Required

WI

Building different styles of shirt from component sleeves and collars.

We now have the mechanisms in place to do some fairly sophisticated renaming of objects. For instance:

"Some Assembly Required"

Garment type is a kind of value. The garment types are vest, t-shirt, polo shirt, mandarin blouse, button-down, shell, experiment.

Every turn:
 assign identities.

When play begins: assign identities.

To assign identities:
 repeat with item running through torsos:

reassess item.

To reassess (item - a torso):

if the number of things which are part of the item is 0:
now garment type of the item is vest;
rule succeeds;
if exactly two short sleeves are part of the item:
if a collar is part of the item,
now garment type of the item is polo shirt;
otherwise now garment type of the item is t-shirt;
rule succeeds;
if exactly two long sleeves are part of the item:
if a collar is part of the item,
now garment type of the item is button-down;
otherwise now garment type of the item is mandarin blouse;
rule succeeds;
if a collar is part of the item and the number of sleeves which are part of the
item is 0, now garment type of the item is shell;
otherwise now garment type of the item is experiment.

Before cutting something which is worn by the player:
try taking off the noun.

Instead of cutting something when something is part of the noun:
say "You cut up [the noun], snipping off [a list of things which are part of the
noun].";
now every thing which is part of the noun is in the holder of the noun.

Instead of cutting something which is part of something:
say "You carefully snip [the noun] free.";
now the player carries the noun.

Rule for printing the name of a torso: say "[garment type]".

A torso is a kind of thing. A torso is always wearable. Understand "shirt" or
"blouse" as a torso. A torso has a garment type. Understand the garment type
property as describing a torso. A sleeve is a kind of thing. A short sleeve is a
kind of sleeve. A long sleeve is a kind of sleeve. A collar is a kind of thing.

Understand "sew [something] to [something]" as affixing it to. Affixing it to is an
action applying to two things. Carry out affixing something to something: now the
noun is part of the second noun. Report affixing something to something: assign
identities; say "You sew [the noun] on, creating [a second noun]." Understand
the command "stitch" as "sew".

Instead of affixing something to something when the second noun is worn: say
"You're wearing [the second noun]!"

Instead of affixing a torso to something:
if the second noun is a torso, say "Couture for Siamese twins is a daring field,
but a bit of a niche market."
otherwise try affixing the second noun to the noun.

Instead of affixing a sleeve to something when at least two sleeves are part of
the second noun: say "[The second noun] already sports [a list of sleeves that

are part of the second noun]."

Instead of affixing a collar to something when a collar is part of the second noun: say "[The second noun] already sports [a list of collars that are part of the second noun]."

Instead of examining something when something is part of the noun: say "Stitched to [the noun] [is-are a list of things which are part of the noun]."

Here is where the issue of precedence arises. We want to encourage Inform to select a cuttable object that is part of something else, rather than one of the spares:

Definition: a thing is removable if it is part of something. Understand "cut [removable thing]" as cutting.

The Boutique is a room. "Still festively strewn with the confetti and streamers of the Grand Opening party, and still almost totally customer-free."

The player carries a torso. The player carries three short sleeves. The player carries two long sleeves. The player carries two collars.

Test me with "sew collar to shirt / i / sew short sleeve to shirt / g / i / x polo shirt / cut collar / i / cut shirt / sew long sleeve to shirt / i / sew long sleeve to shirt / i / sew collar to shirt / g / i / wear button-down".

338

★ **Example Hays Code**

WI

Clark Gable in a pin-striped suit and a pink thong.

The following burlesque was considered too much for the tender readers of Chapter 3, since it involved explicit use of listing and persuasion:

"Hays Code"

The Movie Set is a room. Clark Gable is a man in the Movie Set. "Clark leans on a polystyrene pillar, wearing [a list of unconcealed things worn by Clark] with his usual aplomb." Persuasion rule for asking Clark to try doing something: persuasion succeeds.

Clark is wearing a pin-striped suit and a pink thong. Rule for deciding the concealed possessions of Clark: if the particular possession is the thong and Clark is wearing the suit, yes; otherwise no.

Test me with "clark, remove suit / look / clark, remove thong / look".

242

★★ **Example What Not To Wear**

WI

A general-purpose clothing system that handles a variety of different

clothing items layered in different combinations over different areas of the body.

"What Not To Wear"

Section 1 - Overlying and Underlying

We start by borrowing some of the same ideas from the Bogart example, but we're also going to make a kind called "garment-element". This kind will include both garments (objects of clothing) and body parts (things that can be covered by clothing); using it allows us to restrict the way our underlying and overlying relations apply, which will make them a bit faster at run-time.

A garment-element is a kind of thing.

Underlying relates various garment-elements to various garment-elements with fast route-finding. The verb to underlie means the underlying relation. The verb to be under implies the underlying relation.

Check taking off:

if the noun underlies something (called the impediment) which is worn by the player, say "[The impediment] [are] in the way." instead.

Carry out taking off:

now the noun is not underlaid by anything.

Report taking off something:

say "[We] [are] now wearing [a list of uppermost things worn by the player]." instead.

Definition: a garment-element is uppermost if it is not under something opaque.

Here we've expanded on the previous ideas of 'uppermost' because it is possible for an upper layer to reveal what lies beneath: a tie, a clear plastic trenchcoat, an open-knit sweater, etc. We'll make such items transparent.

Before taking off something which underlies something which is worn by the player:

while the noun underlies something (called the impediment) which is worn by the player:

say "(first removing [the impediment])[command clarification break]";
silently try taking off the impediment;
if the noun underlies the impediment, stop the action.

Overlying relates various garment-elements to various garment-elements. The verb to overlie means the overlying relation.

Covering relates a garment-element (called A) to a garment-element (called B) when the number of steps via the overlying relation from A to B is greater than 0. The verb to cover means the covering relation.

Before wearing something when a garment which covers the noun is worn by the player:

while the player wears a garment (called the impediment) which covers the noun:

say "(first removing [the impediment])[command clarification break]";
silently try taking off the impediment;
if the player is wearing the impediment, stop the action.

Carry out wearing:

repeat with hidden item running through things worn by the player:
if the noun covers the hidden item, now the hidden item underlies the noun.

Instead of looking under something which is worn by the player:

if something (called the underwear) underlies the noun, say "[We] [peek] at [the underwear]. Yup, still there.";
otherwise say "Just [us] in there."

Instead of taking inventory:

say "[if the player carries something][We]['re] carrying [a list of things carried by the player][else][We]['re] empty-handed[end if][if the player wears something]. [We] [are] wearing [a list of uppermost garments worn by the player][end if]."

To peek is a verb.

Section 2 - Regional Coverage

Here we draw in the idea that different clothes go over different areas of the body, and that they should be in competition with each other only if both sets of clothes belong at the same level over the same body area.

Before wearing something:

let N be the layering depth of the noun;
repeat with item running through things worn by the player:
if the layering depth of the item is N and the item covers a body-part which is covered by the noun:
say "(first taking off [the item])[command clarification break]";
silently try taking off the item;
if the player wears the item, stop the action.

This may seem like overkill, but it allows us to create garments that cover different subsets of the body -- pants and shirt vs. a dress, for instance.

To decide what number is the layering depth of (chosen garment - a thing):

let N be 0;
if the chosen garment covers a body-part (called base):
let N be the number of steps via the overlying relation from the chosen garment to the base;
decide on N.

To help with modeling, we'll give everyone body parts, broken down according to their relevance to clothing:

A body-part is a kind of garment-element. A torso, a seat, a head, pair of legs, and pair of feet are kinds of body-part.

If we wanted to allow gloves, we might put in hands as well; but this is enough for now.

One head is part of every person. One torso is part of every person. One pair of legs is part of every person. One pair of feet is part of every person. One seat is part of every person.

And now we make some categories of clothing:

A garment is a kind of garment-element. A garment can be transparent. A pair of pants, a pair of underpants, a foundation garment, a pair of socks, a pair of shoes, a jacket, a hat, a dress, and a shirt are kinds of garment.

The plural of pair of pants is pairs of pants. The plural of pair of underpants is pairs of underpants. The plural of pair of socks is pairs of socks. The plural of pair of shoes is pairs of shoes.

A pair of pants, a pair of underpants, a foundation garment, a pair of socks, a pair of shoes, a jacket, a hat, a dress, and a shirt are usually wearable.

When play begins:

now every pair of socks overlies every pair of feet;
now every pair of shoes overlies every pair of socks;
now every pair of underpants overlies every seat;
now every pair of pants overlies every pair of underpants;
now every foundation garment overlies every torso;
now every jacket overlies every shirt;
now every jacket overlies every dress;
now every hat overlies every head;
now every dress overlies every pair of underpants;
now every dress overlies every foundation garment.

Section 2 - The Scenario

The Dressing Room is a room.

The player carries some capris, some jeans, a corset, a plunge bra, a thong, boy-shorts, black satin D'Orsay pumps, brown leather boots, a camisole, a cocktail dress, a bolero, a cashmere shrug, a sheer wrap, and a linen tunic.

The woolly socks are a pair of socks.

The D'Orsay pumps and the brown leather boots are pairs of shoes.

The thong and the boy-shorts are pairs of underpants.

The capris and the jeans are pairs of pants.

The tunic is a shirt.

The camisole, the corset, and the plunge bra are foundation garments.

The cocktail dress is a dress.

The bolero, the cashmere shrug, and the sheer wrap are jackets. The shrug and the wrap are transparent.

Test me with "wear capris / wear jeans / i / wear thong / i / wear dress / wear corset / wear dress / i / wear wrap / i / wear boots / wear pumps / i".



Example Get Me to the Church on Time

Using kinds of clothing to prevent the player from wearing several pairs of trousers at the same time.

Inform's default handling of wearable things does not make any rules about what can be worn together. Suppose, however, we have a game in which there are a large number of different garments, and we want to keep the player from wearing (say) more than one pair of pants at once:

"Get Me to the Church on Time"

A pair of pants and a shirt are kinds of thing. A pair of pants and a shirt are usually wearable.

Some golf pants are a pair of pants. The description is "Checked in red and green, with tiny frolicking gophers every few inches."

Some tuxedo trousers are a pair of pants. The description is "Black, pressed, and slimming."

The frilly shirt is a shirt. The description of the frilly shirt is "She insisted."

The polo shirt is a shirt. The description is "Turquoise and bright yellow, the colors selected by your golfing buddies."

The player wears the golf pants and the polo shirt. The player carries the tuxedo trousers and the frilly shirt.

The Wedding Chapel Dressing Room is a room. "The bride's dressing room is a lavish suite with closets, hangers, dressmaker's dummies, boxes of straight pins and sewing notions, combs, lotions, brushes, and hair fixatives, plus room for fifteen female attendants and a photographer. Before they shoved you out of the room you even got a glimpse of a small reference library including '1001 French Braids' and 'Corset-Lacing For Beginners.'

This is the groom's dressing room. You get a framed photograph of Elvis, a dusty mirror, and the floor space of an average toilet stall."

The dusty mirror and the photograph of Elvis are scenery in the Dressing Room. The description of the mirror is "You can't really get more than a silhouette impression of yourself." The description of Elvis is "He reminds you that you'd better get out there before the organist switches to Hound Dog."

And now the rule itself, borrowed from a later chapter:

Instead of wearing a pair of pants when the player is wearing a pair of pants (called the wrong trousers):

say "You'll have to take off [the wrong trousers] first."

Instead of wearing a shirt when the player is wearing a shirt (called the wrong top):

say "You'll have to take off [the wrong top] first."

When play begins:

say "From the other side of the door, you hear the organist move on from his instrumental interpretation of 'I Wanna Hold Your Hand' to a somewhat more spirited rendition of 'Help! I Need Somebody!'. Okay, okay, but you've been rushing things along since the 16th fairway, and you can't be more than a half-hour late... Surely that mother of hers can't blame you for that?"

Test me with "i / x trousers / wear trousers / x golf pants / take off golf pants / wear trousers / x frilly shirt / x polo shirt / wear frilly shirt / doff polo shirt / wear frilly shirt".

If we wanted to, we could make similar kinds for hats, shoes, and so on, and have a simple but effective system of clothing. A more complicated treatment might keep track of layering and describe the player's outfit differently depending on which clothes were outermost -- an example for a later chapter.

234



Example Bogart

WI

Clothing for the player that layers, so that items cannot be taken off in the wrong order, and the player's inventory lists only the clothing that is currently visible.

We have two things to keep track of with our layering clothing: what currently is covering something else; and what *can* cover something else. This implementation goes for a fairly simple treatment, assuming that each item of clothing will completely conceal those beneath it, and that we are not implementing entire sets of shirts, jackets, etc. But it will do for a demonstration.

"Bogart"

Section 1 - Clothing Behavior

First we make our relation to represent what **is** underneath another item:

Underlying relates one thing to various things. The verb to underlie means the underlying relation. The verb to be under implies the underlying relation.

And now we prevent taking a lower layer off before the thing that is worn over it:

Before taking off something which underlies something (called the impediment) which is worn by the player:

say "(first removing [the impediment])[command clarification break]";
silently try taking off the impediment;
if the noun underlies something which is worn by the player, stop the action.

Check taking off:

if the noun underlies something (called the impediment) which is worn by the player, say "[The impediment] [are] in the way." instead.

Carry out taking off:

now the noun is not underlaid by anything.

Report taking off something:

say "[We] [are] now wearing [a list of uppermost things worn by the player]."
instead.

Definition: a thing is uppermost if it is not under something.

That covers order of clothing removal, but we also want to restrict what can be worn on top of what else. This time we need Inform to have some idea of what customarily can be layered on top of what other clothing:

Overlying relates one thing to various things. The verb to overlie means the overlying relation.

Covering relates a thing (called A) to a thing (called B) when the number of steps via the overlying relation from A to B is greater than 0. The verb to cover means the covering relation.

With these definitions, we can say that a jacket should go over a shirt and a shirt over an undershirt (say), and then Inform will know that a jacket will cover both shirt and undershirt.

Before wearing something when something (called the impediment) which covers the noun is worn by the player:

while the player wears something which covers the noun:

say "(first removing [the impediment])[command clarification break]";

silently try taking off the impediment;

if the player is wearing the impediment, stop the action.

Carry out wearing:

if the noun covers something (called the hidden item) worn by the player, now the hidden item underlies the noun.

Instead of looking under something which is worn by the player:

if something (called the underwear) underlies the noun, say "[We] [peek] at [the underwear]. Yup, still there.";

otherwise say "Just [us] in there."

Instead of taking inventory:

say "[if the player carries something][We][re] carrying [a list of things carried by the player][else][We][re] empty-handed[end if][if the player wears something]. [We] [are] wearing [a list of uppermost things worn by the player][end if]."

To peek is a verb.

Notice that our inventory only describes the things that the player can see as the upper layer of clothing.

Section 2 - The Scenario

The Trailer is a room. "A full-length mirror is the main amenity in here, and that suits you just fine." The full-length mirror is scenery in the Trailer. Instead of examining or searching the mirror, try taking inventory.

The player wears a fedora, a jacket, a shirt, some undershorts, an undershirt, some slacks, a pair of socks, and a pair of shoes.

The shirt underlies the jacket. The pair of socks underlies the pair of shoes. The undershorts underlie the slacks. The undershirt underlies the shirt.

The jacket overlies the shirt. The shoes overlie the socks. The slacks overlie the undershorts. The shirt overlies the undershirt.

Test me with "x mirror / remove fedora / remove jacket / remove shirt / remove slacks / remove undershirt / remove shoes / remove socks / remove shorts / remove undershorts".

If we further wanted to prevent the player from taking off clothes in inappropriate places, we might add something like this:

Instead of taking off something in the presence of someone who is not the player:

say "[We] [are] far too modest to strip in public."

263

Example Widget Enterprises

WI

Allowing the player to set a price for a widget on sale, then determining the resulting sales based on consumer demand, and the resulting profit and loss.

Suppose the player is responsible for pricing at Widget Enterprises. Widget production entails a certain fixed cost as well as a cost per unit; and somewhere out in the world there are a number of customers interested in purchasing widgets, but the player starts without knowing what this distribution looks like.

We can express the profits as an equation: the total made by selling widgets, minus the cost thereof.

The Table of Customers holds the data about customer preferences, and whenever the player selects a widget price, we consult it to determine how many customers in total would be willing to buy at that price.

"Widget Enterprises"

Widget Stand is a room.

A monetary value is a kind of value. \$1.99 specifies a monetary value with parts dollars and cents.

Equation - Profit Equation

$$P = nV - (F + nC)$$

where P is a monetary value, F is the fixed cost, C is the unit cost, V is a monetary value, and n is a number.

The fixed cost is a monetary value that varies. The fixed cost is \$5.00.
The unit cost is a monetary value that varies. The unit cost is \$10.66.

Table of Customers

	base maximum value
2	\$26.00
5	\$20.00
8	\$15.00
2	\$13.50
1	\$9.00

To decide what number is the units sold at (V - a monetary value):
let total units be 0;
repeat through the Table of Customers:
 if V is less than the maximum value entry:
 increase total units by the base entry;
decide on total units.

Understand "set price to [monetary value]" as setting price to. Setting price to is an action applying to one monetary value.

Carry out setting price to:
 let V be the monetary value understood;
 let n be the units sold at the monetary value understood;
 let P be given by the Profit Equation;
 say "You set the price of your widgets to [V], resulting in sales of [n] unit[s] and
 ",
 if P is less than \$0.00:
 let L be \$0.00 - P ;
 say "a loss of [L].";
 otherwise if P is \$0.00:
 say "break even.";
 otherwise:
 say "a profit of [P].".

Test me with "set price to \$0.00 / set price to \$100.00 / set price to \$15.00 / set price to \$8.00 / set price to \$25.00 / set price to \$14.99".

As written this will be a rather dull guessing game for the player; more interesting would be to enhance it into a fuller economic simulator with more control over fixed costs and customer price points.

264

★ Example Frozen Assets

WI

A treatment of money which keeps track of how much the player has on him, and a BUY command which lets him go shopping.

In our brave new world, everything will have a price, so we had better spell this out.

"Frozen Assets"

Price is a kind of value. \$10.99 specifies a price. A thing has a price. The price of a thing is usually \$0.00. After examining something for sale, say "It can be yours for [the price of the noun]."

Now we assume a simple shopping model in which the player can't take anything without paying for it.

Definition: a thing is free if the price of it is \$0.00.

Definition: a thing is for sale if it is not free.

Instead of taking something for sale:

say "You'll have to pay for that."

Before buying something for sale when the money is not in the wallet:

say "You're broke." instead.

Before buying something for sale when the money is free:

say "You're broke." instead.

Before buying something for sale when the price of the money is less than the price of the noun:

say "Your funds do not run to the price of [the noun]." instead.

Instead of buying something:

decrease the price of the money by the price of the noun;

say "You fork over [the price of the noun] for [the noun], leaving yourself with [the price of the money].";

if the money is free:

now the money is nowhere;

now the price of the noun is \$0.00;

now the player is carrying the noun.

The player's money object is going to be a bit unusual, because it has value but cannot itself be bought.

The player carries a wallet. The wallet contains money. The price of the money is \$4.50. The printed name of the money is "[price of the money] in cash".

Understand "cash" as the money.

Instead of taking the money:

say "Best to leave it alone until you need to buy something."

Instead of buying something free:

say "[The noun] is yours already."

Instead of buying the money:

say "The money belongs to you; you buy things with it."

Now we just need something to buy.

The Dessert Parlor is a room. "An underlit, over-crowded room campily furnished with a lot of gilt-frame mirrors and twinkle lights: it is essentially a brothel of food. The service is slow at best, and on Saturday nights glacial. However. The wares on display more than make up for these trivial inconveniences."

The vanilla ice cream is an edible thing in the Parlor. The price of the ice cream is \$2.45. The description is "In the scale of ice creams, you recognize this as a very inferior vanilla because it has no adjectives in the title."

The raspberry tart is an edible thing in the Parlor. The price of the tart is \$4.50. The description is "An almond-laced shell packed with raspberries-under-glaze."

The syllabub is an edible thing in the Parlor. The price of the syllabub is \$4.25. The description is "Whipped cream, alcohol, and lime juice, a substance without any redeeming food value whatever."

The espresso cake is an edible thing in the Parlor. The price of the espresso cake is \$5.50. The description is "A lethal wedge of purest blackness."

Test me with "inventory / examine syllabub / get syllabub / buy syllabub / drop it / get it / buy raspberry tart".

Implementing caloric units for this scenario is left as an exercise for the reader.

265



Example Money for Nothing

WI

An OFFER price FOR command, allowing the player to bargain with a flexible seller.

"Money for Nothing"

Section 1 - Prices and Bargaining

Price is a kind of value. \$10.99 specifies a price with parts dollars and cents (optional, preamble optional).

A person has a price called wealth. The wealth of the player is \$15.

A thing has a price called minimum value. The minimum value of a thing is usually \$0.50.

A thing has a price called desired value. The desired value of a thing is usually \$5.00.

Offering it for is an action applying to one price and one visible thing.

Understand "offer [price] for [something]" as offering it for.

After taking inventory, say "You have [the wealth of the player]."

Check offering it for:

if the price understood is greater than the wealth of the player, say "You don't have that kind of cash." instead;

if the second noun is not carried by someone, say "There's no one in a position to sell you [the second noun]." instead;

if the second noun is carried by the player, say "[The second noun] is already

yours." instead;

if the minimum value of the second noun is greater than the price understood,
say "[The holder of the second noun] cackles disdainfully. 'If yer just here to
insult me you can take your business elsewhere!' he says." instead;

if the desired value of the second noun is greater than the price understood:
let difference be the desired value of the second noun minus the price
understood;

let difference be difference divided by two;

decrease the desired value of the second noun by difference;

now the last object offered is the second noun;

say "'How about [desired value of the second noun]?' suggests [the holder
of the second noun]." instead;

otherwise:

unless the desired value of the second noun is the price understood:

say "From the avaricious gleam in the eye of [the holder of the second
noun], you guess you could've gotten this purchase for less..."

Carry out offering it for:

increase the wealth of the holder of the second noun by the price understood;

decrease the wealth of the player by the price understood;

move the second noun to the player.

Report offering it for:

say "You spend [the price understood], and now you possess [the second
noun]."

When play begins: now right hand status line is "Your funds: [wealth of the
player]".

Now, since the man does make counter-offers, it would be reasonable to let the
player accept or reject those, as well:

The last object offered is a thing that varies.

Instead of saying yes when the last object offered is carried by a person (called
seller) who is not the player:

if the seller is not visible:

continue the action;

otherwise:

now the price understood is the desired value of the last object offered;

try offering the desired value of the last object offered for the last object
offered.

Instead of saying no when the last object offered is carried by a person (called
seller) who is not the player:

if the seller is not visible:

continue the action;

otherwise:

now the last object offered is the player;

say "You reject the offer firmly."

And we borrow just a line or two from a later chapter to take care of some alternate
syntax the player might try:

Understand "offer [price] to [someone]" as a mistake ("You'll need to specify what you want to buy -- try OFFER \$1000.00 FOR BROOKLYN BRIDGE.").
Understand "offer [someone] [price]" as a mistake ("You'll need to specify what you want to buy -- try OFFER \$1000.00 FOR BROOKLYN BRIDGE.").

Understand "buy [something]" as a mistake ("You'll have to name your price: try OFFER \$1000.00 FOR BROOKLYN BRIDGE.").

Section 2 - The Scenario

The Flea Market is a room. The crotchety man is a man in the Market. "A crotchety man here is selling [the list of things carried by the crotchety man]."
The crotchety man carries a broken television set, a Victorian rhinestone brooch, and a cracked shaving mug.

The minimum value of the brooch is \$2.50.

Test me with "offer \$0.50 for mug / offer \$0.50 to man / offer \$6.00 for mug / offer \$50.00 for brooch / offer \$1.50 for brooch / offer \$4.50 for brooch / no / offer \$4.50 for brooch / yes".

269



Example Fabrication

WI

A system of assembling clothing from a pattern and materials; both the pattern and the different fabrics have associated prices.

When we make a new kind of value, the new named values can themselves have properties. That is convenient because, for instance, we might want to associate a material (itself the property of an object) with certain features, such as price.

"Fabrication"

Section 1 - Procedure

A material is a kind of value. The materials are silk, velvet, cotton, and wool.

Price is a kind of value. \$1.99 specifies a price.

Area is a kind of value. 5 sq yards specifies an area.

Cost is a kind of value.. \$1.99 per sq yard specifies a cost. A cost times an area specifies a price.

A material has a cost.

The cost of silk is usually \$5.75 per sq yard. The cost of velvet is usually \$9.50 per sq yard. The cost of cotton is usually \$2.29 per sq yard. The cost of wool is usually \$4.75 per sq yard.

A pattern is a kind of thing. A pattern has a material. A pattern has an area. A pattern has a price. The price of a pattern is usually \$9.99. Understand "pattern"

as a pattern. Understand "patterns" as the plural of a pattern.

After printing the name of a pattern:

if planning:
do nothing;
otherwise:
say " pattern".

To decide what price is the material price of (chosen item - pattern):

let C be the cost of the material of the chosen item;
let A be the area of the chosen item;
decide on $C * A$.

To decide what price is the overall price of (chosen item - pattern):

let P be the price of the chosen item;
let M be the material price of the chosen item;
decide on $P + M$.

Understand "plan [material] [pattern]" as planning it for.

Planning it for is an action applying to one material and one thing.

Carry out planning it for:

now the material of the second noun is the material understood.

Report planning it for:

say "You lay plans for a [material understood] [second noun], running [material price of the second noun] for materials and [price of the second noun] for the pattern itself, for a total of [overall price of the second noun]."

Section 2 - Scenario

Joanne's Fabrics is a room. Joanne's Fabrics contains a pattern bin.

The cape is a pattern. The material of the cape is velvet. The area of the cape is 9 sq yards.

The bodice is a pattern. The material of the bodice is silk. The area of the bodice is 2 sq yards. The price of the bodice is \$11.99.

The cape and the bodice are in the pattern bin.

Test me with "plan silk bodice / plan velvet bodice / plan velvet cape / plan wool cape".



A more intricate system of money, this time keeping track of the individual denominations of coins and bills, specifying what gets spent at each transaction, and calculating appropriate change.

Typically games which keep track of the player's wealth need only do so as an abstract number, but occasionally it becomes useful to represent money as physical coins and bills. Here is an example that does exactly that:

"Nickel and Dimed"

Section 1 - Currency

Price is a kind of value. \$10.99 specifies a price with parts dollars and cents. A thing has a price. The price of a thing is usually \$0.00.

Money is a kind of thing. Coin is a kind of money.

A dollar bill is a kind of money. The price of a dollar bill is \$1.00. The printed name of a dollar bill is "dollar bill". Rule for printing the plural name of a dollar bill: say "dollar bills". The description of a dollar bill is "It has George Washington's head on the front, which with a bit of creative folding can be scrunched to look like a mushroom. All important things really are learned in kindergarten."

A five-dollar bill is a kind of money. The price of a five-dollar bill is \$5.00. Understand "five" or "five dollar" as the five-dollar bill. The description of a five-dollar bill is "Abraham Lincoln. He looks slightly less dignified here than he does on the penny."

A hundred-dollar bill is a kind of money. The price of the hundred-dollar bill is \$100.00. Understand "hundred" or "hundred dollar" as the hundred-dollar bill. Understand "dollar" as the dollar bill. The description of a hundred-dollar bill is "It's got Benjamin Franklin, who always gets shafted: a denomination too large for anyone to carry conveniently, and a lot of local fame in Philadelphia."

Our choice of understand rules guarantees that "five dollar" will be understood as the five, but "dollar" alone as the single. We will learn more about "understand" in later chapters, but here is a test to check the functionality:

Test bills with "x hundred dollar bill / x five dollar bill / x hundred / x five / x dollar / x dollar bill".

A quarter is a kind of coin. The price of a quarter is \$0.25. The description of a quarter is "One of the old-fashioned variety, not a state quarter."

A dime is a kind of coin. The price of a dime is \$0.10. The description of a dime is "Franklin Roosevelt, trying not to look too annoyed that his coin is so small and thin."

A nickel is a kind of coin. The price of a nickel is \$0.05. The description of a nickel is "A chubby coin, but you've always liked Thomas Jefferson, and the Monticello on the back is a nice touch."

A penny is a kind of coin. The price of a penny is \$0.01. The description of the penny is "A profile of Abe Lincoln. Sometime soon they'll stop minting these, you're sure of it."

Section 2 - Ownership

Ownership relates one person (called the owner) to various things. The verb to own means the ownership relation.

Definition: a thing is owned if the player owns it.

Instead of buying something which is owned by the player:
say "You already own [the noun]."

Instead of going somewhere when the player encloses something (called the stolen goods) which is not owned by the player:

if the owner of the stolen goods is not a person:
now the player owns the stolen goods;
continue the action;
if the owner of the stolen goods can see the player,
say "'Hey there buddy, not so fast,' says [the owner of the stolen goods].
'You going to buy [the stolen goods] first, or am I gonna call the cops?";
otherwise continue the action.

After taking inventory:

say "Altogether, you've got [the player's cash] on your person."

To decide what price is the player's cash:

let sum be the total price of money enclosed by the player;
decide on sum.

To decide what price is the sum in (item - a container):

let sum be the total price of the money in the item;
decide on sum.

When play begins: now every thing carried by the player is owned by the player.

Section 3 - Purchasing and Sales

Definition: a thing is worthless if the price of it is \$0.00. Definition: a thing is valuable if it is not worthless.

A thing can be for sale.

Rule for printing room description details of something (called target) which is for sale (this is the disclose prices in room description rule): say " ([price of the target])".

Before listing contents: group money together giving articles.

Instead of examining a for sale thing (this is the describe things by price rule):

say "[The noun] costs [the price of the noun], payable to [the owner of the noun]."

The cashbox is a theoretical construct, not something the player will ever encounter in the course of the game. It contains all the money that is available for non-player characters to use in making change. If we wanted, we could give each character his own stash of change, but this would increase the likelihood that any given person

would run out of cash to make change with. (And in this example there is only one vendor anyway.)

The cashbox is a container. The cashbox contains 10 pennies. The cashbox contains 10 nickels. The cashbox contains 10 dimes. The cashbox contains 10 quarters. The cashbox contains 10 dollar bills. The cashbox contains 10 five-dollar bills.

The block buying rule is not listed in the check buying rules.

Check buying something:

if the noun is not for sale, say "[The owner of the noun] does not want to sell you [the noun]." instead;

if the player's cash is less than the price of the noun, say "You can't afford the asking price of [the price of the noun] for [the noun]." instead.

Carry out buying something:

let sum paid be \$0.00;
while sum paid is less than the price of the noun:
let current target be the price of the noun minus the sum paid;
let bill offered be the best money from the player for the current target;
if the bill offered is money:
move the bill offered to the owner of the noun;
now the bill offered is spent;
increase the sum paid by the price of the bill offered;
let current target be the price of the noun minus the sum paid;
say "You hand [the owner of the noun] [a list of spent money]. [run paragraph on]";
let change be \$0.00;
if the sum paid is greater than the price of the noun:
now the change is the sum paid minus the price of the noun;
if change is greater than the sum in the cashbox:
now the player carries every spent money;
now every spent thing is fresh;
say "'Whoa,' says [the owner of the noun], handing the cash back to you. 'I can't make change for that, man, sorry.'" instead;
now every spent thing is in the cashbox;
now every spent thing is fresh;
while change is greater than \$0.00:
let change bill be the best money from the cashbox for change;
decrease change by the price of the change bill;
now change bill is spent;
move change bill to player;
if money is spent, say "[The owner of the noun] makes change with [a list of spent money]. [run paragraph on]";
now every spent thing is fresh;
if the noun is not enclosed by the player and the owner of the noun can touch the noun:
say "'Here ya go,' says [the owner of the noun], handing [the noun] to you. [run paragraph on]";
move the noun to the player;
now the player owns the noun.

Money can be spent or fresh.

Report buying something:
if the player owns the noun,
say "Your transaction is now complete, leaving you with [the player's cash]."

We've skipped over defining what makes a denomination the best for a given transaction, so we'd better do that now. Our goal is to avoid ever having the player gratuitously overpay -- he should always offer the smallest amount of money that will meet the price of what he's buying.

We also assume that all money "enclosed by the buyer" -- that is, somewhere in the buyer's possession -- is available for use. This might not be true in a game where the player could pick up, say, a sealed lucite container with a ten-dollar bill inside; in that case we would have to define our terms more rigorously, perhaps by requiring that the bills be both enclosed and touchable by the buyer. The touchability check adds an extra layer of calculation, however, and since it is not necessary in this example (and probably not in most other cases either), we'll leave it out:

Definition: money is costly if its price is \$2.50 or more. Definition: money is cheap if its price is \$0.99 or less.

Functional relation is a kind of value. The functional relations are overpayment, underpayment and irrelevant. Money has a functional relation.

To decide what money is the best money from (buyer - a thing) for (cost - a price):

```
repeat with bill offered running through money:  
  if the bill offered is enclosed by the buyer:  
    if the price of the bill offered is the cost, decide on the bill offered;  
    if the price of the bill offered is greater than the cost, now the functional  
    relation of bill offered is overpayment;  
    otherwise now the functional relation of the bill offered is underpayment;  
  otherwise:  
    now the functional relation of the bill offered is irrelevant;  
    [say "underpayment: [a list of underpayment money]  
    overpayment: [a list of overpayment money]";]  
    if the total price of underpayment money is less than the cost:  
      decide on the cheapest money which is overpayment;  
    otherwise:  
      decide on the costliest money which is underpayment.
```

Notice the "say underpayment/overpayment section..." noted out, above. This is for debugging purposes: when writing complex code, it is sometimes useful to put in lines that will say explicitly what is going on. We can enclose them in brackets and Inform will ignore them as though they were comments; if we run into any problems with the code later, we can erase the brackets and see the diagnostic printed to the screen as we play.

Instead of giving money to someone:
say "Best to keep the transaction simple by buying whatever you want."

Section 4 - The Scenario

The player carries 2 dollar bills. The player carries a nickel. The player carries 2 pennies. The player carries a five-dollar bill. The player carries 1 hundred-dollar

bill.

The Subway Station is a room.

The Bitterly Cold Street is north of the Subway Station. "Even though there is no actual snow or ice, the street is about as cold as you can stand, for which reason walking the twenty blocks uptown is not an acceptable option." The Bitterly Cold Street contains a dollar bill.

The newspaper man is a man in the Subway Station. "A newspaper man in a knit cap and fingerless gloves is hopping up and down behind his stand[if the turn count is 1]. Cold weather, caffeine overdose, or mental illness? You may never know. Welcome to New York[end if]." The description is "Eye contact with strangers is something to avoid around here."

The stand is a supporter in the Station. The stand is scenery.

A copy of the New York Times is on the stand. The price of the New York Times is \$1.25.

A pack of gum is on the stand. The price of the gum is \$0.40.

A paperback novel is on the stand. The price of the paperback novel is \$7.99.

A packet of trading cards is on the stand. The price of the packet is \$0.99.

When play begins:

now every thing on the stand is owned by the newspaper man;

now every thing on the stand is for sale.

We could have done all that by hand, but the initialization requires a little less work.

The ticket machine is a container in the Subway Station. It is fixed in place. The description of the ticket machine is "An LED screen on the front instructs you to insert [remaining ticket total] to complete your purchase. You also notice that the NO CHANGE light is lit up." The light is part of the ticket machine. The printed name of the light is "no change light". Understand "no change" or "no change light" as the light.

The description of the light is "In the whole of your recollection, the ticket machine has actually had change a total of twice. Usually, as now, the no-change light gleams angrily, daring you to put in more than you owe." A cash return button is part of the ticket machine. Instead of pushing the cash return button: say "The ticket machine regurgitates [the list of things in the ticket machine]."; now every thing in the ticket machine is carried by the player. Instead of taking something which is in the ticket machine: say "The ticket machine has swallowed your money, but it can be retrieved (you hope) with the cash return button."

Instead of inserting a hundred-dollar bill into the ticket machine:

say "What, are you nuts?"

To decide what price is the remaining ticket total:

let absolute cost be \$2.25;

let remaining cost be absolute cost minus the total price of things in the ticket machine;
if remaining cost is less than \$0.00, decide on \$0.00;
decide on remaining cost.

Instead of inserting something which is not money into the ticket machine: say "The ticket machine only accepts money, not other tokens of your esteem and regard."

Instead of inserting a penny into the ticket machine:
say "The penny rattles out again mockingly: not even the ticket machine thinks these are worth anything."

A subway pass is a kind of thing. 15 subway passes are in the cashbox. The description of a subway pass is usually "A rectangle of thick lavender paper with a black magnetic stripe running up the back side. It is good for one trip on the subway."

After inserting something into the ticket machine:
if the remaining ticket total is \$0.00:
let purchased ticket be a random subway pass in the cashbox;
if purchased ticket is not a subway pass, say "The ticket machine grunts disobligingly and then the unwelcome word MALFUNCTION parades across the LED screen, three letters at a time." instead;
repeat with item running through things in the machine:
now the item is nowhere;
move purchased ticket to player;
say "The ticket machine beeps obligingly and disgorges a single subway pass.";
otherwise:
say "The ticket machine beeps obligingly and adjusts its price down to [remaining ticket total]."

And because even though the ticket machine is a container, we don't want to say (empty) after it in the room description:

Rule for printing room description details of the ticket machine:
do nothing instead.

Test me with "buy novel / n / get dollar / s / buy novel".

After all that, we should probably give the player a chance to win, as well:

The turnstile is south of the Subway Station. "A turnstile is all that separates you from the subway platform stairs." The turnstile is north of the Platform. The turnstile is a door. Before going down in the presence of the turnstile, try going south instead. The turnstile is openable. The turnstile is open.

Instead of going through the turnstile when the player carries a subway pass: say "You enter the turnstile and begin your journey uptown..."; end the story finally saying "At last". Instead of going through the turnstile: say "You can't go through the turnstile without a subway pass. They're very strict about this."

Instead of inserting money into the turnstile: say "The turnstile takes passes, not money." Instead of inserting a subway pass into the turnstile, try entering the

turnstile.

Test more with "buy times / put all but five-dollar bill in machine / press button / buy gum / buy cards / i / put dollar in machine / g / put quarter in machine / i / d".

In fairness to the Metropolitan Transit Authority, we should admit that most of the ticketing machines in the real New York subway are better than this, and will accept, say, a debit card. But that would be so much less exciting to implement.

283



Example Introduction to Juggling

WI

Assortment of equipment defined with price and description, in a table.

Suppose we have a whole catalog-full of equipment that the player might want to purchase and use. We'll start by defining our purchasing rules:

"Introduction to Juggling"

We allow things to have prices, and the player's price to indicate how much money the player has:

Section 1 - Mail-ordering defined

Price is a kind of value. \$100.99 specifies a price.

The player has a price. The price of the player is \$60.00.

Because we're allowing the player to order things that he can't currently see, we need to borrow a special kind of grammar from the Understanding chapter. All our orderable items in this example are toys, so "any toy" means any object of the toy kind, whether or not it is in view at the moment:

Understand "buy [any toy]" as ordering. Understand the command "order" as something new. Understand the command "order" as "buy".

Ordering is an action applying to one visible thing.

Check ordering:

if the cost of the noun is greater than the price of the player, say "You only have [price of the player], while [the noun] would cost [cost of the noun]." instead.

Carry out ordering:

move the noun to the player;
decrease the price of the player by the cost of the noun.

Report ordering:

say "You order [a noun], which is delivered instantly."

We should also handle the situation where the player orders another of something he has already bought and which is right in front of him:

Instead of buying something:
say "You already have [a noun]."

So much for the general rules for this scenario. Now we move on to particulars: the actual items the player is allowed to order. Each item will have a description, a price, and a difficulty representing how skilled the player must be in order to make use of that item.

Since we are going to use price and difficulty in the table that defines our juggling equipment, we need to mention these kinds of value before the line that says how toys are defined.

Section 2 - The Scenario

Difficulty is a kind of value. The difficulties are easy, moderate, hard. The player has a difficulty. The difficulty of the player is easy.

The plural of toy is toys. A toy is a kind of thing. Some toys are defined by the Table of Juggling Equipment.

Table of Juggling Equipment

toy	cost	restriction	description	difficulty	outcome
an economy bounce ball set	\$10.00	"comes in set of three"	"A fairly ordinary rubber ball, solid color."	moderate	"You create of the balls a cascade of moving color."
an acrylic contact ball	\$14.00	"should be bought with ball polish"	"A large clear ball, not for throwing but for using in various hand tricks."	hard	"You rotate the ball between your fingers and pass it over the backs of your hands."
a UV-reactive contact ball	\$55.00	"appears to glow in dark rooms"	"Similar to the ordinary acrylic contact ball, but UV-reactive."	hard	"The ball glows as it passes between your fingers and over the backs of your hands, rolls up to your wrist, snaps through the air-- all apparently of its own accord."
a ball polish set	\$10.00	"useful only with acrylic contact balls"	"Three bottles of polish and a rag for keeping acrylic contact balls scratch-free."	hard	"You juggle the polish bottles with difficulty, since they are full of sloshing liquid."
a teaching beanbag set	\$8.00	"set of three"	"Soft, easily-juggled bag."	easy	"You juggle the beanbags with basic competence."
a stage ball set	\$13.50	"comes in set of three"	"Not much different in appearance from the economy bounce ball, but larger so as to be visible from a stage."	moderate	"You create of the balls a cascade of moving color, visible from quite a distance."
a fireball set	\$33.00	"will not be sold to minors"	"A ball has wicking and a fuel-source inside so that it will burn while being juggled."	hard	"You juggle the fireballs rapidly, careful never to hold any of them a moment longer than necessary."

Notice that we are allowed to define "description" and other already-known properties in the table as well.

Backstage is a room. "A muffled black room with felt on the floors and walls. A glowing sign over the stage door says SHOW IN PROGRESS."

The Juggling Equipment Catalog is a thing in Backstage.

Instead of examining the Catalog:

say "You read through the offerings, including: [paragraph break]";
repeat through Table of Juggling Equipment:
say "[bold type][toy entry][roman type]: [description entry] [cost entry],
[restriction entry]. [paragraph break]".

When play begins:

now right hand status line is "Budget: [price of the player]";
now left hand status line is "[location], feeling [if the difficulty of player is
easy]incompetent[end if][if the difficulty of player is moderate]moderately
skilled[end if][if the difficulty of player is hard]highly skilled[end if]".

And of course this will be no fun unless the player is allowed to use the equipment:

Understand "juggle [something]" as juggling.

Juggling is an action applying to one thing.

Check juggling:

if the noun is not a toy listed in the Table of Juggling Equipment, say "You
can't juggle [a noun]!" instead;
if the difficulty of the noun is greater than the difficulty of the player, say
"You're not quite ready to juggle something like [the noun]. Better to start with an
easier toy." instead.

Carry out juggling:

if a random chance of 1 in 3 succeeds:
if the difficulty of the player is less than hard and the difficulty of the player
is the difficulty of the noun:
now the difficulty of the player is the difficulty after the difficulty of the
player.

Report juggling:

say "[outcome of the noun][paragraph break]".

Instead of burning the fireball set:

say "It will flame by itself when you use it."

Test me with "read catalog / buy economy / buy beanbag / juggle economy /
juggle beanbag / juggle beanbag / juggle beanbag / juggle beanbag / juggle
beanbag / juggle beanbag / juggle economy / juggle economy / juggle economy /
buy fireball set / juggle fireball".

"Do Pass Go"

Go is a room. "A giant square area, where you and your other pewter ornament friends gather before setting out to purchase London."

The pair of dice is carried by the player.

The pair of dice has a number called first die. The pair of dice has a number called second die. The first die of the pair is 6. The second die of the pair is 6. Rule for printing the name of the pair of dice while taking inventory: say "pair of dice".

Rule for printing the name of the pair of dice: say "pair of dice showing [first die of the pair plus second die of the pair]".

To say detailed state of the dice:

if the first die of the pair is the second die of the pair, say "double [first die of the pair]";

otherwise say "[first die of the pair] and [second die of the pair]".

The description of the pair of dice is "The pair of dice are [if the dice are carried]itching to be rolled[otherwise]showing [detailed state of the dice][end if]."

Rolling is an action applying to one carried thing. Understand "roll [something preferably held]" as rolling.

Check rolling when the noun is not the pair of dice: say "Not something you can roll." instead.

Carry out rolling:

now the pair of dice is in the holder of the actor;

now the first die of the pair of dice is a random number from 1 to 6;

now the second die of the pair of dice is a random number from 1 to 6.

Report rolling:

say "You roll [detailed state of the dice]."

Test me with "i / roll dice / look / x dice / get dice / x dice / roll dice".

Because we remember the states of the individual dice, not just a total, we can make use of the combination rolled.

The doubles count is a number that varies.

After rolling:

if the first die of the pair is the second die of the pair, increment the doubles count;

otherwise now the doubles count is 0;

continue the action.

Jail is a room. "This is Jail, and not the Just Visiting periphery, either."

Every turn when the doubles count is 3:

say "The blue-uniformed policemen blows his whistle and beckons you sternly...";

now the player carries the pair of dice;

now the player is in Jail;

now the doubles count is 0.

Every turn when the doubles count is 1 and the player is in Jail:
say "The warden gruffly releases you.";
now the player carries the pair of dice;
now the player is in Go.

175

Example Wonka's Revenge

WI

A lottery drum which redistributes the tickets inside whenever the player spins it.

"Wonka's Revenge"

The Caribou Lodge is a room. "Hundreds of expectant faces are turned your way from every table." A lottery drum is in the Lodge. "Before you is the lottery drum[if we have spun the drum], ready to disgorge a ticket[otherwise], waiting to be spun[end if]." In the drum are a red ticket, an orange ticket, a yellow ticket, a green ticket, a blue ticket, a purple ticket, and a ticket of pure gold. The drum is closed and openable.

Understand "spin [something]" as spinning.

Spinning is an action applying to one thing.

Check spinning: if the noun is an open container which contains something, say "[The list of things in the noun] would fly out." instead.

Carry out spinning a container:
shuffle the contents of the noun.

Report spinning:
if the noun contains something, say "You rattle [if the noun is transparent][the list of things in the noun][otherwise]the stuff[end if] in [the noun].";
otherwise say "Nothing results of your shaking [the noun]."

Inform keeps track of the order in which things have been put into a container. If we want to change that order without the player's intervention, we can move the things ourselves.

To shuffle the contents of (basket - a container):
let moves be the number of things in the basket;
repeat with counter running from 1 to moves:
move a random thing in the basket to the basket.

After opening the drum when we have spun the drum for the first time:
if something (called the pick) is in the drum:
try searching the drum;
say "[The pick] it is, then.";
silently try taking the pick;
if the pick is the ticket of pure gold, end the story finally;
otherwise end the story saying "Oh well, better luck next time."

Test me with "open drum / look in drum / close drum / spin drum / open drum".

275



Example Jokers Wild

WI

A deck of cards which can be shuffled and dealt from.

Suppose we want a deck of cards which the player can shuffle and draw from. Our first (rather tedious) task is merely to set up the deck as a table:

"Jokers Wild"

Suit is a kind of value. The suits are hearts, clubs, diamonds, and spades.

Table of Cards

suit	value
diamonds	1
diamonds	2
diamonds	3
diamonds	4
diamonds	5
diamonds	6
diamonds	7
diamonds	8
diamonds	9
diamonds	10
diamonds	11
diamonds	12
diamonds	13
spades	1
spades	2
spades	3
spades	4
spades	5
spades	6
spades	7
spades	8
spades	9
spades	10
spades	11
spades	12
spades	13
hearts	1
hearts	2
hearts	3
hearts	4
hearts	5
hearts	6
hearts	7
hearts	8
hearts	9
hearts	10
hearts	11
hearts	12
hearts	13
clubs	1
clubs	2
clubs	3
clubs	4
clubs	5
clubs	6
clubs	7

clubs	8
clubs	9
clubs	10
clubs	11
clubs	12
clubs	13

We're going to describe the higher numbers as face cards, so it helps to write a new "to say" phrase.

To say (count - a number) as a card value:
 choose row with a value of count in the Table of Value Names;
 say "[term entry]".

Table of Value Names

value	term
1	"ace"
2	"deuce"
3	"three"
4	"four"
5	"five"
6	"six"
7	"seven"
8	"eight"
9	"nine"
10	"ten"
11	"jack"
12	"queen"
13	"king"

Now we get the shuffling of the deck from "sort in random order", so:

Understand "shuffle" as shuffling. Shuffling is an action applying to nothing.

Carry out shuffling:
 sort the Table of Cards in random order;
 say "You expertly rearrange the cards.".

When play begins:
 sort the Table of Cards in random order.

This will continue to work properly even as the deck is partially depleted. Speaking of which, suppose we want the player to be able to toss the cards one-by-one into a hat. They are going to need to be removed from the deck, so:

Understand "toss" or "toss a card" or "toss card" as tossing.

Tossing is an action applying to nothing.

Check tossing:
 if the number of filled rows in the Table of Cards is 0, say "The deck is empty."
 instead.

Carry out tossing:
 repeat through the Table of Cards:
 let new value be value entry;
 let new suit be suit entry;

say "You throw the [value entry as a card value] of [suit entry] at the top hat,
and [if a random chance of 1 in 3 succeeds]hit[otherwise]miss[end if].";
blank out the whole row;
rule succeeds.

If we wanted to simulate a slightly more stimulating game, we could instead have a second table to represent the player's hand of cards and record each card drawn. That would get long for the purposes of example, however, so instead we will just admit that the player's life is an empty husk of existence:

The Empty Room is a room. "It has come to this: sitting on the bare floor of Lulu's apartment with nothing to amuse you but a deck of cards and the top hat from last year's act. You reckon [the number of filled rows in the Table of Cards in words] cardtosses are all that stand between you and the utter pointlessness of existence.

Once again you curse Lulu for running off with that joker."

The player is carrying the deck of cards. The top hat is an open container in the Empty Room. It is scenery.

Test me with "toss / again / again / again / again / again / again / again".

318

☆☆☆ Example Tilt 1

WI

A deck of cards with fully implemented individual cards, which can be separately drawn and discarded, and referred to by name.

We've simulated a deck of cards before, but only as entries in a table. This time we're going to do it more completely, with card objects that can be drawn and discarded, and referred to by name. The tedious way to do this would be to make 52 objects by hand and laboriously write out their names and understand rules.

A more sensible way is to make 52 identical card objects, assign them ranks and suits, and allow Inform to generate and parse their names automatically.

So:

"Tilt"

Section 1 - Cards

Suit is a kind of value. The suits are hearts, clubs, diamonds, and spades. Understand "heart" as hearts. Understand "club" as clubs. Understand "diamond" as diamonds. Understand "spade" as spades. [Providing the singular forms means that Inform will also understand >EXAMINE SPADE, >DISCARD CLUB, and so on.]

A card is a kind of thing. A card has a suit. A card has a number called rank. Understand the suit property as describing a card. Understand the rank property as describing a card.

52 cards are in the card repository.

Now, we're going to describe the higher numbers as face cards, so it helps to write a new "to say" phrase, just as we did in *Jokers Wild*. (A subsequent version of this example shows how to print card values with red and black symbols representing the different suits; see "Tilt 3".)

To say (count - a number) as a card value:
choose row count in the Table of Value Names;
say "[term entry]".

Rule for printing the name of a card (called target):
say "[rank of the target as a card value] of [suit of the target]"

Table of Value Names

term	value	topic
"ace"	"1"	"ace/A/one"
"deuce"	"2"	"deuce/two"
"three"	"3"	"three"
"four"	"4"	"four"
"five"	"5"	"five"
"six"	"6"	"six"
"seven"	"7"	"seven"
"eight"	"8"	"eight"
"nine"	"9"	"nine"
"ten"	"10"	"ten"
"jack"	"11"	"jack/knave/J"
"queen"	"12"	"queen/Q"
"king"	"13"	"king/K"

This is enough already to let inform understand things like "ten clubs", but we want to add a couple of refinements. For one thing, we'd like to accept "of" when it appears in phrases such as "ten of clubs" (but not generically otherwise); for another, we'd like the player to be able to use various names for ranks. To this end, we need to borrow from the *Activities* chapter and modify the player's command before attempting to understand it:

After reading a command:
if the player's command includes "of [suit]":
 while the player's command includes "of":
 cut the matched text;
 repeat through the Table of Value Names:
 while the player's command includes topic entry:
 replace the matched text with value entry.
[This allows Inform to understand "ace", "deuce", "king", etc., as numerical ranks.]

It may be a bit confusing that the Table of Value Names has both a topic column and a term column, to all appearances essentially identical. But items in the topic column can be matched against the player's input, whereas items in other kinds of text column can be printed out; the two kinds of text are not treated identically by Inform, so we need to have both. Notice that the topic column contains entries like "jack/knave," which will match either "jack" or "knave" in the player's input.

Now to set up the deck at the outset. With some intelligent looping, we avoid having to declare every combination of suit and number individually:

```
When play begins:  
  reconstitute deck.
```

```
To reconstitute deck:  
  let current suit be hearts;  
  now every card is in the card repository;  
  while a card is in the card repository:  
    repeat with current rank running from 1 to 13:  
      let item be a random card in card repository;  
      now rank of item is current rank;  
      now suit of item is current suit;  
      now item is in the deck of cards;  
    now current suit is the suit after the current suit.
```

And now we need a simple setting and some actions to manipulate the deck with:

Section 2 - The Deck and the Discard Pile

```
The Empty Room is a room. "Nothing to see here."
```

```
The deck of cards is in the Empty Room. It is a closed unopenable container.  
The description is "A standard poker deck."
```

```
The discard pile is a closed unopenable container. The description is "Cards in  
this game are discarded face-down, so the discard pile is not very interesting to  
see. All you can observe is that it currently contains [if the number of cards  
which are in the discard pile is less than ten][the number of cards which are in  
the discard pile in words][otherwise]about [the rounded number of cards which  
are in the discard pile in words][end if] card[s]."
```

```
To decide what number is the rounded number of (described set - a description  
of objects):  
  let N be the number of members of the described set;  
  let R be N divided by 5;  
  let total be R times 5;  
  decide on total.
```

The above phrase rounds a number to the nearest five, so that the player is not autistically able to count a large number of cards in the discard pile at a single glance.

This next bit is an optional borrowing from the Activities chapter: we want to prevent Inform printing things like "You can see a discard pile (closed) here.", since we don't want the player to think of the piles as containers, even though Inform thinks of them in those terms.

```
Rule for printing room description details of something: do nothing instead.
```

Finally, we want the player to use "draw" and "discard" to manipulate his hand of cards:

Section 3 - Drawing and Discarding Actions

Understand the commands "take" and "carry" and "hold" and "get" and "drop" and "throw" and "discard" as something new.

Understand "take [text]" or "get [text]" or "drop [text]" as a mistake ("Here, you only draw and discard. Nothing else matters at the moment.").

Understand "draw" or "draw card" or "draw a card" as drawing. Drawing is an action applying to nothing. The drawing action has an object called the card drawn.

Setting action variables for drawing:

now the card drawn is a random card which is in the deck of cards.

Check drawing:

if the card drawn is nothing, say "The deck is completely depleted." instead.

Check drawing:

if the number of cards carried by the player is greater than four,
say "This is a five-card game; you must discard something before drawing anything further." instead.

Carry out drawing:

move the card drawn to the player.

Report drawing:

say "You draw [a card drawn]."

Understand "discard [card]" as discarding. Discarding is an action applying to one thing.

Check discarding:

if the player does not carry the noun, say "You can only discard cards from your own hand." instead.

Carry out discarding:

now the noun is in the discard pile;

if the discard pile is not visible, move the discard pile to the location.

Report discarding:

say "You toss [the noun] nonchalantly onto the discard pile."

Seeding is an action out of world. Understand "seed" as seeding. Carry out seeding: seed the random-number generator with 5681.

Test me with "seed / draw / g / g / g / g / i / discard seven of spades / draw / discard six / draw / i / discard hearts / discard six of diamonds card / draw / draw / i / discard spades card / draw / discard king card".



A deck of cards with fully implemented individual cards; when the player has a full poker hand, the inventory listing describes the resulting hand accordingly.

In our previous implementations of playing cards, we've gotten as far as creating decks of individual cards that the player can draw and discard. But in a poker game, one doesn't just have a collection of cards: one has a hand of a specific kind.

Here we take on the job of writing an inventory listing for a poker hand that will reflect the real value of what the player has drawn. To do this, we create a rulebook to sort and assess the cards in the player's hand; its possible return values are limited to the kinds of poker hands that exist, from "high card" to "royal flush".

The first three sections, creating the deck of cards and the means to parse their names, are identical to those we've already seen in Tilt 1; new material begins at section 4.

For the purposes of demonstration, we're simulating something akin to five-card draw without wilds; stud or hold-em variations would add some other complexities.

"Tilt"

Section 1 - Cards

Suit is a kind of value. The suits are hearts, clubs, diamonds, and spades. Understand "heart" as hearts. Understand "club" as clubs. Understand "diamond" as diamonds. Understand "spade" as spades.

A card is a kind of thing. A card has a suit. A card has a number called rank. Understand the suit property as describing a card. Understand the rank property as describing a card.

52 cards are in the card repository.

To say (count - a number) as a card value:
choose row count in the Table of Value Names;
say "[term entry]".

Rule for printing the name of a card (called target):
say "[rank of the target as a card value] of [suit of the target]"

Table of Value Names

term	value	topic
"ace"	"1"	"ace/A/one"
"deuce"	"2"	"deuce/two"
"three"	"3"	"three"
"four"	"4"	"four"
"five"	"5"	"five"
"six"	"6"	"six"
"seven"	"7"	"seven"
"eight"	"8"	"eight"
"nine"	"9"	"nine"
"ten"	"10"	"ten"
"jack"	"11"	"jack/knave/J"

"queen" "12" "queen/Q"
"king" "13" "king/K"

After reading a command:

if the player's command includes "of [suit]":
while the player's command includes "of":
cut the matched text;
repeat through the Table of Value Names:
while the player's command includes topic entry:
replace the matched text with value entry.

When play begins:
reconstitute deck.

To reconstitute deck:

let current suit be hearts;
now every card is in the card repository;
while a card is in the card repository:
repeat with current rank running from 1 to 13:
let item be a random card in card repository;
now rank of item is current rank;
now suit of item is current suit;
now item is in the deck of cards;
now current suit is the suit after the current suit.

Section 2 - The Deck and the Discard Pile

The Empty Room is a room. "Nothing to see here."

The deck of cards is in the Empty Room. It is a closed unopenable container.
The description is "A standard poker deck."

The discard pile is a closed unopenable container. The description is "Cards in this game are discarded face-down, so the discard pile is not very interesting to see. All you can observe is that it currently contains [if the number of cards which are in the discard pile is less than ten][the number of cards which are in the discard pile in words][otherwise]about [the rounded number of cards which are in the discard pile in words][end if] card[s]."

To decide what number is the rounded number of (described set - a description of objects):

let N be the number of members of the described set;
let R be N divided by 5;
let total be R times 5;
decide on total.

Rule for printing room description details of something: do nothing instead.

Section 3 - Drawing and Discarding Actions

Understand the commands "take" and "carry" and "hold" and "get" and "drop" and "throw" and "discard" as something new.

Understand "take [text]" or "get [text]" or "drop [text]" as a mistake ("Here, you only draw and discard. Nothing else matters at the moment.").

Understand "draw" or "draw card" or "draw a card" as drawing. Drawing is an action applying to nothing. The drawing action has an object called the card drawn.

Setting action variables for drawing:

now the card drawn is a random card which is in the deck of cards.

Check drawing:

if the card drawn is nothing, say "The deck is completely depleted." instead.

Check drawing:

if the number of cards carried by the player is greater than four,
say "This is a five-card game; you must discard something before drawing anything further." instead.

Carry out drawing:

move the card drawn to the player.

Report drawing:

say "You draw [a card drawn]."

Understand "discard [card]" as discarding. Discarding is an action applying to one thing.

Check discarding:

if the player does not carry the noun, say "You can only discard cards from your own hand." instead.

Carry out discarding:

now the noun is in the discard pile;
if the discard pile is not visible, move the discard pile to the location.

Report discarding:

say "You toss [the noun] nonchalantly onto the discard pile."

New material begins here. We want to start by grouping cards together, but identifying poker hands only if the player holds a full five cards.

Section 4 - Assessing Hands

Before listing contents while taking inventory: group cards together.

Before grouping together cards:

if the number of cards carried by the player is 5:
say "[run paragraph on]";
follow the hand-ranking rules;
if the rule succeeded, say "[the outcome of the rulebook]";
otherwise say "some random cards";
if the outcome of the rulebook is pair outcome, say " of [rank of the first thing held by the player as a card value]s";
otherwise:
say "[number of cards carried by the player in words] assorted cards";
say " (".

Rule for grouping together cards:
say "[list hand]".

To say list hand:
let chosen card be the first thing held by the player;
while chosen card is a card:
say "[chosen card]";
now chosen card is the next thing held after chosen card;
if chosen card is a card, say ", ".

After grouping together cards:
say ")".

The ranking of poker hands traditionally depends on three features: 1) whether all the cards are of the same suit (flush); 2) whether the cards constitute a numerical run of ranks (straight); and 3) how many cards or sets of cards are of matching rank (pairs, three of a kind, and four of a kind). Here we will start by assessing our hand to determine these qualities:

The hand-ranking rules is a rulebook. The hand-ranking rules have outcomes royal flush, straight flush, four of a kind, full house, flush, straight, three of a kind, two pairs, pair, high card.

The hand-ranking rulebook has a truth state called the flushness.
The hand-ranking rulebook has a truth state called the straightness.

The hand-ranking rulebook has a number called the pair count.
The hand-ranking rulebook has a number called the triple count.
The hand-ranking rulebook has a number called the quadruple count.

For convenience in identifying hand features, and for elegance when we print the hand-listing, we start by sorting the cards in the player's hand so that the high-ranked cards are listed first. It is rare that we want to concern ourselves with this, but as we saw in the section on "Looking at containment by hand" in the chapter on Change, Inform keeps an ordered list of the items inside any given container; so it does order the objects in the player's hand, and the ordering depends on which things were added to the hand most recently. By moving something to the player's hand again (even if it was already there), we change this ordering, and wind up with a sorted hand.

A card can be sorted or unsorted. A card is usually unsorted.

Definition: a card is high if its rank is 11 or more.
Definition: a card is low if its rank is 4 or less.

A hand-ranking rule (this is the initial sort rule):
now every card is unsorted;
while the player carries an unsorted card:
let item be the lowest unsorted card held by the player;
move item to the player;
now the item is sorted;
if sort-debugging is true, say "-- after initial sort: [list hand]".

This last printing instruction is there for diagnostic purposes: later we'll add a testing command to turn debugging on and off; when it's on, the game will print out its card list at various stages in sorting, to help us trouble-shoot any problems. In normal play, however, this will be off.

Next up, a check to see whether the player has a flush:

```
A hand-ranking rule (this is the finding flushness rule):
  let called suit be the suit of a random card carried by the player;
  if every card carried by the player is called suit, now flushness is true.
```

Now we check for straights; this is slightly complicated by the fact that an ace can be either the bottom of a low straight (lower than 2) or the top of a high straight (higher than king), so we explicitly check both possibilities.

```
A hand-ranking rule (this is the finding straightness rule):
  now straightness is true;
  let N be the rank of the highest card which is carried by the player;
  repeat with current rank running from N - 4 to N:
    now the test rank is the current rank;
    unless the player carries a matching card:
      if the current rank is N - 4 and the current rank is 9 and the player carries
      an ace card, do nothing; [this covers the case where an ace could be the top
      card of the sequence]
      otherwise now straightness is false.
```

And finally, we need to identify any groups of cards of the same rank. We want to know how many groups there are and how large each group is (though in practice there can only be one group of three or four in a standard-sized poker hand). We also want to mark any grouped cards so that we can move them to the front of the player's hand when we take inventory.

```
A card can be quadrupled, tripled, paired or uncombined.
```

```
Test rank is a number that varies. Definition: a card is matching if its rank is the
test rank.
```

This definition is a convenience so that we don't have to write so many explicit loops in the following rule:

```
A hand-ranking rule (this is the counting multiples rule):
  now every card is uncombined;
  repeat with current rank running from 1 to 13:
    now test rank is current rank;
    let N be the number of matching cards held by the player;
    if N is 4:
      increment the quadruple count;
      now every matching card held by the player is quadrupled;
    if N is 3:
      increment the triple count;
      now every matching card held by the player is tripled;
    if N is 2:
      increment the pair count;
      now every matching card held by the player is paired.
```

Next we tweak our sorting to reflect the make-up of the hand. There are two reasons why this might differ from the straight highest-to-lowest sort we did earlier:

- 1) we want to list aces as high unless they are serving as the bottom of a low straight, in which case they should appear last;
- 2) we want combinations to appear at the front of the list, sorted from highest value to lowest value: larger combinations first, then smaller combinations, and combinations of equal size sorted by rank.

A hand-ranking rule (this is the move aces up unless there's a low straight rule):
unless the straightness is true and the lowest card carried by the player is an ace card and the rank of the highest card carried by the player is 5,
now every ace card which is carried by the player is carried by the player;
if sort-debugging is true, say "-- after ace movement rule: [list hand]".

A hand-ranking rule (this is the move pairs forward rule):
while the player carries a paired card:
let selection be the lowest paired card which is carried by the player;
move the selection to the player;
now the selection is uncombined;
if sort-debugging is true, say "-- after pairs movement: [list hand]".

A hand-ranking rule (this is the raise ace pairs rule):
if the player carries exactly two ace cards:
repeat with item running through ace cards which are carried by the player:
move item to the player;
if sort-debugging is true, say "-- after paired-ace movement: [list hand]".

A hand-ranking rule (this is the move multiples forward rule):
while the player carries a tripled card:
let selection be the lowest tripled card which is carried by the player;
move the selection to the player;
now the selection is uncombined;
while the player carries a quadrupled card:
let selection be the lowest quadrupled card which is carried by the player;
move the selection to the player;
now the selection is uncombined;
if sort-debugging is true, say "-- after multiples movement rule: [list hand]".

Definition: a card is ace if its rank is 1.
Definition: a card is king if its rank is 13.

Now, having determined the salient qualities of our hand, we run through rules in order from the highest kind of poker combination to the lowest. Because of the order of the source, Inform will choose whichever combination applies first.

A hand-ranking rule (this is the royal-flush rule):
if flushness is true and straightness is true and the highest card carried by the player is king and the lowest card carried by the player is ace, royal flush.

A hand-ranking rule (this is the straight-flushes rule):
if flushness is true and straightness is true, straight flush.

A hand-ranking rule (this is the four-of-a-kind rule):
if the quadruple count is 1, four of a kind.

A hand-ranking rule (this is the full-house rule):
if the pair count is 1 and the triple count is 1, full house.

A hand-ranking rule (this is the flushes rule):
if flushness is true, flush.

A hand-ranking rule (this is the straights rule):
if straightness is true, straight.

A hand-ranking rule (this is the three-of-a-kind rule):
if triple count is 1, three of a kind.

A hand-ranking rule (this is the two-pair rule):
if the pair count is 2, two pairs.

A hand-ranking rule (this is the pair rule):
if the pair count is 1, pair.

A hand-ranking rule (this is the default rule):
high card.

And finally, we need to define our debugging variable here, even though we won't give the player the ability to turn it on and off except in the special testing section.

Sort-debugging is a truth state that varies.

For many examples, a test-me script is enough to prove that the example does what it ought. This example, though, is a bit more complicated, and hard to test randomly. The remainder of the source here shows how we might write a test to verify the desired behavior of our rulebook. Those who are only interested in the rulebook itself can stop reading at this point.

Section 5 - Testing hand identification - Not for release

For the sake of testing our rules, we provide an apparatus that will load the player's hand up with sample hands of each kind, then show the result to make sure that the hand is being correctly identified.

Understand "debug sorting" as debugging hand sorting. Debugging hand sorting is an action out of world.

Carry out debugging hand sorting:
if sort-debugging is false, now sort-debugging is true;
otherwise now sort-debugging is false.

Report debugging hand sorting:
say "Sort debugging is now [if sort-debugging is true]on[otherwise]off[end if]."

Test me with "draw / g / g / g / g / force hand / g / g / g / g / g / g / g / g / g / g / g / g / g".

The somewhat rough-and-ready principle of this table is that we will overwrite the cards in the player's hand by resetting their ranks and suits; every five rows of the table represent a new poker hand for the game to attempt to sort and identify. These include one example of each of the major kinds of poker hand, plus a couple of variations involving aces which test the special sorting rules.

Table of Testing Hands

set suit	set rank	
spades	1	[royal flush]
spades	13	
spades	12	
spades	11	
spades	10	
clubs	12	[straight flush]
clubs	11	
clubs	10	
clubs	9	
clubs	8	
diamonds	8	[four of a kind]
hearts	8	
spades	8	
clubs	8	
clubs	3	
clubs	1	[full house]
spades	1	
hearts	10	
spades	10	
clubs	10	
hearts	2	[flush]
hearts	5	
hearts	7	
hearts	11	
hearts	12	
hearts	1	[straight]
spades	13	
diamonds	12	
clubs	11	
hearts	10	
hearts	2	[three of a kind]
spades	2	
clubs	2	
clubs	4	
spades	3	
diamonds	6	[two pairs]
spades	6	
clubs	7	
diamonds	7	
hearts	9	
diamonds	6	[two pairs, ace high]
spades	6	
clubs	1	
diamonds	7	
hearts	1	
hearts	12	[pair]
spades	12	
diamonds	10	
spades	7	
clubs	4	
diamonds	13	[high]
hearts	11	
spades	9	
clubs	7	
diamonds	5	
hearts	1	[tricky sorting: low straight]
diamonds	2	
spades	3	
diamonds	4	

Understand "force hand" as forcing a hand. Forcing a hand is an action out of world.

Current marker is a number that varies.

Carry out forcing a hand:

- repeat with item running through cards which are carried by the player:
 - increment current marker;
 - if current marker is greater than the number of filled rows in the Table of Testing Hands, now current marker is 1;
- choose row current marker in the Table of Testing Hands;
- now the suit of item is the set suit entry;
- now the rank of item is the set rank entry.

Report forcing a hand:

- try taking inventory.

★ Example The Trouble with Printing

Making a READ command, distinct from EXAMINE, for legible objects.

"The Trouble with Printing"

A thing has some text called printing. The printing of a thing is usually "blank".

Understand the command "read" as something new. Understand "read [something]" as reading. Reading is an action applying to one thing, requiring light. Check reading: if the printing of the noun is "blank", say "Nothing is written on [the noun]." instead. Carry out reading: say "You read: [printing of the noun] [line break]". Report reading: do nothing.

The Archive is a room.

Berkeley's report is a thing in the Archive. The description is "A report from Governor Sir William Berkeley of Virginia, in 1671, in answer to the queries sent by the Commissioners of Plantations the year previous. Of this report the better part is burned and only a tail fragment remains." The printing of Berkeley's report is "I thank God, [italic type]there are no free schools[roman type] nor [italic type]printing[roman type], and I hope we shall not have these hundred years; for [italic type]learning[roman type] has brought disobedience, and heresy, and sects into the world, and [italic type]printing[roman type] has divulged them..."

Test me with "examine report / read report".

Since we defined reading as an action requiring light, we could further distinguish reading and examining (if we wanted) by writing some different visibility rules for it.

Example Pages

A book with pages that can be read by number (as in "read page 3 in...") and which accepts relative page references as well (such as "read the last page of...", "read the next page", and so on).

Suppose we have a book that the player must consult page-by-page, and we want to be able to accept all of the following input:

- > READ BOOK (to choose a random page and read it)
- > READ PAGE 1 IN BOOK
- > READ PAGE 2
- > READ THE LAST PAGE OF THE BOOK
- > READ THE NEXT PAGE
- > READ PREVIOUS PAGE IN BOOK
- > READ THE FIRST PAGE

One approach would be to write many different understand rules and actions: one action for reading randomly, one for reading a specific page, one for reading the first page, one for reading the previous page, one for reading the next page, and one for reading the last page. But this gets tedious to construct and maintain.

More usefully, we could consider that all of the last four options are essentially the same action at heart: the player is asking to read a page in the book using a name rather than a number, and we will have to perform a minor calculation to discover what the number should be. Here's an implementation using named values to resolve this problem:

"Pages"

The Library is a room. The sinister book is carried by the player. The sinister book has a number called the last page read. The sinister book has a number called the length. The length of the sinister book is 50.

Understand the command "read" as something new.

Understand "read [something]" or "consult [something]" or "read in/from [something]" as reading. Reading is an action applying to one thing, requiring light.

Understand "read [number] in/from/of [something]" or "read page [number] in/from/of [something]" or "look up page [number] in/from/of [something]" or "consult page [number] in/from/of [something]" as reading it in. Reading it in is an action applying to one number and one thing, requiring light.

Named page is a kind of value. The named pages are first page, last page, next page, previous page.

To decide what number is the effective value of (L - last page):
decide on the length of the book.

To decide what number is the effective value of (F - first page):
decide on 1.

To decide what number is the effective value of (N - next page):
let X be the last page read of the book plus 1;
decide on X.

To decide what number is the effective value of (P - previous page):
let X be the last page read of the book minus 1;
decide on X.

Understand "read [named page] in/from/of [something]" or "read the [named page] in/from/of [something]" as reading it relatively in. Reading it relatively in is an action applying to one named page and one thing, requiring light.

Does the player mean reading something in the sinister book: it is very likely.

This is the book requirement rule:
if the player is not carrying the sinister book, say "You're not reading anything." instead.

Check reading it relatively in:
if the second noun is not the sinister book, say "There are no pages in [the second noun]." instead;
abide by the book requirement rule.

Carry out reading it relatively in:
let N be the effective value of the named page understood;
now the number understood is N;
try reading N in the book.

Check reading it in:
if the second noun is not the sinister book, say "There are no pages in [the second noun]." instead;
abide by the book requirement rule.

Check reading it in:
if the number understood is greater than the length of the sinister book, say "There are only [length of sinister book in words] pages in the book." instead;
if the number understood is less than 1, say "The page numbering begins with 1." instead.

Carry out reading it in:
read page number understood.

Check reading:
if the noun is not the sinister book, say "There are no pages in [the noun]." instead;
abide by the book requirement rule.

Carry out reading:
let N be a random number between 1 and the length of the sinister book; now the number understood is N;
say "You flip the pages randomly and arrive at page [the number understood]: [paragraph break]";
try reading the number understood in the sinister book.

Table of Book Contents

page content
 2 "dhuma jyotih salila marutam / samnipatah kva meghah / samdes arthah kva patukaranaih / pranibhih prapaniyah"
 13 "amathesteron pws eipe kai saphesteron"
 50 "Rrgshilz maplot..."

To read page (N - a number):

now the last page read of the sinister book is N;
 if there is a content corresponding to a page of N in the Table of Book

Contents:

choose row with a page of N in the Table of Book Contents;
 say "You read: '[content entry]';"

otherwise:

say "Page [N] appears to be blank."

To read page (N - 47):

say "Your eyes burn; your ears ring. Beneath your gaze, the dreadful sigils writhe, reminding you of that which lies outside the edges of the universe...";
 end the story saying "You have lost your remaining sanity".

Test me with "read from the sinister book / read the book / read the next page / read page 2 / read previous page / g / read the first page / read the last page of the book / read the next page / read 47 in book".

443

Example The Fourth Body

WI

Notebooks in which the player can record assorted notes throughout play.

Some mystery games supply the player with an in-game system for taking notes, in case he doesn't want to rely on scraps of paper next to the computer. One way of doing this is to write out all the player's notes and observations into a notebook file, whose contents can be retrieved during play (or, indeed, after it).

We'll first invent a general system for writing text into notebooks, by creating a new kind called jotter. Each individual jotter will have its own disc file, and there will be basically three things which can be done with jotters: erasing, reading and writing.

"The Fourth Body"

A jotter is a kind of thing. A jotter has an external file called the text file. A jotter can be fresh or used. A jotter is usually fresh. A jotter has a text called the heading.

The currently erased jotter is an object that varies.

To erase (pad - a jotter):

now the currently erased jotter is the pad;
 write "[heading of the currently erased jotter]" to the text file of the pad;
 now the pad is fresh.

To write in (pad - a jotter):

append "[the time of day]: [topic understood]" to the text file of the

pad;
now the pad is used.

To read (pad - a jotter):
say "You read: [paragraph break][text of the text file of the pad]".

This is all as might be expected, except perhaps for the business of the "currently erased jotter". Why copy "pad" into this - why not simply write "[heading of the pad]"? The answer is that "pad" is a temporary "let" value, and cannot be used inside other phrases, such as the "write ... to ..." phrase.

We want to erase any jotters when play begins, as otherwise text left over from any previous games will still be visible:

When play begins:
repeat with pad running through jotters:
erase the pad.

Now we need to create rules to allow the player to control reading, writing and erasing. Reading we will handle with the ordinary examining action, but we create new actions for writing and erasing. A nice little trick allows WRITE WHATEVER to default to writing WHATEVER in a notebook being carried.

Instead of examining a used jotter (called the pad):
read the pad.

Instead of examining a fresh jotter (called the pad):
say "There is nothing of note in [the pad]."

Understand "write [text] in [something preferably held]" as writing it in.
Understand "write [text]" as writing it in. Writing it in is an action applying to a topic and one thing. Rule for supplying a missing second noun while writing: if a jotter (called the pad) is carried, now the second noun is the pad; otherwise say "You will have to specify what to write that it."

Check writing it in:
if the second noun is not a jotter, say "It would be better to write in a notebook." instead.

Carry out writing it in:
write in the second noun.

Report writing it in:
say "Under the current time, you write '[the topic understood]' into [the second noun]."

Understand "erase [something preferably held]" as erasing. Erasing is an action applying to one carried thing.

Check erasing:
if the noun is not a jotter, say "It's hard to see how." instead.

Carry out erasing:
erase the noun.

Report erasing:
say "You scrub out all the entries in [the noun]."

That completes a general-purpose implementation of jotters, and we put it to use:

The player carries a jotter called your notebook. The file of Player's Observations is called "notebook". The text file of your notebook is the file of Player's Observations. The heading of your notebook is "Observations in the Pottingham Green Case".

The Damp Hillside is a room. "It is just after dawn: among the trees there is misty and pale blue light. [if Havers is in the location]The only saturated color in view is the orange-and-yellow jacket of [Detective Havers]. She is trying unsuccessfully to light a cigarette. [end if][paragraph break]The body itself is further down, closer to the bottom of the ravine. It would be foolish to speculate before seeing it, but the odds are that the corpse was rolled down after death. The ground is not steep enough for the fall itself to be deadly."

Detective Havers is a woman in the Damp Hillside. The description is "She gives you a weak smile when you look at her: you know she hasn't slept more than three hours any of the last few nights." Havers is scenery.

Havers is carrying a jotter called Barbara's notebook. The file of Barbara's Observations is called "barbara". The text file of Barbara's notebook is the file of Barbara's Observations. The heading of Barbara's notebook is "I could murder a cup of tea".

The time of day is 6:32 AM.

Instead of examining your notebook when your notebook is fresh:
say "Your notebook is blank. Back in the office, of course, there are a stack of others. But you brought a fresh notebook in a kind of weary hope. You're going to pretend, just for now, that this body might be unrelated to the graphic string of murders you're already investigating."

324



Example Bibliophilia

WI

A bookshelf with a number of books, where the player's command to examine something will be interpreted as an attempt to look up titles if the bookshelf is present, but otherwise given the usual response.

Suppose we want a bookshelf with a very large number of books on it. They aren't to be taken or carried around in the game, but they should be mentioned, and the player should be allowed to look them up by name. Furthermore, the player's attempts to examine something unrecognized should be understood as an attempt to look up a title -- but only when the player is in the presence of the books. The rest of the time such requests should be rejected in the usual way.

"Bibliophilia"

The Graduate Lounge is a room. "Shabby sofas; plastic cups remaining from the afternoon's pre-lecture espresso; a collection of Xena and Hercules figurines posed for ironic effect. It's somewhat depressing at this hour, when everyone has gone home."

The Classics Reading Room is south of the Lounge. "Not as large a collection as the one in the Library, but it contains copies of everything really essential for reference."

Understand "examine [text]" as examining as a book when the player is in the Reading Room. Understand "look up [text]" as examining as a book when the player is in the Reading Room.

Examining as a book is an action applying to one topic.

Carry out examining as a book:
say "You can't find any such text."

Instead of examining as a book a topic listed in the Table of Book Titles:
say "[description entry][paragraph break]"

Table of Book Titles

topic	title	description
"Reading Greek Death" or "reading/greek/death" or "greek death"	"Reading Greek Death"	"A dense orange paperback treatise on the development of Greek eschatology."
"TAPA/Transactions/134-2"	"TAPA 134-2"	"Transactions of the American Philological Association from 2004."
"Oxford Classical Dictionary" or "OCD/dictionary/classical/oxford"	"Oxford Classical Dictionary"	"A hefty reference with short articles on everything from Greek meter to ancient cosmetics."
"Collected Dialogues of Plato" or "Plato/dialogues/hamilton/cairns"	"Collected Dialogues of Plato"	"All the Platonic dialogues -- some, admittedly, in rather tired translations -- but still a useful single volume, ed. Edith Hamilton and Huntington Cairns."
"Adobe Illustrator CS User Guide" or "user guide" or "adobe illustrator" or "adobe/illustrator/cs/user/guide"	"Adobe Illustrator CS User Guide"	"Hello, how did this get here? A suspiciously familiar name is scribbled inside the front cover..."

Some books are scenery in the Reading Room. Understand "copies" or "book" or "shelf" or "shelves" as the books. Instead of examining the books:
choose a random row in the Table of Book Titles;
say "You scan the shelves and notice, among others, a volume entitled [italic type][title entry][roman type]."

Test me with "south / examine ocd / examine books / examine books / examine plato / n / x hercules / s / x hercules".

Now if we type >X HERCULES in the Lounge, we will get

>x hercules
You can't see any such thing.

thanks to our somewhat slovenly implementation of the Lounge scenery; but in the Reading Room,

>x hercules
You can't find any such text.

In practice we might also want to extend our coverage somewhat to handle a case where the player tried to take books from the bookshelf: currently that would not be understood.

335



Example AARP-Gnosis

WI

An Encyclopedia set which treats volumes in the same place as a single object, but can also be split up.

Suppose we have a complete Encyclopedia in our game. The player is allowed to pick up the whole set (there must not be too many volumes), but also to do things with individual volumes, and indeed to scatter these volumes all over the place. Putting a volume back in the same place as the rest of the Encyclopedia should, however, restore it to the collective. We will start out by defining general rules for collectives like this:

"AARP-Gnosis"

Fitting relates various things to one thing (called the home). The verb to fit means the fitting relation. Definition: a thing is missing if it is not part of the home of it.

A collective is a kind of thing.

Before doing something to something which is part of a collective:
let space be the holder of the home of the noun;
move the noun to the space.

Instead of examining a collective:
say "[The noun] consists of [the list of things which are part of the noun]."

Now the real work begins. One reason to make this an activity is that we might easily want to override it for specific objects; for instance, the generic collecting activity here would not deal properly with collectives of clothing where some items might be worn and others not. In that case, we would want to write another, more specific "collecting" activity to handle the complexities of fashion.

Collecting something is an activity.

Every turn:
repeat with item running through collectives:
carry out the collecting activity with the item.

To remove (item - a thing) when empty:
let space be the holder of the item;
if the number of things which are part of the item is 0:
now the item is nowhere;

if the number of things which are part of the item is 1:
let the last thing be a random thing which is part of the item;
move the last thing to the space;
now the item is nowhere.

Before collecting a thing (called the item):
remove item when empty;
let space be the holder of the item;
if space is not a thing and space is not a room:
if something (called the other space) contains at least two things which fit the item, move item to the other space;
if a room (called the other space) contains at least two things which fit the item, move item to the other space;
if someone (called the owner) carries at least two things which fit the item, move item to the owner.

Rule for collecting a thing (called the item):
let space be the holder of the item;
if space is a thing or space is a room:
repeat with component running through things held by the space:
if the component fits the item, now the component is part of the item;
remove item when empty.

And now for a cheerful scenario:

The Boise Memorial Library is a room. "A concrete box of a room, roughly eight feet by fourteen, which contains all the fallout shelter has to offer by way of entertainment. Someone with a grim sense of humor has tacked a READ! literacy poster to the door, as though there were anything else to do while you await the calming of the Geiger counters." The shelf is a supporter in the Library. "A battered utility shelf stands against the south wall."

The New Idahoan Encyclopedia Set is a collective. Volume A-Aalto fits the Encyclopedia. It is part of the Set. Volume AAM-Aardvark fits the Encyclopedia. It is part of the Set. Volume Aarhus-Aaron fits the Encyclopedia. It is part of the Set. Volume AARP-Gnosis fits the Encyclopedia. It is part of the Set. Volume Gnu-Zygote fits the Encyclopedia. It is part of the Set. The Set is on the shelf.

Let's have the Encyclopedia describe itself differently depending on whether it's all in one place:

After printing the name of the Set when something missing fits the Set:
say "(missing [a list of missing things which fit the Set])"

Before printing the name of the Set when the number of missing things which fit the set is 0:
say "complete".

Test me with "get aarhus-aaron / look / inventory / get aam-aardvark / look / get gnu-zygote / look / get aarp-gnosis / look / inventory / drop set / look / get set / get a-aalto / inventory".



Example The Fifth Body

An expansion on the notebook, allowing the player somewhat more room in which to type his recorded remark.

The implementation here is much like that of the previous example, except that we allow the player to write his notebook input as a separate command, leading to an exchange such as

```
>write in my notebook
You open your notebook and prepare to write in it.
```

```
>>Am beginning to think that HT and BGG are in this together.
You finish writing and fold your notebook away.
```

```
>read my notebook
You read:
```

```
Wednesday morning
```

```
Am beginning to think that HT and BGG are in this together.
```

The opening is much as before:

```
"The Fifth Body"
```

```
A jotter is a kind of thing. A jotter has an external file called the text file. A jotter
can be fresh or used. A jotter is usually fresh. A jotter has a text called the
heading.
```

```
The currently erased jotter is an object that varies.
```

```
To erase (pad - a jotter):
  now the currently erased jotter is the pad;
  write "[heading of the currently erased jotter][paragraph break]" to the text file
of the pad;
  now the pad is fresh.
```

```
To write in (pad - a jotter):
  append "[the time of day]: [player's command][line break]" to the text file of the
pad;
  now the pad is used.
```

```
To read (pad - a jotter):
  say "You read: [paragraph break][text of the text file of the pad]".
```

```
When play begins:
  repeat with pad running through jotters:
    erase the pad.
```

```
Instead of examining a used jotter (called the pad):
  read the pad.
```

Instead of examining a fresh jotter (called the pad):
say "There is nothing of note in [the pad]."

Target jotter is an object that varies. The target jotter is usually nothing.

Understand "write in [something preferably held]" as writing in. Writing in is an action applying to one thing.

Check writing in:
if the noun is not a jotter, say "It would be better to write in a notebook."
instead.

Carry out writing in:
now the command prompt is ">>";
now the target jotter is the noun.

Report writing in:
say "You open [the noun] and prepare to write in it."

Now what happens is that the player, having typed WRITE IN NOTEBOOK, will be faced with a ">>" prompt instead of the usual ">": a sign that the input mode has changed.

The next code is to react to reading a command. Whatever the player types at the >> prompt when the target jotter is set will now be recorded in the notebook, though with a character limit of about 60-100 characters depending on how much upper-case and punctuation he uses. (There are ways to lift the character length restriction as well, but they would take us into deeper waters.)

After reading a command when target jotter is a jotter:
now the command prompt is ">";
write in target jotter;
now target jotter is used;
say "You finish writing and fold your notebook away.";
now the target jotter is nothing;
reject the player's command.

Understand "erase [something preferably held]" as erasing. Erasing is an action applying to one carried thing.

Check erasing:
if the noun is not a jotter, say "It's hard to see how." instead.

Carry out erasing:
erase the noun.

Report erasing:
say "You scrub out all the entries in [the noun]."

The player carries a jotter called your notebook. The file of Player's Observations is called "notebook". The text file of your notebook is the file of Player's Observations. The heading of your notebook is "Sunday Morning".

The Vestry is a room. "[Havers] hangs back by the door: the forensics expert is not finished with a preliminary examination of the body. From here you can't see much, except that the expert has peeled back and laid to one side a liturgical vestment that someone at the church used to cover the corpse until the police came. What was once a cream silk with festive Easter embroidery is now stained with blood-colored handprints."

Detective Havers is a woman in the Vestry. The description is "She looks glumly back. There's still a purple-ish bruise on her cheekbone from the disaster Thursday afternoon." Havers is scenery.

Havers is carrying a jotter called Barbara's notebook. The file of Barbara's Observations is called "barbara". The text file of Barbara's notebook is the file of Barbara's Observations. The heading of Barbara's notebook is "Sun. AM".

The time of day is 9:11 AM.

94



Example Costa Rican Ornithology

WI

A fully-implemented book, answering questions from a table of data, and responding to failed consultation with a custom message such as "You flip through the Guide to Central American Birds, but find no reference to penguins."

The following relies on quite a number of features we haven't met yet: tables, rules for printing names, instructions for understanding the player's commands. It is offered simply as an example of how a fully implemented book might be handled in Inform.

"Costa Rican Ornithology"

A book is a kind of thing. Understand "book" as a book. A book has a table name called the contents.

Instead of consulting a book about a topic listed in the contents of the noun:
say "[reply entry][paragraph break]".

Report consulting a book about:
say "You flip through [the noun], but find no reference to [the topic understood]." instead.

With this "topic understood" phrase, we're telling Inform to print back the word or phrase that the player was attempting to look up. This overrides the more general default response, "You discover nothing of interest in the book."

We now have the essential elements to construct whatever books we like. Now let's have an example of a specific book:

The Guide to Central American Birds is a book carried by the player. The contents of the Guide is the Table of Listed Birds.

We will come back to the idea of tables and table names later, but for now the important thing is that we have instructed Inform to look up its answers to consulting the bird guide in this form:

Table of Listed Birds

topic	reply
"[red]" or "[red] bird/macaw"	"You flip through the Guide for a while and eventually discover a reference to the [scarlet macaw], which appears to correspond with what you see before you."
"quetzal/trogon" or "resplendent trogon"	"The entry on the quetzal is quite lyrical, describing its brilliant plumage, flashing and igniting in the sunshine, which is supposedly sufficient to lure birdwatchers from all over the world. Unfortunately, the quetzal is described as being bright emerald in color, with a pink fuzz on its head and a long soft tail 'like a feather boa'. None of these describes your visitor."

The topic column is a bit special: it matches the player's input, and is not meant to be printed out again. Topic columns will be discussed further in the chapter on Tables. (Note also that, however it may appear in the documentation, the topic column should not be spanning multiple lines in our source text.)

We may also compress long or complicated topics by creating bracketed abbreviations, and in fact it's useful to do so now, to explain the red token we just used:

Understand "red-orange" or "bird" or "red" or "orange" as the scarlet macaw.
Understand "red-orange" or "red" or "orange" or "scarlet" as "[red]".

This technique is discussed further in the chapter on Understanding.

If we wanted more books, we could define those in the same way, giving each its own separate contents table to be used for consultation. But for the sake of the example we will keep it simple, and move on to the scenario itself:

The Veranda is a room. "From here you can see a considerable expanse of dense-growing jungle plants, and eventually the open water beyond."

The scarlet macaw is an animal in the veranda. "A vibrantly-colored [scarlet macaw] perches on the rail."

A thing can be known or unknown.

Before printing the name of the scarlet macaw while consulting:
now the scarlet macaw is known.

Rule for printing the name of the unknown scarlet macaw: if the macaw is unknown, say "red-orange bird of unknown species".

Test me with "look up penguins in the guide / look up quetzal in guide / look up silver nuthatches in the guide / look / look up red bird in the book / look".

and where painterly names for pigments ("cerulean") are accepted alongside everyday ones ("blue").

There are hundreds of traditional pigments, from lampblack to burnt sienna, so we will confine ourselves to just two:

"Palette"

The Atelier is a room. "The floridly untidy loft space used by a moderately unsuccessful artist (you, that is)." The canvas, palette and paint brush are here. Understand "painting" as the canvas.

Colour is a kind of value. The colours are white, red, blue and green.

The canvas has a colour. The canvas is white. The printed name of the canvas is "largely [colour] canvas".

Painting is an action applying to one thing and one colour. Check painting: if the noun is not the canvas, say "Centuries of tradition suggest that canvas is the natural home of paint." instead. Carry out painting: now the colour of the canvas is the colour understood. Report painting: say "You splash away at the now [canvas]."

Understand "paint [something] [a colour]" as painting.

Understand "calico" as white. Understand "cerulean" or "cerulean blue" as blue.

Test me with "examine canvas / paint canvas red / examine canvas / paint canvas cerulean / examine canvas".

36



Example Brown

WI

A red sticky label which can be attached to anything in the game, or removed again.

"Brown"

The Shipping Room is a room. The red sticky label is a thing carried by the player. The description of the red sticky label is "It reads: AIRMAIL[if the label is part of something (called the parent)]. It is stuck to [the parent][end if]."

A black crate is in Shipping. The description is "A boring black crate." The brown crate is a thing in Shipping. The description is "An ordinary brown crate."

After examining something when the label is part of the noun:
say "A bright red sticky label is attached to [the noun]!"

Here is the essential point: whenever we ATTACH LABEL TO something, it becomes part of that object.

Instead of tying the red sticky label to something:
now the red sticky label is part of the second noun;
say "You stick [the label] to [the second noun]."

And of course the label cannot be stuck to itself or to more than one thing at a time.

Before tying the label to something when the label is part of something:
if the label is part of the second noun:
say "[The label] is already stuck to [the second noun]." instead;
otherwise:
say "(first freeing the label)[line break]";
silently try taking the label;
if the label is part of something, stop the action.

Instead of tying the red sticky label to the label:
say "That would ruin the label entirely."

Instead of taking the label when the label is part of something:
now the player carries the label;
say "You peel the label off again."

Much of the rest is just tidying to make sure that the player's commands are redirected into the right syntax.

Instead of tying something to the label:
try tying the label to the noun.

Instead of putting the label on something:
try tying the label to the second noun.

Instead of inserting the label into something:
try tying the label to the second noun.

Understand the commands "stick" or "apply" as "tie".

We could have created a new "sticking" action, but to keep the example short we will use the built-in "tying" action instead, and respond to the command "stick" just as if it were "tie".

Understand "peel [something]" or "peel off [something]" as taking.

Test me with "i / put label on the black crate / look / x black / x label / get the label / apply label to brown crate / look / x brown / peel off label / stick label to label".



A child's set of building blocks, which come in three different colours - red, green and blue - but which can be repainted during play.

This would be a one-star example if it were not for the repainting:

"Early Childhood 1"

A building block is a kind of thing. A red block, a blue block and a green block are kinds of building block.

The Nursery is a room. In the Nursery are six red blocks, four blue blocks and a green block.

Test me with "look / get red block".

But a kind cannot change during play, so this will not do. Instead, the colour will have to be a property of the block. So we might first try this:

"Early Childhood 2"

Colour is a kind of value. The colours are red, blue and green. A block is a kind of thing. A block has a colour. A block is usually blue.

The Nursery is a room. In the Nursery are six red blocks, four blue blocks and a green block.

Test me with "look / get red block".

Which is fine, so far as it goes, but the colour property is not at all visible to the player, who simply sees "eleven blocks". We thought of colour as being something outwardly apparent, but Inform does not know this. To achieve a better effect, we will need features from distant chapters. The first is an activity called "printing the name of":

"Early Childhood 3"

Colour is a kind of value. The colours are red, blue and green. A block is a kind of thing. A block has a colour. A block is usually blue. Before printing the name of a block: say "[colour] ". Before printing the plural name of a block: say "[colour] ".

The Nursery is a room. In the Nursery are six red blocks, four blue blocks and a green block.

Test me with "look / get red block".

This too, however, is unsatisfactory. The individual blocks are correctly described, but we are unable to distinguish them during play: we cannot type "take a green block", for instance. And because the blocks are indistinguishable in play, they are still massed together as "eleven blocks" in room descriptions. We need to go one step further:

"Early Childhood 4"

Colour is a kind of value. The colours are red, blue and green. A block is a kind of thing. A block has a colour. A block is usually blue. Before printing the name

of a block: say "[colour] ". Before printing the plural name of a block: say "[colour] ". Understand the colour property as describing a block.

The Nursery is a room. In the Nursery are six red blocks, four blue blocks and a green block.

And now everything works nicely: the blocks are grouped by colour, and can be referred to by colour, and we can even change the colour of an individual block during play, using a bit of extra trickery from later:

Understand "paint [something] [colour]" as painting it. Painting it is an action applying to one thing and one colour. Check painting it: if the noun is not a block, say "Paints are only for blocks." instead. Carry out painting it: now the colour of the noun is the colour understood. Report painting it: say "The block is now [the colour of the noun]."

Test me with "get red block / get blue block / g / i / look / paint blue block red / i / look / paint me red".

61

★ Example Control Center

WI

Objects which automatically include a description of their component parts whenever they are examined.

It is straightforward to make a rule that anything with parts must mention all those parts during an EXAMINE command:

"Control Center"

After examining a thing when something is part of the noun:
say "[The noun] includes [a list of things which are part of the noun]."

The Control Center is a room. "Here you are at the Control Center of the universe."

The Universe Management Computer is a fixed in place thing in the Control Center. "The Universe Management Computer sits directly before you, unguarded." The description of the Universe Management Computer is "The computer is so large that you would be unable to operate it all from one position. Alas, it does not come with a manual."

A chartreuse indicator light, an ennui meter, a golden knob settable to 15,000 positions, a toothpick dispenser, and a button labeled RESTART are part of the Universe Management Computer.

The command chair is an enterable supporter in the Control Center. It is pushable between rooms. "Because the computer is too large for you to reach all of the front panel from a standing position, there is a command chair on casters which allows you to push back and forth." The description of the command chair is "Quite ordinary, really, but for the heady rush of power that comes of sitting in it.". Some casters are part of the command chair.

Now whenever we look at any object with components, we will first see the description, then a list of parts which belong to it. The following refinement brings in elements of later chapters, but it may be worth noting: because we've written our rule as an "After examining...", anything that pre-empts the operation of the examine command will also prevent that rule from occurring. So for instance:

A hair-thick needle is part of the ennui meter.

Instead of examining the ennui meter: say "You can't be bothered."

...would not result in the needle being mentioned.

Test me with "x chair / x computer / x ennui meter"

427

★ Example What Makes You Tick

WI

Building a fishing pole from several component parts that the player might put together in any order.

Suppose we want to let the player build a fishing pole out of three parts: a hook, a string, and a stick.

There are several things we must account for here. One is that our combination verb should be insensitive to ordering: it shouldn't matter whether the player types COMBINE STICK WITH STRING or COMBINE STRING WITH STICK.

Second, we need to make sure that our implementation handles intervening stages of assembly gracefully. The player should be able to combine string and hook first, or string and stick first, and be able to complete the assembly in either case.

Our implementation here uses a table of lists to determine which combinations of inputs should produce which result object. Because we sort our lists before comparing them, we guarantee that the player's ordering doesn't matter: COMBINE STICK WITH STRING will have the same effect as COMBINE STRING WITH STICK.

What's more, our implementation could be expanded to account for many other assemblages, if we wanted object-building to be a running theme of puzzles in our game.

"What Makes You Tick"

Understand "combine [something] with [something]" as combining it with. Combining it with is an action applying to two carried things. Understand the command "connect" as "combine".

Understand the command "attach" as something new. Understand "attach [something] to [something]" as combining it with.

The combining it with action has an object called the item built.

Setting action variables for combining something with something:

```
let X be a list of objects;  
add the noun to X;  
add the second noun to X;  
sort X;  
repeat through the Table of Outcome Objects:  
  let Y be the component list entry;  
  sort Y;  
  if X is Y:  
    now the item built is the result entry.
```

Check combining it with:

```
if the item built is nothing or the item built is not in limbo,  
  say "You can't combine [the noun] and [the second noun] into anything  
  useful." instead.
```

Carry out combining it with:

```
move the item built to the holder of the noun;  
now the noun is nowhere;  
now the second noun is nowhere.
```

Report combining it with:

```
say "You now have [an item built]."
```

Limbo is a container. Limbo contains a hookless fishing pole, a hooked line, and a complete fishing pole.

Streamside is a room. The player carries a stick, a wire hook, and a string.

Table of Outcome Objects

component list	result
{stick, string}	hookless fishing pole
{wire hook, string}	hooked line
{hooked line, stick}	complete fishing pole
{hookless fishing pole, wire hook}	complete fishing pole

Test me with "combine stick with string / i / combine pole with hook / i".

This kind of implementation makes sense if we don't intend the player to take the fishing pole apart again, or to refer to any of its component parts once it is built. For an alternate approach that does allow assembled objects to be taken apart again, see "Some Assembly Required".

