The `fancyqr` package

Florian Sihler

Version v2.2 – 2024/11/27

`fancyqr` is a simple package to create fancy qr-codes with the help of the *qrcode*-package. You can use the `\fancyqr`-macro just like the normal `\qrcode`.[1]

If you do want to hide a center square (e.g, because you want to embed an image) you can use `\FancyQrDoNotPrintSquare{<x>}{<y>}` to hide a rectangle with radius x and y set from the center (with `\FancyQrDoNotPrintRadius{<factor>}` you can apply a rounding to this!). If you choose this option, the default `\FancyQrRoundCut` that rounds cut corners can be changed with `\FancyQrHardCut`. At the moment, there are six other styles for the qr-code `flat`, `frame`, `blobs`, `glitch`, and `dots`, that you can load (locally) by using `\FancyQrLoad{<name>}`. The default style is named `default` and can be 'reset' by `\FancyQrLoad{default}` or `\FancyQrLoadDefault`.

All of the extra qr-options (you can set all of them with `\fancyqrset{<keys>}`) are showcased in Table 1. The defaults are set like this:

```
\fancyqrset{image padding=0,gradient=true,
        gradient angle=135,r color=teal,l color=purple}
```

Consider the following example (uses fontawesome for the symbol, but you can use include images, ...):

```
\fancyqr[image=\huge\faGithub,image padding=1,color=black
   !90!gray]{https://github.com/EagleoutIce/fancyqr}
```

Which produces:

[1] `\fancyqr[<qr-options>]{<url>}`

Option	Type	Default	Explanation
`classic`	boolean	`false`	Use the classic qr-code style (black with flat rectangles, this loads the `flat` style).
`color`	color		Disables the `gradient` and sets the color accordingly.
`compensate`	length	`0.15pt`	Compensating overlap to add to avoid glitches.
`gradient`	boolean	`true`	Toggle the color gradient
`gradient angle`	angle	`135`	Change the gradient angle.
`image`	LaTeX		Automatically center an image (you have to care for the size and maybe adjust the `version` and `level` to keep the qr-code readable).[2]
`image padding`	number		Additionally hide blocks (x & y) around the image.
`image x padding`	number	`0`	Additionally hide blocks (x) around the image.