57



Example Model Shop

WI

An "on/off button" which controls whatever device it is part of.

Suppose we're particularly mechanically-minded and would like a game in which all of our mechanical devices have buttons to turn them on and off.

"Model Shop"

An on/off button is a kind of thing.

Instead of pushing an on/off button which is part of a switched off device (called the machine):

try switching on the machine.

Here we are making a rule about how our hypothetical buttons will interact with the machines to which they belong. Instead of pushing... is a rule that pertains to actions, and we will learn more about these in the chapter on actions. "...which is part of a switched off device" provides a specific circumstance - this is only to apply to buttons that are stuck to a machines that can be turned on or off. "(called the machine)" tells Inform that if it finds such a device, it should thereafter refer to it as "the machine." (The called syntax is explained further in the chapter on Change.)

A set of three more rules will complete our instructions about using buttons to control devices:

Instead of pushing an on/off button which is part of a switched on device (called the machine):

try switching off the machine.

Instead of switching on an on/off button which is part of a device (called the machine):

try switching on the machine.

Instead of switching off an on/off button which is part of a device (called the machine):

try switching off the machine.

Then we hand out buttons with a free hand:

One on/off button is part of every device.

The Model Shop is a room. A model train is a fixed in place device in the Model Shop. A toy elephant is a device in the Model Shop.

Every turn when the model train is switched on:

say "The model train circles your feet, blowing small puffs of steam."

Every turn when the toy elephant is switched on:

say "The toy elephant waves its trunk at you."

Test me with "push model train's button / push elephant's button / g / switch off model train's button".

And now the game will have a model train's button and a toy elephant's button.

It may be that we want (as an added nuance) to add other names for these items. While we would want an assembly to create objects such as "Lucy's hand" and not "Lucy hand", it is entirely reasonable to want to talk about the model train button or the elephant button. We could define these additional names like so:

Understand "elephant button" or "button on elephant" as the elephant's button.

Understand "model train" or "model" or "train" as "[train]". Understand "[train] button" or "button on [train]" as the model train's button.

In the second case, we are defining [train] to mean any of the three phrases "train", "model", and "model train"; so "[train] button" will match "model train button" or "train button" or "model button" equally well. See the chapter on Understanding for more on how to create alternative phrasings for the player to use.

53



Example Signs and Portents

WI

Signpost that points to various destinations, depending on how the player has turned it.

"Signs and Portents"

Seven Dials is a room. The description of Seven Dials is "There is a signpost, on which seven hands swivel and swing, freely as weathercocks. They make your present road now London, now Abingdon; now Weston-super-Mare, or now Hell."

Seven Dials contains a signpost. The signpost is scenery. Understand "sign" and "post" as the signpost.

Destination is a kind of value. The destinations are London, Abingdon, Luton, Weston-super-Mare, Runnymede, Hell, and Low Noon.

The signpost has a destination.

In order to interact with the signpost, we will need to make use of some action rules:

Instead of examining the signpost:

say "[The signpost] currently puts you on the road to [italic type][the destination of the signpost][roman type], but it swiftly alters again."; now the destination of the signpost is the destination after the destination of the signpost.

Instead of turning the signpost:

now the destination of the signpost is the destination after the destination of the signpost; say "With a hand's touch you turn the signpost to mark your way for [italic type][the destination of the signpost][roman type]."

Instead of going north in Seven Dials when the destination of the signpost is Hell:

say "It is a path that goes gently ever down and down with no stumbling block or any distraction at either side; there are no bandits and no tolls."; end the story.

Instead of going north in Seven Dials when the destination of the signpost is Low Noon:

say "A long road whose scenery does not change, nor anything on the horizon

move but the sun. When at last you come to Noon, she hangs above your head like a hat.";
end the story finally.

Test me with "x signpost / n / turn signpost / n".

Test more with "x signpost / n / turn signpost / n".

66

Example Radio Daze

WI

A radio that produces a cycle of output using varying text.

"Radio Daze" by Jon Ingold

The Living Room is a room. "A long couch, set up so you can see your wireless set. Not that you need to see it, of course."

The long couch is an enterable scenery supporter in the living room. Instead of entering the long couch when the radio was switched off, say "Better turn the radio on before you get comfortable." Report entering the couch: say "You settle yourself down to listen." instead. Instead of listening when the radio is switched on, stop the action.

The radio is a device in the living room. The radio is switched off. "[if switched on]The radio burbles on[otherwise]The radio is off[end if]." Check switching off the radio when the player is on the long couch: say "You can't reach the radio from here." instead.

Every turn when the radio is switched on:

say "[one of]Two characters in the radio play have begun an argument[or]The argument continues[or]The play continues[stopping]: [one of]'Did not!'[or]'Did too!'[or]'Did I?'[or]'You did!'[or]'I couldn't have, Martha!'[or]'But you did, Tom!'[cycling]"

Test me with "sit on couch / turn on radio / sit on couch/ listen / g / g / g / g / g".

311

Example Aspect

WI

Understanding aspect ratios (a unit) in the names of televisions.

Named properties are not the only kind that Inform is able to understand referring to an object. We can also use unit and number properties to distinguish things from one another, as here, where televisions have aspect ratios:

"Aspect"

An aspect ratio is a kind of value. 16:9 specifies an aspect ratio.

A television is a kind of device. A television has an aspect ratio. Understand the aspect ratio property as referring to a television. Understand "European standard" as 16:9.

The Office is a room.

The widescreen TV is a television in the Office. The fifties TV is a television in the Office. The widescreen TV is 16:9. The fifties TV is 4:3.

Test me with "examine european standard tv / x 16:9 tv / x 4:3 tv".

313

★★ Example Channel 1

WI

Understanding channels (a number) in the names of televisions.

We might want to allow every television to be tuned to a channel (a number property) which the player could refer to, so that

WATCH CHANNEL 13
TURN OFF CHANNEL 4

would be directed to the appropriate television object, if any television is turned on and tuned to the correct station. We might now write:

"Channel"

A television is a kind of device. A television has a number called the channel. Understand the channel property as referring to a television. Understand "channel" as a television.

The Office is a room. The widescreen TV is a television in the Office. The fifties TV is a television in the Office.

Changing the channel of it to is an action applying to one thing and one number.

Understand "tune [something] to [number]" or "change channel of [something] to [number]" as changing the channel of it to.

Check changing the channel of something to:
if the noun is not a television, say "[The noun] cannot be tuned to a channel." instead.

Carry out changing the channel of something to:
now the channel of the noun is the number understood.

Report changing the channel of something to:
say "You tune [the noun] to channel [number understood]."

Instead of examining a television:
if the noun is switched off, say "[The noun] is currently turned off." instead;
let the chosen channel be the channel of the noun;

if the chosen channel is a current channel listed in the Table of Television Channels:

choose row with current channel of the chosen channel in the Table of Television Channels;

say "[output entry][paragraph break]";

otherwise:

say "Snow fills the screen of [the noun]."

Table of Television Channels

current channel	output
0	"The screen of [the noun] is completely black."
4	"A gloomy female news anchor describes the latest car bomb in Baghdad: 104 dead today, and no sign of change."
5	"A couple of contestants in spangled scarlet outfits are performing an energetic paso doble."
13	"On-screen, Ichiro is up to bat with one man on second and no outs."

Test me with "change channel of fifties tv to 4 / x channel 4 / switch on fifties / x channel 4 / switch on widescreen / tune fifties tv to 5 / x channel 5 / x fifties tv / x channel 4".

316



Example Channel 2

WI

Understanding channels (a number) in the names of televisions, with more sophisticated parsing of the change channel action.

Our previous implementation of televisions ("Channel 1") doesn't allow the player to type things like

TUNE FIFTIES TELEVISION TO CHANNEL 4

nor does it deal with player input like

TUNE TO CHANNEL 4 ON FIFTIES TELEVISION

or

TUNE TO CHANNEL 4

where no television is specified. When we are designing commands which involve two elements (here, a television and a channel number), it's usually a good idea to allow the player to specify those elements in either order, as we saw demonstrated briefly in "New commands for old grammar".

We might, therefore, want to add a few refinements: first by defining a "[channel]" token that will accept input of the forms "[number]" and "channel [number]", and second by creating some additional "Understand" lines that will accept variant versions of the player's input.

"Channel 2"

Section 1 - Televisions in General

A television is a kind of device.

A television has a number called the channel. Understand the channel property as referring to a television. Understand "channel" as a television.

Changing the channel of it to is an action applying to one thing and one number.

Understand "tune [television] to [channel]" or "change channel of [television] to [channel]" as changing the channel of it to.

Understand "tune [something] to [channel]" or "change channel of [something] to [channel]" as changing the channel of it to.

Understand "tune to [channel] on [television]" or "change to [channel] on [television]" as changing the channel of it to (with nouns reversed).

Understand "tune to [channel] on [something]" or "change to [channel] on [something]" as changing the channel of it to (with nouns reversed).

Understand "[number]" or "channel [number]" as "[channel]".

Check changing the channel of something to:
if the noun is not a television, say "[The noun] cannot be tuned to a channel."
instead.

Carry out changing the channel of something to:
now the channel of the noun is the number understood.

Report changing the channel of something to:
say "You tune [the noun] to channel [number understood]."

Instead of examining a television:
if the noun is switched off, say "[The noun] is currently turned off." instead;
let the chosen channel be the channel of the noun;
if the chosen channel is a current channel listed in the Table of Television Channels:
choose row with current channel of the chosen channel in the Table of Television Channels;
say "[output entry][paragraph break]";
otherwise:
say "Snow fills the screen of [the noun]."

Table of Television Channels

current channel	output
0	"The screen of [the noun] is completely black."

Section 2 - The Scenario

The Office is a room.

The widescreen TV is a television in the Office. The fifties TV is a television in the Office.

And we add the scenario-specific content to our Table of Television Channels; in the case of channel 13, we provide for a changing sequence of events using text variations.

Table of Television Channels (continued)

current channel	output
4	"A gloomy female news anchor describes the latest car bomb in Baghdad: 104 dead today, and no sign of change."
5	"A couple of contestants in spangled scarlet outfits are performing an energetic paso doble."
13	"[one of]On-screen, Ichiro is up to bat with one man on second and no outs.[or]Ichiro has singled to first and the other man is on third.[or]The next batter is in the middle of flying out.[or]Everything looks rosy until the men in black pull off a double-play and retire the side.[or]The channel has cut to a commercial.[stopping]"

Test me with "test one / test two".

Test one with "change channel of fifties tv to 4 / x channel 4 / switch on fifties / x channel 4 / switch on widescreen / tune fifties tv to channel 5 / x channel 5 / x fifties tv / x channel 4".

Test two with "tune to channel 13 / widescreen / tune channel 13 to channel 5 / tune channel 5 to channel 3 / widescreen / x channel 3".

336



Example Aftershock

WI

Modifying the rules for examining a device so that all devices have some specific behavior when switched on, which is described at various times.

The built-in behavior of Inform is to print a line after a device is examined, saying whether the item is on or off. This is often inappropriate, and we could simply turn off that behavior in general by instructing Inform to ignore the "examine described devices rule" (see the chapter on rulebooks).

Perhaps, though, we would like continue to have a short passage about the action of any switched on device; we'd just like a little more control over what it says from time to time. And in that case, we might change the rule to give a new activity control over that portion of the description:

"Aftershock"

Section 1 - Showing actions

Showing action of something is an activity.

Rule for showing action of something (called item):
if the item is switched on, say "[The item] is switched on."
otherwise say "[The item] is switched off."

Borrowing from the rulebooks chapter, we can replace the standard "examine described devices" rule with something that uses this activity.

The new described devices rule is listed instead of the examine devices rule in the carry out examining rules.

This is the new described devices rule:
if the noun is a device:
 carry out the showing action activity with the noun;
 now examine text printed is true.

Thus far we have essentially replicated the original behavior, but we've made it possible to write specialized behavior for devices, and to invoke that behavior in other places:

Report switching on something:
 say "You flip a switch. ";
 carry out the showing action activity with the noun instead.

This might be useful for an electric lamp kind:

Section 2 - Electric Lamps

An electric lamp is a kind of device.

Rule for showing action of an electric lamp (called item):
 if the item is switched on, say "[The item] is lit[if the number of visible lit things is greater than 1], competing with [the list of visible lit things which are not the item][end if].";
 otherwise say "[The item] is dark."

Carry out switching on an electric lamp: now the noun is lit. Carry out switching off an electric lamp: now the noun is unlit.

Section 2 - The Scenario

The time of day is 3:47 AM. When play begins, now the right hand status line is "[time of day]".

The Downstairs Hallway is a dark room. "The only room in the house with no furniture and almost nothing on the walls. At times like this you always notice the crack in the plaster, originating near the light fixture and running almost all the way to the wall."

A plastic jug of filtered water is in the Downstairs Hallway. The description is "Five gallons, not that that will last you very long, hot as it has been lately."

The crack is scenery in the Hallway. The description is "No, the ceiling isn't going to fall on you today."

The light fixture is an electric lamp in the Hallway. It is switched on, lit, and scenery. The description is "A plain globe of frosted glass containing the light

bulb. Nothing special, and you never think about it except when, as now, you are forced to spend hours in this room."

The flashlight is an electric lamp carried by the player. The description is "A shiny red flashlight." The portable radio is a device carried by the player. The description is "A small battery-operated radio which you received for free with your subscription to US News & World Report. It has served you well through many earthquakes past."

And with our activity, we can override the flashlight's electric lamp behavior with new behavior:

Rule for showing action of the flashlight:
if the flashlight is switched on, say "A strong, narrow beam of light shines from the flashlight.";
otherwise say "It is currently switched off."

...or give special actions for the radio:

Rule for showing action of the radio:
if the radio is switched on, say "Through the static, you pick up pieces of discussion: a 6.7 on the Richter scale, epicenter... something about Topanga... but it crackles out again.";
otherwise say "The radio is silent. You're saving the batteries."

Instead of listening in the presence of the switched on radio:
carry out the showing action activity with the radio instead.

Test me with "examine light / switch light off / switch flashlight on / switch radio on / examine radio / examine flashlight".

226



Example Four Cheeses

WI

A system of telephones on which the player can call distant persons and have conversations.

"Four Cheeses"

Section 1 - Telephones and Connections

A telephone is a kind of thing. Understand "phone" as a telephone.

Understand "call [any telephone] on [something]" as calling it on. Understand "call [any telephone]" as calling it on. Understand the commands "dial" or "phone" or "telephone" as "call". Understand "call [any known person]" as calling it by name on.

Connection relates one thing to another (called the other party).

The verb to reach means the connection relation.

Calling it on is an action applying to one visible thing and one thing.

Check calling it on:

if the second noun is not a telephone, say "[The second noun] is unlikely to be much use in that respect." instead;

if the second noun is the noun, say "You get a busy signal." instead.

Carry out calling it on:

if a person (called the listener) can see the noun, now the player reaches the listener.

Because we've said that connection is a reciprocal, one-to-one relationship, Inform will do the rest of the bookkeeping: if (for instance) we telephone someone else, the first connection will be broken automatically.

Report calling it on:

say "'Hello?' says [the other party of the player]."

To avoid annoyance, we should also let the player use CALL ##### as well as CALL ##### ON TELEPHONE. A rule from the chapter on Activities comes in handy here:

Rule for supplying a missing second noun while calling something on:
assign a phone.

To assign a phone:

if the player can touch a telephone (called the current phone):

say "(on [the current phone])[line break]";

now the second noun is the current phone;

otherwise:

say "You don't have a phone handy."

Things might be a little more complicated if we had cell phones that could be moved around, but for right now the player can only touch a maximum of one phone at a time.

Suppose we further want to allow the player to call people up by name, but only if they've already been encountered or are familiar to the player for some reason.

A person can be known or unknown.

Understand "call [any known person]" as calling it by name on.

Understand "call [any known person] on [something]" as calling it by name on.

Rule for supplying a missing second noun while calling something by name on:
assign a phone.

Calling it by name on is an action applying to one visible thing and one thing.

Check calling it by name on:

if the noun is in the location, say "[The noun] is right here." instead.

Carry out calling it by name on:

if the noun can touch a telephone (called the link), try calling the link on the second noun;

otherwise say "You can't reach [the noun]." instead.

Before calling something on something when the player reaches someone:

say "(first ending your conversation with [the other party of the player])
[command clarification break]";
end current conversation.

Understand "hang up [something]" as hanging up.

Hanging up is an action applying to one thing.

Check hanging up:

if the noun is not a telephone, say "You can't hang up [the noun]." instead;
if the player does not reach someone, say "You're not on the line with anyone." instead.

Carry out hanging up:

now the player does not reach anyone.

Report hanging up:

say "You put down [the noun], cutting the connection."

Before going somewhere when the player reaches someone:

say "(first hanging up on [the other party of the player])[command clarification break]";
end current conversation.

And finally we want to make sure that calling random other numbers produces a sensible result:

Understand "call [text]" as misdialling. Misdialling is an action applying to one topic. Carry out misdialling: say "The phone rings and rings but no one answers."

Understand "call 911" or "call 999" or "call police" or "call fire department" as a mistake ("After strict warnings, you've given up making prank calls to emergency services.").

Before misdialling when the player reaches someone:

say "(first ending your conversation with [the other party of the player])
[command clarification break]";
end current conversation.

To end current conversation:

let the current phone be a random telephone which can be touched by the player;
silently try hanging up the current phone.

After deciding the scope of the player while the player reaches someone:

place the other party of the player in scope, but not its contents.

A note about this scope addition: the player can refer to the other party whenever he has the other person on the phone. He can't, however, see or refer to anything that person might be holding or wearing, thanks to the "but not its contents" option.

Furthermore, the player can't actually do anything to that person that requires touching. That's because of the reaching inside rules, which govern whether the player can reach through intervening barriers such as rooms. (See the Advanced Actions chapter for more about changing reachability.) There are two things we might want to be careful about, though.

First, we should specifically disallow the player from looking at the person on the other end of the line. Since sight doesn't require touching, the reaching inside rules will not be consulted about a command such as EXAMINE BOSS or LOOK UNDER BOSS. We can, however, intervene in such cases using the visibility rules, which are consulted for any actions that "require light" (including EXAMINE and LOOK UNDER). Here again we borrow some options from the Advanced Actions chapter:

```
To decide whether acting through the line:
  if the noun is something and the location of the noun is not the location of the
  player:
    yes;
  if the second noun is something and the location of the second noun is not the
  location of the player:
    yes;
  no.
```

```
Visibility rule when acting through the line:
  there is insufficient light.
```

```
Rule for printing a refusal to act in the dark when acting through the line:
  say "You're not on a video phone, so you can only hear." instead.
```

Second, though the existing reaching inside rules are adequate to stop us from touching the person on the other end of the line, the response that's currently printed is a bit generic: it just says "You can't reach into [the room containing the person]." Let's add our own custom reply, instead:

```
A rule for reaching inside a room (called destination):
  if the other party of the player is enclosed by the destination:
    say "Though you're on the line with [the other party of the player], you can't
    physically reach to [the destination].";
  deny access.
```

Section 2 - Conversation over the Phone, In General

This portion supplies a simple method of conversation; but we could substitute some completely different conversation system if appropriate. The effect of the telephones is that we are allowed to talk to characters in distant locations under certain circumstances, after which the usual conversation rules apply.

```
Instead of listening to a telephone when the player reaches someone:
  say "You can hear [the other party of the player] breathing."
```

Before listening to someone when the player cannot touch the noun:
say "[The noun] is waiting for you to carry on the conversation." instead.

A person has a table name called chatter.

Before telling someone about something:
try asking the noun about it instead.

Before answering someone that something:
say "Best to confine your conversation to questions and answers." instead.

Before asking someone about something:
if the topic understood is a topic listed in the chatter of the noun, say "[reply entry][paragraph break]" instead;
otherwise say "[The noun] does not reply." instead.

Section 3 - The Scenario

The Guard House is a room. "Here you spend all your nights. Bullet-proof windows offer a panoramic view of serene cliffs, palm trees, and a moonlit ocean. Occasionally someone is foolish enough to try a cliff ascent or even an attack by helicopter, but lately things have been pretty quiet.

The mansion is up the hill behind you, security lights ablaze."

The grey telephone is a telephone in the Guard House. Understand "6885" as the grey telephone. "Before you is a grey telephone. In black marker someone has written on it: MAIN OFFICE 2802."

Before going a direction in the Guard House, say "And leave your post? The boss would have you flayed. No kidding." instead.

In a game where the player could walk around, we would of course want to add a before rule so that he automatically hung up any phone he was using before leaving the room.

The Main Office is a room. The boss is a known woman in the Main Office. A telephone called the red telephone is in the Main Office. Understand "2802" as the red telephone.

The Guild is a room. The ninja is an unknown man in the Guild. A telephone called the black telephone is in the Guild. Understand "4431" as the black telephone.

Potter's Pizza is a room. The pizza delivery boy is a known man in Pizza. A telephone called the saucy telephone is in Pizza. Understand "8885" as the saucy telephone.

The chatter of the boss is the Table of Boss Conversation. The chatter of the delivery boy is the Table of Pizza Conversation. The chatter of the ninja is the Table of Ninja Conversation.

After calling the red telephone on something for the first time:
say "'Yes?' asks the boss. Her voice is especially husky this evening. Maybe

that night of passion isn't so far off after all."

Table of Boss Conversation

topic	reply
"love/passion/tonight/night" or "night of passion"	"...Sorry, what?' she asks. 'I wasn't listening.' Oh. Maybe she'd go for some pizza, though."
"pizza"	"'I'd love some. No pepperoni, though,' she says, sounding dreamy. Yes, this is definitely time for a call to your old friend, the pizza boy."
"imminent ninja attack"	"'Don't worry about it,' says the boss crisply. 'I have everything under control.'"

Table of Pizza Conversation

topic	reply
"pizza"	"'Pepperoni special tonight!' he says proudly."
"pepperoni"	"'Pepperoni is included free on ALL our pizzas,' he says proudly."
"no pepperoni"	"'Well, I don't see why you'd want that,' replies the boy sniffily. 'It's free!'"
"jalapeno"	"'Sorry, we're out of jalapenos this evening. There was a run on them.'"
"sausage"	"'Sausage, sure, we can do you sausage.'"
"canadian bacon"	"'There's currently an embargo on Canadian pig products.'"
"cheese"	"'We use four kinds,' says the boy, then lowers his voice confidentially. 'Actually, two of them are the same. Nobody ever counts. The stringy one, the one that comes in dollops and the orangey one. You know.'"
"pineapple"	"'We could put pineapple on there, sure,' says the delivery boy, in a tone that lets you know his opinion of people who order fruit-based pizzas."
"pineapple and garlic"	"'What kind of crazy combination is that?' demands the delivery boy, finally losing all self-control."
"delivery"	"'Well, I don't know,' says the boy in a worried voice. 'Last time I came there were attack dogs. And ninjas.'"
"massive gratuity"	"'There's no use in a big tip you don't live to spend,' says the delivery boy quite firmly."

Table of Ninja Conversation

topic	reply
"imminent attack"	"'Yes, still on for tonight,' confirms the voice at the other end of the line."
"pizza delivery boy"	"The voice, in tones of velvet, indicates that it cannot guarantee the safety of any delivery persons whomsoever."

Test me with "call 2802 / examine boss / ask boss about night of passion / ask boss about pizza / listen to telephone / call delivery boy on telephone / ask boy about cheese / tell boy about no pepperoni / ask boy about delivery / tell boy about massive gratuity / attack boy".

...and it more or less writes itself from there.

253



Example Alias

WI

A telephone with phone numbers of the standard American seven-digit length.

Seven-digit telephone numbers are too long for Inform to handle when compiling to the Z-Machine, but they will work under Glux. To have this example succeed, make sure that you have selected the Glux option in your settings menu.

"Alias"

A telephone is a kind of thing. Understand "phone" or "telephone" as a telephone.

A phone number is a kind of value. 999-9999 specifies a phone number.

Now we borrow some techniques from the Understanding chapter to set up dialing actions:

Understand "dial [phone number] on [telephone]" as dialing it on. Understand "dial [phone number] on [something]" as dialing it on.

Understand the commands "phone" or "telephone" or "call" as "dial".

Understand "call [text]" or "phone [text]" or "dial [text]" or "telephone [text]" as a mistake ("That's not a number you know.").

Dialing it on is an action applying to one phone number and one thing.

Report dialing it on:
say "You dial [the phone number understood]."

This much is enough to let us dial telephone numbers and have Inform report that we've done so; it doesn't actually provide a telephone system such that we could reach and converse with other characters (but see the other telephone examples in the recipe book for more on how one might do that).

We'll set up a little political espionage scenario from which our player can make calls:

The Senator's Junior Suite is a room. "The Senator appears, unfortunately, to have very precise habits: little in the room has been moved from its usual place; the trash can is empty; the bed has been remade[if the blue paper is unexamined]. There may in fact be nothing to find here[end if]."

The bed is an enterable scenery supporter in the Junior Suite.

The player is wearing a housekeeping uniform and a brunette wig. The player carries a telephone called a Nokia.

Borrowing again from the chapter on Understanding, we might arrange things so that the player knows and can call a few standard numbers with such syntax as CALL HOME:

Understand "home" as 555-9200.

And what if we'd like to have the player learn some phone numbers during the game?

A thing can be examined or unexamined. Carry out examining something: now the noun is examined.

Understand "Stephen" as 555-2513 when the blue paper is examined.

This will understand CALL STEPHEN once the paper is examined; before that, the player will just get the "That's not a number you know" response that Inform uses for all attempts to call unknown names.

We'd better plant this paper for the player to find:

The blue paper is in the drawer. The description of the blue paper is "It reads: 'Call Stephen - 555-2513'."

The drawer is part of the dresser. It is closed and openable. The dresser is in The Senator's Junior Suite. The lamp is on the dresser. The description of the dresser is "The single drawer is [if the drawer is open]open[otherwise]shut[end if]."

Test me with "dial 555-9999 / call home on the telephone / phone the president / call stephen / open drawer / read paper / call stephen / put phone in drawer / close drawer / call stephen".

180

★ Example Witnessed 2

WI

A piece of ghost-hunting equipment that responds depending on whether or not the meter is on and a ghost is visible or touchable from the current location.

"Witnessed"

The player carries a device called a Trifield Natural EMF Meter. The description of the Meter is "This cost a pretty penny off the internet, but it's worth it: according to the website it has been programmed by PhD physicists to ignore manmade sources of fields and to respond only to paranormal EMF changes.

It also features an optional Tone Alarm, which can be turned on to indicate when readings spike. If the alarm is off, the meter just reads out the magnetic and electric field levels on a scale from 0-100 microteslas, or 0-1000 V/m.

Since both fields are important, you keep the meter set to SUM mode. The meter has its own optional backlighting, so that you can see the reading even if your flashlight is off. Currently it is reading at [meter setting]." A Tone Alarm is part of the Meter. It is a device. The description of the Tone Alarm is "The Tone Alarm will make a noise, if the EMF picks up a spike."

To decide what number is meter setting:
if the meter is switched off, decide on 0;
if a ghost is touchable, decide on 35;
if a ghost is visible, decide on 12;
decide on 0.

After switching on the meter:
say "You turn on the meter. The needle steadies at [meter setting]."

Every turn: if the meter setting is greater than 10 and the Tone Alarm is switched on, say "[The Tone Alarm] shrieks."

Thirtieth Street Station is a room. "A huge, high, rectangular room with coffered ceilings, which looks grand but mostly makes you feel lonely and small. There are long benches in rows down the middle of the room, and an information desk with the train times, and a series of ticket windows, none of which matters very much at the moment."

The benches are an enterable supporter. They are scenery in the Station. The information desk is scenery in the Station. Some ticket windows are scenery in the Station. Instead of examining scenery in the Station: say "You're fairly sure that whatever is going on here has nothing to do with [the noun]." Understand "window" as ticket windows.

The mural is fixed in place in Thirtieth Street. "At the north side of the station is a particularly pointless and empty annex to the main room. It is dominated by a huge relief of sorts, and this is what you remember." Understand "metal" or "relief" or "huge" as the mural. The description of the mural is "It is both stylized and confusing, but you think it might be supposed to represent the various tasks and occupations of Philadelphia's population. The portions closer to the ground look as though they have recently been subjected to a light dusting of talcum powder. No unusual prints are evident."

The wind chimes are fixed in place in Thirtieth Street. "Carefully attached to the wall with a piece of duct tape and a hook is a light-weight set of wind chimes. Someone else has been here before you, it seems." The description is "Several of your friends use wind chimes as a sort of ghost alarm, since ghosts sometimes cause very localized movements of air when there is no natural breeze."

A ghost is a kind of person. The pale figure is a ghost.

At 9:03 AM: move the pale figure to the location; say "You shiver with some sort of presence."

Test me with "turn on alarm / turn on meter / z / z / z / x figure".

299

★ Example Tom's Midnight Garden

WI

A clock kind that can be set to any time using "the time understood"; may be turned on and off; and will advance itself only when running. Time on the face is also reported differently depending on whether the clock is analog or digital.

Time can also be understood as a token, and the time parsed will be recorded as "the time understood". So therefore, if we wish for clocks which may be set:

"Tom's Midnight Garden"

A clock is a kind of device. A clock has a time called the current time. A clock can be analog or digital. The current time of a clock is usually 9:01 AM. The

description of a clock is "It shows the time to be [if analog]about [the current time to the nearest five minutes in words][otherwise][the current time][end if]."

Understand "set [clock] to [time]" as setting it by time. Setting it by time is an action applying to one thing and one time.

Instead of setting a clock to something:

say "[The noun] can be set only to a time of day, such as 8:00 AM, or midnight."

Carry out setting a clock by time:

now the current time of the noun is the time understood.

Report setting a clock by time:

say "You set [the noun] to [time understood]."

Every turn:

repeat with item running through switched on clocks:

now the current time of the item is one minute after the current time of the item.

The Hall is a room. The grandfather clock is a fixed in place analog clock in the Hall. The travel clock is a switched on digital clock in the Hall. When play begins: now the right hand status line is "[time of day]".

Test me with "examine grandfather clock / set it to midnight / switch it on / wait / wait / wait / examine it / set travel clock to 4:12 / examine it".

365



Example Ginger Beer

WI

A portable magic telescope which allows the player to view items in another room of his choice.

Suppose we want to have a pair of linked lenses so that the player can look into one of them and see things which occur in room containing the other lense.

We begin simply with a bit of environment for the player to wander around:

"Ginger Beer"

The Ginger Beer Factory is a room. "In the center of the room is an enormous pot filled with crushed ginger, which seems to be bubbling slightly on its own. The fumes are overwhelming."

The pot is scenery in the Ginger Beer Factory. The description of the pot is "Cast iron." In the pot is a bubbling brew.

Instead of smelling the Ginger Beer Factory: try smelling the brew.

Instead of smelling the brew, say "You blink back tears."

The Storeroom is south of the Ginger Beer Factory. "The walls here are lined with a prodigious number of small, rounded bottles, each with a screw top and a smiling pirate on the label."

The Clippings Room is west of the Ginger Beer Factory. "A clean room lined with steel tables, for preparing ingredients."

Some steel tables are a supporter in the Clippings Room. They are scenery. The description is "They are roughly the size and height of laboratory worksurfaces."

The quantity of dandelion is on the steel tables. The description is "Horrible common weed."

The wooden box is on the steel tables. It is openable and closed. The description is "A large wooden box with a lid, used for ingredient storage. There is a label on the lid."

The label is part of the box. The description is "BURDOCK: the root beaten with a little salt and laid on the place suddenly easeth the pain thereof, and helpeth those that are bit by a mad dog:... the seed being drunk in wine 40 days together doth wonderfully help the sciatica: the leaves bruised with the white of an egg and applied to any place burnt with fire, taketh out the fire, gives sudden ease and heals it up afterwards.... The root may be preserved with sugar for consumption, stone and the lax."

The quantity of burdock is in the box. The description is "It looks like a kind of thistle."

Some bottles are in the Storeroom. They are scenery. The description is "They are smaller than the average bottle, because more potent." Instead of taking the bottles, say "Take one away and the whole lineup will cascade to the floor."

Now for the lenses themselves:

A lense is a kind of thing.

The large end of the telescope is a lense in the Ginger Beer Factory. "There is a large glass lense propped against the wall, in which are reflected all the contents of the room." Understand "glass" or "lense" as the large end.

The small end of the telescope is a lense in the Storeroom. "There is a small glass lense sitting on the floor. Due to some curious effect of the optics, it appears to be giving a view of somewhere else entirely." Understand "glass" or "lense" as the small end. The description is "A gleaming lense about the size of a pound coin."

Here is the critical bit, which needs to be somewhat flexible, since the large end of the telescope could in theory be left anywhere in the game (and should still work).

After deciding the scope of the player while the small end is carried by the player:

- let there be the holder of the large end;
- place there in scope.

Before searching the small end when the small end is not carried by the player:
say "(first picking up [the small end] and holding it to your eye)";
silently try taking the small end.

Instead of searching the small end when the player is not carrying the small end:
say "It's too hard to look through the small end from a distance."

Instead of searching the large end,
say "You see only your own reflection."

We also want to make sure that the player who looks through the small lense does not see the large lense listed among the contents of the other location:

Definition: a thing is recognizable if it is not a lense.

Instead of searching the small end:
let the far side be the holder of the large end of the telescope;
say "You peer into the little lense and through it see, in [the far side], [the list of recognizable things in the far side]."

Test me with "examine lense / south / examine lense / look through lense / north / look through small lense".

And we're done.

273

★ **Example If It Hadn't Been For...**

WI

A sound recording device that records the noises made by player and non-player actions, then plays them back on demand.

We start out by giving ourselves a capacious recording device:

"If It Hadn't Been For..."

The digital recorder is a device. The description is "A noise-activated recorder, which time-stamps each recording segment. It has space for about 60 short recordings."

Every turn:
if the digital recorder is switched on and the number of blank rows in the Table of Recorded Content is 0, now the recorder is switched off.

Table of Recorded Content

time stamp	sound
a time	some text
with 60 blank rows.	

And most of what follows is attaching sounds to various events. (We could have made noises associated with all the actions, but for simplicity we stuck to a few.)

The thing to note here is that the recording happens as part of Carry out, not as part of Report, so sounds will be recorded even when they are the result of non-player action when the player is not even in the room.

Carry out opening something in the presence of the switched on recorder:
record "A creaking noise, as of something being opened."

Carry out someone opening something when the switched on recorder can see the noun:
record "A creaking noise, as of something being opened."

Carry out closing something in the presence of the switched on recorder:
record "A creaking followed by a slam."

Carry out someone closing something in the presence of the switched on recorder:
record "A creaking followed by a slam."

Carry out someone going to a room (called destination) in the presence of the switched on recorder:
if the destination is the holder of the recorder, record "Footsteps, growing louder.";
otherwise record "Footsteps, fading out."

Carry out going to a room (called destination) in the presence of the switched on recorder:
if the destination is the holder of the recorder, record "Footsteps, growing louder.";
otherwise record "Footsteps, fading out."

Carry out someone eating something in the presence of the switched on recorder:
record "Loud uncouth chewing sounds."

Carry out eating something in the presence of the switched on recorder:
record "Distant muffled chewing sounds."

To record (noise - some text):
if the number of blank rows in the Table of Recorded Content is 0, rule succeeds;
choose a blank row in the Table of Recorded Content;
now time stamp entry is the time of day;
now sound entry is noise.

Understand "play [something]" as listening.

Instead of listening to the recorder:
if the number of filled rows in the Table of Recorded Content is 0, say "The recorder remains blank." instead;
repeat through the Table of Recorded Content:
say "[line break][time stamp entry]: [sound entry]";
say paragraph break.

The Haunted House is a room. The squeaky cupboard is an openable enterable closed fixed in place container in the House. The ghost is a man in the cupboard. The Lawn is outside from the Haunted House.

Instead of opening the closed cupboard when the ghost is in the cupboard: say "The cupboard stubbornly refuses to open."

Every turn when the player is not in the House:
if the ghost is in the cupboard:
try the ghost exiting;
otherwise if the cupboard is open:
try the ghost closing the cupboard.

Before someone exiting when the person asked is in a closed container (called the trap):
try the person asked opening the trap.

Before someone entering a closed container: try the person asked opening the noun.

Before going to the House when the House contains the ghost:
try the ghost entering the cupboard;
try the ghost closing the cupboard.

The player carries the recorder, chips, and a sandwich. The sandwich is edible. The chips are edible.

Carry out someone eating the chips in the presence of the switched on recorder: record "An incredible racket of a packet being opened." Carry out eating the chips in the presence of the switched on recorder: record "An incredible racket of a packet being opened."

Test me with "open cupboard / drop recorder / switch it on / eat chips / out / wait / wait / wait / in / switch recorder off / play recorder".

Now we're at liberty to record evidence of the ghost getting out of the cupboard and getting back in, while we ourselves stand about on the lawn.

369

★ Example Originals

WI

Allowing the player to create models of anything in the game world; parsing the name "model [thing]" or even just "[thing]" to refer to these newly-created models; asking "which do you mean, the model [thing] or the actual [thing]" when there is ambiguity.

We rely here on the understanding-by-relations rules we've already learned, but there is an additional trick: we want to make sure that if the player types "original" or "actual", this word will not be taken to refer to the thing modeled:

"Originals"

A model is a kind of thing. 100 models are in the model-repository.

Appearance relates one thing to various models. The verb to be shown by means the appearance relation.

Indication relates a model (called X) to a thing (called Y) when Y is shown by X and Y is suitable.

Understand "actual" or "original" as "[actual]". Understand "[actual]" as something when the item described is not a model.

Definition: a thing is suitable:
if the player's command includes "[actual]", no;
yes.

Understand "[something related by indication]" as a model.

After printing the name of a model (called target): say "[random thing shown by the target]"

Now our duplication command -- for the sake of simplicity, we'll suppose that in this scenario the player is duplicating objects by magic rather than creating them out of physical materials or supplies:

Understand "duplicate [something]" as duplicating. Duplicating is an action applying to one visible thing.

The duplicating action has an object called the selected model.

Setting action variables for duplicating:
let N be a random model in the model-repository;
now the selected model is N.

Check duplicating:
if the selected model is nothing, say "You're out of power." instead.

Carry out duplicating:
now the noun is shown by the selected model;
move the selected model to the player.

Report duplicating:
say "You concentrate and manifest [a selected model]."

Now, the challenge is that we want to print the word "actual" before printing the name of an object, but only during disambiguation questions and only when we are not printing the name of the object as part of a model-name! (If we are not careful about the latter point, we will get responses such as "Which do you mean, the model actual deer or the actual deer?" which of course defeats the whole purpose.

The way around this is to remember that activities stack: we're printing the name of the deer while printing the name of a model that involves the deer. So if we set a flag while printing the name of a model, we can control the way the deer's name prints during the transaction. (We could use our ...while clause to specify while not printing

the name of a model, except that we're already using it for "while asking which do you mean", and these do not stack.) So:

The virtual-context is a truth state that varies. The virtual-context is false.

Before printing the name of a model:
now virtual-context is true.

After printing the name of a model:
now virtual-context is false.

Before printing the name of something (called target) while asking which do you mean:
if the target is not a model and virtual-context is false:
say "actual".

Forest is a room. It contains a deer and a daisy. The deer is an animal.

Test me with "duplicate deer / x model deer / x deer model / drop deer / x deer / actual / x deer / model".

417

✦ Example **Mirror, Mirror**

WI

The sorcerer's mirror can, when held up high, form an impression of its surroundings which it then preserves.

"Mirror, Mirror"

The Sorcerer's Workshop is a room. "The sorcerer's den is a dusty, whispering place. A grandfather clock with skeletal hands reads [the time of day in words]. The floorboards are stained where that porridge just wouldn't come out."

The Apprentice's Pantry is east of the Workshop. "This is where the aproned apprentice traditionally makes the camomile tea, cleans out the jackdaw cages and furtively examines purloined artefacts."

When play begins: erase the mirror.

The player carries a magic mirror. The magic mirror has a text called the mirror vision.

To erase the mirror: now mirror vision of the mirror is "The mirror is polished clean, and has no impression upon it."

To say current room description: try looking.

To expose the mirror:

say "The mirror shines momentarily with a dazzling light.[paragraph break]";
now mirror vision of the mirror is the substituted form of "The hazy image in the mirror preserves a past sight:[line break][current room description]All is distorted and yet living, as though the past and present are coterminous in the mirror."

Understand "hold up [something preferably held]" or "hold [something preferably held] up" as holding aloft. Holding aloft is an action applying to one carried thing. Report holding aloft: say "You hold [the noun] aloft."

Instead of rubbing the mirror: erase the mirror; try examining the mirror. Instead of holding aloft the mirror: expose the mirror.

The description of the mirror is "[mirror vision of the mirror]".

Test me with "look / examine mirror / hold up mirror / z / look / x mirror / rub mirror / east / hold mirror up / west / x mirror".

221

☆☆ Example Actor's Studio

WI

A video camera that records actions performed in its presence, and plays them back with time-stamps.

Here we construct a video camera to track and play back actions:

"The Actor's Studio"

Section 1 - The Video Camera

The video camera is a thing carried by the player.

Table of Videotape

recorded action	time stamp
waiting	9:00 AM
with 25 blank rows.	

Mode is a kind of value. The modes are idle, recording, and playing back. The video camera has a mode. The video camera is idle.

Understand "play back" as playing back. Instead of switching on the camera, try tuning the camera to recording. Instead of switching off the camera, try tuning the camera to idle.

The description of the video camera is "It is currently [mode]; its available settings are idle, recording, and playing back."

Understand "set [camera] to [a mode]" as tuning it to. Tuning it to is an action applying to one thing and one mode.

Instead of setting the camera to something:

say "The available settings are idle, recording, and playing back."

Check tuning it to:

if the noun is not the camera, say "Only the video camera can be set to [the mode understood]." instead.

Carry out tuning it to:
now the mode of the noun is the mode understood.

Report tuning it to:
say "You set [the noun] to [mode understood]."

After an actor doing something when the video camera is recording:
if the current action is tuning the video camera to recording, make no decision;
if the number of blank rows in the Table of Videotape is greater than zero:
choose a blank row in the Table of Videotape;
now the recorded action entry is the current action;
now the time stamp entry is the time of day;
otherwise:
now the video camera is idle;
say "The video camera runs out of recording memory and switches off.";
continue the action.

Every turn when the video camera is playing back:
say "On the camera screen, you see [run paragraph on]";
let starting playback be false;
repeat through the Table of Videotape:
if the recorded action entry is not waiting:
now starting playback is true;
say "[line break] -- [if the actor part of the recorded action entry is the
player]you [end if][the recorded action entry], time stamped at [time stamp
entry]";
blank out the whole row;
if starting playback is false, say "only static.";
otherwise say paragraph break.

Section 2 - The Scenario

The Actor's Studio is a room. Lucas is a man in the Actor's Studio. Persuasion
rule: persuasion succeeds.

The Studio contains an edible thing called a croissant.

Test me with "set camera to recording / x lucas / lucas, take inventory / lucas, eat
croissant / set camera to playing back / z".

Notice that both Lucas' implied taking action (picking up the croissant) and his eating
action are recorded on the same move.

322

☆☆ Example Claims Adjustment

WI

An instant camera that spits out photographs of anything the player
chooses to take a picture of.

We start by creating a camera and a photograph object. As usual when we want to
have a kind of object that can be dispensed in bulk, we start off with a bunch of
identical instances of the object out of play (in this case, kept in an out-of-play

container called "film roll"); we can then move them into play and give them characteristics when they're needed.

Each photograph can depict exactly one thing -- we're assuming that the player is not a landscape photographer here -- so we create a relation to indicate what is shown by each photograph. We'll then use that relation to determine how photographs are described, named, and parsed:

"Claims Adjustment"

A photograph is a kind of thing. 36 photographs are in the film roll.

Appearance relates one thing to various photographs. The verb to be shown by means the appearance relation.

The description of a photograph is usually "It shows [a random thing which is shown by the item described]."

Understand "of [something related by reversed appearance]" as a photograph.

This allows the player to refer to any photograph by its subject: useful if we have a large number of them.

Now we create an action to let the player use the camera and generate these photograph objects:

The player carries a cheap instant camera.

Understand "photograph [something] with [camera]" as photographing.
Understand "photograph [something] with [something preferably held]" as photographing. Photographing is an action applying to one visible thing and one carried thing, requiring light.

The photographing action has an object called the selected film.

Setting action variables for photographing:
let N be a random photograph in the film roll;
now the selected film is N.

Check photographing:
if the second noun is not the camera, say "You need a camera for that purpose." instead.

Check photographing:
if the noun is the camera, say "Sadly impossible." instead.

Check photographing:
if the selected film is nothing, say "You're out of film." instead.

Carry out photographing:
now the noun is shown by the selected film;
move the selected film to the player.

Report photographing:

say "Your camera instantly spits out [a selected film]."

Now we use two activities from the Activities chapter to describe the photographs to the player more elegantly:

After printing the name of a photograph (called target):

say " of [a random thing which is shown by the target]".

After printing the plural name of a photograph (called target):

let N be the holder of the target;

say " of [a list of things which are shown by photographs which are held by N]";

if the number of things which are shown by photographs which are held by N is greater than one, say " (variously)".

And finally we provide a brief scenario to give the player something to take pictures of:

The Treasure Room is a room. "Despite the fancy name, this is no more than a closet -- albeit a closet with its own special circuit on the house alarm."

The Treasure Room contains a small Degas, a Ming vase, and a collection of South African krugerrands. The player is carrying insurance forms, a first-class stamp, and a security envelope.

The description of the forms is "Completely filled out in black ink in block letters: now all you need to do is attach photographic evidence of the objects you wish to insure."

Test me with "photograph degas / i / photograph degas / i / x photograph of degas / photograph me / x photograph of me / i / photograph vase / photograph camera / photograph collection / g / i / test more".

Test more with "x photograph of collection / x photograph of krugerrands / x photograph of collection of south african krugerrands / photograph photograph of degas / x photograph of photograph of degas".

Chapter 10: Physics: Substances, Ropes, Energy and Weight

§10.1. Gases; §10.2. Liquids; §10.3. Dispensers and Supplies of Small Objects; §10.4. Glass and Other Damage-Prone Substances; §10.5. Volume, Height, Weight; §10.6. Ropes; §10.7. Electricity and Magnetism; §10.8. Fire; §10.9. Heat; §10.10. Magic (Breaking the Laws of Physics); §10.11. Mathematics

-  Contents of *The Inform Recipe Book*
-  Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics
-  Chapter 11: Out Of World Actions and Effects
-  Indexes of the examples

§10.1. Gases

Inform normally assumes that everything is solid. It has no built-in support for gases or liquids, because they have so many different behaviours. For instance, is the important thing about gas that it diffuses, or that we breathe it, or that it mixes with other gases to react, or that it sometimes obscures vision? The answer depends on what we are trying to write, and for this reason Inform leaves it up to us.

Gases are easier to deal with than liquids, because they tend to be everywhere in their location at once (unlike a liquid which might form a pool on the floor) and because they diffuse and mix by themselves (rather than being carried around or brought into contact with each other by the player). On the other hand, unlike liquids, gases are compressible: they can be present at low or high pressures, that is, in low or high concentrations.

The simplest approach is the one in **Only You...**, where rooms are either filled with smoke or else smoke-free. Smoke gradually fills through the map, obscuring vision: no attempt is made to conserve the total quantity of smoke, as we assume that some fire is churning it out continuously to replace what diffuses away.

Lethal Concentration 1 and **2** simulate a gas diffusing through a three-dimensional maze of rooms, and becoming dangerous above certain concentrations. There is just one possible gas, and it is modelled by giving each room a number which represents the concentration (in parts per million). This enables us to conserve the total amount of gas, or to have it released or captured by sources and sinks of given capacity.

This could be extended by giving each room similar concentration levels for other gases, and providing for the diffusion rule to notice when different gases come into contact; or by giving a concentration (and also, for realism, a volume) to each closed container, applying rules for capturing and releasing gases as containers are opened and closed.

- ↑ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ← Back to Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics: §9.12. Cameras and Recording Devices
 - Onward to §10.2. Liquids
 - ↓ Example 255:  **Lethal Concentration 1** A poisonous gas that spreads from room to room, incapacitating or killing the player when it reaches sufficient levels.
 - ↓ Example 74:  **Only You...** Smoke which spreads through the rooms of the map, but only every other turn.
 - ↓ Example 257:  **Lethal Concentration 2** Poisonous gas again, only this time it sinks.
-

§10.2. Liquids

Liquids are notoriously difficult to simulate well. A fully thorough approach consumes endless storage and can be very finicky to write and keep realistic. It is essential to decide what aspect of a liquid's behaviour is actually needed in a given story, and to simulate only that. For instance, if we only need a little chemistry, where a player can add (say) water to salt and make a solution, we do not want to fool around with calculating quantities or concentrations: what's important is that "some water" (amount unspecified) combines with "some salt" to produce "some salty water". We should no more calculate precisely here than we would work out where all the furniture is to the nearest inch. Good advice for handling liquids is to simulate the least amount of realism possible, but no less.

Sometimes all we want is a down-in-one drink: we needn't simulate the actual liquid, just the bottle it comes in, and all we need is to handle the "drinking" action. See **Beverage Service**, and also **3 AM**, where carbonated drinks can be shaken - again simulating the vessel, not the liquid.

Some elementary biochemistry in **Xylan** is done simply by... well, the point is that two different liquids are represented by single things each, and a chemical reaction simply switches one for the other.

In **Frizz**, we allow any container to be filled with water (only) and we simulate what happens to any solid objects also inside: some waterproof, some not. **Flotation** provides a well (always full of water), with rules to determine whether things dropped into it should sink or float.

Next we move up to quantitative approaches, where we remember not just whether a liquid is present, but how much of it. In its simplest form, we could have a drinking vessel from which we draw in sips, so that it can be full, half-empty or empty: see **Thirst**.

The example with the best compromise between simulation quality and complexity is **Lemonade**. Here we provide a kind of container called a "fluid container", not just a single cup, and each such vessel has a given "fluid capacity". Each holds only a single liquid at a time (so no mixtures) and can be empty or full to any level (rounded off to the nearest 0.1 fl oz). We can fill one vessel from another (unless it would make a mixture). But liquids leaving these vessels must be consumed - drunk or poured away without trace: we cannot make pools on the floor, or carry liquids in our cupped hands. There is no object representing "lemonade": there are only fluid containers, but which can be called LEMONADE if that is what they now contain.

Savannah is a light elaboration of **Lemonade**, showing how liquids might be poured on other objects, as for instance to extinguish a fire.

Noisy Cricket extends **Lemonade** to allow for mixing, though then the number of different possible mixtures is so large that complexity increases greatly. **Lakeside Living** extends **Lemonade** differently to add a "liquid source" kind, a form of fluid container which has infinite fluid capacity and is scenery - ideal for a lake, river or spring.

★ See **Bags, Bottles, Boxes and Safes** for stoppered bottles which could also be used for carrying liquids around in

★ See **Heat** for keeping liquids warm in insulated containers

- ⬆ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ⬅ Back to §10.1. Gases
 - ➡ Onward to §10.3. Dispensers and Supplies of Small Objects
 - ⬇ Example 117: ★ **Thirst** A waterskin that is depleted as the player drinks from it.
 - ⬇ Example 122: ★ **Beverage Service** A potion that the player can drink.
 - ⬇ Example 403: ★ **Flotation** Objects that can sink or float in a well, depending on their own properties and the state of the surrounding environment.
 - ⬇ Example 288: ★★ **Xylan** Creating a new command that does require an object to be named; and some comments about the choice of vocabulary, in general.
 - ⬇ Example 192: ★★★ **Frizz** Liquid flows within containers and soaks objects that are not waterproof; any contact with a wet object can dampen our gloves.
 - ⬇ Example 193: ★★★ **3 AM** A shake command which agitates soda and makes items thump around in boxes.
 - ⬇ Example 266: ★★★ **Lemonade** Containers for liquid which keep track of how much liquid they are holding and of what kind, and allow quantities to be moved from one container to another.
 - ⬇ Example 267: ★★★ **Savannah** Using the liquid implementation demonstrated in **Lemonade** for putting out fires.
 - ⬇ Example 276: ★★★ **Noisy Cricket** Implementing liquids that can be mixed, and the components automatically recognized as matching one recipe or another.
 - ⬇ Example 333: ★★★ **Lakeside Living** Similar to "Lemonade", but with bodies of liquid that can never be depleted, and some adjustments to the "fill" command so that it will automatically attempt to fill from a large liquid source if possible.
-

§10.3. Dispensers and Supplies of Small Objects

A slightly tricky situation arises in IF when we want to offer the player a simulation of a near-infinite supply of something: a napkin dispenser from which he can keep taking more napkins, or an infinite selection of pebbles on a beach, or something of that nature.

One approach is simply to limit the number of items the player is allowed to pick up at a time, while maintaining the fiction that there are more of these items in existence than the player is allowed to interact with. **Extra Supplies** demonstrates this.

The task becomes harder if we do want to let the player have as many napkins as he wants. In some languages, it is possible to generate new objects on the fly after the story has begun (called "dynamic object creation"), and something like this is possible if we are compiling for Glulx. (See the Inform extensions site for examples.) Usually, though, it is less complicated and almost as effective simply to have a very large supply of existing objects, which are moved in and out of play as the player needs them. **Pizza Prince** demonstrates how to do this with slices of pizza.

★ See **Ropes** for an example involving divisible pieces of string, which relies on similar techniques

- ↑ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ← Back to §10.2. Liquids
 - Onward to §10.4. Glass and Other Damage-Prone Substances
 - ↓ Example 378: ★ **Pizza Prince** Providing a pizza buffet from which the player can take as many pieces as he wants.
 - ↓ Example 124: ★★ **Extra Supplies** A supply of red pens from which the player can take another pen only if he doesn't already have one somewhere in the game world.
-

§10.4. Glass and Other Damage-Prone Substances

Just as Inform normally assumes everything is solid, it also assumes that these solid objects will not buckle, crack, break or deform under pressure, and cannot be fragmented. But breakability adds realism, and breakage need not be negative: sometimes we want the player to break her way in to something.

In IF the word DROP is more often used to mean "put down" or "leave behind" than "let go from a height": so it is perhaps unfair that in **Ming Vase** something fragile, when dropped, shatters (into nothing). In **Spring Cleaning**, fragile objects must be explicitly attacked by the player in order to break, and although they leave no tangible debris behind, their loss is at least remembered. **Kyoto** provides a general-purpose model for things being thrown at other things, with consequences including things moving (even between rooms) as well as breaking each other.

Debris from breakages is to be avoided if possible because it means keeping track of increasing numbers of objects. But we can increase realism by allowing something to have a visibly "broken" state, which it changes to when it breaks, rather than simply vanishing. **Terracottissima** provides for broken and unbroken flowerpots in this way.

Since "part of" allows us to have two objects joined together into what the player sees as one, it also gives us a natural seam which allows the whole to be broken back down into its component parts, and this is the neatest way of providing a breakage into pieces. **Paddington** demonstrates a cutting action which allows component parts to be cut away from their holders but will only make small surface gashes in any individual thing: so the player can cut something up, but only into the pieces we specifically choose to allow. Cutting also forces an opening into containers.

★ See **Combat and Death** for a robot that breaks into its component limbs when shot with a blaster

★ See **Goal-Seeking Characters** for a character who eats donuts, leaving crumbs on the floor

★ See **Volume, Height, Weight** for containers breaking under the weight of their contents

★ See **Ropes** for cutting up string into up to 35 different pieces of different lengths - a limit the player is unlikely to find out about, but a limit all the same, and an expensive solution since we need 35 different things for the "debris" when string is "broken"

- ⬆ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ⬅ Back to §10.3. Dispensers and Supplies of Small Objects
 - ➡ Onward to §10.5. Volume, Height, Weight
 - ⬇ Example 97: **Ming Vase** ATTACK or DROP break and remove fragile items from play.
 - ⬇ Example 123: **Spring Cleaning** A character who sulks over objects that the player has broken (and which are now off-stage).
 - ⬇ Example 195: **Paddington** A CUT [something] WITH [something] command which acts differently on different types of objects.
 - ⬇ Example 314: **Terracottissima** The flowerpots once again, but this time arranged so that after the first breakage all undamaged pots are said to be "unbroken", to distinguish them from the others.
 - ⬇ Example 404: **Kyoto** Expanding the effects of the THROW something AT something command so that objects do make contact with one another.
-

§10.5. Volume, Height, Weight

What should fit into what? Inform has basically three sizes: small, person-sized, and room-sized. The difference between "small" and "person-sized" doesn't appear much, but it's the difference between an ordinary container and an enterable container; the fact that a person cannot get inside an ordinary container is one of the few size-related rules built into Inform. It will not object to, say, a fishing rod being put inside a matchbox.

Inform does have one built-in measure of the size of a container: its "carrying capacity". This is a maximum number of contents:

The carrying capacity of the rucksack is 3.

This of course allows three anvils, while forbidding four postage stamps. To do better, we need units of measurement, and **Dimensions** demonstrates setting these up. **The Speed of Thought**, meanwhile, ventures into the area of unit conversion: having multiple types of unit and being able to express them to the player, or parse these in the player's input.

To be fully realistic in what will fit into what, we need sophisticated three-dimensional models of shapes, both of the items being carried and of the free space remaining inside containers. **Depth** elegantly simplifies this by approximating items as cuboids, with a given

width, length and height: these multiply to give a volume. To fit in a container, a new item's volume must not exceed the volume remaining inside the container, and in addition its three dimensions must also fit in one of the possible arrangements at right angles to the sides. (So this system would indeed prevent a 1x1x100 fishing rod from being put inside a 5x2x1 matchbox, but would also prevent a 12x1x1 pencil from being put into a 10x10x1 box, because it would need to be turned diagonally to fit.)

Lead Cuts Paper provides a different constraint: here we do not let light-weight containers hold heavy objects.

Weight comes in a different way into **Swerve left? Swerve right? Or think about it and die?**, which exploits up/down map connections to work out which way gravity would take a rolling marble.

★ See **Liquids** for containers with liquid capacity

- ↑ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ← Back to §10.4. Glass and Other Damage-Prone Substances
 - Onward to §10.6. Ropes
 - ↓ Example 232: ★ **Swerve left? Swerve right? Or think about it and die?** Building a marble chute track in which a dropped marble will automatically roll downhill.
 - ↓ Example 268: ★ **Depth** Receptacles that calculate internal volume and the amount of room available, and cannot be overfilled.
 - ↓ Example 258: ★★ **Dimensions** This example draws together the previous snippets into a working implementation of the weighbridge.
 - ↓ Example 270: ★★★ **The Speed of Thought** Describing scientifically-measured objects in units more familiar to the casual audience.
 - ↓ Example 259: ★★★★ **Lead Cuts Paper** To give every container a breaking strain, that is, a maximum weight of contents which it can bear - so that to put the lead pig into a paper bag invites disaster.
-

§10.6. Ropes

Ropes, chains and similar long, thin, bendable items present three problems: they are like a liquid in that (unless unbreakable) they can be divided arbitrarily into smaller and smaller portions of themselves, they can be in two or more places at once (even in two or more rooms at once), and they can be tied down at either or both ends, allowing them to occupy an uneasy state in between being "portable" and "fixed in place". Even when all this is simulated, they allow us to pull on one end and so to exert force at the other - allowing action-at-a-distance which Inform's realism rules would ordinarily forbid. Ropes are hard. And it is very difficult to imagine everything a player might try when given a fully practical rope with which to solve puzzles.

Snip solves the divisibility question, allowing string to be cut or retied into lengths of any size, with all the consequences for describing and understanding that this entails.

Otranto provides a lengthy but straightforward approach to the other rope-related issues, subject to the simplifying assumptions that a rope is indivisible, has about the length of the

distance between two adjacent rooms, and cannot be tied to another rope.

-  Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 -  Back to §10.5. Volume, Height, Weight
 -  Onward to §10.7. Electricity and Magnetism
 -  Example 228:  **Otranto** A kind of rope which can be tied to objects and used to anchor the player or drag items from room to room.
 -  Example 261:  **Snip** A string which can be cut into arbitrary lengths, and then tied back together.
-

§10.7. Electricity and Magnetism

Electrons are so tiny, and move so fast, that we will never want to simulate them in ordinary IF. So we simply regard electricity and magnetism as behaviours which are either present or not present, and which have instantaneous effects.

In **Witnessed 1**, batteries provide electricity to enable a "device" to work. Even if switched on, a device with no battery will be ineffective.

Larger voltages are exposed in **Electrified**, which makes certain items untouchable, and ensures that an experienced electrician will not even try.

Rules of Attraction provides for a magnet which attracts metallic items just strongly enough to stick together until pulled apart for any reason.

-  Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 -  Back to §10.6. Ropes
 -  Onward to §10.8. Fire
 -  Example 346:  **Rules of Attraction** A magnet which picks up nearby metal objects, and describes itself appropriately in room descriptions and inventory listings, but otherwise goes by its ordinary name.
 -  Example 408:  **Electrified** Adding a rule before the basic accessibility rule that will prevent the player from touching electrified objects under the wrong circumstances.
 -  Example 139:  **Witnessed 1** A kind of battery which can be put into different devices, and which will lose power after extended use.
-

§10.8. Fire

Fire exhibits some of the properties of a gas: it is only vaguely located and tends to spread out, though it passes by touch rather than on the air. It is hazardous to life, through direct contact, heat, and smoke. Better governed, it provides light and warmth. Worse governed, it consumes almost anything it comes into contact with. Here the problem with "debris" is not so much that we need potentially hundreds of new objects to represent broken items: instead, fire could sweep through a work of IF destroying so much that no play is possible any

longer. Setting up a problem in which the player must defeat a fully-capable fire is difficult to balance.

As with liquids, it is best to simulate the least amount of fire that the design will allow.

Bruneseau's Journey provides a single candle which can be lit, or blown out, but where fire can never transfer from the candle's end to anything else - or vice versa: the player's source of fire, with which to light the candle, is discreetly neglected.

In the more realistic **Thirst 2**, a campfire is lit using a tinderbox, so that fire does transfer from one thing (tinder) to another (the campfire): but it is always confined to just these two items, and can be used only for light and warmth.

The Cow Exonerated provides a class of matches that can light any flammable object, but assumes that burning objects requires only one turn; lighting one thing does not burn another.

In Fire or in Flood provides a complete simulation of what we might call "wild-fire": combustion which spreads through arbitrary objects and rooms, destroying all in its path.

★ See **Examining** for a way to describe objects as charred once they have been partly burnt

★ See **Heat** for one consequence of fire having touched something

★ See **Gases** for an implementation of smoke without fire, if this can exist

★ See **Liquids** for water being used to extinguish a simple fire

★ See **Lighting** for other uses of candles and torches as light sources

-
-  Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 -  Back to §10.7. Electricity and Magnetism
 -  Onward to §10.9. Heat
 -  Example 118: ★ **Thirst 2** A campfire added to the camp site, which can be lit using tinder.
 -  Example 150: ★★ **Bruneseau's Journey** A candle which reacts to lighting and blowing actions differently depending on whether it has already been lit once.
 -  Example 418: ★★★ **The Cow Exonerated** Creating a class of matches that burn for a time and then go out, with elegant reporting when several matches go out at once.
 -  Example 401: ★★★★ **In Fire or in Flood** A BURN command; flammable objects which light other items in their vicinity and can burn for different periods of time; the possibility of having parts or contents of a flaming item which survive being burnt.
-

§10.9. Heat

Since we prefer not to simulate burnt skin, and it is unsporting to kill a player outright merely for touching a hot object, heat is mostly used as a reason why something cannot be picked up at a given moment. This very basic puzzle is demonstrated in **Grilling**.

With the naked eye, it is not always easy to detect what is too hot to touch - a point made in both **Masochism Deli**, where the only solution is to keep picking up potatoes until one doesn't burn, and in **Hot Glass Looks Like Cold Glass**, where infrared goggles turn the scales.

If a hot object is not to be touched, will it stay hot forever? It might well, if it is a steak on an electric grill, but not if it is a recently-baked apple pie sitting on a window-sill. **Entropy** simulates the gradual return of temperature to equilibrium.

★ See **Electricity and Magnetism** for items which shouldn't be touched because they are hot in a different way

- ⬆ Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 - ⬅ Back to §10.8. Fire
 - ➡ Onward to §10.10. Magic (Breaking the Laws of Physics)
 - ⬇ Example 84: ★ **Grilling** A grill, from which the player is not allowed to take anything lest he burn himself.
 - ⬇ Example 182: ★ **Entropy** All objects in the game have a heat, but if not kept insulated they will tend toward room temperature (and at a somewhat exaggerated rate).
 - ⬇ Example 326: ★ **Masochism Deli** Multiple potatoes, with rules to make the player drop the hot potato first and pick it up last.
 - ⬇ Example 331: ★★★ **Hot Glass Looks Like Cold Glass** Responding to references to a property that the player isn't yet allowed to mention (or when not to use "understand as a mistake").
-

§10.10. Magic (Breaking the Laws of Physics)

Every previous section of this chapter has been about adding further realism constraints to Inform, to give it a more advanced understanding of physics (and chemistry). But Inform has plenty of physical laws built into it already, even if they are more mundane: inanimate objects do not spontaneously move or change, one solid object cannot pass through another, there are opaque barriers through which light cannot pass, and so on. These rules stop many attempted actions. (GO EAST - "The oak door is closed." GET EMERALD - "You can't, since the glass display case is in the way.")

In the real world, physics is not negotiable. ("Gravity: it's not just a good idea, it's the law.") But in stories, magic can sometimes happen, and in these examples some of the rules built into Inform are waived in special circumstances, which in effect cancel certain physical laws. Very many other magical tricks could be achieved: if you want to make a given command work despite realism constraints, try typing **ACTIONS** - a testing command which reveals what is happening behind the scenes - and then typing the command you want. This should reveal which named rule is stopping it: you can then try suspending that rule, and seeing if the effect now works.

★ See **Magic Words** for ways to create new single-word spell commands

-  Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 -  Back to §10.9. Heat
 -  Onward to §10.11. Mathematics
 -  Example 215:  **Magneto's Revenge** Kitty Pryde of the X-Men is able to reach through solid objects, so we might implement her with special powers that the player does not have...
 -  Example 225:  **Interrogation** A wand which, when waved, reveals the concealed items carried by people the player can see.
 -  Example 227:  **Transmutations** A machine that turns objects into other, similar objects.
 -  Example 398:  **Access All Areas** The Pointy Hat of Liminal Transgression allows its wearer to walk clean through closed doors.
-

§10.11. Mathematics

It is uncommon, but not absolutely unheard-of, to need to do detailed mathematical calculations in interactive fiction.

The Fibonacci Sequence demonstrates the calculation of a series of numbers; **Sieve of Eratosthenes** shows off how to calculate the prime numbers within a certain range.

Number Study demonstrates conditional relations between numbers.

-  Start of Chapter 10: Physics: Substances, Ropes, Energy and Weight
 -  Back to §10.10. Magic (Breaking the Laws of Physics)
 -  Onward to Chapter 11: Out Of World Actions and Effects: §11.1. Start-Up Features
 -  Example 240:  **Number Study** The parity and joint magnitude relations explored.
 -  Example 434:  **The Fibonacci Sequence** The modest Leonardo Fibonacci of Pisa will be only too happy to construct his sequence on request, using an array.
 -  Example 437:  **Sieve of Eratosthenes** The haughty Eratosthenes of Cyrene will nevertheless consent to sieve prime numbers on request.
-

Examples from Chapter 10: Physics: Substances, Ropes, Energy and Weight

-  Start of this chapter
-  Chapter 11: Out Of World Actions and Effects
-  Indexes of the examples

255

Example **Lethal Concentration 1**

A poisonous gas that spreads from room to room, incapacitating or killing the player when it reaches sufficient levels.

WI

"Lethal Concentration"

A concentration is a kind of value. 200.9ppm specifies concentration. 200.9 ppm specifies concentration.

A room has a concentration called current concentration. A room has a concentration called former concentration.

Probability inverse is a number that varies. [This is expressed as an inverse of the actual probability of diffusion from one room to another, to avoid error.]
Probability inverse is 20. [That is, any given molecule of gas has a 5% chance of leaving by a given doorway at any given minute. Probability inverse should never drop below 10, the maximum number of exits from the room.]

Every turn:
follow the diffusion rules.

The diffusion rules are a rulebook.

A diffusion rule (this is the gas movement rule):
repeat with space running through rooms:
let sum be 0.0 ppm;
repeat with way running through directions:
let second space be the room way from the space;
if second space is a room:
let difference be the former concentration of the second space minus
the former concentration of the space;
increase sum by the difference;
let sum be sum divided by probability inverse;
now current concentration of the space is the former concentration of the
space plus the sum.

A technical note: it would be possible to write "repeat with space running through rooms... repeat with second space running through rooms adjacent to the space" instead, but in practice this loops through all the rooms * all the rooms again * all the directions (to determine adjacency). Phrasing the loop this way omits the second multiplier. For a map of 25 rooms, this means that the loop runs 25 times faster than it would otherwise, and of course for a larger map the effect would be even more dramatic.

A diffusion rule (this is the resetting concentration rule):
repeat with space running through rooms:
now the former concentration of the space is the current concentration of
the space.

The last diffusion rule (this is the lethal dosage rule):
if the current concentration of the location is greater than LC50:
say "The concentration in the air overpowers you...";
end the story;
otherwise:
if the current concentration of the location is greater than TLV-STEL:
say "You feel extremely uncomfortable in this environment."

Instead of doing something when the current concentration of the location is greater than TLV-STEL:
if going, continue the action;
say "You can't work in this environment: your eyes and nose sting and it hurts to breathe."

And, for testing purposes, a square grid of rooms:

Room 1A is west of Room 1B. Room 1B is west of Room 1C. Room 1C is west of Room 1D. Room 1D is west of Room 1E.

Room 2A is south of room 1A and west of Room 2B. Room 2B is west of Room 2C and south of Room 1B. Room 2C is west of Room 2D and south of Room 1C. Room 2D is west of Room 2E and south of Room 1D. Room 2E is south of Room 1E.

Room 3A is south of room 2A and west of Room 3B. Room 3B is west of Room 3C and south of Room 2B. Room 3C is west of Room 3D and south of Room 2C. Room 3D is west of Room 3E and south of Room 2D. Room 3E is south of Room 2E.

Room 4A is south of room 3A and west of Room 4B. Room 4B is west of Room 4C and south of Room 3B. Room 4C is west of Room 4D and south of Room 3C. Room 4D is west of Room 4E and south of Room 3D. Room 4E is south of Room 3E.

Room 5A is south of room 4A and west of Room 5B. Room 5B is west of Room 5C and south of Room 4B. Room 5C is west of Room 5D and south of Room 4C. Room 5D is west of Room 5E and south of Room 4D. Room 5E is south of Room 4E.

The former concentration of room 3C is 800.0 ppm.

For variety of testing, here is another room set-up, this time with some corridors and walls within; uncommenting it, and commenting out the connected grid, will let us explore what would happen in alternative cases, to prove to ourselves that the model works consistently.

[Room 1A is west of Room 1B. Room 1B is west of Room 1C. Room 1C is west of Room 1D. Room 1D is west of Room 1E.

Room 2A is west of Room 2B. Room 2B is west of Room 2C. Room 2C is west of Room 2D. Room 2D is west of Room 2E. Room 2E is south of Room 1E.

Room 3A is south of room 2A and west of Room 3B. Room 3B is west of Room 3C. Room 3C is west of Room 3D. Room 3D is west of Room 3E.

Room 4A is west of Room 4B. Room 4B is west of Room 4C. Room 4C is west of Room 4D. Room 4D is west of Room 4E. Room 4E is south of Room 3E.

Room 5A is south of room 4A and west of Room 5B. Room 5B is west of Room 5C and south of Room 4B. Room 5C is west of Room 5D and south of Room 4C. Room 5D is west of Room 5E and south of Room 4D. Room 5E is south of Room 4E.]

For the sake at least of seeing what's going on in the example, let's also provide the player with the means to view the gas diffusion graphically:

The status grid is a device carried by the player. The status grid is switched on.

Every turn:
try examining the grid.

Instead of examining the status grid:

```
say "[fixed letter spacing][bar][line break]";
say "[state of room 1A][state of room 1B][state of room 1C][state of room
1D][state of room 1E][line break]";
say "[bar][line break]";
say "[state of room 2A][state of room 2B][state of room 2C][state of room
2D][state of room 2E][line break]";
say "[bar][line break]";
say "[state of room 3A][state of room 3B][state of room 3C][state of room
3D][state of room 3E][line break]";
say "[bar][line break]";
say "[state of room 4A][state of room 4B][state of room 4C][state of room
4D][state of room 4E][line break]";
say "[bar][line break]";
say "[state of room 5A][state of room 5B][state of room 5C][state of room
5D][state of room 5E][line break]";
say "[bar][variable letter spacing][line break]".
```

To say bar:
say "-----".

TLV is a concentration that varies. TLV is 30.0ppm. [Long-term exposure maximum, safe for 8 hours a day.]

TLV-STEL is a concentration that varies. TLV-STEL is 50.0ppm. [Short-term exposure maximum, safe for fifteen minutes max.]

TLV-C is a concentration that varies. TLV-C is 150.0ppm. [Absolute exposure ceiling.]

LC50 is a concentration that varies. LC50 is 300.0ppm. [Concentration at which 50 percent of test subjects die of exposure, usually expressed in terms of time and body weight; in our LC50 these are factored in for the player's weight for one minute.]

The values set for these would depend on the type of poisonous gas in question; we'd want to adjust appropriately.

To say state of (space - a room):

```
if the space is the location, say bold type;
if current concentration of space is less than 10.0ppm, say " ";
if current concentration of space is less than 100.0ppm, say " ";
say current concentration of space;
say roman type.
```

Now, in theory we might also want to account for sources and sinks, items that either inject poisonous gas into the environment or remove it again. For simplicity, we will assume that these contributions can also be calculated in ppm and that the total number of inert and poisonous gas molecules in a room never changes (so if poisonous gas molecules are added, an equal number of inert molecules are removed). If room pressure were able to change, our model would have to be improved, so let us assume for now that that never happens. We want this sink/source business to calculate before any other portion of the diffusion rulebook, so set it as a first diffusion rule.

A gas source is a kind of thing. A gas source has a concentration called the contribution. The contribution of a gas source is usually 30.0ppm.

Room 2B contains a gas source called a spigot. The contribution of the spigot is 50.0ppm. Room 5A contains a gas source.

A gas sink is a kind of thing. A gas sink has a concentration called the contribution. The contribution of a gas sink is usually 30.0ppm.

Room 5E contains a gas sink called a fan. The contribution of the fan is 80.0ppm.

The first diffusion rule (this is the sources and sinks rule):
follow the sources rule;
follow the sinks rule.

This is the sinks rule:
repeat with item running through gas sinks:
let space be the location of the item;
decrease the former concentration of the space by the contribution of the item;
if the former concentration of the space is less than 0.0ppm, now the former concentration of the space is 0.0ppm.

This is the sources rule:
repeat with second item running through gas sources:
let space be the location of the second item;
increase the former concentration of the space by the contribution of the second item;
if the former concentration of the space is less than 0.0ppm, now the former concentration of the space is 0.0ppm.

Test me with "z / z / z / z / z / z / z / z / z".

74



Example Only You...

Smoke which spreads through the rooms of the map, but only every other turn.

WI

Suppose we want to have smoke that spreads from room to room, gradually filling the entire map with a clogging smoke. Having it spread every single turn would

make for a pretty rapid diffusion, so we temper this by having it spread only on even-numbered turns, instead. Conveniently, Inform by default already knows about even and odd numbers, so we can write:

"Only You..."

Section 1 - The Procedure

Every turn when the turn count is even:

- if every room is smoky, make no decision;
- let previously smoky be whether or not the location is smoky;
- repeat with area running through smoky rooms:
 - now every room which is adjacent to the area is smoky;
- if previously smoky is false and the location is smoky:
 - say "[The location] is filling rapidly with smoke."

A room can be smoky or unsmoky.

Some air is a backdrop. Air is everywhere. Instead of doing something other than examining or smelling to air: say "It's just air." Understand "smoke" as the air when the location is smoky.

Instead of examining the air in a smoky room: say "A thick layer of smoke lies just under the ceiling."

Instead of smelling the air in a smoky room: say "Agh, acrid." Instead of smelling a smoky room: try smelling the air.

After looking in a smoky room: say "A thick layer of smoke has gathered under the ceiling."

Section 2 - The Scenario

The Guide Lodge is a room. "A very spacious room capable of containing several hundred girls while they eat, talk, or do crafts. It is constructed in a not-unappealing rustic style, with floor-to-ceiling windows overlooking the lake below, and a fieldstone hearth at the center." The Guide Lodge is smoky.

The Kitchen is north of the Guide Lodge. "Multiple eight-burner ranges, ovens, and a walk-in refrigerator: you know the sort of thing."

The Industrial Pantry is east of the Kitchen. "Awe-inspiring quantities of food line every shelf, from the three-gallon tub of mayonnaise to the 50-pound tub of rice. Perhaps the most astonishing item is a bag of marshmallows big enough to double as a futon."

The player is in the Pantry.

The Hallway is west of the Guide Lodge. The description of the Hallway is "A perpetually-crammed hallway which has to handle the overflow line for the toilets." A singed sign is fixed in place in the Hallway. The description of the sign is "Where the edge of the sign has not been burnt, the legible words are '...Can Prevent Forest Fires'."

The Toilets are north of the Hallway. "Always in full use, at least when the 12-to-15s are here."

The Coat Closet is south of the Hallway. "Muddy boots may not be worn inside the lodge; instead, about 250 pair are piled here, along with their owners' damp parkas and umbrellas."

The Craft Supply Room is west of the Hallway. "A holding-depot for jugs of white glue and popsicle sticks."

Test me with "x smoke / z / z / z / z / x smoke / look".

257



Example Lethal Concentration 2

WI

Poisonous gas again, only this time it sinks.

This is a slight variation on the previous gas diffusion example: the main difference is that gas preferentially moves towards lower rooms, and will gradually settle in the bottom floor. We do this by calculating the probability of movement separately for each pair of rooms.

"Lethal Concentration"

A concentration is a kind of value. 200.9ppm specifies concentration. 200.9 ppm specifies concentration.

A room has a concentration called current concentration. A room has a concentration called former concentration.

To decide what number is the probability inverse between (space - a room) and (second space - a room):

- let guess be 20;
- let way be the best route from space to second space;
- if way is up, let guess be 50;
- if way is down, let guess be 10;
- if the guess is less than 10, decide on 10;
- decide on guess.

If we wanted, we could introduce other concerns into the calculation here: open and closed doors, windows between rooms, rooms that are outdoors vs. those that are indoors, and so on. The possibilities are numerous, so we will stick with the simple principle that our poison gas sinks.

Every turn:
follow the diffusion rules.

The diffusion rules are a rulebook.

A diffusion rule (this is the gas movement rule):
repeat with space running through rooms:
let sum be 0.0 ppm;

repeat with way running through directions:
let second space be the room way from the space;
if second space is a room:
let incoming be the former concentration of the second space divided
by the probability inverse between second space and space;
let outgoing be the former concentration of the space divided by the
probability inverse between space and second space;
let difference be incoming minus outgoing;
increase sum by the difference;
now current concentration of the space is the former concentration of the
space plus the sum.

A diffusion rule (this is the resetting concentration rule):
repeat with space running through rooms:
now the former concentration of the space is the current concentration of
the space.

The last diffusion rule (this is the lethal dosage rule):
if the current concentration of the location is greater than LC50:
say "The concentration in the air overpowers you...";
end the story;
otherwise:
if the current concentration of the location is greater than TLV-STEL:
say "You feel extremely uncomfortable in this environment."

Instead of doing something when the current concentration of the location is
greater than TLV-STEL:
if going, continue the action;
say "You can't work in this environment: your eyes and nose sting and it hurts
to breathe."

Room 1A is west of Room 1B. Room 1B is west of Room 1C. Room 1C is west
of Room 1D. Room 1D is west of Room 1E.

Room 2A is west of Room 2B and below room 1A. Room 2B is west of Room 2C
and below Room 1B. Room 2C is west of Room 2D and below Room 1C. Room
2D is west of Room 2E and below Room 1D. Room 2E is south of Room 1E and
below Room 1E.

The former concentration of Room 1C is 800.0 ppm.

The status grid is a device carried by the player. The status grid is switched on.

And just for fun, this time we'll make the grid prettier, too; but this will work only on
the Z-machine setting, not Glulx.

Every turn:
try examining the grid.

Instead of examining the status grid:
say "[unicode box drawings light down and right][top bar][unicode box
drawings light down and left][line break]";
say "[unicode box drawings light vertical]";
say "[state of room 1A][state of room 1B][state of room 1C][state of room 1D]
[state of room 1E] upstairs[line break]";

```
say "[unicode box drawings light vertical and right][middle bar][unicode box
drawings light vertical and left][line break]";
say "[unicode box drawings light vertical]";
say "[state of room 2A][state of room 2B][state of room 2C][state of room 2D]
[state of room 2E] downstairs[line break]";
say "[unicode box drawings light up and right][bottom bar][unicode box
drawings light up and left][variable letter spacing][line break]"
```

Include Unicode Character Names by Graham Nelson.

To say top bar:

```
repeat with N running from 1 to 9:
  if the remainder after dividing N by 2 is 0, say "[unicode box drawings light
down and horizontal]";
  otherwise say "[unicode box drawings light horizontal]" .
```

To say middle bar:

```
repeat with N running from 1 to 9:
  if the remainder after dividing N by 2 is 0, say "[unicode box drawings light
vertical and horizontal]";
  otherwise say "[unicode box drawings light triple dash horizontal]" .
```

To say bottom bar:

```
repeat with N running from 1 to 9:
  if the remainder after dividing N by 2 is 0, say "[unicode box drawings light
up and horizontal]";
  otherwise say "[unicode box drawings light horizontal]" .
```

TLV is a concentration that varies. TLV is 30.0ppm. [Long-term exposure maximum, safe for 8 hours a day.]

TLV-STEL is a concentration that varies. TLV-STEL is 50.0ppm. [Short-term exposure maximum, safe for fifteen minutes max.]

TLV-C is a concentration that varies. TLV-C is 150.0ppm. [Absolute exposure ceiling.]

LC50 is a concentration that varies. LC50 is 300.0ppm. [Concentration at which 50 percent of test subjects die of exposure, usually expressed in terms of time and body weight; in our LC50 these are factored in for the player's weight for one minute.]

Include Basic Screen Effects by Emily Short.

To say state of (space - a room):

```
if the current concentration of space is less than TLV, say blue letters;
if the current concentration of space is TLV, say blue letters;
if the current concentration of space is greater than TLV, say green letters;
if the current concentration of space is greater than TLV-STEL, say yellow
letters;
if the current concentration of space is greater than TLV-C, say red letters;
say "[unicode square with diagonal crosshatch fill]";
say default letters;
say "[unicode box drawings light vertical]" .
```

Test me with "z / z / z / z / z / z / z / z / z".

117

Example Thirst

WI

A waterskin that is depleted as the player drinks from it.

"Thirst"

The player carries a waterskin. The waterskin can be full, partly drained, or empty. The waterskin is full. Understand "water" as the waterskin.

Instead of drinking the waterskin when the waterskin is empty:
say "There is no water left."

Instead of drinking the waterskin: if the waterskin is partly drained, now the waterskin is empty; if the waterskin is full, now the waterskin is partly drained;
say "You drink a long draught."

After printing the name of the waterskin: say " ([waterskin condition])"

Campsite is a room. "It is solid night now, and the stars have come out. Unfamiliar stars. On the other side of the valley -- a valley round-bottomed but shallow, like a soup bowl -- burn other campfires, most likely bandits. Their voices do not carry, but the smoke rises and obscures the starlight over that way."

A sleepsack is an enterable container in the Campsite. "Your sleepsack is laid out in a pocket of sandy soil and coarse grass."

The sandy soil, the stars, the distant campfires, and the coarse grass are scenery in the Campsite. Understand "smoke" as the campfires. Instead of listening in the presence of your campfire: say "All you hear are the reassuring snaps and cracks of the sticks in your fire." Understand "campfires" or "fires" as the distant campfires.

Your campfire is scenery in the Campsite. Instead of pushing, pulling, turning, tasting, or touching your campfire, say "You would burn yourself." Understand "fire" as your campfire. The description of your campfire is "A reassuring protection against wild animals and cold."

The description of the stars is "You invent constellations for them. The slingshot. The scroll. The heart (upside down)."

Instead of going nowhere when the player is in Campsite:
say "Now is not the time for wandering, alone in the dark. Better to keep here[if your campfire is visible], by the fire[end if]."

Singing is an action applying to nothing. Understand "sing" as singing.

Instead of singing:
say "You sing, deep and low, a song from home. It is a good night for singing"

and the song raises your spirits."

Test me with "i / drink water / i / drink water / i".

122

★ Example Beverage Service

WI

A potion that the player can drink.

Some kinds of game objects -- food, for instance -- can only sensibly be used once, and should then be destroyed. The EAT command already implements this, but suppose we also had a category of drinkable potions:

"Beverage Service"

A potion is a kind of thing. The sparkly blue potion is a potion carried by the player.

Level 3 is a room.

Instead of drinking a potion (called the drink):
now the drink is nowhere;
say "You quaff [the drink]. It goes down beautifully."

Test me with "drink sparkly / i".

403

★ Example Flotation

WI

Objects that can sink or float in a well, depending on their own properties and the state of the surrounding environment.

Here we want a rulebook to determine whether objects float or sink, so we create an object-based rulebook for the purpose. The more specific rules here, pertaining to corks and to inflated things, will be consulted first; then, as a default, the general flotation rule.

We also want a switch that can turn flotation off at will. The rule about the big switch will be observed before the others because the when... clause makes it more specific than the other rules in the flotation rulebook.

If we wanted, we could also put these rules into a rulebook in an explicit order, overriding Inform's automatic sorting by specificity.

"Flotation"

The Pumping House is a room.

A well is a fixed in place container in the Pumping House.

Instead of examining the well:

```
say "[if something is in the well]On the surface of the water you can see [a list
of things in the well][otherwise]There is nothing on the surface of the water, nor
can you see into the depths[end if]."
```

The well bottom is a container.

The cork, the rubber ring and a lead ingot are in the Pumping House.

A big switch is a fixed in place device in the Pumping House. "A big switch
labelled 'MAKE EVERYTHING SINK' is mounted on one wall[if switched on]. It
crackles with electricity[otherwise]. It is currently switched off and silent[end if]."

A thing can be inflated or uninflated. A thing is usually uninflated. Before printing
the name of an inflated thing: say "inflated ".

The rubber ring is inflated.

The flotation rules are an object-based rulebook.

A flotation rule for the cork: rule succeeds.

A flotation rule for an inflated thing: rule succeeds.

A flotation rule when the big switch is switched on: rule fails.

After inserting something into the well:

```
follow the flotation rules for the noun;
```

```
if the rule succeeded:
```

```
say "[The noun] bobs on the surface.";
```

```
otherwise:
```

```
move the noun to the well bottom;
```

```
say "[The noun] sinks out of sight."
```

A thing can be sinking, rising, or static. A thing is usually static.

Definition: a thing is wet:

```
if it is in the well, yes;
```

```
if it is in the well bottom, yes;
```

```
no.
```

Every turn:

```
now every thing is static;
```

```
repeat with item running through wet things:
```

```
follow the flotation rules for the item;
```

```
if the rule failed and the item is in the well, now the item is sinking;
```

```
if the rule succeeded and the item is in the well bottom, now the item is
```

```
rising;
```

```
now every rising thing is in the well;
```

```
now every sinking thing is in the well bottom;
```

```
if something is rising, say "[The list of rising things] rise[if the number of rising
things is 1]s[end if] to the surface of the well.";
```

```
if something is sinking, say "[The list of sinking things] sink[if the number of
sinking things is 1]s[end if] out of sight."
```

And finally a few description rules to make things look prettier:

Rule for writing a paragraph about the well when the well contains something:
say "The chief feature of the room is a concrete-sided well in which there
float[if the number of things in the well is 1]s[end if] [a list of things in the well]."

Rule for writing a paragraph about the well:
say "The chief feature of the room is a concrete-sided well full of water."

As we recall from the chapter on activities, "writing a paragraph about..." is an activity; activities are themselves structured as sets of object-based rulebooks. The activity "writing a paragraph about" uses three object-based rulebooks (before writing..., for writing..., after writing...). We could have made a flotation activity as well, but in general it is overkill to make an activity to make success/failure decisions. For that purpose an object-based rulebook is sufficient.

Test me with "get all / put cork in well / put ring in well / put ingot in well / x well /
get cork / get ring / switch switch on / put cork in well / put ring in well / x well /
switch switch off / switch switch on".

288



Example Xylan

WI

Creating a new command that does require an object to be named; and some comments about the choice of vocabulary, in general.

If we wanted to define a brand new verb that did affect a specific object, we might begin like this:

"Xylan"

Understand "hydrolyze [something]" as hydrolyzing. Hydrolyzing is an action applying to one thing.

Carry out hydrolyzing:
say "[The noun] cannot be hydrolyzed."

Instead of hydrolyzing the xylan:
move the xylose to the holder of the xylan;
now the xylan is nowhere;
say "At once the xylan becomes xylose."

Plant Cell Wall is a room.

There is a xylose sample. The xylan sample is a thing in Plant Cell Wall. The description of the xylan is "A polysaccharide. Totally useless. If only you had some xylose, instead!" The description of the xylose is "Awesome!"

Test me with "x xylan / hydrolyze xylan / x xylose".

Of course, how our players will ever solve this problem is another question (especially if their biology and chemistry are both rusty). When adding entirely new commands to a game, it is often a good idea to provide as many ways of phrasing the

command as possible; to drop hints about the correct phrasing within the game's text; or even to tell the player about the expanded command list in some documentation or help at the beginning of the game. So for instance we might also add

Understand "break down [something] with water" or "break [something] down with water" as hydrolyzing.

And these lines will also provide syntax for our new command, without interfering with the previous syntax. It's also good to anticipate alternative (British or American) spellings. People's typing habits are hard to overcome, even if they know you are spelling the word the other way. It is probably best not to annoy them unduly. So:

Understand "hydrolyse [something]" as hydrolyzing.

Then some text in-game might offer a clue, subtle or (since this is an example) blunt:

Instead of examining the player, say "You're a drop of water, which means that you can break down certain chemicals!"

Understand "break down [something]" or "break [something] down" as hydrolyzing.

And finally, we could try adding instructions explicitly:

Understand "help" or "hint" or "hints" or "instructions" or "info" or "about" as asking for help. Asking for help is an action out of world. Carry out asking for help: say "The following commands are understood, in addition to the standard ones: EVAPORATE, FREEZE, HYDROLYZE, SUBLIME..."

Test more with "help / x me / break down xylan"

...though of course in fact these other commands won't be available until we define them, too.

This last approach, defining all the extra commands up front, is especially useful if these commands are very technical or unusual; if they are needed early in the game, before you've a chance to educate the player; or if they are not suggested by any in-game objects. A player who encounters a tool with an obvious use, such as a hairbrush, will likely think of trying to BRUSH things with it. It's harder to rely on his guessing actions that are both outside the range of usual commands and unrelated to any of the visible props, however.



Liquid flows within containers and soaks objects that are not waterproof; any contact with a wet object can dampen our gloves.

Suppose we have some items that get wet in contact with other damp things; in particular, if we touch anything wet while wearing a pair of gloves, the gloves too get

damp. This requires that we be systematic about detecting all cases where contact occurs. So:

"Frizz"

A thing can be waterproof or porous. A container is usually waterproof. An animal is usually waterproof.

A thing can be sodden or dry.

A container can be waterfilled or empty.

Before printing the name of a sodden thing: say "sodden ". Understand the sodden property as describing a thing.

The player wears a pair of woolly gloves. Instead of wearing a sodden thing: say "You dubiously contemplate [the noun], and decide it's best to wear dry clothing."

Every turn:
follow the liquid distribution rules.

The liquid distribution rules is a rulebook.

A liquid distribution rule:
repeat with item running through containers:
if the item is open and the item is empty and the item is in a waterfilled container:
now the item is waterfilled;
if the player can see the item, say "[The item] fills up with water, of course."

A liquid distribution rule:
repeat with item running through things in a waterfilled container:
if the item is porous and the item is dry:
if the player can see the item, say "[The item] soaks through.";
now the item is sodden.

A liquid distribution rule:
repeat with item running through sodden things in the airing cupboard:
if the item is not The Last Man, now the item is dry.

The last liquid distribution rule:
if the player carries a dry copy of The Last Man, end the story finally.

Instead of examining a waterfilled container:
say "[The noun] is full of water[if the noun contains something visible]; it also contains [the list of things in the noun][end if]."

Instead of examining a container:
say "Dry inside[if the noun contains something visible], and containing [a list of things in the noun]."

A book is a kind of thing. Instead of examining a sodden book, say "[The noun] is too soaked to read, thanks to someone's carelessness." Understand "book" as

a book.

The copy of The Last Man is a book. The description of The Last Man is "Mary Shelley's very own original copy, loaned to you under the strictest of agreements[if the pond encloses the copy]. How it came to be in its present position is a long story, and not important at the moment: the critical thing is not to blame oneself - who could have anticipated the cricket bat, anyway? - but to fix it immediately, before anything worse occurs[end if]." The The Last Man is in the ziploc bag. The ziploc bag is waterproof and empty. It is openable, transparent, and closed.

The Back Garden is a room. The Back Garden contains a pond. The pond is a waterfilled container. In the pond is the ziploc bag. A carp is in the pond. The carp is an animal. Instead of taking the carp, say "You're not fast enough."

Rule for writing a paragraph about the pond:

if the pond encloses the Last Man:
say "It is a beautiful day just at the end of spring and beginning of summer. The sun shines, the trees blossom, the world conspires in Edenic cheerfulness. You can take in none of it. Your eye is on [the Last Man].";
otherwise:
say "The pond[if something is in the pond] (containing [the list of things in the pond])[end if] remains a figure of menace in an otherwise lushly verdant landscape."

Before printing the name of the Last Man when the Last Man is enclosed by the Pond: if the Last Man is dry, say "delicate, valuable".

After printing the name of something (called the target) while writing a paragraph about something:

if the Last Man is enclosed by the Pond:
if the target is in something (called the parent), say "- which is in [the parent]".

The Kitchen is west of the Back Garden. "Not in any sense your province: Mrs Peaswell gets agitated if anyone besides herself so much as boils a kettle. In general it is best to sneak through, disturbing as little as possible and preserving the cosy domestic fiction that academics cannot cook.

The nearby stairwell leads up to the second floor."

The Stairwell is above the Kitchen. "Halfway up and halfway down." The airing cupboard is a container in the Stairwell. "An airing cupboard here contains the heating system, and is the ideal place for restoring wet items to a dry state." The airing cupboard is fixed in place.

Now: whenever the player definitely touches something, we want to follow certain rules about the transfer of liquid. These rules need to come after an action's check rules (to make sure the action really occurs) and before the carry out rules (so that nothing has moved or changed yet). So we'll borrow from the chapter on rulebooks to create a whole new stage to the action, occurring between the check and the carry out phases:

The post-check rules are a rulebook.

This is the post-check stage rule:
abide by the post-check rules.

The post-check stage rule is listed before the carry out stage rule in the specific action-processing rules.

A post-check rule (this is the dry glove rule):
if we get wet:
if the player wears the gloves and the gloves are dry:
now the gloves are sodden;
say "(soaking your gloves in the process)";
continue the action.

A post-check rule (this is the wet glove rule):
if the player wears the sodden gloves:
if the The Last Man must be touched and the Last Man is not sodden:
say "(soaking the parched pages of The Last Man with the rude touch of your sodden gloves)";
now the The Last Man is sodden;
continue the action.

Before doing something when the player does not wear the gloves:
if The Last Man must be touched:
if The Last Man is dry, say "[The The Last Man] is too precious to endanger when you are not wearing gloves." instead;
otherwise say "You hesitate instinctively, then recollect that you can hardly harm [The The Last Man] any more than it has already been harmed...".

To decide whether we get wet:
if the noun is not a thing, no;
if the noun dampens us, yes;
if the second noun is not a thing, no;
if the second noun dampens us, yes;
no.

To decide whether (item - a thing) dampens us:
if the item is not liquiferous, no;
if item must be touched, yes;
no.

To decide whether (item - a thing) must be touched:
if the item is the noun and the action requires a touchable noun, yes;
if the item is the second noun and the action requires a touchable second noun, yes;
no.

Definition: a thing is liquiferous:
if it is sodden, yes;
if it is in a waterfilled container, yes;
no.

Test me with "x book / x bag / get bag / take off gloves / w / u / put gloves in cupboard / z / open bag / touch book / push book / turn book / get gloves / wear gloves / get book".

Test disaster with "open bag".

Test mishandling with "get bag / open bag / get book".

193



Example 3 AM

WI

A shake command which agitates soda and makes items thump around in boxes.

"3 AM"

Understand "shake [something preferably held]" as shaking.

Shaking is an action applying to one carried thing.

Carry out shaking:

say "Nothing results of your shaking [the noun]."

Instead of shaking a closed container when something is in the noun:

say "Something rattles inside [the noun]."

Instead of shaking a closed transparent container when something is in the noun:

say "Inside [the noun] there are banging noises produced by [the list of things contained by the noun]."

Instead of shaking an open container which contains something:

say "[The list of things contained by the noun] might fly out."

The Wawa is a room. "A convenience store, if you like to call it that, vending the usual assortment of chips, donuts, soda, and beer. There is something of a line at the sandwich counter."

The box of enrobed cakes is in the Wawa. "A box of Tastykake Enrobed Cakes has fallen off its shelf." The description is "'Enrobed Cakes' is a fancy term for 'strange sponge-like baked good, covered in a thin shell of waxy chocolate'. They are addictive, but not in a way that lets you respect yourself in the morning." The box is a closed openable container. In the box is a cake.

Instead of opening the box, say "The Wawa clerks frown on the consumption of unpurchased foodstuffs."

The can of root beer is a closed openable container carried by the player. The can of root beer is either agitated or calm.

Because the can of root beer should have some reactions to having been shaken later in the game, we need to borrow a few ideas from the chapter on Time:

Instead of shaking the can of root beer:

the can calms down in five turns from now;

say "You give the can a good hard shake.";
now the can is agitated.

Instead of listening to the can: say "It sounds [if
agitated]fizzy[otherwise]calm[end if]!"

At the time when the can calms down:
now the can is calm.

The sticky mess is fixed in place. "There is a sticky mess on the ground."

Instead of opening the agitated can of root beer:
now the can of root beer is nowhere;
now the sticky mess is in the location;
say "You open the can and fizzing sweet soda goes absolutely everywhere."

Instead of opening the calm can of root beer when the can has been agitated:
now the can of root beer is nowhere;
say "The root beer is disappointingly flat. That's what you get for shaking it
up!"

Test me with "get box / shake box / open box / shake box / listen to can / shake
can / listen to can / wait / wait / wait / wait / wait / listen to can / open can".

266



Example Lemonade

WI

Containers for liquid which keep track of how much liquid they are holding and of what kind, and allow quantities to be moved from one container to another.

Liquids, and all substances that can be mixed or broken off in partial amounts, pose a challenge to model in interactive fiction. The following example is a simple one, but adequate for many scenarios.

We start by assuming that all liquids in the game will always appear in containers. The player can pour liquids from one container to another, and the containers keep track of how full they are and describe themselves appropriately. The player can also refer to containers by content.

Mixture, however, is not allowed, nor is it possible to put liquids on other objects, pour them out on the ground, etc. These ideas would require a more complicated set-up.

"Lemonade"

A volume is a kind of value. 15.9 fl oz specifies a volume with parts ounces and tenths (optional, preamble optional).

A fluid container is a kind of container. A fluid container has a volume called a fluid capacity. A fluid container has a volume called current volume.

The fluid capacity of a fluid container is usually 12.0 fl oz. The current volume of a fluid container is usually 0.0 fl oz.

Liquid is a kind of value. The liquids are water, milk, lemonade, and iced tea. A fluid container has a liquid.

Instead of examining a fluid container:

- if the noun is empty,
 - say "You catch just a hint of [the liquid of the noun] at the bottom.";
- otherwise
 - say "[The noun] contains [current volume of the noun in rough terms] of [liquid of the noun]."

To say (amount - a volume) in rough terms:

- if the amount is less than 0.5 fl oz:
 - say "a swallow or two";
 - otherwise if tenths part of amount is greater than 3 and tenths part of amount is less than 7:
 - let estimate be ounces part of amount;
 - say "[estimate in words] or [estimate plus 1 in words] fluid ounces";
- otherwise:
 - if tenths part of amount is greater than 6, increase amount by 1.0 fl oz;
 - say "about [ounces part of amount in words] fluid ounce[s]".

Before printing the name of a fluid container (called the target) while not drinking:

- if the target is empty:
 - say "empty ";
- otherwise:
 - do nothing.

After printing the name of a fluid container (called the target) while not examining:

- unless the target is empty:
 - say " of [liquid of the target]";
 - omit contents in listing.

Instead of inserting something into a fluid container:

- say "[The second noun] has too narrow a mouth to accept anything but liquids."

Definition: a fluid container is empty if the current volume of it is 0.0 fl oz.

Definition: a fluid container is full if the current volume of it is the fluid capacity of it.

Understand "drink from [fluid container]" as drinking.

Instead of drinking a fluid container:

- if the noun is empty:
 - say "There is no more [liquid of the noun] within." instead;
- otherwise:
 - decrease the current volume of the noun by 0.2 fl oz;
 - if the current volume of the noun is less than 0.0 fl oz, now the current volume of the noun is 0.0 fl oz;
 - say "You take a sip of [the liquid of the noun][if the noun is empty], leaving [the noun] empty[end if]."

We have allowed all liquids to be drunk, but it would be possible also to add checking, if we had a game where some liquids were beverages and others were, say, motor oil.

Understand the command "fill" as something new.

Understand "pour [fluid container] in/into/on/onto [fluid container]" as pouring it into. Understand "empty [fluid container] into [fluid container]" as pouring it into. Understand "fill [fluid container] with/from [fluid container]" as pouring it into (with nouns reversed).

Understand "pour [something] in/into/on/onto [something]" as pouring it into. Understand "empty [something] into [something]" as pouring it into. Understand "fill [something] with/from [something]" as pouring it into (with nouns reversed).

Pouring it into is an action applying to two things.

Check pouring it into:

- if the noun is not a fluid container, say "You can't pour [the noun]." instead;
- if the second noun is not a fluid container, say "You can't pour liquids into [the second noun]." instead;
- if the noun is the second noun, say "You can hardly pour [the noun] into itself." instead;
- if the liquid of the noun is not the liquid of the second noun:
 - if the second noun is empty, now the liquid of the second noun is the liquid of the noun;
 - otherwise say "Mixing [the liquid of the noun] with [the liquid of the second noun] would give unsavory results." instead;
- if the noun is empty, say "No more [liquid of the noun] remains in [the noun]." instead;
- if the second noun is full, say "[The second noun] cannot contain any more than it already holds." instead.

Carry out pouring it into:

- let available capacity be the fluid capacity of the second noun minus the current volume of the second noun;
- if the available capacity is greater than the current volume of the noun, now the available capacity is the current volume of the noun;
- increase the current volume of the second noun by available capacity;
- decrease the current volume of the noun by available capacity.

Report pouring it into:

- say "[if the noun is empty][The noun] is now empty;[otherwise][The noun] now contains [current volume of the noun in rough terms] of [liquid of the noun]; [end if]";
- say "[the second noun] contains [current volume of the second noun in rough terms] of [liquid of the second noun][if the second noun is full], and is now full[end if]."

This is probably a drier description than we would actually want in our story, but it does allow us to see that the mechanics of the system are working, so we'll stick with this for the example.

Now we need a trick from a later chapter, which allows something to be described in terms of a property it has. This way, the story will understand not only "pitcher" and

"glass" but also "pitcher of lemonade" and "glass of milk" -- and, indeed, "glass of lemonade", if we empty the glass and refill it with another substance:

Understand the liquid property as describing a fluid container. Understand "of" as a fluid container.

And now the scenario itself:

The Porch is a room. The porch swing is an enterable supporter in the Porch. "An inviting swing hangs here at the end of the porch, allowing you to enjoy the summer with a cool beverage, and watch your neighbor Ted mowing his lawn with the very last manual powerless lawnmower on the block."

The glass is a fluid container carried by the player. The liquid of the glass is milk. The current volume of the glass is 0.8 fl oz.

The pitcher is a fluid container in the Porch. The fluid capacity of the pitcher is 32.0 fl oz. The current volume of the pitcher is 20.0 fl oz. The liquid of the pitcher is lemonade.

Ted's Lawn is outside from the Porch. Ted is a man in Ted's Lawn. "Ted has taken off his shirt, but still seems a bit oppressed by the sun." The description of Ted is "He looks hot. In all senses."

After deciding the scope of the player: place Ted in scope.

Instead of doing something to Ted when the player is in the Porch: say "You can't really interact with Ted from this distance, except in the sense of eyeing him surreptitiously."

Instead of giving an empty fluid container to Ted: say "Yes, taunt the poor man, why don't you?"

Instead of giving a fluid container to Ted when the liquid of the noun is milk: say "Ted looks ruefully at the milk. 'Thanks, but I'm lactose-intolerant,' he says."

The block giving rule is not listed in the check giving it to rules.

Every turn:

if Ted is in the location:

if Ted carries a fluid container (called refreshment):

try Ted drinking the refreshment;

otherwise if a random chance of 1 in 3 succeeds:

say "Ted pushes the ineffective mower over some dandelions."

Instead of someone drinking a fluid container:

if the noun is empty:

try the person asked giving the noun to the player;

otherwise:

decrease the current volume of the noun by 2.0 fl oz;

if the current volume of the noun is less than 0.0 fl oz, now the current volume of the noun is 0.0 fl oz;

say "[The person asked] gulps down some [liquid of the noun]."

After someone giving something to the player:

say "'Here,' says [the person asked], handing [the noun] back to you. 'Thanks, I owe you one.'";
end the story finally.

Test me with "x milk / x lemonade / drink lemonade / drink milk / pour lemonade into glass / drink milk / x milk / drink milk / g / i / fill glass with lemonade / drink lemonade / drop glass / drink lemonade / pitcher".

Test Ted with "out / give milk to ted / drink milk / g / g / g / give glass to ted / in / fill glass with lemonade / out / give lemonade to ted / wait / z / z / z".

267



Example Savannah

WI

Using the liquid implementation demonstrated in Lemonade for putting out fires.

Here we build very slightly on the existing liquid implementation to add a puzzle where the player puts out a fire with a bucket of water. Most of the liquid implementation remains the same as before, but now we understand the names of containers according to the liquids they contain.

The new material, pertaining to extinguishing fires, is at the bottom in section 2.

"Savannah"

Section 1 - Essentials of Liquid

A volume is a kind of value. 15.9 fl oz specifies a volume with parts ounces and tenths (optional, preamble optional).

A fluid container is a kind of container. A fluid container has a volume called a fluid capacity. A fluid container has a volume called current volume.

The fluid capacity of a fluid container is usually 12.0 fl oz. The current volume of a fluid container is usually 0.0 fl oz.

Liquid is a kind of value. A fluid container has a liquid.

Instead of examining a fluid container:

if the noun is empty,
say "You catch just a hint of [the liquid of the noun] at the bottom.";
otherwise
say "[The noun] contains [current volume of the noun in rough terms] of [liquid of the noun]."

To say (amount - a volume) in rough terms:

if the amount is less than 0.5 fl oz:
say "a swallow or two";
otherwise if tenths part of amount is greater than 3 and tenths part of amount is less than 7:

let estimate be ounces part of amount;
say "[estimate in words] or [estimate plus 1 in words] fluid ounces";
otherwise:
if tenths part of amount is greater than 6, increase amount by 1.0 fl oz;
say "about [ounces part of amount in words] fluid ounce[s]".

Before printing the name of a fluid container (called the target) while not drinking:
if the target is empty:
say "empty ";
otherwise:
do nothing.

After printing the name of a fluid container (called the target) while not examining:
unless the target is empty:
say " of [liquid of the target]";
omit contents in listing.

Instead of inserting something into a fluid container:
say "[The second noun] has too narrow a mouth to accept anything but liquids."

Definition: a fluid container is empty if the current volume of it is 0.0 fl oz.
Definition: a fluid container is full if the current volume of it is the fluid capacity of it.

Understand "drink from [fluid container]" as drinking.

Instead of drinking a fluid container:
if the noun is empty:
say "There is no more [liquid of the noun] within." instead;
otherwise:
decrease the current volume of the noun by 0.2 fl oz;
if the current volume of the noun is less than 0.0 fl oz, now the current volume of the noun is 0.0 fl oz;
say "You take a sip of [the liquid of the noun][if the noun is empty], leaving [the noun] empty[end if]."

Understand the command "fill" as something new.

Understand "pour [fluid container] in/into/on/onto [fluid container]" as pouring it into. Understand "empty [fluid container] into [fluid container]" as pouring it into. Understand "fill [fluid container] with/from [fluid container]" as pouring it into (with nouns reversed).

Understand "pour [something] in/into/on/onto [something]" as pouring it into. Understand "empty [something] into [something]" as pouring it into. Understand "fill [something] with/from [something]" as pouring it into (with nouns reversed).

Pouring it into is an action applying to two things.

Check pouring it into:
if the noun is not a fluid container, say "You can't pour [the noun]." instead;
if the second noun is not a fluid container, say "You can't pour liquids into [the second noun]." instead;

if the noun is the second noun, say "You can hardly pour [the noun] into itself."
instead;
if the liquid of the noun is not the liquid of the second noun:
if the second noun is empty, now the liquid of the second noun is the liquid
of the noun;
otherwise say "Mixing [the liquid of the noun] with [the liquid of the second
noun] would give unsavory results." instead;
if the noun is empty, say "No more [liquid of the noun] remains in [the noun]."
instead;
if the second noun is full, say "[The second noun] cannot contain any more
than it already holds." instead.

Carry out pouring it into:

let available capacity be the fluid capacity of the second noun minus the
current volume of the second noun;
if the available capacity is greater than the current volume of the noun, now
the available capacity is the current volume of the noun;
increase the current volume of the second noun by available capacity;
decrease the current volume of the noun by available capacity.

Report pouring it into:

say "[if the noun is empty][The noun] is now empty;[otherwise][The noun] now
contains [current volume of the noun in rough terms] of [liquid of the noun]; [end
if]";
say "[the second noun] contains [current volume of the second noun in rough
terms] of [liquid of the second noun][if the second noun is full], and is now
full[end if]."

Understand the liquid property as describing a fluid container. Understand "of" as
a fluid container.

Section 2 - Putting Out Fires

The Beach is a room. "The Atlantic stretches east to the horizon, though it is at
low tide at the moment. It is dawn: time to pack up and go home."

We will skip implementing the Pacific ocean itself, though the example Lakeside
Living shows how to incorporate large bodies of water into our liquid simulation.

The liquids are seawater. [We could include others, but for the moment...]

Instead of drinking a fluid container when the liquid of the noun is seawater:
say "Blech!"

The bucket is a fluid container carried by the player. The liquid of the bucket is
seawater. The current volume of the bucket is 64.0 fl oz.

The fire is a fixed in place thing in the beach. "A low fire crackles here, left over
from an attempt at s'mores much earlier in the evening."

Instead of touching or rubbing or taking the fire, say "You're not such a glutton
for punishment."

Instead of pouring something into the fire:
now the fire is nowhere;

now the current volume of the noun is 0.0 fl oz;
say "[The second noun] goes out in a great hiss."

Test me with "drink seawater / pour seawater on fire / x bucket / i".

This is still a specific implementation: if we wanted to weave liquids together with a full-scale burning model (as in "In Fire or in Flood"), where pretty much any object in the game can be flaming (currently on fire) or damp (extinguished), we might generalize our rule to

Instead of pouring something into a flaming thing:
now the second noun is damp;
now the current volume of the noun is 0.0 fl oz;
say "[The second noun] goes out in a great hiss."

Of course, the merging of fire and liquids also raises the possibility of gasoline and explosives, of heating and boiling liquids, etc.: as always, it's wise to incorporate a simulation that is only as detailed as the game's interactions really justify.

276



Example Noisy Cricket

WI

Implementing liquids that can be mixed, and the components automatically recognized as matching one recipe or another.

Our previous experiments into liquid have not dealt with the possibility of mixing components, but that is because for most games, tracking the details of mixture is overkill.

But let's suppose that this time we do want to have mixed liquids; moreover, we want a way to describe the mixtures to the player inventively, so that if he hits specific combinations those combinations are recognized: calling the result a martini, say, rather than just "a mixture of vodka and vermouth".

The implementation that follows relies on a fairly simple idea from linear algebra. Any given liquid can be expressed as a vector in N-space, where N is the number of available ingredients and the length of the vector depends on how much of each ingredient is used; then we find the recipe that best describes the liquid by taking the dot product of our liquid vector with a bunch of sample vectors and selecting the one with the largest result.

If this does not make sense, don't worry: it's not necessary to understand the idea to use the code.

Any implementation involving a large number of place values is always a bit challenging in integer arithmetic. This examples assumes that no bodies of liquid will ever be very large, and that the proportions of ingredients in a mixture will not be vastly askew. (No 20-parts-to-1 proportions, for instance.) This probably works reasonably well for the cocktails that we make the basis of the example.

"Noisy Cricket"

Part 1 - Volumes and Mixtures

A volume is a kind of value. 15.9 fl oz specifies a volume with parts ounces and tenths (optional, preamble optional).

A fluid container is a kind of container. A fluid container has a volume called a fluid capacity. A fluid container has a volume called creme de menthe volume. A fluid container has a volume called vodka volume. A fluid container has a volume called cacao volume.

The fluid capacity of a fluid container is usually 12.0 fl oz. The creme de menthe volume of a fluid container is usually 0.0 fl oz. The vodka volume of a fluid container is usually 0.0 fl oz. The cacao volume of fluid container is usually 0.0 fl oz.

To decide what volume is the current volume of (item - a fluid container):
 let total be the creme de menthe volume of the item;
 increase total by the vodka volume of the item;
 increase total by the cacao volume of the item;
 decide on total.

Instead of examining a fluid container:

if the noun is empty,
 say "You catch just a hint of [the nominal descriptor of the noun] at the bottom.";
 otherwise
 say "[The noun] contains [current volume of the noun in rough terms] of [adjectival descriptor of the noun] [nominal descriptor of the noun]."

Adjectival descriptor is a kind of value. The adjectival descriptors are strong, chocolatey, minty, perfect, and pure.

Nominal descriptor is a kind of value. The Nominal descriptors are creme de menthe, vodka, creme de cacao, grasshopper, chocolate vodka, mint vodka, chocolate martini, mintini, chocolate mint martini.

Our table of mixtures is expressed in parts: so if a recipe contains one part X and two parts Y, we would put "1" in the first column and "2" in the second column.

Table of Mixtures

rating	creme de menthe comp	vodka comp	cacao comp	adjectival descriptor	nominal descriptor
0.0 fl oz 1	0	0	0	minty	creme de menthe
0.0 fl oz 0	1	0	0	chocolatey	vodka
0.0 fl oz 0	0	1	0	chocolatey	creme de cacao
0.0 fl oz 1	2	0	0	chocolatey	mintini
0.0 fl oz 1	0	1	1	chocolatey	grasshopper
0.0 fl oz 0	2	1	1	chocolatey	chocolate martini
0.0 fl oz 0	3	1	0	chocolatey	chocolate vodka
0.0 fl oz 1	3	0	0	chocolatey	mint vodka
0.0 fl oz 1	2	1	1	chocolatey	chocolate mint martini

A fluid container has an adjectival descriptor. A fluid container has a nominal descriptor. Understand the adjectival descriptor property as describing a fluid container. Understand the nominal descriptor property as describing a fluid container.

To decide what number is (quantity - a number) squared:
decide on quantity times quantity.

To score mixtures in (item - a fluid container):

repeat through Table of Mixtures:
let total line parts be creme de menthe comp entry squared;
let total line parts be total line parts plus vodka comp entry squared;
let total line parts be total line parts plus cacao comp entry squared;
let creme de menthe score be creme de menthe comp entry times the
creme de menthe volume of item;
let vodka score be vodka comp entry times the vodka volume of item;
let cacao score be cacao comp entry times the cacao volume of item;
let total score be creme de menthe score plus vodka score;
let total score be total score plus cacao score;
let total score be total score times calibration for total line parts;
now rating entry is total score;
if total line parts is 1, now adjectival descriptor entry is pure;
otherwise now adjectival descriptor entry is perfect;
[and for creme de menthe...]
now creme de menthe comp entry is creme de menthe comp entry plus 1;
let total line parts be creme de menthe comp entry squared plus vodka
comp entry squared;
let total line parts be total line parts plus cacao comp entry squared;
let creme de menthe score be creme de menthe comp entry times the
creme de menthe volume of item;
let vodka score be vodka comp entry times the vodka volume of item;
let cacao score be cacao comp entry times the cacao volume of item;
let total score be creme de menthe score plus vodka score;
let total score be total score plus cacao score;
let total score be total score times calibration for total line parts;
if total score is greater than rating entry, now adjectival descriptor entry is
minty;
now creme de menthe comp entry is creme de menthe comp entry minus 1;
[and for vodka...]
now vodka comp entry is vodka comp entry plus 1;
let total line parts be creme de menthe comp entry squared plus vodka
comp entry squared;
let total line parts be total line parts plus cacao comp entry squared;
let creme de menthe score be creme de menthe comp entry times the
creme de menthe volume of item;
let vodka score be vodka comp entry times the vodka volume of item;
let cacao score be cacao comp entry times the cacao volume of item;
let total score be creme de menthe score plus vodka score;
let total score be total score plus cacao score;
let total score be total score times calibration for total line parts;
if total score is greater than rating entry, now adjectival descriptor entry is
strong;
now vodka comp entry is vodka comp entry minus 1;
[and for cacao...]
now cacao comp entry is cacao comp entry plus 1;
let total line parts be creme de menthe comp entry squared plus vodka
comp entry squared;
let total line parts be total line parts plus cacao comp entry squared;
let creme de menthe score be creme de menthe comp entry times the
creme de menthe volume of item;
let vodka score be vodka comp entry times the vodka volume of item;
let cacao score be cacao comp entry times the cacao volume of item;

```

    let total score be creme de menthe score plus vodka score;
    let total score be total score plus cacao score;
    let total score be total score times calibration for total line parts;
    if total score is greater than rating entry, now adjectival descriptor entry is
chocolatey;
    now cacao comp entry is cacao comp entry minus 1.

```

```

To identify mixture in (item - a fluid container):
score mixtures in item;
sort Table of Mixtures in reverse rating order;
choose row 1 in Table of Mixtures;
now nominal descriptor of the item is nominal descriptor entry;
let sample vodka be vodka comp entry; [Now keep track of all these]
let sample creme de menthe be creme de menthe comp entry;
let sample cacao be cacao comp entry;
if rating entry divided by 100 is the current volume of the item:
    now adjectival descriptor of the item is pure;
otherwise:
    now adjectival descriptor of the item is adjectival descriptor entry.

```

```

To decide what number is the raw quantity of (item volume - a volume):
let raw be item volume divided by 0.5 fl oz;
decide on raw.

```

```

To decide what number is calibration for (total - a number):
if total is an initial listed in the table of Multipliers, decide on result entry;
decide on 21.

```

Here we cheat on our arithmetic. The following chart just provides values corresponding roughly to $1/(\text{sqrt}(x))$, but since Inform does not deal very gracefully with square roots or fractions, we will calculate this elsewhere and just supply the answers in the code:

Table of Multipliers

initial	result
1	100
2	71
3	57
4	50
5	44
6	41
7	38
8	35
9	33
10	31
11	30
12	29
13	28
14	27
15	26
16	25
17	24
18	24
19	23
20	22

```

When play begins:
repeat with item running through fluid containers:

```

identify mixture in item.

To say (amount - a volume) in rough terms:

if the amount is less than 0.6 fl oz:
say "half an ounce or less";
otherwise if tenths part of amount is greater than 3 and tenths part of amount is less than 7:
let estimate be ounces part of amount;
say "[estimate in words] or [estimate plus 1 in words] fluid ounces";
otherwise:
if tenths part of amount is greater than 6, increase amount by 1.0 fl oz;
say "about [ounces part of amount in words] fluid ounce[s]".

Before printing the name of a fluid container (called the target) while not drinking or pouring:

if the target is empty:
say "empty ";
otherwise:
do nothing.

After printing the name of a fluid container (called the target) while not examining or pouring:

unless the target is empty:
say " of [adjectival descriptor of the target] [nominal descriptor of the target]";
omit contents in listing.

Instead of inserting something into a fluid container:

say "[The second noun] has too narrow a mouth to accept anything but liquids."

Definition: a fluid container is empty if the current volume of it is 0.0 fl oz.

Definition: a fluid container is full if the current volume of it is the fluid capacity of it.

Understand "drink from [fluid container]" as drinking.

Instead of drinking a fluid container:

if the noun is empty:
say "There is no more [nominal descriptor of the noun] within." instead;
otherwise:
let cacao loss be the consumed cacao of the noun out of sip volume;
let creme de menthe loss be the consumed creme de menthe of the noun out of sip volume;
let vodka loss be the consumed vodka of the noun out of sip volume;
decrease the cacao volume of the noun by the cacao loss;
decrease the creme de menthe volume of the noun by creme de menthe loss;
decrease the vodka volume of the noun by vodka loss;
say "You take a sip of [the nominal descriptor of the noun][if the noun is empty], leaving [the noun] empty[end if].".

Sip volume is a volume that varies. Sip volume is 0.5 fl oz.

To decide what volume is the consumed cacao of (item - a fluid container) out of (total consumption - a volume):

- let new volume be the cacao volume of the item times 100;
- let percentage be the new volume divided by the current volume of the item;
- let consumed volume be the percentage times total consumption;
- let consumed volume be consumed volume divided by 100;
- if consumed volume is greater than the cacao volume of the item, decide on the cacao volume of the item;
- decide on consumed volume.

To decide what volume is the consumed creme de menthe of (item - a fluid container) out of (total consumption - a volume):

- let new volume be the creme de menthe volume of the item times 100;
- let percentage be the new volume divided by the current volume of the item;
- let consumed volume be the percentage times total consumption;
- let consumed volume be consumed volume divided by 100;
- if consumed volume is greater than the creme de menthe volume of the item, decide on the creme de menthe volume of the item;
- decide on consumed volume.

To decide what volume is the consumed vodka of (item - a fluid container) out of (total consumption - a volume):

- let new volume be the vodka volume of the item times 100;
- let percentage be the new volume divided by the current volume of the item;
- let consumed volume be the percentage times total consumption;
- let consumed volume be consumed volume divided by 100;
- if consumed volume is greater than the vodka volume of the item, decide on the vodka volume of the item;
- decide on consumed volume.

Part 2 - Filling

Understand the command "fill" as something new.

Understand "fill [something] with/from [something]" as filling it with.

Filling it with is an action applying to two things. Carry out filling it with: try pouring the second noun into the noun instead.

Understand "pour [fluid container] in/into/on/onto [fluid container]" as pouring it into. Understand "empty [fluid container] into [fluid container]" as pouring it into.

Understand "pour [something] in/into/on/onto [something]" as pouring it into. Understand "empty [something] into [something]" as pouring it into.

Pouring it into is an action applying to two things.

Check pouring it into:

- if the noun is not a fluid container, say "You can't pour [the noun]." instead;
- if the second noun is not a fluid container, say "You can't pour liquids into [the second noun]." instead;
- if the noun is the second noun, say "You can hardly pour [the noun] into itself." instead;
- if the noun is empty, say "No more [nominal descriptor of the noun] remains in [the noun]." instead;

if the second noun is full, say "[The second noun] cannot contain any more than it already holds." instead.

Carry out pouring it into:

let available capacity be the fluid capacity of the second noun minus the current volume of the second noun;

if the available capacity is greater than the current volume of the noun, now the available capacity is the current volume of the noun;

let cacao loss be the consumed cacao of the noun out of available capacity;
let creme de menthe loss be the consumed creme de menthe of the noun out of available capacity;

let vodka loss be the consumed vodka of the noun out of available capacity;

decrease the cacao volume of the noun by the cacao loss;

decrease the creme de menthe volume of the noun by creme de menthe loss;

decrease the vodka volume of the noun by vodka loss;

increase the cacao volume of the second noun by the cacao loss;

increase the creme de menthe volume of the second noun by creme de menthe loss;

increase the vodka volume of the second noun by vodka loss.

Report pouring it into:

identify mixture in noun;

identify mixture in second noun;

say "[if the noun is empty][The noun] is now empty; [otherwise][The noun] now contains [current volume of the noun in rough terms] of [nominal descriptor of the noun]; [end if]";

say "[the second noun] contains [current volume of the second noun in rough terms] of [adjectival descriptor of the second noun] [nominal descriptor of the second noun][if the second noun is full], and is now full[end if]."

Understand "of" as a fluid container.

Part 3 - Scenario

When play begins: say "When you decided to try Mixology WS102 (*cross-listed with Women's Studies), you envisioned yourself writing essays about gender discrimination during the Prohibition, say, or reading essays on male vs. female metabolism of alcohol. But no: MxWS102 turns out to be about... mixing the perfect chocolate mint martini."

The College of Mixology is a room. The bar is a supporter in the college.

The cocktail glass is a fluid container carried by the player. The fluid capacity of the cocktail glass is 4.0 fl oz.

The flask is a fluid container carried by the player. The vodka volume of the flask is 4.0 fl oz.

The jigger is a fluid container carried by the player. The fluid capacity of the jigger is 1.0 fl oz.

The small measure is a fluid container carried by the player. The fluid capacity of the small measure is 0.5 fl oz.

The decanter is a fluid container on the bar. The fluid capacity of the decanter is 32.0 fl oz. The creme de menthe volume of the decanter is 20.0 fl oz.

The bottle is a fluid container carried by the player. The cacao volume of the bottle is 10.0 fl oz.

Test me with "i / pour flask in jigger / pour jigger in glass / pour bottle in jigger / pour jigger in glass / pour bottle in jigger / pour jigger in glass / pour decanter in jigger / pour jigger in glass / drink glass / g / g / x glass / pour flask in glass".

333



Example Lakeside Living

WI

Similar to "Lemonade", but with bodies of liquid that can never be depleted, and some adjustments to the "fill" command so that it will automatically attempt to fill from a large liquid source if possible.

Much of what follows is identical to "Lemonade" earlier; the new material begins at Part 2.

"Lakeside Living"

A volume is a kind of value. 15.9 fl oz specifies a volume with parts ounces and tenths (optional, preamble optional).

A fluid container is a kind of container. A fluid container has a volume called a fluid capacity. A fluid container has a volume called current volume.

The fluid capacity of a fluid container is usually 12.0 fl oz. The current volume of a fluid container is usually 0.0 fl oz.

Liquid is a kind of value. The liquids are water, absinthe, and iced tea. A fluid container has a liquid.

Instead of examining a fluid container:

if the noun is empty,
say "You catch just a hint of [the liquid of the noun] at the bottom.";
otherwise
say "[The noun] contains [current volume of the noun in rough terms] of [liquid of the noun]."

To say (amount - a volume) in rough terms:

if the amount is less than 0.5 fl oz:
say "a swallow or two";
otherwise if tenths part of amount is greater than 3 and tenths part of amount is less than 7:
let estimate be ounces part of amount;
say "[estimate in words] or [estimate plus 1 in words] fluid ounces";
otherwise:
if tenths part of amount is greater than 6, increase amount by 1.0 fl oz;
say "about [ounces part of amount in words] fluid ounce[s]".

Before printing the name of a fluid container (called the target) while not drinking or pouring:

```
if the target is empty:
    say "empty ";
otherwise:
    do nothing.
```

After printing the name of a fluid container (called the target) while not examining or pouring:

```
unless the target is empty:
    say " of [liquid of the target]";
    omit contents in listing.
```

Instead of inserting something into a fluid container:

```
say "[The second noun] has too narrow a mouth to accept anything but liquids."
```

Definition: a fluid container is empty if the current volume of it is 0.0 fl oz.

Definition: a fluid container is full if the current volume of it is the fluid capacity of it.

Understand "drink from [fluid container]" as drinking.

Instead of drinking a fluid container:

```
if the noun is empty:
    say "There is no more [liquid of the noun] within." instead;
otherwise:
    decrease the current volume of the noun by 0.2 fl oz;
    if the current volume of the noun is less than 0.0 fl oz, now the current
    volume of the noun is 0.0 fl oz;
    say "You take a sip of [the liquid of the noun][if the noun is empty], leaving
    [the noun] empty[end if]."
```

Part 2 - Filling

Understand the command "fill" as something new.

Here we want Inform to prefer full liquid sources to other containers when it chooses an end to a player's unfinished or ambiguous command. And so:

Understand "fill [fluid container] with/from [full liquid source]" as filling it with.

Understand "fill [fluid container] with/from [fluid container]" as filling it with.

Both grammar lines point to the same ultimate outcome; the purpose of specifying both is to tell Inform to check thoroughly for full liquid sources before falling back on other fluid containers when making its decisions.

Understand "fill [something] with/from [something]" as filling it with.

Filling it with is an action applying to two things. Carry out filling it with: try pouring the second noun into the noun instead.

Understand "pour [fluid container] in/into/on/onto [fluid container]" as pouring it into. Understand "empty [fluid container] into [fluid container]" as pouring it into.

Understand "pour [something] in/into/on/onto [something]" as pouring it into.
Understand "empty [something] into [something]" as pouring it into.

Pouring it into is an action applying to two things.

Check pouring it into:

- if the noun is not a fluid container, say "You can't pour [the noun]." instead;
- if the second noun is not a fluid container, say "You can't pour liquids into [the second noun]." instead;
- if the noun is the second noun, say "You can hardly pour [the noun] into itself." instead;
- if the liquid of the noun is not the liquid of the second noun:
 - if the second noun is empty, now the liquid of the second noun is the liquid of the noun;
 - otherwise say "Mixing [the liquid of the noun] with [the liquid of the second noun] would give unsavory results." instead;
- if the noun is empty, say "No more [liquid of the noun] remains in [the noun]." instead;
- if the second noun is full, say "[The second noun] cannot contain any more than it already holds." instead.

Carry out pouring it into:

- let available capacity be the fluid capacity of the second noun minus the current volume of the second noun;
- if the available capacity is greater than the current volume of the noun, now the available capacity is the current volume of the noun;
- increase the current volume of the second noun by available capacity;
- decrease the current volume of the noun by available capacity.

Report pouring it into:

- say "[if the noun is empty][The noun] is now empty;[otherwise][The noun] now contains [current volume of the noun in rough terms] of [liquid of the noun]; [end if]";
- say "[the second noun] contains [current volume of the second noun in rough terms] of [liquid of the second noun][if the second noun is full], and is now full[end if]."

Understand the liquid property as describing a fluid container. Understand "of" as a fluid container.

And now we add our liquid source kind, which will represent lakes, absinthe fountains, and any other infinite supplies of liquid we might need. Note that 3276.7 is the largest possible number of fluid ounces available to us.

A liquid source is a kind of fluid container. A liquid source has a liquid. A liquid source is usually scenery. The fluid capacity of a liquid source is usually 3276.7 fl oz. The current volume of a liquid source is usually 3276.7 fl oz. Instead of examining a liquid source: say "[The noun] is full of [liquid of the noun]."

Carry out pouring a liquid source into something: now the current volume of the noun is 3276.7 fl oz.

We want filling things from liquid sources to work the same way as usual, with the distinction that a) the liquid source never depletes in quantity (hence the carry-out

rule resetting its fullness); and b) we should report the results a bit differently as well:

After pouring a liquid source into a fluid container:

say "You fill [the second noun] up with [liquid of the noun] from [the noun]."

On the other hand, pouring liquids into a liquid source needs to work completely differently from pouring liquids into anything else. Let's say we're going to allow any liquid at all to be dumped into rivers and streams (environmental protections evidently are not very well-enforced in this scenario):

Instead of pouring a fluid container into a liquid source:

if the noun is empty, say "[The noun] is already empty." instead;

now the current volume of the noun is 0.0 fl oz;

say "You dump out [the noun] into [the second noun]."

A couple of minor refinements:

Swimming is an action applying to nothing. Understand "swim" or "dive" as swimming.

Instead of swimming in the presence of a liquid source:

say "You don't feel like a dip just now."

Before inserting something into a liquid source: say "[The noun] would get lost and never be seen again." instead.

Part 3 - Scenario

The Lakeside is a room. The Lakeside swing is an enterable supporter in the Lakeside. "Here you are by the lake, enjoying a summery view."

The glass is a fluid container carried by the player. The liquid of the glass is absinthe. The current volume of the glass is 0.8 fl oz.

The pitcher is a fluid container in the Lakeside. The fluid capacity of the pitcher is 32.0 fl oz. The current volume of the pitcher is 20.0 fl oz. The liquid of the pitcher is absinthe.

The lake is a liquid source. It is in the Lakeside.

The player wears a bathing outfit. The description of the bathing outfit is "Stylishly striped in blue and white, and daringly cut to reveal almost all of your calves, and quite a bit of upper arm, as well. You had a moral struggle, purchasing it; but mercifully the lakeshore is sufficiently secluded that no one can see you in this immodest apparel."

Instead of taking off the outfit: say "What odd ideas come into your head sometimes!"

Test me with "fill glass / empty absinthe into lake / fill glass / swim / drink lake / drink / x water / x lake".

★ Example Pizza Prince

Providing a pizza buffet from which the player can take as many pieces as he wants.

Suppose we want the player to have a pizza buffet from which he can take a number of slices. But we don't want to actually put the slices there in front of him, because "you can see 17 slices of pizza here" is not the descriptive effect we want, and because we want to pretend, at least, that the pizza supply is nearly infinite. In fact, we're going to replenish the supply by allowing eaten slices to return to the buffet table (safer in IF than in real life).

To do this, we create one object to stand in for the pizza supply, but whenever the player tries to take it, we give him a different "pizza slice" object instead. Thus:

"Pizza Prince"

The Pizza Prince is a room.

The buffet table is a supporter in Pizza Prince.

The pizza selection is a thing on the buffet table. Understand "slice" as the pizza selection. The description is "They are all cheese-only, and all luke-warm."

Rule for writing a paragraph about the buffet table:

say "On [the buffet table] is [a pizza selection]. [description of the pizza selection][line break]".

Now we introduce our actual pizza slices, which are retained in a container out of play until they're needed:

A pizza slice is a kind of thing. 10 pizza slices are in Pizza Limbo. A pizza slice is always edible. [After a fashion, anyway.]

In this example we've set that supply to be artificially small, to make it easier to test what happens when the player reaches the limit; but we could provide many more slices to start with in Pizza Limbo, and the aim in practice would be to pick a number high enough (such as 50 or 100) that the average player will get bored of TAKE PIZZA long before he reaches the limit.

The main thing to be aware of is that objects consume memory in the game file, so creating a large number of pizza slices might bulk the game out. This is more of a concern if we're compiling for the Z-machine than if we're compiling for Glulx.

Whenever the player tries to take the selection, we want him to wind up holding an individual slice instead; but of course we need to check and make sure that he hasn't exhausted the pizza slice supply.

Instead of taking the pizza selection:

let chosen slice be a random pizza slice in Pizza Limbo;

```

if chosen slice is nothing: [That is, there were no slices remaining]
    say "[manager refusal]";
otherwise:
    move the chosen slice to the player;
    say "Taken (gingerly)."
```

To say manager refusal:

```

say "[one of]'Hey!' barks a hitherto-unseen manager from behind you. 'It's an
'all you can eat' buffet, not an 'all you can stuff down your pants' buffet.'[or]You
are conscious of a disapproving huff from the manager, so you refrain.[stopping]"
```

That's fine for the case where the player is taking a new slice of pizza explicitly, but we need to handle it a little differently if the taking action is generated in response to EAT PIZZA. In that case, we need to take the slice and also change the identity of the noun, because after the implicit take action happens, the game will test whether the player is holding the noun before attempting to eat it. So we need to refocus its attention:

Rule for implicitly taking the pizza selection:

```

let chosen slice be a random pizza slice in Pizza Limbo;
if chosen slice is nothing: [That is, there were no slices remaining]
    say "[manager refusal]";
otherwise:
    move the chosen slice to the player;
    say "(helping yourself from the selection)";
    now the noun is the chosen slice.
```

And finally, a bit of touch-up:

Rule for clarifying the parser's choice of the pizza selection while taking:

```

say "(from the magnificent selection before you)[line break]"
```

For tidiness, we should probably also return the consumed pizza slices to Pizza Limbo so that they can be re-used later:

After eating a pizza slice:

```

move the noun to Pizza Limbo;
continue the action.
```

```

Test me with "i / get pizza / g / i / get pizza / drop pizza / look / get pizza / g / look
/ eat pizza / g / g / g / g / get pizza / g / g / g / g / g / g / g / g / g / g / g / g / i / eat
pizza / take pizza / g".
```



Example Extra Supplies

A supply of red pens from which the player can take another pen only if he doesn't already have one somewhere in the game world.

Suppose we have a supply closet in our game from which the player is allowed to take red pens. To keep modeling simple, we only allow him to have one in play at a

time, and we test this by seeing whether the red pen is "off-stage" before moving it to his possession.

This approach might seem no different from having a single red pen sitting in the closet, but it may be preferable, for two reasons. First, it's not very plausible for a supply closet to contain nothing but a single red pen (well, assuming a well-regulated supplier, anyway); and second, it gives the player a way to get a new red pen should the original be destroyed in a tragic handwriting accident.

"Extra Supplies"

The Supply Closet is a room. A supply of red pens is in the Supply Closet. Understand "pen" as the supply of red pens when the red pen is not visible.

There is a red pen.

Instead of taking the supply of red pens:

if the red pen is off-stage:

move the red pen to the player;

say "You help yourself to a fresh red pen.";

otherwise:

say "You're only allowed one pen at a time. The department secretary is very strict."

South of the Supply Closet is the Furnace Room. The incinerator is a thing in the Furnace Room. It is a container. "The incinerator is here, working full blast."

After inserting something into the incinerator:

now the noun is nowhere;

say "A fiery blast consumes [the noun]!"

Test me with "get pen / i / get pen / get supply / s / put pen in incinerator / n / get pen".

97

Example Ming Vase

WI

ATTACK or DROP break and remove fragile items from play.

In emulation of a certain annoying aspect of the original Adventure, in which there is a Ming vase that cannot safely be dropped:

"Ming Vase"

A thing can be strong or fragile. A thing is usually strong.

Instead of attacking or dropping a fragile thing:

now the noun is nowhere;

say "[The noun] breaks into thousands of pieces!"

The Cave is a room. The Ming vase is carried by the player. The vase is fragile.

We could also implement an additional refinement from Adventure, that a fragile thing is safe if dropped when there is a cushion nearby.

The pillow is a portable supporter. It is carried by the player.

Instead of dropping a fragile thing when the pillow is in the location: try putting the noun on the pillow instead.

After putting a fragile thing on the pillow:
say "You set [the noun] down gently on the pillow."

Test me with "drop pillow / drop vase / get vase / get pillow / drop vase".

In this rule, the pillow is the second noun; if we had a general rule about setting fragile things on soft things, we could say "You set [the noun] down gently on [the second noun]." for the same effect.

If we wanted to be more refined, we would provide extra code so that breaking a container or a supporter would leave behind their contents. We will see how to do that later.

123

★ Example Spring Cleaning

WI

A character who sulks over objects that the player has broken (and which are now off-stage).

Here we have a destruction action that allows the player to break any fragile items. Once destroyed, these things are removed from play, but we can still refer to them: they are now off-stage. This makes it easy for our sulking character to list the ones that have been destroyed:

"Spring Cleaning"

A thing can be tough or fragile. A thing is usually tough.

Instead of attacking something fragile:
say "You smash [the noun] to smithereens!";
now the noun is nowhere.

A knick-knack is a kind of thing which is fragile.

Every turn when a knick-knack is off-stage and Granny Blue can see the player:
say "'Ohh,' whimpers Granny to herself softly. 'How I will miss [the list of off-stage knick-knacks]!'"

The Parlor is a room. Granny Blue is a woman in the Parlor. A china lamb, a porcelain milkmaid, a frolicking Dutch cow, and a crystal unicorn are knick-knacks in the Parlor.

Test me with "break lamb / break milkmaid / break cow / break unicorn".



Example Paddington

A CUT [something] WITH [something] command which acts differently on different types of objects.

Suppose we intend a game in which the player needs to cut things open on a regular basis. We'll want to check whether he has the proper tools handy, and deal graciously with commands such as CUT [something] when no tool is specified. So:

"Paddington"

A blade is a kind of thing.

Understand "cut [something] with [something]" as cutting it with.

Instead of cutting something:

if a blade (called the edge) is held by the player,
try cutting the noun with the edge;
otherwise say "Your fingernails are not sharp enough."

Cutting it with is an action applying to two things.

Check cutting it with:

if the noun is a person, say "That would hurt." instead;
if the second noun is not a blade, say "[The second noun] has not got enough of a blade." instead.

Carry out cutting it with:

increment the count of rips of the noun.

Report cutting it with:

say "You slash [the noun] with [the second noun]."

We'll need a way to account for all these cuts and rips.

Definition: a thing is ripped if the count of rips of it > 0. A thing has a number called the count of rips. After examining something ripped, say "You see [the count of rips of the noun in words] rip[s] in [the noun][if something is in the noun], revealing [a list of things in the noun][end if]."

Moreover, because open containers normally list their contents when examined but we'd prefer Paddington's to be mentioned in the ripping paragraph:

The examine containers rule does nothing when examining the teddy bear.

So far, so good. But suppose that we'd like cutting also to make containers be permanently open and impossible to close again. We could write an "instead" rule, but that would mean that only our instead instructions would take effect, overriding the normal cutting it with rules entirely. Better would be to add a second carry out rule:

Carry out cutting a container with something:
now the noun is open;
now the noun is unopenable.

Now our rule will occur whenever a container is cut, but play will still go on to the reporting stage. And indeed we can add more of these, of varying degrees of specificity:

Carry out cutting something which is part of something with something:
move the noun to the player.

Carry out cutting the quilt with something:
now the description of the quilt is "Horribly tattered."

For that matter, we might want to add a report rule as well, to occur after the "You slash..." rule, so that every time the player cuts something open which has contents, the contents will be listed.

Report cutting it with:
if the noun is open and the noun contains something,
say "Visible within [is-are a list of things in the noun]."

This time we do not add the condition to the rule (i.e., Report cutting an open noun...) If we did, this report rule would be more specific than the general report rule, and would occur first.

The Safehouse is a room.

The teddy bear is a closed thing in the Safehouse. The description is "Fluffy[if the head is part of the bear], with an outsized head[otherwise], but headless[end if]." The head is a closed part of the teddy bear. In the bear is a large wad of stuffing. In the head are a small wad of stuffing and a packet of smuggled diamonds.

The quilt is in the Safehouse. The description is "An old but comforting quilt."

The player carries a blade called a switchblade.

Here is a final nicety to get rid of the "which is closed" statement on our closed unopenable teddy bear, using an "activity" rule:

After printing the name of a closed unopenable container:
omit contents in listing.

Test me with "cut quilt with bear / cut quilt with switchblade / examine quilt / cut bear with switchblade / again / examine bear / cut head with switchblade / get diamonds / mourn loss of innocence".



breakage all undamaged pots are said to be "unbroken", to distinguish them from the others.

This easiest way to do this uses the "printing the name of" activity, which will come up in the following chapter:

"Terracottissima"

A flowerpot is a kind of thing. Understand "pot" as a flowerpot.

A flowerpot can be unbroken or broken. After dropping an unbroken flowerpot: say "Crack!"; now the noun is broken. Understand the broken property as describing a flowerpot.

Before printing the name of a broken flowerpot, say "broken ". Before printing the name of an unbroken flowerpot: if a flowerpot is broken, say "unbroken ".

Before printing the plural name of a broken flowerpot, say "broken ". Before printing the plural name of an unbroken flowerpot: if a flowerpot is broken, say "unbroken ".

The Herb Garden is a room. In the Herb Garden are ten unbroken flowerpots.

Test me with "get three flowerpots / drop all / look".

404



Example Kyoto

WI

Expanding the effects of the THROW something AT something command so that objects do make contact with one another.

Suppose we want to expand the function of the existing THROW SOMETHING AT command so that a thrown object actually does make contact most of the time. A glance at the Actions index tells us that the Throwing it at rulebook currently looks like this:

Throwing something at something (past tense thrown it at)
"drop [something held] at/against/on/onto [something]"

check an actor throwing something at implicitly remove thrown clothing rule
check an actor throwing something at futile to throw things at inanimate objects rule
check an actor throwing something at block throwing at rule

Some of those still look useful. We want to leave the "implicitly remove thrown clothing" rule, for instance -- no fair having the player throw a hat that's on his head. On the other hand, the "futile to throw things at inanimate objects rule" is going to have to go, because that would prevent us from ever being able to complete the throwing command. So let's get rid of that:

"Kyoto"

Part 1 - Throwing Rules

The futile to throw things at inanimate objects rule is not listed in the check throwing it at rules.

That "block throwing at" rule also looks sinister: any "block..." rule in the standard actions library is there to print a message telling the player he can't do what he's asked to do.

But it's not enough to ignore it, the way we did the "futile" rule. Since we are only expanding the command to affect inanimate objects, let's replace the "block throwing at" rule with a different one which will only prevent the player throwing things at people:

The block throwing at people rule is listed instead of the block throwing at rule in the check throwing it at rules.

This is the block throwing at people rule:
if the second noun is a person, say "That might be construed as an attack."
instead.

Now we've changed the command so that some action can sometimes be carried out here -- but we don't have any rules for what happens. It's time to create some rules for our model world.

A thing can be hard or soft. A thing can be fragile or strong. Shape is a kind of value. The shapes are round, flat, and linear. A thing has shape.

If we're actually going to allow throwing, we might want to add a couple of extra checks to the rulebook to make sure that this happens when it ought:

Check throwing it at (this is the block juggling rule):
if the player is carrying the second noun, say "It would be difficult to throw at something you are yourself holding." instead.

Check throwing it at (this is the avoid throwing things into themselves rule):
if the second noun is within the noun, say "That would be a nice magic trick."
instead.

And then the rules for the action itself:

Carry out throwing it at (this is the check aerodynamics rule):
if the noun is flat:
move noun to location;
say "[The noun], unwieldy, flutters to the ground.";
rule succeeds.

That "rule succeeds" ends the action here, if the noun is flat. If not, Inform goes on to the next rule in the carry out throwing it at rulebook:

Carry out throwing it at (this is the contact rule):
say "[The noun] hits [the second noun].[paragraph break]";

if the second noun is fragile and the noun is hard:
destroy the second noun.

Carry out throwing it at (this is the landing rule):
let destination be the location;
if the second noun is on a supporter (called endtable), let destination be the endtable;
if the second noun is a supporter, let destination be the second noun;
move the noun to the destination;
if the noun is fragile and the second noun is hard:
destroy the noun;
rule succeeds;
say "[The noun] lands [if the destination is the location]nearby[otherwise]on [the destination][end if]."

These rules are assuming some backup information, so let's provide that as well:

Reliance relates a thing (called X) to a thing (called Y) when X is part of Y or X is in Y or X is on Y. The verb to be relying on means the reliance relation.

To destroy (item - a thing):
let home be the holder of the item;
if the item is part of something (called the superstructure), let home be the holder of the superstructure;
if the item is visible:
say "[The item] breaks[if something is relying on the item], leaving [a list of things which are relying on the item] behind[end if].";
if something is relying on the item,
now all the things which are relying on the item are in the home;
now the item is nowhere.

Now suppose we'd like to add some further cases for what happens if the player breaks a fragile door this way:

To destroy (item - a door):
now the item is open;
now the item is unopenable;
say "[The item] smashes."

Rule for printing the name of an unopenable open door while not throwing something at something:
say "open doorway".

Understand "door" or "doorway" as a door.

This works, except that objects will continue to "strike" open, unopenable doors, with the result that the player can smash the same door over and over. What we need is another rule, after the aerodynamics rule and before the contact rule, that tells Inform how to handle throwing things at open doors.

This is the flying through doorways rule:
if the second noun is an open door:
let the distant room be the other side of the second noun;
move the noun to the distant room;

say "[The noun] flies out of sight into [the distant room].";
rule succeeds.

If the original rulebook is one we wrote ourselves, we could just add that rule in the proper spot in order. If we got it from an extension, though, we might need to put it in the right place explicitly:

The flying through doorways rule is listed before the contact rule in the carry out throwing it at rules.

The magic of rulebooks is that they allow authors to amend each other's work (or the Standard Rules) with a fair amount of freedom. A well-written extension will give individual names to its rules, to allow subsequent authors to modify the function of the extension without too much trouble.

Now for an actual scenario with which to test this:

Part 2 - The Study

The sliding paper screen is a door. It is north of the Moss Garden and south of the Study. The paper screen is fragile.

The player carries a netsuke and a shamisen. The description of the netsuke is "A weight for the cord on which you wear your purse or your medicine box. This particular one has the shape of a bullfrog, carved from green stone." The netsuke is round, hard, and strong. Understand "green" or "stone" or "bullfrog" as the netsuke.

The description of the shamisen is "An instrument you have only begun to learn to play." The shamisen is linear, soft, and fragile. A neck is part of the shamisen. The neck is linear, strong, and hard. A body is part of the shamisen. The body is round, fragile, and soft. A string is part of the shamisen. The string is linear, soft, and strong. The printed name of the body is "[if the body is not part of the shamisen]shamisen [end if]body". The printed name of the neck is "[if the neck is not part of the shamisen]shamisen [end if]neck". Understand "shamisen" as the body when the body is not part of the shamisen. Understand "shamisen" as the neck when the neck is not part of the shamisen.

The description of the Study is "A restful three-tatami room." The Study contains a calligraphy box and a hanging scroll. The initial appearance of the hanging scroll is "A handsome scroll depicts two women in kimonos crossing a bridge; Mount Fuji is in the background." The calligraphy box contains a brush. The box is openable and closed. The brush is hard, linear, and strong. The calligraphy box is round, soft, and strong. The hanging scroll is flat, soft, and strong.

The description of the Moss Garden is "Earlier today, you arranged three leaves on the moss in imitation of autumn. They must not be disturbed." The leaves are scenery in the Moss Garden. Instead of throwing something at the leaves: say "You spent too long over their placement."

Test me with "test one / test two".

Test one with "open screen / throw netsuke at screen / n / get netsuke / close screen / get scroll / throw scroll at screen / throw netsuke at scroll / get netsuke /

throw netsuke at shamisen / drop netsuke".

Test two with "throw shamisen at netsuke / get all / throw netsuke at screen / get netsuke / throw netsuke at door / s / get netsuke".

232

★ **Example Swerve left? Swerve right? Or think about it and die?**

WI

Building a marble chute track in which a dropped marble will automatically roll downhill.

Suppose we have marbles that roll downhill across our map, in a life-size version of one of those marble-chute toys. We might now want to keep track of both compass relationships and which-room-slopes-into-which, so we make a new relation:

"Swerve left? Swerve right? Or think about it and die?"

Overlooking relates various rooms to various rooms.

The verb to overlook means the overlooking relation.

A thing can be spherical or lumpy. A marble is a kind of thing. A marble is always spherical. The player carries a marble called a red marble. The player carries a marble called an agate marble. The player carries a marble called a blue cloudy marble.

The Long Yellow Slide is north of the Funnel. The Long Yellow Slide overlooks the Blue Funnel. The Ski-jump is below the Blue Funnel. The Blue Funnel overlooks the Ski-jump. The Ski-jump overlooks the Landing Bowl. The Landing Bowl overlooks the Snake Run. The Landing Bowl is north of the Snake Run. The Snake Run overlooks the Goal. The Snake Run is north of the Goal.

Definition: a room is sloping if it overlooks a room.

And let's say we want the player to be allowed to slide, too, since that would be much more fun than just watching the marbles go:

Understand "sit" as sitting down. Sitting down is an action applying to nothing. Check sitting down: if the player is spherical, say "You are already seated." Carry out sitting down: now the player is spherical. Report sitting down: say "You sit, ready to slide wherever fate takes you."

Understand the command "stand" as something new.

Understand "stand" or "stand up" as standing up. Standing up is an action applying to nothing. Check standing up: if the player is lumpy, say "You are already standing." Carry out standing up: now the player is lumpy. Report standing up: say "You get to your feet."

Now a rule to control what happens to all our sliding and rolling objects:

Every turn:
 repeat with item running through spherical things which are in sloping rooms:
 let the current space be the holder of the item;
 let the final space be a random room which is overlooked by the current space;
 if the player can see the item and the item is a marble, say "[The item] rolls out of the room toward [the final space].[line break]";
 if the player is the item, say "You keep sliding...";
 move the item to the final space;
 if the player can see the item and the item is a marble, say "[The item] rolls into the room from [the current space].[line break]".

Since the Ski-jump overlooks the Landing Bowl, the marble will be able to fly through the air to its destination, even though there is no map connection to allow the player to cross. We might want to let the player make it across this barrier also, so:

Instead of jumping in a sloping room:
 say "You leap...";
 move the player to a random room overlooked by the location.

Because overlooking is various-to-various, we could include that element popular in marble chute toys, the splitter:

The Downhill Splitter is north of the Long Yellow Slide. "The green plastic chute runs downhill towards a Y-junction, forcing incoming marbles right or left."

The Downhill Splitter overlooks the Long Yellow Slide and the Purple Snaking Passage. The Purple Snaking Passage is southeast of the Downhill Splitter. The Purple Snaking Passage overlooks the Landing Bowl. The Purple Snaking Passage is above the Landing Bowl.

The player is in the Downhill Splitter.

Test me with "drop red / drop blue / sit / z / stand up / drop agate / sit / z / z / z / z / z".

Receptacles that calculate internal volume and the amount of room available, and cannot be overfilled.

In the following, we pretend that every item has a cuboidal shape. Every thing has a length, width and depth, while a "measured container" also has interior dimensions. (Thus a 10x10x10 container with 1cm-thick sides might have interior dimensions 9x9x9.)

"Depth"

A length is a kind of value. 10 cm specifies a length. An area is a kind of value. 10 sq cm specifies an area. A length times a length specifies an area. A volume

is a kind of value. 10 cu cm specifies a volume. A length times an area specifies a volume.

A thing has a length called height. A thing has a length called width. A thing has a length called depth. The height of a thing is usually 10 cm. The width of a thing is usually 10 cm. The depth of a thing is usually 10 cm.

To decide what volume is the exterior volume of (item - a thing):
let base area be the height of the item multiplied by the width of the item;
let base volume be the base area multiplied by the depth of the item;
decide on the base volume.

In order to see how these shapes might fit together spatially, we need to work out the three dimensions in order of size. (If we were only dealing with portable objects, we could simply insist that the length always be greater than the width which in turn must be greater than the depth, because we could always turn them over in our hands until this was so: but some of the things we deal with may be fixed in place.) A clever way to do this might be to put them in a table of three rows and sort it, but we will write the calculation out longhand:

To decide what length is the largest dimension of (item - a thing):
let long side be the height of item;
if the width of the item is greater than the long side, now the long side is the width of the item;
if the depth of the item is greater than the long side, now the long side is the depth of the item;
decide on the long side.

To decide what length is the middling dimension of (item - a thing):
let longer side be the height of item;
let shorter side be the width of item;
if the width of the item is greater than the height of the item:
let shorter side be the height of item;
let longer side be the width of item;
if the depth of the item is greater than the longer side, decide on the longer side;
if the depth of the item is less than the shorter side, decide on the shorter side;
decide on the depth of the item.

To decide what length is the shortest dimension of (item - a thing):
let short side be the height of item;
if the width of the item is less than the short side, now the short side is the width of the item;
if the depth of the item is less than the short side, now the short side is the depth of the item;
decide on the short side.

When testing this example, the author made use of the following: it's no longer needed, but may be useful to anyone else planning elaborations.

To test the dimensions of (item - a thing):
say "[the item] - height [height of the item], width [width of the item], depth [depth of the item].";

say "largest side [largest dimension of the item], middling [middling dimension of the item], smallest [shortest dimension of the item]."

We now introduce a new kind: a measured container, which not only has exterior dimensions - the height, width and depth which every thing now has - but also interior measurements. A convenient way to do calculations with the hollow interior is to regard it as if it were a solid shape in its own right, and we do this with the aid of something out of world, which the player never sees: the "imaginary cuboid", which is made into the shape of whatever measured container's interior is being thought about.

A measured container is a kind of container. A measured container has a length called interior height. A measured container has a length called interior width. A measured container has a length called interior depth.

There is an imaginary cuboid.

To imagine the interior of (receptacle - a measured container) as a cuboid:
now the height of the imaginary cuboid is the interior height of the receptacle;
now the width of the imaginary cuboid is the interior width of the receptacle;
now the depth of the imaginary cuboid is the interior depth of the receptacle.

To decide what volume is the interior volume of (receptacle - a measured container):
imagine the interior of the receptacle as a cuboid;
decide on the exterior volume of the imaginary cuboid.

If we assume that we could always pack items into a measured container with perfect ease, never wasting any space, then the only volume constraint will be that the total volume of the contents must not exceed the volume of the inside of the container. So we need to calculate the available volume.

To decide what volume is the available volume of (receptacle - a measured container):
let the remaining space be the interior volume of the receptacle;
repeat with item running through things in the receptacle:
decrease the remaining space by the exterior volume of the item;
if the remaining space is less than 0 cu cm, decide on 0 cu cm;
decide on the remaining space.

If we only constrained volume, a 140 cm-long fishing rod could fit into a 12 cm by 12 cm compact disc box. So we also insist the basic shape must fit, in some orientation perpendicular to one of the sides (i.e.: we can turn the item over in any of its three sides, but not turn it diagonally or wedge it in at a tilt). This requires the longest side of the item to be less than the longest side of the receptacle, and the middle-length side, and also the shortest side. The number of these conditions to fail gives us a clue as to how we can best describe the reason why the shape won't squeeze in.

Check inserting something (called the item) into a measured container (called the receptacle):
if the exterior volume of the item is greater than the interior volume of the receptacle, say "[The item] will never fit inside [the receptacle]." instead;

if the exterior volume of the item is greater than the available volume of the receptacle, say "[The item] will not fit into [the receptacle] with [the list of things in the receptacle]." instead;
imagine the interior of the receptacle as a cuboid;
if the largest dimension of the item is greater than the largest dimension of the imaginary cuboid, say "[The item] is too long to fit into [the receptacle]." instead;
if the middling dimension of the item is greater than the middling dimension of the imaginary cuboid, say "[The item] is too wide to fit into [the receptacle]." instead;
if the shortest dimension of the item is greater than the shortest dimension of the imaginary cuboid, say "[The item] is too bulky to fit into [the receptacle]." instead.

And finally a situation to try out these rules.

The Cubist Lab is a room. "A laboratory which, as the art critic Louis Vauxcelles said about Braque's paintings in 1908, is full of little cubes: everyday objects rendered as if cuboidal."

The box is a measured container. The interior height is 10 cm. The interior depth is 5 cm. The interior width is 6 cm. The player carries the box.

A pebble is a kind of thing. The height is usually 2 cm. The depth is usually 2 cm. The width is usually 2 cm. The player carries 25 pebbles.

A red rubber ball is carried by the player. The depth is 5 cm. The width is 5 cm. The height is 5 cm.

An arrow is carried by the player. The height is 40 cm. The width is 1 cm. The depth is 1 cm.

A crusty baguette is carried by the player. The height is 80 cm. The width is 4 cm. The depth is 5 cm.

A child's book is carried by the player. The height is 1 cm. The width is 9 cm. The depth is 9 cm.

A featureless white cube is carried by the player. The height is 6 cm. The width is 6 cm. The depth is 6 cm.

Test me with "put arrow in box / put book in box / put cube in box / put ball in box / put baguette in box / put pebbles in box".

Several warnings about this. First, the numbers can't go very high (if the Settings for the project set the story file format to the Z-machine): while the volume can in theory go to 32,767, in practice this equates to an object 32 cm on a side, which is not very large. One way to avoid this is to use the Glulx format, allowing for sizes in excess of 10 m on a side: or we could simply scale the dimensions to suit our purposes, using a decimeter (10 cm) as the basic unit of measurement, for instance.

Second, the system will require a height, width, and depth for every portable object in the game, which is a large commitment to data entry; it may become tiresome. So it is probably not worth bothering with this kind of simulation unless it is going to be genuinely significant.



This example draws together the previous snippets into a working implementation of the weighbridge.

The following is not a very sophisticated approach, because it does not allow for weight to accumulate: if we put a gold ingot into a paper bag, then put the bag on the balance platform, only the bag's weight will register. But it will do for a first try.

"Dimensions"

A length is a kind of value. 10m specifies a length. An area is a kind of value. 10 sq m specifies an area. A length times a length specifies an area.

A weight is a kind of value. 10kg specifies a weight. Everything has a weight.

The verb to weigh means the weight property. A thing usually weighs 1kg.

Definition: A thing is light if its weight is 3kg or less.

Definition: A thing is heavy if its weight is 10kg or more.

The Weighbridge is a room.

A blackboard is in the Weighbridge. "A blackboard propped against one wall reads: '122/10 is [122 divided by 10] remainder [remainder after dividing 122 by 10]; 122kg/10kg is [122kg divided by 10kg] remainder [remainder after dividing 122kg by 10kg]; 122kg/10 is [122kg divided by 10] remainder [remainder after dividing 122kg by 10].'" The blackboard weighs 10kg.

A feather and a lead pig are in the Weighbridge. The lead pig weighs 45kg.

The balance platform is a supporter in the Weighbridge. "The balance platform is 10m by 8m, giving it an area of [10m multiplied by 8m], currently weighing [the list of things on the platform]. The scale alongside reads: [total weight of things on the platform]. [if two things are on the platform]Average weight is: [the total weight of things on the platform divided by the number of things on the platform]. Heaviest item is [the heaviest thing on the platform], at [weight of the heaviest thing on the platform]. Lightest item is [the lightest thing on the platform], at [weight of the lightest thing on the platform].[otherwise]It seems to be able to weigh several things at once."

Test me with "get feather / put it on platform / look / get pig / put it on platform / look".



Suppose that we have a number of objects in the game that are sized in some conventional unit (such as meters), but which we would like to describe in slightly less formal terms. To do this, we will start with measurements as defined in the built-in extension Metric Units, so we don't have to recreate all these.

We'll add our own set of "conceptual units" -- things we're familiar with in real life. As we'll see below, Inform will automatically choose a unit of the right order to express a given distance if we tell it to print a length "in conceptual units".

Note: the following will compile only if you have settings set for Glulx. (To change this, go to the Settings panel and click on the Glulx option.) The Glulx virtual machine is capable of handling larger numbers than the Z-machine.

"The Speed of Thought"

Section 1 - Procedure

Include Metric Units by Graham Nelson.

1 quarter (in conceptual units, in quarters, singular) or 2 quarters (in conceptual units, in quarters, plural) specifies a length equivalent to 24mm.

1 pencil (in conceptual units, in pencils, singular) or 2 pencils (in conceptual units, in pencils, plural) specifies a length equivalent to 18cm.

1 bathtub (in conceptual units, in bathtubs, singular) or 2 bathtubs (in conceptual units, in bathtubs, plural) specifies a length equivalent to 152cm.

1 Olympic swimming pool (in conceptual units, in Olympic swimming pools, singular) or 2 Olympic swimming pools (in conceptual units, in Olympic swimming pools, plural) specifies a length equivalent to 50 meters.

1 Empire state building (in conceptual units, in Empire State buildings, singular) or 2 Empire State buildings (in conceptual units, in Empire State buildings, plural) specifies a length equivalent to 443m.

1 credit card (in conceptual units, in credit cards, singular) or 2 credit cards (in conceptual units, in credit cards, plural) specifies an area equivalent to 46 sq cm.

1 letter sheet (in conceptual units, in letter sheets, singular) or 2 letter sheets (in conceptual units, in letter sheets, plural) specifies an area equivalent to 603 sq cm.

1 queen-sized mattress (in conceptual units, in queen-sized mattresses, singular) or 2 queen-sized mattresses (in conceptual units, in queen-sized mattresses, plural) specifies an area equivalent to 3 square meters.

1 football field (in conceptual units, in football fields, singular) or 2 football fields (in conceptual units, in football fields, plural) specifies an area equivalent to 5351 square meters.

Understand "report [something]" as reporting. Reporting is an action applying to one thing.

Check reporting:

if the noun is not a fact:

say "The public doesn't want to hear about [the noun]." instead.

Carry out reporting:

now the noun is nowhere.

Report reporting:

if the extent of the noun is greater than 0mm and the surface of the noun is greater than 0 sq cm:

contextualize "[The noun] has a length of [about] [extent of the noun in conceptual units] and an area of [about] [surface of the noun in conceptual units].";

otherwise if the extent of the noun is greater than 0mm:

contextualize "[The noun] has a length of [about] [extent of the noun in conceptual units].";

otherwise if the surface of the noun is greater than 0 sq cm:

contextualize "[The noun] has an area of [about] [surface of the noun in conceptual units].";

otherwise:

say "[The noun] is... pretty hard to imagine,' you say weakly. That's not going to go over well."

To say about:

say "[one of]roughly[or]about[or]around[or]approximately[at random]";

To contextualize (chosen information - text):

say "[one of]You turn to the camera and speak:[or][or]Turning to another camera angle, you add:[or][stopping] ";

say "[chosen information] ";

say "[one of][line break][or]Right now the station will be cutting over to a visual of that.[or][line break][or]Pity the kids in audiovisual who have to scare that image together in a hurry.[or]You smile brightly.[stopping]";

Section 2 - Scenario

The Science Journalism Desk is a room. "From here you, the Science Anchor, have the privilege of reporting the latest and most fascinating stories to an eager public."

After looking:

try thinking.

Instead of thinking:

say "Currently you have to report on the International Space Station. Your story could include [the list of facts carried by the player]."

Instead of taking inventory:

say "It looks foolish to be fiddling with your possessions on camera."

Instead of dropping a fact:

say "You decide to omit [the noun] from your lineup."; now the noun is nowhere.

A fact is a kind of thing. Every fact is carried by the player. A fact has a length called the extent. A fact has an area called the surface.

The experiment module is a fact. The extent is 1116cm.

The logistics module is a fact. The extent is 421cm.

The solar array is a fact. The surface is 375 sq m. The extent is 58m.

An individual solar cell is a fact. The surface is 8 sq cm.

The orbit height is a fact.

Report reporting the orbit height:

contextualize ""The station orbits at heights between [about] [278km in conceptual units] and [460km in conceptual units] above the earth."" instead.

Every turn:

if the player carries no facts:

say "And that's all! The channel cuts to weather.";

end the story saying "Time for lunch".

Test me with "report experiment module / report logistics / report height / report array / report solar cell".

259



Example Lead Cuts Paper

WI

To give every container a breaking strain, that is, a maximum weight of contents which it can bear - so that to put the lead pig into a paper bag invites disaster.

The following shows the kind of "realism" rules which could be introduced using weights. Not entirely realistic: we do not bother to rupture containers out of the player's sight.

"Lead Cuts Paper"

A weight is a kind of value. 10kg specifies a weight. Everything has a weight. A thing usually has weight 1kg.

A container has a weight called breaking strain. The breaking strain of a container is usually 50kg. Definition: A container is bursting if the total weight of things in it is greater than its breaking strain.

A lead pig, a feather, a silver coin and a paper bag are in a room called the Metallurgy Workshop. The paper bag is a container with breaking strain 2kg. The lead pig has weight 50kg.

Every turn when a container (called the sack) held by someone visible (called the chump) is bursting:

say "[The sack] splits and breaks under the weight! [if the player is the chump]You discard[otherwise][The chump] discards[end if] its ruined remains, looking miserably down at [the list of things in the sack] on the floor.";

now all of the things in the sack are in the location;

now the sack is nowhere.

Test me with "get bag / get feather / put feather in bag / get pig / put pig in bag / look".

228



Example Otranto

WI

A kind of rope which can be tied to objects and used to anchor the player or drag items from room to room.

The range of things one might want to do with a rope in a work of interactive fiction is fairly overwhelming. One might, in theory, swing from ropes; use them to tie containers shut; cut them up into smaller ropes; tie them together into longer ropes; employ them as fuses; bind other characters with them, or the player character.

Our rope implementation is, by these lights, reasonably simple, but it does account for the possibility of tying and untying both ends; using ropes to descend into lower rooms; pulling objects tied to the far end of the rope; and dragging objects from place to place.

"Otranto"

We start by coming up with a rope.

A rope is a kind of thing.

Definition: a thing is nonrope if it is not a rope. [The perfect idiocy of this statement notwithstanding, having a shortcut will come in very handy later]

Attachment relates things to each other in groups. The verb to be stuck to means the attachment relation.

Definition: a thing is tied if the number of things stuck to it is greater than 1.

Definition: a thing is free if it is not tied.

Definition: a rope is free if the number of nonrope things stuck to it is less than 2.

Definition: a thing is hindering if it is stuck to the noun and it is not within the location.

A thing can be round or unevenly shaped. A thing is usually round.

Definition: something is anchored if it is fixed in place or it is scenery or it is part of an anchored thing.

Definition: something is draggable if it is not had by the player and it is not the player and it is not anchored.

Now, we want a rope to be described in terms of the way it is tied, when it's described in a room description.

Rule for writing a paragraph about a rope (called the coil):
if the coil is stuck to something which is in a room (called the next room) which is not the location:
let the way be the best route from the location to the next room;
if the way is up or the way is down:
say "[The coil] runs [way] into [the next room].";
otherwise:
say "[The coil] snakes across the floor [way] towards [the next room].";
otherwise:

say "There is [a coil] here[if the coil is stuck to a visible nonrope thing], tied to [the list of nonrope visible things which are stuck to the coil][end if]."

To decide what room is the home of (item - a thing):

if item is a door:
let front cut be the number of moves from the location to the front side of the item;
let back cut be the number of moves from the location to the back side of the item;
if front cut is -1, let front cut be 999;
if back cut is -1, let back cut be 999;
if the location encloses the item, decide on the location;
if front cut is greater than back cut, decide on the back side of the item;
decide on the front side of the item;
decide on the location of the item.

Rule for writing a paragraph about a nonrope thing (called the anchor) which is stuck to a rope (called the coil):

if the coil is in an adjacent room:
let the next room be the home of the coil;
let the way be the best route from the location to the next room;
if the way is up or the way is down:
say "[The coil] runs [way] from [the anchor] into [the next room].";
otherwise:
say "From [the anchor] runs [a coil], heading off toward [the way].";
otherwise:
if the coil is stuck to something which is not visible,
say "[The coil] is tied to [the anchor][if the coil is stuck to something in an adjacent room (called the next room)], and from there runs off towards [the next room][end if]."

We need a way to account for it when it's being carried, as well.

After printing the name of a rope (called the tied object) while taking inventory:

if something nonrope is stuck to the tied object:
say " (attached to [the list of nonrope things which are stuck to the tied object])";
otherwise:
say " (with both ends free)".

And, indeed, whenever the player examines a rope, we should see what's connected.

Instead of examining a rope (called the cord) when something is stuck to the cord:

say "[The noun] is tied to [the list of secondary things which are stuck to the noun]."

Similarly, any time the player looks at something tied to a rope.

After examining the player when the player is stuck to something which is not the player:

say "You're currently lashed to [the list of secondary things stuck to the noun]."

After examining something which is stuck to something secondary:

say "[The noun] is currently attached to [the list of secondary things stuck to

the noun]."

We also need to make sure that the rope can be interacted with properly even when it's partly in the next room.

After deciding the scope of the player:

if something stuck to a rope (called the coil) is in the location, place the coil in scope.

A reaching inside rule:

if the noun is a rope:

let the anchor be a random visible thing stuck to the noun;

if the anchor is touchable, allow access.

Now tying:

Before tying something to a rope:

if the noun is stuck to the second noun, say "[The noun] and [the second noun] are already tied together." instead;

if the second noun is not free, say "[The second noun] has no ends free." instead;

if the noun is round, say "You can't realistically tie anything to [the noun]." instead.

Instead of tying a rope to something:

try tying the second noun to the noun.

Instead of tying something to a rope:

now the noun is stuck to the second noun;

say "You loop [the second noun] around [the noun] and knot firmly."

Instead of tying something to a nonrope tied thing:

let the coil be a random rope stuck to the second noun;

try tying the noun to the coil.

Instead of tying a nonrope tied thing to something:

let the coil be a random rope stuck to the noun;

try tying the second noun to the coil.

Instead of tying a free nonrope thing to a free nonrope thing:

if the player carries a free rope (called the coil):

try tying the noun to the coil;

if the noun is stuck to the coil and the coil is free:

try tying the second noun to the coil;

otherwise:

say "You lack the requisite spare rope."

Understand "untie [something] from [something]" as untying it from. Understand "untie [something]" as untying it from.

Rule for supplying a missing second noun while untying something from:

if the number of secondary things stuck to the noun is 0, say "[The noun] is already entirely free." instead;

if the noun is a rope:

```

    if the number of touchable nonrope things which are stuck to the noun > 1:
        say "You'll have to say which thing you want to untie [the noun] from.";
        rule fails;
    otherwise:
        if the number of touchable nonrope things stuck to the noun is 0, say
        "You can't reach [the random nonrope thing stuck to the noun]." instead;
        let the tied object be a random touchable nonrope thing which is stuck to
        the noun;
        say "(from [the tied object])[line break]";
        now the second noun is the tied object;
    otherwise:
        if the noun is stuck to a rope (called the tied object):
            say "(from [the tied object])[line break]";
            now the second noun is the tied object.

```

Untying it from is an action applying to two things.

Before untying a rope from something: try untying the second noun from the noun instead.

```

Before untying something from a rope:
    if the second noun is not held:
        say "(first picking up [the second noun])[line break]";
        try taking the second noun.

```

```

Check untying it from:
    unless the noun is stuck to the second noun or the second noun is stuck to
    the noun,
        say "[The noun] and [the second noun] are already not tied together."
    instead.

```

```

Carry out untying it from:
    now the noun is not stuck to the second noun.

```

```

Report untying it from:
    say "Untied."

```

Another part of the fun of a rope is that you can drag things from another room:

After reading a command: now every thing is unmentioned.

Before pulling something anchored: say "[The noun] is firmly anchored." instead.

```

Instead of pulling something tied:
    if the noun is unmentioned:
        say "The impulse is transmitted to [the list of pullable things stuck to the
        noun].";
        repeat with item running through pullable things stuck to the noun:
            say "[item]: [run paragraph on]";
            try pulling the item;
        if the noun is a rope and the noun is not within the location:
            if the number of nonrope hindering things is 0, move the noun to the
            location;
        otherwise:
            continue the action.

```

Before pulling something which is not visible:
 if the noun is anchored:
 say "[The noun] resists, for whatever reason." instead;
 otherwise:
 let space be the holder of the noun;
 let way be the best route from the space to the location;
 if the way is a direction:
 move the noun to the location;
 say "[The noun] [if the way is up]rises[otherwise]slides[end if] into view."
 instead;
 otherwise:
 move the noun to the location;
 say "[The noun] slides into view." instead.

Definition: a thing is secondary if it is not the noun. Definition: a thing is pullable if it is not the noun and it is not the player.

A player who is tied to things should also have some restrictions on his ability to move.

Before going a direction (called the way) when the player has something (called the link) which is stuck to something anchored (called the anchor):
 let the next room be the home of the anchor;
 if the next room is not a room, continue the action;
 if the next room is the location:
 if the link is stuck to at least two anchored things,
 say "You can't go far while you're carrying [the link] tied to [the list of anchored things stuck to the link]." instead;
 otherwise:
 let the safe way be the best route from the location to the next room;
 if the safe way is the way:
 if the player is not stuck to the anchor, say "(coiling up your rope again as you go...)";
 otherwise:
 if the safe way is a direction,
 say "While you have [the link] you can't really head any direction but [best route from the location to the next room]." instead;
 otherwise say "You're tied up here." instead.

Before going a direction (called the way) when the player is stuck to something anchored (called the anchor):
 let the next room be the home of the anchor;
 if the next room is not a room, continue the action;
 if the next room is the location:
 if the player is stuck to at least two anchored things,
 say "You can't go far while you're tied to [the list of anchored things stuck to the player]." instead;
 otherwise:
 if the best route from the location to the next room is the way:
 say "(coiling up your rope again as you go...)";
 otherwise:
 say "Your attachments prevent you going any way but [best route from the location to the next room]." instead.

Sometimes, if the player is tied to a movable object, the moved object will move with him.

After going somewhere when the player has something (called the link) which is stuck to something draggable:

if the player is not stuck to the link:
say "You drag along behind you [the list of draggable things which are stuck to the link].";
now every draggable thing which is stuck to the link is in the location;
continue the action.

Report going somewhere when the player is stuck to something draggable:
say "You drag along behind you [the list of draggable things which are stuck to the player].";
now every draggable thing which is stuck to the player is in the location.

And now the actual game and puzzles.

Use full-length room descriptions.

The Fallow Field is a room. "The very land is gloomy, the earth plowed into untended rows that yield no fruit, shadowed by the castle to the north. A chasm, no doubt the product of some upheaval of the earth, opens before your feet.". An oak stump is fixed in place in the Field. "From an oak stump, a few hopeful shoots grow." A hempen rope is a rope in the field. It is stuck to the oak stump and the wooden chest. The stump is unevenly shaped.

The Chasm is below the Field. "Your person is most uncomfortably pressed on every side by the closeness of the walls; to which you may add as a further inconvenience, that the irregularity of the floor making it difficult to walk upright." An iron key is in the Chasm. "An iron key nestles in the cleft of earth, its age indicated by its implausibly great size."

The wooden chest is a unevenly shaped closed openable container in the Chasm. The description of the wooden chest is "A handsome, solid case not long committed to its dank enclosure, or it would long since have rotted." Rule for printing the name of the wooden chest when the chest is not handled: say "deadweight". Understand "dead" or "weight" or "deadweight" as the chest. Before pulling the wooden chest: now the chest is handled.

In the chest is a heavy dagger. The description of the dagger is "Set with red jewels and of a wicked aspect."

Before going down from the Field when the player is not stuck to something anchored:
say "You don't quite dare simply leap into the darkness without some anchor." instead.

Before going down from the Field:
let anchor be a random anchored thing which is stuck to the player;
say "You lower yourself gingerly, hoping that [the anchor] holds your weight..."

Before going up from the Chasm:
if the player cannot touch a rope which is stuck to an anchored thing which is in the Field, say "And how, precisely, do you mean to do that?" instead.

The Castle Hall is north of the Field. "All is desolate: the great hall has no roof, nor is there any glass in the windows. A staircase without banister ascends

inside the wall to a musician's gallery without song."

The Musician's Gallery is above the Castle Hall. "Of its former cheery aspect only this remains to the Gallery: that chevrons of red and yellow are painted on the wall. But as these are streaked with rain and grime, the banister pulled away, the roof open to the sky, and the corners made a nesting place for birds, the consolation thereby afforded is but slight."

The pointed door is north of the Musician's Gallery and south of the Sinister Attic. It is a closed locked openable door. "A pointed door of particularly grim and uninviting aspect leads north." The pointed door is lockable and unevenly shaped. The description of the pointed door is "A door coming to a gothic point and fitted with iron fittings of great strength. The handle looks particularly well-attached." The iron key unlocks the pointed door.

A rule for reaching inside the Musician's Gallery:
allow access.

A rule for reaching inside the Sinister Attic:
allow access.

Instead of opening the pointed door for the first time:
say "When you rattle at the door, there arises from beyond a terrible shrill noise as though something beyond exults in its imminent release."

After opening the trapped pointed door when the player can see the pointed door:
say "Thousands of bats fly from the pointed door, attacking you!";
end the story.

After opening the pointed door when the player cannot see the pointed door:
now the pointed door is untrapped; continue the action.

The pointed door can be trapped or untrapped. The pointed door is trapped.

Before pulling the pointed door: try opening the pointed door instead.

The player is unevenly shaped.

After going to the sinister attic:
say "You have arrived at the goal of your quest!";
end the story finally.

Test me with "x rope / pull rope / get chest / untie rope from chest / tie rope to me / down / get key / up / untie rope from stump / north / up / unlock pointed door with key / open it / tie rope to door / down / pull rope / up / north".

Test death with "x rope / pull rope / get chest / untie rope from chest / tie rope to me / down / get key / up / untie rope from stump / north / up / unlock pointed door with key / open it / g"



A string which can be cut into arbitrary lengths, and then tied back together.

"Snip!"

Length is a kind of value. 30 inch specifies a length. 20 in specifies a length. 50 inches specifies a length.

A string is a kind of thing. A string has a length. The length of a string is usually 36 inches.

Before printing the name of a string, say "[length] piece of ". Rule for printing the plural name of a string: say "[length] pieces of string".

Understand the command "cut" as something new. Understand "cut [length] from/off [something]" as trimming it by (with nouns reversed). Understand "cut [something] by [length]" as trimming it by. Understand the command "trim" as "cut".

Trimming it by is an action applying to one thing and one length.

Check trimming it by:

if the length understood is 0 inches, say "You're approaching Zeno's string at this point." instead;

if the length understood is greater than the length of the noun, say "[The noun] is only [length of the noun] long to start with." instead;

if the length understood is the length of the noun, say "[The noun] is already exactly [length of the noun] long." instead.

Carry out trimming it by:

now the length of the noun is the length of the noun minus the length understood;

let the other half be a random string in the string repository;

now the length of the other half is the length understood;

move the other half to the player.

Report trimming it by:

reset string lengths; [we will define this in a moment; it helps guarantee that our descriptions of the strings are correct when we write the output list]

say "You now have [a list of strings carried by the player]."

Understand "cut [something] in half" as halving. Halving is an action applying to one thing.

Carry out halving:

let half measure be the length of the noun divided by 2;

now the length understood is half measure;

try trimming the noun by half measure.

This fudges slightly, since an odd-length string will be divided into uneven halves. Keeping track of fractional inches would complicate matters, though, so let's assume for now that this doesn't matter.

The player carries a string.

The Scissors Room is a room.

The string repository contains 35 strings.

Since our initial string is 36 inches long and it is impossible for the player to divide it into pieces smaller than an inch each, we need a total of 36 items to represent all the string-bits: one that the player carries at the outset, and 35 others. We should bear in mind that it is usually a good idea to use the smallest number of spare objects we can get away with: writing a game that required 1000 strings in the string repository would place silly demands on the resources of the system, so it's best to avoid that sort of thing if possible.

Now with a bit of fiddling we can also teach Inform to recognize descriptors such as "the shortest string":

Ordinariness is a kind of value. The ordinarinesses are longest, medium, shortest. A string has an ordinariness. Understand the ordinariness property as referring to a string.

Definition: a string is small if its length is 2 in or less. Definition: a string is large if its length is 20 in or more.

Before reading a command:
reset string lengths.

To reset string lengths:
let upper measure be the length of the largest visible string;
let lower measure be the length of the smallest visible string;
repeat with item running through strings:
now the ordinariness of the item is medium;
if the length of the item is the upper measure, now the item is longest;
if the length of the item is the lower measure, now the item is shortest.

After reading a command:
if the player's command includes "shorter", replace the matched text with "shortest";
if the player's command includes "longer", replace the matched text with "longest".

Instead of tying a string to a string:
move the second noun to the string repository;
now the length of the noun is the length of the noun plus the length of the second noun;
decrease the length of the noun by 1 inch;
say "You end up with [a noun], as some is taken up by the knot."

This is still a little incomplete because we cannot refer to strings by their lengths, as in "the 2 inch string" and so on. To do this, we borrow a line from the chapter on Understanding:

Understand the length property as referring to a string.

Test me with "trim string by 4 in / cut longer string in half / cut longest string in half / cut shortest string in half / g / g / tie longest string to shortest string / tie longest string to medium string / i / x 16 inch string / drop 8 inch string / i".

346

Example Rules of Attraction

WI

A magnet which picks up nearby metal objects, and describes itself appropriately in room descriptions and inventory listings, but otherwise goes by its ordinary name.

Often we have some salient features of an object that we want to make sure the player notices whenever looking at the item in a room or in inventory. At other times, we may prefer to allow the name of the item to be printed bare. So for instance:

"Rules of Attraction"

A metal form is a kind of thing. A magnet is a kind of metal form.

Every turn:

repeat with item running through nonmagnetic metal forms which are not part of something:

if item is in a container which contains a magnet (called attractor):

say "[The item] sticks to [the attractor].";

now the item is part of the attractor.

The horseshoe magnet is a magnet carried by the player. The nail is a metal form carried by the player. The Barn is a room. In the Barn is a bucket. In the bucket is a metal form called the iron hook.

Definition: a thing is nonmagnetic if it is not a magnet.

Rule for printing room description details of a magnet (called attractor): if something is part of the attractor, say " (stuck to which [is-are the list of things which are part of the attractor])".

After printing the name of a magnet (called attractor) while taking inventory:

if something is part of the attractor, say " (stuck to which [is-are the list of things which are part of the attractor])".

Before taking a touchable thing which is part of a magnet (called attractor):

move the noun to the holder of the attractor.

Test me with "i / put horseshoe in bucket / look / get horseshoe / i / drop horseshoe / i / look / get all / put all in bucket / i / x magnet / get nail / i".

408

Example Electrified

WI

Adding a rule before the basic accessibility rule that will prevent the player from touching electrified objects under the wrong circumstances.

Suppose we want to prevent the player from touching anything electrified -- not just as a response to TOUCH OBJECT, but at any time when the action would require contact with the object in question.

"Electrified"

A thing can be safe or electrified. A thing is usually safe.

The Open Field is a room. "At this end of the field is a wire fence separating farm country from the government testing grounds beyond." The wire fence is an electrified thing in Open Field. It is scenery. The description of the wire fence is "Built into the fence is [a list of things which are part of the fence]." The scary box is an electrified container. It is part of wire fence. In the scary box is an alluring prize.

The player carries a flashlight, a grappling hook, a very thick rubber glove, and a length of rope. The glove is wearable.

This is the electrocution-wisdom rule:

- if the player wears the very thick rubber glove, make no decision;
- if the action requires a touchable noun and the noun is electrified, say "You fear touching [the noun]." instead;
- if the action requires a touchable second noun and the second noun is electrified, say "You fear touching [the second noun]." instead.

The electrocution-wisdom rule is listed before the basic accessibility rule in the action-processing rules.

Before touching the scary box:

say "You can't help noticing a bright red sticker on the surface of the box."
[This rule will fire even if we are not wearing the glove, because Before rules occur before basic accessibility.]

Instead of opening the scary box:

say "The scary box seems to be super-glued shut." [This one won't, because Instead rules occur after basic accessibility.]

Test me with "touch fence / touch box / open box / wear glove / open box".

139



Example Witnessed 1

WI

A kind of battery which can be put into different devices, and which will lose power after extended use.

The following example makes fairly ample use of material that we haven't seen yet, but gives some idea of the flexibility of every turn rules. Suppose we want to have a number of electrical devices, all of which may be powered by a set of batteries. The batteries will all need to be discharged as they are used (regardless of what device they happen to be controlling at the moment). So:

"Witnessed"

A battery is a kind of thing. A battery has a number called charge. The charge of a battery is usually 15.

Every turn:

- repeat with hollow running through battery compartments:
 - if the hollow is part of a switched on device (called the machine):
 - if a battery (called cell) is in the hollow:
 - decrement the charge of the cell;
 - carry out the warning about failure activity with the machine;
 - if the cell is discharged, carry out the putting out activity with the machine;
 - otherwise:
 - carry out the putting out activity with the machine.

Warning about failure of something is an activity.

Rule for warning about failure of a device (called the machine):

- if a random battery compartment which is part of the machine contains a battery (called the power source):
 - if the charge of the power source is 2, say "[The machine] is obviously going to go out quite soon."

Putting out something is an activity.

Rule for putting out a device (called the machine):

- say "[The machine] loses power and switches off!";
- silently try switching off the machine.

A battery compartment is a kind of container. A battery compartment is usually closed and openable. One battery compartment is part of every device. Instead of inserting something which is not a battery into a battery compartment, say "Only batteries should go in a battery compartment."

And to get rid of annoying messages like "Which would you like to close, the flashlight or the flashlight's battery compartment?" when only the compartment is closable, we might add some understanding instructions:

Understand "turn on [device]" as switching on.

Understand "turn off [device]" as switching off.

Understand "open [openable closed thing]" as opening.

Understand "close [openable open thing]" as closing.

Understand "put [something] in [container]" as inserting it into.

Instead of opening a device, try opening a random battery compartment which is part of the noun. Instead of closing a device, try closing a random battery compartment which is part of the noun. Instead of inserting a battery into a device, try inserting the noun into a random battery compartment which is part of the second noun.

Instead of switching on an empty device:

say "Nothing happens, perhaps because there isn't a charged battery in [the noun]."

Instead of switching on a battery compartment which is part of a device (called the power user), try switching on the power user.

Definition: a device is empty:

if a random battery compartment which is part of it contains a battery (called the power source):

if the power source is discharged, yes;

no;

yes.

Definition: a battery is discharged if its charge < 1 .

A light source is a kind of device. Carry out switching on a light source: now the noun is lit. Carry out switching off a light source: now the noun is unlit.

The flashlight is a light source. A D battery is a battery carried by the player.

The cassette recorder is a device. Every turn: if the cassette recorder is switched on, say "The cassette recorder hisses faintly."

Rule for warning about failure of the cassette recorder:

if a random battery compartment which is part of the cassette recorder contains a battery (called the power source):

if the charge of the power source is 2, say "The hiss from [the cassette recorder] begins to warble ominously."

The player wears a backpack. The backpack is openable. In the backpack is the flashlight and the cassette recorder.

The description of the cassette recorder is "Useful both for recording your notes and for capturing any odd ghostly sounds you may hear."

The description of the backpack is "An old familiar pack, which you know so well that you can find all its pockets and take things in and out of it in pitch darkness. To avoid it showing up oddly in photographs, it is entirely black, with no shiny or metallic attachments."

The description of the flashlight is "You bought a new one just for this occasion, because you were worried about bringing something too small or light. This is a heavy-duty flashlight with an adjustable-focus beam. The case is made of metal, rather than plastic, and there is a spare light-bulb inside as well. You've put a band of masking tape around the handle and written in your initials in red marker.

There is a piece of red cellophane attached to the business end of the flashlight to keep it from being overly bright."

The red cellophane is part of the flashlight.

Instead of doing something to the red cellophane: say "You need the cellophane on the flashlight so that using it does not completely destroy your night vision."

Thirtieth Street Station is a room. "A huge, high, rectangular room with coffered ceilings, which looks grand but mostly makes you feel lonely and small. There are long benches in rows down the middle of the room, and an information desk with the train times, and a series of ticket windows, none of which matters very much at the moment."

The benches are an enterable supporter. They are scenery in the Station. The information desk is scenery in the Station. Some ticket windows are scenery in the Station. Instead of examining scenery in the Station: say "You're fairly sure that whatever is going on here has nothing to do with [the noun]." Understand "window" as ticket windows.

The mural is fixed in place in Thirtieth Street. "At the north side of the station is a particularly pointless and empty annex to the main room. It is dominated by a huge relief of sorts, and this is what you remember." Understand "metal" or "relief" or "huge" as the mural. The description of the mural is "It is both stylized and confusing, but you think it might be supposed to represent the various tasks and occupations of Philadelphia's population. The portions closer to the ground look as though they have recently been subjected to a light dusting of talcum powder. No unusual prints are evident."

The wind chimes are fixed in place in Thirtieth Street. "Carefully attached to the wall with a piece of duct tape and a hook is a light-weight set of wind chimes. Someone else has been here before you, it seems." The description is "Several of your friends use wind chimes as a sort of ghost alarm, since ghosts sometimes cause very localized movements of air when there is no natural breeze."

And this last bit, borrowed from the chapter on Understanding, adds some special instructions to help Inform decide when the player is likely to be referring to a compartment and when he's likely to be referring to the device itself.

Does the player mean doing something other than searching to a battery compartment: it is unlikely. [We discourage Inform from choosing a compartment when the player uses just the name of a device or the word 'battery'.]

We also need to deal with commands like PUT BATTERY IN FLASHLIGHT, where Inform might construe BATTERY as the D battery, the flashlight's battery compartment, or the cassette recorder's battery compartment -- and might also construe FLASHLIGHT as either the flashlight's battery compartment or the flashlight itself.

Does the player mean inserting into a battery compartment:
if the noun is nothing:
it is very likely;
otherwise:
make no decision.

Does the player mean inserting a battery compartment into: it is very unlikely.

Does the player mean inserting something into a device: it is unlikely.

Does the player mean searching a battery compartment: it is very likely.

Test me with "test first / test second".

Test first with "i / open flashlight compartment / put battery in it / turn on flashlight / take d battery / open cassette compartment / turn on cassette / put battery in cassette compartment / turn on cassette / z / z / z / z".

Test second with "get d battery / put d battery in flashlight compartment / turn on flashlight / z / z / z / z / z / z / z / turn off flashlight / z / z / turn on flashlight / z".

118

★ Example Thirst 2

WI

A campfire added to the camp site, which can be lit using tinder.

"Thirst"

The player carries a waterskin. The waterskin can be full, partly drained, or empty. The waterskin is full. Understand "water" as the waterskin.

Instead of drinking the waterskin when the waterskin is empty:
say "There is no water left."

Instead of drinking the waterskin: if the waterskin is partly drained, now the waterskin is empty; if the waterskin is full, now the waterskin is partly drained; say "You drink a long draught."

After printing the name of the waterskin: say " ([waterskin condition])"

Campsite is a room. "It is solid night now, and the stars have come out. Unfamiliar stars. On the other side of the valley -- a valley round-bottomed but shallow, like a soup bowl -- burn other campfires, most likely bandits. Their voices do not carry, but the smoke rises and obscures the starlight over that way."

A sleepsack is an enterable container in the Campsite. "Your sleepsack is laid out in a pocket of sandy soil and coarse grass."

The sandy soil, the stars, the distant campfires, and the coarse grass are scenery in the Campsite. Understand "smoke" as the campfires. Instead of listening in the presence of your campfire: say "All you hear are the reassuring snaps and cracks of the sticks in your fire." Understand "campfires" or "fires" as the distant campfires.

Your campfire is scenery in the Campsite. Instead of pushing, pulling, turning, tasting, or touching your campfire, say "You would burn yourself." Understand "fire" as your campfire. The description of your campfire is "A reassuring protection against wild animals and cold."

The description of the stars is "You invent constellations for them. The slingshot. The scroll. The heart (upside down)."

Instead of going nowhere when the player is in Campsite:

say "Now is not the time for wandering, alone in the dark. Better to keep here[if your campfire is visible], by the fire[end if]."

Singing is an action applying to nothing. Understand "sing" as singing.

Instead of singing:

say "You sing, deep and low, a song from home. It is a good night for singing and the song raises your spirits."

The player carries a tinderbox. The tinderbox contains a flint, a steel, some tinder, and a patch of carbonized cloth. The description of the flint is "A flat grey stone with flaked edges." The description of the steel is "Curved so that you can hold it over the knuckles of your right hand and strike it against the flint. There is a knack to it. Those without the knack end up with bloody knuckles and no fire." The steel is wearable. The description of the tinder is "Dried grass and similar." The description of the cloth is "The little, precious, spark-preserving scraps without which the fire would never begin."

Instead of attacking the flint when the steel is not worn by the player:

say "You must wear the steel over your knuckles, in order to hit the flint at the best angle."

Instead of attacking the flint when the steel is worn and the cloth is not in the location:

say "Though you strike the flint sharply with the steel and throw sparks, they have nothing to catch on, since the patch of cloth does not lie beneath."

Instead of attacking the flint when the steel is worn and the cloth is in the location and the cloth is not unlit:

say "The patch of cloth has already caught."

Instead of attacking the flint:

now the cloth is glowing;

say "You strike the flint against the steel and throw sparks onto the patch of cloth; they make tiny circles of orange there, which will only prosper if blown into flame."

Realistically, we ought to attach a randomization to this so that each step of the fire-starting has a good chance of failure. But because our player may not be as patient as someone who actually needs a fire started, we allow him to succeed the first time in every case.

Ignition is a kind of value. The ignitions are whole, fading, glowing, flaming. A thing has an ignition.

Blowing on is an action applying to one thing. Understand "breathe on [something]" or "blow on [something]" as blowing on.

Instead of blowing on the whole cloth:

say "There is no point, since no sparks have caught there."

Instead of blowing on the fading cloth:

now the cloth is flaming;

say "You blow on the faint sparks on the cloth and turn them into the beginnings of flame."

Instead of burning the whole tinder:

if the cloth is not flaming:
say "The patch of cloth must catch flame before you can light anything with it.";
otherwise:
now the tinder is flaming;
now the cloth is nowhere;
say "You light the tinder with the patch of cloth, and have the elements of a fire."

Every turn:

unless the cloth is flaming or the cloth is whole:
now the ignition of the cloth is the ignition before the ignition of the cloth;
say "Now the patch of cloth is [ignition of the cloth]."

Some kindling is in the campsite.

Instead of burning the whole kindling:

if the tinder is not flaming:
say "You need the tinder to be flaming, first.";
otherwise:
now the tinder is nowhere;
now the kindling is nowhere;
move the campfire to the location;
say "You succeed in lighting yourself a proper campfire.";
now the printed name of Campsite is "By The Campfire".

Test me with "i / drink water / i / drink water / i / wear steel / get flint / get cloth / drop cloth / get tinder / hit flint / blow on cloth / burn tinder / burn kindling / look".

150



Example Bruneseau's Journey

WI

A candle which reacts to lighting and blowing actions differently depending on whether it has already been lit once.

"Sire," said the Minister of the Interior to Napoleon, "yesterday I saw the most intrepid man in your Empire." - "What man is that?" said the Emperor brusquely, "and what has he done?" - "He wants to do something, Sire." - "What is it?" - "To visit the sewers of Paris."

This man existed and his name was Bruneseau.

- Victor Hugo, Les Miserables

Let's say that our intrepid explorer has a candle that can be lit and blown out again, and should accordingly appear unlit, burning, or partly burnt:

"Bruneseau's Journey"

The Sewer Beneath St Denis is a room. "A narrow, stone-lined passageway, with only a little ledge to walk above the level of the refuse that flows down towards the Seine."

The candle is carried by the player. The description of the candle is "A candle, [if the candle has been lit]partially burnt[otherwise]still in pristine condition with untouched wick[end if]."

Instead of examining the lit candle, say "It burns with a pure heart."

The block burning rule is not listed in the check burning rules.

Instead of burning the lit candle:
say "The candle is already lit."

Check burning:
if the noun is not the candle, say "[The noun] cannot profitably be set on fire."

Carry out burning the candle:
now the candle is lit.

Report burning:
if the candle had been lit, say "You relight the candle.";
otherwise say "You light the candle for the first time."

Understand "blow out [something]" as blowing out. Understand "blow [something] out" as blowing out. Blowing out is an action applying to one thing.

Carry out blowing out the candle:
now the candle is unlit.

Report blowing out:
if the noun is the candle and the candle was lit, say "You blow out [the noun].";
otherwise say "You blow on [the noun], to little effect."

Test me with "x candle / light candle / x candle / blow out candle / x candle".

We must be careful: "if the noun was lit" would throw errors because past-tense rules can only be applied to specific items, not to variables that could be anything.

Creating a class of matches that burn for a time and then go out, with elegant reporting when several matches go out at once.

Here we create a class of matches that can be used to burn other objects. Our objectives are as follow:

Burned objects other than matches should be removed from play instantly (just as edible objects are instantly eaten). We could give everything its own burning

duration, but that complicates matters and allows for fire to spread from one object to another; for an example of how to do that, see the example "In Fire or in Flood".

Matches should be described to show whether they are burning or extinguished, and when the parser chooses one of several identical matches, it should make very clear which match it has selected.

The game must sensibly select and, if necessary, automatically light new matches to carry out a >BURN THING command.

The matches must burn for a set number of turns before going out, never to be used again.

And finally, the part for which the text will be useful: when several matches go out in the same turn, we want the game to print

Four matches go out.

rather than

A match goes out.
A match goes out.
A match goes out.
A match goes out.

This last function appears down in Section 3, if we wish to skip ahead and look at it.

"The Cow Exonerated"

Section 1 - Simple Burning

Understand the commands "light" and "burn" as something new.

Understand "burn [something] with [striable-match]" as burning it with.
Understand "burn [something] with [something preferably held]" as burning it with. Burning it with is an action applying to one thing and one carried thing.

Understand the command "light" as "burn".

A thing can be flammable or impervious. A thing is usually impervious.

Check burning something with something (this is the burn only with flaming matches rule):

if the second noun is not a striable-match, say "You can only light things with matches." instead;

if the second noun is not flaming, say "[The second noun] needs to be burning first." instead.

Check burning something with something (this is the burn only flammable things rule):

if the noun is impervious, say "[The noun] cannot be burned." instead.

Check burning something with something (this is the burn only things not held rule):

say "[one of]It occurs to you to set down [the noun] before burning, just for safety's sake. [or]Again, you decide to put down [the noun] prior to burning. [or]You try setting down [the noun] as usual. [stopping][run paragraph on]";
silently try the player dropping the noun;
if the player encloses the noun, stop the action.

Carry out burning something with something (this is the simplistic burning rule):
now the noun is nowhere.

Report burning something with something:
say "You burn up [the noun]."

Rule for implicitly taking the second noun while burning something with something which is not a strikable-match:
say "You can only light things with matches.";
stop the action.

Section 2 - Matches

The word "matches" is used by Inform to compare snippets of text (see "Reading a command" in the Activities chapter). This can sometimes cause awkwardness if we also have a kind called "match", so for the occasion we will give our matches a more specialized name, never visible to the player:

A strikable-match is a kind of thing. The plural of strikable-match is s-matches.

A strikable-match has a number called duration. The duration of a strikable-match is usually 3.

Rule for printing the name of a strikable-match: say "match".
Rule for printing the plural name of a strikable-match: say "matches".

Understand "match" as a strikable-match. Understand "matches" as a strikable-match.

Flame-state is a kind of value. The flame-states are burnt, flaming, and new.
Understand "burning" or "lit" as flaming. Understand "unused" as new.

A strikable-match has a flame-state. A strikable-match is usually new.
Understand the flame-state property as describing a strikable-match.

Before printing the name of a strikable-match while asking which do you mean:
say "[flame-state]".

Before printing the name of a strikable-match while taking inventory:
say "[flame-state]".
Before printing the plural name of a strikable-match while taking inventory:
say "[flame-state]".

Before printing the name of a strikable-match while clarifying the parser's choice of something:
if not taking inventory, say "[flame-state]".

After printing the name of a strikable-match (called special-target) while clarifying the parser's choice of something:

- if the player carries the special-target:
 - say " you're carrying";
- otherwise if the special-target is in the location:
 - say " on the ground";
- otherwise:
 - say " [if the holder of the special-target is a container]in[otherwise]on[end if] [the holder of the special-target]".

Understand "strike [something]" as attacking.

Understand "strike [strikable-match]" as striking. Striking is an action applying to one carried thing.

Understand "burn [strikable-match]" as striking.

Does the player mean striking a new strikable-match:
it is very likely.

Does the player mean striking a burnt strikable-match:
it is unlikely.

Check striking a strikable-match (this is the strike only new matches rule):

- if the noun is burnt, say "[The noun] has already burnt down and cannot be relit." instead;
- if the noun is flaming, say "[The noun] is already burning." instead.

Carry out striking a strikable-match (this is the standard striking rule):

- now the noun is flaming;
- now the noun is lit.

Report striking a strikable-match (this is the standard report striking rule):
say "You light [the noun]."

Before burning something with a new strikable-match (this is the prior lighting rule):

- say "(first [if the player does not carry the second noun]taking and [end if]lighting [the second noun])[command clarification break]";
- silently try striking the second noun;
- if the second noun is not flaming, stop the action.

Rule for implicitly taking a strikable-match (called target) while striking:
try silently taking the target.

Does the player mean burning something with a flaming strikable-match:
it is very likely.

Does the player mean burning something with a new strikable-match:
it is likely.

Does the player mean burning something with a burnt strikable-match:
it is unlikely.

Instead of burning a burnt strikable-match with something:
say "[The noun] is completely consumed and cannot be relit."

Section 3 - Putting the Matches Out

Every turn:

```
let N be 0; [here we track how many matches are being put out during this
turn, so that we don't have to mention each match individually if several go out
during the same move]
repeat with item running through flaming s-matches:
  decrement the duration of the item;
  if the duration of the item is 0:
    now the item is burnt;
    now the item is unlit;
    if the item is visible, increment N;
if N is 1:
  say "[if the number of visible flaming s-matches is greater than 0]One of the
matches [otherwise if the number of burnt visible s-matches is greater than
1]Your last burning match [otherwise]The match [end if]goes out.";
  otherwise if N is greater than 1:
    let enumeration be "[N in words]";
    if N is the number of visible s-matches:
      if N is two, say "Both";
      otherwise say "All [enumeration]";
    otherwise:
      say "[enumeration in title case]";
  say " matches go out[if a visible strikable-match is flaming], leaving [number
of visible flaming s-matches in words] still lit[end if]."
```

Section 4 - Scenario

Old Chicago is a room.

The player carries a flammable thing called a log. Understand "wooden" and
"wood" as the log.

The player carries two s-matches. The matchbox is an open openable container.
It contains five s-matches. The player carries the matchbox.

When play begins:

```
now every strikable-match carried by the player is flaming;
now every strikable-match carried by the player is lit.
```

Test me with "i / burn match / i / i / burn log with match / burn matchbox with
match / i".



"In Fire or in Flood"

Part I - Rules for combustion

Heat is a kind of value. The heats are whole, damp, and flaming. A thing has a heat. A thing is usually whole.

A thing has a number called endurance. The endurance of a thing is usually 5. A thing has a number called turns of burning. A thing can be flammable or flame-retardant.

Before printing the name of something flaming:
say "flaming ".

Before burning something when the player is not carrying something flaming:
if a flaming portable thing (called the lighter) is touchable:
say "(with [the lighter], which you first take)[command clarification break]";
try taking the lighter.

Instead of burning something when the player is not carrying something flaming:
say "You would first need a fire source."

Instead of burning something flame-retardant:
say "[The noun] is not the sort of thing that catches fire."

Instead of burning something flammable when the player is carrying something flaming (called the flame source):
say "You light [the noun] with [the flame source].";
now the heat of the noun is flaming.

Instead of burning something when the player is in the noun:
say "That seems dangerous given that you yourself are in [the noun]."

Instead of burning something when the player is on the noun:
say "That seems dangerous given that you yourself are on [the noun]."

Instead of examining something:
say "Hm, the [printed name] appears to be [heat]."

Before taking a flaming thing:
let turns remaining be the endurance of the noun minus the turns of burning of the noun;
if turns remaining is less than two, say "There's no portion of [the noun] sufficiently cool for you to pick up." instead.

But that's only a small part of the battle. The thing about fire is that it keeps on doing fiery things even when the player is otherwise occupied: destroying items that are on fire, and spreading to other things nearby. So we need a set of rules for the fire's behavior.

Every turn when something is flaming:
follow the fire rules.

The fire rules is a rulebook.

A fire rule (this is the can't hold flaming objects rule):
repeat with item running through flaming things:
if the item is held by the player:
let turns remaining be the endurance of the item minus the turns of
burning of the item;
if turns remaining is less than two:
say "[The item] becomes too hot to hold! ";
try dropping the item;
if the item is held by the player, say "This is certainly painful."

A fire rule (this is the flames spread rule):
repeat with item running through flaming things:
if the turns of burning of the item is one:
spread the flames from the item.

A fire rule (this is the fire destroys things rule):
now started printing is false;
repeat with item running through flaming things:
increment the turns of burning of the item;
if the turns of burning of the item is greater than the endurance of the item,
destroy the item;
if started printing is true, say "[paragraph break]";
now started printing is false.

Because we've labelled all the fire rules, we could swap their order, or turn some of them off, while allowing the others run as usual. For instance, if there were a pair of fireproof gloves in the game, we might want to turn off the "can't hold flaming objects rule" whenever the player is wearing them.

This sort of flexibility is especially useful in the context of extensions. Someone writing an extension about burning would have no way of anticipating the need for a Fireproof Gauntlet of Thog, but the author would nonetheless be able to implement one easily.

Definition: a thing is vulnerable if it is flammable and it is whole.

The contact between things is a critical factor when it comes to fire, so we might add a couple of conditional relations do determine what is touching what.

Reliance relates a thing (called X) to a thing (called Y) when X is part of Y or X is in Y or X is on Y. The verb to be relying on means the reliance relation.

Contact relates a thing (called X) to a thing (called Y) when X is relying on Y or Y is relying on X. The verb to be joined to means the contact relation.

Having these at our disposal makes it much tidier to write what happens next:

To spread the flames from (item - a thing):
now started printing is false;
if the item is joined to a flammable whole thing (called the sacrifice):
if the sacrifice is visible:
now started printing is true;
say "Flames engulf [the list of flammable whole things which are joined to

the item].";
now all the flammable whole things joined to the item are flaming.

Started printing is a truth state that varies. Started printing is false.

To destroy (item - a thing):
let home be the holder of the item;
if the item is part of something (called the superstructure), let home be the holder of the superstructure;
if the item is visible:
now started printing is true;
say "[The item] burns away[if something is relying on the item], leaving [a list of things which are relying on the item] behind[end if]. ";
if something is relying on the item,
now all the things which are relying on the item are in the home;
now the item is nowhere;
now the item is damp;
now every flaming thing which is part of the item is damp.

To destroy (item - a door):
let home be the holder of the item;
if item is visible:
now started printing is true;
say "[The item] burns away[if something flame-retardant is part of the item], leaving [a list of flame-retardant parts of the item] behind[end if]. ";
if home is a room, now all of the flame-retardant parts of the item are in the home;
now the item is damp;
now the item is open;
now the item is unopenable.

Before printing the name of a damp door:
say "burnt-out frame of ".

Instead of opening or closing a damp door:
say "[The noun] can no longer be opened or closed in any meaningful sense."

Instead of doing something other than examining or dropping to a flaming thing when the turns of burning of the noun is greater than 1:
say "Fire has too thoroughly engulfed [the noun] for that to be a good idea."

Instead of taking something when the noun is in a flaming thing (called the receptacle):
say "You don't quite dare reach into [the receptacle]."

Instead of touching something which is within a flaming thing (called the receptacle):
say "It seems a little risky since [the receptacle] is on fire."

Instead of turning something when the noun is contained in a flaming thing (called the receptacle):
say "It seems a little risky since [the receptacle] is on fire."

Instead of pushing or pulling something when the noun is inside a flaming thing (called the receptacle):

say "[The receptacle] deters you."

Before burning something which is in a container when the holder of the noun contains the player:

say "This could make things toasty for you..."

And that completes the rules which cover burning: things can catch fire, fire will spread, and gradually consume the world in flames. All of that was general and could be used in any setting, but we now provide a small game to show it off.

Part II - Escape from the Library of the Dead

The Library of the Dead is a room. "This room -- little, dank, stone -- is filling with some miasma you do not quite dare breathe. It is imperative that you get out."

The desk is a flammable supporter in the Library. A drawer is part of the desk. The drawer is a flammable closed container. It is openable, lockable, and locked. The desk is scenery.

A box is in the Library. A metal hinge is part of the box. The hinge is flame-retardant. The box is open, flammable, and openable. The shroud of Laertes is a flammable thing in the box.

Instead of examining something when something is part of the noun:

say "You note [the list of things which are part of the noun]."

The world's last manuscript of the Psychagogoi by Aeschylus is on the desk. The manuscript is flammable. The manuscript has endurance 1.

The torch is a flammable flaming thing carried by the player. It has endurance 60. The asbestos sack is a flame-retardant player's holdall in the drawer.

The trapdoor is up of the Library and east of the Plaza. The trapdoor is a door. It is flammable, closed, lockable, and locked. "A trapdoor in the ceiling is your only hope of escape[if flaming]. Fortunately, it is rapidly burning through[end if]." The trapdoor has endurance 15.

Instead of going through the closed trapdoor, say "[The trapdoor] is closed."

We can then add a special fire rule to handle the trapdoor, which will be called as part of the same sequence. Again, this would be most important if the fire rules were part of a standard extension, and the trapdoor fire rule the author's own addition.

A fire rule:

if the trapdoor is flaming and a random chance of 1 in 3 succeeds:
let the caught thing be a random flammable whole thing which can be touched by the trapdoor;
if the caught thing is a thing:
say "A spark from [the trapdoor] catches [the caught thing]!";
now the caught thing is flaming.

Instead of going to the Plaza:

say "Out at last!";
end the story finally.

Test me with "get manuscript / get shroud / light desk / look / g / open drawer / look / g / g / g / get sack / put shroud in sack / put manuscript in sack / close sack / light trapdoor / look / g / g / g / g / g / g / g / g / g / g / up".

84

★ Example Grilling

WI

A grill, from which the player is not allowed to take anything lest he burn himself.

Descriptions of objects can be used in "Instead" rules: we can not only say "Instead of taking the steak", but also "Instead of taking something" or "Instead of taking something which is on the grill".

That last rule is useful if, for example, we want to prevent the player from interacting with anything on a specific supporter:

"Grilling"

The Patio is a room. The Patio contains a grill and an ice chest. The ice chest contains a cold beer. On the grill are a steak and a hot dog.

Mom is a woman in the Patio.

Instead of taking something which is on the grill:
say "'Hey, you'll burn yourself,' says Mom."

Test me with "get steak / get all from grill / get all".

We could just as easily adapt this rule to affect a container: "Instead of taking something which is in the ice chest," for example.

Note also that in older versions of Inform, the pattern "get all from..." was treated differently from "get steak", and had to be accounted for separately. This is no longer the case; this instead of taking... rule will handle all the phrasings which the player might use to try to acquire this object.

182

★ Example Entropy

WI

All objects in the game have a heat, but if not kept insulated they will tend toward room temperature (and at a somewhat exaggerated rate).

"Entropy"

Heat is a kind of value. The heats are frosty, cold, cool, room temperature, warm, hot, and scalding. Everything has a heat. The heat of a thing is usually room temperature.

Every turn:

repeat with item running through things which are not in an insulated container:

if the heat of the item is greater than room temperature, now the heat of the item is the heat before the heat of the item;

if the heat of the item is less than room temperature, now the heat of the item is the heat after the heat of the item.

Definition: a container is insulated if it is closed and it is opaque.

The vacuum thermos is an opaque closed openable container carried by the player. In the vacuum thermos is a frosty thing called an ice cube.

Every turn:

if the heat of the ice cube is greater than cold:

if the ice cube is visible, say "The ice cube melts! 'HA ha,' says Maxwell, in a very unsporting, some might say demonic, way.";

now the ice cube is nowhere.

Before printing the name of something: say "[heat] ".

Equilibrium is a room. "A perfectly smooth chamber sealed from the outside world. You can't at this moment work out where the exit is, though possibly that is just because the lighting is so very very even and diffuse. And doesn't come from anywhere that you can see, either."

Maxwell is a man in Equilibrium. "Maxwell perches awkwardly on a stool across from you[if Maxwell has something], holding [a list of things carried by Maxwell] [end if]." He is carrying a box of Chinese food. The Chinese food is scalding. "A discarded [item described] lies on the floor." The description of Maxwell is "He has the faintly peevish look of one who has not been properly fed."

Every turn when Maxwell has the food:

if the heat of the Chinese food is greater than warm, say "Maxwell takes a bite, and swears.";

if the heat of the Chinese food is warm, say "Maxwell eats as fast as he can, enjoying the food while it's at just the right temperature.";

if the heat of the Chinese food is less than warm:

say "Maxwell sadly stabs at his leftovers with a chopstick, but does not try to eat any more.";

move the food to the location.

Test me with "z / z / open thermos / close thermos / open thermos".

326

★ Example Masochism Deli

WI

Multiple potatoes, with rules to make the player drop the hot potato first and pick it up last.

Here the player has several potatoes; we would like to make him more likely to drop the hot one, and more likely to pick up the cold one, all else being equal. At the same

time, we want to phrase our rules so that they don't make the player try to take something he's already holding, or drop something he isn't.

So:

"Masochism Deli"

The Masochism Deli is a room. "Recent restructurings of corporate policy restrict the 'lunch hour' to exactly thirty-two minutes, which means that no one has time to go out. Instead, you and your coworkers eat here, in the company's very own themed lunch room."

The plural of potato is potatoes. A potato is a kind of thing. A potato is edible.

Temperature is a kind of value. The temperatures are hot and cold. A potato has a temperature. A potato is usually cold.

Understand the temperature property as describing a potato. Before printing the name of a potato (called subject): say "[temperature of subject] ". Before printing the plural name of a potato (called subject): say "[temperature of subject] ".

Does the player mean dropping a hot potato which is carried by the player: it is very likely.

Does the player mean taking a cold potato which is not carried by the player: it is very likely.

The player carries three potatoes.

After dropping a hot potato:

say "The guy from Cube B sneers at your lack of potato-holding stamina."

When play begins: now a random potato is hot.

Test me with "inventory / drop potato / g / g / get potato / g / i / get potato".

331



Example Hot Glass Looks Like Cold Glass

WI

Responding to references to a property that the player isn't yet allowed to mention (or when not to use "understand as a mistake").

Suppose we have a situation where the player is allowed to talk about the heat of an object only if he's properly equipped to detect it.

"Hot Glass Looks Like Cold Glass"

Use scoring.

Heat is a kind of value. The heats are hot, warm, room temperature, and cold. A thing has a heat.

Understand the heat property as referring to a thing when the player wears the infrared goggles.

The Test Kitchen is a room. "Your own personal lab, ready for scrupulously scientific recipe research. You hope. The previous three runs of this did not go well." The pair of infrared goggles is carried by the player. The description is "A pair of head-mounted IR goggles which look very foolish when worn." The goggles are wearable.

A glass dish is a kind of container. A glass dish is transparent. Three room temperature glass dishes are on the counter. Two hot glass dishes are on the counter. Two cold glass dishes are on the counter. The counter is scenery in the Test Kitchen.

Instead of doing something other than examining to a hot glass dish:
say "Ow! Crikey! You swear, and Claudia makes a sympathetic hiss. 'You're going to have a mark from that for sure,' she comments.";
decrease the score by 2.

Instead of doing something when a hot glass dish is the second noun:
say "You brush [the second noun], and wince, but manage to conceal that from Claudia.";
decrement the score.

Before printing the name of a glass dish when the player wears the goggles: say "[heat] "

Before printing the plural name of a glass dish when the player wears the goggles: say "[heat] "

So far, so good. Now, what if the player tries to GET HOT DISH when the goggles are off? "You can't see any such thing." doesn't seem like quite the right response: he can see such a thing. He just doesn't know which it is.

We could go on to write a mistake rule that would scold the player for trying "get [heat] [text]" when not wearing the goggles. The problem is that this would not cover any other phrasing of the command, nor would it account for all the many other things the player might try to do with an object specified by heat.

What we really want is to catch all instances of the player using the property name when not allowed to do so; and for this purpose we can borrow a trick from the chapter on Activities:

After reading a command:
if the player wears the goggles, make no decision;
if the player's command includes "[heat]":
say "Without the IR goggles on, you cannot tell hot things from cold at sight.";
rule succeeds.

Claudia is a woman in the Test Kitchen. "Your assistant Claudia stands by with [a list of things carried by Claudia]." The description of Claudia is "Infinitely patient and a very good stenographer. She is studiously avoiding giving you any sort of

look that might be construed as mocking." Claudia carries a notepad, a brined chicken breast, a blowtorch, and a cup of heavy cream.

Instead of asking Claudia for something which is carried by Claudia:

move the second noun to the player;
say "Check, [second noun],' repeats Claudia, in the tone of one who has seen too many medical dramas. She does hand it over, though."

Instead of asking Claudia for something:

say "She clears her throat faintly and glances at [the second noun], as though to say that it's not hers to give."

Test me with "get all / drop all / look / wear goggles / look / ask claudia for cream / put cream in hot dish / put cream in cold dish / remove goggles / get hot dish".

215

★ Example Magneto's Revenge

WI

Kitty Pryde of the X-Men is able to reach through solid objects, so we might implement her with special powers that the player does not have...

"Magneto's Revenge"

The School for Gifted Youngsters is a room. Kitty Pryde is a woman in the School for Gifted Youngsters. "Kitty Pryde waits for you to say why you summoned her out of class." The description is "You see nothing special about Kitty Pryde. But that is only because she has learned to conceal her mutant powers: in fact she has the alternative name Shadowcat, and the ability to phase through solid objects."

The glass box is a container in the School for Gifted Youngsters. It is closed, openable, and transparent. In the box is a quantity of poisonous gas. In the box is a thing called the message from Magneto.

Instead of examining the message when the player is not carrying the message:
say "You'd need to be holding it before you could read it."

Instead of opening the glass box: say "You're deterred by the swirling bottle-green mist within."

Check someone taking the gas (this is the gaseous object rule): rule fails.

Unsuccessful attempt by someone taking the gas: say "The gas isn't something one can pick up in one's bare hands."

Every turn:

if the player can touch the gas:
say "The gas has reached your lungs!";
end the story.

A rule for reaching inside something:
if the person reaching is Kitty, allow access.

Persuasion rule for asking someone to try doing something: persuasion succeeds.

Test me with "get message / kitty, get message / open box / kitty, open box".

Now the player can get Kitty to take the message without releasing the poisonous gas:

Kitty Pride waits for you to tell her why you summoned her out of class.

You can also see a glass box (closed) (in which are some poisonous gas and a message from Magneto) here.

>kitty, get message
Kitty Pryde picks up a message from Magneto.

225

★ Example Interrogation

WI

A wand which, when waved, reveals the concealed items carried by people the player can see.

"Interrogation"

The X-Ray Vision Wand is carried by the player.

Instead of waving the X-Ray Vision Wand:
say "Disappointingly, nothing happens."

Instead of waving the X-Ray Vision Wand when the player can see someone who is concealing something:

say "The wand glows green. Immediately you see on the monitor [a list of things which are concealed by people who can be seen by the player]."

After printing the name of a thing (called target) which is carried by someone while waving the wand:

say "(carried by [a random person who carries the target])"

The Interrogation Chamber is a room. "Despite its gothic name, this room is a rather civilized place for your work, with large plate-glass windows and a fitted carpet."

A thing can be secret or obvious. A thing is usually obvious.

Brian is in the Interrogation Chamber. "Brian lounges against the wall." Brian carries a quantity of plastic explosive. The explosive is secret.

Janine is in the Interrogation Chamber. "Janine toys nervously with a laptop bag." Janine carries a chocolate biscuit, a laptop bag, and a microfilm. The

microfilm is secret. The laptop bag is wearable. In the bag is a laptop computer.

Rule for deciding the concealed possessions of something: if the particular possession is secret, yes; otherwise no.

Instead of examining someone: say "[The noun] is openly carrying [a list of unconcealed things carried by the noun]."

Test me with "wave wand / examine janine / examine brian".

227

★ Example Transmutations

WI

A machine that turns objects into other, similar objects.

Suppose we want to have a machine in our game that can transmute one item into another, similar object with different properties: a bag of jelly beans into a bag of jewels, for instance. Thus each item will be associated with some number of equivalents -- the other objects it can turn into. This is a handy use for group relations:

"Transmutations"

Workshop is a room.

Transmutation relates things to each other in groups. The verb to become means the transmutation relation.

Definition: a thing is transmutable if it becomes more than one thing. [* It always becomes itself.]

A thing can be valuable. Something valuable called a bag of jewels is carried by the player. It becomes the bag of gunpowder and the bag of jelly beans.

A thing can be dangerous. The bag of gunpowder is a dangerous thing.

The bag of jelly beans is an edible thing.

The machine is fixed in place in the workshop.

The can't insert into what's not a container rule does nothing when inserting something into the machine.

Check inserting something which is not transmutable into the machine:
instead say "You can't transmute that."

To decide which thing is new form of (obj - edible thing): decide on a random valuable thing which becomes obj.

To decide which thing is new form of (obj - dangerous thing): decide on a random edible thing which becomes obj.

To decide which thing is new form of (obj - valuable thing): decide on a random dangerous thing which becomes obj.

Carry out inserting something into the machine:
now the noun is nowhere;
now the player carries the new form of the noun.

Report inserting something edible into the machine:
say "The machine clicks, whirrs, and spits out [a new form of the noun]. You're rich!";
rule succeeds.

Report inserting something dangerous into the machine:
say "The machine clicks, whirrs, and in a shower of flavor crystals, spits out [a new form of the noun].";
rule succeeds.

Report inserting something valuable into the machine:
say "The machine clicks, whirrs, and with a violent roar, spits out [a new form of the noun].";
rule succeeds.

Test me with "i / put jewels in machine / i / put gunpowder in machine / i / put beans in machine".

In this example we have only defined a single set of transmutable objects, but we could easily expand to include other groups.

(Thanks to Jesse McGrew for proposing this example.)

398

Example Access All Areas

WI

The Pointy Hat of Liminal Transgression allows its wearer to walk clean through closed doors.

If somebody tries to walk through a closed door, the "can't go through closed doors rule" usually stops them. This is a rule belonging to the "check going" rulebook. These names are fairly explanatory when written out, but hard to remember: fortunately we don't need to remember them, as the Index panel contains a full inventory of the check, carry out and report rules for every action, showing all of their names and the order in which they are checked. (We can also find out which rules are stopping an action by typing the testing command ACTIONS.)

Here we make the rule do nothing provided a condition holds:

"Access All Areas"

The extremely difficult door is north of the Standing Room and south of the Room of Walking Upside Down. It is a locked door.

The player is carrying the Pointy Hat of Liminal Transgression. The hat is wearable.

The can't go through closed doors rule does nothing when the Hat is worn.

Test me with "n / wear hat / n".

(The Pointy Hat may be useful in debugging a game, even if it never makes it into the final published work.)

240

★ Example Number Study

WI

The parity and joint magnitude relations explored.

This is the same case given above, but expanded just slightly to demonstrate that the names of the relations can also be printed, if we like:

"Number Study"

Abstraction is a room.

Parity relates a number (called N) to a number (called M) when N minus M is even.

Joint magnitude relates a number (called N) to a number (called M) when N plus M is greater than 7.

To chart (R - a relation of numbers):
repeat with N running from 1 to 5:
repeat with M running from 1 to 5:
if R relates N to M, say "[N] <=> [M] by [R]";

When play begins:
let L be { parity relation, joint magnitude relation };
repeat with R running through L:
chart R.

As this shows, we can even form lists of relations. The kind of L is "list of relations of numbers".

434

★ Example The Fibonacci Sequence

WI

The modest Leonardo Fibonacci of Pisa will be only too happy to construct his sequence on request, using an array.

Fibonacci (a posthumous nickname) spread Arabic mathematical learning across Europe in the 13th century, and it's curious that his name lives on only for a single sequence.

"The Fibonacci Sequence"

Pisa is a room. Leonardo Fibonacci is a man in Pisa. "The modest Italian mathematician, Leonardo Fibonacci (1170-1250), beams at you."

Sequencing is an action applying to one number. Understand "sequence [number]" as sequencing.

Instead of sequencing, say "You make a feeble attempt, sketching in the sand, but it goes nowhere. Leonardo is sympathetic. 'Often goes wrong for me, too, actually. I didn't even invent the thing - the ancient Indians knew about it first.'"

Persuasion rule for asking Leonardo to try sequencing: persuasion succeeds.

Report Leonardo sequencing:

let N be the number understood;
say "Leonardo scratches his head and makes self-deprecating remarks, before coming up with [the first N terms of the Fibonacci sequence]."

An array need not be fixed in length, as the following example shows:

To decide what list of numbers is the first (F - a number) terms of the Fibonacci sequence:

```
let the Fibonacci sequence be {1, 1};  
let N be 3;  
while N < F:  
  let the last term be entry (N - 1) of the Fibonacci sequence;  
  let the penultimate term be entry (N - 2) of the Fibonacci sequence;  
  let the next term be the last term plus the penultimate term;  
  add the next term to the Fibonacci sequence;  
  increment N;  
decide on the Fibonacci sequence.
```

Test me with "sequence 20 / leonardo, sequence 20".

The result of "the first 20 terms of the Fibonacci sequence" is "1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584 and 4181". This is a sequence which has a knack of turning up in odd places - it was found in the 1970s to be related to the rings of florets in a sunflower, for instance - and here it is in a book about interactive fiction.

437

★ Example Sieve of Eratosthenes

The haughty Eratosthenes of Cyrene will nevertheless consent to sieve prime numbers on request.

WI

In the words of Wikipedia: "Eratosthenes of Cyrene (Greek Ερατοσθένης; 276 BC-194 BC) was a Greek mathematician, poet, athlete, geographer and astronomer." In the words of Tom Lehrer: "It's people like that who make you realise how little you've achieved in life."

A prime number is a number greater than 1 which is not a multiple of anything, so we can find the primes by starting with all the numbers and sieving out all the multiples of 2, then all the multiples of 3, and so on. Here we make our sieve of the unacceptable numbers (the "composite" or non-prime ones) first, then form a list of all the numbers, then sieve out the composites: what are left must be the primes.

"Sieve of Eratosthenes"

Alexandria is a room. Eratosthenes is a man in Alexandria. "The haughty Greek mathematician, Eratosthenes, glowers at you."

Sieving is an action applying to one number. Understand "sieve [number]" as sieving.

Instead of sieving, say "You make a feeble attempt, sketching in the sand, but it goes nowhere. Eratosthenes smirks. 'I expect your friends call you gamma, then?'"

Persuasion rule for asking Eratosthenes to try sieving: persuasion succeeds.

Report Eratosthenes sieving:

```
let N be the number understood;
let the composites be a list of numbers;
let I be 2;
while I times I is at most N:
  if I is not listed in the composites:
    let J be I times 2;
    while J is at most N:
      add J to the composites, if absent;
      increase J by I;
    increment I;
sort the composites;
let the primes be a list of numbers;
repeat with P running from 2 to N:
  add P to the primes;
remove the composites from the primes;
say "Eratosthenes sketches lines in the sand with the air of much practice.
'The primes up to [N] are [the primes]. The composites are [the composites]."
```

Test me with "sieve 10 / eratosthenes, sieve 100".

While this could all be done more efficiently with an array, that's only because what we are sieving are numbers: sieving is a technique which can be used for non-numerical decisions, too.

Chapter 11: Out Of World Actions and Effects

§11.1. Start-Up Features; §11.2. Saving and Undoing; §11.3. Helping and Hinting; §11.4. Scoring; §11.5. Settings and Status Checks During Play; §11.6. Ending The Story

-  Contents of *The Inform Recipe Book*
-  Chapter 10: Physics: Substances, Ropes, Energy and Weight
-  Chapter 12: Typography, Layout, and Multimedia Effects
-  Indexes of the examples

§11.1. Start-Up Features

When the story file starts up, it often prints a short introductory passage of text (the "overture") and then a heading describing itself, together with some version numbering (the "banner"). It is traditional that the banner must appear eventually (and one of the few requirements of the Inform licence is that the author acknowledge Inform somewhere, for which the banner is sufficient) but some designs call for a multi-turn prologue before the banner finally appears, and marks the start of play in earnest. **Bikini Atoll** demonstrates this.

If a story file represents the latest in a sequence of story files representing chapters in some larger narrative, it will need some way to pick up where its predecessor left off. This can be done with the aid of external files (in the Glulx format, at least). **Alien Invasion Part 23** shows how.

Another task we might want to perform at the beginning of play is to arrange any randomized features that are supposed to change from one playing to the next. We can add such instructions with "When play begins" rule, as in:

```
When play begins:  
  now the priceless treasure is in a random room.
```

Since we may want to do something a bit more complicated than this, **Hatless** demonstrates effective and ineffective methods of distributing multiple objects (in this case, one randomly-selected hat per person).

★ See **Map** for a way to generate a randomized maze at the start of play

★ See **Food** for a way to choose a random piece of candy to be poisonous

★ See **Getting Acquainted** for a way to choose a murderer from among the characters at the start of each story

-  Start of Chapter 11: Out Of World Actions and Effects
 -  Back to Chapter 10: Physics: Substances, Ropes, Energy and Weight: §10.11. Mathematics
 -  Onward to §11.2. Saving and Undoing
 -  Example 382:  **Bikini Atoll** Delaying the banner for later.
 -  Example 440:  **Alien Invasion Part 23** Keeping a preference file that could be loaded by any game in a series.
 -  Example 126:   **Hatless** It's tempting to use "now..." to distribute items randomly at the start of play, but we need to be a little cautious about how we do that.
-

§11.2. Saving and Undoing

A very few titles in the IF literature - very few being still too many, some would say - restrict the player's ability to save the story.

Removing the player's ability to UNDO is also a risky choice. Inform does provide the facility with the use option

[Use undo prevention.](#)

which makes it impossible to UNDO at any time (unless, that is, the player is playing on an interpreter that itself has a built-in UNDO feature -- these do exist). When it works, undo prevention safeguards a randomized story or combat session against brute-force solutions, but it also means that the player who makes even a minor mistake of typing will be stuck with the undesired results.

In many cases it may be preferable to use some subtler method to enforce random effects in a story. Several extensions exist for Inform that either allow selective manipulation of the UNDO command or rig randomization to prevent UNDO and replay attempts.

-  Start of Chapter 11: Out Of World Actions and Effects
 -  Back to §11.1. Start-Up Features
 -  Onward to §11.3. Helping and Hinting
 -  Example 210:  **Spellbreaker** P. David Lebling's classic "Spellbreaker" (1986) includes a room where the game cannot be saved: here is an Inform implementation.
 -  Example 211:    **A point for never saving the game** In some of the late 1970s "cave crawl" adventure games, an elaborate scoring system might still leave the player perplexed as to why an apparently perfect play-through resulted in a score which was still one point short of the supposed maximum. Why only 349 out of 350? The answer varied, but sometimes the last point was earned by never saving the game - in other words by playing it right through with nothing to guard against mistakes (except perhaps UNDO for the last command), and in one long session.
-

§11.3. Helping and Hinting

IF is difficult to play: often harder than the writer ever suspects. Players are held up by what is "obvious", and they stumble into unforeseen combinations, or spend inordinate amounts of time working on the "wrong" problems. Too much of this and they give up, or post questions on online forums. Against this, many IF authors like to include in-story hints.

There are many approaches, which differ on two main issues.

First: do we spontaneously offer help to the player? The difficulty here is detecting the player's need: **Y ask Y?** tries to spot aimlessness, while **Solitude** has a novice mode where it is reasonable to assume that help is almost always needed. On the other hand, suppose we require that the initiative come from the player. Will a novice know to type HELP? **Query** shows how to redirect any attempt to ask a direct question into a HELP request. At the other end of the scale, wearily experienced players may type HELP all the time, out of habit, cheating themselves of the fun of frustration: if so, **Real Adventurers Need No Help** provides the nicotine patch against this addiction.

Second: how do we decide what help is needed? Normally the player only types HELP, which is unspecific. The simplest approach offers a menu, diagnosing the player's problem by obliging him to make choices: see **Food Network Interactive**. Listing all the possible problems in the story may give away too much, though, since players may not have reached the puzzles in question yet; so some authors prefer to create menus that adapt to the current state of the story (commonly called "adaptive hints").

Failing this, we can also try to parse commands like HELP ABOUT MICRODOT, as in **Ish**. **Trieste** takes a similar tack, except that instead of offering hints about puzzles, it offers help on story features (such as how to save), and lists all the available topics if the player types simply HELP.

Finally, and perhaps most stylishly, we can try to deduce what the player is stuck on from his immediate circumstances and from what is not yet solved: this needs a powerful adaptive hints system like the one in **The Unexamined Life**.

★ See **Getting Started with Conversation** for a way to redirect a player using the wrong conversation commands

★ See **Footnotes** for another medium by which hints could perhaps be transmitted

- ⬆ Start of Chapter 11: Out Of World Actions and Effects
 - ⬅ Back to §11.2. Saving and Undoing
 - ➡ Onward to §11.4. Scoring
 - ⬇ Example 111: ★ **Y ask Y?** Noticing when the player seems to be at a loss, and recommending the use of hints.
 - ⬇ Example 284: ★ **Food Network Interactive** Using a menu system from an extension, but adding our own material to it for this game.
 - ⬇ Example 296: ★ **Ish.** A (very) simple HELP command, using tokens to accept and interpret the player's text whatever it might be.
 - ⬇ Example 329: ★ **Query** Catching all questions that begin with WHO, WHAT, WHERE, and similar question words, and responding with the instruction to use commands, instead.
 - ⬇ Example 285: ★★ **Trieste** Table amendment to adjust HELP commands provided for the player.
 - ⬇ Example 400: ★★★ **Solitude** Novice mode that prefaces every prompt with a list of possible commands the player could try, and highlights every important word used, to alert players to interactive items in the scenery.
 - ⬇ Example 54: ★★★★ **Real Adventurers Need No Help** Allowing the player to turn off all access to hints for the duration of a game, in order to avoid the temptation to rely on them overmuch.
 - ⬇ Example 230: ★★★★★ **The Unexamined Life** An adaptive hint system that tracks what the player needs to have seen or to possess in order to solve a given puzzle, and doles out suggestions accordingly. Handles changes in the game state with remarkable flexibility, and allows the player to decide how explicit a nudge he wants at any given moment.
-

§11.4. Scoring

Not every work of IF allots a numerical score to the player: for some authors, this emphasises the idea of a story rather than a narrative. The simple sentence

Use scoring.

introduces the concept. Once this is included, Inform will provide built-in support for a single number measuring progress ("score"), and will expect to measure this against a maximum possible ("maximum score", which can either be set by hand or worked out automatically from a table of ranks).

In a story in which scoring exists, the player may choose to turn score notifications (such as "[Your score has just gone up by one point.]") on or off. The commands to do this are NOTIFY ON and NOTIFY OFF; the actions are called switching score notification on and switching score notification off. In the event that we need to amend the behavior of notification, we could do so by adding, removing, or modifying the elements of the check and carry out rulebooks for these commands; as in

Check switching score notification off:

if the turn count is less than 10:

say "You are still a novice, grasshopper. Allow your teacher to give you advice until such time as you are ready to go on alone."

If we wish to change the wording of the default message ("[Your score has..."), we may want to use the Responses system.

An especially insidious style of bug allows the player to type the same sequence of commands over and over, earning score endlessly for the same insight, and to avoid this it is usually safest to write source like:

```
After taking the Picasso miniature when the Picasso miniature is not handled:  
  increase the score by 10;  
  say "As they say in Montmartre: dude!"
```

We might also write our condition with "for the first time", like so:

```
After jumping for the first time:  
  increase the score by 5;  
  say "Boing! That was certainly entertaining."
```

But we should be careful not to use "for the first time" in scoring situations where it's possible for the player to try the action but fail. Inform counts even unsuccessful attempts towards the number of times an action is understood to have occurred, so if the player tries to jump and fails, his "for the first time" will be used up and he will never receive the score points.

If there are many "treasure" items like the Picasso miniature, it is best to be systematic, as in **No Place Like Home**. **Bosch** takes another approach to the same idea, by creating a table of point-earning actions that the player will be rewarded for doing; the FULL SCORE command will then play these back.

Mutt's Adventure demonstrates how we might add a scored room feature, such that the player earns a point when he first arrives at a special room.

A single number does not really sum up a life, or even an afternoon, and **Goat-Cheese and Sage Chicken** and **Panache** offer more detailed citations. Works that are more story than story may prefer to offer a plot summary of the player's experience to date in lieu of more conventional scoring.

Finally, **Rubies** provides a scoreboard that keeps track of the ten highest-scoring players from one playthrough to the next.

- ↑ Start of Chapter 11: Out Of World Actions and Effects
 - ← Back to §11.3. Helping and Hinting
 - Onward to §11.5. Settings and Status Checks During Play
 - ↓ Example 219: ★ **Bosch** Creating a list of actions that will earn the player points, and using this both to change the score and to give FULL SCORE reports.
 - ↓ Example 136: ★★ **Mutt's Adventure** Awarding points for visiting a room for the first time.
 - ↓ Example 137: ★★★ **No Place Like Home** Recording a whole table of scores for specific treasures.
 - ↓ Example 166: ★★★★ **Panache** Replacing the score with a plot summary that records the events of the plot, scene by scene.
 - ↓ Example 280: ★★★★★ **Goat-Cheese and Sage Chicken** Implementing a FULL SCORE command which lists more information than the regular SCORE command, adding times and rankings, as an extension of the example given in this chapter.
 - ↓ Example 442: ★★★★★ **Rubies** A scoreboard that keeps track of the ten highest-scoring players from one playthrough to the next, adding the player's name if he has done well enough.
-

§11.5. Settings and Status Checks During Play

Several default actions allow the player some control over the presentation of the story, or permit the player to request information about what is going on. In addition to the standard commands described elsewhere in this section (SCORE, SAVE, UNDO, QUIT, RESTART, and RESTORE), Inform has the following actions that control the player's experience:

- Preferring abbreviated room descriptions (SUPERBRIEF)
- Preferring unabbreviated room descriptions (VERBOSE)
- Preferring sometimes abbreviated room descriptions (BRIEF)
- Switching score notification on (NOTIFY ON)
- Switching score notification off (NOTIFY OFF)

The first three of these allow the player to change the way rooms are described on first and subsequent versions; the last two, when used in a story that provides a score feature, toggle on and off reports such as "[Your score has just gone up by three points.]" These are discussed elsewhere in the Recipe Book (see cross-references below).

These provide immediate feedback about the status of the story file being played:

- Verifying the story file (VERIFY)
- Requesting the story file version (VERSION)
- Requesting the pronoun meanings (PRONOUNS)

VERIFY examines checksums to make sure that the story file being run is intact and correct. This is less often an issue now than it was in the days when story files were distributed by highly corruptible floppy disk, but the command persists and is very occasionally useful. VERSION gives the full banner text associated with the story, including title, author, release number, IFID, and other bibliographical data; it follows this with a list of the included extensions.

PRONOUNS announces to the player what the story is currently understanding as the antecedents of "him", "her", "it", and "them". This is often useful during testing, but sometimes also during play.

The following allow the player (when supported by his interpreter) to create a log of play:

[Switching the story transcript on \(TRANSCRIPT ON\)](#)
[Switching the story transcript off \(TRANSCRIPT OFF\)](#)

It is rarely a good idea to change the default performance of such commands: they are often finicky and closely tied to the interpreter in which the story runs. Moreover, disabling the "version" command means that the story file is not able to display attribution information for Inform and any included extensions, in violation of their respective licenses.

★ See **Looking** for a way to set the story's verbosity level for the player

★ See **Scoring** for a discussion of score notification

★ See **Testing** for some examples of status-check commands created for alpha- or beta-testing of a story

-
-  Start of Chapter 11: Out Of World Actions and Effects
 -  Back to §11.4. Scoring
 -  Onward to §11.6. Ending The Story
-

§11.6. Ending The Story

Play can end in many ways, at the writer's discretion:

[end the story;](#)
[end the story finally;](#)
[end the story saying "You have reached an impasse, a stalemate";](#)
[end the story finally saying "You have succeeded.";](#)

The phrase "end the story" by itself will finish play, printing "*** The End ***". Using one of the phrases with "saying..." allows us to specify some other text with which to conclude. Including "finally" means that the player has earned access to AMUSING text and other notes, if any of these are provided.

We can eliminate the asterisked headline entirely by removing the rule that prints it, thus:

[The print obituary headline rule is not listed in any rulebook.](#)

The next step is to print the player's score and, if applicable, the rank he achieved. By default a story doesn't feature scoring, but the following use option will incorporate it:

[Use scoring.](#)

Then, if we want to allow a score but alter the way it is reported, we may remove or modify the print final score rule, as in

The print final score rule is not listed in any rulebook.

or perhaps something like

The chatty final score rule is listed instead of the print final score rule in for printing the player's obituary.

This is the chatty final score rule: say "Wow, you achieved a whole [score in words] point[s] out of a possible [maximum score in words]! I'm very proud of you. This was a triumph. I'm being so sincere right now."

What happens next is normally that the player is invited to RESTART, RESTORE (from a saved story), QUIT or UNDO the last command. The presence of the question can somewhat undercut a tragedy, and **Battle of Ridgefield** shows another way to go out.

If we do leave the question in, the text is formed by the Table of Final Question Options, which by default looks like this:

Table of Final Question Options

final question wording	only if victorious	topic	final response rule	final response activity
"RESTART"	false	"restart"	immediately restart the VM rule	--
"RESTORE a saved story"	false	"restore"	immediately restore saved story rule	--
"see some suggestions for AMUSING things to do"	true	"amusing"	--	amusing a victorious player
"QUIT"	false	"quit"	immediately quit rule	--
"UNDO the last command"	false	"undo"	immediately undo rule	--

Because this is a table, we may alter the behavior by changing entries or continuing the table. **Finality** shows how we might take out the option to UNDO the last command, for instance.

Using an ending phrase that includes "finally" tells Inform to include the options that are marked "only if victorious". One common use is to let the player read some special bit of additional text, perhaps describing easter eggs he might have missed in the story or presenting some authorial notes. **Xerxes** demonstrates a simple AMUSING command to read final information, while **Jamaica 1688** shows how to add completely new elements to the list of options.

Old-school adventures expected their adventurers to die early and die often. **Labyrinth of Ghosts** shows how the residue of such past attempts can be preserved into subsequent attempts, using an external file. **Big Sky Country** shows how a player can be resurrected by, let us say, some beneficent god, so that a player can even die more than once in the same attempt.

- ↑ Start of Chapter 11: Out Of World Actions and Effects
 - ← Back to §11.5. Settings and Status Checks During Play
 - Onward to Chapter 12: Typography, Layout, and Multimedia Effects: §12.1. Typography
 - ↓ Example 383: ★ **Battle of Ridgefield** Completely replacing the endgame text and stopping the game without giving the player a chance to restart or restore.
 - ↓ Example 384: ★ **Finality** Not mentioning UNDO in the final set of options.
 - ↓ Example 385: ★ **Jamaica 1688** Adding a feature to the final question after victory, so that the player can choose to reveal notes about items in the game.
 - ↓ Example 386: ★★ **Xerxes** Offering the player a menu of things to read after winning the game.
 - ↓ Example 441: ★★★ **Labyrinth of Ghosts** Remembering the fates of all previous explorers of the labyrinth.
 - ↓ Example 138: ★★★ **Big Sky Country** Allowing the player to continue play after a fatal accident, but penalizing him by scattering his possessions around the game map.
-

Examples from Chapter 11: Out Of World Actions and Effects

- ↑ Start of this chapter
- Chapter 12: Typography, Layout, and Multimedia Effects
- ↘ Indexes of the examples

382

★ **Example Bikini Atoll**
 Delaying the banner for later.

WI

"Bikini Atoll" by Edward Teller

The Hut and the Tropical Beach are rooms.

The conch shell is in the Hut. After taking the shell for the first time: say "As you gather the oddly-warm conch shell into your arms, you experience a sudden flash of deja-vu...[banner text]"; move the player to the Tropical Beach.

Rule for printing the banner text when the player is not carrying the shell: do nothing.

Test me with "look / examine shell / get shell / look".

(By tradition, and as a courtesy to all the people who have worked on Inform, authors ensure that the banner is printed some time near the beginning of each game played. So please only defer it, rather than suppress it altogether.)

440

★ **Example Alien Invasion Part 23**
 Keeping a preference file that could be loaded by any game in a series.

WI

Suppose that we have a series of games each of which allows the player to select a puzzle difficulty level. When the player plays a new game in the series, we want him to start out by default with the same difficulty level he faced earlier on, so we store this information in a small preferences file, as follows:

"Alien Invasion Part 23"

A difficulty is a kind of value. The difficulties are easy, moderate, hard, and fiendish.

Understand "use [difficulty] puzzles" as selecting difficulty. Selecting difficulty is an action out of world, applying to one difficulty.

Carry out selecting difficulty:
choose row 1 in the Table of Preference Settings;
now challenge level entry is difficulty understood;
say "Puzzles will be [challenge level entry] from now on."

The File of Preferences is called "prefs".

When play begins:
if File of Preferences exists:
read File of Preferences into the Table of Preference Settings;
choose row 1 in the Table of Preference Settings;
say "(The current puzzle difficulty is set to [challenge level entry].)"

Check quitting the game:
write File of Preferences from the Table of Preference Settings.

Table of Preference Settings
challenge level
easy

The Sewer Junction is a room.

Our preference file is restricted to a single option here for simplicity's sake, but we could keep track of more information -- whether the player preferred verbose or brief room descriptions, screen configurations, and so on.

If we were disposed to be somewhat crueler, we could use a similar method to make the player finish each episode of the series in order to "unlock" the next. All we would need to do is store a numerical password in our preferences file when the player finished a given level; the next level would check, say, the Table of Completed Levels for that password, and refuse to play unless the right number were present.



Suppose we want a game in which each scenario starts with the characters wearing hats -- randomly passed out. We might be tempted to write our scenario like this:

"Hatless"

The Costumery is a room. Larry, Curly, and Moe are men in the Costumery.
Janine is a woman in the Costumery.

Rule for writing a paragraph about a person (called the target) who wears a hat (called attire):

say "[The target] is here, looking stylish in [an attire]."

Rule for writing a paragraph about a hatless person (called the target):

say "[The target] mopes about, hatless."

A hat is a kind of thing. A hat is always wearable. Definition: a person is hatless if he is not the player and he does not wear a hat.

The indigo bowler, the polka-dotted fedora, the pink beret, and the scarlet cloche are hats.

When play begins:

now every hat is worn by a random hatless person.

And we might hope that this would choose a new hatless person for each hat, but we would be wrong. It will instead choose one hatless person and put all the hats on him -- and everyone else has to go bare-headed. That's clearly no good. Let's try again:

"Hatless 2"

The Costumery is a room. Larry, Curly, and Moe are men in the Costumery.
Janine is a woman in the Costumery.

Rule for writing a paragraph about a person (called the target) who wears a hat (called attire):

say "[The target] is here, looking stylish in [an attire]."

Rule for writing a paragraph about a hatless person (called the target):

say "[The target] mopes about, hatless."

A hat is a kind of thing. A hat is always wearable. Definition: a person is hatless if he is not the player and he does not wear a hat.

The indigo bowler, the polka-dotted fedora, the pink beret, and the scarlet cloche are hats.

When play begins:

now every hatless person wears a random hat.

But this selects one random hat and assigns it to each hatless person in turn -- so it will only wind up being worn by the last of them (since Inform knows that only one person can wear a hat at a time).

In this case, we do have to expand out our loop so that the game makes an explicit distribution:

"Hatless 3"

The Costumery is a room. Larry, Curly, and Moe are men in the Costumery.
Janine is a woman in the Costumery.

Rule for writing a paragraph about a person (called the target) who wears a hat (called attire):

say "[The target] is here, looking stylish in [an attire]."

Rule for writing a paragraph about a hatless person (called the target):

say "[The target] mopes about, hatless."

A hat is a kind of thing. A hat is always wearable. Definition: a person is hatless if he is not the player and he does not wear a hat.

The indigo bowler, the polka-dotted fedora, the pink beret, and the scarlet cloche are hats.

When play begins:

repeat with item running through hats:

now the item is worn by a random hatless person.

Each time Inform considers the instruction "now the item is worn by a random hatless person", there is one fewer such person to choose from -- so we can guarantee that the hats are distributed one per customer and that all hats are distributed.

Hatless 3 is only guaranteed to work because the number of hats is less than or equal to the number of people; otherwise the final use of random will return "nothing" and then a problem message will appear during play.

210

★ Example Spellbreaker

WI

P. David Lebling's classic "Spellbreaker" (1986) includes a room where the game cannot be saved: here is an Inform implementation.

The answer is easy, but there is a trap:

Check saving the game when the location is the Vault: say "That spell does not work here." instead.

The trap is that "Before saving the game...", which might have been our first guess, does not work: because out of world actions are exempt from Before, Instead and After rules.

"Spellbreaker" pulls this unpleasant, but in context witty, stunt as part of a situation which is engineered to force the player to reason through a weighing-objects puzzle using the perfect strategy rather than by guesswork. The illusion that the situation is

fair - not rigged against the player, that is - would collapse if the player could save the game and keep retrying possibilities in the light of knowledge gained from earlier attempts. The moral of this story is that any attempt to use in-world situations to influence out-of-world commands should be *extremely uncommon*.

211



Example A point for never saving the game

WI

In some of the late 1970s "cave crawl" adventure games, an elaborate scoring system might still leave the player perplexed as to why an apparently perfect play-through resulted in a score which was still one point short of the supposed maximum. Why only 349 out of 350? The answer varied, but sometimes the last point was earned by never saving the game - in other words by playing it right through with nothing to guard against mistakes (except perhaps UNDO for the last command), and in one long session.

Here is one way to score this point with Inform:

Check saving the game for the first time: decrement the score.

That has the right effect, but it just isn't sneaky enough. Instead let us quietly keep track of how many times the player saves:

Check saving the game: increment the number of saves.
When play ends: if the number of saves is 0, increment the score.

Sneakier, certainly, but now we could get the bonus even if the game ends earlier on, so finally:

When play ends: if the number of saves is 0 and the score is 349, increment the score.

111



Example Y ask Y?

WI

Noticing when the player seems to be at a loss, and recommending the use of hints.

Suppose we'd like to watch for signs that the player is floundering, and if we see them, recommend that he try the hints. There are probably more sophisticated diagnostics, but as a first cut, let's assume that a player who repeatedly reviews descriptions of objects he's already seen, looks around the room, and takes inventory, is at a loss for more productive activities. So then...

"Y ask Y?"

A thing can be examined or unexamined. A thing is usually unexamined. Carry out examining something: now the noun is examined.

Taking inventory is acting confused. Looking is acting confused. Examining an examined thing is acting confused.

After acting confused for the sixth turn:

say "(If you are feeling lost, try typing HELP for suggestions.)"

And now we write a scenario which will, alas, rather encourage even a deft and clueful player to play as though he were hopelessly confused:

The story headline is "or: Bad Author, No Biscuit".

The description of a thing is usually "Hm. [The item described] reminds you quite a lot of [a random visible thing which is not the item described]."

The Yurt is a room.

Food is a kind of thing. Food is always edible. In the Yurt are a yam and a dish of yakitori. The yam and the yakitori are food. The description of food is "Well, at least it's not [a random edible thing which is not the item described]."

In the Yurt is an animal called a yapok.

The player wears a yukata. The player carries a yataghan.

Every turn:

if a random chance of 1 in 2 succeeds and something is examined:

say "Your eye is attracted by some kind of surreptitious movement from [the random examined thing].";

otherwise if the player carries something and a random chance of 1 in 3 succeeds:

say "[The random thing carried by the player] tries to slip from your grasp."

Test me with "x yam / x yam / look / x yam / i / look / i / help / quit".

And finally a little dollop of perversity from a later chapter:

Check quitting the game:

say "You're sure? ";

if player consents, say "[line break]You were getting close to a breakthrough, you know.[line break]";

otherwise stop the action.

Understand "help" as a mistake ("You're doing fine! Just keep at what you're doing now.").

"Basic Help Menu" is an extension that uses tables to provide a menu of instructions. Suppose we wanted to include this menu in our own game, but add a few custom menu items of our own:

"Food Network Interactive"

Include Basic Screen Effects by Emily Short. Include Menus by Emily Short.
 Include Basic Help Menu by Emily Short.

Table of Basic Help Options (continued)

title	subtable	description	toggle
"Recipes in This Game"	Table of Recipes	--	--
"Contacting the Author"	--	"If you have any difficulties with [story title], please contact me at fakeaddress@gmail.com."	--

This table is one that is pre-defined by the extension. By continuing it, we add a few additional items to the list.

And since we've promised a sub-menu of recipes:

Table of Recipes

title	subtable	description	toggle
"Salmon Tartare"	--	"First, be sure to buy extremely fresh salmon. Raw fish should be served on the day it was caught, if possible. To guarantee this, visit an Asian supermarket or specialty store, and buy salmon marked 'sashimi grade'..."	--
"Pecan Brownies"	--	"Begin by shelling half a pound of pecans..."	--

Whole Foods is a room.

To test it, type HELP and then experiment.

Example Ish.

A (very) simple HELP command, using tokens to accept and interpret the player's text whatever it might be.

"Ish."

Ichiro's Dubious Sushi Hut is a room. "Despite the allure of the dusty plastic sushi models in the window, you're beginning to have second thoughts about the selection of this particular restaurant for your rendezvous with Agent Fowler. There are no other patrons, for one thing. Afternoon sunlight filters lazily through the window and illuminates a number of empty glass-topped tables, at each of which is a chopstick dispenser (in form of cute ceramic cat) and a pitcher of soy sauce (sticky).

The sushi bar itself is what gives the most pause, however. Behind it sits an angry-looking Japanese woman, aggressively eating a Quarter Pounder with

Cheese."

We can, when necessary, accept any text at all as a token:

Understand "help [text]" or "help about [text]" as getting help about. Understand the commands "instructions" or "hint" or "hints" or "menu" or "info" or "about" as "help".

Getting help about is an action applying to one topic.

After that, we can use "the topic understood" to refer to the text we read:

Carry out getting help about:

if the topic understood is a topic listed in the Table of Standard Help:

say "[explanation entry][paragraph break]";

otherwise:

say "You're out of ideas."

Table of Standard Help

topic	title	summary	explanation
"sushi"	"sushi"	"Really it's just vinegary rice"	"Popular misconception says that sushi inevitably entails raw fish, but it is in fact just rice with rice vinegar on it. It's just that the really good kinds have raw fish in."
"cucumber roll" or "cucumber"	"Cucumber roll"	"Sushi for people who are afraid of sushi"	"It is just rice and slivers of cucumber in the middle, and as long as you don't go too crazy with the wasabi, all should be well."
"california roll" or "california"	"California roll"	"Travesty of the sushi concept"	"It's. Fake. Crab."
"monkfish liver"	"monkfish liver"	"Expert eaters only"	"The odds of Ichiro's having this unusual delicacy is near zero."
"microdot"	"microdot"	"What you came here to deliver"	"There'll be time enough for that later. If Fowler ever turns up. Where is she, anyway?"

Since the player may not know what all the help options are, we might as well let him get an overview, as well.

Understand "help" as summoning help. Summoning help is an action applying to nothing.

Carry out summoning help:

say "Help is available about the following topics. Typing HELP followed by the name of a topic will give further information.[paragraph break]";

repeat through the Table of Standard Help:

say " [title entry]: [summary entry][line break]".

Test me with "help / help about microdot / help cucumber / help california roll".

First, we create a single "[query]" token so that we can capture all instances of such sentences in a single line:

```
"Query"
```

```
Blank Room is a room.
```

```
Understand "who" or "what" or "when" or "where" or "why" or "how" or "who's" or "what's" or "when's" or "where's" or "why's" or "how's" as "[query]".
```

```
Understand "[query] [text]" as a mistake ("[story title] understands commands, such as '[command prompt]examine [a random thing that can be seen by the player]', but not questions. For more instructions, type HELP.").
```

```
Test me with "who am I? / who are you? / where is this place?".
```

Now the game will respond to all questions novice players might type with this reminder to look for help information.

285



Example Trieste

WI

Table amendment to adjust HELP commands provided for the player.

Suppose we are using an extension in which another author has defined some help topics for the player, and we want to amend them for our game.

We'll start with the portion of the text that we have inherited from the extension:

```
"Trieste"
```

```
Section 1 - Procedure
```

```
A help-topic is a kind of value. Some help-topics are defined by the Table of Standard Instructions.
```

```
Table of Standard Instructions
```

```
help-topic reply
commands "This game recognizes 150 common commands for forms of military attack. These include..."
saving      "To save the game, type SAVE. You will be prompted to supply a file-name for your saved game. If
            you'd like to return to play at that point again later, RESTORE the saved game."
```

```
Understand "help [help-topic]" as asking for help about. Asking for help about is an action out of world, applying to one help-topic.
```

```
Understand "help" or "help [text]" as a mistake ("Help is available on the following topics: [help-topics list]").
```

```
To say help-topics list:
```

```
repeat through the Table of Standard Instructions:
```

```
say "[line break] [help-topic entry]";
```

Carry out asking for help about:
repeat through the Table of Standard Instructions:
if the help-topic understood is the help-topic entry:
say "[reply entry][paragraph break]";
break.

Section 2 - Scenario

Now, let's imagine our game is a special one in which only a very limited supply of moves are allowed. In that case, we'll want to replace the information on commands:

Table of Standard Instructions (amended)

help-topic reply
commands "The only commands this game recognizes are HOLD, MOVE, CONVOY, SUPPORT MOVE, and SUPPORT HOLD. No others are necessary."

Board Room is a room. Mark is a man in the Board Room. "Russia (played by Mark) is also hovering over the board."

Guest Bathroom is south of Board Room. Lena and Rob are in the Guest Bathroom. Lena is a woman. Rob is a man.

Rule for writing a paragraph about Lena when Lena is in the Guest Bathroom and Rob is in the Guest Bathroom:

say "[Lena] (Italy) and [Rob] (Great Britain) are having a hushed conversation while leaning against your good towels. They stop and stare at you when you come in."

Test me with "help / help commands / help saving".

400



Example Solitude

WI

Novice mode that prefaces every prompt with a list of possible commands the player could try, and highlights every important word used, to alert players to interactive items in the scenery.

Observation of novice IF players suggests that they often have a hard time figuring out how to get started, especially if they are encountering the game in a context where they don't have time to settle in and read instructions. Here we provide some training wheels to help them learn to communicate.

This is divided into several parts. The first part is the system of rules for general guidance, which could be excerpted and used anywhere. The second part is a scenario using these rules.

"Solitude"

Part 1 - General Rules

When play begins:

say "Have you played interactive fiction before? >";
now novice mode is whether or not the player consents.

The rationale for asking the question this way, and not another, is that novices asked whether they would like instructions very often say no, even if they need them.

Novice mode is a truth state that varies. Novice mode is true.

Stopping novice mode is an action out of world.
Starting novice mode is an action out of world.

Understand "novice mode off" or "novice off" as stopping novice mode.
Understand "novice mode on" or "novice on" as starting novice mode.

Carry out stopping novice mode: now novice mode is false.
Carry out starting novice mode: now novice mode is true.

Report stopping novice mode: say "Novice mode is now off."
Report starting novice mode: say "Novice mode is now on."

Before reading a command when novice mode is true:
say "[line break]Some options to try:[line break]";
follow the novice suggestion rules.

The novice suggestion rules is a rulebook.

A novice suggestion rule (this is the suggestion that he look rule):
if not looking and not going, say " [bold type]look[roman type]".

A novice suggestion rule (this is the suggestion that he check inventory rule):
if the player carries something and we are not taking inventory, say " [bold type]inventory[roman type] (I)".

A novice suggestion rule (this is the suggestion that he put things on rule):
if the player carries something and a free-standing supporter is relevant, say " [bold type]put[roman type] something [bold type]on[roman type] [the list of relevant supporters]".

A novice suggestion rule (this is the suggestion that he take things rule):
if a gettable thing is relevant, say " [bold type]take[roman type] [the list of gettable relevant things]".

A novice suggestion rule (this is the suggestion that he examine things rule):
if an unexamined thing is relevant, say " [bold type]examine[roman type] (X) [the list of unexamined relevant things]".

A novice suggestion rule (this is the suggestion that he enter things rule):
if a relevant thing is worth entering, say " [bold type]enter[roman type] [the list of worth entering relevant things], or [bold type]get out[roman type]".

A novice suggestion rule (this is the suggestion that he open things rule):
if an unlocked openable thing is relevant, say " [bold type]open[roman type] or [bold type]close[roman type] [the list of unlocked openable relevant things]".

A novice suggestion rule (this is the suggestion that he lock things rule):
if a closed lockable thing is relevant, say " [bold type]lock[roman type] or [bold type]unlock[roman type] [the list of closed lockable relevant things]".

A novice suggestion rule (this is the suggestion that he eat things rule):
if the player carries an edible relevant thing, say " [bold type]eat[roman type] [the list of edible relevant things carried by the player]".

A novice suggestion rule (this is the suggestion that he wear things rule):
if the player carries a wearable relevant thing, say " [bold type]wear[roman type] [the list of wearable relevant things carried by the player]".

A novice suggestion rule (this is the suggestion that he turn things on rule):
if a device is relevant, say " [bold type]turn on[roman type] or [bold type]turn off[roman type] [the list of relevant devices]".

A novice suggestion rule (this is the suggestion that he go places rule):
if a room is adjacent, say " [bold type]go[roman type][exit list][if in darkness] or try other directions in the dark[otherwise]".

A novice suggestion rule (this is the suggestion that he enter doors rule):
if an open door is relevant, say " [bold type]go through[roman type] [the list of relevant open doors]".

A novice suggestion rule (this is the suggestion that he interact with people rule):
if another person is relevant, say " [bold type]kiss[roman type] or [bold type]wake[roman type] [the list of relevant other people][if the player carries something], or [bold type]give[roman type] things [bold type]to[roman type] someone[end if]".

A novice suggestion rule (this is the suggestion that he ask for help rule):
say " [bold type]help[roman type] to see a more complete set of instructions".

A novice suggestion rule (this is the suggestion that he turn off help rule):
say " [bold type]novice mode off[roman type] to turn off this guidance".

Last novice suggestion rule:
say "[line break]".

The suggestion about asking for help is no good unless we provide some. This might take any of a number of forms, but for the sake of example we'll use an easy way out:

Include Basic Screen Effects by Emily Short. Include Menus by Emily Short.
Include Basic Help Menu by Emily Short.

After taking inventory when novice mode is true: say "To get rid of any of these objects, [bold type]drop[roman type] it."

A thing can be examined or unexamined. Carry out examining something: now the noun is examined.

A thing can be seen or unseen. A thing is usually unseen.

Definition: a thing is relevant if it is seen and it is visible. Before printing the name of something (called the target): now the target is seen; if novice mode is true, say "[bold type]". After printing the name of something: say "[roman type]".

Definition: a supporter is worth entering:
if the player carries it, no;
if it is enterable, yes.

Definition: a container is worth entering:
if the player carries it, no;
if it is enterable and it is open, yes.

Definition: a person is other if it is not the player. Definition: a person is another if it is other.

Definition: a thing is free-standing if it is in a room.

To say exit list:
let place be location;
let count be 0;
repeat with way running through directions:
let place be the room way from the location;
if place is a room:
increment count;
say "[if count is greater than 1] or[end if] [bold type][way][roman type]".

Definition: a thing is gettable:
if it is scenery, no;
if it is fixed in place, no;
if it is a person, no;
if the player is carrying it, no;
if the player is wearing it, no;
yes.

Part 2 - On the Ground

Antarctic Research Station is a room. "Though not always the most stimulating of environments, the station is far from your ex-wife and most of the things in the world that annoy you, namely the other 6+ billion people. There is a second room to the south." The station contains a radio. The radio is a device. It is fixed in place.

South of the Station is Sitting Room. The description of the Sitting Room is "Just big enough for a very [comfortable chair]." The Sitting Room contains an enterable supporter called a comfortable chair. The chair is scenery. A monograph about penguins is in the Sitting Room.

Blistering Cold is a room. "It is white out here and very very very cold." The white door is a door. "[The white door] leads to [the other side of the white door]." It is west of the Blistering Cold and east of the Antarctic Research Station.

Test me with "i / x radio / x door / s / i / x chair / x monograph / sit in chair / get up / n / open door / enter door".



Example Real Adventurers Need No Help

Allowing the player to turn off all access to hints for the duration of a game, in order to avoid the temptation to rely on them overmuch.

Suppose we have an action called "asking for help" that gives the player some hints on request. We've also made it possible to turn this feature off, if the player would like to discourage himself from using the hints too much. Now we need a value that varies to keep track of whether hints are currently permitted or currently not permitted. So we might write:

"Real Adventurers Need No Help"

A permission is a kind of value. The permissions are allowed and denied.

Hint usage is a permission that varies. Hint usage is allowed.

And under the right circumstances, we change hint usage to denied:

Check asking for help for the first time:

say "Sometimes the temptation to rely on hints becomes overwhelming, and you may prefer to turn off hints now. If you do so, your further requests for guidance will be unavailing. Turn off hints? >";

if player consents:

now hint usage is denied;

say "[line break]Truly, a real adventurer does not need hints." instead.

Then we can refer back to this value later to decide whether we want to display the hint menu or not:

Check asking for help:

if hint usage is denied, say "You have chosen to eschew hints in this game. Be strong! Persevere!" instead.

Asking for help is an action out of world. Understand "help" or "hint" or "hints" as asking for help.

The Realm of Terribly Unjust Puzzles is a room.

Carry out asking for help:

say "Fine, since you're weak enough to ask: here is a complete walkthrough:
GET EGG. PEEL EGG. SMELL EGG. DIVIDE YOLK INTO THREE PORTIONS.
GIVE THE SMALLEST PORTION OF YOLK TO THE GOLDEN GOOSE. ASK
THE GOOSE ABOUT WHETHER THE SWAN IS TO BE TRUSTED. GIVE THE
LARGEST PORTION OF YOLK TO THE SWAN. DANCE CONGA. EAT
MEDIUM PORTION. STAND ON HEAD. WEST."

Test me with "hint".

Note that it would probably be kinder to offer the player some intermediate level of help, in the actual event.



Example The Unexamined Life

An adaptive hint system that tracks what the player needs to have seen or to possess in order to solve a given puzzle, and doles out suggestions accordingly. Handles changes in the game state with remarkable flexibility, and allows the player to decide how explicit a nudge he wants at any given moment.

Hint systems in IF come in a variety of flavors: some are a static, prewritten set of guidelines (which might exist in a menu or outside the game entirely); others are built in as part of the program, and attempt to adapt to the situation the player currently faces. Adaptive hints have the advantage that they are less likely to reveal information for which the player is not ready, and the disadvantage that they are more work for the author.

The exercise here is to write an adaptive hint system that will both respond in agile ways to the state of the world model and require a minimum of authorial fussing. We also want the player to be able to ask for a hint about any object he encounters in the game world: this will let him be specific and avoid accidentally receiving hints about the wrong puzzles.

Our baseline assumption is that a player may find a puzzle unsolvable for one of two reasons: he either hasn't seen the relevant clue, or he hasn't got the relevant equipment. If these are true, then he should be given hints about how to find this information, and then once he has it, more specific hints about the puzzle itself -- ending, as a last resort, with the exact command(s) he will need to use in order to bring about the solution.

In practice, there are other possibilities, but this will do for an example.

We begin by defining our relations:

"The Unexamined Life"

Use scoring.

Explaining relates one thing to various things. The verb to explain means the explaining relation.

Instead of hinting about something when something unexamined (called the clue) explains the noun:

say "You're still missing some information that might be useful to understanding the problem. [More]";
if player consents, try hinting about the clue.

Requiring relates one thing to various things. The verb to require means the requiring relation.

Instead of hinting about something when the noun requires something (called the implement) which is not carried by the player:

say "You're missing an object that might be useful to resolving this problem.

[More]";
if player consents, try hinting about the implement.

Hinting about is an action applying to one visible thing. Understand "hint about [any thing]" as hinting about.

This allows us to create the most absolutely generic sort of hint -- boring, perhaps, but in practice the player often just needs a nudge about what part of the game world he should be examining for a solution:

Carry out hinting about:
if something explains the noun, say "You might want to review [the list of things which explain the noun]. ";
if the noun requires something:
say "You should be sure that you have [the list of things required by the noun]. ";
otherwise:
say "Sorry, I can't advise you further on that.".

These things cover hinting about objects that are themselves puzzles. But what if the player asks for hints about a tool or piece of information because he doesn't know how to apply it yet? We might want to give some guidance there, as well.

Carry out hinting about something which explains something (called target):
if target is unseen, say "[The noun] might prove useful information, sooner or later." instead;
otherwise say "You could examine [the noun]." instead.

Carry out hinting about something which is required by something:
say "[The noun] might be useful to have. [More]";
if player consents:
if a seen thing requires the noun, say "[The noun] may help with [the list of seen things which require the noun]." instead;
otherwise say "There are [number of things which require the noun in words] problems for which [the noun] might come in handy." instead.

Now we have these general hints written, but we want to pre-empt them if the player has not yet fulfilled all the prerequisites.

Instead of hinting about something unseen:
if the noun is visible:
now the noun is seen;
continue the action;
say "Perhaps you should explore further. ";
if the ultimate location of the noun is an unvisited room:
try hinting about the ultimate location of the noun;
otherwise:
if the ultimate location of the noun is the location:
say "You're in the correct room right now[if the visible shell of the noun is a thing]. Try further exploring [the visible shell of the noun][end if].";
otherwise:
try hinting about the ultimate location of the noun.

Instead of hinting about a visited room:

- say "There's a room you've visited, but you haven't exhausted all there is to see there. [More]";
- if player consents:
 - say "Try going back to [the noun]. [More]";
 - if player consents, direct player to the noun.

Instead of hinting about an unvisited room:

- say "There's a room you haven't yet visited. [More]";
- if player consents, direct player to the noun.

To direct player to (goal - a room):

- let way be the best route from location to the goal, using even locked doors;
- if way is a direction, say "Try going [way] to start your explorations.";
- otherwise say "Sorry, the route is an indirect one."

Instead of hinting about a portable seen thing which is not visible:

- if the noun is scenery, continue the action;
- say "You have seen the item you need to solve this problem, but it's not in sight at the moment. [More]";
- if player consents:
 - try hinting about the ultimate location of the noun.

And this business of "seen" things requires, of course, that we keep track:

- A thing can be seen or unseen. A thing is usually unseen. The player is seen.
- After printing the name of something (called target): now the target is seen.

That "After printing..." rule means that as soon as the game automatically prints the name of an object, it tags that object as having been "seen" by the player. This requires just a little care on our part, that we never mention an object without using the game's printing rules. Still, it is much easier than most other possible forms of bookkeeping.

We also need to deal with the question of whether the player has examined an object, for those objects whose descriptions carry vital information:

- A thing can be examined or unexamined. A thing is usually unexamined. Carry out examining something: now the noun is examined.

In practice, there might be other ways of getting vital facts, and in a more sophisticated puzzle game we might need a more sophisticated model to track this. But examined or unexamined will do for now.

To decide what room is the ultimate location of (item - a thing):

- let place be the holder of the item;
- while the place is a thing:
 - let the place be the holder of the place;
- if the place is a room, decide on the place.

To decide what thing is the visible shell of (item - a thing):

- if item is visible, decide on the item;
- let place be the holder of the item;
- while place is a thing and place is not visible:

let place be the holder of the place;
if the place is visible, decide on the place.

To say more:
say "[paragraph break]Shall I go on? > ".

That covers most of the generic hints, but let's also add some slightly more precise hints about a few kinds of objects that are especially important in the model world. These hints will probably not be very interesting to a seasoned IF veteran, but a novice player who does not know the wording or cannot guess what something might be for may still find them useful:

Carry out hinting about a locked lockable thing:
say "You could unlock [the noun] with [the matching key of the noun]." instead.

Instead of hinting about a locked thing when the matching key of the noun is not carried by the player:

if the player can see the matching key of the noun:
say "Perhaps [the matching key of the noun] would help.";
otherwise:
say "[The noun] is locked. There must be a key around somewhere.
[More]";
if player consents, try hinting about the matching key of the noun.

Carry out hinting about a closed openable unlocked thing:
say "You could open [the noun]." instead.

Carry out hinting about an open door:
say "You could enter [the noun]." instead.

Carry out hinting about an unexamined thing:
say "You might find out something if you examine [the noun]." instead.

Carry out hinting about an edible thing:
say "You could eat [the noun]." instead.

Carry out hinting about a wearable thing:
say "You could wear [the noun]." instead.

Carry out hinting about a pushable between rooms thing:
say "You could push [the noun] some direction." instead.

Now to the actual objects in the game:

The Crypt is a room. "This squat, barrel-vaulted chamber runs roughly north-south. Along either side are the graves of Saxon kings and early bishops of the church long since gone to dust -- one [tomb] in particular looks undisturbed."

Notice that we used the bracketed tomb here: the tomb is scenery, and if we do not use the name-printing function, Inform will not register that we have mentioned it to the player.

The tomb is scenery in the Crypt. The tomb is openable and closed. The silver dagger is a thing in the tomb. Understand "tombs" as the tomb. The description of the silver dagger is "Gleaming in a soft light all its own. Its blade is figured with running deer and its hilt is made of horn." The wight requires the silver dagger. The tomb requires the pry bar.

Instead of opening the tomb when the player does not carry the pry bar:
say "The lids are stone, too heavy for you to raise without some implement."

Now we can add specific hints to replace the generic ones:

Carry out hinting about the tomb:
say "The lids are heavy, but you can open them when you carry the pry bar."

The rest of the hint system ensures that the player will not see this final suggestion until he has the pry bar, since the tomb "requires" the pry bar. Having the hint there doesn't excuse us from providing some alternate wording in case the player solves this not-very-difficult conundrum on his own, though:

Understand "pry [something] with [something preferably held]" as unlocking it with. Understand the commands "lever" or "prise" as "pry".

Instead of unlocking something with the pry bar, try opening the noun.

The wight is a man in the Crypt. "[The wight] lurks near the south exit." The description of wight is "Old English [*wiht*]: a thing, a creature. It is little more than the memory of a life ill-lived, but it lingers here." Understand "wiht" or "creature" or "ghost" as the wight.

Instead of going south in the presence of wight:
say "The wight breathes chill into your face."

Your head swims, and you are aware that you no longer have the willpower to go in that direction."

Fresh Air is south from the Crypt.

After going to Fresh Air:
increment the score;
say "Congratulations, you have escaped!";
end the story finally.

The inscription is fixed in place in the Crypt. "Someone has painstakingly carved [an inscription] into the wall above the door." The description is "Squinting, you decipher the Latin text: [*Silver causes harm to those that live though dead*]." The inscription explains wight.

The Treasure Chamber is north of the Crypt. "The walls are thick, the high windows promisingly barred with iron. But for all this there is no hint of any valuable stores remaining."

The pry bar is in the Treasure Chamber. "One of the window bars, rusted from its place, lies in a puddle of water." Understand "window" or "bars" as the pry bar.

The description of the pry bar is "A few feet long, and not entirely rusted into uselessness yet."

Instead of giving the dagger to wight:
say "The wight recoils, appalled."

Carry out hinting about wight:
say "You will have to find some way to get wight to come in physical contact with the silver dagger, which he will certainly not do willingly. [More]";
if player consents, say "You could, for instance, throw it at him." instead;
otherwise stop the action.

Understand "touch [something] with [something]" as putting it on (with nouns reversed). Understand "hit [someone] with [something]" as putting it on (with nouns reversed).

Instead of attacking the wight:
say "You can't force yourself to approach close enough for hand to hand combat: if, indeed, the wight has hands."

Instead of putting the dagger on wight:
say "The wight fades out of your way without ever coming into contact with the dagger. Perhaps a more projectile method would work better."

Instead of putting something on wight:
say "The wight dodges you."

Instead of throwing the dagger at wight:
now the wight is nowhere;
move the dagger to the location;
increment the score;
say "The dagger passes through its airy form with a rending like the rip of silk. The fragments dissipate at once."

The maximum score is 2.

Test me with "hint about wight / north / get bar / south / open tomb / get dagger / south / hint about wight / read inscription / hint about wight / attack wight / throw dagger at wight / south".

Note that, if using TEST ME to run through the solution on the Z-machine, we will have to answer a few yes/no questions along the way.

For Glulx, the code should instead read something like

Test me with "hint about wight / y / north / get bar / south / open tomb / get dagger / south / hint about wight / y / read inscription / hint about wight / y / attack wight / throw dagger at wight / south".

We could, if we wanted, make a table of stored actions all of which represent things that will earn points for the player. For instance:

"Bosch"

Use scoring.

The Garden of Excess is a room. The gilded lily is an edible thing in the Garden of Excess.

The Pathway to Desire is west of the Garden of Excess. The emerald leaf is in the Pathway.

Table of Valuable Actions

relevant action	point value	turn stamp
taking the emerald leaf	15	-1
eating the gilded lily	5	-1

(And our list would presumably continue from there, in the full game.)

The maximum score is 25.

After doing something:

- repeat through Table of Valuable Actions:
 - if the current action is the relevant action entry and turn stamp entry is less than 0:
 - now the turn stamp entry is the turn count;
 - increase the score by the point value entry;
 - continue the action.

Understand "full score" or "full" as requesting the complete score. Requesting the complete score is an action out of world.

Check requesting the complete score:

- if the score is 0, say "You have not yet achieved anything of note." instead.

Carry out requesting the complete score:

- say "So far you have received points for the following: [line break]";
- sort the Table of Valuable Actions in turn stamp order;
- repeat through the Table of Valuable Actions:
 - if the turn stamp entry is greater than 0:
 - say "[line break] [relevant action entry]: [point value entry] points";
- say line break.

Test me with "eat lily / w / full score / get leaf / full".

This system is tidy, but limited: we cannot give actions interesting names in the score list, like "seducing the pirate's daughter" or "collecting a valuable artifact". So it will not be ideal in all situations, but it has the virtue of being easy to extend, and of listing all of the player's successes in the order in which they occurred in his play-through.

**Example Mutt's Adventure**

Awarding points for visiting a room for the first time.

Suppose we want to reward the player the first time he reaches a given room. The "unvisited" attribute is useful for this: unlike such constructions as "going to a room for the first time", it doesn't develop false positives when the player has merely tried to go to the room in question. "Every turn when the player is in a room for the first time" is also unhelpful, because it continues to be true as long as the player is in a room on his first visit there.

"Mutt's Adventure"

Use scoring.

Section 1 - Procedure

A room can be scored or unscored.

Carry out going to a unvisited scored room:
increment the score.

Section 2 - Scenario

The Incan Palace Compound is a room. "After numerous false leads through the jungles of Peru, and an arduous trek along the Amazon, you have arrived, at last, here: at Atagon, the lost city of untold treasure."

The startlingly intricate door is a door. It is inside from Incan Palace Compound and outside from the Treasure Room. "A door carved all over with figures of ancient gods, and protected by an assortment of gears and latches, [if open]stands open[otherwise]blocks progress[end if] towards [the other side of the intricate door]."

The description of the Treasure Room is "To your considerable surprise, the treasure room is stocked with art objects from a vast range of eras and geographical locations: beside the expected pre-Columbian gold there are Cycladic figurines, Chinese Tang-dynasty pottery, purses that might have been stolen from Sutton Hoo. [one of]If the British Museum developed a nasty expectorant cough, this is what you'd find in its hanky.[or][stopping]".

The Treasure Room is scored.

Test me with "in / out / in".

**Example No Place Like Home**

Recording a whole table of scores for specific treasures.

Suppose we want to assign scores for a whole range of objects the player might pick up. One systematic way to do this would be with a table of point values for things:

"No Place Like Home"

Use scoring.

The Hall of the Gnome King is a room. The emerald cow is a thing in the Hall of the Gnome King. The ivory chessman is a thing in the Hall of the Gnome King. The book of incantations is a thing in the Hall of the Gnome King.

Table of Point Values

item	score
cow	10
incantations	4
chessman	1

Report taking an item listed in the Table of Point Values:
increase the score by the score entry;
blank out the whole row.

Test me with "take all".

"Blank out the whole row" removes the line from the table, so that each award will occur only once. The player will not be able to earn more and more points by dropping and taking the same item again.

166



Example Panache

WI

Replacing the score with a plot summary that records the events of the plot, scene by scene.

If we have a plot that branches and has multiple kinds of outcome, we might well want to assemble these into a plot summary in place of the more traditional score. One way to approach this is to build the scene information into a table, adding information when each scene ends.

We begin with a bit of setup:

"Panache"

The player is in a room called Beneath Roxane's Balcony. Christian is a man in the Balcony. "Christian stands in a spot of moonlight and tries to avoid too obviously glancing at the shadows that conceal you." The description of Christian is "Like you, Christian loves Roxane. Unlike you, he is handsome enough to receive her favor in return. He is the beauty to your brain."

Roxane is a woman in the Balcony. "Above you in the night is Roxane." Roxane can be wooed, skeptical, confused, or annoyed. Roxane is skeptical. The

description of Roxane is "The brightest, the most radiant of women -- and in love with an utter fool."

Empty Street is a room. "No one is about at this hour, all alone under a pale moon."

Telling someone about something is speech. Asking someone about something is speech. Answering someone that something is speech.

This next portion borrows from the Advanced Actions chapter to allow us to command Christian to do things:

A persuasion rule for asking Christian to try speech: persuasion succeeds.

Carry out Christian answering someone that something:
now Roxane is wooed;
say "[noun], [the topic understood]."

Carry out Christian answering the player that something:
say "Christian parrots your words back to you." instead.

Carry out Christian telling a skeptical Roxane about something:
now Roxane is confused;
say "Christian turns to [the noun]. 'I must tell you about [the topic understood],' he says, and comes to a halt, looking at you for further direction.

Perhaps you'd better give him exact lines to say. Surely he can't mess up an instruction like 'say hello to Roxane.'" instead.

Carry out Christian asking a skeptical Roxane about something:
now Roxane is confused;
say "'So,' says Christian nervously to [the noun]. 'Did you know about [the topic understood]?' But Roxane merely seems puzzled." instead.

Carry out Christian telling a confused Roxane about something:
now Roxane is annoyed;
say "Christian begins rambling on witlessly about [the topic understood]." instead.

Carry out Christian asking a confused Roxane about something:
now Roxane is annoyed;
say "Christian puts another confused question about [the topic understood]." instead.

And now we have enough material to begin writing the scenes:

Courting Roxane is a scene. Courting Roxane begins when play begins. Courting Roxane ends in success when Roxane is wooed. Courting Roxane ends in failure when Roxane is annoyed.

When Courting Roxane ends in success:
record "Seduction by Proxy" in the Table of Events;
say "Roxane, deeply moved by this sentiment, invites Christian up to her balcony. He scrambles up the ivy and disappears into her bedroom; the last

thing you hear is a girlish giggle from above.";
now Roxane is nowhere; now Christian is nowhere;
move the player to Empty Street.

When Courting Roxane ends in failure:
record "Ruining Christian's Chances" in the Table of Events;
say "Roxane sighs heavily and goes back into her room, slamming the door behind her.

'Thanks very much,' says Christian to you, striding off down the street.";
now Roxane is nowhere; now Christian is nowhere;
move the player to Empty Street.

Sulky Ramble is a scene. Sulky Ramble begins when Courting Roxane ends in success. Sulky Ramble ends when the time since Sulky Ramble began is 2 minutes. When Sulky Ramble ends: record "Wandering the Streets, Sulking" in the Table of Events.

Every turn during Sulky Ramble:
say "You find yourself kicking fenceposts quite without thinking about it."

Smug Ramble is a scene. Smug Ramble begins when Courting Roxane ends in failure. Smug Ramble ends when the time since Smug Ramble began is 2 minutes. When Smug Ramble ends: record "Wandering the Streets, Exultant" in the Table of Events; say "Of course, you will regret this soon enough."

Every turn during Smug Ramble:
say "You find yourself smiling fiercely at the moon."

To record (occurrence - text) in (target table - a table name):
choose a blank row in the target table;
now the event entry is the occurrence.

Table of Events
event
"A Duel of Insults"
with 30 blank rows.

The plot summary rule is listed instead of the announce the score rule in the carry out requesting the score rules.

This is the plot summary rule:
say "The Plot So Far: [paragraph break]";
let act number be 0;
repeat through the table of Events:
increment act number;
say " Act [act number]: [event entry][line break]".

Test me with "christian, ask roxane about love / christian, say your breath smells like ripe taleggio to roxane / score / z / z / score".



Implementing a FULL SCORE command which lists more information than the regular SCORE command, adding times and rankings, as an extension of the example given in this chapter.

Some games provide a FULL SCORE command that gives more information about the player's achievements than SCORE alone. Supposing we wanted to include a FULL SCORE in our game that gave the kind of score reading described in this chapter:

"Goat-Cheese and Sage Chicken"

Use scoring.

The story headline is "An interactive recipe"

Table of Tasks Achieved

Points	Citation	Time
3	"sauteeing onions"	a time
3	"reconstituting apricots"	
1	"flattening chicken"	
1	"unwrapping goat cheese"	

To record (T - text) as achieved:

choose row with a citation of T in the Table of Tasks Achieved;

if there is no time entry:

now time entry is the time of day;

increase the score by the points entry.

Requesting the full score is an action out of world. Understand "full" or "full score" as requesting the full score.

Carry out requesting the full score:

if the score is 0, say "You have achieved nothing towards supper." instead;

repeat through the Table of Tasks Achieved in reverse time order:

say "[time entry]: [citation entry] ([points entry])."

Table of Rankings

Score	Rank
0	"Rank Amateur"
2	"would-be Bobby Flay"
5	"Alton Brown"
8	"Julia Child"

The Kitchen is a room. The description of the Kitchen is "Equipped with many familiar friends: refrigerator, stove, oven; countertop; cabinet for pans and bowls, and a drawer for your tools."

The stove is scenery in the kitchen. It is a supporter. The oven is a container. It is part of the stove. It is closed and openable. The stove's switch is a device. It is switched on. It is part of the stove. The oven's dial is a device. It is switched off. It is part of the oven.

A thing can be heatproof.

Instead of putting something which is not heatproof on the stove when the stove's switch is switched on:

say "You catch yourself just at the last minute: not a good idea to put [the noun] directly on the stove while it's turned on."

Instead of switching on the stove, try switching on the stove's switch. Instead of switching off the stove, try switching off the stove's switch. Instead of switching on the oven, try switching on the oven's dial. Instead of switching off the oven, try switching off the oven's dial.

Before switching on the oven's dial when the oven is open:

say "(closing the oven so that it will heat properly)[command clarification break]";
try closing the oven.

The frying pan is a heatproof unopenable open container on the stove.

The cabinet is a closed openable container in the kitchen. It is scenery. It contains an open unopenable container called a mixing bowl. It contains a portable supporter called a platter. An open unopenable heatproof container called a Calphalon baking dish is in the cabinet. The baking dish has the description "One of those marvelous pieces of kitchen equipment which goes on the stove or in the oven, as you will. The chief thing is never ever to touch it when it is hot, since the handles are metal and the heat retention excellent."

The counter is a supporter in the kitchen. It is scenery. The kettle is a heatproof openable closed container on the counter. Some water is in the kettle.

The water can be cool, warm, or boiling. The printed name of the water is "[water condition] water".

The refrigerator is a closed openable container in the kitchen. It is scenery. Understand "fridge" as the refrigerator.

An ingredient is a kind of thing.

Some onions, some apricots, and some sage are ingredients on the counter. A chicken breast, an egg, and goat cheese are ingredients in the refrigerator.

The goat cheese can be wrapped, snipped open, or unwrapped. The printed name of the goat cheese is "[goat cheese condition] goat cheese".

The sage can be unwashed, clean, or julienned. The sage is unwashed. The printed name of the sage is "[sage condition] sage".

The apricots can be dried, reconstituted, or chopped. The apricots are dried. The printed name of the apricots is "[apricots condition] apricots".

The chicken breast can be whole, flattened, stuffed, rolled, coated, browned, or baked. The printed name of the chicken breast is "[chicken breast condition] chicken breast".

The onions can be unpeeled, peeled, diced, sauteed, or burnt. [The printed name of the onions is "[onions condition] onions".]

The can of chicken broth is a closed container on the counter. The bottle of white cooking wine and the bottle of Thurston Wolfe PGV are a closed containers in the refrigerator.

The description of the Thurston Wolfe is "A Washington State Pinot Gris-Viognier, 2003. It is said to have 'peach aromas', and, startlingly, the untutored person can detect these without resorting to fantasy.

(It is also supposed to possess a delicate perfume and a moderate body; the label author at least stopped short of 'good sense of humor and likes long walks on the beach')."

Understand the commands "wash" and "rinse" as "clean".

Instead of rubbing the unwashed sage:
now the sage is clean;
say "You rinse off the sage. There -- ready to slice."

Instead of cutting the sage:
say "You'd need to have a knife in hand, first."

Instead of cutting the clean sage when the player is carrying the butcher knife:
now the sage is julienned;
say "You slice the sage into thin strips."

Instead of cutting the unwashed sage:
say "It came from the garden, so it won't have any strange chemicals on it, but you should still give it a rinse for dirt and bugs and so on before using it."

Instead of doing something other than examining or rubbing with the unwashed sage:
say "It needs to be washed off."

Understand "peel [something]" as peeling.

Peeling is an action applying to one thing.

Instead of peeling the unpeeled onions:
now the onions are peeled;
say "You tear away the shining outer skin of the onions, leaving them pale and nekkid. Poor things."

Instead of cutting the diced onions:
say "That seems unnecessary now."

Instead of cutting the sauteed onions:
say "Too late; you're well past that stage."

Instead of cutting the burnt onions:
say "There's no rescuing 'em -- the carbon isn't going to flake off, you know."

Instead of cutting the unpeeled onions:
say "It would help to peel them first."

Instead of cutting the peeled onions:
say "You'd need to have a knife in hand, first."

Instead of cutting the peeled onions when the player is carrying the butcher knife:
now the onions are diced;
say "You dice the onions neatly. Your own skill brings tears to your eyes."

Instead of opening the goat cheese:
try peeling the goat cheese instead.

Instead of peeling the unwrapped goat cheese:
say "The goat cheese is already unwrapped. (Stay focused, stay focused...)"

Before peeling the wrapped goat cheese when the shears are held by the player:
try cutting the goat cheese.

Instead of peeling the snipped open goat cheese:
now the goat cheese is unwrapped;
record "unwrapping goat cheese" as achieved;
say "Ah, success. The goat cheese is now free of its packet."

Instead of peeling the wrapped goat cheese:
say "It would help to have a pair of scissors or something -- the packet resists being torn."

Instead of cutting the goat cheese:
say "No need, at this point."

Before cutting the wrapped goat cheese when the shears are not held by the player and the shears are visible:
say "(first picking up the shears)[command clarification break]";
try taking the shears.

Instead of cutting the wrapped goat cheese:
say "Something to cut with would be useful."

Instead of cutting the wrapped goat cheese when the shears are held by the player:
now the goat cheese is snipped open;
say "You neatly snip through the packaging with the shears."

Instead of examining the whole chicken breast:
say "It is still entire and has yet to be pounded flat."

Instead of examining the flattened chicken breast:
say "It has been hammered to a thickness of about a half inch. (The recipe said a quarter inch but you're pretty sure it was joking. You have never been able to achieve a quarter inch.)"

Instead of attacking the whole chicken breast:
say "You need something heavy enough to flatten it with."

Instead of attacking the whole chicken breast when the player is holding the wooden mallet:
now the chicken breast is flattened;
record "flattening chicken" as achieved;
say "You hammer away at the chicken breast, turning all your aggressions into culinary goodness. Several minutes pass. When you are done you have a broad flat chickeny pancake suitable for wrapping about a stuffing."

Before printing the name of onions:
say "[onions condition]".

The drawer is an openable closed container. It is part of the counter.

A tool is a kind of thing. A spatula, a spoon, a wooden mallet, some shears, and a ball of twine are tools in the drawer. A butcher knife is a tool carried by the player. Understand "scissors" as the shears.

Instead of burning something:
say "You'll have to do that the hard way."

Some steam is fixed in place. "Dense clouds of steam fill the room."

Some smoke is fixed in place. "Smoke is beginning to collect near the ceiling."

Sauteeing Onions is a scene. Sauteeing Onions begins when the diced onions are in a hot container.

Definition: a container is hot if it is on the stove and the stove's switch is switched on.

Instead of touching the hot pan:
say "Ow!"

Scorching Onions is a scene.

Preheating the Oven is a scene. Preheating the Oven begins when the oven's dial is heating.

Definition: a oven's dial is heating if the oven's dial has been switched on for exactly one turn.

Preheating the Oven ends when the time since Preheating the Oven began is five minutes.

When Preheating the Oven begins:
say "The oven begins to warm up."

When Preheating the Oven ends:
say "The oven beeps to inform you that it has reached the desired hotness."

Every turn during Sauteeing Onions:
say "The onions sizzle in the pan."

Every turn during Scorching Onions:
say "The onions are past their prime and are getting blacker by the moment."

Every turn during Hearing the Kettle Whistle:
say "The kettle continues to whistle."

Instead of listening to during Hearing the Kettle Whistle:
say "The only thing you can really hear just at the moment is the kettle."

Instead of smelling the Kitchen during Sauteeing Onions:
try smelling the onions.

Instead of smelling the onions during Sauteeing Onions:
say "The onions smell marvelous."

Instead of opening the oven during Preheating the Oven:
say "It'll never heat if you open it up while it's warming."

Heating Kettle is a scene. Heating Kettle begins when the hot kettle contains cool water.

Before printing the name of the kettle when the kettle is hot:
say "hot "

When Heating Kettle begins:
say "The kettle begins to heat up."

Heating Kettle ends when the time since Heating Kettle began is 7 minutes.

Hearing the Kettle Whistle is a scene. Hearing the Kettle Whistle begins when Heating Kettle ends. Hearing the Kettle Whistle ends when the kettle is not hot.

When Hearing the Kettle Whistle begins:
now the water is boiling;
say "The kettle begins to burble and whistle shrilly."

When Hearing the Kettle Whistle ends:
say "The kettle's screaming dies off."

Idling is a scene. Idling begins when play begins. Idling ends when Sauteeing Onions begins.

Sauteeing Onions ends in disaster when Scorching Onions begins.

Sauteeing Onions ends in success when the onions are sauteed and onions are not in a hot container.

Definition: a thing is alone if it is in a container which contains exactly one thing.

Sauteeing Onions ends in mixture when the sauteed onions are not alone.

When Sauteeing Onions ends in mixture:
say "The mixture of things in [the holder of the onions] stops them cooking quite so fast."

When Sauteeing Onions ends in success:
say "Nice work with the onions."

Every turn:
if diced onions have been in a hot pan for ten turns:
say "The onions are starting to look ready.";
now the onions are sauteed.

Scorching Onions begins when Sauteeing Onions ends in disaster. Scorching Onions begins when the alone sauteed onions are in a hot container.

Scorching Onions ends horribly when the time since Scorching Onions began is three minutes. Scorching Onions ends in reprieve when the sauteed onions are not in a hot container. Scorching Onions ends in mixture when the sauteed onions are not alone.

When Scorching Onions ends in mixture:
record "sauteeing onions" as achieved;
say "The mixture of things in [the holder of the onions] stops them cooking quite so fast."

When Scorching Onions ends horribly:
move smoke to Kitchen;
now the onions are burnt.

When Scorching Onions ends in reprieve:
record "sauteeing onions" as achieved;
say "You've got the onions off heat before they can scorch -- a good sign."

Instead of taking the onions when the onions are in the pan: try taking the pan.

Instead of smelling in the presence of the smoke:
say "The scent of the late disaster lingers in the air."

Reconstituting the Apricots is a scene.

Reconstituting the Apricots begins when the dried apricots are in a container which contains boiling water.

When Reconstituting the Apricots begins:
say "The apricots slowly begin to plump up again."

Reconstituting the Apricots ends when the dried apricots are not in a container which contains boiling water.

Every turn:
if dried apricots have been in a container which contains boiling water for ten turns:
say "The apricots have turned plump(ish).";

now the apricots are reconstituted;
record "reconstituting apricots" as achieved.

Test sautee with "peel onions / cut onions / get onions / put onions in pan / get sage / wash sage / cut sage / wait / get pan".

Test apricots with "get kettle / open kettle / get apricots / put apricots in kettle / put kettle on stove / wait / wait".

Test chicken with "open refrigerator / get chicken / open drawer / get mallet / hit breast".

Test cheese with "get scissors / get cheese / unwrap cheese".

Test me with "full score / test sautee / full score / test apricots / full score / test chicken / full score / test cheese / full score".

And... at that point you're a lot less close to being done than you think. The filling -- onions, sage, apricot, and cheese -- must be assembled and put in the chicken breasts; these tied up in string; each roll dipped in egg yolk and rolled in panko crumbs; these arranged in the Calphalon pan and baked. Then later, the whole retrieved from the oven, and the breasts transferred to a plate while we deglaze the pan and concoct the sauce with the chicken broth, wine, butter, etc. Then the chicken is sliced and plated, and the sauce poured over top. Usually one also wants a side dish or two. A number of things can go interestingly wrong in this process, of course, and implementing it would require, among other things, an intelligent management of all the possible mixtures that result.

442



Example Rubies

WI

A scoreboard that keeps track of the ten highest-scoring players from one playthrough to the next, adding the player's name if he has done well enough.

The trick here is that we need a table with text in order to keep track of the players' names.

Part 1 largely replicates the source from "Identity Theft"; new material starts at Part 2.

"Rubies"

Use scoring.

Part 1 - Collecting Names

The player's forename is a text that varies. The player's full name is a text that varies.

When play begins:
now the command prompt is "What is your name? > ".

To decide whether collecting names:
if the command prompt is "What is your name? > ", yes;
no.

After reading a command when collecting names:
if the number of words in the player's command is greater than 5:
say "[paragraph break]Who are you, a member of the British royal family?
No one has that many names. Let's try this again.";
reject the player's command;
now the player's full name is the player's command;
now the player's forename is word number 1 in the player's command;
now the command prompt is ">";
say "Hi, [player's forename]!";
say "[banner text]";
move the player to the location;
reject the player's command.

Instead of looking when collecting names: do nothing.

Rule for printing the banner text when collecting names: do nothing.

Rule for constructing the status line when collecting names: do nothing.

Part 2 - Adding the Leaderboard

File of Leaderboard is called "leaderboard".

When play begins:
if the File of Leaderboard exists:
read File of Leaderboard into the Table of Leaders;
sort the Table of Leaders in reverse scored amount order.

When play ends:
choose row 10 in the Table of Leaders; [we've sorted the table, so the lowest score will be the one at the bottom]
if the score is greater than scored amount entry:
now name entry is the player's forename;
now the scored amount entry is the score;
show leaderboard;
write the File of Leaderboard from the Table of Leaders.

To show leaderboard:
sort the Table of Leaders in reverse scored amount order;
say "Current leading scores: [paragraph break]";
say fixed letter spacing;
repeat through Table of Leaders:
if scored amount entry is greater than 0:
say " [name entry]";
let N be 25 minus the number of characters in name entry; [here we want to space out the scores so they make a neat column]
if N is less than 1, now N is 1;
say N spaces;

say "[scored amount entry][line break]";
say variable letter spacing.

To say (N - a number) spaces:
repeat with index running from 1 to N:
say " ".

Table of Leaders

scored amount	name
0	"Smithee"

And now we introduce a scenario that allows different players to come up with different scores -- admittedly not a very interesting scenario, but it will do for now:

Part 3 - Scenario

Carry out taking something which is not handled:
increment score.

The Big Treasure Chamber is a room. It contains a ruby, an emerald, a gold tooth, an antique katana, and a silver coin.

Instead of waiting, end the story finally.

Test me with "get ruby / z".

383

★ Example Battle of Ridgefield

WI

Completely replacing the endgame text and stopping the game without giving the player a chance to restart or restore.

Occasionally, a piece of IF is sufficiently serious that it feels bathetic to offer the player the usual restore-restart-undo-quit options at the end. The following would replace "*** You have died ***" with a centered epitaph, then quit the game when the player hits a key.

This example relies on a standard extension to avoid any fancy programming:

"Battle of Ridgefield"

Include Basic Screen Effects by Emily Short.

Ridgefield is a room.

Instead of doing something when the turn count is greater than 1: say "Alas, you no longer have the strength."; end the story.

Rule for printing the player's obituary:

```
say paragraph break;
center "In defense of American Independence";
center "at the Battle of Ridgefield, April 27, 1777,";
center "died Eight Patriots who were laid in this ground,";
center "Companied by Sixteen British Soldiers,";
center "Living, their enemies,";
center "Dying, their guests";
say paragraph break;
wait for any key;
stop game abruptly;
rule succeeds.
```

384

★ Example Finality

WI

Not mentioning UNDO in the final set of options.

By default, Inform reminds the player that he has the option of typing UNDO after a story-ending action. This is generally good practice, especially for the sake of novice players who might not be aware of this possibility otherwise, and might be frustrated by a loss they could easily step back from.

Just occasionally, though, we may decide that the player does not deserve any such notification:

"Finality"

Cliff Edge is a room. "This narrow strip overlooks a gorge many hundreds of feet deep, at whose bottom is a river of molten lava. The walls of the gorge are lined with poison-tipped spikes. Furthermore, the birds that inhabit this valley spit balls of fire. Good thing you're safe up here."

The Table of Final Question Options determines what options are to be given to the player after the story ends. We can change what is mentioned there by altering the entries. (The example Jamaica 1688 explains this table in more detail, and demonstrates some other things that we might do with it.)

When play begins:

```
choose row with a final response rule of immediately undo rule in the Table of
Final Question Options;
blank out the final question wording entry.
```

Instead of jumping:

```
say "If you insist.";
end the story.
```

And if we decided that we didn't want the player to be able to undo the command at all, we should add the use option

Use undo prevention.

Test me with "jump".

385

★ Example Jamaica 1688

WI

Adding a feature to the final question after victory, so that the player can choose to reveal notes about items in the game.

The options offered to the player at the end of the game are listed in the Table of Final Question Options, which means that we can add to them simply by continuing the table; what's more, the table gives us the opportunity to create a "final response rule", a rule that the game should follow in order to parse the player's input at this point.

So, for instance, if we wanted the player to be allowed to ask for notes about any of the rooms, characters, or objects in a historical game:

"Jamaica 1688"

Use scoring.

Section 1 - Procedure

Table of Final Question Options (continued)

final question wording	only if victorious	topic	final response rule	final response activity
"REVEAL the inspiration for something or somewhere"	true	"reveal [any thing]"	investigate something rule	--
--	true	"reveal [any room]"	investigate something rule	--

This is the investigate something rule:
 repeat through the Table of Footnotey Stuff:
 if the player's command matches the topic entry:
 say "[revelation entry][paragraph break]";
 rule succeeds;
 say "I'm afraid I have no revelation to vouchsafe there."

Section 2 - Scenario

The Upper Deck is a room. Lucius is a man in the Upper Deck.

The maximum score is 501.

When play begins: now the score is 501; end the story finally.

Table of Footnotey Stuff

topic	revelation
"reveal [Lucius]"	"Lucius is based on a historical buccaneer who sailed with William Dampier. The original did carry a Greek New Testament, from which he read aloud when the men were stranded in the jungles near

Panama."
"reveal [Upper Deck]"
"The Callisto is a simplified and tidied representation of a pirate sloop ca. 1688."

386



Example Xerxes

WI

Offering the player a menu of things to read after winning the game.

Building a menu is moderately tedious, so we will rely on the standard menu extensions provided. Thus:

"Xerxes"

Include Basic Screen Effects by Emily Short. Include Menus by Emily Short.

Table of Amusing Matter

title	subtable	description	toggle
"Cult Revisions"	--	"Did you try... [paragraph break] banning the worship of Seth? [line break] of Dionysus? [line break] assigning all your priests to Re? [line break] assigning male priests to Cybele? [line break] assigning married priestesses to Hestia? [line break] identifying one god as another (e.g., Isis and Hecate)? [line break] identifying a mortal as a god (e.g., Alexander as Helios-Apollo)?"	--
"Military Revisions"	--	"Did you try... [paragraph break] allying a Greek city-state with the Persians? (try >MEDIZE) [line break] playing Athens as a land-based power?"	--

Rule for amusing a victorious player:
now the current menu is the Table of Amusing Matter;
now the current menu title is "Things to Try";
carry out the displaying activity;
clear the screen.

Omitting about a half million words from this rigorous and educational but nonetheless enthralling simulation of centuries of history, culture, and religion, we will skip directly to:

Athens is a room.

Use scoring.

Every turn:
if the score is greater than 10000, end the story finally.

When play begins: now the score is 10001.

Test me with "z".

441



Example Labyrinth of Ghosts

WI

Remembering the fates of all previous explorers of the labyrinth.

A tradition among Nethack-like computer games of the old school is that a player's death in a given place leaves a ghost behind to haunt subsequent players. Information about past lives is sometimes stored in a "bones file", and in this example we do exactly that, for a grievously unfair little dungeon.

To begin with, the labyrinth itself. We create a kind of value to remember possible means of death in these tunnels, and we assign a coordinate position in some grid to each location. (We do this because grid positions can safely be stored in tables saved out to external files, whereas room names cannot - they represent data which changes each time we amend the source.)

"Labyrinth of Ghosts"

Use scoring.

A demise is a kind of value. The demises are drowned, buried by a rockfall, pierced by an arrow and slain. The latest demise is a demise that varies.

A grid location is a kind of value. (1,19) specifies a grid location. A room has a grid location called coordinates.

The Gateway is a room. "For the foolhardy adventurer, the perilous labyrinth lies north, east or south." The coordinates are (6,6). The Tomb is east of the Gateway. The coordinates are (7,6). The Rockfall Cave is north of the Gateway. "This partly fallen cave may perhaps extend further north." The coordinates are (6,5). Instead of going north in the Rockfall Cave, have the player buried by a rockfall. The Archery Canyon is south of the Gateway. "No telling why this canyon is named after archery, but perhaps if you wait around you'll find out." The coordinates are (6,7). Instead of waiting in the Archery Canyon, have the player pierced by an arrow. The Rock Pool is east of the Tomb. The coordinates are (8,6). The cold mountain pool is in the Rock Pool. The cold pool is fixed in place. Instead of entering the cold mountain pool, have the player drowned.

Every turn when a random chance of 1 in 10 succeeds:
say "A dwarf appears out of nowhere, and throws a nasty little knife.";
have the player slain.

And as compensation for these hazards:

Some silver bars are in the Tomb. The emerald is in the Rock Pool. The platinum pyramid is in the Canyon.

Table of Point Values
item score

silver bars	3
platinum pyramid	10
emerald	4

Report taking an item listed in the Table of Point Values:
increase the score by the score entry;
blank out the whole row.

We are now ready for the actual undertaking. The Table of Ghostly Presences holds up to twenty death notices, and is initially blank. Deaths are sequentially numbered, and this number is stored in the sequence column.

Table of Ghostly Presences

haunted position score at death turns at death manner of death sequence
a grid location a number a number a demise a number
with 19 blank rows.

As the story file starts up, we look to see if a ghosts file already exists. If one does, we load up the Table of Ghostly Presences with it: and if not, as will be the case the first time the player explores, we leave the table blank. We sort the table so that it has earlier deaths (lower sequence numbers) first.

The File of Ghosts is called "ghosts".

When play begins:

if the File of Ghosts exists, read File of Ghosts into the Table of Ghostly Presences;
sort the Table of Ghostly Presences in sequence order.

How will ghosts manifest themselves? Because this is only a small example, we will simply tell the player that he senses something. If several ghosts are present in the same place, the most aggrieved (that is, the most recent) is sensed first...

After looking:

repeat through the Table of Ghostly Presences in reverse sequence order:
if the haunted position entry is the coordinates of the location, say "You sense the ghostly presence of an adventurer, [manner of death entry] with a score of [score at death entry] in [turns at death entry] turns."

(For instance, "You sense the ghostly presence of an adventurer, buried by a rockfall with a score of 10 in 5 turns.") That just leaves the rule for bumping off the player. When the Table is full, and there are already 20 ghosts, the one who died longest ago (with the lowest sequence count) is eliminated, and his row blanked out. (This will always be row 1 since we sorted the table in sequence order on reading it in.)

To have the player (sticky end - a demise):

let the new sequence number be 0;
repeat through the Table of Ghostly Presences:
let S be the sequence entry;
if S is greater than the new sequence number, let the new sequence number be S;
increment the new sequence number;
if the number of blank rows in the Table of Ghostly Presences is 0:
choose row 1 in the Table of Ghostly Presences;
blank out the whole row;
choose a blank row in the Table of Ghostly Presences;
now the sequence entry is the new sequence number;
now the manner of death entry is the sticky end;
now the turns at death entry is the turn count;
now the score at death entry is the score;
now the haunted position entry is the coordinates of the location;

write the File of Ghosts from the Table of Ghostly Presences;
now the latest demise is the sticky end;
end the story saying "You have been [latest demise]".

Strictly speaking we ought to worry that after 2,147,483,647 deaths, the sequence numbers would grow too large to store in a single value, and then the sequence of ghosts will be erratic. But it seems unlikely that anyone will play this example 2.1 billion times.

138



Example Big Sky Country

WI

Allowing the player to continue play after a fatal accident, but penalizing him by scattering his possessions around the game map.

Some older games allowed the player to be resurrected after a death, but punished him by distributing his possessions far and wide. Here we emulate that effect.

"Big Sky Country"

Use scoring.

When play begins: say "There's a bit of a drive over from Anaconda, Montana, and then through a couple or three ghost towns, but finally you find what you're looking for, and strike out on foot..."

Entrance to Devil's Canyon is a room. "You are at the top of a steep road, which proceeds down into the canyon proper." A sign is in Devil's Canyon. It is fixed in place. "An ominous sign has been put up by the local sheriff's office." The description is "PROCEED AT OWN RISK - NO RESCUES!"

Instead of going down when a random chance of 1 in 3 succeeds:
say "Whooooops, your footing is not as secure as you thought...";
end the story.

Dusty Path is below Entrance. "A dusty path, with grey-brown thorny bushes on either side. Immediately to your right is a sheer drop; far below you can see the rusting remains of a Model T that some fool tried to drive by here."

Hairpin is below Dusty Path. "A sharp bend in the road, doubling back down towards the bottom of the canyon. Just north of here there is also a small cavern of some kind[if the stick pin is in the cavern], which attracts your eye with some glittery thing[end if]."

The Cavern is north of Hairpin. "Really not much more than a little hollow in the side of the canyon." In the cavern are a snake and a diamond stick pin. The snake is an animal. The description of the snake is "You're no expert, but it looks like a rattler."

Instead of taking the diamond stick pin in the presence of the snake: say "Turns out the snake is partial to that there pin, and takes exception to your intending to make off with it."; end the story.

In a fuller implementation of this game, we might make it possible to get by the snake, but in this version, it's just going to remain troublesome.

Crooked Path is below Hairpin. "You're about two thirds of the way down to the bottom of the cavern at this point."

At the Spot is below Crooked Path. "This'll be it: a bare patch of ground that might as well have an X painted right on it."

Rule for supplying a missing noun while digging:
now noun is the location.

Understand "dig" or "dig hole/here" or "dig in ground/dirt/earth" as digging.
Digging is an action applying to one thing.

Instead of digging at the spot:

say "You dig and dig, and after a half hour or so, sure enough, you do turn up a big box of gold! You're going to be richer than God and Bill Gates put together.";

increase the score by 5;
end the story finally.

Instead of digging at the spot when the player does not carry the shovel:
say "What, without your shovel? That won't work too well."

The player carries a walking stick. The player wears a hat, a whistle, and a daypack. The daypack contains a mylar blanket, a granola bar, a cellular phone, a water bottle, a folding shovel, and a photocopied map. The granola bar is edible. Instead of drinking the water, say "You quench your thirst, for the time being." The description of the map is "The map shows the winding path of Devil's Canyon, with a large X down by the south end. That would be where your uncle Jesse buried the gold from the train robbery."

The maximum score is 5.

When play ends when the story has not ended finally:

say "Oh dear, that ought to be fatal! However, if you like I can get you out of it..."

Shall I? >;
if the player consents:
repeat with item running through things had by the player:
move the item to a random visited room;
say "A strong wind picks you up and sets you back at [the location], though perhaps minus a few of your things.";
resume the story;
try looking.

"If the player consents" is just a convenient way to ask a yes/no question that the player must answer before going on with the game.

Chapter 12: Typography, Layout, and Multimedia Effects

§12.1. Typography; §12.2. The Status Line; §12.3. Footnotes; §12.4. Timed Input; §12.5. Glulx Multimedia Effects

-  Contents of *The Inform Recipe Book*
-  Chapter 11: Out Of World Actions and Effects
-  Chapter 13: Testing and Publishing
-  Indexes of the examples

§12.1. Typography

Story files produced by Inform tend not to contain elaborate typographical effects. They would only distract. Like a novel, a classic work of IF is best presented in an elegant but unobtrusive font. Inform does, however, provide for italic and bold-face, and also for a typewriter-style fixed pitch of lettering:

"This is an [italic type]italicised[roman type] word."

"This is an [bold type]emboldened[roman type] word."

"This is a [fixed letter spacing]typewritten[variable letter spacing] word."

Authors making very frequent use of these might like to borrow the briefer definitions in **Chanel Version 1**.

A very wide range of letter-forms is normally available (and even more in quoted text), so that the writer seldom needs to not worry whether, say, a sentence like

A ticket to Tromsø via Østfold is in the Íslendingabók.

will work. **The Über-complète clavier** is an exhaustive test of such exotica.

Coloured type is trickier, and its availability depends on the story file format. For a Z-machine story, **Garibaldi 2** demonstrates this.

Finally, **Tilt 3** combines unusual letterforms (suit symbols) with red and black colours to render hands of cards typographically.

- ⬆ Start of Chapter 12: Typography, Layout, and Multimedia Effects
 - ⬅ Back to Chapter 11: Out Of World Actions and Effects: §11.6. Ending The Story
 - ➡ Onward to §12.2. The Status Line
 - ⬇ Example 69: ★ **Garibaldi 2** Adding coloured text to the example of door-status readouts.
 - ⬇ Example 453: ★ **Chanel Version 1** Making paired italic and boldface tags like those used by HTML for web pages.
 - ⬇ Example 450: ★★ **Tilt 3** Displaying the card suits from our deck of cards with red and black colored unicode symbols.
 - ⬇ Example 70: ★★★ **The Über-complète clavier** This example provides a fairly stringent test of exotic lettering.
-

§12.2. The Status Line

The status line is the reverse-coloured bar along the top of the window during play, which conventionally, but not necessarily, shows the current location, the score (or sometimes the time of day) and the number of turns so far. It has been highly traditional since the early 1980s (Infocom's customer newsletter was for many years called "The Status Line"): it has become the visual identifier of IF. It plays the same role for IF that a header with chapter name and page number plays in a printed book.

The status line is ordinarily printed from two named pieces of text, the "left hand status line" and the "right hand status line". These can be changed during play, so for instance,

When play begins, change the right hand status line to "Time: [time of day]".

The examples below offer miscellaneous alternatives, and are fairly self-descriptive.

★ See **Viewpoint** for a way to make the status line list the player's current identity

- ↑ Start of Chapter 12: Typography, Layout, and Multimedia Effects
 - ← Back to §12.1. Typography
 - Onward to §12.3. Footnotes
 - ↓ Example 114: ★ **Politics as Usual** Have the status line indicate the current region of the map.
 - ↓ Example 350: ★ **Ways Out** A status line that lists the available exits from the current location.
 - ↓ Example 387: ★ **Blankness** Emptying the status line during the first screen of the game.
 - ↓ Example 412: ★ **Capital City** To arrange that the location information normally given on the left-hand side of the status line appears in block capitals.
 - ↓ Example 452: ★ **Status line with centered text, the hard way** A status line which has only the name of the location, centered.
 - ↓ Example 351: ★★ **Guided Tour** A status line that lists the available exits from the current location, changing the names of these exits depending on whether the room has been visited or not.
 - ↓ Example 115: ★★★ **Centered** Replacing the two-part status line with one that centers only the room name at the top of the screen.
-

§12.3. Footnotes

Ibid. provides a version of the traditional Infocom-style approach to footnotes, which number off in the order encountered.

- ↑ Start of Chapter 12: Typography, Layout, and Multimedia Effects
 - ← Back to §12.2. The Status Line
 - Onward to §12.4. Timed Input
 - ↓ Example 300: ★★★ **Ibid.** A system which allows the author to assign footnotes to descriptions, and permits the player to retrieve them again by number, using "the number understood". Footnotes will automatically number themselves, according to the order in which the player discovers them.
-

§12.4. Timed Input

Inform normally expects a purely turn-based story: the player acts, the story responds and waits for the player to act again.

Occasionally, however, we may want to offer a different mode of interaction, for instance with turns in which the player has limited time to come up with his next act. Likewise, we might want to have text that printed itself to the screen gradually, to represent dialogue with pauses, or the speed of a typewriter placing letters on a page.

It's best to be careful with these effects: overdone, they can be very annoying to players who prefer to read at a faster speed. Nonetheless, they do have their uses.

Inform does not have standard syntax to handle real-time delays and output, but there are several extensions that provide this capacity. Erik Temple's extension Real Time Delays, for instance, allows us to specify a delay in milliseconds before continuing with whatever aspect of the story is currently in progress.

★ See **The Passage Of Time** for ways to keep track of clock-time within the story

 Start of Chapter 12: Typography, Layout, and Multimedia Effects

 Back to §12.3. Footnotes

 Onward to §12.5. Glulx Multimedia Effects

§12.5. Glulx Multimedia Effects

Glulx is one of the two basic story file formats to which Inform can work. It is the more powerful of the two, and modern-day Inform uses it by default. At one time it was a less universally playable format, but today players rarely have any trouble getting it to work.

Among its powers are the ability to display images, play back sound effects, and read and write external files to the disc. With care and a certain amount of fuss, this can even give a playing story file limited Internet connectivity, although it should be stressed that this can only be done if the player sets up his computer just right and runs an auxiliary program beside the story itself. That will mostly be too much to ask, if the player is playing offline, but when the story file is being run on an interpreter running at a server - so that the player simply sends commands to it and sees responses back on a web page - one could easily imagine setting up the server to provide these auxiliary programs, without any extra difficulty for the player.

Many of the more advanced multimedia abilities of Glulx are best unlocked using extensions available from the Inform website or the Public Library. As of this writing, extensions exist to help authors create complex multi-windowed displays (including per-location pictures, visual status bars, and even limited animations and gradually-revealed maps).

There is also work in progress on sound-management to allow the author to play sounds in multiple channels at once, to change sound volumes, and to create fade-in and fade-out effects.

Without extensions, all these abilities are within reach for an author who is willing to do some fairly advanced programming.

 Start of Chapter 12: Typography, Layout, and Multimedia Effects

 Back to §12.4. Timed Input

 Onward to Chapter 13: Testing and Publishing: §13.1. Testing

 Example 445:  **Flathead News Network** Using external files, together with a simple Unix script running in the background, to provide live news headlines inside a story file.

Examples from Chapter 12: Typography, Layout, and Multimedia Effects



Start of this chapter



Chapter 13: Testing and Publishing



Indexes of the examples

69

Example Garibaldi 2

WI

Adding coloured text to the example of door-status readouts.

The extension "Basic Screen Effects" provides a few more type styles, in the form of coloured lettering. The colours available are red, yellow, green, blue, white, magenta, and cyan, as well as the usual black; and to restore the player's default screen colour, we say "default letters".

Thus if we wanted to highlight locked and unlocked doors in our security readout example:

"Garibaldi"

Include Basic Screen Effects by Emily Short.

The security readout is a device. The description of the readout is "The screen is blank."

Instead of examining the switched on security readout:

```
say "The screen reads: [fixed letter spacing]";
say line break;
repeat with item running through doors:
  say line break;
  say " [item] ([front side of the item]/[back side of the item]): [if the item is
locked][green letters]LOCKED[default letters][otherwise][red
letters]UNLOCKED[default letters][end if]";
  say variable letter spacing;
  say paragraph break.
```

The player carries the security readout.

The Docking Bay is a room. The inner airlock is a door. It is north of the Docking Bay and south of the Zocalo. The inner airlock is lockable and unlocked. The outer airlock is lockable and locked. It is a door. It is south of the Docking Bay and north of Space.

The quarantine seal is a door. It is west of the Zocalo and east of Medlab. Quarantine seal is locked.

The security pass unlocks the inner airlock. The player carries the security pass.

Test me with "x readout / turn on readout / x readout / lock inner airlock with security pass / x readout".

Note that this extension does not currently produce the desired effects when compiling with the Glulx setting; to see it working, make sure that the settings tab is set to compile to the Z-machine.

453

★ Example Chanel Version 1

WI

Making paired italic and boldface tags like those used by HTML for web pages.

HTML uses angled brackets to achieve effects, and places italicised text between `<i>` and `</i>` tags; and similarly boldface between `` and ``. We can mimic this very easily by setting each up as a segmented substitution:

"Chanel Version 1"

To say `i` -- beginning `say_i` -- running on: (- style underline; -).

To say `/i` -- ending `say_i` -- running on: (- style roman; -).

To say `b` -- beginning `say_b` -- running on: (- style bold; -).

To say `/b` -- ending `say_b` -- running on: (- style roman; -).

Place Vendôme is a room. "[i]Fashion fades, only style remains the same[/i] ([b]Coco Chanel[/b]). And this elegant drawing-room, once a milliner's shop, is a case in point."

Instead of going nowhere, say "[i]Don't spend time beating on a wall, hoping to transform it into a door.[/i] ([b]Coco Chanel[/b]) This one is a wall."

Test me with "look / e".

We have had to use square instead of angle brackets, but then, "in order to be irreplaceable one must always be different" (Coco Chanel).

(Marking these as substitutions which run on prevents unexpected paragraph breaks if they should appear immediately after the end of a sentence.)

450

★★ Example Tilt 3

WI

Displaying the card suits from our deck of cards with red and black colored unicode symbols.

Books and articles about card-playing traditionally abbreviate card names into a simple two-symbol notation: a number or letter representing the card rank, followed by a symbol indicating the card suit. Suppose that we want to emulate this notation when taking inventory in our poker game.

The trick here is that colored output is done in different ways by the Z-Machine and by Glulx, so we'll need two different versions of the same section in order to produce this output. The relevant source is right at the beginning:

"Tilt"

Section 0 - Colored Output in Two Forms

For the suit symbols, we'll want the Unicode extension included with Inform:

Include Unicode Character Names by Graham Nelson.

Rule for printing the name of a card (called target) while grouping together:
say "[rank of the target as abbreviated value][suit of the target as symbol]".

To say (current suit - a suit) as symbol:
if current suit is diamonds, say "[red letters][unicode black diamond suit][default letters]";
if current suit is spades, say "[unicode black spade suit]";
if current suit is clubs, say "[unicode black club suit]";
if current suit is hearts, say "[red letters][unicode black heart suit][default letters]".

Section 0Z (for Z-machine only)

The Basic Screen Effects extension bundled with Inform includes mechanisms to change the text color, so for the Z-machine, we need only include this:

Include Basic Screen Effects by Emily Short.

Section 0G (for Glulx only)

Under Glulx, we need slightly more set-up: Glulx requires that we define special user font styles when we plan to make display changes. A fuller discussion of this (and of how to define new colors) appears in the documentation of "Glulx Text Effects", but an implementation sufficient to our purposes would be

Include Glulx Text Effects by Emily Short.

Table of User Styles (continued)

style name	glulx color
special-style-1	g-pure-red

Table of Common Color Values (continued)

glulx color value	assigned number
g-pure-red	16711680

To say red letters: say first custom style.

To say default letters: say roman type.

From here, the rest of the source is mostly as we've seen in previous examples:

Section 1 - Cards

Suit is a kind of value. The suits are hearts, clubs, diamonds, and spades.
Understand "heart" as hearts. Understand "club" as clubs. Understand
"diamond" as diamonds. Understand "spade" as spades.

A card is a kind of thing. A card has a suit. A card has a number called rank.
Understand the suit property as describing a card. Understand the rank property
as describing a card.

52 cards are in the card repository.

To say (count - a number) as a card value:
choose row count in the Table of Value Names;
say "[term entry]".

Rule for printing the name of a card (called target):
say "[rank of the target as a card value] of [suit of the target]"

To say (count - a number) as abbreviated value:
choose row count in the Table of Value Names;
say "[abbrev entry]".

Table of Value Names

term	value	abbrev	topic
"ace"	"1"	"A"	"ace/A"
"deuce"	"2"	"2"	"deuce/two"
"three"	"3"	"3"	"three"
"four"	"4"	"4"	"four"
"five"	"5"	"5"	"five"
"six"	"6"	"6"	"six"
"seven"	"7"	"7"	"seven"
"eight"	"8"	"8"	"eight"
"nine"	"9"	"9"	"nine"
"ten"	"10"	"10"	"ten"
"jack"	"11"	"J"	"jack/knave/J"
"queen"	"12"	"Q"	"queen/Q"
"king"	"13"	"K"	"king/K"

After reading a command:
if the player's command includes "of [suit]":
while the player's command includes "of":
cut the matched text;
repeat through the Table of Value Names:
while the player's command includes topic entry:
replace the matched text with value entry.

When play begins:
reconstitute deck.

To reconstitute deck:
let current suit be hearts;
now every card is in the card repository;
while a card is in the card repository:
repeat with current rank running from 1 to 13:
let item be a random card in card repository;
now rank of item is current rank;
now suit of item is current suit;

now item is in the deck of cards;
now current suit is the suit after the current suit.

Section 2 - The Deck and the Discard Pile

The Empty Room is a room. "Nothing to see here."

The deck of cards is in the Empty Room. It is a closed unopenable container.
The description is "A standard poker deck."

The discard pile is a closed unopenable container. The description is "Cards in this game are discarded face-down, so the discard pile is not very interesting to see. All you can observe is that it currently contains [if the number of cards which are in the discard pile is less than ten][the number of cards which are in the discard pile in words][otherwise]about [the rounded number of cards which are in the discard pile in words][end if] card[s]."

To decide what number is the rounded number of (described set - a description of objects):

let N be the number of members of the described set;
let R be N divided by 5;
let total be R times 5;
decide on total.

Rule for printing room description details of something: do nothing instead.

Section 3 - Drawing and Discarding Actions

Understand the commands "take" and "carry" and "hold" and "get" and "drop" and "throw" and "discard" as something new.

Understand "take [text]" or "get [text]" or "drop [text]" as a mistake ("Here, you only draw and discard. Nothing else matters at the moment.").

Understand "draw" or "draw card" or "draw a card" as drawing. Drawing is an action applying to nothing. The drawing action has an object called the card drawn.

Setting action variables for drawing:

now the card drawn is a random card which is in the deck of cards.

Check drawing:

if the card drawn is nothing, say "The deck is completely depleted." instead.

Check drawing:

if the number of cards carried by the player is greater than four,
say "This is a five-card game; you must discard something before drawing anything further." instead.

Carry out drawing:

move the card drawn to the player.

Report drawing:

say "You draw [a card drawn]."

Understand "discard [card]" as discarding. Discarding is an action applying to one thing.

Check discarding:

if the player does not carry the noun, say "You can only discard cards from your own hand." instead.

Carry out discarding:

now the noun is in the discard pile;

if the discard pile is not visible, move the discard pile to the location.

Report discarding:

say "You toss [the noun] nonchalantly onto the discard pile."

Section 4 - Assessing Hands

Before listing contents while taking inventory: group cards together.

Before grouping together cards:

if the number of cards carried by the player is 5:

say "[run paragraph on]";

follow the hand-ranking rules;

if the rule succeeded, say "[the outcome of the rulebook]";

otherwise say "some random cards";

if the outcome of the rulebook is pair outcome, say " of [rank of the first thing held by the player as a card value]s";

otherwise:

say "[number of cards carried by the player in words] assorted cards";

say " (".

Rule for grouping together cards:

say "[list hand]".

To say list hand:

let chosen card be the first thing held by the player;

while chosen card is a card:

say "[chosen card]";

now chosen card is the next thing held after chosen card;

if chosen card is a card, say "-".

After grouping together cards:

say ")".

The hand-ranking rules is a rulebook. The hand-ranking rules have outcomes royal flush, straight flush, four of a kind, full house, flush, straight, three of a kind, two pairs, pair, high card.

The hand-ranking rulebook has a truth state called the flushness.

The hand-ranking rulebook has a truth state called the straightness.

The hand-ranking rulebook has a number called the pair count.

The hand-ranking rulebook has a number called the triple count.

The hand-ranking rulebook has a number called the quadruple count.

A card can be sorted or unsorted. A card is usually unsorted.

Definition: a card is high if its rank is 11 or more.

Definition: a card is low if its rank is 4 or less.

A hand-ranking rule (this is the initial sort rule):

now every card is unsorted;

while the player carries an unsorted card:

let item be the lowest unsorted card held by the player;

move item to the player;

now the item is sorted;

if sort-debugging is true, say "-- after initial sort: [list hand]".

A hand-ranking rule (this is the finding flushness rule):

let called suit be the suit of a random card carried by the player;

if every card carried by the player is called suit, now flushness is true.

A hand-ranking rule (this is the finding straightness rule):

now straightness is true;

let N be the rank of the highest card which is carried by the player;

repeat with current rank running from N - 4 to N:

now the test rank is the current rank;

unless the player carries a matching card:

if the current rank is N - 4 and the current rank is 9 and the player carries an ace card, do nothing;

otherwise now straightness is false.

A card can be quadrupled, tripled, paired or uncombined.

Test rank is a number that varies. Definition: a card is matching if its rank is the test rank.

A hand-ranking rule (this is the counting multiples rule):

now every card is uncombined;

repeat with current rank running from 1 to 13:

now test rank is current rank;

let N be the number of matching cards held by the player;

if N is 4:

increment the quadruple count;

now every matching card held by the player is quadrupled;

if N is 3:

increment the triple count;

now every matching card held by the player is tripled;

if N is 2:

increment the pair count;

now every matching card held by the player is paired.

A hand-ranking rule (this is the move aces up unless there's a low straight rule):

unless the straightness is true and the lowest card carried by the player is an ace card and the rank of the highest card carried by the player is 5,

now every ace card which is carried by the player is carried by the player;

if sort-debugging is true, say "-- after ace movement rule: [list hand]".

A hand-ranking rule (this is the move pairs forward rule):

while the player carries a paired card:

let selection be the lowest paired card which is carried by the player;

move the selection to the player;

now the selection is uncombined;
if sort-debugging is true, say "-- after pairs movement: [list hand]".

A hand-ranking rule (this is the raise ace pairs rule):
if the player carries exactly two ace cards:
repeat with item running through ace cards which are carried by the player:
move item to the player;
if sort-debugging is true, say "-- after paired-ace movement: [list hand]".

A hand-ranking rule (this is the move multiples forward rule):
while the player carries a tripled card:
let selection be the lowest tripled card which is carried by the player;
move the selection to the player;
now the selection is uncombined;
while the player carries a quadrupled card:
let selection be the lowest quadrupled card which is carried by the player;
move the selection to the player;
now the selection is uncombined;
if sort-debugging is true, say "-- after multiples movement rule: [list hand]".

Definition: a card is ace if its rank is 1.
Definition: a card is king if its rank is 13.

A hand-ranking rule (this is the royal-flush rule):
if flushness is true and straightness is true and the highest card carried by the player is king and the lowest card carried by the player is ace, royal flush.

A hand-ranking rule (this is the straight-flushes rule):
if flushness is true and straightness is true, straight flush.

A hand-ranking rule (this is the four-of-a-kind rule):
if the quadruple count is 1, four of a kind.

A hand-ranking rule (this is the full-house rule):
if the pair count is 1 and the triple count is 1, full house.

A hand-ranking rule (this is the flushes rule):
if flushness is true, flush.

A hand-ranking rule (this is the straights rule):
if straightness is true, straight.

A hand-ranking rule (this is the three-of-a-kind rule):
if triple count is 1, three of a kind.

A hand-ranking rule (this is the two-pair rule):
if the pair count is 2, two pairs.

A hand-ranking rule (this is the pair rule):
if the pair count is 1, pair.

A hand-ranking rule (this is the default rule):
high card.


```

hearts 12 [pair]
spades 12
diamonds 10
spades 7
clubs 4
diamonds 13 [high]
hearts 11
spades 9
clubs 7
diamonds 5
hearts 1 [tricky sorting: low straight]
diamonds 2
spades 3
diamonds 4
diamonds 5

```

Understand "force hand" as forcing a hand. Forcing a hand is an action out of world.

Current marker is a number that varies.

Carry out forcing a hand:

```

repeat with item running through cards which are carried by the player:
  increment current marker;
  if current marker is greater than the number of filled rows in the Table of
Testing Hands, now current marker is 1;
  choose row current marker in the Table of Testing Hands;
  now the suit of item is the set suit entry;
  now the rank of item is the set rank entry.

```

Report forcing a hand:
try taking inventory.

70



Example The Über-complète clavier

WI

This example provides a fairly stringent test of exotic lettering.

The following example puts Inform's support for exotic lettering through its paces. It was useful in testing Inform but is not a very instructive read: still, it does provide a test story file for interpreters, so we are including the source here as an example.

"The Über-complète clavier"

The story headline is "Pushing the Limits of Unicode in IF". The story description is "This is a demanding test for Unicode compliance by Z-machine interpreters."

Include Unicode Character Names by Graham Nelson.

Include Basic Screen Effects by Emily Short.

The Château Bibliothèque Français is east of the Deutsche Universität Bücherei. "From this Borgesian construction, doorways lead into anterooms in each of the four cardinal directions." South of the Bibliothèque is the Miscellany Mañana. North of the Bibliothèque is the Íslendingabók. East of the Bibliothèque is Alphabet Soup.

A framed photograph of Icelandic Prime Minister Halldór Ásgrímsson, a ruler measuring Ångströms, a Bokmål-Lëtzebuergesch Lëtzebuergesch-Bokmål dictionary and a ticket to Tromsø via Østfold are in the Íslendingabók.

A paper by Karl Weierstraß, a general feeling of Ärger, an old Österreich passport and the Bach Clavier-Übung open at the fugue à 4 are in the Bücherei.

The painting of École normale supérieure students singing Ça ira, the frankly lesser-known journal of Niccolò Polo, Così fan tutte on CD, an extract of Herodotus concerning Artaÿctes and the exit sign reading À BIENTÔT are in the Bibliothèque.

A wicker basket marked CHLOË is in the Bibliothèque. A ginger cat is in the basket.

A guide to Æsop for naïve aesthetes, Lönnrot's Kalevala, a creed according to the Bahá'ís, FALARÃO magazine, an Estonian poem by Tõnu Trubetsky, a Portuguese-Italian recipe for macarrão, a stripy hanging CANDY PIÑATA bag, a ζ;ιPunctuation Turned Upside Downζ;ι pamphlet, an Italian brewers' anti-violence poster declaring BÓTTE NON BÓTTE, a map of È and a dusty book titled The Parnasum of Luís Vaz bearing CAMÕES on its spine are in Miscellany Mañana.

The description of the map is "È is a province in the People's Republic of China."

In Mañana is something called ÂÊÎÔÛ - The Official Journal of the Society for Vowels bearing Circumflexes.

In Mañana is something called âêîôû comic - the youth edition.

The description of Alphabet Soup is "A bewildering place of glyphs, sigils and signs. The Library proper leads back west: steps lead upwards to an Observatory, or downwards into what seems to be a dangerous area. A gaming lounge lies to the south."

The Greek Alphabet, the Cyrillic Alphabet, the Hebrew alphabet, and the embossed plaque are in Alphabet Soup. The description of the Greek alphabet is "αβγδεζηθικλμνξοπρστυφχψω.". The description of the Hebrew alphabet is "תשרקצףפףעסונמםלכלךיטחזוזבגדחא.". The description of the Cyrillic alphabet is "абвгдежзийклмнопрстуфхцчшщъыьэюя."

Instead of examining the plaque:

say "It seems to be a sign in Braille: ";
say unicode Braille pattern dots-24, " (I), ",
unicode Braille pattern dots-1345, " (N), ",
unicode Braille pattern dots-124, " (F), ",
unicode Braille pattern dots-135, " (O), ",
unicode Braille pattern dots-1235, " (R), ",
unicode Braille pattern dots-134, " (M)."

The Gaming Lounge is south of Alphabet Soup. The chess position and the book of puzzle canons are in the Gaming Lounge.

The Georges de la Tour painting Le Tricheur is in the Gaming Lounge. "Hanging on one wall is Georges de la Tour's masterpiece Le Tricheur (the card-sharp). Visible are 8[unicode black diamond suit], 9[unicode black diamond suit],

A[unicode black diamond suit], A[unicode black spade suit], 6[unicode black club suit] but not one of them has a [unicode black heart suit]."

The description of Le Tricheur is "If they'd been dice-players instead, they might have thrown [unicode die face-1], [unicode die face-2], [unicode die face-3], [unicode die face-4], [unicode die face-5] or [unicode die face-6], but as it is they stick to cards."

The description of the book of canons is "A typical fugue is no. 13 (Tovey: [unicode eighth note] = 110) in F[unicode music sharp sign] minor, but you can also make out keys like A[unicode music flat sign] and G[unicode music natural sign]."

The empty square text is text that varies. To say empty: say the empty square text.

To display the board:

```
say empty, empty, empty, empty, empty, empty, empty, unicode black chess king,
empty, line break;
say empty, empty, empty, unicode black chess queen, empty, empty, unicode
black chess pawn, empty, line break;
say unicode black chess pawn, empty, empty, unicode black chess bishop,
unicode black chess pawn, empty, empty, unicode black chess pawn, line break;
say empty, empty, empty, unicode black chess pawn, empty, unicode black
chess rook, empty, empty, line break;
say empty, unicode black chess pawn, empty, unicode white chess pawn,
unicode black chess pawn, empty, empty, empty, line break;
say empty, empty, empty, unicode black chess bishop, unicode white chess
queen, empty, unicode white chess pawn, unicode white chess pawn, line break;
say unicode white chess pawn, unicode white chess pawn, empty, unicode
white chess bishop, empty, unicode black chess rook, unicode white chess
bishop, empty, line break;
say empty, unicode white chess knight, empty, empty, unicode white chess
rook, empty, unicode white chess rook, unicode white chess king, line break.
```

Instead of examining the chess position:

```
say "Fritz Saemisch - Aron Nimzowitsch, Copenhagen 1923: the Immortal
Zugzwang Game. Nimzowitsch (black), observing that white will very soon have
to play a terrible move, has just advanced his h pawn for no reason other than to
wait. So it is white to play...";
say "[fixed letter spacing].....k. [line break]...q..p. [line break]p..bp..p [line
break]...p.r.. [line break].p.Pp... [line break]...bQ.PP [line break]PP.B.rB. [line
break].N..R.RK [variable letter spacing][line break]";
say "'White must, willy-nilly, eventually throw himself upon the sword', in
Nimzowitsch's commentary. ";
say "We will now try to display the same position using chess-piece symbols
in a Unicode font.";
say fixed letter spacing;
now the empty square text is " ";
display the board;
say variable letter spacing.
```

The Astrological Observatory is above Alphabet Soup.

The planets are in the Observatory. "Diagrams of the planets are scattered across the dome: Sun [unicode Sun], Mercury [unicode Mercury], Venus

[unicode Female Sign], Earth [unicode Earth], Moon [unicode First Quarter Moon] and [unicode Last Quarter Moon], Mars [unicode Male Sign], Jupiter [unicode Jupiter], Saturn [unicode Saturn], Uranus [unicode Uranus], Neptune [unicode Neptune], Pluto [unicode Pluto] and one or two comets [unicode Comet]. Fainter, but all around, you see stars black [unicode black star] and white [unicode white star]."

The constellations are in the Observatory. "Ringing the dome are the constellations: Aries [unicode Aries], Taurus [unicode Taurus], Gemini [unicode Gemini], Cancer [unicode Cancer], Leo [unicode Leo], Virgo [unicode Virgo], Libra [unicode Libra], Scorpius [unicode Scorpius], Sagittarius [unicode Sagittarius], Capricorn [unicode Capricorn], Aquarius [unicode Aquarius], Pisces [unicode Pisces]."

The weather almanac is in the Observatory. The description of the almanac is "Here nightly observers scrawl in hasty abbreviations for the current weather conditions: clear weather [unicode Black Sun with Rays], cloudy [unicode cloud], rain [unicode umbrella], snow [unicode snowman], lightning [unicode lightning], thunderstorm [unicode thunderstorm]."

The Danger Zone is below Alphabet Soup. The printed name of the Danger Zone is "[unicode skull and crossbones] Danger Zone [unicode skull and crossbones]".

The warning signs are in the Danger Zone. "A variety of international-standard warning standards suggest that this may not be the safest place: [unicode skull and crossbones], [unicode caution sign], [unicode radioactive sign], [unicode biohazard sign]."

This example text was used to produce a story file which has been tried against both Zoom for Mac OS X and Windows Frotz. The Latin, Greek, Cyrillic and Hebrew text all functioned perfectly on both, but a point of difference showed when writing the Hebrew alphabet: Zoom wrote this right-to-left, Windows Frotz left-to-right. The exotic symbols displayed on Zoom (though others not mentioned above, such as "[unicode staff of hermes]", did not): but most appeared only as black squares on Windows Frotz, exceptions being the astrological signs for Venus and Mars and the musical note.

Example Politics as Usual

Have the status line indicate the current region of the map.

Suppose a game with a large map entirely subdivided into regions. We could define:

"Politics as Usual"

When play begins:

now the right hand status line is "[map region of the location]".

Washington is west of Idaho.

Red is a region. Blue is a region. Idaho is in red. Washington is in blue.

Test me with "e / w".

Note that, since regions can be stacked, we technically can be within more than one region at once. In the Port Royal example, for instance, the Tavern region is inside the Inland region. If there is any ambiguity, "the map region of the location" will be construed as "the smallest region that the location belongs to": so we would see "Tavern" rather than "Inland" in the status bar, when the player was in the Feathers or the Feathers Bedroom.

Some extra finesse would be necessary if the names of map regions were very long or if there were some rooms that were not considered to belong to any region at all.

350

Example Ways Out

WI

A status line that lists the available exits from the current location.

A not-uncommon device in games with large maps is a list of available exits printed in the status bar. We might do this so:

"Ways Out"

When play begins:

now left hand status line is "Exits: [exit list]";
now right hand status line is "[location]".

To say exit list:

let place be location;
repeat with way running through directions:
let place be the room way from the location;
if place is a room, say "[way]".

We may find that printing out full directions makes the status line unpleasantly crowded. Fortunately, it isn't hard to provide a set of abbreviations to use in this context:

Rule for printing the name of a direction (called the way) while constructing the status line:

choose row with a heading of the way in the Table of Abbreviation;
say "[shortcut entry]".

Table of Abbreviation

heading	shortcut
north	"N"
northeast	"NE"
northwest	"NW"
east	"E"
southeast	"SE"
south	"S"
southwest	"SW"
west	"W"
up	"U"
down	"D"
inside	"IN"

outside "OUT"

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Test me with "w / n / e / e / s / e".

Everywhere else, the names of directions will still be printed out in full in the usual way.

387

★ Example Blankness

WI

Emptying the status line during the first screen of the game.

Occasionally we want to print something as our first screen and then pause the game. By default, Inform will print a rather odd status line, with "You" on the left side and "0" on the right. This is because the left hand status line is set to display the location, but (because we're not done with the when-play-begins rules) the player has not yet even been moved to a room.

We can tidy this up in the "starting the virtual machine" activity, by temporarily changing the status line content. We will not provide game-pausing code here, because that is easily done by extension; so:

"Blankness"

Include Basic Screen Effects by Emily Short.

```
When play begins:  
  say "take me home";  
  wait for any key;  
  say " yeah";  
  wait for any key;  
  say " yeah";  
  pause the game;  
  now the left hand status line is "[location]";  
  now the right hand status line is "[turn count]".
```

```
Before starting the virtual machine:  
  now the left hand status line is "";  
  now the right hand status line is "".
```

Paradise City is a room. The description of Paradise City is "The grass is green and the girls are pretty."

Quite a modest effect, but occasionally useful.

412

★ Example Capital City

WI

To arrange that the location information normally given on the left-hand side of the status line appears in block capitals.

Not much is needed for this. The only noteworthy point is that it doesn't work by changing the LHSL to "[the player's surroundings in upper case]": it cannot do this because "the player's surroundings" is not a value. Instead, "[the player's surroundings]" is a text substitution sometimes printing the name of a room, sometimes printing "Darkness", and so on. We must therefore load it into a text first, and then apply "...in upper case".

"Capital City"

Capital City is a room. East is Lower Caissons. South of Lower Caissons is San Seriffe. East of San Seriffe is a dark room.

To say the player's capitalised surroundings:
let the masthead be "[the player's surroundings]" in upper case;
say the masthead.

When play begins:
now the left hand status line is "[the player's capitalised surroundings]".

Test me with "e / s / e".

452

★ Example Status line with centered text, the hard way

WI

A status line which has only the name of the location, centered.

Making major changes to display features, such as the construction of the status line, sometimes requires that we rely on Inform 6 in bulk; here is how we might produce the Trinity-style status line, with the location centered at the top center of the screen.

"Corner of No and Where"

No is a room. Where is west of No.

Rule for constructing the status line:
print the location in the center of the status line;
rule succeeds.

To print the location in the center of the status line:
(- PrintCenteredStatus(); -).

Include (-

```

Array printed_text --> 64;

[ PrintCenteredStatus i j;
  @set_cursor 1 0;
  i = 0->33;
  spaces(i);
  @output_stream 3 printed_text;
  print (name) location;
  @output_stream -3;
  j = (i - (printed_text-->0))/2;
  @set_cursor 1 j;
  print (name) location;
  spaces j-1;
];

-)

Test me with "w / e".

```

In fact, as we've already seen, many extra modifications to the display behavior are possible using Basic Screen Effects.

351



Example Guided Tour

WI

A status line that lists the available exits from the current location, changing the names of these exits depending on whether the room has been visited or not.

It may sometimes be helpful to prompt the player with a list of exits printed up in the status line. For instance, here is a status line that will print the names of nearby rooms, as well as all the doors the player can see:

"Guided Tour"

When play begins:

```

now left hand status line is "Nearby: [if a room is adjacent][the list of adjacent
rooms][end if][if a room is adjacent and a door is visible] and [end if][if a door is
visible][the list of visible doors][end if]";
now right hand status line is "".

```

Of course, we may not want to tell the player what glories are to be found in locations he hasn't yet explored.

Rule for printing the name of an unvisited room (called the target) while constructing the status line:

```

let aim be the best route from the location to the target;
say "something [aim]".

```

Even when we have seen a room, we might still want a reminder about how to get there:

After printing the name of a visited room (called the target) while constructing the status line:

```
let aim be the best route from the location to the target;
say " ([aim])".
```

We may also find that printing out full directions makes the status line unpleasantly crowded. Fortunately, it isn't hard to provide a set of abbreviations to use in this context:

Rule for printing the name of a direction (called the aim) while constructing the status line:

```
choose row with a heading of the aim in the Table of Abbreviation;
say "[shortcut entry]".
```

Table of Abbreviation

heading	shortcut
north	"N"
northeast	"NE"
northwest	"NW"
east	"E"
southeast	"SE"
south	"S"
southwest	"SW"
west	"W"
up	"U"
down	"D"
inside	"in"
outside	"out"

Everywhere else, the names of directions will still be printed out in full in the usual way. And now we give it a little map to work with:

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Test me with "n / w / s".

Note that while this looks fine in some places, other locations exceed the limits of what the status-line can hold: if any given room is going to have a large number of exits, this kind of listing will almost certainly not fit. So apply cautiously.



If we want to lay out the status line in some other way than with left-hand and right-hand entries, it is possible to do this as well. Later we will learn about the "rule for constructing the status line", but here is a basic effect using this rule and an Inform extension included as part of the standard distribution, called Basic Screen Effects.

"Centered"

When play begins:

say "After months of boring through the Earth's crust in this metal-jawed vehicle, you break through..."

The Hollow Core is a room. "Truly a magnificent sight: the land curves up away from you in every direction, covered with the cities and fields of the Core People. Molten rock runs in the canals, bringing heat and light to every home.

At the center of the Earth hangs a dense black sun."

Include Basic Screen Effects by Emily Short.

Rule for constructing the status line:

center "[location]" at row 1;
rule succeeds.

Test me with "look".

Basic Screen Effects also provides a mechanism for building complicated status lines of more than one row. To read its documentation, we include the extension, press Go!, and then consult the contents index that results.

300



Example Ibid.

WI

A system which allows the author to assign footnotes to descriptions, and permits the player to retrieve them again by number, using "the number understood". Footnotes will automatically number themselves, according to the order in which the player discovers them.

"Hitchhiker's Guide to the Galaxy" introduced the idea of footnoted descriptions, and various IF games since have toyed with the idea. The recommended implementation in Inform 6 involved keeping an assortment of footnote objects around, but in Inform 7 the table is a much tidier way of handling the same problem.

"Ibid."

The Ship Inn is a room. "Here you are in a lovely pub which your guidebook assures you is extremely authentic. [1 as a footnote].

To your left sits a party of Italians, with their guidebook.

To your right is a silent, but not unappealing, young man."

A party of Italians and a silent young man are people in the Ship Inn. The Italians and the young man are scenery.

The table is a supporter in the Ship Inn. On the table is a mysterious pie. The description of the pie is "Your waitress told you it was the specialty of the day, Steak and Owl Pie. [2 as a footnote]." The pie is edible.

Table of Footnotes

```
assignment note
a number "Francis Drake ate here, if the sign on the door is to be believed"
-- "this is unlikely, considering that owls are protected animals in England these days [3 as a footnote]"
-- "moreover, you can't imagine that owl would be very tasty"
```

Footnotes mentioned is a number that varies.

Whenever we mention a footnote for the first time, we need to assign it a number, which we will use consistently thereafter. And it's probably a good idea to protect ourselves against the author accidentally using a number too large for the footnote table, too. So:

```
To say (footnote - a number) as a footnote:
  if footnote > number of filled rows in the Table of Footnotes:
    say "Programming error: footnote assignment out of range.";
  otherwise:
    choose row footnote in the Table of Footnotes;
    if there is an assignment entry:
      say "([assignment entry])";
    otherwise:
      increment footnotes mentioned;
      choose row footnote in the Table of Footnotes;
      now assignment entry is footnotes mentioned;
      say "([assignment entry])".
```

Now, in order to let the player view these footnotes, we'll need to parse numbers.

Understand "footnote [number]" as looking up a footnote.

Looking up a footnote is an action applying to one number.

```
Check looking up a footnote:
  if the number understood > footnotes mentioned, say "You haven't seen any
such footnote." instead;
  if the number understood < 1, say "Footnotes are numbered from 1."
```

```
Carry out looking up a footnote:
  choose row with assignment of number understood in the Table of Footnotes;
  say "([assignment entry]): [note entry]."
```

Test me with "footnote 1 / examine pie / footnote 2 / footnote 3".

This method does require us to keep track of where a footnote appears in the table. If we found this inconvenient, we could add a column to the footnote table so that we could invoke it with tags like "[appearance quip as a footnote]".



Example Flathead News Network

Using external files, together with a simple Unix script running in the background, to provide live news headlines inside a story file.

This example can only work if we have a separate program running in the background while the story file is being played, and as such it will only work if we set things up in a special way. Exactly how to do this will vary from platform to platform.

First, the source text for the Inform end of the communication line:

"Flathead News Network"

The file of RSS Requests is called "rssrequest".

The file of RSS Responses (owned by project "RSS-SCRIPT") is called "rssreply".

To request (RSS feed address - text):

mark the file of RSS Responses as not ready to read;
write the RSS feed address to the file of RSS Requests.

Newsroom is a room. "This is the secret nerve-centre of FNN, the Flathead News Network."

The BBC button is in the Newsroom. Instead of pushing the BBC button: say "Bong!"; request "newsrss.bbc.co.uk/rss/newsonline_uk_edition/front_page/rss.xml".

The NASA button is in the Newsroom. Instead of pushing the NASA button: say "Bang!"; request "www.nasa.gov/rss/breaking_news.rss".

The WHO button is in the Newsroom. Instead of pushing the WHO button: say "Bing!"; request "http://www.doctorwhonews.net".

A screen is in the Newsroom.

Instead of examining the screen:

if ready to read the file of RSS Responses, say "From the screen you read:
[line break][text of the file of RSS Responses]";
otherwise say "The screen remains blank for now."

As far as the story file is concerned, then, it sends a request down the communication line by writing the chosen RSS feed to the file named "rssrequest", and expects a reply to come back down the line by being written to the file "rssreply". However, the story file needs to expect that this might take some time. (Maybe forever, if there is no program responding, or if the Internet connection is not working.) The story file marks the "rssreply" file as not ready before it makes a request; if it subsequently

finds that the file is now ready, that must mean that the other program has done the honours, and that all is well. In the mean time, "The screen remains blank for now."

Now for the RSS-SCRIPT program. The following provides a crude but workable program suitable for running as a Perl script on a system which provides the standard Internet fetching program "curl": Mac OS X, for instance. (If you have OS X, you can paste the following into a (Unix-format) text file called "rss-script.pl", place it in your home folder, open the Terminal utility, type "perl rss-script.pl", and then hide the Terminal window again.)

```
for (;;) { # repeat forever:
  system("sleep 1"); # wait 1 second
  open REQUEST, "rssrequest.glkdata" or next;
  # the request file has been detected:
  $header_line = <REQUEST>; # the header line
  $rss_feed = <REQUEST>; # the actual content - the RSS feed URL
  close REQUEST;
  if ($header_line =~ m/^\*/) { # if the request file is marked ready
    $rss = system("curl $rss_feed >rawrss.txt"); # download the RSS feed
    # read the RSS XML into a single Perl string:
    open RAWRSS, "rawrss.txt" or next;
    $raw = "";
    while ($nl = <RAWRSS>) {
      $raw = $raw." ".$nl;
    }
    close RAWRSS;
    # look for the title and description of the first item:
    if ($raw =~ m/^\<item\>\<title.*?\>(.*?)\</title\>.*?\<description.*?\>(.*?)\</description\>/) {
      # write the reply:
      open REPLY, ">rssreply.glkdata" or next;
      print REPLY "*" //RSS-SCRIPT// rssreply\n", $1, "\n", $2, "\n";
      close REPLY;
      # request safely dealt with, so we can remove it:
      system("rm 'rssrequest.glkdata'");
    }
  }
}
```

Chapter 13: Testing and Publishing

§13.1. Testing; §13.2. Publishing

-  Contents of *The Inform Recipe Book*
-  Chapter 12: Typography, Layout, and Multimedia Effects
-  Indexes of the examples

§13.1. Testing

There are no recipes for testing, alas, although most experienced IF authors have their preferred ways of going about it. Briefly: the best advice is to build and maintain a Skein which holds complete play-throughs of the piece, so that it is easy to check with a single click that all is still well; to have beta-testers who play through (substantial) drafts and send back their transcripts of play, along with suggestions; and to listen to these suggestions, and treat the beta-testers as editors rather than galley slaves.

Alpha shows a way to gracefully accept beta-testers' annotations without advancing time in the story.

Most large works of IF have historically provided secret commands for testing and debugging - commands removed (or sometimes accidentally not) in the final released product. Inform does this automatically: the commands SHOWME, ACTIONS and SCENES are always present except in a released story file. It also allows us to write passages of source text which apply only for the testing phase, so that we can define new testing commands, or other checks that all is well: **Bic** demonstrates this, and is also useful in its own right.

-
-  Start of Chapter 13: Testing and Publishing
 -  Back to Chapter 12: Typography, Layout, and Multimedia Effects: §12.5. Glulx Multimedia Effects
 -  Onward to §13.2. Publishing
 -  Example 2:  **Bic** Testing to make sure that all objects have been given descriptions.
 -  Example 414:  **Alpha** Creating a beta-testing command that matches any line starting with punctuation.
-

§13.2. Publishing

These three examples simply show what can be done using Inform's EPS-format map output, if one is willing to tweak the design in some vector-art program. Unfortunately, at present, there are few alternatives to Adobe Illustrator - a superb but very expensive program - in the field of EPS editing, and indeed, of vector art generally: this is especially the case for Windows users. Inkscape is a usable free alternative, but it needs to have EPS files translated

to PDFs before they can be used. On Mac OS X, the built-in Preview application can do this; otherwise the open-source Ghostscript might be used, but it can be a pain to install. Still, for the IF author who does have EPS editing facilities available, Inform will play nicely with them.

-
-  Start of Chapter 13: Testing and Publishing
 -  Back to §13.1. Testing
 -  Example 446:  **Baedeke** Creating a floorplan of the cathedral using the locations from previous examples.
 -  Example 447:  **Port Royal 5** Port Royal scenario given instructions for an EPS map.
 -  Example 448:  **Bay Leaves and Honey Wine** Creating a map of Greece using the locations from previous examples.
-

Examples from Chapter 13: Testing and Publishing

-  Start of this chapter
-  Indexes of the examples

2

Example **Bic**

Testing to make sure that all objects have been given descriptions.

WI

It may occasionally be useful to check whether all objects in our game have a given property. Here we have a "not for release" section that will run at the start of the game and alert us to any objects lacking description:

"Bic"

Section 1 - Testing descriptions - Not for release

When play begins (this is the run property checks at the start of play rule):

repeat with item running through things:

if description of the item is "":

say "[item] has no description."

Section 2 - Story

The Staff Break Room is a room.

The player carries an orange, a Bic pen, and a napkin. The description of the orange is "It's a small hard pinch-skinned thing from the lunch room, probably with lots of pips and no juice."

The description of the napkin is "Slightly crumpled."

Example **Alpha**

Sometimes we want to let testers of a game insert their own comments during a transcript, without those comments wasting turns of the game or producing lengthy or inappropriate parser errors. Many testers have a habit of prefacing comments with a punctuation mark, so let's say that we'd like to catch any command that starts with any punctuation at all:

"Alpha"

When play begins:

say "Hi, Larry! Thanks for testing my game!!"

Unimplemented Room is a room. "Room description goes here..."

The scary troll is a man in Unimplemented Room.

After reading a command (this is the ignore beta-comments rule):

if the player's command matches the regular expression "^\\p":

say "(Noted.)";

reject the player's command.

Test me with "x me / x troll / !this game is a bit dull so far / kiss troll / ? does this troll do anything? / :yawn".

★ Example Baedeker

Creating a floorplan of the cathedral using the locations from previous examples.

If our map is largely or entirely set inside a single building, we might want to produce something that resembles a floorplan. It's possible to do this with a little tweaking, like so:

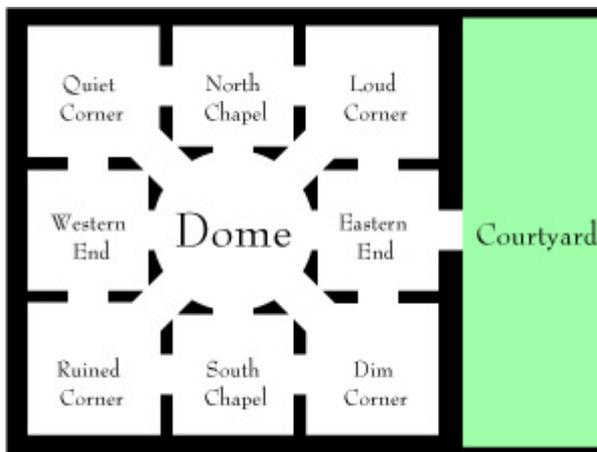
"Baedeker"

Dome is a room. North of Dome is North Chapel. South of the Dome is South Chapel. West of the Dome is Western End. Quiet Corner is northwest of the Dome, north of Western End, and west of North Chapel. Loud Corner is east of North Chapel, northeast of Dome, and north of Eastern End. Eastern End is north of Dim Corner and east of Dome. Dim Corner is southeast of Dome and east of South Chapel. Ruined Corner is southwest of Dome, west of South Chapel, and south of Western End.

The church door is east of Eastern End and west of the Courtyard. The church door is a door.

Index map with
room-shape set to "square" and
room-size set to 60 and
room-name-size set to 9 and
room-name-length set to 13 and
route-thickness set to 20 and
room-outline set to off and
map-outline set to off and
route-colour set to "White" and
room-colour set to "White" and
room-shape of Dome set to "circle" and
room-size of Dome set to 80 and
EPS file.

Now we have a map made of white lines and boxes over a white background, which is not very exciting. If, however, we put a layer of black under this and slightly adjust the room shapes (using an image editor such as Adobe Illustrator), we can produce something that plausibly resembles a floorplan or museum map, like so:



447

★ Example Port Royal 5

Port Royal scenario given instructions for an EPS map.

WI

"1691"

Fort James is a room.

Thames Street End is south of Fort James.

Lime Street is south of Thames Street End. West of Thames Street End is north of Fisher's Row. The description of Fisher's Row is "A waterfront street that runs south towards Chocolata Hole, where the small craft are harboured. It also continues north around the tip of the peninsula from here, turning into the east-west Thames Street."

Water Lane is east of Thames Street End.

East of Water Lane is a room called Thames Street at the Wherry Bridge. Thames Street at the Wherry Bridge has the description "To the southwest is the fishmarket; directly across the street is the entrance to a private alley through a brick archway."

The Fishmarket is southwest of Thames Street at the Wherry Bridge.

The Private Alley is south of Thames Street at the Wherry Bridge.

Thames Street by the King's House is east of Thames Street at the Wherry Bridge.

Thames Street before Fort Carlisle is east of Thames Street by the King's House.

South of Thames Street before Fort Carlisle is a room called Fort Carlisle. The description of Fort Carlisle is "Handsomely arrayed with cannons which you could fire at any moment -- though of course there are ships at dock which might be in the way."

Queen Street End is south of Lime Street.

Queen Street Middle is east of Queen Street End.

Queen Street East is east of Queen Street Middle and south of Private Alley.

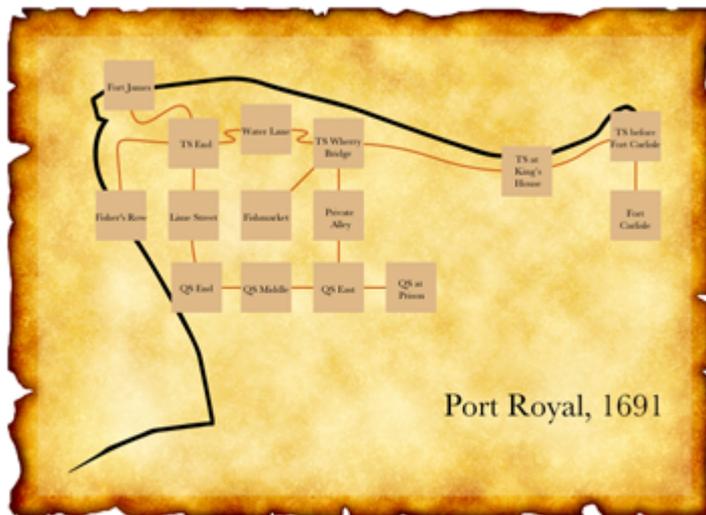
Queen Street at the Prison is east of Queen Street East.

Index map with an EPS file and
Fisher's Row mapped southwest of Thames Street End,
room-size set to 50 and room-name-size set to 10,
room-name-length set to 15,
route-thickness set to 2,
room-outline set to off,
map-outline set to off,
route-colour set to "Chocolate",
room-colour set to "Burly Wood",
title set to "Port Royal, 1691",
font set to "Baskerville",
room-offset of Thames Street by the King's House set to 160&-40,
room-offset of Thames Street before Fort Carlisle set to 210&10,
room-offset of Fort Carlisle set to 210&0,
room-offset of Fort James set to -90&-20,
room-offset of Water Lane set to 0&20,
room-offset of Queen Street End set to 5&0.

This sentence has become a long catalogue of specifications. To break it down: we disambiguate the placement of Thames Street End, which otherwise is hard to locate because the directions to and from the room are not symmetrical. Then we apply some general rules about size, font, and color. Finally, we add instructions about offsetting the room locations of a few specific rooms.

This last part is a bit finicky and will not be necessary in many cases, but our goal this time is to create a map diagram that can be superimposed on the real coastal

outline of Port Royal at the time. With a bit of editing, the result looks like this:



448

★ Example Bay Leaves and Honey Wine

WI

Creating a map of Greece using the locations from previous examples.

The map-maker can be used in quite versatile ways, in short; though the default is a schematic line-and-box affair, that is hardly the only option. While the EPS created is not always the result of our dreams, Inform usually can be made to do most of the hard and boring part, leaving the author to do only a bit of aesthetic touchup.

In many previous examples, we have sent hapless deities wandering around a map of Greece; we might like to chart that for ourselves, in a semi-realistic fashion. So:

"Bay Leaves and Honey Wine"

Corinth is a room. Athens is east of Corinth. Epidaurus is southeast of Corinth and east of Mycenae. Mycenae is south of Corinth. Olympia is west of Mycenae. Argos is south of Mycenae. Thebes is northwest of Athens. Pylos is south of Olympia. Sparta is east of Pylos and south of Argos. Delphi is northwest of Thebes.

Index map with an EPS file and
room-size set to 8,
map-outline set to off,
room-name-offset set to 40&-40,
room-outline set to off,
room-colour set to "White",
route-colour set to "White",
room-name-colour set to "White",
room-name-length set to 25,
room-shape set to "circle".

This produces a line-and-dot map, where the names of rooms do not appear inside the city-circles, but rather (thanks to "room-name-offset") off to one side. We specify a long room-name-length because we want all the names of the cities spelled out in full; and we make all the elements white because we intend to place them over a black background layer.

We can then superimpose this on a vector map of Greece and tweak the exact positions of cities a little by hand (in Adobe Illustrator, as it happens, but other programs would also allow this level of editing). The result:



Alphabetical Index of Examples

-  Examples in Thematic Order
-  Examples in Numerical Order
-  General Index
-  Part I. Writing with Inform
-  Part II. The Inform Recipe Book

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

A&E

AARP-Gnosis

The Abolition of Love

About Inform's regular expression support

About the examples

Accented and exotic letters and symbols

Access All Areas

Actaeon

Actions prohibited in the presence of a dangerous character

Actor's Studio

Adverbs used in commands

Aftershock

Age of Steam

Ahem

Air Conditioning is Standard

Alias

Alien Invasion Part 23

All actions given different durations

All Roads Lead to Mars

Allowing EXAMINE to see multiple objects with a single command

Allowing the player to describe his character before starting play

Alpaca Farm

Alpha

Altering response text programmatically

America Stands Tall

AMUSING menu shown at the endgame

Anchorite

Annoytron Jr

Another, distinct from the player

Ant-Sensitive Sunglasses

Anteaters

Apples

Appointments at future times

Appraisal

The Art of Noise

ASK person TO do something, understood

ASK someone ABOUT an object the same as SHOW

ASKing about a conversation topic other than the one the player typed

Asking the player a Yes/No question

Asking the player to choose a gender

Aspect

Assigning attributes randomly at the start of play

ATTACK and DROP break fragile things

Attitude rules

An automatically closing door
Awarding points for visiting a room

- B**
- Backdrops that move conditionally
 - Backdrops which can only be examined
 - Backus-Naur form for rules
 - Bad Hair Day**
 - Baedeker**
 - Baking soda which has a different name when next to a smelly thing
 - Ballpark**
 - Banner printing at appropriate times
 - Baritone, Bass**
 - Barter Barter**
 - Basic room, container, and supporter descriptions
 - Basic switchable light for a room
 - Basic window similar to a door
 - Battle of Ridgefield**
 - Bay Leaves and Honey Wine**
 - Beachfront**
 - Bee Chambers**
 - Beekeeper's Apprentice**
 - Being Peter**
 - Being Prepared**
 - Belfry**
 - Beneath the Surface**
 - The Best Till Last**
 - Beverage Service**
 - Bibliophilia**
 - Bic**
 - The Big Sainsbury's**
 - Big Sky Country**
 - Bikini Atoll**
 - Blackout**
 - Blanking the status line before play
 - Blankness**
 - Blink**
 - Bogart**
 - Book with numbered pages
 - Book with various contents
 - Bookshelf with numerous books
 - Bosch**
 - Boston Cream**
 - Bottles with removable stoppers
 - Bowler Hats and Baby Geese**
 - Bribery**
 - Broken and unbroken flowerpots
 - Brown**
 - Bruneseau's Journey**
 - Bumping into Walls**
 - BURN command, and flammable objects
 - Buttons**
 - BUY command and a system of money
 - BUY command and money, complete with denominations

- C**
- Cactus Will Outlive Us All**
 - Camp Bethel**
 - Campfire lit
 - Candy**
 - Capital City**
 - Capitalised status line
 - Car which can only be driven on roads

Carnivale

Carousel Room from which exits lead to random locations

Case change for text produced by to say phrases

Casino Banale

Cave Entrance

Cave-troll

Celadon

Centered

Chanel Version 1

Change of Basis

Changing the final question after victory

Channel

1. Television that can be referred to by channel

2. Television with more advanced parsing

Character who learns new actions by watching the player

Characters and objects with titles and special articles

Characters killed off by trigger actions

Characters referred to when absent

Characters renamed in play

Check rules modified for taking something we're inside

Checking for missed properties

Cheese-makers

Chest with a supporting lid

Child who asks if we're there yet

Chronic Hinting Syndrome

Cinco

Circle of Misery

Claims Adjustment

Cloak of Darkness

Clocks that can be SET TO any time

Clothes made of priced fabrics

Clothing kinds

Clothing that layers

Clothing with layering and regions of the body

Clouds with random descriptions

Cloves

Clueless

Colour names for exotic pigments

COMBINE action for connecting multiple objects into one

Command applying to an object added to the story

Command prompt changing to reflect state of combat

Commercial break

Compass directions renamed

Complimentary Peanuts

Concealed pet who would yip at you if it could see you

Concealing clothing for other characters

Contact with objects detected

Containers and supporters given a common kind

Containers referred to by contents

Containers that calculate internal volume and available room

Containers which have a breaking strain

Continuous space with distant objects visible

Continuous space with distant objects visible and automatic room description

Continuous space, simple

Control Center

Controlling the order in which multiple-object actions occur

Conversation based on keyword recognition

Conversation system with recap of past exchanges

Conversation where characters seek logical connection to foregoing topics

Conversation with multiple options for each conversation topic

Copper River

Costa Rican Ornithology

The Cow Exonerated

The Crane's Leg

1. Descriptions based on how objects compare with player's expectations
2. Descriptions based on player-character's observation abilities

Creating our own text variations rules

Creating our own text variations rules, part two

Croft

Crusoe

Curare

CUT and TIE for lengths of string

CUT [something] WITH [something] command

Cycle of day and night scenes

Cyclical randomization of named objects

D

Dangerous objects not to be touched

The Dark Ages Revisited

Dark rooms given non-standard description

A Day For Fresh Sushi

Day One

Deadbolted door unlockable without a key on one side

Dearth and the Maiden

Death message replaced

Deck of cards

Delayed Gratification

Delicious, Delicious Rocks

Democratic Process

Depth

Describing action outcomes using a random choice of verb

Describing action outcomes using a random choice of verb or describing the world state that results

Describing objects with parts

Describing ongoing character behavior with participles

Describing relations using a random choice of verb

Description from inside a vehicle

Device to remember and play back actions

DIAGNOSE command

Dimensions

Dinner is Served

Disambiguating based on properties

Disambiguating body parts

Disambiguation question giving items more explicit names

Disambiguation question worded in a new way entirely

Disenchantment Bay

1. The charter boat

2. Scenery

3. Glacier view

4. Descriptions of things

5. Cabin door and deck

6. Locked-up fishing rods

7. Radar and instruments

8. Pushable chest of ice

9. The Captain

10. Clothing and possessions

11. The player's backpack

12. Complete playable scenario

Disorderly conduct

Do Pass Go

Don Pedro's Revenge

Door described and parsed differently depending on where it is

Door described differently depending on where it is

Door kind that describes the destination

Doors as seen by NPCs

Doors that open automatically

Down Below**Down in Oodville**

Drawers where thing is always in the last opened

DRINKing a potion which then goes away

DROP applies even to objects the player carries indirectly

DROP modified to distinguish between large and small enterable supporters

DROPPING into and onto things

Dubai

Duplicated blue, red and green building blocks

E Early Childhood**Eddystone**

Electric light kind of device

Electrified

An elevator operated simply by walking in and out

An elevator which takes the player to alternate floors

Elsie**Emma****Empire**

Encyclopedia set

Endurance

Enterable objects

Entrapment**Entrevaux****Entropy**

Equation used for price curves

Equipment List**Escape**

Escape action for non-player characters only

Escape from the Seraglio

Every turn an opponent plays

Exact numbers replaced with vague quantifiers

EXAMINE multiple things at once

EXAMINE produces special results for a group of objects

Examining everything at once

Examining the player

An example bouquet of flowers

Exits added to a room

Exits and rooms listed in the status line

Exits only listed in the status line

Extension design advice

Extra Supplies**The Eye of the Idol****Eyes, Fingers, Toes****F Fabrication****The Facts Were These**

Falling into a pit when going from darkness to darkness

Fallout Enclosure**Farewell****Fate Steps In**

Faux-infinite supply of red pens

Feline Behavior**Ferragamo Again****The Fibonacci Sequence****Fido****The Fifth Body****Fifty Times Fifty Ways****Fifty Ways to Leave Your Larva**

Filtering text output in room names

Finality

FIND command
 Finding the bottom of a pile
Fine Laid
Finishing School
 First look text added for a newly entered room
First Name Basis
 Fish of ambiguous species
Flashlight
Flathead News Network
Flotation
 FOLLOW command
Food Network Interactive
 Food with ingredients affecting the player
 Footnote system
For Demonstration Purposes
Fore
 Formal syntax of sentences
Formicidae
Four Cheeses
Four Stars

1. Dark room which relies on other senses when light is off
2. Scope for listening different from scope for seeing

The Fourth Body
Fragment of a Greek Tragedy
Frizz
Frozen Assets
Full Moon
 FULL SCORE using a list of stored actions
 FULL SCORE with achievements table and rankings
Fun with Participles
 Furniture, resentment of
Further Reasons Why All Poets Are Liars

G

Garibaldi

1. Readout of all doors in the game
2. Coloured text styles

Generation X
Get Axe
 GET DOWN and DOWN understood as EXIT
Get Me to the Church on Time
Ginger Beer
 GIVE action for other characters
 GIVE command extended
 GIVE extended to multiple objects at once
 GO BACK command
 GO equivalent to GO OUT
 GO OUT and GO IN determine most appropriate direction if none is defined
 GO TO ROOM command
 GO TO ROOM with automatic door-handling en route
 GO UP and GO DOWN determined by room altitude
Goat-Cheese and Sage Chicken
 GOing an invalid direction produces a list of valid exits
Going Going
Gopher-wood
The Gorge at George
 Graph-theory view of relations
Grilling
Guided Tour

H

The Hang of Thursdays
Happy Hour
Hatless

A Haughty Spirit

Hays Code

Hayseed

Heat death

HELP command is recommended if the player seems lost

HELP command with specific set of topics

HELP menu from extension, with added content

HELP with a simple menu

A hexagonal map grid for movement

Hiding things under other things

Higher Calling

Hint access able to be turned off for the duration of the game

Hints leading the player through conversation

Hints leading the player through physical puzzles

Histories of previous games saved to external file

History Lab

Hohmann Transfer

Horribly heavy box described differently after being taken

Hot Glass Looks Like Cold Glass

Hotel Stechelberg

Hour of the Wren

Hover

HTML-style italic and boldface tags

Hudsucker Industries

A Humble Wayside Flower

Hunger eventually killing the player

Hymenaeus

The Hypnotist of Blois

I I Didn't Come All The Way From Great Portland Street

Ibid.

Identifying nouns by the last thing done to them

Identity Theft

If It Hadn't Been For...

Igpay Atinlay

Implicit takes require time

In Fire or in Flood

Incorporating changeable objects into the body of the room description

Indirection

Infiltration

Instant EXAMINE and LOOK

Interrogation

Introduction to Juggling

INVENTORY handling of properties made more flexible

INVENTORY listing with separate carried and worn items

INVENTORY revised for other characters

Inventory standard response changed

INVENTORY wide and tall

IPA

IQ Test

Ish.

J Jamaica 1688

Joke command added

Jokers Wild

K

Keywords in place of ordinary commands

Kinds given new singular and plural synonyms

Kiwi

Kyoto

L

- Labyrinth of Ghosts**
- Lakeside Living**
- Lanista**
 - 1. Simple Randomized Combat
 - 2. Randomized Combat with Weapons
- Latin Lessons**
- Latris Theon**
- Laura**
- Lead Cuts Paper**
- Lean and Hungry**
- The Left Hand of Autumn**
- Lemonade**
- Leopard-skin**
- Lethal Concentration**
 - 1. Gas diffusion from a source through adjacent rooms
 - 2. Gas diffusion in which gas is heavier than air
- Letters described as a group
- Levels of lighting
- Lies**
 - Light levels variable depending on the number of candles the player has lit
 - Light-filled rooms as a route through a space
 - Lighting and relighting a candle
 - Liquid containers with measured contents
 - Liquid model with large bodies of liquid added
 - Liquids in mixtures, identified by recipe
 - LISTEN command extended
 - LOCK and UNLOCK with altered carrying requirements
- Lollipop Guild**
 - LOOK [direction] command
 - LOOK mentions only currently useful objects
 - LOOK UNDER shows nothing unless the player has a light
- Low Light**
- Lucy**
- Lugubrious Pete's Delicatessen**

M

- M. Melmoth's Duel**
 - Machine that transmutes objects
 - Magnet which picks up nearby metal objects
- Magneto's Revenge**
- The Man of Steel**
- The Man of Steel Excuses Himself**
 - Map region listed in status line
 - Marble chute toy
- Masochism Deli**
 - Matches that set fires
 - Mathematical view of relations
- Matreshka**
- Mattress King**
 - Maze escaped by performing an exact sequence of actions
 - Maze with randomized room links
- Meet Market**
 - Menu of numerical options
- Merlin**
- Meteoric I and II**
- Midsummer Day**
- Mimicry**
- Ming Vase**
- Minimal Movement**
 - Mirror into which the player can look
 - Mirror remembering a room description from the past
- Mirror, Mirror**
- Misadventure**

Missing nouns supplied in commands to characters

Mistress of Animals

Model objects referred to by the thing modeled

Model Shop

Modern Conveniences

Modifying and re-parsing an entered command

Money for Nothing

Morning After

Moving a backdrop during play

Mr. Burns' Repast

MRE

Multiple-object actions given modified descriptions

Murder on the Orient Express

Murderer chosen randomly at start of play

Mutt's Adventure

N

Name of a dog set by the player

Name of player character selected at start of play

Nameless

Names of objects change with player's diction

Narration which changes as the player learns new facts

Narrative Register

Near-infinite supply of pizza

Neighborhood Watch

Nickel and Dimed

Night and Day

The Night Before

Night Sky

Nine AM Appointment

No Place Like Home

No Relation

Noisemaking

Noisy Cricket

North by Northwest

North-northwest and similar directions

Northstar

Novice mode offers suggestions before each prompt

Number Study

Numberless

Numbers in relations

O

Object-based rulebooks

Observation stage after report

Odin

Odyssey

OFFER price FOR command allowing player to bargain

Olfactory Settings

On/off button for devices

One of Those Mornings

One Short Plank

Only You...

Orange Cones

Originals

Otherwise if demonstrated

Otranto

Owen's Law

Oyster Wide Shut

P

Paddington

Pages

Pair of dice for rolling

Palette**Panache**

Paragraphs of flexible descriptions
Parser errors involving the any token
Past tense conditions

Patient Zero**Peeled****Peers**

People changing their titles during play
People commanded to obey
People introduced by relative
People listed as a group, with their possessions afterward
People perform errands and randomly spread disease
People who do new things each time the player looks
People who interact with each other each turn
People who move around a party on their own
People who must be greeted before conversation
People who reject categories of instruction

Persephone

Person capable of reaching through solid objects
Person who answers questions based on a common pool of knowledge
Person who answers what, where, and when questions differently
Person who comments on the player's every action
Person who finds a path to a destination specified by player
Person who follows the player
Person who gets fed up after being asked to do many implausible things
Person who laments broken objects
Person who moves on a pre-determined path
Person who moves randomly
Person who obeys almost all instructions
Person who opens a container before trying to get something from it
Person who picks up treasures
Person who pursues his own goals each turn

Peugeot

Photographs referred to by subject matter
Phrase to repeat a rule
Phrases applying to specific things
Pig Latin for the player's commands

Pine

1. Waking the Princess
2. Chatting with the Princess
3. Flashback scenes described to the Princess
4. Flashback with multiple endings

Pink or Blue**Pizza Prince**

Plank bridge breaks on being crossed when the player is carrying something
Player controls multiple player-characters in turn
Player's erroneous command recorded for later
Player's height randomized at the start of play
Plot summary in place of a score
Pocket added to every jacket
A point for never saving the game
The Pointy Hat of Liminal Transgression

Polarity**Police State****Politics as Usual****Polygons****Port Royal**

1. A landscape from Jamaica, 1691
2. With one-way connections added
3. Divided into regions
4. Character names

5. Producing an EPS map
Postures for sitting, standing, and lying down

Pot of Petunias

POUR that pours liquid on a fire

The Power of the Keys

The Prague Job

Preferences file loaded on replaying

Princess and the Pea

Printed names for complex-named objects

Printing the description of something activity

Priority Lab

Prisoner's Dilemma

The Problem of Edith

Producing an EPS floorplan

Producing an EPS map of Greece

Prolegomena

Properties, checking for

Proposal

Provenance Unknown

Puff of Orange Smoke

Puncak Jaya

PUSH something A DIRECTION modified

PUSH something a direction with extra phrasing

PUSH SOMETHING TO check rules modified

Pushing a box between named internal positions in a room

PUT and INSERT automatically TAKE first

PUT and INSERT automatically TAKE first even with multiples

Q The Queen of Sheba

Query

Questionable Revolutions

Questions asked of the player

Quiz Show

R Radio Daze

Radio producing cycling programming

Radios and other devices active when switched on

Raised supporter kind that conceals its contents

Random distribution at the start of play

Random variations within text

Randomized pedestrian passer-by

rBGH

READ command separate from EXAMINE

Real Adventurers Need No Help

Recorder that records sounds made by player and non-player actions

Red Cross

Red sticky label

Redirecting actions to new objects

Redirecting all actions from one object to another

Reflections

Refusing to understand "hot"

Region off-limits to a player without VIP pass

Regions, traveling from

Relations and their uses

Relations applying to multiple values of the same kind

Relations syntax explored

Relevant Relations

REMEMBER command keyed to topics

Removal

Reordering multiple objects for dramatic effect

Replanting

Reporting rules for other characters' behavior
Responding to questions starting with WHO, WHAT, etc.

Responsive

Restrictions preventing inappropriate behavior
Results of a rule
Resuming play after an accidental death

Revenge of the Fussy Table

Rip Van Winkle

Robo

1. Robot that stores only one program at a time
2. Robot able to store and perform fifteen named scripts

Rock Garden

Rocket Man

Room description changes after first visit
Room description changes after note is read
Room description changes at each of several visits
A room where the game cannot be saved
Rooms player is forced to visit in order
Rooms with sounds of nearby water, illuminated by flickering light
Rooms with sounds of nearby water, illuminated by flickering light
Rooms with visible exteriors
Rope, able to be tied to things and dragged betwixt rooms
An RSS reader connecting to the Internet

Rubies

Rules of Attraction

S

Sackcloth

Safari Guide

Safe that can be opened with a combination
Safe with a multi-number combination

Safety

Saint Eligius

Sand

Sanity-check stage before the Before rules

Savannah

Saying a number in round numbers
Scene that ends as soon as the player has embarrassed himself
Scene which plays through a series of events in order, then ends
Scene which starts and finishes
Scene with random atmospheric events
Scenes which change as the player explores his inventory
Scenes which restrict movement
Scenes with props lists and location properties
Scheduling an eclipse
Scope approaches compared
Scope for large objects visible far away
Scoreboard preserved between games
Scored items listed in a table

Search and Seizure

SEARCH [room] action that opens every visible unlocked container
SEARCHing for an item which moves into play

The Second Oldest Problem

Setting pronouns to respond to sudden changes
SHAKE command

Shawn's Bad Day

Shifting alliances among people
Shiny items are described as reflecting torchlight
Shipboard directions

Shipping Trunk

Shirts constructed from component parts
SHOOT command, and ray gun disintegrating objects
Shops open and close at specific hours

SHOW something without taking it first
 Shuffling the arrangement of items in a container
Sieve of Eratosthenes
 Signpost that can be turned
 Signposts to distant rooms
Signs and Portents
 Single yellow onion in a sack
Situation Room
 Sleep and waking
Slightly Wrong
Slogar's Revenge
Slouching
 Smart window
 Smoke which spreads, not tracking concentration
 Smuggler carrying hidden items
Snip
Solitude
Some Assembly Required
Something Narsty
Space Patrol - Stranded on Jupiter!
The Speed of Thought
Spellbreaker
Spring Cleaning
 Staircase kind of door which diverts the CLIMB command
 Staircase kind of door, always open and unopenable
 Standard IF demonstration
Starry Void
Stately Gardens
 Status line with centered text, the easy way
 Status line with centered text, the hard way
Stone
 Story driven by internal rules
Straw Boater
Straw Into Gold
Streaming
Strictly Ballroom
Studios
 Substituting insults and boasts
 Supporter from which the player cannot take things
 Supporters that don't list contents in room descriptions
Sweeney
Swerve left? Swerve right? Or think about it and die?
Swigmore U.
Sybil

1. ASK, TELL, and ANSWER conflated
2. YES and NO directed to people

T Table of juggling equipment
 TAKE action reports where the noun came from
 TAKE prints a description
 TAKE scenery producing a custom response
 TALK TO command used with scenes
 TALK TO usage corrected
Tamed
Technological Terror
 Telephones reaching distant parties
 Telephones with standard American-length phone numbers
 Telescope allowing view of another room
Tense Boxing
Terracotta
Terracottissima
Terracottissima Maxima

Terror of the Sierra Madre

Testing command

Testing for empty properties

Testing the locale description order

Text Foosball**Thirst****Thirst 2****3 AM**

THROW [something] AT [something] command enhanced

Tilt

1. Deck of cards that can be referred to by name

2. Identifying poker hands

3. Playing cards with colored unicode suits

Time expressed in units other than minutes

Time told in 24-hour military style

Timeless**Tiny Garden**

To say rules for insults and boasts

Today Tomorrow**Tom's Midnight Garden**

Torch understood as flaming or extinguished

Totality**Tour des Maillots****Trachypachidae Maturin 1803**

Train that makes station stops

Transmutations

Travel requiring a vehicle

Travel to a room described

Trieste**The Trouble with Printing****Trying Taking Manhattan**

Turns take 15 minutes each

Turns take a quarter day each

U**U-Stor-It****Unblinking****The Unbuttoned Elevator Affair****Uncommon Ground****Under Contract**

Understanding names of things, basic

Understanding new commands involving compass directions

Understanding text

Understanding units as part of the name of a thing

The Undertomb**The Undertomb 1****The Undertomb 2**

UNDO not mentioned in the final question

The Unexamined Life

Unit conversions

Unpeeled**Unthinkable Alliances****Up and Up****Uptempo****Uptown Girls**

USE action which divines rational behavior for a wide range of possible nouns

Using lists as arrays

Using lists as queues

Using lists as rings

Using lists as sets

Using lists as sieves

Using lists as stacks

- V**
- Van Helsing**
 - Variety**
 - Variety 2**
 - Variety of postures
 - Varying room description text
 - Verbosity**
 1. Brief descriptions by default
 2. Full-length descriptions mandatory
 - Veronica**
 - Versailles**
 - Video camera that records all actions
 - A View of Green Hills**
 - Virtue**
 - Vitrine**
 - Vouvray**
- W**
- Wainwright Acts**
 - WAIT [number] MINUTES command
 - WAIT UNTIL [time] command
 - Waiting Room**
 - Walls and Noses**
 - Wand which reveals a person's concealed possessions
 - Waterskin being depleted
 - Waterworld**
 - Ways Out**
 - Weathering**
 - Weighbridge
 - Wesponses**
 - Western Art History 305**
 - What Makes You Tick**
 - What Not To Wear**
 - When?**
 - Whence?**
 - Whither?**
 - Widget Enterprises**
 - Window connecting two rooms
 - Windows in high places
 - Winds of Change**
 - Witnessed**
 1. Batteries, swappable between devices
 2. EMF Meter
 - Wonderland**
 - Wonka's Revenge**
 - The World of Charles S. Roberts**
 - Would you...?**
 - WRITE IN command that will open a free-text mode
 - WRITE text IN command that will store text
 - WXPQ**
- X**
- Xerxes**
 - Xot**
 - Xylan**
 - XYZZY**
- Y**
- Y ask Y?**
 - Yolk of Gold**
 - You see nothing special... message replaced
 - Your Mother Doesn't Work Here**
- Z**
- Zero**

Zodiac
Zorb
Zork II
Zorn of Zorna
Zqlran Era 8



The Über-complète clavier

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Examples in Numerical Order

- ➔ Examples in Alphabetical Order
 - ➔ Examples in Thematic Order
 - ➔ General Index
 - ↩ Part I. Writing with Inform
 - ↩ Part II. The Inform Recipe Book
-

Chapter 1: Welcome to Inform

§1.1. Preface

- 1 **★ Example About the examples** RB
An explanation of the examples in this documentation, and the asterisks attached to them. Click the heading of the example, or the example number, to reveal the text.
-

Chapter 3: Things

§3.1. Descriptions

- 2 **★ Example Bic** RB
Testing to make sure that all objects have been given descriptions.
- 3 **★ Example Verbosity 1** RB
Making rooms give brief room descriptions when revisited.
- 4 **★★ Example Slightly Wrong** RB
A room whose description changes slightly after our first visit there.

§3.2. Rooms and the map

- 5 **★ Example Port Royal 1** RB
A partial implementation of Port Royal, Jamaica, set before the earthquake of 1692 demolished large portions of the city.
- 6 **★★ Example Up and Up** RB
Adding a short message as the player approaches a room, before the room description itself appears.
- 7 **★★★ Example Starry Void** RB
Creating a booth that can be seen from the outside, opened and closed, and entered as a separate room.

§3.3. One-way connections

- ★ Example Port Royal 2** RB

8

Another part of Port Royal, with less typical map connections.

RB

9

★ Example The Unbuttoned Elevator Affair

A simple elevator connecting two floors which is operated simply by walking in and out, and has no buttons or fancy doors.

RB

§3.4. Regions and the index map

10

★ Example Port Royal 3

Division of Port Royal into regions.

RB

§3.5. Kinds

11

★ Example First Name Basis

Allowing the player to use different synonyms to refer to something.

RB

12

★ Example Midsummer Day

A few sentences laying out a garden together with some things which might be found in it.

RB

§3.6. Either/or properties

13

★ Example Tamed

Examples of a container and a supporter that can be entered, as well as nested rooms.

RB

§3.7. Properties depend on kind

14

★ Example Disenchantment Bay 1

A running example in this chapter, Disenchantment Bay, involves chartering a boat. This is the first step: creating the cabin.

RB

§3.8. Scenery

15

★ Example Disenchantment Bay 2

Disenchantment Bay: creating some of the objects in the cabin's description.

RB

16

★ Example Replanting

Changing the response when the player tries to take something that is scenery.

RB

§3.9. Backdrops

17

★ Example Disenchantment Bay 3

Disenchantment Bay: adding a view of the glacier.

RB

§3.11. Two descriptions of things

18

★ Example Disenchantment Bay 4

Disenchantment Bay: fleshing out the descriptions of things on the boat.

RB

★★ Example Laura

19

Some general advice about creating objects with unusual or awkward names, and a discussion of the use of printed names.

RB

§3.12. Doors

20

★ **Example Disenchantment Bay 5**

Disenchantment Bay: adding the door and the deck to our charter boat.

RB

21

★★ **Example Escape**

Window that can be climbed through or looked through.

RB

22

★★★ **Example Garibaldi 1**

Providing a security readout device by which the player can check on the status of all doors in the game.

RB

§3.13. Locks and keys

23

★ **Example Disenchantment Bay 6**

Disenchantment Bay: locking up the charter boat's fishing rods.

RB

24

★★ **Example Neighborhood Watch**

A locked door that can be locked or unlocked without a key from one side, but not from the other.

RB

§3.14. Devices and descriptions

25

★ **Example Disenchantment Bay 7**

Disenchantment Bay: making the radar and instruments switch on and off.

RB

26

★★ **Example Down Below**

A light switch which makes the room it is in dark or light.

RB

§3.16. Vehicles and pushable things

27

★ **Example Peugeot**

A journey from one room to another that requires the player to be on a vehicle.

RB

28

★★ **Example Disenchantment Bay 8**

Disenchantment Bay: a pushable chest of ice for the boat.

RB

29

★★★ **Example Hover**

Letting the player see a modified room description when he's viewing the place from inside a vehicle.

RB

§3.17. Men, women and animals

30

★ **Example Disenchantment Bay 9**

Disenchantment Bay: enter the charter boat's Captain.

RB

§3.18. Articles and proper names

31  **Example Belfry** RB
You can see a bat, a bell, some woodworm, William Snelson, the sexton's wife, a bellringer and your local vicar here.

32  **Example Gopher-wood** RB
Changing the name of a character in the middle of play, removing the article.

§3.20. Possessions and clothing

33  **Example Disenchantment Bay 10** RB
Disenchantment Bay: things for the player and the characters to wear and carry.

§3.21. The player's holdall

34  **Example Disenchantment Bay 11** RB
Disenchantment Bay: making a holdall of the backpack.

§3.23. Parts of things

35  **Example Fallout Enclosure** RB
Adding an enclosure kind that includes both containers and supporters in order to simplify text that would apply to both.

36  **Example Brown** RB
A red sticky label which can be attached to anything in the game, or removed again.

37  **Example Disenchantment Bay 12** RB
A final trip to Disenchantment Bay: the scenario turned into a somewhat fuller scene, with various features that have not yet been explained.

§3.24. Concealment

38  **Example Search and Seizure** RB
A smuggler who has items, some of which are hidden.

§3.25. The location of something

39  **Example Van Helsing** RB
A character who approaches the player, then follows him from room to room.

§3.26. Directions

40  **Example Prisoner's Dilemma** RB
A button that causes a previously non-existent exit to come into being.

41  **Example The World of Charles S. Roberts** RB
Replacing the ordinary compass bearings with a set of six directions to

impose a hexagonal rather than square grid on the landscape.

42



Example Fore

Understand "fore", "aft", "port", and "starboard", but only when the player is on a vessel.

RB

Chapter 4: Kinds

§4.1. New kinds

43



Example Vouvray

Adding synonyms to an entire kind of thing.

RB

§4.3. Degrees of certainty

44



Example Odin

Replacing "You see nothing special..." with a different default message for looking at something nondescript.

RB

45



Example Something Narsty

A staircase always open and never openable.

RB

§4.4. Plural assertions

46



Example Get Me to the Church on Time

Using kinds of clothing to prevent the player from wearing several pairs of trousers at the same time.

RB

§4.7. New either/or properties

47



Example Change of Basis

Implementing sleeping and wakeful states.

RB

§4.8. New value properties

48



Example Would you...?

Adding new properties to objects, and checking for their presence.

RB

49



Example Straw Boater

Using text properties that apply only to some things and are not defined for others.

RB

§4.9. Using new kinds of value in properties

50



Example The Undertomb 1

A small map of dead ends, in which the sound of an underground river has different strengths in different caves.

RB

51



Example The Undertomb 2

Flickering lantern-light effects added to the Undertomb.

RB

52



Example The Crane's Leg 1

RB

A description text that automatically highlights the ways in which the object differs from a standard member of its kind.

53

☆☆☆ **Example Signs and Portents**

Signpost that points to various destinations, depending on how the player has turned it.

RB

§4.12. *Values that vary*

54

☆☆☆ **Example Real Adventurers Need No Help**

Allowing the player to turn off all access to hints for the duration of a game, in order to avoid the temptation to rely on them overmuch.

RB

§4.14. *Duplicates*

55

☆☆☆ **Example Early Childhood**

A child's set of building blocks, which come in three different colours - red, green and blue - but which can be repainted during play.

RB

§4.15. *Assemblies and body parts*

56

☆ **Example Being Prepared**

A kind for jackets, which always includes a container called a pocket.

RB

57

☆☆ **Example Model Shop**

An "on/off button" which controls whatever device it is part of.

RB

58

☆☆☆ **Example U-Stor-It**

A "chest" kind which consists of a container which has a lid as a supporter.

RB

59

☆☆☆ **Example The Night Before**

Instructing Inform to prefer different interpretations of EXAMINE NOSE, depending on whether the player is alone, in company, or with Rudolph the Red-nosed Reindeer.

RB

Chapter 5: Text

§5.4. *Text with numbers*

60

☆☆☆ **Example Ballpark**

A new "to say" definition which allows the author to say "[a number in round numbers]" and get verbal descriptions like "a couple of" or "a few" as a result.

RB

§5.5. *Text with lists*

61

☆☆ **Example Control Center**

Objects which automatically include a description of their component parts whenever they are examined.

RB

☆☆ **Example Tiny Garden**

62 A lawn made up of several rooms, with part of the description written automatically. RB

§5.6. Text with variations

63  **Example When?** RB
A door whose description says "...leads east" in one place and "...leads west" in the other.

64  **Example Whence?** RB
A kind of door that always automatically describes the direction it opens and what lies on the far side (if that other room has been visited).

65  **Example Persephone** RB
Separate the player's inventory listing into two parts, so that it says "you are carrying..." and then (if the player is wearing anything) "You are also wearing...".

§5.7. Text with random alternatives

66  **Example Radio Daze** RB
A radio that produces a cycle of output using varying text.

67  **Example Camp Bethel** RB
Creating characters who change their behavior from turn to turn, and a survey of other common uses for alternative texts.

§5.8. Line breaks and paragraph breaks

68  **Example Beekeeper's Apprentice** RB
Making the SEARCH command examine all the scenery in the current location.

§5.9. Text with type styles

69  **Example Garibaldi 2** RB
Adding coloured text to the example of door-status readouts.

§5.11. Unicode characters

70  **Example The Über-complète clavier** RB
This example provides a fairly stringent test of exotic lettering.

§5.13. Making new substitutions

71  **Example Fifty Ways to Leave Your Larva** RB
Using text substitution to make characters reply differently under the same circumstances.

72  **Example Fifty Times Fifty Ways** RB
Writing your own rules for how to carry out substitutions.

Chapter 6: Descriptions

§6.4. Defining new adjectives

73

Example Finishing School

The "another" adjective for rules such as "in the presence of another person".

RB

§6.5. Defining adjectives for values

74

Example Only You...

Smoke which spreads through the rooms of the map, but only every other turn.

RB

§6.9. Which and who

75

Example Versailles

A mirror which will reflect some random object in the room.

RB

§6.13. To be able to see and touch

76

Example Lean and Hungry

A thief who will identify and take any valuable thing lying around that he is able to touch.

RB

§6.14. Adjacent rooms and routes through the map

77

Example Mistress of Animals

A person who moves randomly between rooms of the map.

RB

78

Example All Roads Lead to Mars

Layout where the player is allowed to wander any direction he likes, and the map will arrange itself in order so that he finds the correct "next" location.

RB

79

Example Hotel Stechelberg

Signposts such as those provided on hiking paths in the Swiss Alps, which show the correct direction and hiking time to all other locations.

RB

80

Example A View of Green Hills

A LOOK [direction] command which allows the player to see descriptions of the nearby landscape.

RB

81

Example Unblinking

Finding a best route through light-filled rooms only, leaving aside any that might be dark.

RB

§6.15. All, each and every

82

Example Revenge of the Fussy Table

A small game about resentful furniture and inconvenient objects.

RB

§6.16. Counting while comparing

83

☆☆ Example Yolk of Gold

Set of drawers where the item the player seeks is always in the last drawer he opens, regardless of the order of opening.

RB

Chapter 7: Basic Actions*§7.2. Instead rules*

84

☆☆ Example Grilling

A grill, from which the player is not allowed to take anything lest he burn himself.

RB

85

☆☆ Example Bad Hair Day

Change the player's appearance in response to EXAMINE ME.

RB

§7.3. Before rules

86

☆☆☆☆ Example Democratic Process

Make PUT and INSERT commands automatically take objects if the player is not holding them.

RB

87

☆☆☆☆ Example Sand

Extend PUT and INSERT handling to cases where multiple objects are intended at once.

RB

§7.4. Try and try silently

88

☆☆ Example Fine Laid

Making writing that can be separately examined from the paper on which it appears, but which directs all other actions to the paper.

RB

89

☆☆ Example Hayseed

A refinement of our staircase kind which can be climbed.

RB

§7.5. After rules

90

☆☆ Example Morning After

When the player picks something up which he hasn't already examined, the object is described.

RB

§7.6. Reading and talking

91

☆☆ Example Sybil 1

Direct all ASK, TELL, and ANSWER commands to ASK, and accept multiple words for certain cases.

RB

92

☆☆ Example Lucy

Redirecting a question about one topic to ask about another.

RB

93

☆☆☆☆ Example Sybil 2

Making the character understand YES, SAY YES TO CHARACTER, TELL CHARACTER YES, ANSWER YES, and CHARACTER, YES.

RB

94

☆☆☆ Example Costa Rican Ornithology

A fully-implemented book, answering questions from a table of data, and responding to failed consultation with a custom message such as "You flip through the Guide to Central American Birds, but find no reference to penguins."

RB

§7.7. The other four senses

95

☆☆☆ Example The Art of Noise

Things are all assigned their own noise (or silence). Listening to the room in general reports on all the things that are currently audible.

RB

§7.9. All actions and exceptional actions

96

☆☆ Example Zodiac

Several variations on "doing something other than...", demonstrating different degrees of restriction.

RB

§7.10. The noun and the second noun

97

☆☆ Example Ming Vase

ATTACK or DROP break and remove fragile items from play.

RB

§7.12. In the presence of, and when

98

☆☆ Example Beachfront

An item that the player can't interact with until he has found it by searching the scenery.

RB

99

☆☆☆ Example Today Tomorrow

A few notes on "In the presence of" and how it interacts with concealed objects.

RB

§7.13. Going from, going to

100

☆☆ Example Veronica

An effect that occurs only when the player leaves a region entirely.

RB

101

☆☆☆ Example A&E

Using regions to block access to an entire area when the player does not carry a pass, regardless of which entrance he uses.

RB

102

☆☆☆☆ Example Bumping into Walls

Offering the player a list of valid directions if he tries to go in a direction that leads nowhere.

RB

103

☆☆☆☆ Example Polarity

A "go back" command that keeps track of the direction from which the player came, and sends him back.

RB

§7.14. Going by, going through, going with

104

★ Example No Relation

A car which must be turned on before it can be driven, and can only go to roads.

RB

105

★ Example Mattress King

Adding extra phrasing to the action to PUSH something in a direction.

RB

106

★★ Example One Short Plank

A plank bridge which breaks if the player is carrying something when he goes across it. Pushing anything over the bridge is forbidden outright.

RB

107

★★★ Example Provenance Unknown

Allowing something like PUSH TELEVISION EAST to push the cart on which the television rests.

RB

108

★★★★ Example Zorb

Replacing the message the player receives when attempting to push something that isn't pushable, and also to remove the restriction that objects cannot be pushed up or down.

RB

§7.15. Kinds of action

109

★ Example Dearth and the Maiden

Our heroine, fallen among gentleman highwaymen, is restrained by her own modesty and seamliness.

RB

110

★★★ Example Mimicry

People who must be greeted before conversation can begin.

RB

§7.16. Repeated actions

111

★ Example Y ask Y?

Noticing when the player seems to be at a loss, and recommending the use of hints.

RB

112

★★★★ Example A Day For Fresh Sushi

A complete story by Emily Short, called "A Day for Fresh Sushi", rewritten using Inform 7. Noteworthy is the snarky commenter who remarks on everything the player does, but only the first time each action is performed.

RB

Chapter 8: Change*§8.2. Changing the command prompt*

113

★★★★ Example Don Pedro's Revenge

Combat scenario in which the player's footing and position changes from move to move, and the command prompt also changes to reflect that.

RB

§8.3. Changing the status line

114  **Example Politics as Usual** RB
Have the status line indicate the current region of the map.

115  **Example Centered** RB
Replacing the two-part status line with one that centers only the room name at the top of the screen.

§8.4. Change of either/or properties

116  **Example Vitrine** RB
An electrochromic window that becomes transparent or opaque depending on whether it is currently turned on.

§8.5. Change of properties with values

117  **Example Thirst** RB
A waterskin that is depleted as the player drinks from it.

118  **Example Thirst 2** RB
A campfire added to the camp site, which can be lit using tinder.

§8.7. Moving things

119  **Example Meteoric I and II** RB
A meteor in the night sky which is visible from many rooms, so needs to be a backdrop, but which does not appear until 11:31 PM.

§8.8. Moving backdrops

120  **Example Orange Cones** RB
Creating a traffic backdrop that appears in all road rooms except the one in which the player has laid down orange cones.

§8.9. Moving the player

121  **Example Terror of the Sierra Madre** RB
Multiple player characters who take turns controlling the action.

§8.10. Removing things from play

122  **Example Beverage Service** RB
A potion that the player can drink.

123  **Example Spring Cleaning** RB
A character who sulks over objects that the player has broken (and which are now off-stage).

124  **Example Extra Supplies** RB
A supply of red pens from which the player can take another pen only if he doesn't already have one somewhere in the game world.

§8.11. Now...

125 **★ Example Bee Chambers** RB
A maze with directions between rooms randomized at the start of play.

126 **★★ Example Hatless** RB
It's tempting to use "now..." to distribute items randomly at the start of play, but we need to be a little cautious about how we do that.

127 **★★★ Example Technological Terror** RB
A ray gun which destroys objects, leaving their component parts behind.

§8.15. Calling names

128 **★ Example Higher Calling** RB
All doors in the game automatically attempt to open if the player approaches them when they are closed.

§8.18. Randomness

129 **★ Example Do Pass Go** RB
A pair of dice which can be rolled, and are described with their current total when not carried, and have individual scores when examined.

130 **★ Example Lanista 1** RB
Very simple randomized combat in which characters hit one another for a randomized amount of damage.

131 **★ Example Weathering** RB
The automatic weather station atop Mt. Pisgah shows randomly fluctuating temperature, pressure and cloud cover.

132 **★★★ Example Uptown Girls** RB
A stream of random pedestrians who go by the player.

§8.19. Random choices of things

133 **★ Example Candy** RB
One of several identical candies chosen at the start of play to be poisonous.

134 **★ Example Zork II** RB
A "Carousel Room", as in Zork II, where moving in any direction from the room leads (at random) to one of the eight rooms nearby.

Chapter 9: Time

§9.1. When play begins

135 **★ Example Clueless** RB
A murderer for the mystery is selected randomly at the beginning of the game.

§9.2. Awarding points

136

★★★ Example **Mutt's Adventure**

Awarding points for visiting a room for the first time.

RB

137

★★★★ Example **No Place Like Home**

Recording a whole table of scores for specific treasures.

RB

§9.4. When play ends

138

★★★★ Example **Big Sky Country**

Allowing the player to continue play after a fatal accident, but penalizing him by scattering his possessions around the game map.

RB

§9.5. Every turn

139

★★★★ Example **Witnessed 1**

A kind of battery which can be put into different devices, and which will lose power after extended use.

RB

140

★★★★ Example **Text Foosball**

A game of foosball which relies heavily on every-turn rules.

RB

§9.6. The time of day

141

★★★ Example **IPA**

Shops which each have opening and closing hours, so that it is impossible to go in at the wrong times, and the player is kicked out if he overstays his welcome.

RB

§9.7. Telling the time

142

★★ Example **Situation Room**

Printing the time of day in 24-hour time, as in military situations.

RB

§9.11. Future events

143

★★ Example **MRE**

Hunger that eventually kills the player, and foodstuffs that can delay the inevitable by different amounts of time.

RB

144

★★★ Example **Totality**

To schedule an eclipse of the sun, which involves a number of related events.

RB

145

★★★ Example **Empire**

A train which follows a schedule, stopping at a number of different locations.

RB

146

★★★★ Example **Hour of the Wren**

Allowing the player to make an appointment, which is then kept.

RB

§9.12. Actions as conditions

147 **★ Example Night Sky** RB
A room which changes its description depending on whether an object has been examined.

148 **★★★ Example Zero** RB
A box which called "horribly heavy box" after the player has tried to take it the first time.

§9.13. The past and perfect tenses

149 **★ Example Tense Boxing** RB
An overview of all the variations of past and present tenses, and how they might be used.

150 **★★ Example Bruneseau's Journey** RB
A candle which reacts to lighting and blowing actions differently depending on whether it has already been lit once.

151 **★★ Example Elsie** RB
A door that closes automatically one turn after the player opens it.

§9.14. How many times?

152 **★ Example Infiltration** RB
A room whose description changes depending on the number of times the player has visited.

§9.15. How many turns?

153 **★ Example Annoyotron Jr** RB
A child who after a certain period in the car starts asking annoying questions.

Chapter 10: Scenes

§10.2. Creating a scene

154 **★ Example Pine 1** RB
Pine: Using a scene to watch for the solution of a puzzle, however arrived-at by the player.

155 **★★★ Example Entrapment** RB
A scene in which the player is allowed to explore as much as he likes, but another character strolls in as soon as he has gotten himself into an awkward or embarrassing situation.

§10.3. Using the Scene index

156 **★ Example Age of Steam** RB
The railway-station examples so far put together into a short game called "Age of Steam".

§10.4. During scenes

- 157  **Example Full Moon** RB
Random atmospheric events which last the duration of a scene.
- 158  **Example Space Patrol - Stranded on Jupiter!** RB
We'll be back in just a moment, with more exciting adventures of the...
Space Patrol!
- 159  **Example Bowler Hats and Baby Geese** RB
Creating a category of scenes that restrict the player's behavior.
- 160  **Example Day One** RB
A scene which plays through a series of events in order, then ends when
the list of events is exhausted.

§10.5. Linking scenes together

- 161  **Example Pine 2** RB
Pine: Adding a conversation with the princess, in which a basic set of
facts must be covered before the scene is allowed to end.

§10.6. More general linkages

- 162  **Example The Prague Job** RB
Scenes used to provide pacing while the player goes through his
possessions.
- 163  **Example Entrevaux** RB
Organizing the game by scenes, where each scene has a location and
prop lists so that it can be set up automatically.

§10.7. Multiple beginnings and repeats

- 164  **Example Night and Day** RB
Cycling through a sequence of scenes to represent day and night
following one another during a game.
- 165  **Example Pine 3** RB
Pine: Allowing the player to visit aspects of the past in memory and
describe these events to the princess, as a break from the marriage-
proposal scene.

§10.8. Multiple endings

- 166  **Example Panache** RB
Replacing the score with a plot summary that records the events of the
plot, scene by scene.
- 167  **Example Pine 4** RB
Pine: Adding a flashback scene that, instead of repeating endlessly,
repeats only until the Princess has understood the point.

§10.9. Why are scenes designed this way?

168



Example Cheese-makers

Scenes used to control the way a character reacts to conversation and comments, using a TALK TO command.

RB

Chapter 11: Phrases

§11.3. Pattern matching

169



Example Ahem

Writing a phrase, with several variant forms, whose function is to follow a rule several times.

RB

170



Example Ferragamo Again

Using the same phrase to produce different results with different characters.

RB

§11.5. Conditions and questions

171



Example Proposal

Asking the player a yes/no question which he must answer, and another which he may answer or not as he chooses.

RB

§11.7. Begin and end

172



Example Matreshka

A SEARCH [room] action that will open every container the player can see, stopping only when there don't remain any that are closed, unlocked, and openable.

RB

173



Example Princess and the Pea

The player is unable to sleep on a mattress (or stack of mattresses) because the bottom one has something uncomfortable under it.

RB

§11.8. Otherwise

174



Example Numberless

A simple exercise in printing the names of random numbers, comparing the use of "otherwise if...", a switch statement, or a table-based alternative.

RB

§11.10. Repeat

175



Example Wonka's Revenge

A lottery drum which redistributes the tickets inside whenever the player spins it.

RB

§11.11. Repeat running through

176



Example Strictly Ballroom

RB

People who select partners for dance lessons each turn.

§11.14. Phrase options

177

★★★ **Example Equipment List**

Overview of all the phrase options associated with listing, and examples of how to change the inventory list into some other standard formats.

RB

§11.15. Let and temporary variables

178

★ **Example M. Melmoth's Duel**

Three basic ways to inject random or not-so-random variations into text.

RB

§11.16. New conditions, new adjectives

179

★★★★ **Example Owen's Law**

OUT always means "move to an outdoors room, or else to a room with more exits than this one has"; IN always means the opposite.

RB

§11.17. Phrases to decide other things

180

★ **Example Witnessed 2**

A piece of ghost-hunting equipment that responds depending on whether or not the meter is on and a ghost is visible or touchable from the current location.

RB

181

★★★★ **Example A Haughty Spirit**

Windows overlooking lower spaces which will prevent the player from climbing through if the lower space is too far below.

RB

§11.18. The value after and the value before

182

★ **Example Entropy**

All objects in the game have a heat, but if not kept insulated they will tend toward room temperature (and at a somewhat exaggerated rate).

RB

183

★★★★ **Example The Hang of Thursdays**

Turns take a quarter day each, and the game rotates through the days of the week.

RB

Chapter 12: Advanced Actions

§12.3. Giving instructions to other people

184

★ **Example Virtue**

Defining certain kinds of behavior as inappropriate, so that other characters will refuse indignantly to do any such thing.

RB

185

★★★★ **Example Latris Theon**

A person who can accept instructions to go to new destinations and

RB

move towards them according to the most reasonable path.

§12.4. Persuasion

186

★ **Example The Hypnotist of Blois**

A hypnotist who can make people obedient and then set them free again.

RB

187

★ **Example Police State**

Several friends who obey you; a policeman who doesn't (but who takes a dim view of certain kinds of antics).

RB

§12.5. Unsuccessful attempts

188

★★ **Example Generation X**

A person who goes along with the player's instructions, but reluctantly, and will get annoyed after too many repetitions of the same kind of unsuccessful command.

RB

§12.6. Spontaneous actions by other people

189

★ **Example IQ Test**

Introducing Ogg, a person who will unlock and open a container when the player tells him to get something inside.

RB

190

★★★★ **Example Boston Cream**

A fuller implementation of Ogg, giving him a motivation of his own and allowing him to react to the situation created by the player.

RB

§12.7. New actions

191

★ **Example Red Cross**

A DIAGNOSE command which allows the player to check on the health of someone.

RB

192

★★★★ **Example Frizz**

Liquid flows within containers and soaks objects that are not waterproof; any contact with a wet object can dampen our gloves.

RB

193

★★★★ **Example 3 AM**

A shake command which agitates soda and makes items thump around in boxes.

RB

§12.9. Check, carry out, report

194

★ **Example The Dark Ages Revisited**

An electric light kind of device which becomes lit when switched on and dark when switched off.

RB

195

★★ **Example Paddington**

A CUT [something] WITH [something] command which acts differently on different types of objects.

RB

196

**Example Delicious, Delicious Rocks**

Adding a "sanity-check" stage to decide whether an action makes any sense, which occurs before any before rules, implicit taking, or check rules.

RB

197

**Example Noisemaking**

Creating a stage after the report stage of an action, during which other characters may observe and react.

RB

§12.10. Action variables

198

**Example Removal**

TAKE expanded to give responses such as "You take the book from the shelf." or "You pick up the toy from the ground."

RB

199

**Example Further Reasons Why All Poets Are Liars**

The young William Wordsworth, pushing a box about in his room, must struggle to achieve a Romantic point of view.

RB

200

**Example The Second Oldest Problem**

Adapting the going action so that something special can happen when going from a dark room to another dark room.

RB

201

**Example Puff of Orange Smoke**

A system in which every character has a body, which is left behind when the person dies; attempts to do something to the body are redirected to the person while the person is alive.

RB

202

**Example Croft**

Adding special reporting and handling for objects dropped when the player is on a supporter, and special entering rules for moving from one supporter to another.

RB

§12.11. Making actions work for other people

203

**Example The Man of Steel**

An escaping action which means "go to any room you can reach from here", and is only useful to non-player characters.

RB

204

**Example Trying Taking Manhattan**

Replacing the inventory reporting rule with another which does something slightly different.

RB

205

**Example Under Contract**

Creating a person who accepts most instructions and reacts correctly when a request leads implicitly to inappropriate behavior.

RB

§12.12. Check rules for actions by other people

206

**Example Get Axe**

Changing the check rules to try automatically leaving a container before

RB

attempting to take it. (And arranging things so that other people will do likewise.)

207

☆☆☆ **Example Barter Barter**

Allowing characters other than the player to give objects to one another, accounting for the possibility that some items may not be desired by the intended recipients.

RB

§12.13. Report rules for actions by other people

208

☆☆ **Example The Man of Steel Excuses Himself**

Elaborating the report rules to be more interesting than "Clark goes west."

RB

209

☆☆☆ **Example Fate Steps In**

Fate entity which attempts to make things happen, by hook or by crook, including taking preliminary actions to set the player up a bit.

RB

§12.15. Out of world actions

210

☆☆ **Example Spellbreaker**

P. David Lebling's classic "Spellbreaker" (1986) includes a room where the game cannot be saved: here is an Inform implementation.

RB

211

☆☆☆ **Example A point for never saving the game**

In some of the late 1970s "cave crawl" adventure games, an elaborate scoring system might still leave the player perplexed as to why an apparently perfect play-through resulted in a score which was still one point short of the supposed maximum. Why only 349 out of 350? The answer varied, but sometimes the last point was earned by never saving the game - in other words by playing it right through with nothing to guard against mistakes (except perhaps UNDO for the last command), and in one long session.

RB

§12.16. Reaching inside and reaching outside rules

212

☆☆ **Example Carnivale**

An alternative to backdrops when we want something to be visible from a distance but only touchable from one room.

RB

§12.17. Visible vs touchable vs carried

213

☆☆ **Example Eddystone**

Creating new commands involving the standard compass directions.

RB

214

☆☆☆ **Example Slogar's Revenge**

Creating an amulet of tumblers that can be used to lock and unlock things even when it is worn, overriding the usual requirement that keys be carried.

RB

§12.18. Changing reachability

215 **★ Example Magneto's Revenge** RB
Kitty Pryde of the X-Men is able to reach through solid objects, so we might implement her with special powers that the player does not have...

216 **★ Example Waterworld** RB
A backdrop which the player can examine, but cannot interact with in any other way.

217 **★★ Example Dinner is Served** RB
A window between two locations. When the window is open, the player can reach through into the other location; when it isn't, access is barred.

§12.19. Changing visibility

218 **★ Example Flashlight** RB
Visibility set so that looking under objects produces no result unless the player has a light source to shine there (regardless of the light level of the room).

§12.20. Stored actions

219 **★ Example Bosch** RB
Creating a list of actions that will earn the player points, and using this both to change the score and to give FULL SCORE reports.

220 **★ Example Cactus Will Outlive Us All** RB
For every character besides the player, there is an action that will cause that character to wither right up and die.

221 **★★ Example Actor's Studio** RB
A video camera that records actions performed in its presence, and plays them back with time-stamps.

222 **★★★ Example Anteaters** RB
The player carries a gizmo that is able to record actions performed by the player, then force him to repeat them when the gizmo is dropped. This includes storing actions that apply to topics, as in "look up anteater colonies in the guide".

Chapter 13: Relations

§13.2. What sentences are made up from

223 **★★★★ Example Formal syntax of sentences** RB
A more formal description of the sentence grammar used by Inform for both assertions and conditions.

§13.4. To carry, to wear, to have

224 **★ Example Celadon** RB

Using the enclosure relation to let the player drop things which he only indirectly carries.

225

★ **Example Interrogation**

A wand which, when waved, reveals the concealed items carried by people the player can see.

RB

§13.6. Making reciprocal relations

226

★★★ **Example Four Cheeses**

A system of telephones on which the player can call distant persons and have conversations.

RB

§13.7. Relations in groups

227

★ **Example Transmutations**

A machine that turns objects into other, similar objects.

RB

228

★★★ **Example Otranto**

A kind of rope which can be tied to objects and used to anchor the player or drag items from room to room.

RB

§13.9. Defining new assertion verbs

229

★ **Example Unthinkable Alliances**

People are to be grouped into alliances. To kiss someone is to join his or her faction, which may make a grand alliance; to strike them is to give notice of quitting, and to become a lone wolf.

RB

230

★★★ **Example The Unexamined Life**

An adaptive hint system that tracks what the player needs to have seen or to possess in order to solve a given puzzle, and doles out suggestions accordingly. Handles changes in the game state with remarkable flexibility, and allows the player to decide how explicit a nudge he wants at any given moment.

RB

§13.10. Defining new prepositions

231

★ **Example The Abolition of Love**

A thorough exploration of all the kinds of relations established so far, with the syntax to set and unset them.

RB

232

★ **Example Swerve left? Swerve right? Or think about it and die?**

Building a marble chute track in which a dropped marble will automatically roll downhill.

RB

233

★ **Example Beneath the Surface**

An "underlying" relation which adds to the world model the idea of objects hidden under other objects.

RB

234

★★★ **Example Bogart**

Clothing for the player that layers, so that items cannot be taken off in

RB

the wrong order, and the player's inventory lists only the clothing that is currently visible.

§13.11. Indirect relations

235

 **Example The Problem of Edith**

A conversation in which the main character tries to build logical connections between what the player is saying now and what went immediately before.

RB

§13.12. Relations which express conditions

236

 **Example Wainwright Acts**

A technical note about checking the location of door objects when characters other than the player are interacting with them.

RB

237

 **Example A Humble Wayside Flower**

Relations track the relationships between one character and another. Whenever the player meets a relative of someone he already knows, he receives a brief introduction.

RB

§13.13. Relations involving values

238

 **Example Meet Market**

A case in which relations give characters multiple values of the same kind.

RB

239

 **Example For Demonstration Purposes**

A character who learns new actions by watching the player performing them.

RB

§13.14. Relations as values in their own right

240

 **Example Number Study**

The parity and joint magnitude relations explored.

RB

§13.16. What are relations for?

241

 **Example Murder on the Orient Express**

A number of sleuths (the player among them) find themselves aboard the Orient Express, where a murder has taken place, and one of them is apparently the culprit. Naturally they do not agree on whom, but there is physical evidence which may change their minds...

RB

242

 **Example What Not To Wear**

A general-purpose clothing system that handles a variety of different clothing items layered in different combinations over different areas of the body.

RB

243

 **Example Mathematical view of relations**

Some notes on relations from a mathematical point of view, provided only to clarify some technicalities for those who are interested.

RB

244

**Example Graph-theory view of relations**

Some notes on relations from the point of view of graph theory.

RB

Chapter 14: Adaptive Text and Responses*§14.3. More on adapting verbs*

245

**Example Fun with Participles**

Creating dynamic room descriptions that contain sentences such as "Clark is here, wasting time" or "Clark is here, looking around" depending on Clark's idle activity.

RB

246

**Example Variety**

Suppose we want all of our action responses to display some randomized variety. We could do this by laboriously rewriting all of the response texts, but this example demonstrates an alternative.

RB

247

**Example Variety 2**

This builds on the Variety example to add responses such as "You are now carrying the fedora" that describe relations that result from a given verb, as alternate responses.

RB

248

**Example Narrative Register**

Suppose we want all of our action responses to vary depending on some alterable quality of the narrator, so that sometimes they're slangy, sometimes pompous or archaic.

RB

§14.6. Adapting demonstratives and possessives

249

**Example Olfactory Settings**

Some adaptive text for smelling the flowers, or indeed, anything else.

RB

§14.9. Verbs as values

250

**Example History Lab**

We create phrases such as "the box we took" and "the newspaper Clark looked at" based on what has already happened in the story.

RB

251

**Example Relevant Relations**

An example of how to create room descriptions that acknowledge particular relations using their assigned verbs, rather than by the heavily special-cased code used by the standard library.

RB

§14.11. Changing the text of responses

252

**Example Responsive**

Altering the standard inventory text for when the player is carrying nothing.

RB

Chapter 15: Numbers and Equations

§15.2. Numbers and real numbers

253

☆☆☆ **Example Alias**

A telephone with phone numbers of the standard American seven-digit length.

RB

§15.8. Units

254

☆☆ **Example rBGH**

The player character's height is selected randomly at the start of play.

RB

255

☆☆☆ **Example Lethal Concentration 1**

A poisonous gas that spreads from room to room, incapacitating or killing the player when it reaches sufficient levels.

RB

256

☆☆ **Example Wonderland**

Hiking Mount Rainier, with attention to which locations are higher and which lower than the present location.

RB

257

☆☆☆ **Example Lethal Concentration 2**

Poisonous gas again, only this time it sinks.

RB

§15.12. Making the verb "to weigh"

258

☆☆ **Example Dimensions**

This example draws together the previous snippets into a working implementation of the weighbridge.

RB

259

☆☆☆ **Example Lead Cuts Paper**

To give every container a breaking strain, that is, a maximum weight of contents which it can bear - so that to put the lead pig into a paper bag invites disaster.

RB

§15.15. The parts of a number specification

260

☆☆☆ **Example Zqlran Era 8**

Creating an alternative system of time for our game, using new units.

RB

§15.16. Understanding specified numbers

261

☆☆☆ **Example Snip**

A string which can be cut into arbitrary lengths, and then tied back together.

RB

§15.17. Totals

262

☆☆☆ **Example Nickel and Dimed**

A more intricate system of money, this time keeping track of the individual denominations of coins and bills, specifying what gets spent at each transaction, and calculating appropriate change.

RB

§15.18. Equations

- 263  **Example Widget Enterprises** RB
Allowing the player to set a price for a widget on sale, then determining the resulting sales based on consumer demand, and the resulting profit and loss.

§15.19. Arithmetic with units

- 264  **Example Frozen Assets** RB
A treatment of money which keeps track of how much the player has on him, and a BUY command which lets him go shopping.

- 265  **Example Money for Nothing** RB
An OFFER price FOR command, allowing the player to bargain with a flexible seller.

- 266  **Example Lemonade** RB
Containers for liquid which keep track of how much liquid they are holding and of what kind, and allow quantities to be moved from one container to another.

- 267  **Example Savannah** RB
Using the liquid implementation demonstrated in Lemonade for putting out fires.

§15.20. Multiplication of units

- 268  **Example Depth** RB
Receptacles that calculate internal volume and the amount of room available, and cannot be overfilled.

- 269  **Example Fabrication** RB
A system of assembling clothing from a pattern and materials; both the pattern and the different fabrics have associated prices.

- 270  **Example The Speed of Thought** RB
Describing scientifically-measured objects in units more familiar to the casual audience.

Chapter 16: Tables

§16.3. Corresponding entries

- 271  **Example Dubai** RB
An elevator which connects any of 27 floors in a luxury hotel.

§16.6. Repeating through tables

- 272  **Example Port Royal 4** RB
A cell window through which the player can see people who were in Port Royal in the current year of game-time.

§16.9. Blank rows

273

★ Example If It Hadn't Been For...

A sound recording device that records the noises made by player and non-player actions, then plays them back on demand.

RB

§16.10. Adding and removing rows

274

★★ Example Odyssey

A person who follows a path predetermined and stored in a table, and who can be delayed if the player tries to interact with her.

RB

§16.11. Sorting

275

★★ Example Jokers Wild

A deck of cards which can be shuffled and dealt from.

RB

§16.12. Listed in...

276

★★★ Example Noisy Cricket

Implementing liquids that can be mixed, and the components automatically recognized as matching one recipe or another.

RB

§16.13. Topic columns

277

★ Example Merlin

A REMEMBER command which accepts any text and looks up a response in a table of recollections.

RB

278

★★★ Example Questionable Revolutions

An expansion on the previous idea, only this time we store information and let characters answer depending on their expertise in a given area.

RB

279

★★★ Example The Queen of Sheba

Allowing the player to use question words, and using that information to modify the response given by the other character.

RB

§16.14. Another scoring example

280

★★★ Example Goat-Cheese and Sage Chicken

Implementing a FULL SCORE command which lists more information than the regular SCORE command, adding times and rankings, as an extension of the example given in this chapter.

RB

§16.15. Varying which table to look at

281

★★★ Example Farewell

People who respond to conversational gambits, summarize what they said before if asked again, and provide recap of conversation that is past.

RB

*§16.16. Defining things with tables***★★★ Example Sweeney**

282 A conversation where each topic may have multiple questions and answers associated with it, and where a given exchange can lead to new additions to the list. RB

283  **Example Introduction to Juggling** RB
Assortment of equipment defined with price and description, in a table.

§16.18. Table continuations

284  **Example Food Network Interactive** RB
Using a menu system from an extension, but adding our own material to it for this game.

§16.19. Table amendments

285  **Example Trieste** RB
Table amendment to adjust HELP commands provided for the player.

Chapter 17: Understanding

§17.1. Understand

286  **Example Indirection** RB
Renaming the directions of the compass so that "white" corresponds to north, "red" to east, "yellow" to south, and "black" to west.

287  **Example XYZZY** RB
Basics of adding a new command reviewed, for the case of the simple magic word XYZZY.

288  **Example Xylan** RB
Creating a new command that does require an object to be named; and some comments about the choice of vocabulary, in general.

§17.2. New commands for old grammar

289  **Example Alpaca Farm** RB
A generic USE action which behaves sensibly with a range of different objects.

290  **Example Anchorite** RB
By default, Inform understands GET OFF, GET UP, or GET OUT when the player is sitting or standing on an enterable object. We might also want to add GET DOWN and DOWN as exit commands, though.

291  **Example Cloak of Darkness** RB
Implementation of "Cloak of Darkness", a simple example game that for years has been used to demonstrate the features of IF languages.

§17.3. Overriding existing commands

 **Example The Trouble with Printing** RB

292 Making a READ command, distinct from EXAMINE, for legible objects. RB

293 **★★ Example Lanista 2** RB
Randomized combat in which the damage done depends on what weapons the characters are wielding, and in which an ATTACK IT WITH action is created to replace regular attacking. Also folds a new DIAGNOSE command into the system.

§17.4. Standard tokens of grammar

294 **★ Example Shawn's Bad Day** RB
Allowing the player to EXAMINE ALL.

295 **★★★★ Example The Left Hand of Autumn** RB
The possibility of using a [things] token opens up some interesting complications, because we may want actions on multiple items to be reported differently from actions on just one. Here we look at how to make a multiple examination command that describes groups in special ways.

§17.5. The text token

296 **★ Example Ish.** RB
A (very) simple HELP command, using tokens to accept and interpret the player's text whatever it might be.

297 **★★ Example Nameless** RB
ASKing someone about an object rather than about a topic.

§17.6. Actions applying to kinds of value

298 **★ Example Safety** RB
A safe whose dial can be turned with SPIN SAFE TO 1131, and which will open only with the correct combination.

299 **★ Example Tom's Midnight Garden** RB
A clock kind that can be set to any time using "the time understood"; may be turned on and off; and will advance itself only when running. Time on the face is also reported differently depending on whether the clock is analog or digital.

300 **★★ Example Ibid.** RB
A system which allows the author to assign footnotes to descriptions, and permits the player to retrieve them again by number, using "the number understood". Footnotes will automatically number themselves, according to the order in which the player discovers them.

§17.7. Understanding any, understanding rooms

301 **★ Example One of Those Mornings** RB
A FIND command that allows the player to find a lost object anywhere

302

Example Actaeon

A FOLLOW command allowing the player to pursue a person who has just left the room.

RB

§17.9. Understanding kinds of value

303

Example Pages

A book with pages that can be read by number (as in "read page 3 in...") and which accepts relative page references as well (such as "read the last page of...", "read the next page", and so on).

RB

304

Example Down in Oodville

Offering the player a choice of numbered options at certain times, without otherwise interfering with his ability to give regular commands.

RB

305

Example Straw Into Gold

Creating a Rumpelstiltskin character who is always referred to as "dwarf", "guy", "dude", or "man" -- depending on which the player last used -- until the first time the player refers to him as "Rumpelstiltskin".

RB

§17.10. Commands consisting only of nouns

306

Example Misadventure

A going by name command which does respect movement rules, and accepts names of rooms as commands.

RB

307

Example Safari Guide

The same functionality, but making the player continue to move until he reaches his destination or a barrier, handling all openable doors on the way.

RB

§17.11. Understanding values

308

Example Palette

An artist's workshop in which the canvas can be painted in any colour, and where painterly names for pigments ("cerulean") are accepted alongside everyday ones ("blue").

RB

309

Example Baritone, Bass

Letting the player pick a gender (or perhaps other characteristics) before starting play.

RB

§17.13. New tokens

310

Example Lies

Commands to allow the player to lie down in three different ways.

RB

§17.15. Understanding things by their properties

311

Example Aspect

Understanding aspect ratios (a unit) in the names of televisions.

RB

312

★ Example Hymenaeus

Understanding "flaming torch" and "extinguished torch" to refer to torches when lit and unlit.

RB

313

★★ Example Channel 1

Understanding channels (a number) in the names of televisions.

RB

314

★★ Example Terracottissima

The flowerpots once again, but this time arranged so that after the first breakage all undamaged pots are said to be "unbroken", to distinguish them from the others.

RB

315

★★ Example Peers

The peers of the English realm come in six flavours - Baron, Viscount, Earl, Marquess, Duke and Prince - and must always be addressed properly. While a peerage is for life, it may at the royal pleasure be promoted.

RB

316

★★★ Example Channel 2

Understanding channels (a number) in the names of televisions, with more sophisticated parsing of the change channel action.

RB

317

★★★ Example Terracottissima Maxima

Flowerpots with textual names that might change during play.

RB

318

★★★ Example Tilt 1

A deck of cards with fully implemented individual cards, which can be separately drawn and discarded, and referred to by name.

RB

§17.16. Understanding things by their relations

319

★ Example Cinco

A taco shell that can be referred to (when it contains things) in terms of its contents.

RB

320

★ Example Puncak Jaya

When a character is not visible, responding to such commands as EXAMINE PETER and PETER, HELLO with a short note that the person in question is no longer visible.

RB

321

★ Example Whither?

A door whose description says where it leads; and which automatically understands references such as "the west door" and "the east door" depending on which direction it leads from the location.

RB

322

★★ Example Claims Adjustment

An instant camera that spits out photographs of anything the player chooses to take a picture of.

RB

§17.17. Context: understanding when

323

★ Example Quiz Show

In this example by Mike Tarbert, the player can occasionally be quizzed

RB

on random data from a table; the potential answers will only be understood if a question has just been asked.

324

★★ Example Bibliophilia

A bookshelf with a number of books, where the player's command to examine something will be interpreted as an attempt to look up titles if the bookshelf is present, but otherwise given the usual response.

RB

§17.18. Changing the meaning of pronouns

325

★ Example Pot of Petunias

Responding sensibly to a pot of petunias falling from the sky.

RB

§17.19. Does the player mean...

326

★ Example Masochism Deli

Multiple potatoes, with rules to make the player drop the hot potato first and pick it up last.

RB

§17.20. Multiple action processing

327

★★ Example The Best Till Last

Reordering multiple objects for dramatic effect.

RB

328

★★ Example Western Art History 305

Allowing EXAMINE to see multiple objects with a single command.

RB

§17.21. Understanding mistakes

329

★ Example Query

Catching all questions that begin with WHO, WHAT, WHERE, and similar question words, and responding with the instruction to use commands, instead.

RB

330

★ Example The Gorge at George

If the player tries to TALK TO a character, suggest alternative modes of conversation.

RB

331

★★★ Example Hot Glass Looks Like Cold Glass

Responding to references to a property that the player isn't yet allowed to mention (or when not to use "understand as a mistake").

RB

§17.22. Precedence

332

★ Example Some Assembly Required

Building different styles of shirt from component sleeves and collars.

RB

333

★★★★ Example Lakeside Living

Similar to "Lemonade", but with bodies of liquid that can never be depleted, and some adjustments to the "fill" command so that it will automatically attempt to fill from a large liquid source if possible.

RB

Chapter 18: Activities

§18.1. What are activities?

334

★ Example Ant-Sensitive Sunglasses

What are activities good for? Controlling output when we want the same action to be able to produce very flexible text depending on the state of the world -- in this case, making highly variable room description and object description text.

RB

§18.5. New activities

335

★★ Example AARP-Gnosis

An Encyclopedia set which treats volumes in the same place as a single object, but can also be split up.

RB

336

★★★ Example Aftershock

Modifying the rules for examining a device so that all devices have some specific behavior when switched on, which is described at various times.

RB

337

★★★ Example Crusoe

Adding a "printing the description of something" activity.

RB

§18.9. Deciding the concealed possessions of something

338

★ Example Hays Code

Clark Gable in a pin-striped suit and a pink thong.

RB

§18.10. Printing the name of something

339

★ Example Shipping Trunk

A box of baking soda whose name changes to "completely ineffective baking soda" when it is in a container with something that smells funny.

RB

340

★★★ Example Trachypachidae Maturin 1803

Bottles with removable stoppers: when the stopper is in the bottle, the bottle is functionally closed, but the stopper can also be removed and used elsewhere. Descriptions of the bottle reflect its state intelligently.

RB

341

★★★★ Example Chronic Hinting Syndrome

Using name-printing rules to keep track of whether the player knows about objects, and also to highlight things he might want to follow up.

RB

§18.11. Printing the plural name of something

342

★★★ Example Hudsucker Industries

Letters which are described differently as a group, depending on whether the player has read none, some, or all of them, and on whether they are alike or unlike.

RB

§18.12. Printing a number of something

343

★ Example Prolegomena

Replacing precise numbers with "some" or other quantifiers when too many objects are clustered together for the player to count at a glance.

RB

§18.13. Listing contents of something

344

★ Example Unpeeled

Calling an onion "a single yellow onion" when (and only when) it is being listed as the sole content of a room or container.

RB

§18.15. Issuing the response text of something

345

★ Example Wesponses

Parser messages that are delivered with a speech impediment.

RB

§18.16. Printing room description details of something

346

★ Example Rules of Attraction

A magnet which picks up nearby metal objects, and describes itself appropriately in room descriptions and inventory listings, but otherwise goes by its ordinary name.

RB

§18.18. Printing a refusal to act in the dark

347

★★★ Example Zorn of Zorna

Light levels vary depending on the number of candles the player has lit, and this determines whether or not he is able to examine detailed objects successfully.

RB

§18.22. Printing the description of a dark room

348

★★★ Example Hohmann Transfer

Changing the way dark rooms are described to avoid the standard Inform phrasing.

RB

349

★★★ Example Four Stars 1

An elaboration of the idea that when light is absent, the player should be given a description of what he can smell and hear, instead.

RB

§18.23. Constructing the status line

350

★ Example Ways Out

A status line that lists the available exits from the current location.

RB

351

★★★ Example Guided Tour

A status line that lists the available exits from the current location, changing the names of these exits depending on whether the room has been visited or not.

RB

*§18.24. Writing a paragraph about***★ Example Reflections**

352 Emphasizing the reflective quality of shiny objects whenever they are described in the presence of the torch. RB

353 **☆☆ Example Emma**
Social dynamics in which groups of people form and circulate during a party. RB

354 **☆☆☆☆ Example Air Conditioning is Standard**
Uses "writing a paragraph about" to make person and object descriptions that vary considerably depending on what else is going on in the room, including some randomized NPC interactions with objects or with each other. RB

§18.25. Listing nondescript items of something

355 **☆☆ Example Rip Van Winkle**
A simple way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description. RB

356 **☆☆ Example Happy Hour**
Listing visible characters as a group, then giving some followup details in the same paragraph about specific ones. RB

357 **☆☆ Example The Eye of the Idol**
A systematic way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description, and to control whether supporters list their contents or not. RB

§18.26. Printing the locale description of something

358 **☆☆ Example Priority Lab**
A debugging rule useful for checking the priorities of objects about to be listed. RB

§18.27. Choosing notable locale objects for something

359 **☆☆ Example Low Light**
An object that is only visible and manipulable when a bright light fixture is on. RB

360 **☆☆☆☆ Example Casino Banale**
Creating room descriptions and object descriptions that change as the player learns new facts and pieces things together. RB

§18.28. Printing a locale paragraph about

361 **☆☆ Example Kiwi**
Creating a raised supporter kind whose contents the player can't see or take from the ground. RB

362

**Example Copper River**

Manipulating room descriptions so that only interesting items are mentioned, while objects that are present but not currently useful to the player are ignored.

RB

§18.29. Deciding the scope of something

363

**Example Peeled**

Two different approaches to adjusting what the player can interact with, compared.

RB

364

**Example Four Stars 2**

Using "deciding the scope" to change the content of lists such as "the list of audible things which can be touched by the player".

RB

365

**Example Ginger Beer**

A portable magic telescope which allows the player to view items in another room of his choice.

RB

366

**Example Rock Garden**

A simple open landscape where the player can see between rooms and will automatically move to touch things in distant rooms.

RB

367

**Example Stately Gardens**

An open landscape where the player can see landmarks in nearby areas, with somewhat more complex room descriptions than the previous example, and in which we also account for size differences between things seen at a distance.

RB

§18.31. Asking which do you mean

368

**Example Apples**

Prompting the player on how to disambiguate otherwise similar objects.

RB

369

**Example Originals**

Allowing the player to create models of anything in the game world; parsing the name "model [thing]" or even just "[thing]" to refer to these newly-created models; asking "which do you mean, the model [thing] or the actual [thing]" when there is ambiguity.

RB

370

**Example Walls and Noses**

Responding to "EXAMINE WALL" with "In which direction?", and to "EXAMINE NOSE" with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

RB

§18.32. Supplying a missing noun/second noun

371

**Example Latin Lessons**

Supplying missing nouns and second nouns for other characters besides the player.

RB

372

**Example Minimal Movement**

RB

Supplying a default direction for "go", so that "leave", "go", etc., are always interpreted as "out".

§18.33. Reading a command

373

☆☆☆ **Example Cloves**

Accepting adverbs anywhere in a command, registering what the player typed but then cutting them out before interpreting the command.

RB

374

☆☆☆ **Example Fragment of a Greek Tragedy**

Responding to the player's input based on keywords only, and overriding the original parser entirely.

RB

375

☆☆☆ **Example North by Northwest**

Creating additional compass directions between those that already exist (for instance, NNW) -- and dealing with an awkwardness that arises when the player tries to type "north-northwest". The example demonstrates a way around the nine-character limit on parsed words.

RB

376

☆☆☆☆ **Example Complimentary Peanuts**

A character who responds to keywords in the player's instructions and remarks, even if there are other words included.

RB

§18.34. Implicitly taking something

377

☆☆ **Example The Big Sainsbury's**

Making implicit takes add a minute to the clock, just as though the player had typed TAKE THING explicitly.

RB

378

☆☆ **Example Pizza Prince**

Providing a pizza buffet from which the player can take as many pieces as he wants.

RB

379

☆☆☆☆ **Example Lollipop Guild**

Overriding the rules to allow the player to show something to another character without first taking it.

RB

§18.35. Printing a parser error

380

☆☆ **Example WXPQ**

Creating a more sensible parser error than "that noun did not make sense in this context".

RB

381

☆☆☆☆ **Example Xot**

Storing an invalid command to be repeated as text later in the game.

RB

§18.37. Printing the banner text

382

☆☆ **Example Bikini Atoll**

Delaying the banner for later.

RB

§18.38. Printing the player's obituary

383 **★ Example Battle of Ridgefield** RB
Completely replacing the endgame text and stopping the game without giving the player a chance to restart or restore.

384 **★ Example Finality** RB
Not mentioning UNDO in the final set of options.

385 **★ Example Jamaica 1688** RB
Adding a feature to the final question after victory, so that the player can choose to reveal notes about items in the game.

§18.39. Amusing a victorious player

386 **★★ Example Xerxes** RB
Offering the player a menu of things to read after winning the game.

§18.40. Starting the virtual machine

387 **★ Example Blankness** RB
Emptying the status line during the first screen of the game.

Chapter 19: Rulebooks

§19.2. Named rules and rulebooks

388 **★ Example Nine AM Appointment** RB
A WAIT [number] MINUTES command which advances through an arbitrary number of turns.

389 **★★ Example Delayed Gratification** RB
A WAIT UNTIL [time] command which advances until the game clock reaches the correct hour.

§19.3. New rules

390 **★ Example The Crane's Leg 2** RB
A description text generated based on the propensities of the player-character, following different rulebooks for different characters.

391 **★ Example Stone** RB
A soup to which the player can add ingredients, which will have different effects when the player eats.

392 **★★ Example Bribery** RB
A GIVE command that gets rid of Inform's default refusal message in favor of something a bit more sophisticated.

§19.4. Listing rules explicitly

393 **★ Example Saint Eligius** RB
Adding a first look rule that comments on locations when we visit them

for the first time, inserting text after objects are listed but before any "every turn" rules might occur.

394

★ **Example Uptempo**

Adjust time advancement so the game clock moves fifteen minutes each turn.

RB

395

★ **Example Verbosity 2**

Making rooms give full descriptions each time we enter, even if we have visited before, and disallowing player use of BRIEF and SUPERBRIEF.

RB

396

★★ **Example Slouching**

A system of postures allowing the player and other characters to sit, stand, or lie down explicitly or implicitly on a variety of enterable supporters or containers, or in location.

RB

397

★★ **Example Swignore U.**

Adding a new kind of supporter called a perch, where everything dropped lands on the floor.

RB

§19.5. Changing the behaviour of rules

398

★ **Example Access All Areas**

The Pointy Hat of Liminal Transgression allows its wearer to walk clean through closed doors.

RB

§19.7. The preamble of a rule

399

★★★ **Example Backus-Naur form for rules**

The full grammar Inform uses to parse rule definitions, in a standard computer-science notation.

RB

§19.8. New rulebooks

400

★★ **Example Solitude**

Novice mode that prefaces every prompt with a list of possible commands the player could try, and highlights every important word used, to alert players to interactive items in the scenery.

RB

401

★★★ **Example In Fire or in Flood**

A BURN command; flammable objects which light other items in their vicinity and can burn for different periods of time; the possibility of having parts or contents of a flaming item which survive being burnt.

RB

402

★★★★ **Example Patient Zero**

People who wander around the map performing various errands, and in the process spread a disease which only the player can eradicate.

RB

§19.9. Basis of a rulebook

403

★ **Example Flotation**

RB

Objects that can sink or float in a well, depending on their own properties and the state of the surrounding environment.

§19.11. Success and failure

404

Example Kyoto

Expanding the effects of the THROW something AT something command so that objects do make contact with one another.

RB

§19.12. Named outcomes

405

Example Being Peter

A set of rules determining the attitude a character will take when asked about certain topics.

RB

§19.13. Rulebooks producing values

406

Example Feline Behavior

A cat which reacts to whatever items it has handy, returning the result of a rulebook for further processing.

RB

407

Example Tilt 2

A deck of cards with fully implemented individual cards; when the player has a full poker hand, the inventory listing describes the resulting hand accordingly.

RB

§19.15. Two rulebooks used internally

408

Example Electrified

Adding a rule before the basic accessibility rule that will prevent the player from touching electrified objects under the wrong circumstances.

RB

409

Example Timeless

A set of actions which do not take any game time at all.

RB

410

Example Endurance

Giving different actions a range of durations using a time allotment rulebook.

RB

411

Example Escape from the Seraglio

Replacing the usual response to TAKE ALL so that instead of output such as "grapes: Taken. orange: Taken.", Inform produces variable responses in place of "grapes:".

RB

Chapter 20: Advanced Text

§20.4. Upper and lower case letters

412

Example Capital City

To arrange that the location information normally given on the left-hand side of the status line appears in block capitals.

RB

413

★ Example Rocket Man

Using case changes on any text produced by a "to say..." phrase.

RB

§20.6. Regular expression matching

414

★ Example Alpha

Creating a beta-testing command that matches any line starting with punctuation.

RB

415

★ Example About Inform's regular expression support

Some footnotes on Inform's regular expressions, and how they compare to those of other programming languages.

RB

§20.7. Making new text with text substitutions

416

★ Example Identity Theft

Allowing the player to enter a name to be used for the player character during the game.

RB

417

★ Example Mirror, Mirror

The sorcerer's mirror can, when held up high, form an impression of its surroundings which it then preserves.

RB

418

★★★ Example The Cow Exonerated

Creating a class of matches that burn for a time and then go out, with elegant reporting when several matches go out at once.

RB

§20.8. Replacements

419

★ Example Blackout

Filtering the names of rooms printed while in darkness.

RB

420

★ Example Fido

A dog the player can name and un-name at will.

RB

421

★ Example Igpay Atinlay

A pig Latin filter for the player's commands.

RB

422

★★★ Example Mr. Burns' Repast

Letting the player guess types for an unidentifiable fish.

RB

423

★★★ Example Northstar

Making Inform understand ASK JOSH TO TAKE INVENTORY as JOSH, TAKE INVENTORY. This requires us to use a regular expression on the player's command, replacing some of the content.

RB

424

★★★ Example Cave-troll

Determining that the command the player typed is invalid, editing it, and re-examining it to see whether it now reads correctly.

RB

Chapter 21: Lists

§21.3. Saying lists of values

425

★ Example **Oyster Wide Shut**

Replacing Inform's default printing of properties such as "(closed)", "(open and providing light)", etc., with our own, more flexible variation.

RB

§21.5. Building lists

426

★ Example **Robo 1**

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

RB

§21.6. Lists of objects

427

★ Example **What Makes You Tick**

Building a fishing pole from several component parts that the player might put together in any order.

RB

428

★★ Example **Formicidae**

Manipulating the order in which items are handled after TAKE ALL.

RB

§21.9. Accessing entries in a list

429

★★★ Example **Robo 2**

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

RB

§21.10. Lengthening or shortening a list

430

★ Example **Leopard-skin**

A maze that the player can escape if he performs an exact sequence of actions.

RB

431

★★ Example **The Facts Were These**

Creating a variant GIVE action that lets the player give multiple objects simultaneously with commands like GIVE ALL TO ATTENDANT or GIVE THREE DOLLARS TO ATTENDANT or GIVE PIE AND HAT TO ATTENDANT. The attendant accepts the gifts only if their total combined value matches some minimum amount.

RB

§21.11. Variations: arrays, logs, queues, stacks, sets, sieves and rings

432

★ Example **Circle of Misery**

Retrieving items from an airport luggage carousel is such fun, how can we resist simulating it, using a list as a ring buffer?

RB

433

★ Example **Eyes, Fingers, Toes**

A safe with a multi-number combination, meant to be dialed over multiple turns, is implemented using a log of the last three numbers dialed. The log can then be compared to the safe's correct combination.

RB

434

★ Example The Fibonacci Sequence

RB

The modest Leonardo Fibonacci of Pisa will be only too happy to construct his sequence on request, using an array.

435

★ Example I Didn't Come All The Way From Great Portland Street

RB

In this fiendishly difficult puzzle, which may perhaps owe some inspiration to a certain BBC Radio panel game (1967-), a list is used as a set of actions to help enforce the rule that the player must keep going for ten turns without hesitation, repetition, or deviating from the subject on the card.

436

★ Example Lugubrious Pete's Delicatessen

RB

In this evocation of supermarket deli counter life, a list is used as a queue to keep track of who is waiting to be served.

437

★ Example Sieve of Eratosthenes

RB

The haughty Eratosthenes of Cyrene will nevertheless consent to sieve prime numbers on request.

438

★ Example Your Mother Doesn't Work Here

RB

Your hard-working mother uses a list as a stack: urgent tasks are added to the end of the list, interrupting longer-term plans.

Chapter 22: Advanced Phrases*§22.2. Descriptions as values*

439

★ Example Curare

RB

A phrase that chooses and names the least-recently selected item from the collection given, allowing the text to cycle semi-randomly through a group of objects.

Chapter 23: Figures, Sounds and Files*§23.13. Writing and reading tables to external files*

440

★ Example Alien Invasion Part 23

RB

Keeping a preference file that could be loaded by any game in a series.

441

★★ Example Labyrinth of Ghosts

RB

Remembering the fates of all previous explorers of the labyrinth.

442

★★★ Example Rubies

RB

A scoreboard that keeps track of the ten highest-scoring players from one playthrough to the next, adding the player's name if he has done well enough.

*§23.14. Writing, reading and appending text to files***★ Example The Fourth Body**

443 Notebooks in which the player can record assorted notes throughout play. RB

444  **Example The Fifth Body** RB
An expansion on the notebook, allowing the player somewhat more room in which to type his recorded remark.

§23.15. Exchanging files with other programs

445  **Example Flathead News Network** RB
Using external files, together with a simple Unix script running in the background, to provide live news headlines inside a story file.

Chapter 25: Releasing

§25.23. Titling and abbreviation

446  **Example Baedeker** RB
Creating a floorplan of the cathedral using the locations from previous examples.

447  **Example Port Royal 5** RB
Port Royal scenario given instructions for an EPS map.

448  **Example Bay Leaves and Honey Wine** RB
Creating a map of Greece using the locations from previous examples.

Chapter 27: Extensions

§27.5. A simple example extension

449  **Example Modern Conveniences** RB
Exemplifying the kind of source we might use in writing extensions for kitchen and bathroom appliances.

§27.7. Extensions and story file formats

450  **Example Tilt 3** RB
Displaying the card suits from our deck of cards with red and black colored unicode symbols.

§27.15. Defining phrases in Inform 6

451  **Example Pink or Blue** RB
Asking the player to select a gender to begin play.

§27.19. Longer extracts of Inform 6 code

452  **Example Status line with centered text, the hard way** RB
A status line which has only the name of the location, centered.

§27.28. *Segmented substitutions*

453

★ **Example Chanel Version 1**

Making paired italic and boldface tags like those used by HTML for web pages.

RB

§27.30. *To say one of*

454

★ **Example Blink**

Making a "by atmosphere" token, allowing us to design our own text variations such as "[one of]normal[or]gloomy[or]scary[by atmosphere]".

RB

455

★★ **Example Uncommon Ground**

Making a "by viewpoint" token, allowing us to design our own text variations such as "[show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show]" depending on the identity of the player at the moment.

RB

Examples in Thematic Order

- ➔ Examples in Alphabetical Order
 - ➔ Examples in Numerical Order
 - ➔ General Index
 - ↶ Part I. Writing with Inform
 - ↶ Part II. The Inform Recipe Book
-

Chapter 1: How to Use The Recipe Book

§1.1. Preface

- 1 **★ Example About the examples** WI
An explanation of the examples in this documentation, and the asterisks attached to them. Click the heading of the example, or the example number, to reveal the text.
- 12 **★ Example Midsummer Day** WI
A few sentences laying out a garden together with some things which might be found in it.

§1.3. Disenchantment Bay

- 14 **★ Example Disenchantment Bay 1** WI
A running example in this chapter, Disenchantment Bay, involves chartering a boat. This is the first step: creating the cabin.
- 15 **★ Example Disenchantment Bay 2** WI
Disenchantment Bay: creating some of the objects in the cabin's description.
- 17 **★ Example Disenchantment Bay 3** WI
Disenchantment Bay: adding a view of the glacier.
- 18 **★ Example Disenchantment Bay 4** WI
Disenchantment Bay: fleshing out the descriptions of things on the boat.
- 20 **★ Example Disenchantment Bay 5** WI
Disenchantment Bay: adding the door and the deck to our charter boat.
- 23 **★ Example Disenchantment Bay 6** WI
Disenchantment Bay: locking up the charter boat's fishing rods.
- 25 **★ Example Disenchantment Bay 7** WI
Disenchantment Bay: making the radar and instruments switch on and off.
- ★ Example Disenchantment Bay 9** WI

- 30 Disenchantment Bay: enter the charter boat's Captain. WI
- 33 **★ Example Disenchantment Bay 10** WI
Disenchantment Bay: things for the player and the characters to wear and carry.
- 34 **★ Example Disenchantment Bay 11** WI
Disenchantment Bay: making a holdall of the backpack.
- 28 **★★ Example Disenchantment Bay 8** WI
Disenchantment Bay: a pushable chest of ice for the boat.
- 37 **★★★ Example Disenchantment Bay 12** WI
A final trip to Disenchantment Bay: the scenario turned into a somewhat fuller scene, with various features that have not yet been explained.

§1.4. Information Only

- 415 **★ Example About Inform's regular expression support** WI
Some footnotes on Inform's regular expressions, and how they compare to those of other programming languages.
- 223 **★★★ Example Formal syntax of sentences** WI
A more formal description of the sentence grammar used by Inform for both assertions and conditions.
- 243 **★★★ Example Mathematical view of relations** WI
Some notes on relations from a mathematical point of view, provided only to clarify some technicalities for those who are interested.
- 244 **★★★ Example Graph-theory view of relations** WI
Some notes on relations from the point of view of graph theory.
- 399 **★★★ Example Backus-Naur form for rules** WI
The full grammar Inform uses to parse rule definitions, in a standard computer-science notation.

Chapter 2: Adaptive Prose

§2.1. Varying What Is Written

- 71 **★ Example Fifty Ways to Leave Your Larva** WI
Using text substitution to make characters reply differently under the same circumstances.
- 169 **★ Example Ahem** WI
Writing a phrase, with several variant forms, whose function is to follow a rule several times.
- 174 **★ Example Numberless** WI
A simple exercise in printing the names of random numbers, comparing

the use of "otherwise if...", a switch statement, or a table-based alternative.

- 178  **Example M. Melmoth's Duel** WI

Three basic ways to inject random or not-so-random variations into text.
- 249  **Example Olfactory Settings** WI

Some adaptive text for smelling the flowers, or indeed, anything else.
- 252  **Example Responsive** WI

Altering the standard inventory text for when the player is carrying nothing.
- 343  **Example Prolegomena** WI

Replacing precise numbers with "some" or other quantifiers when too many objects are clustered together for the player to count at a glance.
- 345  **Example Wesponses** WI

Parser messages that are delivered with a speech impediment.
- 413  **Example Rocket Man** WI

Using case changes on any text produced by a "to say..." phrase.
- 419  **Example Blackout** WI

Filtering the names of rooms printed while in darkness.
- 439  **Example Curare** WI

A phrase that chooses and names the least-recently selected item from the collection given, allowing the text to cycle semi-randomly through a group of objects.
- 454  **Example Blink** WI

Making a "by atmosphere" token, allowing us to design our own text variations such as "[one of]normal[or]gloomy[or]scary[by atmosphere]".
- 245  **Example Fun with Participles** WI

Creating dynamic room descriptions that contain sentences such as "Clark is here, wasting time" or "Clark is here, looking around" depending on Clark's idle activity.
- 246  **Example Variety** WI

Suppose we want all of our action responses to display some randomized variety. We could do this by laboriously rewriting all of the response texts, but this example demonstrates an alternative.
- 247  **Example Variety 2** WI

This builds on the Variety example to add responses such as "You are now carrying the fedora" that describe relations that result from a given verb, as alternate responses.
- 250  **Example History Lab** WI

We create phrases such as "the box we took" and "the newspaper Clark looked at" based on what has already happened in the story.

251

★★ Example **Relevant Relations**

An example of how to create room descriptions that acknowledge particular relations using their assigned verbs, rather than by the heavily special-cased code used by the standard library.

WI

60

★★★ Example **Ballpark**

A new "to say" definition which allows the author to say "[a number in round numbers]" and get verbal descriptions like "a couple of" or "a few" as a result.

WI

72

★★★ Example **Fifty Times Fifty Ways**

Writing your own rules for how to carry out substitutions.

WI

248

★★★ Example **Narrative Register**

Suppose we want all of our action responses to vary depending on some alterable quality of the narrator, so that sometimes they're slangy, sometimes pompous or archaic.

WI

305

★★★ Example **Straw Into Gold**

Creating a Rumpelstiltskin character who is always referred to as "dwarf", "guy", "dude", or "man" -- depending on which the player last used -- until the first time the player refers to him as "Rumpelstiltskin".

WI

§2.2. *Varying What Is Read*

11

★ Example **First Name Basis**

Allowing the player to use different synonyms to refer to something.

WI

323

★ Example **Quiz Show**

In this example by Mike Tarbert, the player can occasionally be quizzed on random data from a table; the potential answers will only be understood if a question has just been asked.

WI

325

★ Example **Pot of Petunias**

Responding sensibly to a pot of petunias falling from the sky.

WI

19

★★ Example **Laura**

Some general advice about creating objects with unusual or awkward names, and a discussion of the use of printed names.

WI

43

★★ Example **Vouvray**

Adding synonyms to an entire kind of thing.

WI

375

★★ Example **North by Northwest**

Creating additional compass directions between those that already exist (for instance, NNW) -- and dealing with an awkwardness that arises when the player tries to type "north-northwest". The example demonstrates a way around the nine-character limit on parsed words.

WI

§2.3. *Using the Player's Input*

- 421

★ Example Igpay Atinlay

A pig Latin filter for the player's commands.

WI
 - 422

★★ Example Mr. Burns' Repast

Letting the player guess types for an unidentifiable fish.

WI
 - 317

★★★ Example Terracottissima Maxima

Flowerpots with textual names that might change during play.

WI
 - 381

★★★ Example Xot

Storing an invalid command to be repeated as text later in the game.

WI
-

Chapter 3: Place

§3.1. Room Descriptions

- 147

★ Example Night Sky

A room which changes its description depending on whether an object has been examined.

WI
- 152

★ Example Infiltration

A room whose description changes depending on the number of times the player has visited.

WI
- 334

★ Example Ant-Sensitive Sunglasses

What are activities good for? Controlling output when we want the same action to be able to produce very flexible text depending on the state of the world -- in this case, making highly variable room description and object description text.

WI
- 355

★ Example Rip Van Winkle

A simple way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description.

WI
- 358

★ Example Priority Lab

A debugging rule useful for checking the priorities of objects about to be listed.

WI
- 359

★ Example Low Light

An object that is only visible and manipulable when a bright light fixture is on.

WI
- 4

★★ Example Slightly Wrong

A room whose description changes slightly after our first visit there.

WI
- 357

★★★ Example The Eye of the Idol

A systematic way to allow objects in certain places to be described in the room description body text rather than in paragraphs following the room description, and to control whether supporters list their contents or not.

WI

☆☆☆ Example **Copper River**

Manipulating room descriptions so that only interesting items are mentioned, while objects that are present but not currently useful to the player are ignored.

§3.2. Map

5

☆☆ Example **Port Royal 1**

A partial implementation of Port Royal, Jamaica, set before the earthquake of 1692 demolished large portions of the city.

WI

8

☆☆ Example **Port Royal 2**

Another part of Port Royal, with less typical map connections.

WI

10

☆☆ Example **Port Royal 3**

Division of Port Royal into regions.

WI

78

☆☆ Example **All Roads Lead to Mars**

Layout where the player is allowed to wander any direction he likes, and the map will arrange itself in order so that he finds the correct "next" location.

WI

125

☆☆ Example **Bee Chambers**

A maze with directions between rooms randomized at the start of play.

WI

134

☆☆ Example **Zork II**

A "Carousel Room", as in Zork II, where moving in any direction from the room leads (at random) to one of the eight rooms nearby.

WI

286

☆☆ Example **Indirection**

Renaming the directions of the compass so that "white" corresponds to north, "red" to east, "yellow" to south, and "black" to west.

WI

40

☆☆☆ Example **Prisoner's Dilemma**

A button that causes a previously non-existent exit to come into being.

WI

41

☆☆☆ Example **The World of Charles S. Roberts**

Replacing the ordinary compass bearings with a set of six directions to impose a hexagonal rather than square grid on the landscape.

WI

101

☆☆☆ Example **A&E**

Using regions to block access to an entire area when the player does not carry a pass, regardless of which entrance he uses.

WI

§3.3. Position Within Rooms

199

☆☆ Example **Further Reasons Why All Poets Are Liars**

The young William Wordsworth, pushing a box about in his room, must struggle to achieve a Romantic point of view.

WI

7

☆☆☆☆ Example **Starry Void**

Creating a booth that can be seen from the outside, opened and closed, and entered as a separate room.

WI

§3.4. Continuous Spaces and The Outdoors

- 216  **Example Waterworld** WI

A backdrop which the player can examine, but cannot interact with in any other way.
- 62  **Example Tiny Garden** WI

A lawn made up of several rooms, with part of the description written automatically.
- 79  **Example Hotel Stechelberg** WI

Signposts such as those provided on hiking paths in the Swiss Alps, which show the correct direction and hiking time to all other locations.
- 212  **Example Carnivale** WI

An alternative to backdrops when we want something to be visible from a distance but only touchable from one room.
- 213  **Example Eddystone** WI

Creating new commands involving the standard compass directions.
- 366  **Example Rock Garden** WI

A simple open landscape where the player can see between rooms and will automatically move to touch things in distant rooms.
- 80  **Example A View of Green Hills** WI

A LOOK [direction] command which allows the player to see descriptions of the nearby landscape.
- 367  **Example Stately Gardens** WI

An open landscape where the player can see landmarks in nearby areas, with somewhat more complex room descriptions than the previous example, and in which we also account for size differences between things seen at a distance.

§3.5. Doors, Staircases, and Bridges

- 45  **Example Something Narsty** WI

A staircase always open and never openable.
- 63  **Example When?** WI

A door whose description says "...leads east" in one place and "...leads west" in the other.
- 89  **Example Hayseed** WI

A refinement of our staircase kind which can be climbed.
- 128  **Example Higher Calling** WI

All doors in the game automatically attempt to open if the player approaches them when they are closed.
- 236  **Example Wainwright Acts** WI

A technical note about checking the location of door objects when

characters other than the player are interacting with them.

- 321  **Example Whither?** WI
A door whose description says where it leads; and which automatically understands references such as "the west door" and "the east door" depending on which direction it leads from the location.
- 24  **Example Neighborhood Watch** WI
A locked door that can be locked or unlocked without a key from one side, but not from the other.
- 106  **Example One Short Plank** WI
A plank bridge which breaks if the player is carrying something when he goes across it. Pushing anything over the bridge is forbidden outright.
- 151  **Example Elsie** WI
A door that closes automatically one turn after the player opens it.
- 22  **Example Garibaldi 1** WI
Providing a security readout device by which the player can check on the status of all doors in the game.
- 64  **Example Whence?** WI
A kind of door that always automatically describes the direction it opens and what lies on the far side (if that other room has been visited).

§3.6. Windows

- 116  **Example Vitrine** WI
An electrochromic window that becomes transparent or opaque depending on whether it is currently turned on.
- 21  **Example Escape** WI
Window that can be climbed through or looked through.
- 217  **Example Dinner is Served** WI
A window between two locations. When the window is open, the player can reach through into the other location; when it isn't, access is barred.
- 272  **Example Port Royal 4** WI
A cell window through which the player can see people who were in Port Royal in the current year of game-time.
- 181  **Example A Haughty Spirit** WI
Windows overlooking lower spaces which will prevent the player from climbing through if the lower space is too far below.

§3.7. Lighting

- 194  **Example The Dark Ages Revisited** WI
An electric light kind of device which becomes lit when switched on and dark when switched off.

- 312 **★ Example Hymenaeus** WI
 Understanding "flaming torch" and "extinguished torch" to refer to torches when lit and unlit.
- 352 **★ Example Reflections** WI
 Emphasizing the reflective quality of shiny objects whenever they are described in the presence of the torch.
- 363 **★ Example Peeled** WI
 Two different approaches to adjusting what the player can interact with, compared.
- 26 **★★ Example Down Below** WI
 A light switch which makes the room it is in dark or light.
- 51 **★★ Example The Undertomb 2** WI
 Flickering lantern-light effects added to the Undertomb.
- 348 **★★ Example Hohmann Transfer** WI
 Changing the way dark rooms are described to avoid the standard Inform phrasing.
- 81 **★★★ Example Unblinking** WI
 Finding a best route through light-filled rooms only, leaving aside any that might be dark.
- 347 **★★★ Example Zorn of Zorna** WI
 Light levels vary depending on the number of candles the player has lit, and this determines whether or not he is able to examine detailed objects successfully.
- 349 **★★★ Example Four Stars 1** WI
 An elaboration of the idea that when light is absent, the player should be given a description of what he can smell and hear, instead.
- 291 **★★★ Example Cloak of Darkness** WI
 Implementation of "Cloak of Darkness", a simple example game that for years has been used to demonstrate the features of IF languages.

§3.8. Sounds

- 50 **★ Example The Undertomb 1** WI
 A small map of dead ends, in which the sound of an underground river has different strengths in different caves.
- 364 **★ Example Four Stars 2** WI
 Using "deciding the scope" to change the content of lists such as "the list of audible things which can be touched by the player".
- 95 **★★★ Example The Art of Noise** WI
 Things are all assigned their own noise (or silence). Listening to the room in general reports on all the things that are currently audible.

§3.9. Passers-By, Weather and Astronomical Events

- 131 ★ **Example Weathering** WI
The automatic weather station atop Mt. Pisgah shows randomly fluctuating temperature, pressure and cloud cover.
 - 157 ★ **Example Full Moon** WI
Random atmospheric events which last the duration of a scene.
 - 164 ★ **Example Night and Day** WI
Cycling through a sequence of scenes to represent day and night following one another during a game.
 - 144 ★★ **Example Totality** WI
To schedule an eclipse of the sun, which involves a number of related events.
 - 120 ★★★ **Example Orange Cones** WI
Creating a traffic backdrop that appears in all road rooms except the one in which the player has laid down orange cones.
 - 132 ★★★ **Example Uptown Girls** WI
A stream of random pedestrians who go by the player.
-

Chapter 4: Time and Plot

§4.1. The Passage Of Time

- 142 ★ **Example Situation Room** WI
Printing the time of day in 24-hour time, as in military situations.
- 377 ★ **Example The Big Sainsbury's** WI
Making implicit takes add a minute to the clock, just as though the player had typed TAKE THING explicitly.
- 394 ★ **Example Uptempo** WI
Adjust time advancement so the game clock moves fifteen minutes each turn.
- 409 ★ **Example Timeless** WI
A set of actions which do not take any game time at all.
- 410 ★★ **Example Endurance** WI
Giving different actions a range of durations using a time allotment rulebook.
- 183 ★★★ **Example The Hang of Thursdays** WI
Turns take a quarter day each, and the game rotates through the days of the week.
- 260 ★★★ **Example Zqlran Era 8** WI
Creating an alternative system of time for our game, using new units.

§4.2. Scripted Scenes

- 162  **Example The Prague Job** WI
Scenes used to provide pacing while the player goes through his possessions.
- 155  **Example Entrapment** WI
A scene in which the player is allowed to explore as much as he likes, but another character strolls in as soon as he has gotten himself into an awkward or embarrassing situation.
- 159  **Example Bowler Hats and Baby Geese** WI
Creating a category of scenes that restrict the player's behavior.
- 160  **Example Day One** WI
A scene which plays through a series of events in order, then ends when the list of events is exhausted.

§4.3. Event Scheduling

- 141  **Example IPA** WI
Shops which each have opening and closing hours, so that it is impossible to go in at the wrong times, and the player is kicked out if he overstays his welcome.
- 146  **Example Hour of the Wren** WI
Allowing the player to make an appointment, which is then kept.
- 354  **Example Air Conditioning is Standard** WI
Uses "writing a paragraph about" to make person and object descriptions that vary considerably depending on what else is going on in the room, including some randomized NPC interactions with objects or with each other.

§4.4. Scene Changes

- 156  **Example Age of Steam** WI
The railway-station examples so far put together into a short game called "Age of Steam".
- 119  **Example Meteoric I and II** WI
A meteor in the night sky which is visible from many rooms, so needs to be a backdrop, but which does not appear until 11:31 PM.
- 158  **Example Space Patrol - Stranded on Jupiter!** WI
We'll be back in just a moment, with more exciting adventures of the... Space Patrol!
- 163  **Example Entrevaux** WI
Organizing the game by scenes, where each scene has a location and prop lists so that it can be set up automatically.

§4.5. Flashbacks

165

☆☆☆ Example Pine 3

Pine: Allowing the player to visit aspects of the past in memory and describe these events to the princess, as a break from the marriage-proposal scene.

WI

167

☆☆☆ Example Pine 4

Pine: Adding a flashback scene that, instead of repeating endlessly, repeats only until the Princess has understood the point.

WI

§4.6. Plot Management

209

☆☆☆ Example Fate Steps In

Fate entity which attempts to make things happen, by hook or by crook, including taking preliminary actions to set the player up a bit.

WI

Chapter 5: The Viewpoint Character*§5.1. The Human Body*

254

☆☆ Example rBGH

The player character's height is selected randomly at the start of play.

WI

396

☆☆☆ Example Slouching

A system of postures allowing the player and other characters to sit, stand, or lie down explicitly or implicitly on a variety of enterable supporters or containers, or in location.

WI

59

☆☆☆ Example The Night Before

Instructing Inform to prefer different interpretations of EXAMINE NOSE, depending on whether the player is alone, in company, or with Rudolph the Red-nosed Reindeer.

WI

§5.2. Traits Determined By the Player

416

☆☆ Example Identity Theft

Allowing the player to enter a name to be used for the player character during the game.

WI

309

☆☆☆ Example Baritone, Bass

Letting the player pick a gender (or perhaps other characteristics) before starting play.

WI

451

☆☆☆ Example Pink or Blue

Asking the player to select a gender to begin play.

WI

§5.3. Characterization

73

☆☆ Example Finishing School

The "another" adjective for rules such as "in the presence of another person".

WI

☆☆ Example Bad Hair Day

85 Change the player's appearance in response to EXAMINE ME. WI

109  **Example Dearth and the Maiden**
Our heroine, fallen among gentleman highwaymen, is restrained by her own modesty and seamliness. WI

§5.4. Background

277  **Example Merlin**
A REMEMBER command which accepts any text and looks up a response in a table of recollections. WI

301  **Example One of Those Mornings**
A FIND command that allows the player to find a lost object anywhere WI

§5.5. Memory and Knowledge

149  **Example Tense Boxing**
An overview of all the variations of past and present tenses, and how they might be used. WI

320  **Example Puncak Jaya**
When a character is not visible, responding to such commands as EXAMINE PETER and PETER, HELLO with a short note that the person in question is no longer visible. WI

148  **Example Zero**
A box which called "horribly heavy box" after the player has tried to take it the first time. WI

360  **Example Casino Banale**
Creating room descriptions and object descriptions that change as the player learns new facts and pieces things together. WI

§5.6. Viewpoint

390  **Example The Crane's Leg 2**
A description text generated based on the propensities of the player-character, following different rulebooks for different characters. WI

455  **Example Uncommon Ground**
Making a "by viewpoint" token, allowing us to design our own text variations such as "[show to yourself]quaint[to Lolita]thrilling[to everyone else]squalid[end show]" depending on the identity of the player at the moment. WI

52  **Example The Crane's Leg 1**
A description text that automatically highlights the ways in which the object differs from a standard member of its kind. WI

121  **Example Terror of the Sierra Madre**
Multiple player characters who take turns controlling the action. WI

Chapter 6: Commands

§6.3. Modifying Existing Commands

214

☆☆☆ Example Slogar's Revenge

WI

Creating an amulet of tumblers that can be used to lock and unlock things even when it is worn, overriding the usual requirement that keys be carried.

§6.4. Looking

3

★ Example Verbosity 1

WI

Making rooms give brief room descriptions when revisited.

395

★ Example Verbosity 2

WI

Making rooms give full descriptions each time we enter, even if we have visited before, and disallowing player use of BRIEF and SUPERBRIEF.

§6.5. Examining

44

★ Example Odin

WI

Replacing "You see nothing special..." with a different default message for looking at something nondescript.

68

☆☆ Example Beekeeper's Apprentice

WI

Making the SEARCH command examine all the scenery in the current location.

295

☆☆☆ Example The Left Hand of Autumn

WI

The possibility of using a [things] token opens up some interesting complications, because we may want actions on multiple items to be reported differently from actions on just one. Here we look at how to make a multiple examination command that describes groups in special ways.

337

☆☆☆ Example Crusoe

WI

Adding a "printing the description of something" activity.

§6.6. Looking Under and Hiding

98

★ Example Beachfront

WI

An item that the player can't interact with until he has found it by searching the scenery.

172

★ Example Matreshka

WI

A SEARCH [room] action that will open every container the player can see, stopping only when there don't remain any that are closed, unlocked, and openable.

218

★ Example Flashlight

WI

Visibility set so that looking under objects produces no result unless the

player has a light source to shine there (regardless of the light level of the room).

233

★ **Example Beneath the Surface**

An "underlying" relation which adds to the world model the idea of objects hidden under other objects.

WI

§6.7. Inventory

425

★ **Example Oyster Wide Shut**

Replacing Inform's default printing of properties such as "(closed)", "(open and providing light)", etc., with our own, more flexible variation.

WI

177

★★ **Example Equipment List**

Overview of all the phrase options associated with listing, and examples of how to change the inventory list into some other standard formats.

WI

65

★★★ **Example Persephone**

Separate the player's inventory listing into two parts, so that it says "you are carrying..." and then (if the player is wearing anything) "You are also wearing...".

WI

204

★★★ **Example Trying Taking Manhattan**

Replacing the inventory reporting rule with another which does something slightly different.

WI

§6.8. Taking, Dropping, Inserting and Putting

16

★ **Example Replanting**

Changing the response when the player tries to take something that is scenery.

WI

90

★ **Example Morning After**

When the player picks something up which he hasn't already examined, the object is described.

WI

198

★ **Example Removal**

TAKE expanded to give responses such as "You take the book from the shelf." or "You pick up the toy from the ground."

WI

224

★ **Example Celadon**

Using the enclosure relation to let the player drop things which he only indirectly carries.

WI

86

★★★ **Example Democratic Process**

Make PUT and INSERT commands automatically take objects if the player is not holding them.

WI

202

★★★ **Example Croft**

Adding special reporting and handling for objects dropped when the player is on a supporter, and special entering rules for moving from one supporter to another.

WI

379

☆☆☆ Example Lollipop Guild

Overriding the rules to allow the player to show something to another character without first taking it.

WI

87

☆☆☆☆ Example Sand

Extend PUT and INSERT handling to cases where multiple objects are intended at once.

WI

§6.9. Going, Pushing Things in Directions

100

☆☆ Example Veronica

An effect that occurs only when the player leaves a region entirely.

WI

105

☆☆ Example Mattress King

Adding extra phrasing to the action to PUSH something in a direction.

WI

200

☆☆ Example The Second Oldest Problem

Adapting the going action so that something special can happen when going from a dark room to another dark room.

WI

306

☆☆ Example Misadventure

A going by name command which does respect movement rules, and accepts names of rooms as commands.

WI

372

☆☆ Example Minimal Movement

Supplying a default direction for "go", so that "leave", "go", etc., are always interpreted as "out".

WI

393

☆☆ Example Saint Eligius

Adding a first look rule that comments on locations when we visit them for the first time, inserting text after objects are listed but before any "every turn" rules might occur.

WI

6

☆☆☆ Example Up and Up

Adding a short message as the player approaches a room, before the room description itself appears.

WI

256

☆☆☆ Example Wonderland

Hiking Mount Rainier, with attention to which locations are higher and which lower than the present location.

WI

307

☆☆☆ Example Safari Guide

The same functionality, but making the player continue to move until he reaches his destination or a barrier, handling all openable doors on the way.

WI

102

☆☆☆☆ Example Bumping into Walls

Offering the player a list of valid directions if he tries to go in a direction that leads nowhere.

WI

103

☆☆☆☆ Example Polarity

A "go back" command that keeps track of the direction from which the player came, and sends him back.

WI

107 **☆☆☆ Example Provenance Unknown** WI
Allowing something like PUSH TELEVISION EAST to push the cart on which the television rests.

108 **☆☆☆ Example Zorb** WI
Replacing the message the player receives when attempting to push something that isn't pushable, and also to remove the restriction that objects cannot be pushed up or down.

179 **☆☆☆ Example Owen's Law** WI
OUT always means "move to an outdoors room, or else to a room with more exits than this one has"; IN always means the opposite.

§6.10. Entering and Exiting, Sitting and Standing

206 **☆☆ Example Get Axe** WI
Changing the check rules to try automatically leaving a container before attempting to take it. (And arranging things so that other people will do likewise.)

290 **☆☆ Example Anchorite** WI
By default, Inform understands GET OFF, GET UP, or GET OUT when the player is sitting or standing on an enterable object. We might also want to add GET DOWN and DOWN as exit commands, though.

310 **☆☆ Example Lies** WI
Commands to allow the player to lie down in three different ways.

§6.11. Waiting, Sleeping

388 **☆☆ Example Nine AM Appointment** WI
A WAIT [number] MINUTES command which advances through an arbitrary number of turns.

47 **☆☆☆ Example Change of Basis** WI
Implementing sleeping and wakeful states.

389 **☆☆☆ Example Delayed Gratification** WI
A WAIT UNTIL [time] command which advances until the game clock reaches the correct hour.

§6.13. Magic Words

287 **☆☆ Example XYZZY** WI
Basics of adding a new command reviewed, for the case of the simple magic word XYZZY.

§6.14. Remembering, Converting and Combining Actions

88 **☆☆ Example Fine Laid** WI
Making writing that can be separately examined from the paper on which it appears, but which directs all other actions to the paper.

- 92 **★ Example Lucy** WI
 Redirecting a question about one topic to ask about another.
- 220 **★ Example Cactus Will Outlive Us All** WI
 For every character besides the player, there is an action that will cause that character to wither right up and die.
- 430 **★ Example Leopard-skin** WI
 A maze that the player can escape if he performs an exact sequence of actions.
- 435 **★ Example I Didn't Come All The Way From Great Portland Street** WI
 In this fiendishly difficult puzzle, which may perhaps owe some inspiration to a certain BBC Radio panel game (1967-), a list is used as a set of actions to help enforce the rule that the player must keep going for ten turns without hesitation, repetition, or deviating from the subject on the card.
- 222 **★★ Example Anteaters** WI
 The player carries a gizmo that is able to record actions performed by the player, then force him to repeat them when the gizmo is dropped. This includes storing actions that apply to topics, as in "look up anteater colonies in the guide".

§6.15. Actions on Multiple Objects

- 294 **★ Example Shawn's Bad Day** WI
 Allowing the player to EXAMINE ALL.
- 327 **★★ Example The Best Till Last** WI
 Reordering multiple objects for dramatic effect.
- 328 **★★ Example Western Art History 305** WI
 Allowing EXAMINE to see multiple objects with a single command.
- 411 **★★ Example Escape from the Seraglio** WI
 Replacing the usual response to TAKE ALL so that instead of output such as "grapes: Taken. orange: Taken.", Inform produces variable responses in place of "grapes:".
- 428 **★★ Example Formicidae** WI
 Manipulating the order in which items are handled after TAKE ALL.
- 431 **★★ Example The Facts Were These** WI
 Creating a variant GIVE action that lets the player give multiple objects simultaneously with commands like GIVE ALL TO ATTENDANT or GIVE THREE DOLLARS TO ATTENDANT or GIVE PIE AND HAT TO ATTENDANT. The attendant accepts the gifts only if their total combined value matches some minimum amount.

§6.17. Clarification and Correction

- 289

★ Example Alpaca Farm

A generic USE action which behaves sensibly with a range of different objects.

WI
- 368

★ Example Apples

Prompting the player on how to disambiguate otherwise similar objects.

WI
- 380

★ Example WXPQ

Creating a more sensible parser error than "that noun did not make sense in this context".

WI
- 370

★★★ Example Walls and Noses

Responding to "EXAMINE WALL" with "In which direction?", and to "EXAMINE NOSE" with "Whose nose do you mean, Frederica's, Betty's, Wilma's or your own?"

WI
- 424

★★★ Example Cave-troll

Determining that the command the player typed is invalid, editing it, and re-examining it to see whether it now reads correctly.

WI

§6.18. Alternatives To Standard Parsing

- 304

★★★ Example Down in Oodville

Offering the player a choice of numbered options at certain times, without otherwise interfering with his ability to give regular commands.

WI
- 373

★★★ Example Cloves

Accepting adverbs anywhere in a command, registering what the player typed but then cutting them out before interpreting the command.

WI
- 374

★★★ Example Fragment of a Greek Tragedy

Responding to the player's input based on keywords only, and overriding the original parser entirely.

WI

Chapter 7: Other Characters

§7.1. Getting Acquainted

- 31

★ Example Belfry

You can see a bat, a bell, some woodworm, William Snelson, the sexton's wife, a bellringer and your local vicar here.

WI
- 135

★ Example Clueless

A murderer for the mystery is selected randomly at the beginning of the game.

WI
- 238

★ Example Meet Market

A case in which relations give characters multiple values of the same kind.

WI
- 32

★★★ Example Gopher-wood

Changing the name of a character in the middle of play, removing the

WI

article.

315

 **Example Peers**

The peers of the English realm come in six flavours - Baron, Viscount, Earl, Marquess, Duke and Prince - and must always be addressed properly. While a peerage is for life, it may at the royal pleasure be promoted.

WI

237

 **Example A Humble Wayside Flower**

Relations track the relationships between one character and another. Whenever the player meets a relative of someone he already knows, he receives a brief introduction.

WI

§7.2. Liveliness

153

 **Example Annoytron Jr**

A child who after a certain period in the car starts asking annoying questions.

WI

67

 **Example Camp Bethel**

Creating characters who change their behavior from turn to turn, and a survey of other common uses for alternative texts.

WI

76

 **Example Lean and Hungry**

A thief who will identify and take any valuable thing lying around that he is able to touch.

WI

140

 **Example Text Foosball**

A game of foosball which relies heavily on every-turn rules.

WI

§7.3. Reactive Characters

96

 **Example Zodiac**

Several variations on "doing something other than...", demonstrating different degrees of restriction.

WI

154

 **Example Pine 1**

Pine: Using a scene to watch for the solution of a puzzle, however arrived-at by the player.

WI

187

 **Example Police State**

Several friends who obey you; a policeman who doesn't (but who takes a dim view of certain kinds of antics).

WI

38

 **Example Search and Seizure**

A smuggler who has items, some of which are hidden.

WI

197

 **Example Noisemaking**

Creating a stage after the report stage of an action, during which other characters may observe and react.

WI

82

 **Example Revenge of the Fussy Table**

A small game about resentful furniture and inconvenient objects.

WI

112

**Example A Day For Fresh Sushi**

A complete story by Emily Short, called "A Day for Fresh Sushi", rewritten using Inform 7. Noteworthy is the snarky commenter who remarks on everything the player does, but only the first time each action is performed.

WI

§7.4. Barter and Exchange

392

**Example Bribery**

A GIVE command that gets rid of Inform's default refusal message in favor of something a bit more sophisticated.

WI

207

**Example Barter Barter**

Allowing characters other than the player to give objects to one another, accounting for the possibility that some items may not be desired by the intended recipients.

WI

§7.5. Combat and Death

130

**Example Lanista 1**

Very simple randomized combat in which characters hit one another for a randomized amount of damage.

WI

191

**Example Red Cross**

A DIAGNOSE command which allows the player to check on the health of someone.

WI

201

**Example Puff of Orange Smoke**

A system in which every character has a body, which is left behind when the person dies; attempts to do something to the body are redirected to the person while the person is alive.

WI

293

**Example Lanista 2**

Randomized combat in which the damage done depends on what weapons the characters are wielding, and in which an ATTACK IT WITH action is created to replace regular attacking. Also folds a new DIAGNOSE command into the system.

WI

113

**Example Don Pedro's Revenge**

Combat scenario in which the player's footing and position changes from move to move, and the command prompt also changes to reflect that.

WI

127

**Example Technological Terror**

A ray gun which destroys objects, leaving their component parts behind.

WI

§7.6. Getting Started with Conversation

330

**Example The Gorge at George**

If the player tries to TALK TO a character, suggest alternative modes of conversation.

WI

110



Example Mimicry

People who must be greeted before conversation can begin.

WI

§7.7. Saying Simple Things

91



Example Sybil 1

Direct all ASK, TELL, and ANSWER commands to ASK, and accept multiple words for certain cases.

WI

93



Example Sybil 2

Making the character understand YES, SAY YES TO CHARACTER, TELL CHARACTER YES, ANSWER YES, and CHARACTER, YES.

WI

171



Example Proposal

Asking the player a yes/no question which he must answer, and another which he may answer or not as he chooses.

WI

297



Example Nameless

ASKing someone about an object rather than about a topic.

WI

§7.8. Saying Complicated Things

281



Example Farewell

People who respond to conversational gambits, summarize what they said before if asked again, and provide recap of conversation that is past.

WI

282



Example Sweeney

A conversation where each topic may have multiple questions and answers associated with it, and where a given exchange can lead to new additions to the list.

WI

168



Example Cheese-makers

Scenes used to control the way a character reacts to conversation and comments, using a TALK TO command.

WI

376



Example Complimentary Peanuts

A character who responds to keywords in the player's instructions and remarks, even if there are other words included.

WI

§7.10. Character Emotion

405



Example Being Peter

A set of rules determining the attitude a character will take when asked about certain topics.

WI

170



Example Ferragamo Again

Using the same phrase to produce different results with different characters.

WI

§7.11. Character Knowledge and Reasoning



Example Murder on the Orient Express

241

A number of sleuths (the player among them) find themselves aboard the Orient Express, where a murder has taken place, and one of them is apparently the culprit. Naturally they do not agree on whom, but there is physical evidence which may change their minds...

WI

235

☆☆☆ **Example The Problem of Edith**

A conversation in which the main character tries to build logical connections between what the player is saying now and what went immediately before.

WI

278

☆☆☆ **Example Questionable Revolutions**

An expansion on the previous idea, only this time we store information and let characters answer depending on their expertise in a given area.

WI

279

☆☆☆ **Example The Queen of Sheba**

Allowing the player to use question words, and using that information to modify the response given by the other character.

WI

341

☆☆☆ **Example Chronic Hinting Syndrome**

Using name-printing rules to keep track of whether the player knows about objects, and also to highlight things he might want to follow up.

WI

§7.12. Characters Following a Script

426

☆☆ **Example Robo 1**

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

WI

438

☆☆ **Example Your Mother Doesn't Work Here**

Your hard-working mother uses a list as a stack: urgent tasks are added to the end of the list, interrupting longer-term plans.

WI

161

☆☆☆ **Example Pine 2**

Pine: Adding a conversation with the princess, in which a basic set of facts must be covered before the scene is allowed to end.

WI

429

☆☆☆ **Example Robo 2**

A robot which watches and records the player's actions, then tries to repeat them back in the same order when he is switched into play-back mode.

WI

§7.13. Traveling Characters

77

☆☆ **Example Mistress of Animals**

A person who moves randomly between rooms of the map.

WI

39

☆☆ **Example Van Helsing**

A character who approaches the player, then follows him from room to room.

WI

274

☆☆ **Example Odyssey**

A person who follows a path predetermined and stored in a table, and

WI

who can be delayed if the player tries to interact with her.

302

Example Actaeon

A FOLLOW command allowing the player to pursue a person who has just left the room.

WI

185

Example Latris Theon

A person who can accept instructions to go to new destinations and move towards them according to the most reasonable path.

WI

402

Example Patient Zero

People who wander around the map performing various errands, and in the process spread a disease which only the player can eradicate.

WI

§7.14. Obedient Characters

184

Example Virtue

Defining certain kinds of behavior as inappropriate, so that other characters will refuse indignantly to do any such thing.

WI

186

Example The Hypnotist of Blois

A hypnotist who can make people obedient and then set them free again.

WI

371

Example Latin Lessons

Supplying missing nouns and second nouns for other characters besides the player.

WI

188

Example Generation X

A person who goes along with the player's instructions, but reluctantly, and will get annoyed after too many repetitions of the same kind of unsuccessful command.

WI

423

Example Northstar

Making Inform understand ASK JOSH TO TAKE INVENTORY as JOSH, TAKE INVENTORY. This requires us to use a regular expression on the player's command, replacing some of the content.

WI

239

Example For Demonstration Purposes

A character who learns new actions by watching the player performing them.

WI

205

Example Under Contract

Creating a person who accepts most instructions and reacts correctly when a request leads implicitly to inappropriate behavior.

WI

§7.15. Goal-Seeking Characters

189

Example IQ Test

Introducing Ogg, a person who will unlock and open a container when the player tells him to get something inside.

WI

Example The Man of Steel

203 An escaping action which means "go to any room you can reach from here", and is only useful to non-player characters. WI

208 **☆☆ Example The Man of Steel Excuses Himself**
Elaborating the report rules to be more interesting than "Clark goes west." WI

190 **☆☆☆☆ Example Boston Cream**
A fuller implementation of Ogg, giving him a motivation of his own and allowing him to react to the situation created by the player. WI

§7.16. Social Groups

229 **☆ Example Unthinkable Alliances**
People are to be grouped into alliances. To kiss someone is to join his or her faction, which may make a grand alliance; to strike them is to give notice of quitting, and to become a lone wolf. WI

231 **☆ Example The Abolition of Love**
A thorough exploration of all the kinds of relations established so far, with the syntax to set and unset them. WI

436 **☆ Example Lugubrious Pete's Delicatessen**
In this evocation of supermarket deli counter life, a list is used as a queue to keep track of who is waiting to be served. WI

176 **☆☆ Example Strictly Ballroom**
People who select partners for dance lessons each turn. WI

353 **☆☆ Example Emma**
Social dynamics in which groups of people form and circulate during a party. WI

356 **☆☆ Example Happy Hour**
Listing visible characters as a group, then giving some followup details in the same paragraph about specific ones. WI

Chapter 8: Vehicles, Animals and Furniture

§8.1. Bicycles, Cars and Boats

27 **☆ Example Peugeot**
A journey from one room to another that requires the player to be on a vehicle. WI

104 **☆ Example No Relation**
A car which must be turned on before it can be driven, and can only go to roads. WI

49 **☆☆ Example Straw Boater**
Using text properties that apply only to some things and are not defined for others. WI

29  **Example Hover** WI
Letting the player see a modified room description when he's viewing the place from inside a vehicle.

§8.2. Ships, Trains and Elevators

9  **Example The Unbuttoned Elevator Affair** WI
A simple elevator connecting two floors which is operated simply by walking in and out, and has no buttons or fancy doors.

145  **Example Empire** WI
A train which follows a schedule, stopping at a number of different locations.

42  **Example Fore** WI
Understand "fore", "aft", "port", and "starboard", but only when the player is on a vessel.

271  **Example Dubai** WI
An elevator which connects any of 27 floors in a luxury hotel.

§8.3. Animals

406  **Example Feline Behavior** WI
A cat which reacts to whatever items it has handy, returning the result of a rulebook for further processing.

420  **Example Fido** WI
A dog the player can name and un-name at will.

99  **Example Today Tomorrow** WI
A few notes on "In the presence of" and how it interacts with concealed objects.

§8.4. Furniture

13  **Example Tamed** WI
Examples of a container and a supporter that can be entered, as well as nested rooms.

173  **Example Princess and the Pea** WI
The player is unable to sleep on a mattress (or stack of mattresses) because the bottom one has something uncomfortable under it.

361  **Example Kiwi** WI
Creating a raised supporter kind whose contents the player can't see or take from the ground.

432  **Example Circle of Misery** WI
Retrieving items from an airport luggage carousel is such fun, how can we resist simulating it, using a list as a ring buffer?

 **Example Yolk of Gold** WI

- 83 Set of drawers where the item the player seeks is always in the last drawer he opens, regardless of the order of opening. WI
- 397 **★★** **Example Swigmore U.** Adding a new kind of supporter called a perch, where everything dropped lands on the floor. WI
- 58 **★★★** **Example U-Stor-It** A "chest" kind which consists of a container which has a lid as a supporter. WI

§8.5. Kitchen and Bathroom

- 75 **★★** **Example Versailles** A mirror which will reflect some random object in the room. WI
- 449 **★★** **Example Modern Conveniences** Exemplifying the kind of source we might use in writing extensions for kitchen and bathroom appliances. WI

Chapter 9: Props: Food, Clothing, Money, Toys, Books, Electronics

§9.1. Food

- 48 **★** **Example Would you...?** Adding new properties to objects, and checking for their presence. WI
- 133 **★** **Example Candy** One of several identical candies chosen at the start of play to be poisonous. WI
- 143 **★** **Example MRE** Hunger that eventually kills the player, and foodstuffs that can delay the inevitable by different amounts of time. WI
- 391 **★** **Example Stone** A soup to which the player can add ingredients, which will have different effects when the player eats. WI
- 196 **★★★** **Example Delicious, Delicious Rocks** Adding a "sanity-check" stage to decide whether an action makes any sense, which occurs before any before rules, implicit taking, or check rules. WI

§9.2. Bags, Bottles, Boxes and Safes

- 298 **★** **Example Safety** A safe whose dial can be turned with SPIN SAFE TO 1131, and which will open only with the correct combination. WI
- 319 **★** **Example Cinco** A taco shell that can be referred to (when it contains things) in terms of WI

its contents.

339

★ **Example Shipping Trunk**

A box of baking soda whose name changes to "completely ineffective baking soda" when it is in a container with something that smells funny.

WI

344

★ **Example Unpeeled**

Calling an onion "a single yellow onion" when (and only when) it is being listed as the sole content of a room or container.

WI

433

★ **Example Eyes, Fingers, Toes**

A safe with a multi-number combination, meant to be dialed over multiple turns, is implemented using a log of the last three numbers dialed. The log can then be compared to the safe's correct combination.

WI

340

★★ **Example Trachypachidae Maturin 1803**

Bottles with removable stoppers: when the stopper is in the bottle, the bottle is functionally closed, but the stopper can also be removed and used elsewhere. Descriptions of the bottle reflect its state intelligently.

WI

342

★★ **Example Hudsucker Industries**

Letters which are described differently as a group, depending on whether the player has read none, some, or all of them, and on whether they are alike or unlike.

WI

35

★★★ **Example Fallout Enclosure**

Adding an enclosure kind that includes both containers and supporters in order to simplify text that would apply to both.

WI

§9.3. Clothing

56

★ **Example Being Prepared**

A kind for jackets, which always includes a container called a pocket.

WI

332

★ **Example Some Assembly Required**

Building different styles of shirt from component sleeves and collars.

WI

338

★ **Example Hays Code**

Clark Gable in a pin-striped suit and a pink thong.

WI

242

★★ **Example What Not To Wear**

A general-purpose clothing system that handles a variety of different clothing items layered in different combinations over different areas of the body.

WI

46

★★★ **Example Get Me to the Church on Time**

Using kinds of clothing to prevent the player from wearing several pairs of trousers at the same time.

WI

234

★★★ **Example Bogart**

Clothing for the player that layers, so that items cannot be taken off in the wrong order, and the player's inventory lists only the clothing that is currently visible.

WI

§9.4. Money

- 263 **★ Example Widget Enterprises** WI
Allowing the player to set a price for a widget on sale, then determining the resulting sales based on consumer demand, and the resulting profit and loss.
- 264 **★ Example Frozen Assets** WI
A treatment of money which keeps track of how much the player has on him, and a BUY command which lets him go shopping.
- 265 **★★ Example Money for Nothing** WI
An OFFER price FOR command, allowing the player to bargain with a flexible seller.
- 269 **★★★ Example Fabrication** WI
A system of assembling clothing from a pattern and materials; both the pattern and the different fabrics have associated prices.
- 262 **★★★★ Example Nickel and Dimed** WI
A more intricate system of money, this time keeping track of the individual denominations of coins and bills, specifying what gets spent at each transaction, and calculating appropriate change.
- 283 **★★★★★ Example Introduction to Juggling** WI
Assortment of equipment defined with price and description, in a table.

§9.5. Dice and Playing Cards

- 129 **★ Example Do Pass Go** WI
A pair of dice which can be rolled, and are described with their current total when not carried, and have individual scores when examined.
- 175 **★ Example Wonka's Revenge** WI
A lottery drum which redistributes the tickets inside whenever the player spins it.
- 275 **★★ Example Jokers Wild** WI
A deck of cards which can be shuffled and dealt from.
- 318 **★★★ Example Tilt 1** WI
A deck of cards with fully implemented individual cards, which can be separately drawn and discarded, and referred to by name.
- 407 **★★★★ Example Tilt 2** WI
A deck of cards with fully implemented individual cards; when the player has a full poker hand, the inventory listing describes the resulting hand accordingly.

§9.6. Reading Matter

- 292 **★ Example The Trouble with Printing** WI
Making a READ command, distinct from EXAMINE, for legible

objects.

303

★ **Example Pages**

A book with pages that can be read by number (as in "read page 3 in...") and which accepts relative page references as well (such as "read the last page of...", "read the next page", and so on).

WI

443

★ **Example The Fourth Body**

Notebooks in which the player can record assorted notes throughout play.

WI

324

★★ **Example Bibliophilia**

A bookshelf with a number of books, where the player's command to examine something will be interpreted as an attempt to look up titles if the bookshelf is present, but otherwise given the usual response.

WI

335

★★ **Example AARP-Gnosis**

An Encyclopedia set which treats volumes in the same place as a single object, but can also be split up.

WI

444

★★ **Example The Fifth Body**

An expansion on the notebook, allowing the player somewhat more room in which to type his recorded remark.

WI

94

★★★ **Example Costa Rican Ornithology**

A fully-implemented book, answering questions from a table of data, and responding to failed consultation with a custom message such as "You flip through the Guide to Central American Birds, but find no reference to penguins."

WI

§9.7. Painting and Labeling Devices

308

★ **Example Palette**

An artist's workshop in which the canvas can be painted in any colour, and where painterly names for pigments ("cerulean") are accepted alongside everyday ones ("blue").

WI

36

★★★ **Example Brown**

A red sticky label which can be attached to anything in the game, or removed again.

WI

55

★★★ **Example Early Childhood**

A child's set of building blocks, which come in three different colours - red, green and blue - but which can be repainted during play.

WI

§9.8. Simple Machines

61

★ **Example Control Center**

Objects which automatically include a description of their component parts whenever they are examined.

WI

427

★ **Example What Makes You Tick**

Building a fishing pole from several component parts that the player

WI

might put together in any order.

57

Example Model Shop

An "on/off button" which controls whatever device it is part of.

WI

53

Example Signs and Portents

Signpost that points to various destinations, depending on how the player has turned it.

WI

§9.9. Televisions and Radios

66

Example Radio Daze

A radio that produces a cycle of output using varying text.

WI

311

Example Aspect

Understanding aspect ratios (a unit) in the names of televisions.

WI

313

Example Channel 1

Understanding channels (a number) in the names of televisions.

WI

316

Example Channel 2

Understanding channels (a number) in the names of televisions, with more sophisticated parsing of the change channel action.

WI

336

Example Aftershock

Modifying the rules for examining a device so that all devices have some specific behavior when switched on, which is described at various times.

WI

§9.10. Telephones

226

Example Four Cheeses

A system of telephones on which the player can call distant persons and have conversations.

WI

253

Example Alias

A telephone with phone numbers of the standard American seven-digit length.

WI

§9.11. Clocks and Scientific Instruments

180

Example Witnessed 2

A piece of ghost-hunting equipment that responds depending on whether or not the meter is on and a ghost is visible or touchable from the current location.

WI

299

Example Tom's Midnight Garden

A clock kind that can be set to any time using "the time understood"; may be turned on and off; and will advance itself only when running. Time on the face is also reported differently depending on whether the clock is analog or digital.

WI

Example Ginger Beer

365 A portable magic telescope which allows the player to view items in another room of his choice. WI

§9.12. Cameras and Recording Devices

273 **★ Example If It Hadn't Been For...** WI
A sound recording device that records the noises made by player and non-player actions, then plays them back on demand.

369 **★ Example Originals** WI
Allowing the player to create models of anything in the game world; parsing the name "model [thing]" or even just "[thing]" to refer to these newly-created models; asking "which do you mean, the model [thing] or the actual [thing]" when there is ambiguity.

417 **★ Example Mirror, Mirror** WI
The sorcerer's mirror can, when held up high, form an impression of its surroundings which it then preserves.

221 **★★ Example Actor's Studio** WI
A video camera that records actions performed in its presence, and plays them back with time-stamps.

322 **★★ Example Claims Adjustment** WI
An instant camera that spits out photographs of anything the player chooses to take a picture of.

Chapter 10: Physics: Substances, Ropes, Energy and Weight

§10.1. Gases

255 **★★ Example Lethal Concentration 1** WI
A poisonous gas that spreads from room to room, incapacitating or killing the player when it reaches sufficient levels.

74 **★★★ Example Only You...** WI
Smoke which spreads through the rooms of the map, but only every other turn.

257 **★★★ Example Lethal Concentration 2** WI
Poisonous gas again, only this time it sinks.

§10.2. Liquids

117 **★ Example Thirst** WI
A waterskin that is depleted as the player drinks from it.

122 **★ Example Beverage Service** WI
A potion that the player can drink.

403 **★ Example Flotation** WI
Objects that can sink or float in a well, depending on their own

properties and the state of the surrounding environment.

288  **Example Xylan** WI
Creating a new command that does require an object to be named; and some comments about the choice of vocabulary, in general.

192  **Example Frizz** WI
Liquid flows within containers and soaks objects that are not waterproof; any contact with a wet object can dampen our gloves.

193  **Example 3 AM** WI
A shake command which agitates soda and makes items thump around in boxes.

266  **Example Lemonade** WI
Containers for liquid which keep track of how much liquid they are holding and of what kind, and allow quantities to be moved from one container to another.

267  **Example Savannah** WI
Using the liquid implementation demonstrated in Lemonade for putting out fires.

276  **Example Noisy Cricket** WI
Implementing liquids that can be mixed, and the components automatically recognized as matching one recipe or another.

333  **Example Lakeside Living** WI
Similar to "Lemonade", but with bodies of liquid that can never be depleted, and some adjustments to the "fill" command so that it will automatically attempt to fill from a large liquid source if possible.

§10.3. Dispensers and Supplies of Small Objects

378  **Example Pizza Prince** WI
Providing a pizza buffet from which the player can take as many pieces as he wants.

124  **Example Extra Supplies** WI
A supply of red pens from which the player can take another pen only if he doesn't already have one somewhere in the game world.

§10.4. Glass and Other Damage-Prone Substances

97  **Example Ming Vase** WI
ATTACK or DROP break and remove fragile items from play.

123  **Example Spring Cleaning** WI
A character who sulks over objects that the player has broken (and which are now off-stage).

195  **Example Paddington** WI
A CUT [something] WITH [something] command which acts

differently on different types of objects.

314  **Example Terracottissima** WI
The flowerpots once again, but this time arranged so that after the first breakage all undamaged pots are said to be "unbroken", to distinguish them from the others.

404  **Example Kyoto** WI
Expanding the effects of the THROW something AT something command so that objects do make contact with one another.

§10.5. Volume, Height, Weight

232  **Example Swerve left? Swerve right? Or think about it and die?** WI
Building a marble chute track in which a dropped marble will automatically roll downhill.

268  **Example Depth** WI
Receptacles that calculate internal volume and the amount of room available, and cannot be overfilled.

258  **Example Dimensions** WI
This example draws together the previous snippets into a working implementation of the weighbridge.

270  **Example The Speed of Thought** WI
Describing scientifically-measured objects in units more familiar to the casual audience.

259  **Example Lead Cuts Paper** WI
To give every container a breaking strain, that is, a maximum weight of contents which it can bear - so that to put the lead pig into a paper bag invites disaster.

§10.6. Ropes

228  **Example Otranto** WI
A kind of rope which can be tied to objects and used to anchor the player or drag items from room to room.

261  **Example Snip** WI
A string which can be cut into arbitrary lengths, and then tied back together.

§10.7. Electricity and Magnetism

346  **Example Rules of Attraction** WI
A magnet which picks up nearby metal objects, and describes itself appropriately in room descriptions and inventory listings, but otherwise goes by its ordinary name.

408  **Example Electrified** WI
Adding a rule before the basic accessibility rule that will prevent the

player from touching electrified objects under the wrong circumstances.

139



Example Witnessed 1

A kind of battery which can be put into different devices, and which will lose power after extended use.

WI

§10.8. Fire

118



Example Thirst 2

A campfire added to the camp site, which can be lit using tinder.

WI

150



Example Bruneseau's Journey

A candle which reacts to lighting and blowing actions differently depending on whether it has already been lit once.

WI

418



Example The Cow Exonerated

Creating a class of matches that burn for a time and then go out, with elegant reporting when several matches go out at once.

WI

401



Example In Fire or in Flood

A BURN command; flammable objects which light other items in their vicinity and can burn for different periods of time; the possibility of having parts or contents of a flaming item which survive being burnt.

WI

§10.9. Heat

84



Example Grilling

A grill, from which the player is not allowed to take anything lest he burn himself.

WI

182



Example Entropy

All objects in the game have a heat, but if not kept insulated they will tend toward room temperature (and at a somewhat exaggerated rate).

WI

326



Example Masochism Deli

Multiple potatoes, with rules to make the player drop the hot potato first and pick it up last.

WI

331



Example Hot Glass Looks Like Cold Glass

Responding to references to a property that the player isn't yet allowed to mention (or when not to use "understand as a mistake").

WI

§10.10. Magic (Breaking the Laws of Physics)

215



Example Magneto's Revenge

Kitty Pryde of the X-Men is able to reach through solid objects, so we might implement her with special powers that the player does not have...

WI

225



Example Interrogation

A wand which, when waved, reveals the concealed items carried by people the player can see.

WI

227

★ Example Transmutations

A machine that turns objects into other, similar objects.

WI

398

★ Example Access All Areas

The Pointy Hat of Liminal Transgression allows its wearer to walk clean through closed doors.

WI

§10.11. Mathematics

240

★ Example Number Study

The parity and joint magnitude relations explored.

WI

434

★ Example The Fibonacci Sequence

The modest Leonardo Fibonacci of Pisa will be only too happy to construct his sequence on request, using an array.

WI

437

★ Example Sieve of Eratosthenes

The haughty Eratosthenes of Cyrene will nevertheless consent to sieve prime numbers on request.

WI

Chapter 11: Out Of World Actions and Effects*§11.1. Start-Up Features*

382

★ Example Bikini Atoll

Delaying the banner for later.

WI

440

★ Example Alien Invasion Part 23

Keeping a preference file that could be loaded by any game in a series.

WI

126

★★ Example Hatless

It's tempting to use "now..." to distribute items randomly at the start of play, but we need to be a little cautious about how we do that.

WI

§11.2. Saving and Undoing

210

★ Example Spellbreaker

P. David Lebling's classic "Spellbreaker" (1986) includes a room where the game cannot be saved: here is an Inform implementation.

WI

211

★★★ Example A point for never saving the game

In some of the late 1970s "cave crawl" adventure games, an elaborate scoring system might still leave the player perplexed as to why an apparently perfect play-through resulted in a score which was still one point short of the supposed maximum. Why only 349 out of 350? The answer varied, but sometimes the last point was earned by never saving the game - in other words by playing it right through with nothing to guard against mistakes (except perhaps UNDO for the last command), and in one long session.

WI

§11.3. Helping and Hinting

- 111  **Example Y ask Y?** WI
Noticing when the player seems to be at a loss, and recommending the use of hints.
- 284  **Example Food Network Interactive** WI
Using a menu system from an extension, but adding our own material to it for this game.
- 296  **Example Ish.** WI
A (very) simple HELP command, using tokens to accept and interpret the player's text whatever it might be.
- 329  **Example Query** WI
Catching all questions that begin with WHO, WHAT, WHERE, and similar question words, and responding with the instruction to use commands, instead.
- 285  **Example Trieste** WI
Table amendment to adjust HELP commands provided for the player.
- 400  **Example Solitude** WI
Novice mode that prefaces every prompt with a list of possible commands the player could try, and highlights every important word used, to alert players to interactive items in the scenery.
- 54  **Example Real Adventurers Need No Help** WI
Allowing the player to turn off all access to hints for the duration of a game, in order to avoid the temptation to rely on them overmuch.
- 230  **Example The Unexamined Life** WI
An adaptive hint system that tracks what the player needs to have seen or to possess in order to solve a given puzzle, and doles out suggestions accordingly. Handles changes in the game state with remarkable flexibility, and allows the player to decide how explicit a nudge he wants at any given moment.

§11.4. Scoring

- 219  **Example Bosch** WI
Creating a list of actions that will earn the player points, and using this both to change the score and to give FULL SCORE reports.
- 136  **Example Mutt's Adventure** WI
Awarding points for visiting a room for the first time.
- 137  **Example No Place Like Home** WI
Recording a whole table of scores for specific treasures.
- 166  **Example Panache** WI
Replacing the score with a plot summary that records the events of the plot, scene by scene.
-  **Example Goat-Cheese and Sage Chicken** WI

280 Implementing a FULL SCORE command which lists more information than the regular SCORE command, adding times and rankings, as an extension of the example given in this chapter. WI

442 **☆☆☆ Example Rubies**
A scoreboard that keeps track of the ten highest-scoring players from one playthrough to the next, adding the player's name if he has done well enough. WI

§11.6. Ending The Story

383 **★ Example Battle of Ridgefield**
Completely replacing the endgame text and stopping the game without giving the player a chance to restart or restore. WI

384 **★ Example Finality**
Not mentioning UNDO in the final set of options. WI

385 **★ Example Jamaica 1688**
Adding a feature to the final question after victory, so that the player can choose to reveal notes about items in the game. WI

386 **☆☆ Example Xerxes**
Offering the player a menu of things to read after winning the game. WI

441 **☆☆ Example Labyrinth of Ghosts**
Remembering the fates of all previous explorers of the labyrinth. WI

138 **☆☆☆ Example Big Sky Country**
Allowing the player to continue play after a fatal accident, but penalizing him by scattering his possessions around the game map. WI

Chapter 12: Typography, Layout, and Multimedia Effects

§12.1. Typography

69 **★ Example Garibaldi 2**
Adding coloured text to the example of door-status readouts. WI

453 **★ Example Chanel Version 1**
Making paired italic and boldface tags like those used by HTML for web pages. WI

450 **☆☆ Example Tilt 3**
Displaying the card suits from our deck of cards with red and black colored unicode symbols. WI

70 **☆☆☆ Example The Über-complète clavier**
This example provides a fairly stringent test of exotic lettering. WI

§12.2. The Status Line

- 114

★ Example Politics as Usual

Have the status line indicate the current region of the map.

WI
- 350

★ Example Ways Out

A status line that lists the available exits from the current location.

WI
- 387

★ Example Blankness

Emptying the status line during the first screen of the game.

WI
- 412

★ Example Capital City

To arrange that the location information normally given on the left-hand side of the status line appears in block capitals.

WI
- 452

★ Example Status line with centered text, the hard way

A status line which has only the name of the location, centered.

WI
- 351

★★ Example Guided Tour

A status line that lists the available exits from the current location, changing the names of these exits depending on whether the room has been visited or not.

WI
- 115

★★★ Example Centered

Replacing the two-part status line with one that centers only the room name at the top of the screen.

WI

§12.3. Footnotes

- 300

★★★ Example Ibid.

A system which allows the author to assign footnotes to descriptions, and permits the player to retrieve them again by number, using "the number understood". Footnotes will automatically number themselves, according to the order in which the player discovers them.

WI

§12.5. Glulx Multimedia Effects

- 445

★★★ Example Flathead News Network

Using external files, together with a simple Unix script running in the background, to provide live news headlines inside a story file.

WI

Chapter 13: Testing and Publishing

§13.1. Testing

- 2

★ Example Bic

Testing to make sure that all objects have been given descriptions.

WI
- 414

★ Example Alpha

Creating a beta-testing command that matches any line starting with punctuation.

WI

§13.2. Publishing

446

★ **Example Baedeker**

Creating a floorplan of the cathedral using the locations from previous examples.

WI

447

★ **Example Port Royal 5**

Port Royal scenario given instructions for an EPS map.

WI

448

★ **Example Bay Leaves and Honey Wine**

Creating a map of Greece using the locations from previous examples.

WI

General Index

-  Examples in Alphabetical Order
-  Examples in Thematic Order
-  Examples in Numerical Order
-  Part I. Writing with Inform
-  Part II. The Inform Recipe Book

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- #
- !, ending sentences [2.3](#), [5.2](#)
 - "[']" *say phrase* [5.2](#)
 - "
 - defining texts [2.3](#), [3.1](#), [5.1](#), [5.2](#)
 - not usable in unit notations [15.14](#)
 - "[bold type]" *say phrase* [12.1](#)
 - "[fixed letter spacing]" *say phrase* [12.1](#)
 - "[italic type]" *say phrase* [12.1](#)
 - "[roman type]" *say phrase* [12.1](#)
 - "[variable letter spacing]" *say phrase* [12.1](#)
 - ', meaning " [2.3](#), [5.2](#), [25.2](#)
 - (arithmetic value) * (arithmetic value)* *phrase* [15.5](#)
 - (arithmetic value) + (arithmetic value)* *phrase* [15.5](#)
 -
 - optional words in phrase definitions [11.3](#)
 - optional words in Understand grammar [17.12](#)
 - , as a blank table entry [16.7](#)
 - , separating cases [11.8](#)
 - (arithmetic value) - (arithmetic value)* *phrase* [15.5](#)
 - >, in phrase kinds [22.3](#)
 - .
 - before text substitutions [5.6](#)
 - ending sentences with line breaks [2.3](#), [5.2](#), [5.8](#)
 - (arithmetic value) / (arithmetic value)* *phrase* [15.5](#)
 - /
 - separating synonymous words in grammar [17.4](#), [17.12](#), [17.13](#)
 - separating synonymous words in phrase definitions [11.3](#)
 - separating synonymous words in topics [16.13](#)
 - : [6.4](#), [11.2](#), [11.7](#), [11.16](#), [19.3](#)
 - ,
 - comma replacing colon [7.2](#)
 - displaying serial comma [2.12](#), [5.5](#)
 - in list constants, requiring spaces [21.2](#)
 - separating phrase options [11.14](#)
 - ?, ending sentences [2.3](#), [5.2](#)
 - []
 - comments [2.3](#)
 - text substitutions [2.3](#), [5.1](#), [5.2](#)
 - (+ +), for including Inform 7 code in Inform 6 [27.19](#)
 - (- -), for including Inform 6 code in Inform 7 [27.15](#), [27.19](#)
 - ()
 - for calling values [8.15](#)
 - for defining mistakes [17.21](#)
 - for implicit multiplication [15.18](#)

- for phrase parameters [11.3](#)
 - in parser clarifications [18.30](#)
 - in response names [14.10](#)
- { }
- for constant lists [21.2](#)
 - for including Inform 7 expressions in Inform 6 [27.15](#)
- |, for paragraph break [2.3](#)
- The Über-complète clavier* [example 5.11 ex 70, 12.1 ex 70](#)
- A** *A&E* [example 7.13 ex 101, 3.2 ex 101](#)
- a / an / the [3.18, 5.3, 5.5](#)
- AARP-Gnosis* [example 18.5 ex 335, 9.6 ex 335](#)
- abandon the (activity) activity [phrase 18.7](#)
- abandon the (activity on values) activity with (value) [phrase 18.7](#)
- abbreviated room descriptions [use option 2.12](#)
- abbreviations
 - chains of equality tests [11.8](#)
 - object names [3.2](#)
 - property names, omitting of the [8.6](#)
 - rule actions with doing it [7.17](#)
- abide by (rule) [phrase 19.14](#)
- abide by (values based rule producing values) for (value) [phrase 19.14](#)
- able to see [relation verb see visibility](#)
- able to touch [relation verb see touchability](#)
- The Abolition of Love* [example 13.10 ex 231, 7.16 ex 231](#)
- About Inform's regular expression support* [example 20.6 ex 415, 1.4 ex 415](#)
- About the examples* [example 1.1 ex 1, 1.1 ex 1](#)
- absolute value of (real number) [phrase 15.6](#)
- ABSTRACT [testing command 24.4](#)
- accented letters [see characters \(letters\)](#)
- Access All Areas* [example 19.5 ex 398, 10.10 ex 398](#)
- access, deny / allow access (reaching inside / outside) [outcome phrase 12.18](#)
- accessibility text for figures and sounds [23.9](#)
- Actaeon* [example 17.7 ex 302, 7.13 ex 302](#)
- action name [12.20](#)
- action name part of (action) [phrase 12.20](#)
- action of (an action) [phrase 12.20](#)
- action processing [rulebook 19.15](#)
- actions [7.1, 12.1, 12.21](#)
 - applying to multiple objects [17.4, 6.15](#)
 - as conditions [9.12](#)
 - as values [12.20](#)
 - continue the action [phrase 7.3, 7.5](#)
 - defining new actions [12.7, 6.1](#)
 - defining new actions in Inform 6 [27.22](#)
 - exiting [7.14](#)
 - going, special cases [7.13, 7.14](#)
 - guidelines for action rules [12.21](#)
 - instructing other characters
 - persuasion rules [12.4](#)
 - unsuccessful attempts [12.5](#)
 - instructing other people [12.3, 7.14](#)
 - involving senses [7.7, 3.8](#)
 - involving speech [7.6](#)
 - irregular verbs in action names [12.8](#)
 - multiple action processing [17.20](#)
 - new kinds of action [7.15](#)
 - not the same as activities [18.1](#)
 - out of world [12.15, 12.21](#)
 - performing with try [7.4, 12.1, 12.6](#)
 - processing sequence [12.2, 19.15, 6.3](#)
 - pushing objects to other rooms [7.14](#)
 - reaching inside / outside rules [12.16, 12.18](#)

- reactions by other characters [7.3](#)
- redirecting actions [6.14](#)
- referring to current action with **action** [12.7](#)
- requirements for actions [12.16](#), [12.17](#), [12.18](#), [12.19](#), [6.2](#)
- rules applying only when characters are present [7.12](#)
- rules applying to rooms and regions [7.11](#), [7.13](#)
- rules applying under arbitrary conditions, using **when** [7.12](#)
- rules for actions found in rulebooks [12.2](#), [6.3](#)
- rules for all but some actions [7.9](#)
- rules for any actor (including the player) [12.14](#)
- rules for consecutive repetitions [7.17](#)
- rules for multiple actions [7.8](#)
- rules for new actions [12.9](#), [6.2](#)
- rules for other characters [12.4](#), [12.5](#), [12.11](#), [12.12](#), [12.13](#)
- rules for repetitions [7.16](#)
- spontaneous by other people [12.6](#)
- stop the action** *phrase* [7.3](#)
- stored actions [12.20](#)
- suppressing success messages with **try silently** [7.4](#), [12.1](#)
- understanding as mistakes [17.21](#), [5.3](#)
- understanding in player's command [17.1](#)
- using after rules to change output [7.5](#)
- using before rules to test or change outcomes first [7.3](#)
- using instead rules to change outcome [7.2](#)
- variables for actions [12.10](#)
- ACTIONS** *testing command* [7.1](#), [12.1](#), [12.5](#), [12.12](#), [24.3](#), [10.10](#), [13.1](#)
- Actions page of Index panel *user interface* [12.5](#), [12.21](#), [17.1](#), [17.2](#), [17.3](#), [24.3](#)
- active / inactive** *(use option) adjective* [27.18](#)
- activities [18.1](#), [18.2](#)
 - abandoning [18.7](#), [18.25](#)
 - beginning/ending by hand [18.7](#)
 - catalogue [18.8](#)
 - amusing a victorious player [18.39](#)
 - asking which do you mean [18.31](#)
 - choosing notable locale objects for something [18.27](#)
 - clarifying the parser's choice of something [18.30](#)
 - constructing the status line [18.23](#)
 - deciding the concealed possessions of something [3.24](#), [18.9](#)
 - deciding the scope of something [18.29](#), [5.5](#)
 - deciding whether all includes [18.36](#), [6.15](#)
 - grouping together something [18.14](#)
 - implicitly taking something [12.17](#), [18.34](#), [6.8](#)
 - issuing the response text of something [18.15](#)
 - listing contents of something [18.13](#)
 - listing nondescript items of something [18.25](#), [3.1](#)
 - printing a locale paragraph about [18.28](#), [3.1](#)
 - printing a number of something [18.12](#)
 - printing a parser error [18.35](#)
 - printing a refusal to act in the dark [18.18](#)
 - printing inventory details of something [18.17](#)
 - printing room description details of something [18.16](#), [3.1](#)
 - printing the announcement of darkness [18.19](#)
 - printing the announcement of light [18.20](#)
 - printing the banner text [18.37](#)
 - printing the description of a dark room [18.22](#)
 - printing the locale description of something [18.26](#)
 - printing the name of a dark room [18.21](#)
 - printing the name of something [18.10](#)
 - printing the player's obituary [18.38](#)
 - printing the plural name of something [18.11](#)
 - reading a command [18.33](#)
 - starting the virtual machine [18.40](#)

- supplying a missing noun/second noun [18.32](#)
 - writing a paragraph about [18.24, 3.1](#)
- checking for ongoing activities in rules [18.4](#)
- defining [18.5](#)
- empty / non-empty *adjective* [18.5](#)
- golden rules for constructing [18.2, 18.7](#)
- not the same as actions [18.1](#)
- rules for activities [18.3](#)
- variables for activities [18.6](#)
- Actor's Studio example* [12.20 ex 221, 9.12 ex 221](#)
- actor part of (*action*) *phrase* [12.20](#)
- actor — *person action variable* [12.3, 12.14, 14.4](#)
- actors *see* characters (people)
- adaptive displayed text [14.2, 2.1](#)
- contractions [14.7, 14.8](#)
 - pronouns [14.2, 14.4, 14.5, 14.6, 14.8](#)
 - verbs [14.2, 14.3](#)
- add (*list of values*) at entry (*number*) in (*list of values*) *phrase* [21.5](#)
- add (*list of values*) to (*list of values*) *phrase* [21.5](#)
- add (*value*) at entry (*number*) in (*list of values*) *phrase* [21.5](#)
- add (*value*) to (*list of values*) *phrase* [21.1, 21.5](#)
- adjacency *relation* [6.14, 13.8](#)
- adjacent (*room*) *adjective* [6.14](#)
- adjectives [6.3](#)
 - defining [6.4](#)
 - with Inform 6 [27.24](#)
 - with to decide whether [11.16](#)
 - for numeric quantities [6.6, 6.7, 6.8](#)
 - for numeric quantities with units [15.12](#)
 - for values [6.5](#)
 - in descriptions [6.2](#)
- adverbs [6.18](#)
- (*time*) after (*time*) *phrase* [9.9](#)
- (*name of kind*) after (*enumerated value*) *phrase* [11.18](#)
- after (*action*) *rulebook* [7.5](#)
 - guidelines for use [12.21](#)
 - in action processing sequence [12.2, 6.3](#)
 - not used for out of world actions [12.15](#)
- after (*activity*) *rulebook* [18.3](#)
- after, ordering for rules [19.4](#)
- Aftershock example* [18.5 ex 336, 9.9 ex 336](#)
- Age of Steam example* [10.3 ex 156, 4.4 ex 156](#)
- Ahem example* [11.3 ex 169, 2.1 ex 169](#)
- Air Conditioning is Standard example* [18.24 ex 354, 4.3 ex 354](#)
- Albaugh, Lea [25.9](#)
- Alias example* [15.2 ex 253, 9.10 ex 253](#)
- Alien Invasion Part 23 example* [23.13 ex 440, 11.1 ex 440](#)
- all except, in descriptions [6.16](#)
- all, in descriptions [6.15](#)
- ALL, including things in [18.36, 6.15](#)
- All Roads Lead to Mars example* [6.14 ex 78, 3.2 ex 78](#)
- allow access (*reaching inside / outside*) *outcome phrase* [12.18](#)
- almost all, in descriptions [6.15](#)
- Alpaca Farm example* [17.2 ex 289, 6.17 ex 289](#)
- Alpha example* [20.6 ex 414, 13.1 ex 414](#)
- alter the multiple object list to (*list of objects*) *phrase* [17.20](#)
- always / never, in assertions [4.3](#)
- always, in defining constant values [4.13](#)
- amended, for tables [16.19](#)
- American dialect *use option* [2.12, 2.13, 5.4](#)
- AMUSING [9.4, 18.39, 11.6](#)
- amusing a victorious player *activity* [18.39](#)

an / a / the [3.18, 5.3, 5.5](#)
Anchorite example [17.2 ex 290, 6.10 ex 290](#)
 animals *kind* [3.17, 4.1, 8.3](#)
 rideable [3.16, 8.3](#)
 announcement of darkness -- printing the announcement of darkness *activity* [18.19](#)
 announcement of light -- printing the announcement of light *activity* [18.20](#)
Annoyotron Jr example [9.15 ex 153, 7.2 ex 153](#)
 anonymously abide by (rule) *phrase* [19.14](#)
 anonymously abide by (values based rule producing values) for (value) *phrase* [19.14](#)
 another, in defining relations [13.6](#)
 answering (actor) *action* [7.6, 17.10](#)
Ant-Sensitive Sunglasses example [18.1 ex 334, 3.1 ex 334](#)
Anteaters example [12.20 ex 222, 6.14 ex 222](#)
 [any] *grammar token* [17.7, 5.5](#)
 [anybody] *grammar token* [17.7](#)
 anybody, in descriptions [6.2](#)
 [anyone] *grammar token* [17.7](#)
 anyone, in descriptions [6.2](#)
 [anything] *grammar token* [17.7](#)
 anything, in descriptions [6.2](#)
 [anywhere] *grammar token* [17.7](#)
 anywhere, in descriptions [6.2](#)
 apostrophe *see* [punctuation: apostrophe, meaning quotation mark](#)
 appearance *see* [descriptions \(displayed\)](#)
 append (text) to (external file) *phrase* [23.14](#)
Apples example [18.31 ex 368, 6.17 ex 368](#)
 (phrase nothing -> value) applied *phrase* [22.3](#)
 (phrase (value, value, value) -> value) applied to (value) and (value) and (value) *phrase* [22.3](#)
 (phrase (value, value) -> value) applied to (value) and (value) *phrase* [22.3](#)
 (phrase value -> value) applied to (value) *phrase* [22.3](#)
 (phrase value -> value) applied to (list of values) *phrase* [22.5](#)
 apply (phrase nothing -> nothing) *phrase* [22.3](#)
 apply (phrase value -> nothing) to (value) *phrase* [22.3](#)
 apply (phrase (value, value) -> nothing) to (value) and (value) *phrase* [22.3](#)
 apply (phrase (value, value, value) -> nothing) to (value) and (value) and (value) *phrase* [22.3](#)
 applying to, in defining actions [12.7, 17.1](#)
 Approximate Metric Units *extension* [15.13](#)
 arccosine of (real number) *phrase* [15.7](#)
 Archive *web site see* [IF Archive](#)
 arcsine of (real number) *phrase* [15.7](#)
 arctangent of (real number) *phrase* [15.7](#)
 "[are]" *say phrase* [14.2](#)
 "[aren't]" *say phrase* [14.8](#)
 arguments *see* [parameters](#)
 arithmetic *see* [calculation](#)
 arithmetic value [22.1, 22.6](#)
The Art of Noise example [7.7 ex 95, 3.8 ex 95](#)
 articles [3.18](#)
 displaying and capitalizing [7.10](#)
 as something new [17.3](#)
 asking (actor) about *action* [7.6](#)
 asking (actor) to try *action* [12.3](#)
 asking the player yes/no [11.5](#)
 asking which do you mean *activity* [18.31](#)
Aspect example [17.15 ex 311, 9.9 ex 311](#)
 assemblies [4.15, 5.1, 9.8](#)
 assertions [2.1, 3.1](#)
 plural [4.4](#)
 at least *relation verb* [15.5](#)
 at least, in descriptions [6.16](#)
 at most *relation verb* [15.5](#)
 at most, in descriptions [6.16](#)

at (time)... assertion 9.11, 19.7, 4.3
 (rule) at (time) phrase 9.11
 audio *see* sounds
 author *bibliographic data* 2.5, 18.37, 25.2
 of an extension 27.4
 authorial modesty *use option* 27.6
B Babel, Treaty of Babel 25.5
 back side of (object) phrase 3.12
 backdrops *kind* 3.9, 3.4, 3.9
 changing the location of backdrops 8.8
Backus-Naur form for rules example 19.7 ex 399, 1.4 ex 399
Bad Hair Day example 7.2 ex 85, 5.3 ex 85
Baedeker example 25.23 ex 446, 13.2 ex 446
Ballpark example 5.4 ex 60, 2.1 ex 60
 banner text 18.37, 25.4, 11.1, 11.5
Baritone, Bass example 17.11 ex 309, 5.2 ex 309
Barter Barter example 12.12 ex 207, 7.4 ex 207
 Basic Screen Effects *extension* 27.7, 4.5
 basis of rulebook 19.9
Battle of Ridgefield example 18.38 ex 383, 11.6 ex 383
Bay Leaves and Honey Wine example 25.23 ex 448, 13.2 ex 448
Beachfront example 7.12 ex 98, 6.6 ex 98
Bee Chambers example 8.11 ex 125, 3.2 ex 125
Beekeeper's Apprentice example 5.8 ex 68, 6.5 ex 68
 (time) before (time) phrase 9.9
 (name of kind) before (enumerated value) phrase 11.18
 before (action) rulebook 7.3
 guidelines for use 12.21
 in action processing sequence 12.2, 6.3
 not used for out of world actions 12.15
 before (activity) rulebook 18.3
 before, ordering for rules 19.4
 begin, in phrases 11.7, 11.9
 begin the (activity) activity phrase 18.7
 begin the (activity on values) activity with (value) phrase 18.7
 beginning of scene *see* scenes: rules run at beginning
 beginning of story *see* initial state of the world
Being Peter example 19.12 ex 405, 7.10 ex 405
Being Prepared example 4.15 ex 56, 9.3 ex 56
Belfry example 3.18 ex 31, 7.1 ex 31
Beneath the Surface example 13.10 ex 233, 6.6 ex 233
 best route from (object) to (object) phrase 6.14
 best route from (object) to (object) through (description of objects) phrase 6.14
The Best Till Last example 17.20 ex 327, 6.15 ex 327
Beverage Service example 8.10 ex 122, 10.2 ex 122
 bibliographic data 2.5, 18.37, 25.2, 25.4, 25.15, 25.17, 26.3, 11.5
Bibliophilia example 17.17 ex 324, 9.6 ex 324
Bic example 3.1 ex 2, 13.1 ex 2
The Big Sainsbury's example 18.34 ex 377, 4.1 ex 377
Big Sky Country example 9.4 ex 138, 11.6 ex 138
Bikini Atoll example 18.37 ex 382, 11.1 ex 382
 binary, when declaring data files 23.12
Blackout example 20.8 ex 419, 2.1 ex 419
 blank entries in tables *see* tables: blank entries
 Blank, Marc 2.13
 blank out (a table entry) phrase 16.10
 blank out the whole of (table) phrase 16.10
 blank out the whole row phrase 16.10
 blank out the whole (table column) column in (table) phrase 16.10
Blankness example 18.40 ex 387, 12.2 ex 387
 blessed transcripts *see* transcripts: in the Skein
Blink example 27.30 ex 454, 2.1 ex 454

blorb file [23.6](#), [23.10](#), [25.4](#), [25.7](#), [26.3](#)
 blurb *bibliographic data* *see* [story description](#)
 body parts [4.15](#), [5.1](#)
Bogart example [13.10 ex 234](#), [9.3 ex 234](#)
 book *of source text* [2.5](#)
 books (implementing) [9.6](#)
 consulting [7.6](#), [9.6](#)
 with pages [9.6](#)
Bosch example [12.20 ex 219](#), [11.4 ex 219](#)
Boston Cream example [12.6 ex 190](#), [7.15 ex 190](#)
Bowler Hats and Baby Geese example [10.4 ex 159](#), [4.2 ex 159](#)
 boxed quotations [5.12](#)
 brace notation -- "[\[\(list of values\) in brace notation\]](#)" *say phrase* [21.3](#)
 braces *see* [punctuation: curly braces](#)
 brackets *see* [punctuation: brackets](#)
 break *phrase* [11.12](#)
 breakable things (implementing) [10.4](#)
 breaks *see* [paragraph breaks](#)
Bribery example [19.3 ex 392](#), [7.4 ex 392](#)
 BRIEF room descriptions *use option* [2.12](#), [6.4](#), [11.5](#)
 Brooks, Frederick P. [1.1](#)
Brown example [3.23 ex 36](#), [9.7 ex 36](#)
Bruneseau's Journey example [9.13 ex 150](#), [10.8 ex 150](#)
 bugs in Inform [2.15](#)
Bumping into Walls example [7.13 ex 102](#), [6.9 ex 102](#)
 burning *action* [10.8](#)
 buying *action* [9.4](#)

C

Cactus Will Outlive Us All example [12.20 ex 220](#), [6.14 ex 220](#)
 calculation [10.11](#)
 arithmetic [15.5](#)
 arithmetic with units [15.19](#), [15.20](#)
 average of a numeric property [15.17](#)
 conversion during calculations [15.3](#)
 decrementing [8.12](#), [11.18](#)
 exponential functions [15.6](#)
 given by equations [15.18](#)
 incrementing [8.12](#), [11.18](#)
 logarithms [15.6](#)
 rounding to the nearest multiple [9.8](#), [15.5](#)
 rounding to the nearest whole number [15.3](#), [15.6](#)
 total of a numeric property [15.17](#)
 trigonometric functions [15.7](#)
 with time [9.9](#), [9.10](#)

called

 in creating assemblies [4.16](#)
 in creating rooms [3.2](#)
 in creating things [3.5](#)
 in defining action variables [12.10](#)
 in defining activity variables [18.6](#)
 in defining adjectives [6.4](#)
 in defining relations [13.5](#), [13.6](#), [13.12](#)
 in defining rulebook variables [19.10](#)
 in describing things [8.15](#), [11.11](#)
Camp Bethel example [5.7 ex 67](#), [7.2 ex 67](#)
 "[can't]" *say phrase* [14.7](#)
 "[can]" *say phrase* [14.2](#), [14.7](#)
 "[can see]" *say phrase* [14.2](#)
 "[can touch]" *say phrase* [14.2](#)
Candy example [8.19 ex 133](#), [9.1 ex 133](#)
Capital City example [20.4 ex 412](#), [12.2 ex 412](#)
Carnivale example [12.16 ex 212](#), [3.4 ex 212](#)
 carried (*thing*) *adjective* [3.20](#), [8.13](#)

Carroll, Lewis [5.12](#)

"[carry]" *say phrase* [14.2](#)

carry out (*action*) *rulebook* [12.2](#), [12.9](#), [6.3](#)
 guidelines for use [12.21](#)

carry out the (*activity*) *activity phrase* [18.5](#)

carry out the (*activity on values*) *activity with (value) phrase* [18.5](#)

carrying *relation* [3.20](#), [8.7](#), [8.13](#), [13.4](#)
 requirement in defining actions [12.7](#), [12.17](#)

carrying capacity of (*container/supporter/person*) *property* [3.19](#), [10.5](#)
 avoid overflowing player's capacity using a hold-all [3.21](#)

case sensitivity
 checking texts [20.4](#)
 in matching texts [20.5](#)
 in real number constants [15.2](#)
 in regular expressions [20.6](#)
 in text substitutions [5.13](#)
 in text substitutions with lists [5.5](#)
 in text substitutions with objects [5.3](#), [7.10](#)
 in text substitutions with pronouns [14.4](#), [14.5](#), [14.6](#)

Casino Banale example [18.27](#) *ex 360*, [5.5](#) *ex 360*

Cater, John [23.10](#)

Cave-troll example [20.8](#) *ex 424*, [6.17](#) *ex 424*

ceiling of (*real number*) *phrase* [15.6](#)

Celadon example [13.4](#) *ex 224*, [6.8](#) *ex 224*

Centered example [8.3](#) *ex 115*, [12.2](#) *ex 115*

chance *see* [randomness](#)

Chanel Version 1 example [27.28](#) *ex 453*, [12.1](#) *ex 453*

change *see* [now \(a condition\)](#)

change (*direction*) *exit of (room) to nothing/nowhere phrase* [8.5](#)

change (*direction*) *exit of (room) to (room) phrase* [8.5](#)

change (*list of values*) *to have (number) entries/entry phrase* [21.10](#)

Change of Basis example [4.7](#) *ex 47*, [6.11](#) *ex 47*

change the text of the player's command to (*text*) *phrase* [18.33](#)

Channel 1 example [17.15](#) *ex 313*, [9.9](#) *ex 313*

Channel 2 example [17.15](#) *ex 316*, [9.9](#) *ex 316*

chapter *of source text* [2.5](#)

character number (*number*) *in (text) phrase* [20.3](#)

characters (letters) [5.11](#)
 capital letters as kind variables [22.7](#)
 counting [20.3](#)
 in text values [20.1](#)
 replacing in text [20.8](#)
 special / accented [5.10](#)
 special meanings in regular expressions [20.6](#), [20.9](#)
 Unicode (arbitrary symbols) [5.11](#), [12.1](#)
 defining new names for [27.25](#)

characters (people)
 accepting gifts [7.4](#), [9.4](#)
 acting spontaneously [12.6](#), [7.2](#)
 characterization [7.10](#)
 combat [7.5](#)
 dialogue *see* [dialogue](#)
 giving instructions to other characters [12.3](#), [7.14](#)
 goal-seeking [7.15](#)
 groups of people [7.16](#)
 Inanimate Listeners *extension* [7.7](#)
 memory *see* [knowledge \(in story world\)](#)
 moving around [7.13](#)
 names and titles [7.1](#)
 persuading [12.4](#)
 reacting to each other [7.16](#)
 reacting to the player's actions [7.3](#)

- reporting unsuccessful actions [12.5](#)
- rules applying only when characters are present [7.12](#)
- scripted conversations [7.12](#)
- check** (*action*) *rulebook* [12.2](#), [12.9](#), [6.3](#)
 - for other characters [12.12](#)
 - guidelines for use [12.21](#)
- Cheese-makers* *example* [10.9](#) ex [168](#), [7.8](#) ex [168](#)
- Chimento, Philip [1.2](#)
- choose a/the/-- blank row in/from (*table name*) *phrase* [16.10](#)
- choose a/the/-- random row in/from (*table name*) *phrase* [16.5](#)
- choose a/the/-- row (*number*) in/from (*table name*) *phrase* [16.5](#)
- choose a/the/-- row with (*table column*) of (*value*) in/from (*table name*) *phrase* [16.5](#)
- choosing notable locale objects for something *activity* [18.27](#)
- Chronic Hinting Syndrome* *example* [18.10](#) ex [341](#), [7.11](#) ex [341](#)
- Cinco* *example* [17.16](#) ex [319](#), [9.2](#) ex [319](#)
- Circle of Misery* *example* [21.11](#) ex [432](#), [8.4](#) ex [432](#)
- Claims Adjustment* *example* [17.16](#) ex [322](#), [9.12](#) ex [322](#)
- clarifying the parser's choice of something *activity* [18.30](#)
- Cloak of Darkness* *example* [17.2](#) ex [291](#), [3.7](#) ex [291](#)
- clocks (implementing) [9.11](#)
- closed / open (*container/door*) *property* [3.6](#), [3.13](#), [3.15](#), [9.2](#)
- clothing *see* [wearing](#)
- Cloves* *example* [18.33](#) ex [373](#), [6.18](#) ex [373](#)
- Clueless* *example* [9.1](#) ex [135](#), [7.1](#) ex [135](#)
- colon *see* [punctuation: colon](#)
- columns, in tables *see* [tables](#)
- combat [7.5](#)
- comma *see* [punctuation: comma](#)
- command prompt** — *text* *global variable* [8.2](#), [9.7](#)
- commands (of the player) *see* [understanding](#)
- comments
 - in source text [2.3](#)
 - in transcripts from beta testers [17.21](#), [24.7](#), [13.1](#)
- comparisons
 - counting [6.16](#)
 - equality [11.6](#); *see also* [equality](#)
 - of enumerated values [11.18](#)
 - of numbers [15.5](#)
 - of numbers using adjectives [6.7](#)
 - of numbers with units using adjectives [15.12](#)
 - of text [20.5](#)
 - with regular expressions [20.6](#)
- compass directions *see* [directions](#)
- Complimentary Peanuts* *example* [18.33](#) ex [376](#), [7.8](#) ex [376](#)
- components [3.23](#)
 - breaking apart [10.4](#)
 - for kinds [4.15](#)
 - incorporation relation *see* [incorporation](#)
 - of bodies [5.1](#)
 - of machines [9.8](#)
- concealed possessions [3.24](#), [18.9](#)
- concealed** (*thing*) *adjective* [3.24](#), [18.9](#)
- concealment *relation* [3.24](#), [13.4](#), [18.9](#)
 - in a room *see* [hiding things from room descriptions by making them scenery](#)
 - under other things *see* [hiding things under other things](#)
- condition**, naming properties [4.10](#)
- conditions
 - changing conditions of rules [19.5](#)
 - defined by phrases [11.5](#)
 - defining with **to decide whether** [11.16](#)
 - equality [11.6](#)
 - expressed as relations [13.12](#)

- for activity rules [18.4](#)
- for rules [7.12](#), [9.5](#), [19.5](#), [19.7](#)
- for scenes [10.4](#)
- for **understand** [17.17](#), [2.2](#)
- historical [9.13](#)
 - number of times [9.14](#)
 - number of turns [9.15](#)
 - warning concerning variables and other varying qualities [9.13](#), [9.15](#)
- testing [11.6](#)
- with **otherwise/else** [11.8](#)
- connections between rooms [3.2](#), [3.2](#)
 - changing [8.5](#)
 - doors [3.12](#), [3.5](#)
 - inconsistent directions [3.3](#)
 - one-way [3.3](#)
 - rules applying to map connections [7.13](#)
 - to nowhere [3.3](#)
- consents -- if player consents** *phrase* [11.5](#)
- constant values [4.13](#)
- constructing the status line** *activity* [18.23](#)
- consulting (thing) about** *action* [7.6](#)
- container in question — object** *global variable* [12.18](#)
- containers *kind* [3.5](#), [8.4](#), [9.2](#)
 - carrying capacity [3.19](#), [10.5](#)
 - containment relation *see* [containment](#)
 - enterable [3.6](#)
 - liquids in containers [10.2](#)
 - locked [3.13](#)
 - open / closed [3.6](#), [3.15](#)
 - pockets in clothing [9.3](#)
 - transparent / opaque [3.6](#), [3.15](#)
 - vehicle as kind of container [3.16](#)
- containment *relation* [6.11](#), [8.13](#), [13.4](#)
 - hierarchy, low-level interface [8.17](#)
 - indirect [3.25](#), [6.11](#)
 - placing the contents of something in scope [18.29](#)
- Contents page of Index panel *user interface* [2.6](#), [23.5](#), [27.10](#)
- continue the action** *phrase* [7.3](#), [7.5](#), [19.11](#)
- continue the activity** *phrase* [18.5](#)
- continued**, for tables [16.18](#)
- contractions, adaptive contractions in displayed text [14.7](#), [14.8](#)
- Control Center* *example* [5.5](#) *ex 61*, [9.8](#) *ex 61*
- conversation *see* [dialogue](#)
- Copper River* *example* [18.28](#) *ex 362*, [3.1](#) *ex 362*
- Cornelson, David [1.2](#)
- corresponding to**, in tables [16.3](#)
- cosine of (real number)** *phrase* [15.7](#)
- Costa Rican Ornithology* *example* [7.6](#) *ex 94*, [9.6](#) *ex 94*
- "[could]" *say phrase* [14.7](#)
- "[couldn't]" *say phrase* [14.7](#)
- counted loops [11.10](#)
- counting [6.16](#)
 - actions with **for the Nth time** [7.16](#), [11.4](#)
 - actions with **for the Nth turn** [7.17](#)
 - characters / words / paragraphs / lines [20.3](#)
 - conditions with **for the Nth time** [9.14](#)
 - conditions with **for the Nth turn** [9.15](#)
 - moves along a route [6.14](#)
 - steps in an indirect relation [13.11](#)
 - things [8.16](#)
- cover art [25.8](#)
- The Cow Exonerated* *example* [20.7](#) *ex 418*, [10.8](#) *ex 418*

The Crane's Leg 1 *example* [4.9 ex 52](#), [5.6 ex 52](#)
The Crane's Leg 2 *example* [19.3 ex 390](#), [5.6 ex 390](#)
 creating *see* [defining](#)
Croft *example* [12.10 ex 202](#), [6.8 ex 202](#)
 Crowther, Will [3.1](#)
Crusoe *example* [18.5 ex 337](#), [6.5 ex 337](#)
 cube root of (*arithmetic value*) *phrase* [15.5](#)
Curare *example* [22.2 ex 439](#), [2.1 ex 439](#)
 curly braces *see* [punctuation: curly braces](#)
 current action *phrase* [12.20](#)
 cut (*snippet*) *phrase* [18.33](#)

D

Daly, Liza [1.2](#)
 damaging things (implementing) [10.4](#)
 dark / lighted (*room*) *property* [3.15](#), [3.7](#)
The Dark Ages Revisited *example* [12.9 ex 194](#), [3.7 ex 194](#)
 darkness [3.15](#), [6.13](#), [3.7](#)
 printing a refusal to act in the dark *activity* [18.18](#)
 printing the announcement of darkness *activity* [18.19](#)
 printing the announcement of light *activity* [18.20](#)
 printing the description of a dark room *activity* [18.22](#)
 printing the name of a dark room *activity* [18.21](#)
 rules to change perception of [12.19](#)
 testing whether in darkness [11.5](#)
 data files *see* [files \(data files\)](#)
A Day For Fresh Sushi *example* [7.16 ex 112](#), [7.3 ex 112](#)
Day One *example* [10.4 ex 160](#), [4.2 ex 160](#)
Dearth and the Maiden *example* [7.15 ex 109](#), [5.3 ex 109](#)
 death [9.4](#), [18.38](#), [18.39](#), [11.6](#)
 debug version (of the story) *see* [test version \(of the story\)](#); [test version \(of the story\)](#)
 debugging commands *see* [testing commands](#)
 decide no *phrase* [11.16](#)
 decide on (*value*) *phrase* [11.17](#)
 decide yes *phrase* [11.16](#)
 deciding the concealed possessions of something *activity* [3.24](#), [18.9](#)
 deciding the scope of something *activity* [18.29](#), [5.5](#)
 deciding whether all includes *activity* [18.36](#), [6.15](#)
 decimals *see* [real numbers](#)
 say "[*real number* in decimal notation]" *phrase* [15.4](#)
 say "[*real number* to (*number*) decimal places]" *phrase* [15.4](#)
 decrease (*a stored value*) by (*value*) *phrase* [8.12](#)
 decrement (*a stored value*) *phrase* [8.12](#)
 default outcome of a rulebook [19.11](#), [19.12](#)
 default properties of kinds [4.3](#); *see also* [kinds: default values of kinds](#)
 default value of (*name of kind*) *phrase* [4.11](#)
 defining
 action variables [12.10](#)
 actions [12.7](#), [6.1](#)
 activities [18.5](#)
 activity variables [18.6](#)
 adaptive displayable verbs [14.3](#)
 adjectives [6.4](#)
 for numeric quantities [6.6](#), [6.7](#), [6.8](#)
 for numeric quantities with units [15.12](#)
 for values [6.5](#)
 with **to decide whether** [11.16](#)
 constant values [4.13](#)
 data files [23.12](#)
 directions [3.26](#)
 either/or properties [4.7](#)
 equations for values with units [15.18](#)
 figures [23.5](#)
 global variables [4.12](#)

- grammar tokens [17.13](#)
 - producing values [17.14](#)
- Inform 6 entities *see* [Inform 6 inclusions](#)
- input grammar [17.1](#)
- kinds [4.1](#)
 - multiply using tables [16.16](#)
- kinds of thing [4.1](#), [4.2](#), [4.4](#)
- kinds of value [4.5](#), [4.9](#), [4.10](#)
 - defining with a table [16.17](#)
- lists [21.2](#)
 - by description [21.6](#), [21.7](#)
- phrases [11.2](#)
- plurals [4.4](#), [2.2](#)
- prepositions for relations [13.10](#)
- previewing [23.5](#)
- relations [13.5](#), [13.13](#)
- relations expressing conditions [13.12](#)
- rooms [3.1](#)
- rulebook variables [19.10](#)
- rulebooks [19.8](#)
- rules [2.2](#), [19.3](#), [19.7](#)
- sentence verbs for properties [15.12](#)
- sentence verbs for relations [13.9](#)
- sounds [23.8](#)
- tables [16.1](#)
- temporary variables [11.15](#)
- testing commands [2.9](#), [13.1](#)
- text substitutions [5.13](#)
- things [3.1](#)
 - multiply using tables [16.16](#)
 - multiply with counts [4.14](#)
 - using plurals [4.4](#)
 - using relations [4.15](#), [4.16](#)
- units of measure [15.8](#), [15.9](#), [15.10](#)
- units of measure with parts [15.14](#), [15.15](#)
- value properties [4.8](#), [4.9](#), [4.10](#)

definite articles

- when creating things [3.18](#)
- when displaying lists [5.5](#), [21.3](#)
- when displaying things [5.3](#)

definition *assertion* [6.4](#), [19.7](#)

- for numeric quantities [6.6](#), [6.7](#), [6.8](#)
- for values [6.5](#)
- with phrase logic [11.16](#)

(real number) **degrees** *phrase* [15.7](#)

Delayed Gratification *example* [19.2](#) *ex 389*, [6.11](#) *ex 389*

deleting things from the world [8.10](#)

Delicious, Delicious Rocks *example* [12.9](#) *ex 196*, [9.1](#) *ex 196*

Democratic Process *example* [7.3](#) *ex 86*, [6.8](#) *ex 86*

deny access *(reaching inside / outside)* *outcome phrase* [12.18](#)

Depth *example* [15.20](#) *ex 268*, [10.5](#) *ex 268*

described / undescribed *(thing)* *property* [3.24](#)

describing, in understand as [17.15](#)

description of a dark room -- printing the description of a dark room *activity* [18.22](#)

description of *(room)* *property* [3.10](#), [3.11](#)

description of *(scene)* *property* [10.2](#)

description of *(thing)* *property* [3.11](#)

descriptions (displayed)

- activities
 - choosing notable locale objects for something [18.27](#)
 - grouping together something [18.14](#)
 - listing contents of something [18.13](#)

- listing nondescript items of something [18.25, 3.1](#)
- printing a locale paragraph about something [18.28, 3.1](#)
- printing a number of something [18.12](#)
- printing inventory details of something [18.17](#)
- printing room description details of something [18.16, 3.1](#)
- printing the description of a dark room [18.22](#)
- printing the locale description of something [18.26](#)
- printing the name of something [18.10](#)
- printing the plural name of something [18.11](#)
- writing a paragraph about [18.24, 3.1](#)
- default descriptions using new kinds [4.2](#)
- full-length room descriptions / abbreviated room descriptions [2.12](#)
- hiding things from room descriptions [3.8](#)
- initial appearance of thing [3.11, 8.5, 3.1](#)
- miscellaneous things in room [18.25, 3.1](#)
- notable things in room [18.24, 3.1](#)
- room contents [18.26, 18.27, 18.28, 3.1, 6.4](#)
- room descriptions with substituted text [5.1](#)
- scene descriptions [10.2](#)
- varying with properties [3.14](#)
- descriptions (references to things) [6.1](#)
 - adjectives for magnitude [6.6](#)
 - adjectives for values [6.5](#)
 - as values [22.2](#)
 - comparative adjectives [6.7](#)
 - complex phrases [6.9](#)
 - counting [6.16](#)
 - defining adjectives for descriptions [6.4](#)
 - determiners (all / none) [6.15](#)
 - existence [6.10](#)
 - listed in table [16.12](#)
 - non-existence [6.10, 6.12](#)
 - relative [6.9](#)
 - superlative adjectives [6.8](#)
 - syntax [6.2](#)
- determiners (all / none) [6.15](#)
- devices *kind* [3.14, 9.8, 10.7](#)
 - displaying the on/off state [6.5](#)
- dialogue [7.6](#)
 - ASK / TELL** [7.6, 7.7, 7.8](#)
 - keyword-based dialogue [7.8](#)
 - menu-based dialogue [7.8](#)
 - scripted conversations [7.12](#)
 - simple terms [7.7](#)
 - structure of conversation [7.9](#)
 - with inanimate objects [7.7](#)
- DICT_WORD_SIZE** *use option* [2.2](#)
- Dimensions example* [15.12 ex 258, 10.5 ex 258](#)
- Dinner is Served example* [12.18 ex 217, 3.6 ex 217](#)
- direction of (**door**) **from** (**room**) *phrase* [3.12](#)
- directions *kind* [3.2, 3.5, 3.26, 4.1, 3.2, 6.9](#)
 - defining [3.26](#)
 - finding adjacent rooms [6.14](#)
 - shipboard directions, **FORE/AFT/PORT/STARBOARD** [8.2](#)
- disambiguation
 - of player commands [17.19, 6.17](#)
 - displaying parser's chosen thing [18.30](#)
 - printing disambiguation request [18.31](#)
 - of source text [2.6](#)
 - precedence of phrase definitions [22.9, 22.10](#)
 - precedence of rules [19.2, 19.4, 19.6, 19.16](#)
- Disenchantment Bay 1 example* [3.7 ex 14, 1.3 ex 14](#)

Disenchantment Bay 2 example [3.8 ex 15](#), [1.3 ex 15](#)
Disenchantment Bay 3 example [3.9 ex 17](#), [1.3 ex 17](#)
Disenchantment Bay 4 example [3.11 ex 18](#), [1.3 ex 18](#)
Disenchantment Bay 5 example [3.12 ex 20](#), [1.3 ex 20](#)
Disenchantment Bay 6 example [3.13 ex 23](#), [1.3 ex 23](#)
Disenchantment Bay 7 example [3.14 ex 25](#), [1.3 ex 25](#)
Disenchantment Bay 8 example [3.16 ex 28](#), [1.3 ex 28](#)
Disenchantment Bay 9 example [3.17 ex 30](#), [1.3 ex 30](#)
Disenchantment Bay 10 example [3.20 ex 33](#), [1.3 ex 33](#)
Disenchantment Bay 11 example [3.21 ex 34](#), [1.3 ex 34](#)
Disenchantment Bay 12 example [3.23 ex 37](#), [1.3 ex 37](#)
display (*figure name*) *phrase* [23.6](#)
display the boxed quotation (*text*) *phrase* [5.12](#)
 displaying descriptions *see* [descriptions \(displayed\)](#)
 distance
 measuring with **number of moves** [6.14](#)
 measuring with **number of steps** [13.11](#)
 (*arithmetic value*) **divided by** (*arithmetic value*) *phrase* [15.5](#)
do nothing *phrase* [11.1](#)
Do Pass Go example [8.18 ex 129](#), [9.5 ex 129](#)
does the player mean *rulebook* [17.19](#)
doing it, a repeated action [7.17](#)
doing something / anything, describing actions [7.9](#)
 "[don't]" *say phrase* [14.8](#)
Don Pedro's Revenge example [8.2 ex 113](#), [7.5 ex 113](#)
door (*direction*) **from/of** (*room*) *phrase* [6.14](#)
 doors *kind* [3.12](#), [3.5](#)
 locked [3.13](#)
 rules for going through doors [7.14](#)
 double-dash *see* [punctuation: double-dash](#)
Down Below example [3.14 ex 26](#), [3.7 ex 26](#)
Down in Oodville example [17.9 ex 304](#), [6.18 ex 304](#)
drinking *action* [10.2](#)
dropping *action* [6.8](#)
Dubai example [16.3 ex 271](#), [8.2 ex 271](#)
 duplicates [4.14](#)
 taken from a dispenser [10.3](#)
during (*scene*) [10.4](#), [19.7](#)
dynamic memory allocation *use option* [20.2](#)

E

e — *real number constant* [15.2](#)
each, in descriptions [6.15](#)
each other, in defining relations [13.6](#), [13.7](#)
Early Childhood example [4.14 ex 55](#), [9.7 ex 55](#)
eating *action* [3.22](#), [9.1](#)
Eddystone example [12.17 ex 213](#), [3.4 ex 213](#)
edible / inedible (*thing*) *property* [3.22](#), [9.1](#)
 either/or properties [3.6](#), [4.6](#)
 changing [8.4](#)
 defining [4.7](#)
 used as adjectives [6.3](#)
 electricity (implementing) [10.7](#)
Electrified example [19.15 ex 408](#), [10.7 ex 408](#)
else (*a phrase*) *phrase* [11.8](#)
else if (*a condition*) *phrase* [11.8](#)
else unless (*a condition*) *phrase* [11.8](#)
Elsie example [9.13 ex 151](#), [3.5 ex 151](#)
Emma example [18.24 ex 353](#), [7.16 ex 353](#)
Empire example [9.11 ex 145](#), [8.2 ex 145](#)
empty / non-empty *adjective*
 for activities [18.5](#)
 for lists [21.1](#), [21.2](#)
 for relations [13.14](#)

- for rulebooks 19.8
 - for tables 16.9
 - for texts 6.5, 20.3
- enclosure *relation* 3.25
- end if 11.7
- end, in phrases 11.7, 11.9
- end of scene *see* scenes: rules run at end
- end of story 9.4, 18.38, 18.39, 11.6
- end the (activity) activity *phrase* 18.7
- end the (activity on values) activity with (value) *phrase* 18.7
- end the story *phrase* 9.4, 11.6
- end the story finally *phrase* 9.4, 11.6
- end the story finally saying (text) *phrase* 9.4, 11.6
- end the story saying (text) *phrase* 9.4, 11.6
- end while 11.9
- ended -- if (scene) has ended *phrase* 10.4
- Endurance *example* 19.15 ex 410, 4.1 ex 410
- engineering notation *use option* 15.2
- English
 - defining plural forms 4.4, 2.2
 - displaying numbers in words 5.4
 - displaying simple plurals 5.4
 - displaying times in words 9.7
 - forms for sentence verbs 13.9
 - forms of to be 13.10
 - irregular verbs 12.8, 13.9
 - narrative viewpoint, of standard responses 14.1, 5.6
 - really understanding 2.16
 - simple superlative forms 6.8
 - tense
 - of conditions in rules 9.12, 9.13
 - of standard responses 14.1, 5.6
- enterable (container) *property* 3.6, 9.2
 - vehicles made automatically enterable 3.16
- entering *action* 6.10
- Entire Game — scene *constant* 10.3
- Entrapment *example* 10.2 ex 155, 4.2 ex 155
- Entrevaux *example* 10.6 ex 163, 4.4 ex 163
- entries, in tables *see* tables
- Entropy *example* 11.18 ex 182, 10.9 ex 182
- entry (number) of (list) 21.9
- enumerated value 22.1, 22.6; *see also* kinds: of value
- Epistemology *extension* 5.5
- EPS file *see* index map: customizing for release
- equality *relation* 13.8
- equations, for values with units 15.18
- Equipment List *example* 11.14 ex 177, 6.7 ex 177
- equivalence (relation) *adjective* 13.14
- equivalent to (unit) 15.10
- error messages
 - abject failures of Inform 2.15
 - for player commands 18.35, 6.17
 - for source text 2.4
- Escape *example* 3.12 ex 21, 3.6 ex 21
- Escape from the Seraglio *example* 19.15 ex 411, 6.15 ex 411
- Euler, Leonhard 15.2
- Eve, Eric 1.2, 5.5
- even / odd (number) *adjective* 4.6, 6.5
- events in the future 9.11
- every
 - creating assemblies 4.15, 4.16, 5.1
 - in descriptions 6.15

every turn *rulebook* 9.5, 7.2
 not for **out of world** actions 12.15
everywhere, placing backdrops 3.9, 8.8
exactly
 counting conditions 9.14, 9.15
 in descriptions 6.16
examining *action* 6.5
 handling **READ** separately 9.6
except
 describing actions 7.9
 in descriptions 6.16
 exclamation mark *see* punctuation: exclamation mark, ending sentences
 existence
 descriptions of any such thing 6.2
 testing that such a thing exists 6.10
existent / nonexistent (*real number*) *adjective* 15.3
exiting *action* 7.14, 6.9, 6.10
 exits *see* connections between rooms
exponential of (*real number*) *phrase* 15.6
extend (*list of values*) **to** (*number*) **entries/entry** *phrase* 21.10
 extensions
 built-in extensions 27.3
 Inform 6 code in extensions 27.14
 Inform 6 template layer 27.26, 6.3
 installing 2.10, 27.3
 licensing of use 27.1, 1.1
 listing credits for 27.6, 11.5
 project-specific extensions 27.3
 specific extensions
 Approximate Metric Units 15.13
 Basic Screen Effects 27.7, 4.5
 Epistemology 5.5
 Inanimate Listeners 7.7, 9.6
 Locksmith 2.10, 2.11, 3.5, 9.2
 Metric Units 15.13, 15.18, 15.20
 Mistype 6.17
 Object Response Tests 24.6
 Punctuation Removal 6.13, 7.1, 7.7
 Real Time Delays 12.4
 Rideable Vehicles 3.16, 8.3
 Skeleton Keys 9.2
 Standard Rules 27.2, 27.14
 Unicode Character Names 5.11, 27.25
 Unicode Full Character Names 5.11, 27.25
 Vorple system 25.12
 using 2.11
 using specific versions 27.6
 writing
 author *bibliographic data* 27.4
 documenting 27.11, 27.12
 example 27.5
 for a specific virtual machine 27.7
 giving examples 27.12
 including other extensions 27.8, 27.9
 modifying other extensions 27.9
 version number 27.6
 external files *see* files (data files)
Extra Supplies *example* 8.10 ex 124, 10.3 ex 124
The Eye of the Idol *example* 18.25 ex 357, 3.1 ex 357
Eyes, Fingers, Toes *example* 21.11 ex 433, 9.2 ex 433
F *Fabrication* *example* 15.20 ex 269, 9.4 ex 269
The Facts Were These *example* 21.10 ex 431, 6.15 ex 431

failure

- for actions [7.1](#), [12.2](#)
- for persuasion [12.4](#)
- for rules [19.11](#)

Fallout Enclosure [example 3.23 ex 35, 9.2 ex 35](#)

false / true [11.5](#)

Farewell [example 16.15 ex 281, 7.8 ex 281](#)

fast route-finding [use option 6.14](#)

- for particular relations [13.11](#)

Fate Steps In [example 12.13 ex 209, 4.6 ex 209](#)

feelies [25.7](#)

Feline Behavior [example 19.13 ex 406, 8.3 ex 406](#)

female / male / neuter (person) property [3.17](#)

Ferragamo Again [example 11.3 ex 170, 7.10 ex 170](#)

The Fibonacci Sequence [example 21.11 ex 434, 10.11 ex 434](#)

Fido [example 20.8 ex 420, 8.3 ex 420](#)

The Fifth Body [example 23.14 ex 444, 9.6 ex 444](#)

Fifty Times Fifty Ways [example 5.13 ex 72, 2.1 ex 72](#)

Fifty Ways to Leave Your Larva [example 5.13 ex 71, 2.1 ex 71](#)

figure name [23.10](#)

figures [23.2](#), [12.5](#)

- accessibility text [23.9](#)
- as values [23.10](#)
- cover art [25.8](#)
- declaring [23.5](#)
- displaying [23.6](#)
- formats allowed [23.4](#)
- placing in materials folder [23.4](#)

files (compiling)

- blorb file [see blorb file](#)
- extensions [2.10](#), [27.3](#)
- materials folder [see materials folder](#)
- Options.txt [2.13](#)

files (data files) [23.11](#), [12.5](#)

- declaring [23.12](#)
- exchanging with other programs [23.12](#), [23.15](#)
- loading initial state from a data file [11.1](#)
- reading / writing / appending text [23.14](#)
- reading / writing tables [23.13](#)
- storing records of death in a data file [11.6](#)

filter to (description of values) of (list of values) phrase [22.5](#)

Final Question Options -- Table of Final Question Options [11.6](#)

Finality [example 18.38 ex 384, 11.6 ex 384](#)

finally, ending the story [9.4](#), [18.39](#), [11.6](#)

FIND [5.4](#)

Fine Laid [example 7.4 ex 88, 6.14 ex 88](#)

Finishing School [example 6.4 ex 73, 5.3 ex 73](#)

finite / infinite (real number) adjective [15.3](#)

fire (implementing) [10.8](#)

first, listed first in rulebook [19.4](#), [19.7](#)

First Name Basis [example 3.5 ex 11, 2.2 ex 11](#)

first person singular / plural [14.1](#)

first thing held by (object) phrase [8.17](#)

first time, counting actions [7.16](#)

first time, counting conditions [9.14](#)

first value of (name of kind) phrase [11.18](#)

Fisher, David [1.2](#)

fixed in place / portable (thing) property [3.6](#), [3.7](#), [8.7](#)

Flashlight [example 12.19 ex 218, 6.6 ex 218](#)

Flathead News Network [example 23.15 ex 445, 12.5 ex 445](#)

Fleming, Ian [6.10](#)

floating-point [see real numbers](#)

[floor of \(real number\) phrase](#) [15.6](#)
[Flotation example](#) [19.9 ex 403](#), [10.2 ex 403](#)
[fluids](#) *see* [liquids \(implementing\)](#)
[follow \(rule\) phrase](#) [19.8](#)
[follow \(values based rule producing values\) for \(value\) phrase](#) [19.9](#)
[fonts](#)
 [fixed-width / variable-width](#) [5.9](#), [12.1](#)
 [italic / bold / roman](#) [5.9](#), [12.1](#)
[food](#) [3.22](#), [9.1](#)
[Food Network Interactive example](#) [16.18 ex 284](#), [11.3 ex 284](#)
[footnotes \(implementing\)](#) [12.3](#)
[for \(activity\) rulebook](#) [18.3](#)
[For Demonstration Purposes example](#) [13.13 ex 239](#), [7.14 ex 239](#)
[for release only](#) [2.9](#), [25.6](#), [13.1](#)
[Fore example](#) [3.26 ex 42](#), [8.2 ex 42](#)
[Formal syntax of sentences example](#) [13.2 ex 223](#), [1.4 ex 223](#)
[Formicidae example](#) [21.6 ex 428](#), [6.15 ex 428](#)
[Four Cheeses example](#) [13.6 ex 226](#), [9.10 ex 226](#)
[Four Stars 1 example](#) [18.22 ex 349](#), [3.7 ex 349](#)
[Four Stars 2 example](#) [18.29 ex 364](#), [3.8 ex 364](#)
[The Fourth Body example](#) [23.14 ex 443](#), [9.6 ex 443](#)
[fragile things \(implementing\)](#) [10.4](#)
[Fragment of a Greek Tragedy example](#) [18.33 ex 374](#), [6.18 ex 374](#)
[Frizz example](#) [12.7 ex 192](#), [10.2 ex 192](#)
[from now -- \(rule\) in \(number\) turn/turns from now phrase](#) [9.11](#)
[from now -- \(rule\) in \(time\) from now phrase](#) [9.11](#)
[from -- inside / outside from](#) [3.2](#)
[front side of \(object\) phrase](#) [3.12](#)
[Frozen Assets example](#) [15.19 ex 264](#), [9.4 ex 264](#)
[full-length room descriptions](#) *use option* [2.12](#)
[Full Moon example](#) [10.4 ex 157](#), [3.9 ex 157](#)
[full stop](#) *see* [punctuation: full stop](#)
[Fun with Participles example](#) [14.3 ex 245](#), [2.1 ex 245](#)
[Further Reasons Why All Poets Are Liars example](#) [12.10 ex 199](#), [3.3 ex 199](#)
[future / present / past tense](#) [14.1](#)
[future events](#) [9.11](#)

G

[Galactic Patrol](#) [20.6](#)
[game structure](#) *see* [story structure](#)
[games, dice and cards \(implementing\)](#) [9.5](#)
[Garibaldi 1 example](#) [3.12 ex 22](#), [3.5 ex 22](#)
[Garibaldi 2 example](#) [5.9 ex 69](#), [12.1 ex 69](#)
[gases \(implementing\)](#) [10.1](#)
[Generation X example](#) [12.5 ex 188](#), [7.14 ex 188](#)
[genre](#) *bibliographic data* [25.2](#), [25.3](#)
[Get Axe example](#) [12.12 ex 206](#), [6.10 ex 206](#)
[Get Me to the Church on Time example](#) [4.4 ex 46](#), [9.3 ex 46](#)
[Ginger Beer example](#) [18.29 ex 365](#), [9.11 ex 365](#)
[given, in calculations with equations](#) [15.18](#)
[giving it to action](#) [7.4](#)
[global constants](#) *see* [constant values](#)
[global variables](#) *see* [variables: global](#)
[Glulx](#) [2.14](#), [23.3](#)
 [extensions for Glulx only](#) [27.7](#)
 [memory limits](#) [2.14](#), [20.2](#), [21.10](#)
 [no character set limits](#) [20.4](#)
 [no support for Vorple](#) [25.12](#)
 [numeric limits](#) [11.10](#), [15.2](#), [15.10](#), [15.14](#)
 [real numbers available](#) [15.2](#)
 [required for multimedia](#) [23.1](#), [12.5](#)
[Glulx resource ID of \(figure name\) phrase](#) [23.10](#)
[Glulx resource ID of \(sound name\) phrase](#) [23.10](#)
[Go button](#) *user interface* [1.4](#)

Goat-Cheese and Sage Chicken example [16.14 ex 280](#), [11.4 ex 280](#)
 God's own teacup [16.18](#)
going *action* [7.13](#), [6.9](#)
 default failure response [7.13](#)
 going by *action* [7.14](#)
 going from *action* [7.13](#)
 going nowhere *action* [7.13](#)
 going through *action* [7.14](#)
 going to *action* [7.13](#)
 going with *action* [7.14](#)
 other characters moving around [7.13](#)
 rules for going by vehicles [7.14](#)
 rules for going through doors [7.14](#)
 rules for going to and from places [7.13](#)
 rules for going with pushed things [7.14](#)
going on (activity) *adjective* [18.2](#)
 GONEAR *testing command* [24.4](#)
Gopher-wood example [3.18 ex 32](#), [7.1 ex 32](#)
The Gorge at George example [17.21 ex 330](#), [7.6 ex 330](#)
 grammar of player's command *see* [understanding](#)
 grammar tokens [17.1](#), [17.4](#), [6.2](#)
 defining [17.13](#)
 defining in Inform 6 [27.23](#)
 for kinds of value [17.9](#), [17.11](#)
 for numbers [17.6](#), [17.9](#)
 for rooms [17.7](#), [5.5](#)
 for text [17.5](#)
 for things not in scope [17.7](#), [5.5](#)
 producing values [17.14](#)
Graph-theory view of relations example [13.16 ex 244](#), [1.4 ex 244](#)
 graphics *see* [figures](#)
greater than *relation verb* [15.5](#)
Grilling example [7.2 ex 84](#), [10.9 ex 84](#)
group (description of objects) together *phrase* [18.13](#)
group (description of objects) together as (text) *phrase* [18.13](#)
group (description of objects) together giving articles *phrase* [18.13](#)
 group relations [13.7](#)
grouping together something *activity* [18.14](#)
Guided Tour example [18.23 ex 351](#), [12.2 ex 351](#)

H

had been, for past conditions [9.13](#)
handled (thing) *property* [3.11](#)
The Hang of Thursdays example [11.18 ex 183](#), [4.1 ex 183](#)
happened -- if (scene) has happened *phrase* [10.4](#)
happening (scene) *adjective* [10.4](#)
Happy Hour example [18.25 ex 356](#), [7.16 ex 356](#)
 Hardy, Thomas [4.5](#)
has *relation verb* *see* [possession](#)
has been, for past conditions [9.13](#)
has, for defining properties [4.8](#)
Hatless example [8.11 ex 126](#), [11.1 ex 126](#)
A Haughty Spirit example [11.17 ex 181](#), [3.6 ex 181](#)
 "[have]" *say phrase* [14.2](#)
 "[haven't]" *say phrase* [14.8](#)
Hays Code example [18.9 ex 338](#), [9.3 ex 338](#)
Hayseed example [7.4 ex 89](#), [3.5 ex 89](#)
 Hazel, Philip [20.6](#)
 headings [2.5](#), [2.6](#), [2.9](#), [25.6](#)
 in extension documentation [27.12](#)
 in extensions [27.9](#), [27.10](#)
 headline *bibliographic data* [18.37](#), [25.2](#)
 heat
 on fire [10.8](#)

too hot to touch [10.9](#)

held by -- first thing held by (object) phrase [8.17](#)

held by -- next thing held after (object) phrase [8.17](#)

held in relation verb see [containment](#)

held (thing) adjective [3.20](#), [8.13](#)

help systems [11.3](#)

HER, in player commands [17.18](#), [2.2](#)

hiding things carried by other characters [3.24](#), [18.9](#)

hiding things from room descriptions by making them scenery [3.8](#)

hiding things under other things [6.6](#)

Higher Calling example [8.15](#) ex [128](#), [3.5](#) ex [128](#)

HIM, in player commands [17.18](#), [2.2](#)

hint systems [11.3](#)

historical conditions [9.13](#)

 times [9.14](#)

 turns [9.15](#)

 warning concerning variables and other varying qualities [9.13](#), [9.15](#)

History Lab example [14.9](#) ex [250](#), [2.1](#) ex [250](#)

Hohmann Transfer example [18.22](#) ex [348](#), [3.7](#) ex [348](#)

hold-all [3.21](#)

holder of (object) phrase [8.17](#)

Hot Glass Looks Like Cold Glass example [17.21](#) ex [331](#), [10.9](#) ex [331](#)

Hotel Stechelberg example [6.14](#) ex [79](#), [3.4](#) ex [79](#)

Hour of the Wren example [9.11](#) ex [146](#), [4.3](#) ex [146](#)

(number) hours phrase [9.8](#)

hours part of (time) phrase [9.10](#)

Hover example [3.16](#) ex [29](#), [8.1](#) ex [29](#)

Hudsucker Industries example [18.11](#) ex [342](#), [9.2](#) ex [342](#)

A Humble Wayside Flower example [13.12](#) ex [237](#), [7.1](#) ex [237](#)

Hunter, Andrew [1.2](#)

Hymenaeus example [17.15](#) ex [312](#), [3.7](#) ex [312](#)

hyperbolic arccosine of (real number) phrase [15.7](#)

hyperbolic arcsine of (real number) phrase [15.7](#)

hyperbolic arctangent of (real number) phrase [15.7](#)

hyperbolic cosine of (real number) phrase [15.7](#)

hyperbolic sine of (real number) phrase [15.7](#)

hyperbolic tangent of (real number) phrase [15.7](#)

The Hypnotist of Blois example [12.4](#) ex [186](#), [7.14](#) ex [186](#)

I I Didn't Come All The Way From Great Portland Street example [21.11](#) ex [435](#), [6.14](#) ex [435](#)

Ibid. example [17.6](#) ex [300](#), [12.3](#) ex [300](#)

IDE interface see [user interface](#)

Identity Theft example [20.7](#) ex [416](#), [5.2](#) ex [416](#)

if (a condition) phrase [2.2](#), [8.11](#), [8.13](#), [11.6](#)

 with generic conditions [8.14](#)

 with otherwise/else [11.8](#)

if (a condition) , (a phrase) phrase [11.6](#)

if a paragraph break is pending phrase [5.8](#)

if a random chance of (number) in (number) succeeds phrase [8.18](#)

if (action) involves (object) phrase [12.20](#)

if action requires a carried noun phrase [12.7](#)

if action requires a carried second noun phrase [12.7](#)

if action requires a touchable noun phrase [12.7](#)

if action requires a touchable second noun phrase [12.7](#)

if action requires light phrase [12.7](#)

IF Archive web site [26.3](#), [26.4](#), [26.6](#)

if (external file) exists phrase [23.13](#)

if handling (activity) activity phrase [18.7](#)

if handling (activity on values) activity with (value) phrase [18.7](#)

if in darkness phrase [11.5](#)

If It Hadn't Been For... example [16.9](#) ex [273](#), [9.12](#) ex [273](#)

if (name of kind) relates to (value) by (relation of values to values) phrase [13.13](#)

if player consents phrase [11.5](#)

if ready to read (*external file*) *phrase* [23.15](#)
 if rule failed *phrase* [19.11](#)
 if rule succeeded *phrase* [19.11](#)
 if (*scene*) has ended *phrase* [10.4](#)
 if (*scene*) has happened *phrase* [10.4](#)
 if (*scene*) has not ended *phrase* [10.4](#)
 if (*scene*) has not happened *phrase* [10.4](#)
 if (*snippet*) does not include (*topic*) *phrase* [18.33](#)
 if (*snippet*) does not match (*topic*) *phrase* [18.33](#)
 if (*snippet*) includes (*topic*) *phrase* [18.33](#)
 if (*snippet*) matches (*topic*) *phrase* [18.33](#)
 if story has ended *phrase* [9.4](#)
 if story has ended finally *phrase* [9.4](#)
 if story has not ended *phrase* [9.4](#)
 if story has not ended finally *phrase* [9.4](#)
 if (*text*) exactly matches the regular expression (*text*) *phrase* [20.6](#)
 if (*text*) exactly matches the text (*text*) *phrase* [20.5](#)
 if (*text*) is in lower case *phrase* [20.4](#)
 if (*text*) is in upper case *phrase* [20.4](#)
 if (*text*) matches the regular expression (*text*) *phrase* [20.6](#)
 if (*text*) matches the text (*text*) *phrase* [20.5](#)
 if there is (*a table entry*) *phrase* [16.7](#)
 if there is no (*a table entry*) *phrase* [16.7](#)
 if (*time*) is after (*time*) *phrase* [9.9](#)
 if (*time*) is before (*time*) *phrase* [9.9](#)
 if (*value*) is *phrase* [11.8](#)
 if (*value*) is listed in (*list of values*) *phrase* [21.4](#)
 if (*value*) is not listed in (*list of values*) *phrase* [21.4](#)
 if (*value*) matches (*description of values*) *phrase* [22.2](#)
 if (*value*) relates to (*name of kind*) by (*relation of values to values*) *phrase* [13.13](#)
 IFDB *web site* [25.8](#), [26.3](#), [26.4](#), [26.6](#)
 IFID *bibliographic data* [23.12](#), [23.15](#), [25.5](#), [26.6](#), [11.5](#)
 ifwiki *web site* [26.5](#), [26.7](#), [1.2](#)
 Igpay Atinlay *example* [20.8](#) ex 421, [2.3](#) ex 421
 images *see* [figures](#)
 immobile things
 fixed in place [3.6](#), [8.7](#)
 moving anyway [8.7](#)
 scenery [3.8](#)
 supporters made automatically immobile [3.7](#)
 implicitly taking something *activity* [12.17](#), [18.34](#), [6.8](#)
 improper-named / proper-named (*thing*) *property* [3.18](#)
 in *relation verb* *see* [containment](#)
 In Fire or in Flood *example* [19.8](#) ex 401, [10.8](#) ex 401
 in groups, in defining relations [13.7](#)
 in, location conditions for rules [7.11](#), [7.13](#)
 (*text*) in lower case *phrase* [20.4](#)
 (*text*) in sentence case *phrase* [20.4](#)
 in the presence of [7.12](#), [18.29](#)
 (*rule*) in (*time*) from now *phrase* [9.11](#)
 (*text*) in title case *phrase* [20.4](#)
 (*rule*) in (*number*) turn/turns from now *phrase* [9.11](#)
 (*text*) in upper case *phrase* [20.4](#)
 inactive / active (*use option*) *adjective* [27.18](#)
 Inanimate Listeners *extension* [7.7](#), [9.6](#)
 include (*extension*) by (*extension author*) *assertion* [2.11](#)
 incorporation *relation* [3.23](#), [4.15](#), [8.13](#), [13.4](#)
 increase (*a stored value*) by (*value*) *phrase* [8.12](#), [9.2](#)
 increment (*a stored value*) *phrase* [8.12](#)
 indefinite article of (*object*) *property* [3.18](#)
 indefinite articles
 when creating things [3.18](#)

- when displaying lists [5.5](#), [21.3](#)
 - when displaying things [5.3](#)
- indentation *of source text* [11.7](#), [11.9](#)
- index *of source text* *see* [headings](#)
- index map [3.2](#), [3.4](#), [3.26](#), [3.2](#)
 - customizing for release [25.18](#), [13.2](#)
- Index panel *user interface* [1.6](#)
 - Actions page [12.5](#), [12.21](#), [17.1](#), [17.2](#), [17.3](#), [24.3](#)
 - checking story correctness [24.1](#)
 - Contents page [2.6](#), [23.5](#), [27.10](#)
 - Kinds page [4.1](#), [4.6](#), [4.8](#), [4.11](#), [4.12](#), [27.10](#)
 - Map page [2.7](#), [3.4](#), [3.26](#), [25.18](#)
 - Phrases page [2.2](#), [11.2](#)
 - Rules page [19.6](#), [19.16](#)
 - Scenes page [10.3](#), [10.9](#)
- indexed text, now the same as text *see* [text](#)
- indirect containment [3.25](#), [6.11](#)
- Indirection example* [17.1 ex 286](#), [3.2 ex 286](#)
- inedible / edible** (*thing*) *property* [3.22](#), [9.1](#)
- Infiltration example* [9.14 ex 152](#), [3.1 ex 152](#)
- infinite / finite** (*real number*) *adjective* [15.3](#)
- Infocom [2.12](#), [9.3](#), [25.7](#), [12.2](#), [12.3](#)
- inform7.com *web site* [2.15](#), [27.3](#)
- Inform 6 [1.1](#), [2.14](#), [27.14](#)
- Inform 6 Designer's Manual* [18.29](#), [27.21](#), [27.23](#), [27.27](#)
- Inform 6 equivalent
 - attributes [4.7](#)
 - BeforeParsing** [18.33](#)
 - box** [5.12](#)
 - break** [11.12](#)
 - child** [8.17](#)
 - classes [4.1](#)
 - clothing** [3.20](#)
 - code blocks [11.7](#)
 - constants [4.13](#)
 - continue** [11.12](#)
 - daemons [9.5](#)
 - darkness [12.19](#)
 - each_turn** [9.5](#)
 - for** [11.10](#)
 - functions [11.2](#)
 - functions with optional parameters [11.14](#)
 - functions with parameters [11.3](#)
 - global variable [4.12](#)
 - IndirectlyContains** [3.25](#)
 - initial** [3.11](#)
 - local variables [11.15](#)
 - meta** [12.15](#)
 - objectloop** [11.11](#)
 - parent** [8.17](#)
 - procedures [11.2](#)
 - properties [4.8](#)
 - return** [11.13](#)
 - return value [11.17](#)
 - rtrue/rfalse** [11.16](#)
 - SACK_OBJECT** [3.21](#)
 - scope rules [12.16](#), [12.18](#), [18.29](#)
 - sibling** [8.17](#)
 - switch/case** [11.8](#)
 - switchable** [3.14](#)
 - synonyms [12.7](#)
 - timers [9.11](#)

- verbs [12.7](#)
- while [11.9](#)
- with_key [3.13](#)
- Inform 6 inclusions [27.14](#)
 - actions [27.22](#)
 - adjectives [27.24](#)
 - classes [27.21](#)
 - global variables [27.22](#)
 - grammar tokens [27.23](#)
 - Inform 6 template layer [27.26](#), [6.3](#)
 - long sections of code [27.19](#)
 - objects [27.21](#)
 - phrase elements [27.28](#), [27.29](#), [27.30](#)
 - phrase options [27.17](#)
 - phrases [27.15](#)
 - phrases to decide a value [27.16](#)
 - phrases to decide whether/if [27.16](#)
 - properties [27.21](#), [27.22](#)
 - rules [27.20](#)
 - understanding grammar [27.23](#)
 - use options [27.18](#)
- information *see* [knowledge \(in story world\)](#)
- Ingold, Jon [1.2](#)
- initial appearance of (thing) property [3.11](#), [8.5](#)
- initial location of player [3.1](#)
- initial state of the world [2.1](#), [3.1](#)
 - scenes that begin at the beginning [10.6](#)
 - setting in when play begins [9.1](#), [11.1](#)
- initially, in defining global variables [4.12](#)
- inserting it into action [6.8](#)
- inside *relation verb* *see* [containment](#)
- inside from [3.2](#)
- instead of (action) *rulebook* [7.2](#), [7.3](#)
 - as failure of persuasion [12.4](#)
 - guidelines for use [12.21](#)
 - in action processing sequence [12.2](#), [6.3](#)
 - not used for out of world actions [12.15](#)
- instead of, replacing rules [19.4](#)
- instead, to stop the action [7.3](#), [12.9](#), [19.11](#), [6.3](#)
- instructing other characters [12.3](#), [7.14](#)
- insufficient / sufficient light, there is insufficient / sufficient light (*visibility rules*) *outcome phrase* [12.19](#)
- integers *see* [numbers](#)
- interface *see* [user interface](#)
- Interrogation *example* [13.4 ex 225](#), [10.10 ex 225](#)
- intfiction forum *web site* [24.8](#), [26.3](#)
- introduction *see* [initial state of the world](#)
- Introduction to Juggling *example* [16.16 ex 283](#), [9.4 ex 283](#)
- introductory booklet and postcard [25.9](#)
- inventory details -- printing inventory details of something *activity* [18.17](#)
- inventory -- taking inventory *action* [6.7](#)
- invisible / visible (thing) *adjective* [6.13](#)
- IPA *example* [9.6 ex 141](#), [4.3 ex 141](#)
- IQ Test *example* [12.6 ex 189](#), [7.15 ex 189](#)
- is [2.1](#), [13.1](#), [13.2](#), [13.8](#)
 - equality relation *see* [equality](#)
- Ish. *example* [17.5 ex 296](#), [11.3 ex 296](#)
- issuing the response text of something *activity* [18.15](#)
- "[it]" *say phrase* [14.5](#)
- it does / it does not (*deciding whether all includes*) *outcome phrase* [18.36](#)
- IT, in player commands [17.18](#), [2.2](#)
- it, in source text [11.16](#)

- it is likely / possible / unlikely (*does the player mean*) *outcome phrase* [17.19](#)
- item described — *object global variable* [17.17](#), [2.2](#)
- J**
 - Jamaica 1688 example* [18.38](#) ex [385](#), [11.6](#) ex [385](#)
 - Jerome, Jerome K. [2.5](#), [25.2](#)
 - Jerz, Dennis [26.9](#)
 - Jokers Wild example* [16.11](#) ex [275](#), [9.5](#) ex [275](#)
- K**
 - Kesserich, Sonja [1.2](#)
 - keyword-style parsing [6.18](#)
 - Kinder, David [1.2](#)
 - kinds [3.5](#), [4.1](#), [22.1](#)
 - catalogue
 - animal [3.17](#)
 - backdrop [3.9](#); *see also* [backdrops](#)
 - container [3.5](#); *see also* [containers](#)
 - device [3.14](#), [9.8](#), [10.7](#)
 - direction [3.5](#), [3.26](#)
 - door [3.12](#); *see also* [doors](#)
 - man [3.17](#)
 - person [3.17](#); *see also* [person](#)
 - player's holdall [3.21](#)
 - region [3.4](#); *see also* [regions](#)
 - room [3.5](#); *see also* [rooms](#)
 - supporter [3.5](#); *see also* [supporters](#)
 - thing [3.5](#); *see also* [things](#)
 - vehicle [3.16](#), [8.1](#)
 - woman [3.17](#)
 - default properties [4.3](#)
 - default values of kinds [4.8](#), [4.11](#), [4.12](#), [22.4](#)
 - defining [4.1](#)
 - in Inform 6 [27.21](#)
 - kinds of thing [4.1](#), [4.2](#), [4.4](#)
 - kinds of value [4.5](#), [4.9](#), [4.10](#)
 - defining with a table [16.17](#)
 - multiple kinds with a table [16.16](#)
 - documenting in extensions [27.10](#)
 - enumerated values [4.5](#), [4.9](#), [4.10](#)
 - hierarchy of kinds [4.1](#)
 - in phrase parameters [11.3](#), [22.8](#)
 - kind variables [22.1](#), [22.7](#)
 - must define before using [4.2](#)
 - of action [7.15](#)
 - of description [22.2](#)
 - of kinds [22.1](#), [22.6](#)
 - of list [21.1](#), [21.2](#)
 - of phrase [22.3](#)
 - default values for [22.4](#)
 - of relation [13.14](#)
 - of thing [4.1](#)
 - of value [4.1](#), [4.5](#)
 - of values with units of measure [15.8](#)
 - uses of new kinds [4.2](#)
 - Kinds page of Index panel *user interface* [4.1](#), [4.6](#), [4.8](#), [4.11](#), [4.12](#), [27.10](#)
 - Kiwi example* [18.28](#) ex [361](#), [8.4](#) ex [361](#)
 - Kleene, Stephen [20.6](#)
 - Knight, Cedric [6.17](#)
 - knowledge (in story world)
 - other characters' knowledge [7.11](#)
 - player's knowledge [5.4](#), [5.5](#)
 - Knuth, Donald [1.1](#)
 - Kyoto example* [19.11](#) ex [404](#), [10.4](#) ex [404](#)
 - L**
 - Labyrinth of Ghosts example* [23.13](#) ex [441](#), [11.6](#) ex [441](#)

Lakeside Living *example* [17.22 ex 333](#), [10.2 ex 333](#)
Lanista 1 *example* [8.18 ex 130](#), [7.5 ex 130](#)
Lanista 2 *example* [17.3 ex 293](#), [7.5 ex 293](#)
last, listed last in rulebook [19.4](#), [19.7](#)
last value of (*name of kind*) *phrase* [11.18](#)
latest parser error — *command parser error* *global variable* [18.35](#)
Latin Lessons *example* [18.32 ex 371](#), [7.14 ex 371](#)
Latris Theon *example* [12.3 ex 185](#), [7.13 ex 185](#)
Laura *example* [3.11 ex 19](#), [2.2 ex 19](#)
laws for sorting rulebooks [19.6](#), [19.16](#)
Le Guin, Ursula K. [3.12](#)
Lead Cuts Paper *example* [15.12 ex 259](#), [10.5 ex 259](#)
leading zeroes, when defining units [15.15](#)
Lean and Hungry *example* [6.13 ex 76](#), [7.2 ex 76](#)
Lebling, Dave [2.5](#), [2.13](#)
The Left Hand of Autumn *example* [17.4 ex 295](#), [6.5 ex 295](#)
left hand status line — *text* *global variable* [8.3](#), [18.23](#), [12.2](#)
Leinonen, Juhana [1.2](#), [24.6](#), [25.12](#)
Lemonade *example* [15.19 ex 266](#), [10.2 ex 266](#)
Leopard-skin *example* [21.10 ex 430](#), [6.14 ex 430](#)
less than *relation verb* [15.5](#)
less than, in descriptions [6.16](#)
let (*a name not so far used*) **be** (*description of relations of values to values*) *phrase* [13.15](#)
let (*a name not so far used*) **be given by** (*equation name*) *phrase* [15.18](#)
let (*a name not so far used*) **be** (*name of kind*) *phrase* [11.15](#)
let (*a name not so far used*) **be** (*value*) *phrase* [11.15](#)
let (*a temporary named value*) **be given by** (*equation name*) *phrase* [15.18](#)
let (*a temporary named value*) **be** (*value*) *phrase* [11.15](#)
Lethal Concentration 1 *example* [15.8 ex 255](#), [10.1 ex 255](#)
Lethal Concentration 2 *example* [15.8 ex 257](#), [10.1 ex 257](#)
letters *see* [characters \(letters\)](#)
Lies *example* [17.13 ex 310](#), [6.10 ex 310](#)
light [3.15](#), [6.13](#), [3.7](#)
 lack of *see* [darkness](#)
 printing the announcement of light *activity* [18.20](#)
 requirement in defining actions [12.7](#), [12.17](#)
 rules to override sufficiency of [12.19](#)
lighted / dark (*room*) *property* [3.15](#), [3.7](#)
likely / possible / unlikely, it is likely / possible / unlikely (*does the player mean*) *outcome*
 phrase [17.19](#)
limits
 for extending lists [21.10](#)
 for manipulating text [20.2](#)
 of Inform 6 compiler [2.14](#)
 of numeric values [11.10](#), [15.2](#)
 of numeric values of units [15.10](#), [15.14](#)
 of scene changing mechanism [10.9](#)
 of story file [2.14](#)
 on changing lists being repeated through [21.4](#)
line breaks [5.6](#), [5.8](#)
 counting lines in text [20.3](#)
 in formatting tables [16.1](#)
 produced by sentence-ending punctuation [2.3](#), [5.2](#), [5.8](#)
line number (*number*) **in** (*text*) *phrase* [20.3](#)
liquids (implementing) [10.2](#)
"[list of (*description*)]" *say phrase* [6.1](#)
list of (*description of values*) *phrase* [21.6](#)
list of (*name of kind*) **that/which/who** relate to (*value*) **by** (*relation of values to values*) *phrase* [13.13](#)
list of (*name of kind*) **that/which/whom** (*relation of values to values*) **relates** *phrase* [13.13](#)
list of (*name of kind*) **that/which/whom** (*relation of values to values*) **relates to** *phrase* [13.13](#)
list of (*name of kind*) **that/which/whom** (*value*) **relates to by** (*relation of values to values*) *phrase* [13.13](#)
list of (*name of kind*) **to which/whom** (*relation of values to values*) **relates** *phrase* [13.13](#)

list of *(name of kind)* to which/whom *(value)* relates by *(relation of values to values)* phrase [13.13](#)
 list the contents of *(object)* phrase [11.14](#)
 listed in / before / after / instead of, for placing rules in rulebooks [19.4](#)
 listening to *action* [7.7](#), [3.8](#)
 listing contents of something *activity* [18.13](#)
 listing group size — *number* *global variable* [18.12](#), [18.14](#)
 listing nondescript items of something *activity* [18.25](#), [3.1](#)
 listing rules
 for grouping together [18.14](#)
 for listing contents [18.13](#)
 lists [13.13](#), [21.1](#)
 adding entries [21.1](#), [21.5](#)
 changing entries [21.9](#)
 changing length [21.10](#)
 constant lists [21.2](#)
 creating [21.2](#)
 by description [21.6](#), [21.7](#)
 displaying [5.5](#), [21.3](#)
 empty / non-empty *adjective* [21.1](#), [21.2](#)
 extending [21.10](#)
 filtering (with a phrase or description) [22.5](#)
 finding entries [21.4](#), [21.9](#)
 finding length [21.1](#), [21.9](#)
 mapping (with a phrase) [22.5](#)
 memory limits [21.10](#)
 reducing (with a phrase, to a value) [22.5](#)
 removing entries [21.5](#)
 repeating through [21.4](#)
 reversing [21.8](#)
 rotating [21.8](#)
 sorting [21.8](#)
 the multiple object list [17.20](#), [6.15](#)
 truncating [21.10](#)
 variations on use of lists [21.11](#)
 Z-machine limitations [21.10](#)
 lit / unlit *(thing)* *property* [3.15](#), [3.7](#)
 local variables *see* [temporary variables](#)
 locale description -- printing the locale description of something *activity* [18.26](#)
 locale paragraph -- printing a locale paragraph about something *activity* [18.28](#), [3.1](#)
 location of *(object)* phrase [3.25](#)
 location — *object* *global variable* [3.25](#), [4.12](#)
 locations *see* [rooms](#)
 lockable *(container/door)* *property* [3.13](#), [9.2](#)
 locked / unlocked *(container/door)* *property* [3.13](#), [9.2](#)
 locksmith *extension* [2.10](#), [2.11](#), [3.5](#), [9.2](#)
 logarithm to base *(number)* of *(real number)* phrase [15.6](#)
 Lollipop Guild *example* [18.34](#) ex [379](#), [6.8](#) ex [379](#)
 looking *action* [12.19](#), [3.1](#), [6.4](#)
 as part of going [6.9](#)
 loops
 counted [11.10](#)
 in phrases [11.9](#)
 interrupting [11.12](#)
 through objects [11.11](#)
 through values [11.10](#)
 losing the game [9.4](#)
 losing the story [18.38](#), [18.39](#), [11.6](#)
 Low Light *example* [18.27](#) ex [359](#), [3.1](#) ex [359](#)
 lower case -- *(text)* in lower case phrase [20.4](#)
 Lucy *example* [7.6](#) ex [92](#), [6.14](#) ex [92](#)
 Lugubrious Pete's Delicatessen *example* [21.11](#) ex [436](#), [7.16](#) ex [436](#)
 M. Melmoth's Duel *example* [11.15](#) ex [178](#), [2.1](#) ex [178](#)

M machines (implementing) [9.8](#), [10.7](#)
magic words [24.4](#), [6.13](#)
Magneto's Revenge *example* [12.18](#) *ex 215*, [10.10](#) *ex 215*
magnets (implementing) [10.7](#)
make no decision *phrase* [19.11](#)
male / female / neuter (*person*) *property* [3.17](#)
man *kind* [3.17](#), [4.1](#)
The Man of Steel *example* [12.11](#) *ex 203*, [7.15](#) *ex 203*
The Man of Steel Excuses Himself *example* [12.13](#) *ex 208*, [7.15](#) *ex 208*
map *see* [index](#) [map](#)
Map page of Index panel *user interface* [2.7](#), [3.4](#), [3.26](#), [25.18](#)
mapping (*direction*) *relation* [6.14](#)
mark (*external file*) **as not ready to read** *phrase* [23.15](#)
mark (*external file*) **as ready to read** *phrase* [23.15](#)
marked for listing / unmarked for listing (*thing*) *property* [18.25](#), [3.1](#)
Masochism Deli *example* [17.19](#) *ex 326*, [10.9](#) *ex 326*
mass nouns [3.18](#)
matched as, in describing action variables [12.10](#)
matched text — *snippet* *global variable* [18.33](#)
matching key of (*container/door*) *property* [3.13](#)
materials folder
 cover art [25.8](#)
 feelies [25.7](#)
 figures [23.4](#)
 introductory booklet / postcard [25.9](#)
 map [25.18](#), [13.2](#)
 pre-existing story file [25.15](#)
 project-specific extensions [27.3](#)
 released story file [25.6](#)
 sounds [23.8](#)
 source text [25.17](#)
 walkthrough [25.16](#)
 web-playable story [25.11](#), [26.5](#)
 web pages for the story [25.10](#), [26.5](#)
 templates [25.13](#), [25.14](#)
Mathematical view of relations *example* [13.16](#) *ex 243*, [1.4](#) *ex 243*
maths *see* [calculation](#)
Matreshka *example* [11.7](#) *ex 172*, [6.6](#) *ex 172*
Mattress King *example* [7.14](#) *ex 105*, [6.9](#) *ex 105*
maximum score — *number* *global variable* [9.2](#), [9.3](#), [11.4](#)
maximum text length *use option* [20.2](#)
maximum things understood at once *use option* [4.14](#)
"[may]" *say phrase* [14.7](#)
"[mayn't]" *say phrase* [14.7](#)
Mazaitis, Paul [1.2](#)
McGrew, Jesse [27.4](#), [1.2](#)
McKee, Kate [1.2](#)
meaning *relation* [14.9](#)
meaningful / meaningless (*verb*) *adjective* [14.9](#)
means, defining sentence verbs
 the verb to be (P) means (*relation*) *assertion* [13.10](#)
 the verb (V) means (*property*) *assertion* [15.12](#)
 the verb (V) means (*relation*) *assertion* [13.9](#)
Meet Market *example* [13.13](#) *ex 238*, [7.1](#) *ex 238*
member of (*description*) [22.2](#)
memory economy *use option* [2.14](#)
memory limits [2.14](#)
 for extending lists [21.10](#)
 for manipulating text [20.2](#)
memory of player *see* [knowledge](#) (*in story world*)
men *kind* *see* [man](#)
mentioned / unmentioned (*thing*) *property* [18.24](#), [18.26](#), [18.28](#), [3.1](#)

Merlin example [16.13 ex 277](#), [5.4 ex 277](#)
 metadata *see* [bibliographic data](#)
Meteoric I and II example [8.7 ex 119](#), [4.4 ex 119](#)
 Metric Units *extension* [15.13](#), [15.18](#), [15.20](#)
Midsummer Day example [3.5 ex 12](#), [1.1 ex 12](#)
 "[might]" *say phrase* [14.7](#)
 "[mightn't]" *say phrase* [14.7](#)
Mimicry example [7.15 ex 110](#), [7.6 ex 110](#)
Ming Vase example [7.10 ex 97](#), [10.4 ex 97](#)
Minimal Movement example [18.32 ex 372](#), [6.9 ex 372](#)
 (arithmetic value) minus (arithmetic value) *phrase* [15.5](#)
 minus infinity — real number *constant* [15.3](#)
 (number) minutes *phrase* [9.8](#)
 minutes part of (time) *phrase* [9.10](#)
Mirror, Mirror example [20.7 ex 417](#), [9.12 ex 417](#)
Misadventure example [17.10 ex 306](#), [6.9 ex 306](#)
 missing noun -- supplying a missing noun/second noun *activity* [17.1](#), [18.32](#)
 mistakes, in the player's command [17.21](#), [17.22](#), [5.3](#)
Mistress of Animals example [6.14 ex 77](#), [7.13 ex 77](#)
 Mistype *extension* [6.17](#)
 modal / non-modal (verb) *adjective* [14.7](#), [14.9](#)
Model Shop example [4.15 ex 57](#), [9.8 ex 57](#)
Modern Conveniences example [27.5 ex 449](#), [8.5 ex 449](#)
Money for Nothing example [15.19 ex 265](#), [9.4 ex 265](#)
 money (implementing) [9.4](#)
 monospaced text [5.9](#), [12.1](#)
 Montfort, Nick [1.2](#)
 Monty Python [11.7](#)
 more than, in descriptions [6.16](#)
 Moriarty, Brian [5.12](#)
Morning After example [7.5 ex 90](#), [6.8 ex 90](#)
 most, in descriptions [6.15](#)
 move (backdrop) to (region) *phrase* [8.8](#)
 move (object) backdrop to all (description of objects) *phrase* [8.8](#)
 move (object) to (object) *phrase* [8.7](#)
Mr. Burns' Repast example [20.8 ex 422](#), [2.3 ex 422](#)
MRE example [9.11 ex 143](#), [9.1 ex 143](#)
 multiple copies of things *see* [duplicates](#)
 multiple object list *phrase* [17.20](#), [6.15](#)
 (arithmetic value) multiplied by (arithmetic value) *phrase* [15.5](#)
Murder on the Orient Express example [13.16 ex 241](#), [7.11 ex 241](#)
 Musante, Mark [1.2](#)
 music *see* [sounds](#)
 "[must]" *say phrase* [14.7](#)
 "[mustn't]" *say phrase* [14.7](#)
Mutt's Adventure example [9.2 ex 136](#), [11.4 ex 136](#)

N

Nameless example [17.5 ex 297](#), [7.7 ex 297](#)
 names

- for things named generally [8.15](#)
- of characters [7.1](#)
- of phrases [22.3](#)
- of rules [19.2](#), [19.3](#), [19.7](#)
- printed name of (object) *property* [8.5](#)
- printing the name of a dark room *activity* [18.21](#)
- printing the name of something *activity* [18.10](#)
- printing the plural name of something *activity* [18.11](#)

NaN (not a number) *see* [nonexistent / existent \(real number\)](#)
Narrative Register example [14.3 ex 248](#), [2.1 ex 248](#)
 narrative viewpoint, of standard responses [14.1](#), [5.6](#)
 natural/-- logarithm of (real number) *phrase* [15.6](#)
 natural language, really understanding [2.16](#)
 nearest whole number -- (real number) to the nearest whole number *phrase* [15.3](#)

nearest -- *(arithmetic value) to the nearest (arithmetic value) phrase* [9.8](#), [15.5](#)
negative / positive (number) adjective [6.5](#)
Neighborhood Watch example [3.13 ex 24](#), [3.5 ex 24](#)
 Nelson, Toby [1.2](#)
neuter / male / female (person) property [3.17](#)
never / always, in assertions [4.3](#)
 Newcomb, Ron [1.2](#)
next *phrase* [11.12](#)
next step via (relation of values to values) from (object) to (object) phrase [13.11](#)
next thing held after (object) phrase [8.17](#)
Nickel and Dimed example [15.17 ex 262](#), [9.4 ex 262](#)
Night and Day example [10.7 ex 164](#), [3.9 ex 164](#)
The Night Before example [4.15 ex 59](#), [5.1 ex 59](#)
Night Sky example [9.12 ex 147](#), [3.1 ex 147](#)
Nine AM Appointment example [19.2 ex 388](#), [6.11 ex 388](#)
no *phrase* [11.16](#)
no, in descriptions [6.15](#)
No Place Like Home example [9.2 ex 137](#), [11.4 ex 137](#)
NO, querying yes/no [11.5](#)
No Relation example [7.14 ex 104](#), [8.1 ex 104](#)
NO, responding to a character [7.7](#)
nobody, testing that no such person exists [6.10](#)
Noisemaking example [12.9 ex 197](#), [7.3 ex 197](#)
Noisy Cricket example [16.12 ex 276](#), [10.2 ex 276](#)
non-empty / empty adjective
 for activities [18.5](#)
 for lists [21.1](#), [21.2](#)
 for relations [13.14](#)
 for rulebooks [19.8](#)
 for tables [16.9](#)
 for texts [6.5](#), [20.3](#)
 non-existence
 removing things from play [8.10](#)
 starting things out of play [3.25](#), [6.10](#), [8.10](#)
 testing that no such thing exists [6.10](#), [6.12](#)
non-modal / modal (verb) adjective [14.9](#)
non-recurring / recurring (scene) property [10.4](#), [10.7](#)
nondescript items -- listing nondescript items of something activity [18.25](#), [3.1](#)
none, in descriptions [6.15](#)
nonexistent / existent (real number) adjective [15.3](#)
North by Northwest example [18.33 ex 375](#), [2.2 ex 375](#)
Northstar example [20.8 ex 423](#), [7.14 ex 423](#)
not for release [2.9](#), [24.5](#), [25.6](#), [13.1](#)
not listed in, for removing rules from rulebooks [19.4](#)
notable locale objects -- choosing notable locale objects for something activity [18.27](#)
nothing
 as a value [6.12](#)
 do nothing *phrase* [11.1](#)
 in removing room connections [8.5](#)
 in understanding actions [17.3](#)
 testing that no such thing exists [6.10](#), [6.12](#)
NOTIFY ON/OFF [9.2](#), [11.4](#), [11.5](#)
noun part of (action) phrase [12.20](#)
noun — object global variable [7.10](#)
 nouns, in descriptions [6.2](#)
 nouns, reversed, in understanding actions [17.2](#)
now (a condition) phrase [2.2](#), [8.1](#), [8.11](#)
 changing either/or properties [8.4](#)
 changing global variables [8.1](#)
 changing list entries [21.9](#)
 changing table entries [16.4](#)
 changing temporary variables [11.15](#)

- changing the locations of backdrops [8.8](#)
- changing the locations of things [8.7](#), [8.10](#)
- changing the player's identity [8.9](#), [5.6](#)
- changing the player's location [8.9](#)
- changing value properties [8.5](#)
- for arbitrary conditions [8.11](#)
- now** -- *(rule) in (number) turn/turns from now* *phrase* [9.11](#)
- now** -- *(rule) in (time) from now* *phrase* [9.11](#)
- nowhere**
 - in removing room connections [3.3](#), [8.5](#)
 - placing backdrops [3.9](#), [8.8](#)
 - placing things [8.10](#)
 - rules for going nowhere [7.13](#)
- NPC** *see* [characters \(people\)](#)
- [number]** *grammar token* [17.6](#), [17.9](#)
- number of blank rows in/from** *(table name) phrase* [16.10](#)
- number of characters in** *(text) phrase* [20.3](#)
- number of** *(description of values) phrase* [6.1](#), [8.16](#)
- number of entries in/of** *(list of values) phrase* [21.1](#), [21.9](#)
- number of filled rows in/from** *(table name) phrase* [16.10](#)
- number of lines in** *(text) phrase* [20.3](#)
- number of moves from** *(object) to (object) phrase* [6.14](#)
- number of moves from** *(object) to (object) through (description of objects) phrase* [6.14](#)
- number of paragraphs in** *(text) phrase* [20.3](#)
- number of punctuated words in** *(text) phrase* [20.3](#)
- number of rows in/from** *(table name) phrase* [16.2](#)
- number of steps via** *(relation of values to values) from (object) to (object) phrase* [13.11](#)
- number of times** *(text) matches the regular expression (text) phrase* [20.6](#)
- number of times** *(text) matches the text (text) phrase* [20.5](#)
- number of unpunctuated words in** *(text) phrase* [20.3](#)
- number of words in** *(text) phrase* [20.3](#)
- Number Study example* [13.14](#) *ex 240*, [10.11](#) *ex 240*
- number understood** — *number global variable* [17.9](#)
- Numberless example* [11.8](#) *ex 174*, [2.1](#) *ex 174*
- numbers** [15.1](#), [15.2](#)
 - adjectives comparing magnitude [6.7](#), [6.8](#)
 - adjectives describing magnitude [6.6](#)
 - arithmetic [15.5](#)
 - calculation with *see* [calculation](#)
 - conversion to/from real numbers [15.3](#)
 - converting to times [9.10](#)
 - displaying [5.3](#), [5.4](#)
 - given by equations [15.18](#)
 - limits of numeric values [11.10](#), [15.2](#)
 - limits of numeric values of units [15.10](#), [15.14](#)
 - of things in descriptions [6.16](#)
 - printing a number of something** *activity* [18.12](#)
 - random numbers *see* [randomness](#)
 - real numbers [15.2](#)
 - conversion to/from numbers [15.3](#)
 - displaying [15.4](#)
 - exponential functions [15.6](#)
 - infinite and nonexistent [15.3](#)
 - trigonometric functions [15.7](#)
 - understanding [17.6](#), [17.9](#)
 - units of measure *see* [units of measure](#)
- Nyman, Jeff [1.2](#)
- O** **object loops** [11.11](#)
- Object Response Tests** *extension* [24.6](#)
- objects** *see* [things](#)
- odd / even** *(number) adjective* [4.6](#), [6.5](#)
- Odin example* [4.3](#) *ex 44*, [6.5](#) *ex 44*

Odyssey example [16.10 ex 274](#), [7.13 ex 274](#)
 of, for properties [4.9](#)
 off-stage / on-stage (thing) adjective [8.10](#)
Olfactory Settings example [14.6 ex 249](#), [2.1 ex 249](#)
 omit contents in listing phrase [18.10](#)
 omitting code, for release version [2.9](#), [24.5](#), [25.6](#), [13.1](#)
 on-stage / off-stage (thing) adjective [8.10](#)
 on relation verb *see* [support](#)
 on top of relation verb *see* [support](#)
 once, counting conditions [9.14](#)
 one-to-one (relation) adjective [13.14](#)
 one-to-various (relation) adjective [13.14](#)
 one-way connections [3.3](#)
One of Those Mornings example [17.7 ex 301](#), [5.4 ex 301](#)
One Short Plank example [7.14 ex 106](#), [3.5 ex 106](#)
 only, counting conditions [9.14](#), [9.15](#)
Only You... example [6.5 ex 74](#), [10.1 ex 74](#)
 opaque / transparent (container) property [3.6](#), [3.15](#), [9.2](#)
 open / closed (container/door) property [3.6](#), [3.13](#), [3.15](#), [9.2](#)
 openable / unopenable (container) property [3.6](#), [9.2](#)
 opposite of (direction) property [3.26](#)
 Options.txt [2.13](#)
 options, for phrases [11.14](#)
 or more / less [6.6](#), [6.7](#), [6.8](#)
Orange Cones example [8.8 ex 120](#), [3.9 ex 120](#)
 ordering *see* [precedence](#)
 ordering other characters [12.3](#), [7.14](#)
Originals example [18.31 ex 369](#), [9.12 ex 369](#)
 other people *see* [characters \(people\)](#)
 other side of (door) from (room) phrase [3.12](#)
 [other things] grammar token [17.4](#)
 otherwise (a phrase) phrase [11.8](#)
 otherwise if (a condition) phrase [11.8](#)
 otherwise unless (a condition) phrase [11.8](#)
Otranto example [13.7 ex 228](#), [10.6 ex 228](#)
 "[our]" say phrase [14.4](#)
 out of play [3.25](#), [6.10](#), [8.10](#)
 out of world [12.15](#), [12.21](#)
 outcome of the rulebook phrase [19.12](#)
 outline of source text [2.6](#)
 outside from [3.2](#)
Owen's Law example [11.16 ex 179](#), [6.9 ex 179](#)
Oyster Wide Shut example [21.3 ex 425](#), [6.7 ex 425](#)
P *Paddington example* [12.9 ex 195](#), [10.4 ex 195](#)
Pages example [17.9 ex 303](#), [9.6 ex 303](#)
 paint (implementing) [9.7](#)
Palette example [17.11 ex 308](#), [9.7 ex 308](#)
Panache example [10.8 ex 166](#), [11.4 ex 166](#)
 paragraph breaks [5.6](#), [5.8](#)
 counting paragraphs in text [20.3](#)
 for confusing the normal display of objects [18.24](#)
 for fooling persuasion refusal [12.4](#)
 with vertical stroke [2.3](#)
 paragraph number (number) in (text) phrase [20.3](#)
 paragraph -- writing a paragraph about activity [18.24](#), [3.1](#)
 parameters
 of phrases [11.3](#)
 kind values [22.8](#)
 of rulebooks [19.9](#)
 parentheses *see* [punctuation: brackets](#)
 parser's choice -- clarifying the parser's choice of something activity [18.30](#)
 parser error messages [18.35](#), [6.17](#)

parsing *see* [understanding](#)
 part of source text [2.5](#)
 part of relation verb *see* [incorporation](#)
 part of, for units with multiple components [15.15](#)
 particular possession — *thing* global variable [3.24](#), [18.9](#)
 past / present / future tense [14.1](#)
 past tense, of conditions in rules [9.12](#), [9.13](#)
 pathfinding [6.14](#)
 arbitrary relations [13.11](#)
Patient Zero example [19.8 ex 402](#), [7.13 ex 402](#)
 pattern matching
 phrase definitions [11.3](#)
Peeled example [18.29 ex 363](#), [3.7 ex 363](#)
Peers example [17.15 ex 315](#), [7.1 ex 315](#)
 people *see* [person; characters \(people\)](#)
 perfect tense, of conditions in rules [9.13](#)
 period *see* [punctuation: full stop](#)
Persephone example [5.6 ex 65](#), [6.7 ex 65](#)
 person kind [3.17](#)
 carrying capacity [3.19](#)
 characters *see* [characters \(people\)](#)
 person asked — *object* global variable [12.3](#)
 person reaching — *object* global variable [12.18](#)
 persuasion *rulebook* [12.4](#)
 persuasion succeeds / fails (*persuasion rules*) *outcome phrase* [12.4](#)
Peugeot example [3.16 ex 27](#), [8.1 ex 27](#)
 phrases [11.1](#)
 as values [22.3](#)
 default values [22.4](#)
 deciding a condition [11.5](#), [11.16](#)
 deciding a value [11.17](#)
 defining [11.2](#)
 defining in Inform 6 [27.15](#), [27.16](#), [27.17](#)
 interrupting [11.13](#)
 named phrases [22.3](#)
 options for small variations [11.14](#)
 phrase parameters [11.3](#)
 kind values [22.8](#)
 precedence of definitions [22.9](#), [22.10](#)
 temporary variables in phrases [11.15](#)
 Phrases page of Index panel *user interface* [2.2](#), [11.2](#)
 pi — *real number* constant [15.2](#)
 pictures *see* [figures](#)
Pine 1 example [10.2 ex 154](#), [7.3 ex 154](#)
Pine 2 example [10.5 ex 161](#), [7.12 ex 161](#)
Pine 3 example [10.7 ex 165](#), [4.5 ex 165](#)
Pine 4 example [10.8 ex 167](#), [4.5 ex 167](#)
Pink or Blue example [27.15 ex 451](#), [5.2 ex 451](#)
Pizza Prince example [18.34 ex 378](#), [10.3 ex 378](#)
 place (object) in scope *phrase* [18.29](#)
 place the/-- contents of (object) in scope *phrase* [18.29](#)
 places *see* [rooms](#)
 play (*sound name*) *phrase* [23.8](#)
 player's command — *snippet* global variable [18.33](#)
 player's holdall kind [3.21](#)
 player
 as opposed to the actor [12.3](#), [12.11](#)
 carrying capacity [3.19](#)
 changing the identity of the player [8.9](#), [5.6](#)
 customizing name or traits [5.2](#)
 initial location [3.1](#)
 memory *see* [knowledge \(in story world\)](#)

- moving the player [8.7](#), [8.9](#)
- plot structure *see* [story structure](#)
- Plotkin, Andrew [1.2](#), [25.9](#)
- plural-named / singular-named (*thing*) *property* [3.18](#)
- plural of
 - in player command [17.8](#)
 - in source text [4.4](#)
- plurals
 - defining [4.4](#), [2.2](#)
 - displaying [5.4](#)
 - of units of measure [15.9](#)
 - plural convention for adaptive pronouns [14.4](#)
 - printing the plural name of something *activity* [18.11](#)
 - understanding (in commands) [17.8](#)
- (*arithmetic value*) plus (*arithmetic value*) *phrase* [15.5](#)
- plus infinity — *real number constant* [15.3](#)
- A point for never saving the game example* [12.15](#) *ex* [211](#), [11.2](#) *ex* [211](#)
- points *see* [scoring](#)
- Polarity example* [7.13](#) *ex* [103](#), [6.9](#) *ex* [103](#)
- Police State example* [12.4](#) *ex* [187](#), [7.3](#) *ex* [187](#)
- Politics as Usual example* [8.3](#) *ex* [114](#), [12.2](#) *ex* [114](#)
- Port Royal 1 example* [3.2](#) *ex* [5](#), [3.2](#) *ex* [5](#)
- Port Royal 2 example* [3.3](#) *ex* [8](#), [3.2](#) *ex* [8](#)
- Port Royal 3 example* [3.4](#) *ex* [10](#), [3.2](#) *ex* [10](#)
- Port Royal 4 example* [16.6](#) *ex* [272](#), [3.6](#) *ex* [272](#)
- Port Royal 5 example* [25.23](#) *ex* [447](#), [13.2](#) *ex* [447](#)
- portable / fixed in place (*thing*) *property* [3.6](#), [3.7](#), [8.7](#)
- positive / negative (*number*) *adjective* [6.5](#)
- possession *relation* [8.13](#), [13.4](#)
- possessions [3.20](#)
 - a hold-all for the player's possessions [3.21](#)
 - deciding the concealed possessions of something *activity* [3.24](#), [18.9](#)
- "[possessive]" *say phrase* [14.6](#)
- possible / likely / unlikely, it is possible / likely / unlikely (*does the player mean*) *outcome phrase* [17.19](#)
- Pot of Petunias example* [17.18](#) *ex* [325](#), [2.2](#) *ex* [325](#)
- powers, of numbers [15.6](#)
- The Prague Job example* [10.6](#) *ex* [162](#), [4.2](#) *ex* [162](#)
- preamble optional, when defining units [15.15](#)
- precedence
 - of commands understood [17.22](#)
 - of displayed items [18.26](#), [18.27](#), [3.1](#)
 - of phrase definitions [22.9](#), [22.10](#)
 - of rules [19.2](#), [19.4](#), [19.6](#), [19.16](#)
- prepositions, for sentence verbs [13.10](#)
- presence of [7.12](#), [18.29](#)
- present / past / future tense [14.1](#)
- Princess and the Pea example* [11.7](#) *ex* [173](#), [8.4](#) *ex* [173](#)
- printed name of (*object*) *property* [8.5](#), [8.6](#), [18.10](#)
- printed plural name of (*object*) *property* [18.11](#)
- printing a locale paragraph about *activity* [18.28](#), [3.1](#)
- printing a number of something *activity* [18.12](#)
- printing a parser error *activity* [18.35](#)
- printing a refusal to act in the dark *activity* [18.18](#)
- printing an abbreviated room description [8.7](#), [8.9](#)
- printing inventory details of something *activity* [18.17](#)
- printing room description details of something *activity* [18.16](#), [3.1](#)
- printing the announcement of darkness *activity* [18.19](#)
- printing the announcement of light *activity* [18.20](#)
- printing the banner text *activity* [18.37](#)
- printing the description of a dark room *activity* [18.22](#)
- printing the locale description of something *activity* [18.26](#)

printing the name of a dark room *activity* [18.21](#)
 printing the name of something *activity* [18.10](#)
 printing the player's obituary *activity* [18.38](#)
 printing the plural name of something *activity* [18.11](#)
 priority *see* [precedence](#)
Priority Lab example [18.26](#) [ex 358](#), [3.1](#) [ex 358](#)
Prisoner's Dilemma example [3.26](#) [ex 40](#), [3.2](#) [ex 40](#)
 private / public, in release elements [25.16](#), [25.17](#)
 privately-named / publicly-named (*object*) *adjective* [17.17](#), [2.2](#)
 problem messages
 abject failures of Inform [2.15](#)
 for source text [2.4](#)
The Problem of Edith example [13.11](#) [ex 235](#), [7.11](#) [ex 235](#)
 (*name of kind*) produced by (*rule producing values*) *phrase* [19.13](#)
 (*name of kind*) produced by (*values based rule producing values*) for (*value*) *phrase* [19.13](#)
Prolegomena example [18.12](#) [ex 343](#), [2.1](#) [ex 343](#)
 prompt, changing the prompt [8.2](#), [9.7](#)
PRONOUNS [11.5](#)
 pronouns [3.17](#)
 adaptive contractions in displayed text [14.8](#)
 adaptive pronouns in displayed text [14.2](#), [14.4](#), [14.5](#), [14.6](#)
 displaying with **PRONOUNS** [11.5](#)
 HIM/HER, in player commands [17.18](#), [2.2](#)
 IT, in player commands [17.18](#), [2.2](#)
 it, in source text [11.16](#)
 setting pronouns [17.18](#), [2.2](#)
 something, anything... [6.2](#)
 THEM, in player commands [17.18](#), [2.2](#)
 proper-named / improper-named (*thing*) *property* [3.18](#)
 properties [3.6](#), [3.7](#), [4.6](#)
 catalogue
 carrying capacity of (*container/supporter/person*) [3.19](#), [10.5](#)
 closed (*container/door*) [3.6](#), [3.13](#), [3.15](#), [9.2](#)
 dark (*room*) [3.15](#), [3.7](#)
 described (*thing*) [3.24](#)
 description of (*room*) [3.10](#), [3.11](#)
 description of (*scene*) [10.2](#)
 description of (*thing*) [3.11](#)
 edible (*thing*) [3.22](#), [9.1](#)
 enterable (*container*) [3.6](#), [9.2](#)
 female (*person*) [3.17](#)
 fixed in place (*thing*) [3.6](#), [3.7](#), [8.7](#)
 handled (*thing*) [3.11](#)
 improper-named (*thing*) [3.18](#)
 indefinite article of (*object*) [3.18](#)
 inedible (*thing*) [3.22](#), [9.1](#)
 initial appearance of (*thing*) [3.11](#), [8.5](#)
 lighted (*room*) [3.15](#), [3.7](#)
 lit (*thing*) [3.15](#), [3.7](#)
 lockable (*container/door*) [3.13](#), [9.2](#)
 locked (*container/door*) [3.13](#), [9.2](#)
 male (*person*) [3.17](#)
 marked for listing (*thing*) [18.25](#), [3.1](#)
 matching key of (*container/door*) [3.13](#)
 mentioned (*thing*) [18.24](#), [18.26](#), [18.28](#), [3.1](#)
 neuter (*person*) [3.17](#)
 non-recurring (*scene*) [10.4](#), [10.7](#)
 opaque (*container*) [3.6](#), [3.15](#), [9.2](#)
 open (*container/door*) [3.6](#), [3.13](#), [3.15](#), [9.2](#)
 openable (*container*) [3.6](#), [9.2](#)
 opposite of (*direction*) [3.26](#)
 plural-named (*thing*) [3.18](#)

- portable (*thing*) [3.6](#), [3.7](#), [8.7](#)
- printed name of (*object*) [8.5](#), [8.6](#), [18.10](#)
- printed plural name of (*object*) [18.11](#)
- properly-named (*thing*) [3.18](#)
- pushable between rooms (*thing*) [3.16](#)
- recurring (*scene*) [10.4](#), [10.7](#)
- scenery (*thing*) [3.8](#)
- singular-named (*thing*) [3.18](#)
- specification of (*thing*) [27.10](#)
- switched off (*device*) [3.14](#), [9.8](#), [10.7](#)
- switched on (*device*) [3.14](#), [9.8](#), [10.7](#)
- transparent (*container*) [3.6](#), [3.15](#), [9.2](#)
- undescribed (*thing*) [3.24](#)
- unlit (*thing*) [3.15](#), [3.7](#)
- unlocked (*container/door*) [3.13](#), [9.2](#)
- unmarked for listing (*thing*) [18.25](#), [3.1](#)
- unmentioned (*thing*) [18.24](#), [18.26](#), [18.28](#), [3.1](#)
- unopenable (*container*) [3.6](#), [9.2](#)
- wearable (*thing*) [3.20](#), [9.3](#)
- changing multiple things with **now** [8.11](#)
- defining properties in Inform 6 [27.21](#), [27.22](#)
- defining properties with a table [16.16](#), [16.17](#)
- defining sentence verbs for properties [15.12](#)
- dependence on kind [3.7](#)
- either/or properties [3.6](#), [4.6](#)
 - changing [8.4](#)
 - defining [4.7](#)
 - used as adjectives [6.3](#)
- holding a new kind of value [4.9](#), [4.10](#)
- holding text [3.10](#), [4.6](#)
- implications [27.13](#)
- numeric comparatives, adjectives for [6.7](#), [15.12](#)
- numeric magnitudes, adjectives for [6.6](#)
- numeric superlatives, adjectives for [6.8](#), [15.12](#)
- of scenes [10.2](#), [10.4](#)
- shorthand name, omitting **of the** [8.6](#)
- understanding things by properties [17.15](#)
- value properties [3.10](#), [4.6](#)
 - changing [8.5](#)
 - defining [4.8](#), [4.9](#), [4.10](#)
 - used as adjectives [6.3](#)
- proportional-spaced text [5.9](#), [12.1](#)
- Proposal* *example* [11.5](#) *ex 171*, [7.7](#) *ex 171*
- propriety [7.15](#)
- Provenance Unknown* *example* [7.14](#) *ex 107*, [6.9](#) *ex 107*
- provides**
 - provision relation *see* [provision](#)
- provision *relation* [13.8](#)
- public / private**, in release elements [25.16](#), [25.17](#)
- publication data *see* [bibliographic data](#)
- publicly-named / privately-named** (*object*) *adjective* [17.17](#), [2.2](#)
- Puff of Orange Smoke* *example* [12.10](#) *ex 201*, [7.5](#) *ex 201*
- Puncak Jaya* *example* [17.16](#) *ex 320*, [5.5](#) *ex 320*
- punctuated word number** (*number*) *in* (*text*) *phrase* [20.3](#)
- punctuation [2.3](#)
 - apostrophe, meaning quotation mark [2.3](#), [5.2](#), [25.2](#)
 - brackets
 - for calling values [8.15](#)
 - for defining mistakes [17.21](#)
 - for implicit multiplication [15.18](#)
 - for phrase parameters [11.3](#)
 - in parser clarifications [18.30](#)

- in response names [14.10](#)
- colon [6.4](#), [11.2](#), [11.7](#), [11.16](#), [19.3](#)
- comma
 - comma replacing colon [7.2](#)
 - displaying serial comma [2.12](#), [5.5](#)
 - in list constants, requiring spaces [21.2](#)
 - separating phrase options [11.14](#)
- curly braces
 - for constant lists [21.2](#)
 - for including Inform 7 expressions in Inform 6 [27.15](#)
- double-dash
 - optional words in phrase definitions [11.3](#)
 - optional words in Understand grammar [17.12](#)
- exclamation mark, ending sentences [2.3](#), [5.2](#)
- full stop
 - before text substitutions [5.6](#)
 - ending sentences with line breaks [2.3](#), [5.2](#), [5.8](#)
- in counting words [20.3](#), [20.8](#)
- period *see* [punctuation: full stop](#)
- question mark, ending sentences [2.3](#), [5.2](#)
- quotation marks
 - defining texts [2.3](#), [3.1](#), [5.1](#), [5.2](#)
 - not usable in unit notations [15.14](#)
- removing from player's command [18.33](#), [6.13](#), [7.1](#), [7.7](#)
- slash
 - separating synonymous words in grammar [17.4](#), [17.12](#), [17.13](#)
 - separating synonymous words in phrase definitions [11.3](#)
 - separating synonymous words in topics [16.13](#)
- square brackets
 - comments [2.3](#)
 - text substitutions [2.3](#), [5.1](#), [5.2](#)
- vertical stroke, for paragraph break [2.3](#)
- Punctuation Removal *extension* [6.13](#), [7.1](#), [7.7](#)
- PURLOIN *testing command* [2.8](#), [24.4](#)
- pushable between rooms (*thing*) *property* [3.16](#)

pushing things [3.16](#)

rules for pushing things [7.14](#), [6.9](#)

"[put]" *say phrase* [14.3](#)

putting it on *action* [6.8](#)

Q

quantifiers

existential [6.2](#)

universal [6.15](#)

The Queen of Sheba *example* [16.13](#) *ex 279*, [7.11](#) *ex 279*

Query *example* [17.21](#) *ex 329*, [11.3](#) *ex 329*

querying the player yes/no [11.5](#)

question mark *see* [punctuation: question mark, ending sentences](#)

Questionable Revolutions *example* [16.13](#) *ex 278*, [7.11](#) *ex 278*

QUIT [11.6](#)

Quiz Show *example* [17.17](#) *ex 323*, [2.2](#) *ex 323*

quotation marks *see* [punctuation: quotation marks](#)

quotations, displaying in boxes [5.12](#)

R

Radio Daze *example* [5.7](#) *ex 66*, [9.9](#) *ex 66*

radios (implementing) [9.9](#)

RANDOM *testing command* [24.3](#)

a/-- random (*description of values*) *phrase* [8.19](#)

a random (*name of kind*) between (*arithmetic value*) and (*arithmetic value*) *phrase* [8.18](#)

a random (*name of kind*) between (*enumerated value*) and (*enumerated value*) *phrase* [8.18](#)

a random (*name of kind*) from (*arithmetic value*) to (*arithmetic value*) *phrase* [8.18](#)

a random (*name of kind*) from (*enumerated value*) to (*enumerated value*) *phrase* [8.18](#)

randomness

choosing a random table row [16.5](#)

- combat with random results [7.5](#)
- dice and cards [9.5](#)
- enumerated values [8.18](#)
- random numbers [8.18](#)
- random things [8.19](#)
- randomizing the initial state of the world [11.1](#)
- seeding the generator [8.18](#), [24.3](#)
- sorting lists in random order [21.8](#)
- sorting tables in random order [16.11](#)
- text variations [5.7](#), [2.1](#)
- value kinds [8.18](#)
- rankings *see* [scoring](#)
- Raszewski, L. Ross [1.2](#)
- rather than**, in defining adjectives [6.4](#)
- rBGH example* [15.8 ex 254](#), [5.1 ex 254](#)
- "['re]" *say phrase* [14.8](#)
- reaching inside / outside** *rulebook* [12.16](#), [12.18](#)
- read (external file) into (table name)** *phrase* [23.13](#)
- reading *see* [examining](#)
- reading a command** *activity* [18.33](#)
- Real Adventurers Need No Help example* [4.12 ex 54](#), [11.3 ex 54](#)
- real numbers [15.2](#)
 - arithmetic [15.5](#)
 - conversion to/from numbers [15.3](#)
 - displaying [15.4](#)
 - exponential functions [15.6](#)
 - given by equations [15.18](#)
 - infinite and nonexistent [15.3](#)
 - limits of numeric values [15.2](#)
 - limits of numeric values of units [15.10](#), [15.14](#)
 - trigonometric functions [15.7](#)
 - units of measure *see* [units of measure](#)
- real square root of (arithmetic value)** *phrase* [15.5](#)
- Real Time Delays *extension* [12.4](#)
- reason the action failed** — *rule global variable* [12.5](#), [12.12](#), [19.14](#)
- reciprocal of (real number)** *phrase* [15.6](#)
- reciprocal relations [13.6](#)
- recording devices (implementing) [9.12](#)
- recurring / non-recurring (scene)** *property* [10.4](#), [10.7](#)
- Red Cross example* [12.7 ex 191](#), [7.5 ex 191](#)
- (phrase (value, value) -> value)* **reduction of (list of values)** *phrase* [22.5](#)
- referring to, in understand as** [17.15](#)
- Reflections example* [18.24 ex 352](#), [3.7 ex 352](#)
- refusal to act -- printing a refusal to act in the dark** *activity* [18.18](#)
- "[regarding (number)]" *say phrase* [14.5](#)
- "[regarding (thing)]" *say phrase* [14.5](#), [14.6](#)
- regional-containment *relation* [6.11](#)
- regions *kind* [3.4](#), [4.1](#)
 - backdrops in regions [3.9](#)
 - moving a backdrop to a new region [8.8](#)
 - regional containment [6.11](#)
 - rules applying to regions [7.11](#)
 - things in regions [6.11](#)
- regular expressions [18.33](#), [20.6](#), [20.8](#), [20.9](#)
 - syntax reference [20.9](#)
- reject the player's command** *phrase* [18.33](#)
- related by**, for understanding things by relations [17.16](#)
- (relation)* **relates...** *assertion* [13.5](#), [13.12](#), [13.13](#)
- relates**, for arbitrary relation conditions [13.14](#), [13.15](#)
- relates**, for finding related values [13.13](#)
- relations [4.6](#), [13.3](#)
 - creating things in relation to [4.15](#), [4.16](#)

- defining [13.5](#), [13.13](#)
- defining prepositions for relations [13.10](#)
- defining sentence verbs for relations [13.9](#)
- difference from sentence verbs [13.3](#)
- distance through [13.11](#)
- empty / non-empty *adjective* [13.14](#)
- expressing conditions [13.12](#)
- group relations [13.7](#)
- guidelines for use [13.16](#)
- indirect [13.11](#)
- involving values [13.13](#)
- kinds of relation [13.14](#)
- paths through [13.11](#)
- predefined
 - adjacency [6.14](#), [13.8](#)
 - carrying [3.20](#), [8.7](#), [8.13](#), [13.4](#); *see also* [carrying](#)
 - concealment [3.24](#), [13.4](#), [18.9](#); *see also* [concealment](#)
 - containment [6.11](#), [8.13](#), [13.4](#); *see also* [containment](#); [containers](#)
 - enclosure [3.25](#)
 - equality [13.8](#)
 - incorporation [3.23](#), [4.15](#), [8.13](#), [13.4](#); *see also* [incorporation](#); [components](#)
 - mapping (*direction*) [6.14](#)
 - meaning [14.9](#)
 - possession [8.13](#), [13.4](#)
 - regional-containment [6.11](#)
 - support [8.13](#), [13.4](#); *see also* [supporters](#)
 - touchability [6.13](#); *see also* [touchability](#)
 - visibility [6.13](#); *see also* [visibility](#)
 - wearing [3.20](#), [8.7](#), [13.4](#)
- reciprocal relations [13.6](#)
- relationship to sentence verbs [13.8](#)
- temporary relations [13.15](#)
- understanding things by relations [17.16](#)
- used as values [13.14](#)
- used in sentences [13.1](#)
- RELATIONS *testing command* [13.7](#), [24.3](#)
- release along with... *assertion*
 - a file of (*description*) called (*filename*) [25.7](#)
 - a solution [25.16](#)
 - a website [25.10](#), [26.5](#)
 - an existing story file [25.15](#)
 - an interpreter [25.11](#), [25.12](#), [26.5](#)
 - an introductory booklet [25.9](#)
 - an introductory postcard [25.9](#)
 - cover art [25.8](#)
 - public / private elements [25.16](#), [25.17](#)
 - the library card [25.17](#)
 - the source text [25.17](#)
- Release button *user interface* [2.7](#), [25.6](#)
- release number *bibliographic data* [18.37](#), [25.2](#)
- release version (of the story) [2.9](#), [25.6](#), [13.1](#)
- Relevant Relations *example* [14.9](#) ex [251](#), [2.1](#) ex [251](#)
- remainder after dividing (*arithmetic value*) by (*arithmetic value*) *phrase* [15.5](#)
- REMEMBER [5.4](#)
- Removal *example* [12.10](#) ex [198](#), [6.8](#) ex [198](#)
- remove entries (*number*) to (*number*) from (*list of values*) *phrase* [21.5](#)
- remove entry (*number*) from (*list of values*) *phrase* [21.5](#)
- remove (*list of values*) from (*list of values*) *phrase* [21.5](#)
- remove (*object*) from play *phrase* [8.10](#)
- remove (*value*) from (*list of values*) *phrase* [21.5](#)
- repeat through (*table name*) *phrase* [16.6](#)
- repeat through (*table name*) in reverse order *phrase* [16.6](#)

repeat through *(table name)* in reverse *(table column)* order *phrase* [16.6](#)
 repeat through *(table name)* in *(table column)* order *phrase* [16.6](#)
 repeat with *(a name not so far used)* running from *(arithmetic value)* to *(arithmetic value)* *phrase* [11.10](#)
 repeat with *(a name not so far used)* running from *(enumerated value)* to *(enumerated value)* *phrase* [11.10](#)
 repeat with *(a name not so far used)* running through *(description of values)* *phrase* [11.11](#)
 repeat with *(a name not so far used)* running through *(list of values)* *phrase* [21.4](#)
 replace character number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace line number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace paragraph number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace punctuated word number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace *(snippet)* with *(text)* *phrase* [18.33](#)
 replace the punctuated word *(text)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace the regular expression *(text)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace the text *(text)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace the word *(text)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace unpunctuated word number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 replace word number *(number)* in *(text)* with *(text)* *phrase* [20.8](#)
 Replanting *example* [3.8](#) ex [16](#), [6.8](#) ex [16](#)
 Replay button *user interface* [1.5](#), [1.7](#)
 report *(action)* *rulebook* [12.2](#), [12.9](#), [6.3](#)
 for other characters [12.13](#)
 guidelines for use [12.21](#)
 requiring, in defining actions [12.7](#), [12.17](#), [17.1](#), [6.2](#)
 resource IDs [23.10](#)
 RESPONSES *testing command* [14.12](#), [24.3](#)
 responses (library messages) [14.10](#), [6.16](#)
 changing [14.11](#)
 issuing the response text of something *activity* [18.15](#)
 listing during development [14.12](#), [24.3](#)
 Responsive *example* [14.11](#) ex [252](#), [2.1](#) ex [252](#)
 RESTART [11.6](#)
 RESTORE [11.6](#)
 Results panel *user interface* [1.6](#)
 resume the story *phrase* [9.4](#)
 Revenge of the Fussy Table *example* [6.15](#) ex [82](#), [7.3](#) ex [82](#)
 reverse *(list of values)* *phrase* [21.8](#)
 reverse order, for table rows [16.6](#), [16.11](#)
 reversed nouns, in understanding actions [17.2](#)
 Rideable Vehicles *extension* [3.16](#), [8.3](#)
 right hand status line — *text global variable* [8.3](#), [18.23](#), [12.2](#)
 Rip Van Winkle *example* [18.25](#) ex [355](#), [3.1](#) ex [355](#)
 Roberts, Mike [1.2](#)
 Robo 1 *example* [21.5](#) ex [426](#), [7.12](#) ex [426](#)
 Robo 2 *example* [21.9](#) ex [429](#), [7.12](#) ex [429](#)
 Rock Garden *example* [18.29](#) ex [366](#), [3.4](#) ex [366](#)
 Rocket Man *example* [20.4](#) ex [413](#), [2.1](#) ex [413](#)
 room-describing action — *action name action variable* [6.9](#)
 room-or-door *(direction)* from/of *(room)* *phrase* [6.14](#)
 room description details -- printing room description details of something *activity* [18.16](#), [3.1](#)
 room *(direction)* from/of *(room)* *phrase* [6.14](#)
 room gone from/to — *object action variable* [19.16](#)
 rooms *kind* [3.5](#), [4.1](#), [3.2](#)
 adjacent [6.14](#), [13.8](#)
 changing connections between rooms [8.5](#)
 connections between rooms [3.2](#), [3.2](#)
 creating [3.1](#)
 descriptions [2.12](#), [3.1](#), [6.4](#)
 moving without printing one [8.7](#), [8.9](#)
 divided into smaller areas [3.3](#)
 grouping into regions [3.4](#), [6.11](#)
 listening to (with no object) [7.7](#), [3.8](#)
 moving rooms [8.2](#)

- name of a dark room [18.21](#)
- one-way connections between rooms [3.3](#)
- printing the room contents [18.26](#), [18.27](#), [18.28](#), [3.1](#)
- rules applying to rooms [7.11](#), [7.13](#)
- smelling (with no object) [7.7](#)
- things in more than one room [3.9](#), [3.4](#), [3.9](#)
- roots, of numbers [15.5](#)
- ropes (implementing) [10.6](#)
- Rossing, Eric [1.2](#)
- rotate** (*list of values*) *phrase* [21.8](#)
- rotate** (*list of values*) **backwards** *phrase* [21.8](#)
- route-finding, fast/slow [6.14](#), [13.11](#)
- rows, in tables *see* [tables](#)
- rows -- choose a/the/-- row** (*number*) **in/from** (*table name*) *phrase* [16.5](#)
- rows -- choose a/the/-- row with** (*table column*) **of** (*value*) **in/from** (*table name*) *phrase* [16.5](#)
- rows -- number of rows in/from** (*table name*) *phrase* [16.2](#)
- Rubies example* [23.13](#) *ex 442*, [11.4](#) *ex 442*
- rule fails** *phrase* [19.11](#)
- rule succeeds** *phrase* [19.11](#)
- rule succeeds with result** (*value*) *phrase* [19.13](#)
- rulebooks [19.1](#), [19.2](#)
 - as a kind of rule [19.8](#)
 - basis of rulebook [19.9](#)
 - defining [19.8](#)
 - empty / non-empty** *adjective* [19.8](#)
 - for actions [12.2](#), [6.3](#)
 - for activities [18.3](#), [18.5](#)
 - for new actions [12.9](#), [6.2](#)
 - invoking a rulebook [19.8](#)
 - and abiding by its outcome [19.14](#)
 - multiple action processing rules [17.20](#)
 - order of rules [19.2](#), [19.4](#), [19.6](#), [19.16](#)
 - outcomes [19.11](#), [19.12](#), [19.13](#)
 - rulebook parameters [19.9](#)
 - stopping execution [19.11](#)
 - variables for rulebooks [19.10](#)
- rules [7.2](#)
 - abbreviating actions with **doing it** [7.17](#)
 - access to things referred to in actions [12.16](#), [12.18](#)
 - actions referred to in conditions [9.12](#)
 - after rules [7.5](#), [12.2](#), [12.15](#), [18.3](#), [6.3](#)
 - applying only when characters are present [7.12](#)
 - applying only while performing activities [18.4](#)
 - applying to all but some actions [7.9](#)
 - applying to arbitrary things [7.10](#)
 - applying to consecutively repeated actions [7.17](#)
 - applying to multiple actions [7.8](#)
 - applying to repeated actions [7.16](#)
 - applying to rooms and regions [7.11](#), [7.13](#)
 - arbitrary conditions for rules, using **when** [7.12](#), [9.5](#)
 - arbitrary conditions for rules, using **while** [18.4](#)
 - before rules [7.3](#), [12.2](#), [12.15](#), [18.3](#), [6.3](#)
 - changing conditions of rules [19.5](#)
 - counting repetitions for conditions [9.14](#)
 - counting turns for conditions [9.15](#)
 - creating new kinds of action to reduce redundancy [7.15](#)
 - defining [2.2](#), [19.3](#), [19.7](#)
 - defining in Inform 6 [27.20](#)
 - failure of rules [19.11](#)
 - for actions found in rulebooks [12.2](#), [6.3](#)
 - for activities [18.3](#), [18.5](#)
 - for any actor (including the player) [12.14](#)

- for **exiting** [7.14](#)
- for **going**, special cases [7.13](#), [7.14](#)
- for new actions [12.9](#), [6.2](#)
- for new actions for other characters [12.11](#)
- for other characters [12.4](#), [12.5](#), [12.11](#), [12.12](#), [12.13](#)
- for **pushing** objects to other rooms [7.14](#)
- for scenes [10.2](#), [4.4](#)
- for senses [7.7](#)
- for speech [7.6](#)
- for visibility [12.19](#)
- guidelines for action rules [12.21](#)
- in rulebooks [19.2](#)
- instead rules [7.2](#), [7.3](#), [12.2](#), [12.4](#), [12.15](#), [6.3](#)
- invoking a rule [19.8](#)
 - and abiding by its outcome [19.14](#)
- naming [19.3](#), [19.7](#)
- order in rulebooks [19.2](#), [19.4](#), [19.6](#), [19.16](#)
- past tense of conditions [9.12](#), [9.13](#)
- perfect tense of conditions [9.13](#)
- persuasion rules for instructing characters [12.4](#)
- preamble [19.7](#)
- reaching inside / outside [12.16](#), [12.18](#)
- removing [19.4](#), [19.5](#), [6.3](#)
- replacing [19.4](#), [19.5](#), [6.3](#)
- run at beginning of scene [10.2](#), [4.4](#)
- run at beginning of story [9.1](#), [19.15](#), [11.1](#)
- run at end of scene [10.2](#), [4.4](#)
- run at end of story [9.4](#), [19.15](#), [11.6](#)
- run every turn [9.5](#), [7.2](#)
- success of rules [19.11](#), [19.13](#)
- unsuccessful attempts by other characters [12.5](#)

RULES *testing command* [24.3](#), [6.2](#)

Rules of Attraction example [18.16 ex 346](#), [10.7 ex 346](#)

Rules page of Index panel *user interface* [19.6](#), [19.16](#)

running from, looping through values [11.10](#)

running through, looping through objects [11.11](#)

S

sack object *see* [hold-all](#)

Safari Guide example [17.10 ex 307](#), [6.9 ex 307](#)

Safety example [17.6 ex 298](#), [9.2 ex 298](#)

Saint Eligius example [19.4 ex 393](#), [6.9 ex 393](#)

same (property) as [6.7](#)

Sand example [7.3 ex 87](#), [6.8 ex 87](#)

Savannah example [15.19 ex 267](#), [10.2 ex 267](#)

say "[']" *phrase* [5.2](#)

say "['re]" *phrase* [14.8](#)

say "['ve]" *phrase* [14.8](#)

say "[a list of (description of objects)]" *phrase* [5.5](#)

say "[A list of (description of objects)]" *phrase* [5.5](#)

say "[a list of (description of objects) including contents]" *phrase* [5.5](#)

say "[A (object)]" *phrase* [5.3](#)

say "[a (object)]" *phrase* [5.3](#)

say "[adapt (verb)]" *phrase* [14.9](#)

say "[adapt (verb) from (narrative viewpoint)]" *phrase* [14.9](#)

say "[adapt (verb) in (grammatical tense)]" *phrase* [14.9](#)

say "[adapt (verb) in (grammatical tense) from (narrative viewpoint)]" *phrase* [14.9](#)

say "[an (object)]" *phrase* [5.3](#)

say "[An (object)]" *phrase* [5.3](#)

say "[apostrophe/']" *phrase* [5.2](#)

say "[are]" *phrase* [14.2](#)

say "[aren't]" *phrase* [14.8](#)

say "[as decreasingly likely outcomes]" *phrase* [5.7](#)

say "[at random]" *phrase* [5.7](#)

say "[bold type]" [phrase 5.9, 12.1](#)
 say "[bracket]" [phrase 5.2](#)
 say "[can]" [phrase 14.2, 14.7](#)
 say "[can't]" [phrase 14.7](#)
 say "[can see]" [phrase 14.2](#)
 say "[can touch]" [phrase 14.2](#)
 say "[carry]" [phrase 14.2](#)
 say "[close bracket]" [phrase 5.2](#)
 say "[*(column name)* in/from table *(table name)*]" [phrase 16.4](#)
 say "[command clarification break]" [phrase 5.8](#)
 say "[conditional paragraph break]" [phrase 5.8](#)
 say "[could]" [phrase 14.7](#)
 say "[couldn't]" [phrase 14.7](#)
 say "[current table row]" [phrase 16.4](#)
 say "[cycling]" [phrase 5.7](#)
 say "[don't]" [phrase 14.8](#)
 say "[else]" [phrase 5.6](#)
 say "[end if]" [phrase 5.6](#)
 say "[end unless]" [phrase 5.6](#)
 say "[first time]" [phrase 5.7](#)
 say "[fixed letter spacing]" [phrase 5.9, 12.1](#)
 say "[have]" [phrase 14.2](#)
 say "[haven't]" [phrase 14.8](#)
 say "[here]" [phrase 14.1](#)
 say "[if *(a condition)*]" [phrase 5.6](#)
 say "[in random order]" [phrase 5.7](#)
 say "[infinitive of *(verb)*]" [phrase 14.9](#)
 say "[is-are a list of *(description of objects)*]" [phrase 5.5](#)
 say "[is-are list of *(description of objects)*]" [phrase 5.5](#)
 say "[is-are the list of *(description of objects)*]" [phrase 5.5](#)
 say "[it]" [phrase 14.5](#)
 say "[italic type]" [phrase 5.9, 12.1](#)
 say "[line break]" [phrase 5.8](#)
 say "[list of *(description)*]" [phrase 6.1](#)
 say "[list of *(description of objects)*]" [phrase 5.5](#)
 say "[*(list of objects)* with definite articles]" [phrase 21.3](#)
 say "[*(list of objects)* with indefinite articles]" [phrase 21.3](#)
 say "[*(list of values)* in brace notation]" [phrase 21.3](#)
 say "[may]" [phrase 14.7](#)
 say "[mayn't]" [phrase 14.7](#)
 say "[might]" [phrase 14.7](#)
 say "[mightn't]" [phrase 14.7](#)
 say "[must]" [phrase 14.7](#)
 say "[mustn't]" [phrase 14.7](#)
 say "[negate *(verb)*]" [phrase 14.9](#)
 say "[negate *(verb)* from *(narrative viewpoint)*]" [phrase 14.9](#)
 say "[negate *(verb)* in *(grammatical tense)*]" [phrase 14.9](#)
 say "[negate *(verb)* in *(grammatical tense)* from *(narrative viewpoint)*]" [phrase 14.9](#)
 say "[no line break]" [phrase 5.8](#)
 say "[now]" [phrase 14.1](#)
 say "[*(number)* in words]" [phrase 5.4](#)
 say "[one of]" [phrase 5.7](#)
 say "[only]" [phrase 5.7](#)
 say "[or]" [phrase 5.7](#)
 say "[otherwise]" [phrase 5.6](#)
 say "[otherwise/else if *(a condition)*]" [phrase 5.6](#)
 say "[otherwise/else unless *(a condition)*]" [phrase 5.6](#)
 say "[our]" [phrase 14.4](#)
 say "[paragraph break]" [phrase 5.8](#)
 say "[past participle of *(verb)*]" [phrase 14.9](#)
 say "[possessive]" [phrase 14.6](#)
 say "[present participle of *(verb)*]" [phrase 14.9](#)

say "[purely at random]" [phrase 5.7](#)
 say "[put]" [phrase 14.3](#)
 say "[quotation mark]" [phrase 5.2](#)
 say "[*(real number)* in decimal notation]" [phrase 15.4](#)
 say "[*(real number)* in scientific notation]" [phrase 15.4](#)
 say "[*(real number)* to *(number)* decimal places]" [phrase 15.4](#)
 say "[*(real number)* to *(number)* decimal places in decimal notation]" [phrase 15.4](#)
 say "[*(real number)* to *(number)* decimal places in scientific notation]" [phrase 15.4](#)
 say "[regarding *(number)*]" [phrase 14.5](#)
 say "[regarding *(thing)*]" [phrase 14.5, 14.6](#)
 say "[roman type]" [phrase 5.9, 12.1](#)
 say "[row *(number)* in/from table *(table name)*]" [phrase 16.4](#)
 say "[run paragraph on]" [phrase 5.8](#)
 say "[run paragraph on with special look spacing]" [phrase 5.8](#)
 say "[s]" [phrase 5.4](#)
 say "[*(sayable value)*]" [phrase 5.3](#)
 say "[should]" [phrase 14.7](#)
 say "[shouldn't]" [phrase 14.7](#)
 say "[sticky random]" [phrase 5.7](#)
 say "[stopping]" [phrase 5.7](#)
 say "[text of *(external file)*]" [phrase 23.14](#)
 say "[text of *(response)*]" [phrase 14.10](#)
 say "[the/-- banner text]" [phrase 18.37](#)
 say "[the/-- complete list of extension credits]" [phrase 27.6](#)
 say "[the/-- list of extension credits]" [phrase 27.6](#)
 say "[the/-- player's surroundings]" [phrase 8.3](#)
 say "[the list of *(description of objects)*]" [phrase 5.5](#)
 say "[The list of *(description of objects)*]" [phrase 5.5](#)
 say "[The *(object)*]" [phrase 5.3](#)
 say "[the *(object)*]" [phrase 5.3](#)
 say "[their]" [phrase 14.5](#)
 say "[then at random]" [phrase 5.7](#)
 say "[then purely at random]" [phrase 5.7](#)
 say "[there]" [phrase 14.5](#)
 say "[they]" [phrase 14.2, 14.5](#)
 say "[those]" [phrase 14.6](#)
 say "[*(time)* in words]" [phrase 9.7](#)
 say "[unless *(a condition)*]" [phrase 5.6](#)
 say "[us]" [phrase 14.4](#)
 say "[variable letter spacing]" [phrase 5.9, 12.1](#)
 say "[we]" [phrase 14.4](#)
 say "[wear]" [phrase 14.2](#)
 say "[won't]" [phrase 14.8](#)
 say "[would]" [phrase 14.7](#)
 say "[wouldn't]" [phrase 14.7](#)
 sayable value [22.1, 22.6](#)
 scaled at [15.10](#)
 scaled up / down by [15.9, 15.10](#)
 scenery *(thing)* [property 3.8](#)
 backdrops made automatically scenery [3.9](#)
 scenes [10.1](#)
 beginning when play begins [10.6](#)
 conditions for scenes [10.4](#)
 descriptions (displayed when scene begins) [10.2](#)
 Entire Game [10.3](#)
 for the passage of story time [4.1](#)
 linking to other scenes [10.5, 10.6, 10.7, 10.8](#)
 no way to force scenes to begin / end [10.9](#)
 properties of scenes [10.2, 10.4](#)
 recurring scenes [10.4, 10.7](#)
 rules run at beginning [10.2, 4.4](#)
 rules run at end [10.2, 4.4](#)

- scripting story events in scenes [4.2](#)
- times of scenes beginning/ending [10.3](#)
- variations in ending [10.8](#)
- SCENES** *testing command* [10.3](#), [24.3](#), [13.1](#)
- Scenes page of Index panel *user interface* [10.3](#), [10.9](#)
- scientific notation [15.2](#), [15.4](#)
- scope [17.7](#), [18.29](#), [3.6](#), [5.5](#)
- SCOPE** *testing command* [24.4](#)
- score** — *number global variable* [8.3](#), [9.2](#), [11.4](#)
- scoring [9.2](#), [11.4](#)
 - enabling [2.12](#), [9.2](#)
 - in status line [8.3](#)
 - maximum score [9.2](#)
 - ranking table [9.3](#)
 - reporting at end of story [11.6](#)
 - reporting during story [9.2](#), [11.4](#), [11.5](#)
- scoring** *use option* [2.12](#), [9.2](#), [11.4](#)
- screen top [8.3](#), [18.23](#), [12.2](#)
- SCRIPT ON/OFF** *see* **TRANSCRIPT ON/OFF**
- Search and Seizure* *example* [3.24](#) *ex 38*, [7.3](#) *ex 38*
- searching** *action* [6.6](#)
- second noun part of (action) phrase** [12.20](#)
- second noun** — *object global variable* [7.10](#)
- The Second Oldest Problem* *example* [12.10](#) *ex 200*, [6.9](#) *ex 200*
- second person singular / plural** [14.1](#)
- section** *of source text* [2.5](#)
- see** *relation verb* *see* [visibility](#)
- seed the random-number generator with (number) phrase** [8.18](#)
- seldom / usually**, in assertions [4.3](#)
- senses [7.7](#), [3.8](#)
- sentence case -- (text) in sentence case phrase** [20.4](#)
- sentence verbs [13.1](#), [13.2](#)
 - as values [14.9](#)
 - defining for properties [15.12](#)
 - defining for relations [13.9](#)
 - defining for relations with prepositions [13.10](#)
 - difference from relations [13.3](#)
 - predefined [13.8](#)
- serial comma** *use option* [2.12](#), [2.13](#), [5.5](#), [21.3](#)
- set pronouns from (object) phrase** [17.18](#)
- set the locale priority of (object) to (number) phrase** [18.27](#)
- setting action variables** *rulebook* [12.10](#)
- Shawn's Bad Day* *example* [17.4](#) *ex 294*, [6.15](#) *ex 294*
- Shipping Trunk* *example* [18.10](#) *ex 339*, [9.2](#) *ex 339*
- Short, Emily [1.2](#), [2.10](#), [18.37](#), [25.9](#), [27.4](#), [27.7](#), [3.5](#), [4.5](#), [6.13](#)
- shorthand *see* [abbreviations](#)
- "[should]" *say phrase* [14.7](#)
- "[shouldn't]" *say phrase* [14.7](#)
- show relation (relation of values to values) phrase** [13.7](#)
- SHOWHEAP** *testing command* [24.4](#)
- SHOWME** *testing command* [2.7](#), [24.2](#), [24.3](#), [24.4](#), [13.1](#)
- showme the contents of (table name) phrase** [16.4](#)
- showme (value) phrase** [11.4](#)
- SHOWVERB** *testing command* [24.4](#)
- siblings, in containment hierarchy [8.17](#)
- Sieve of Eratosthenes* *example* [21.11](#) *ex 437*, [10.11](#) *ex 437*
- Signs and Portents* *example* [4.9](#) *ex 53*, [9.8](#) *ex 53*
- silently try (action) phrase** [7.4](#), [12.1](#)
- sine of (real number) phrase** [15.7](#)
- singular-named / plural-named (thing) property** [3.18](#)
- Situation Room* *example* [9.7](#) *ex 142*, [4.1](#) *ex 142*
- Skein panel *user interface* [1.7](#), [1.8](#), [1.9](#), [25.16](#), [13.1](#)

Skeleton Keys *extension* [9.2](#)
 slash *see* [punctuation: slash](#)
 sleeping *action* [6.11](#)
 Slightly Wrong *example* [3.1 ex 4](#), [3.1 ex 4](#)
 Slogar's Revenge *example* [12.17 ex 214](#), [6.3 ex 214](#)
 Slouching *example* [19.4 ex 396](#), [5.1 ex 396](#)
 slow route-finding *use option* [6.14](#)
 for particular relations [13.11](#)
 smelling *action* [7.7](#)
 smoke (implementing) [10.1](#)
 Snip *example* [15.16 ex 261](#), [10.6 ex 261](#)
 snippets [17.5](#), [18.33](#)
 Solitude *example* [19.8 ex 400](#), [11.3 ex 400](#)
 some
 article [3.18](#)
 in descriptions [6.15](#)
 Some Assembly Required *example* [17.22 ex 332](#), [9.3 ex 332](#)
 somebody, in descriptions [6.2](#)
 someone
 in actions [12.12](#), [12.13](#)
 in descriptions [6.2](#)
 [someone] *grammar token* [17.1](#), [17.4](#)
 something
 in descriptions [6.2](#)
 naming value of [8.15](#)
 [something] *grammar token* [17.1](#), [17.2](#), [17.4](#)
 Something Narsty *example* [4.3 ex 45](#), [3.5 ex 45](#)
 something new, in understanding actions [17.3](#)
 [something preferably held] *grammar token* [17.4](#)
 [something related by (relation)] *grammar token* [17.16](#)
 somewhere, in descriptions [6.2](#)
 sort (*list of objects*) in (*property*) order *phrase* [21.8](#)
 sort (*list of objects*) in reverse (*property*) order *phrase* [21.8](#)
 sort (*list of values*) *phrase* [21.8](#)
 sort (*list of values*) in random order *phrase* [21.8](#)
 sort (*list of values*) in reverse order *phrase* [21.8](#)
 sort (*table name*) in random order *phrase* [16.11](#)
 sort (*table name*) in reverse (*table column*) order *phrase* [16.11](#)
 sort (*table name*) in (*table column*) order *phrase* [16.11](#)
 sound name [23.10](#)
 sounds [23.7](#), [12.5](#)
 accessibility text [23.9](#)
 as values [23.10](#)
 declaring [23.8](#)
 formats allowed [23.7](#)
 placing in materials folder [23.8](#)
 source text
 releasing with the story [25.17](#)
 subdivisions [2.5](#), [2.6](#), [2.9](#), [25.6](#)
 Space Patrol - Stranded on Jupiter! *example* [10.4 ex 158](#), [4.4 ex 158](#)
 spaces, in list constants, after commas [21.2](#)
 spaces vs tabs (in tables) [9.3](#), [16.1](#)
 specification of (*thing*) *property* [27.10](#)
 (*spec*) specifies (*a unit*) *assertion* [15.8](#)
 The Speed of Thought *example* [15.20 ex 270](#), [10.5 ex 270](#)
 Spellbreaker *example* [12.15 ex 210](#), [11.2 ex 210](#)
 Spencer, Henry [20.6](#)
 Spring Cleaning *example* [8.10 ex 123](#), [10.4 ex 123](#)
 square brackets *see* [punctuation: square brackets](#)
 square root of (*arithmetic value*) *phrase* [15.5](#)
 Standard Rules *extension* [27.2](#), [27.14](#)
 standing *see* [exiting](#)

Starry Void example [3.2 ex 7](#), [3.3 ex 7](#)
 starting conditions [2.1](#), [3.1](#)
 scenes that begin at the beginning [10.6](#)
 setting in **when play begins** [9.1](#), [11.1](#)
starting the virtual machine activity [18.40](#)
Stately Gardens example [18.29 ex 367](#), [3.4 ex 367](#)
 status line [8.3](#), [18.23](#), [12.2](#)
Status line with centered text, the hard way example [27.19 ex 452](#), [12.2 ex 452](#)
Stone example [19.3 ex 391](#), [9.1 ex 391](#)
stop phrase [11.13](#)
stop the action phrase [7.3](#), [19.11](#)
 using **instead** *see* [\(instead\)](#), to stop the action
 stored actions [12.20](#)
 story description *bibliographic data* [25.2](#), [26.3](#)
 story structure
 beginning [9.1](#), [11.1](#); *see also* [initial state of the world](#)
 changing viewpoint [5.6](#)
 characterization [5.3](#), [7.10](#)
 cut scenes [4.5](#), [7.12](#)
 dialogue [7.6](#), [7.9](#)
 dynamic plot management [4.6](#)
 ending [9.4](#), [11.6](#); *see also* [end of story](#)
 flashbacks [4.5](#)
 providing background information [5.4](#)
 scenes in different environments [4.4](#)
 scenes with scripted events [4.2](#)
story tense — *tense global variable* [14.1](#)
story viewpoint — *narrative viewpoint global variable* [14.1](#)
Straw Boater example [4.8 ex 49](#), [8.1 ex 49](#)
Straw Into Gold example [17.9 ex 305](#), [2.1 ex 305](#)
Strictly Ballroom example [11.11 ex 176](#), [7.16 ex 176](#)
 string *see* [ropes \(implementing\)](#)
 strings of characters *see* [text](#)
 subheadings [2.5](#), [2.6](#), [2.9](#), [25.6](#)
substituted / unsubstituted (text) adjective [20.7](#)
substituted form of (text) phrase [20.7](#)
 success
 for actions [7.1](#), [12.2](#)
 for persuasion [12.4](#)
 for rules [19.11](#), [19.13](#)
sufficient / insufficient light, there is sufficient / insufficient light (visibility rules) outcome phrase [12.19](#)
SUPERBRIEF room descriptions use option [2.12](#), [6.4](#), [11.5](#)
 superlatives
 of numbers using adjectives [6.8](#)
 of numbers with units using adjectives [15.12](#)
supplying a missing noun/second noun activity [17.1](#), [18.32](#)
 support *relation* [8.13](#), [13.4](#)
supporter in question — *object global variable* [12.18](#)
 supporters *kind* [3.5](#), [8.4](#)
 carrying capacity [3.19](#)
 default immobility [3.7](#)
 mentioned because of supported things [3.8](#)
 support relation *see* [support](#)
Sweeney example [16.16 ex 282](#), [7.8 ex 282](#)
Swerve left? Swerve right? Or think about it and die? example [13.10 ex 232](#), [10.5 ex 232](#)
Swigmore U. example [19.4 ex 397](#), [8.4 ex 397](#)
switched on / off (device) property [3.14](#), [9.8](#), [10.7](#)
Sybil 1 example [7.6 ex 91](#), [7.7 ex 91](#)
Sybil 2 example [7.6 ex 93](#), [7.7 ex 93](#)
 symbols *see* [characters \(letters\)](#)
symmetric (relation) adjective [13.14](#)

synonyms [12.7](#), [6.2](#)
T **table name** [16.15](#)
Table of Final Question Options [11.6](#)
 tables [9.3](#), [16.1](#)

- adding rows [16.10](#)
- amending tables [16.19](#)
- blank columns [16.8](#), [16.10](#)
- blank entries [16.7](#), [16.10](#)
- blank rows [16.9](#), [16.10](#)
- changing entries [16.4](#)
- choosing rows [16.5](#)
- continuing tables [16.18](#)
- creating multiple things with a table [16.16](#)
- defining [16.1](#)
- defining a kind of value with a table [16.17](#)
- defining multiple kinds with a table [16.16](#)
- describing things listed in tables [16.12](#)
- empty / non-empty** *adjective* [16.9](#)
- formatting [16.1](#)
- looking up by cross-reference [16.3](#)
- looking up by row number [16.2](#)
- ranking score [9.3](#)
- reading / writing to data files [23.13](#)
- removing rows [16.10](#)
- repeating through [16.6](#)
- sorting tables [16.11](#)
- tabs in formatting tables [9.3](#), [16.1](#)
- topic columns [16.13](#)
- used as values [16.15](#)

 tabs, for indentation *of source text* [11.7](#), [11.9](#)
 tabs vs spaces, in tables *of source text* [9.3](#), [16.1](#)
taking *action* [7.1](#), [6.8](#)

- implicit taking [12.17](#), [18.34](#), [6.8](#)

taking inventory *action* [6.7](#)
Tamed example [3.6 ex 13](#), [8.4 ex 13](#)
tangent of (real number) *phrase* [15.7](#)
 Tarbert, Mike [1.2](#)
tasting *action* [7.7](#)
Technological Terror example [8.11 ex 127](#), [7.5 ex 127](#)
telemetry recordings *use option* [2.13](#)
 telephones (implementing) [9.10](#)
 televisions (implementing) [9.9](#)
telling (actor) about *action* [7.6](#)
 templates, for web pages for the story [25.13](#), [25.14](#)
 templates, Inform 6 template layer [27.26](#), [6.3](#)
 Temple, Erik [1.2](#), [12.4](#)
 temporary variables [11.15](#)

- displaying [20.7](#)

 tense

- of conditions in rules [9.12](#), [9.13](#)
- of standard responses [14.1](#), [5.6](#)

Tense Boxing example [9.13 ex 149](#), [5.5 ex 149](#)
Terracottissima example [17.15 ex 314](#), [10.4 ex 314](#)
Terracottissima Maxima example [17.15 ex 317](#), [2.3 ex 317](#)
Terror of the Sierra Madre example [8.9 ex 121](#), [5.6 ex 121](#)
 Tessman, Kent [1.2](#)
TEST *testing command* [2.8](#), [24.2](#)
test (test name) with (commands) *assertion* [2.8](#), [24.2](#)
 test version (of the story) [2.9](#), [24.5](#), [13.1](#)
 testing commands [2.9](#), [13.1](#)

- ABSTRACT** [24.4](#)
- ACTIONS** [7.1](#), [12.1](#), [12.5](#), [12.12](#), [24.3](#), [10.10](#), [13.1](#)

- checking story correctness in the Index [24.1](#)
- comments in transcripts from beta testers [17.21](#), [24.7](#), [13.1](#)
- defining [2.9](#), [24.5](#), [13.1](#)
- GONEAR [24.4](#)
- PURLOIN [2.8](#), [24.4](#)
- RANDOM [24.3](#)
- RELATIONS [13.7](#), [24.3](#)
- RESPONSES [14.12](#), [24.3](#)
- RULES [24.3](#), [6.2](#)
- SCENES [10.3](#), [24.3](#), [13.1](#)
- SCOPE [24.4](#)
- SHOWHEAP [24.4](#)
- SHOWME [2.7](#), [24.2](#), [24.3](#), [24.4](#), [13.1](#)
- SHOWVERB [24.4](#)
- TEST [2.8](#), [24.2](#)
- TRACE [24.4](#)
- TREE [24.4](#)
- text [5.1](#), [20.1](#)
 - accessibility text for figures and sounds [23.9](#)
 - changing case [20.4](#)
 - counting characters / words / paragraphs / lines [20.3](#)
 - displaying [5.1](#)
 - empty / non-empty *adjective* [6.5](#), [20.3](#)
 - in properties [3.10](#)
 - indexed text, now the same as text [20.1](#)
 - lower case [20.4](#)
 - matching [20.5](#)
 - memory limits [20.2](#)
 - reading / writing / appending to data files [23.14](#)
 - recording from the player's command [2.3](#)
 - regular expressions [18.33](#), [20.6](#), [20.8](#), [20.9](#)
 - replacing parts of a text [20.8](#)
 - substituted / unsubstituted *adjective* [20.7](#)
 - substitutions *see* text substitutions
 - upper case [20.4](#)
 - when text substitutions are substituted [20.7](#)
 - Z-machine limitations [20.2](#), [20.4](#)
- [text] *grammar token* [17.5](#)
 - in understanding mistakes [17.21](#)
- text files *see* files (data files)
- Text Foosball *example* [9.5](#) ex 140, [7.2](#) ex 140
- text matching regular expression *phrase* [20.6](#)
- text matching subexpression (*number*) *phrase* [20.6](#)
- text substitutions [3.14](#), [2.1](#)
 - defining [5.13](#)
 - enumerated values [4.9](#)
 - lists [5.5](#), [21.3](#)
 - new kinds of value [4.9](#)
 - numbers [5.3](#), [5.4](#)
 - punctuation [5.2](#)
 - random variations [5.7](#)
 - real numbers [15.4](#)
 - sayable values [22.1](#), [22.6](#)
 - sequential variations [5.7](#)
 - text [5.1](#)
 - things [5.3](#)
 - units of measure [15.8](#), [15.11](#), [15.15](#)
 - values [5.3](#)
 - variations [5.6](#)
 - when text is substituted [20.7](#)
- (*name of kind*) that/which/who relates to (value) by (*relation of values to values*) *phrase* [13.13](#)
- (*name of kind*) that/which/whom (value) relates to by (*relation of values to values*) *phrase* [13.13](#)

that, adaptive pronouns in displayed text [14.6](#)

that, in description clauses [6.9](#)

that varies, in defining global variables [4.12](#)

the / a / an [3.18](#), [5.3](#), [5.5](#)

"[their]" *say phrase* [14.5](#)

THEM, in player commands [17.18](#), [2.2](#)

"[there]" *say phrase* [14.5](#)

there is, declaring existence [6.10](#)

there is sufficient / insufficient light (*visibility rules*) *outcome phrase* [12.19](#)

"[they]" *say phrase* [14.2](#), [14.5](#)

things *kind* [3.5](#), [4.1](#)

- changing location of [8.7](#), [8.8](#), [8.9](#)
- changing multiples with **now** [8.11](#)
- counting [8.16](#)
- creating [3.1](#)
 - in Inform 6 [27.21](#)
 - multiple identical things with counts [4.14](#)
 - multiple things with a table [16.16](#)
 - using plurals [4.4](#)
 - using relations [4.15](#), [4.16](#)
- displaying [5.3](#)
- immobile
 - because they're supporters [3.7](#)
 - fixed in place** [3.6](#), [8.7](#)
 - moving anyway [8.7](#)
 - scenery** [3.8](#)
- in more than one room [3.9](#)
- kinds of [4.1](#), [4.2](#), [4.4](#)
- location of [3.25](#)
- parts of [4.15](#)
- parts of things [3.23](#)
- pushable between rooms [3.16](#), [7.14](#), [6.9](#)
- references to (descriptions) [6.1](#)
- referring to arbitrary things in rules [7.10](#)
- removing from play [8.10](#)
- starting out of play [3.25](#), [6.10](#), [8.10](#)

[things] *grammar token* [17.4](#)

[things inside] *grammar token* [17.4](#)

THINK ABOUT [5.4](#)

third person singular / plural [14.1](#)

Thirst example [8.5 ex 117](#), [10.2 ex 117](#)

Thirst 2 example [8.5 ex 118](#), [10.8 ex 118](#)

this is, in naming phrases [22.3](#)

this is the (name) rule... *assertion* [19.3](#), [19.7](#)

Thompson, Ken [20.6](#)

Thornton, Adam [1.2](#)

"[those]" *say phrase* [14.6](#)

3 AM example [12.7 ex 193](#), [10.2 ex 193](#)

through *relation verb* [3.12](#)

Tilt 1 example [17.15 ex 318](#), [9.5 ex 318](#)

Tilt 2 example [19.13 ex 407](#), [9.5 ex 407](#)

Tilt 3 example [27.7 ex 450](#), [12.1 ex 450](#)

time [3.9](#), [4.1](#)

- calculations [9.9](#), [9.10](#)
- clocks (implementing) [9.11](#)
- comparing [9.9](#)
- conditions and rules referring back in time [9.13](#)
- counting actions with **for the Nth time** [7.16](#), [11.4](#)
- counting conditions with **for the Nth time** [9.14](#)
- displaying approximations [9.8](#)
- displaying times [9.7](#)
- events in the future [9.11](#)

- not passing for **out of world** actions [12.15](#)
- scene beginning/ending times [10.3](#)
- scripting story events by time [4.3](#)
- time of day [9.6](#)
- timed input [12.4](#)
- waiting intervals of time [6.11](#)
- [time]** *grammar token* [17.9](#)
- time of day** — *time global variable* [8.2](#), [8.3](#), [9.6](#), [9.7](#)
- [time period]** *grammar token* [17.9](#)
- time since (scene) began** *phrase* [10.3](#)
- time since (scene) ended** *phrase* [10.3](#)
- time when (scene) began** *phrase* [10.3](#)
- time when (scene) ended** *phrase* [10.3](#)
- Timeless example* [19.15](#) *ex 409*, [4.1](#) *ex 409*
- (arithmetic value) times (arithmetic value)** *phrase* [15.5](#)
- Tiny Garden example* [5.5](#) *ex 62*, [3.4](#) *ex 62*
- title *bibliographic data* [2.5](#), [18.37](#), [25.2](#)
- title case -- (text) in title case** *phrase* [20.4](#)
- to be *see is*
- to decide what/which (value)** *assertion* [11.17](#)
- to decide whether/if** *assertion* [11.16](#)
- to (phrase name)...** *assertion* [11.2](#), [19.7](#)
- to say (substitution)...** *assertion* [5.13](#)
- (arithmetic value) to the nearest (arithmetic value)** *phrase* [9.8](#), [15.5](#)
- (real number) to the nearest whole number** *phrase* [15.3](#)
- (real number) to the power (real number)** *phrase* [15.6](#)
- to (V) is a verb** *assertion* [14.3](#), [14.7](#)
- (name of kind) to which/whom (value) relates by (relation of values to values)** *phrase* [13.13](#)
- Today Tomorrow example* [7.12](#) *ex 99*, [8.3](#) *ex 99*
- together -- group (description of objects) together** *phrase* [18.13](#)
- tokens of grammar *see grammar tokens*
- Tom's Midnight Garden example* [17.6](#) *ex 299*, [9.11](#) *ex 299*
- topic understood — snippet** *global variable* [17.5](#)
- topics
 - in table columns [16.13](#)
 - understanding [17.5](#)
- total (arithmetic values valued property) of (description of values)** *phrase* [15.17](#)
- Totality example* [9.11](#) *ex 144*, [3.9](#) *ex 144*
- touchability *relation* [6.13](#)
 - in action processing sequence [12.2](#), [12.16](#)
 - requirement in defining actions [12.7](#), [12.17](#), [12.18](#)
- touchable / untouchable (thing)** *adjective* [6.13](#)
- touching** *action* [7.7](#)
- TRACE** *testing command* [24.4](#)
- Trachypachidae Maturin 1803 example* [18.10](#) *ex 340*, [9.2](#) *ex 340*
- TRANSCRIPT ON/OFF** [24.7](#), [11.5](#)
- Transcript panel *user interface* [1.8](#), [1.9](#)
- transcripts
 - creating with the TEST command [2.8](#)
 - in the Skein [1.8](#), [1.9](#)
- translates as...** *assertion*
 - actions [27.22](#)
 - global variables [27.22](#)
 - kinds [27.21](#)
 - properties [27.21](#), [27.22](#)
 - rules [27.20](#)
 - things [27.21](#)
 - understanding (grammar tokens) [27.23](#)
 - Unicode characters [27.25](#)
 - use options [27.18](#)
- Transmutations example* [13.7](#) *ex 227*, [10.10](#) *ex 227*
- transparent / opaque (container)** *property* [3.6](#), [3.15](#), [9.2](#)

Treaty of Babel [25.5](#)
TREE, *testing command* [24.4](#)
Trieste example [16.19 ex 285](#), [11.3 ex 285](#)
The Trouble with Printing example [17.3 ex 292](#), [9.6 ex 292](#)
true / false [11.5](#)
truncate (*list of values*) to (*number*) entries/entry *phrase* [21.10](#)
truncate (*list of values*) to the first (*number*) entries/entry *phrase* [21.10](#)
truncate (*list of values*) to the last (*number*) entries/entry *phrase* [21.10](#)
truth state [11.5](#)
try (*action*) *phrase* [7.4](#), [12.1](#)
 in **asking action** *see* **asking (actor) to try**
try (actor) trying (action) *phrase* [12.6](#)
try silently (*action*) *phrase* [7.4](#)
Trying Taking Manhattan example [12.11 ex 204](#), [6.7 ex 204](#)
turn count — *number* *global variable* [5.4](#), [8.3](#), [9.5](#)
turn sequence *rulebook* [19.15](#)
turning devices on / off [3.14](#), [9.8](#), [10.7](#)
turns
 counting actions with **for the Nth turn** [7.17](#)
 counting conditions with **for the Nth turn** [9.15](#)
 events in the future [9.11](#)
 not passing for **out of world** actions [12.15](#)
 not passing for specific actions [4.1](#)
 rules run every turn [9.5](#)
 turn sequence [19.15](#)
 using time of day instead of turns [9.6](#)
twice, counting conditions [9.14](#)
type-checking [8.1](#), [8.4](#)
 for global variables [4.12](#)
 for temporary variables [11.15](#)
 in conditions [11.6](#)
 in grammar tokens [17.14](#)
 in input grammar [17.6](#)
 in lists [21.1](#)
 in phrase definitions [11.3](#)
 in phrases that decide values [11.17](#)
 in tables [16.1](#), [16.4](#)
 of relation types [13.14](#)
 of units of measure [15.8](#), [15.19](#), [15.20](#)
types *see* **kinds**

U *U-Stor-It example* [4.15 ex 58](#), [8.4 ex 58](#)
unabbreviated object names *use option* [3.2](#)
Unblinking example [6.14 ex 81](#), [3.7 ex 81](#)
The Unbuttoned Elevator Affair example [3.3 ex 9](#), [8.2 ex 9](#)
Uncommon Ground example [27.30 ex 455](#), [5.6 ex 455](#)
Under Contract example [12.11 ex 205](#), [7.14 ex 205](#)
understand nothing as (action) *assertion* [17.3](#)
understand (nouns) as (thing) *assertion* [17.8](#), [17.11](#), [17.15](#), [17.16](#)
understand (property) as describing/referring to (thing) *assertion* [17.15](#)
understand the command (verb) as (verb) *assertion* [17.2](#)
understand (verb) as (action) *assertion* [12.7](#), [12.8](#), [17.1](#), [17.2](#), [17.3](#), [6.2](#)
understand (verb) as something new *assertion* [17.3](#)
understand (words) as a mistake *assertion* [17.21](#), [5.3](#)
understanding [17.1](#), [2.2](#)
 adding synonyms for verbs [17.2](#)
 alternatives to standard parsing [6.18](#)
 arbitrary objects [17.4](#)
 arbitrary text [17.5](#)
 for conversations and consulting [7.6](#)
 choosing between multiple grammar lines [17.22](#)
 conditionally [17.17](#), [2.2](#)
 deciding whether **ALL** includes something [18.36](#), [6.15](#)

disambiguation *see* [disambiguation: of player commands](#)
 error messages for parsing errors [18.35](#), [6.17](#)
 grammar tokens *see* [grammar tokens](#)
 kinds of things [17.11](#)
 limiting cases where understand rules apply [17.17](#), [2.2](#)
 mistakes [17.21](#), [5.3](#)
 noun-first sentences [17.10](#)
 numbers [17.6](#), [17.9](#)
 numbers with units [15.16](#)
 precedence of commands understood [17.22](#)
 properties as describing things [17.15](#)
 recording the player's command [2.3](#)
 relations as describing things [17.16](#)
 removing existing definitions of verbs [17.3](#), [17.3](#)
 requirements for objects in actions [12.7](#), [12.17](#), [12.18](#), [6.2](#)
 sentences without verbs [17.10](#)
 special processing of keyboard input [18.33](#)
 synonyms [17.8](#), [17.12](#)
 as grammar tokens [17.13](#)
 for things and values [17.11](#)
 things
 by properties [17.15](#)
 by relations [17.16](#)
 kinds of things [17.11](#)
 not in scope with **any** [17.7](#), [5.5](#)
 specific things [17.8](#)
 values [17.11](#)
 grammar tokens producing values [17.14](#)
 kinds of value [17.9](#)
 numbers [17.6](#)
 verbs [12.7](#), [12.8](#), [17.1](#), [17.2](#), [17.3](#), [6.2](#)
when clauses for [17.17](#)
 with Inform 6 functions [27.23](#)
(kind) understood — *kind of value* *global variable* [17.9](#)
The Undertomb 1 example [4.9 ex 50](#), [3.8 ex 50](#)
The Undertomb 2 example [4.9 ex 51](#), [3.7 ex 51](#)
undescribed / described (*thing*) *property* [3.24](#)
UNDO [2.12](#), [7.5](#), [11.2](#), [11.6](#)
undo prevention *use option* [2.12](#), [7.5](#), [11.2](#)
The Unexamined Life example [13.9 ex 230](#), [11.3 ex 230](#)
 Unicode *see* [characters \(letters\)](#)
 Unicode Character Names / Full Character Names *extension* [5.11](#), [27.25](#)
unindexed, headings in extensions [27.10](#)
 units of measure [4.5](#), [15.1](#)
 arithmetic [15.19](#)
 averages [15.17](#)
 defining [15.8](#), [15.9](#), [15.10](#), [15.14](#), [15.15](#)
 displaying [15.8](#), [15.11](#), [15.15](#)
 for capacity of containers [10.5](#)
 Metric Units *extension* [15.13](#), [15.18](#), [15.20](#)
 multiple notations for a unit [15.9](#), [15.10](#), [15.11](#)
 multiplication of units [15.20](#)
 totals [15.17](#)
 understanding [15.16](#)
 with multiple numeric components [15.14](#), [15.15](#)
unless (*a condition*) *phrase* [11.6](#)
unless (*a condition*) , (*a phrase*) *phrase* [11.6](#)
unlikely / possible / likely, it is unlikely / possible / likely (*does the player mean*) *outcome phrase* [17.19](#)
unlit / lit (*thing*) *property* [3.15](#), [3.7](#)
unlocked / locked (*container/door*) *property* [3.13](#), [9.2](#)
unlocks *relation verb* [3.13](#)

- unmarked for listing / marked for listing (*thing*) *property* [18.25](#), [3.1](#)
- unmentioned / mentioned (*thing*) *property* [18.24](#), [18.26](#), [18.28](#), [3.1](#)
- unopenable / openable (*container*) *property* [3.6](#), [9.2](#)
- Unpeeled example* [18.13](#) *ex 344*, [9.2](#) *ex 344*
- unpunctuated word number (*number*) in (*text*) *phrase* [20.3](#)
- unsubstituted / substituted (*text*) *adjective* [20.7](#)
- unsuccessful attempt by *rulebook* [12.5](#), [12.12](#)
 - not used for spontaneous actions [12.6](#)
- Unthinkable Alliances example* [13.9](#) *ex 229*, [7.16](#) *ex 229*
- untouchable / touchable (*thing*) *adjective* [6.13](#)
- Up and Up example* [3.2](#) *ex 6*, [6.9](#) *ex 6*
- update backdrop positions *phrase* [8.8](#)
- upper case -- (*text*) in upper case *phrase* [20.4](#)
- Uptempo example* [19.4](#) *ex 394*, [4.1](#) *ex 394*
- Uptown Girls example* [8.18](#) *ex 132*, [3.9](#) *ex 132*
- "[us]" *say phrase* [14.4](#)
- USE [6.17](#)
- use (*options...*) *assertion* [2.12](#)
- use options [2.12](#)
 - active / inactive *adjective* [27.18](#)
 - catalogue
 - abbreviated room descriptions [2.12](#)
 - American dialect [2.12](#), [2.13](#), [5.4](#)
 - authorial modesty [27.6](#)
 - BRIEF room descriptions [2.12](#), [6.4](#)
 - DICT_WORD_SIZE [2.2](#)
 - dynamic memory allocation [20.2](#)
 - engineering notation [15.2](#)
 - fast route-finding [6.14](#)
 - full-length room descriptions [2.12](#)
 - Inform 6 memory limits [2.14](#)
 - maximum text length [20.2](#)
 - maximum things understood at once [4.14](#)
 - memory economy [2.14](#)
 - scoring [2.12](#), [9.2](#), [11.4](#)
 - slow route-finding [6.14](#)
 - SUPERBRIEF room descriptions [2.12](#), [6.4](#)
 - telemetry recordings [2.13](#)
 - the serial comma [2.12](#), [2.13](#), [5.5](#), [21.3](#)
 - unabbreviated object names [3.2](#)
 - undo prevention [2.12](#), [7.5](#), [11.2](#)
 - VERBOSE room descriptions [2.12](#), [6.4](#)
 - defining in Inform 6 [27.18](#)
- user interface [1.3](#)
 - facing pages layout [1.3](#)
 - Go button [1.4](#)
 - Index panel [1.6](#)
 - Actions page [12.5](#), [12.21](#), [17.1](#), [17.2](#), [17.3](#), [24.3](#)
 - checking story correctness [24.1](#)
 - Contents page [2.6](#), [23.5](#), [27.10](#)
 - Kinds page [4.1](#), [4.6](#), [4.8](#), [4.11](#), [4.12](#), [27.10](#)
 - Map page [2.7](#), [3.4](#), [3.26](#), [25.18](#)
 - Phrases page [2.2](#), [11.2](#)
 - Rules page [19.6](#), [19.16](#)
 - Scenes page [10.3](#), [10.9](#)
 - Release button [2.7](#), [25.6](#)
 - Replay button [1.5](#), [1.7](#)
 - Results panel [1.6](#)
 - Skein panel [1.7](#), [1.8](#), [1.9](#), [25.16](#), [13.1](#)
 - Transcript panel [1.8](#), [1.9](#)
- using doors, in route-finding [6.14](#)
- usually / seldom, in assertions [4.3](#)

V value properties [3.10](#), [4.6](#)
 changing [8.5](#)
 defining [4.8](#), [4.9](#), [4.10](#)
 numeric comparatives, adjectives for [6.7](#), [15.12](#)
 numeric magnitudes, adjectives for [6.6](#)
 numeric superlatives, adjectives for [6.8](#), [15.12](#)
 used as adjectives [6.3](#)

values
 actions as values [12.20](#)
 basis of rulebooks [19.9](#)
 constant *see* [constant values](#)
 descriptions as values [22.2](#)
 displaying [4.9](#), [5.3](#)
 figures as values [23.10](#)
 in relations [13.13](#)
 kinds of kinds of value [22.1](#), [22.6](#)
 kinds of value [4.5](#), [4.9](#), [4.10](#)
 comparisons [11.18](#)
 default values [4.11](#)
 defining with a table [16.17](#)
 understanding [17.9](#), [17.14](#)
 phrases as values [22.3](#)
 produced by rulebooks [19.13](#)
 relations as values [13.14](#)
 sounds as values [23.10](#)
 tables as values [16.15](#)
 truth state values [11.5](#)
 understanding [17.6](#)
 varying *see* [variables: global](#)

Van Helsing example [3.25 ex 39](#), [7.13 ex 39](#)

variables
 changing global variables [8.1](#)
 changing temporary variables [11.15](#)
 for actions [12.10](#)
 for activities [18.6](#)
 for rulebooks [19.10](#)
 global [4.12](#)
 defining in Inform 6 [27.22](#)
 kind variables [22.1](#), [22.7](#)
 temporary [11.15](#)
 displaying [20.7](#)

Variety example [14.3 ex 246](#), [2.1 ex 246](#)
Variety 2 example [14.3 ex 247](#), [2.1 ex 247](#)

various-to-one (*relation*) *adjective* [13.14](#)
various-to-various (*relation*) *adjective* [13.14](#)
various, in defining relations [13.5](#)
"['ve]" *say phrase* [14.8](#)

vehicles *kind* [3.16](#), [8.1](#)
 moving rooms [8.2](#)
 rules for going by vehicles [7.14](#)

the verb (V) means (property) assertion [15.12](#)
the verb (V) means (relation) assertion [13.9](#)
the verb to be (P) means (relation) assertion [13.10](#)

VERBOSE room descriptions *use option* [2.12](#), [6.4](#), [11.5](#)

Verbosity 1 example [3.1 ex 3](#), [6.4 ex 3](#)
Verbosity 2 example [19.4 ex 395](#), [6.4 ex 395](#)

verbs
 adaptive verbs in displayed text [14.2](#)
 as values [14.9](#)
 defining adaptive displayable verbs [14.3](#)
meaningful / meaningless *adjective* [14.9](#)
modal / non-modal *adjective* [14.7](#), [14.9](#)

sentence verbs *see* [sentence verbs](#)
 understanding verbs in player's command *see* [understanding](#)
VERIFY [24.4](#), [11.5](#)
Veronica *example* [7.13](#) *ex* [100](#), [6.9](#) *ex* [100](#)
Versailles *example* [6.9](#) *ex* [75](#), [8.5](#) *ex* [75](#)
VERSION [18.37](#), [23.12](#), [26.6](#), [27.5](#), [27.6](#), [11.5](#)
 version number (of extension) [27.6](#)
 version number (of story) *see* [release number](#)
 vertical stroke *see* [punctuation: vertical stroke, for paragraph break](#)
very likely / unlikely *see* [likely / possible / unlikely, it is likely / possible / unlikely \(does the player mean\)](#)
 Vesine, Justin de [1.2](#)
 victory [9.4](#), [18.38](#), [18.39](#), [11.6](#)
A View of Green Hills *example* [6.14](#) *ex* [80](#), [3.4](#) *ex* [80](#)
 virtual machine [2.14](#), [23.3](#)

- character set limits [20.4](#)
- extensions for Z-machine / Glulx only [27.7](#)
- Glulx *see* [Glulx](#)
- memory limits [2.14](#), [20.2](#), [21.10](#)
- multimedia support [23.1](#), [12.5](#)
- numeric limits [11.10](#), [15.2](#), [15.10](#), [15.14](#)
- starting the virtual machine** *activity* [18.40](#)
- Z-machine *see* [Z-machine](#)

Virtue *example* [12.3](#) *ex* [184](#), [7.14](#) *ex* [184](#)
visibility *rulebook* [12.19](#)
 visibility *relation* [6.13](#)

- in action processing sequence [12.2](#)
- requirement in defining actions [12.7](#), [12.17](#), [12.19](#), [17.1](#)
- rules for determining darkness [12.19](#)

visible / invisible (thing) *adjective* [6.13](#)
Vitrine *example* [8.4](#) *ex* [116](#), [3.6](#) *ex* [116](#)
volume *of source text* [2.5](#)
 Vorple system *extension* [25.12](#)
Vouvray *example* [4.1](#) *ex* [43](#), [2.2](#) *ex* [43](#)
W *Wainwright Acts* *example* [13.12](#) *ex* [236](#), [3.5](#) *ex* [236](#)
waiting *action* [6.11](#)
 walkthrough, generated from Skein [25.16](#)
Walls and Noses *example* [18.31](#) *ex* [370](#), [6.17](#) *ex* [370](#)
was, for past conditions [9.13](#)
 water *see* [liquids \(implementing\)](#)
Waterworld *example* [12.18](#) *ex* [216](#), [3.4](#) *ex* [216](#)
Ways Out *example* [18.23](#) *ex* [350](#), [12.2](#) *ex* [350](#)
 "[we]" *say phrase* [14.4](#)
 "[wear]" *say phrase* [14.2](#)
wearable (thing) *property* [3.20](#), [9.3](#)
 wearing *relation* [3.20](#), [8.7](#), [13.4](#)
Weathering *example* [8.18](#) *ex* [131](#), [3.9](#) *ex* [131](#)
Wesponses *example* [18.15](#) *ex* [345](#), [2.1](#) *ex* [345](#)
Western Art History 305 *example* [17.20](#) *ex* [328](#), [6.15](#) *ex* [328](#)
What Makes You Tick *example* [21.6](#) *ex* [427](#), [9.8](#) *ex* [427](#)
What Not To Wear *example* [13.16](#) *ex* [242](#), [9.3](#) *ex* [242](#)
When? *example* [5.6](#) *ex* [63](#), [3.5](#) *ex* [63](#)
when (condition)

- arbitrary conditions for rules [7.12](#), [9.5](#), [19.7](#)
- arbitrary conditions for **understand** [17.17](#)

when play begins *rulebook* [8.2](#), [9.1](#), [19.15](#), [11.1](#)

- printing the banner text** *activity* [18.37](#)

when play ends *rulebook* [9.4](#), [19.15](#), [11.6](#)

- amusing a victorious player *activity* [18.39](#)
- printing the player's obituary *activity* [18.38](#)

when (scene) begins / ends [10.2](#)
Whence? *example* [5.6](#) *ex* [64](#), [3.5](#) *ex* [64](#)
whether or not (a condition) *phrase* [11.5](#)

which, in description clauses [6.9](#)
 which varies, in defining global variables [4.12](#)
 which -- asking which do you mean *activity* [18.31](#)
 while (a condition) *phrase* [11.9](#)
 while (condition)
 arbitrary conditions for rules [18.4](#), [19.7](#)
Whither? *example* [17.16 ex 321](#), [3.5 ex 321](#)
 who, in description clauses [6.9](#)
 whole number -- (real number) to the nearest whole number *phrase* [15.3](#)
Widget Enterprises *example* [15.18 ex 263](#), [9.4 ex 263](#)
 Wild, Campbell [1.2](#)
 Willis, Dannii [1.2](#)
 windows, extra story windows [12.5](#)
 winning the story [9.4](#), [18.38](#), [18.39](#), [11.6](#)
 with nouns reversed [17.2](#)
 within *relation verb* *see* [containment](#)
 without leading zeroes, when defining units [15.15](#)
 without printing a room description [8.7](#), [8.9](#)
Witnessed 1 *example* [9.5 ex 139](#), [10.7 ex 139](#)
Witnessed 2 *example* [11.17 ex 180](#), [9.11 ex 180](#)
 woman *kind* [3.17](#), [4.1](#)
 "[won't]" *say phrase* [14.8](#)
Wonderland *example* [15.8 ex 256](#), [6.9 ex 256](#)
Wonka's Revenge *example* [11.10 ex 175](#), [9.5 ex 175](#)
 word number (number) in (text) *phrase* [20.3](#)
The World of Charles S. Roberts *example* [3.26 ex 41](#), [3.2 ex 41](#)
 worn *relation verb* *see* [wearing](#)
 worn (thing) *adjective* [3.20](#), [8.13](#)
 "[would]" *say phrase* [14.7](#)
Would you...? *example* [4.8 ex 48](#), [9.1 ex 48](#)
 "[wouldn't]" *say phrase* [14.7](#)
 write (external file) from (table name) *phrase* [23.13](#)
 write (text) to (external file) *phrase* [23.14](#)
 writing a paragraph about *activity* [18.24](#), [3.1](#)
WXPQ *example* [18.35 ex 380](#), [6.17 ex 380](#)
X *Xerxes* *example* [18.39 ex 386](#), [11.6 ex 386](#)
Xot *example* [18.35 ex 381](#), [2.3 ex 381](#)
Xylan *example* [17.1 ex 288](#), [10.2 ex 288](#)
XYZZY *example* [17.1 ex 287](#), [6.13 ex 287](#)
XYZZY [3.1](#), [17.1](#), [20.5](#), [26.7](#), [26.10](#), [6.13](#)
Y *Y ask Y?* *example* [7.16 ex 111](#), [11.3 ex 111](#)
 yes *phrase* [11.16](#)
 YES, querying yes/no [11.5](#)
 YES, responding to a character [7.7](#)
Yolk of Gold *example* [6.16 ex 83](#), [8.4 ex 83](#)
 you, adaptive pronouns in displayed text [14.4](#)
Your Mother Doesn't Work Here *example* [21.11 ex 438](#), [7.12 ex 438](#)
 yourself — person *constant* [3.24](#), [6.13](#), [8.9](#)
Z *Z-machine* [2.14](#), [23.3](#)
 character set limits [20.2](#), [20.4](#)
 extensions for Z-machine only [27.7](#)
 memory limits [2.14](#), [20.2](#), [21.10](#)
 no support for multimedia [23.1](#), [12.5](#)
 numeric limits [11.10](#), [15.2](#), [15.10](#), [15.14](#)
 real numbers not available [15.2](#)
Zero *example* [9.12 ex 148](#), [5.5 ex 148](#)
Zodiac *example* [7.9 ex 96](#), [7.3 ex 96](#)
Zorb *example* [7.14 ex 108](#), [6.9 ex 108](#)
Zork II *example* [8.19 ex 134](#), [3.2 ex 134](#)
Zorn of Zorna *example* [18.18 ex 347](#), [3.7 ex 347](#)
Zqlran Era 8 *example* [15.15 ex 260](#), [4.1 ex 260](#)

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z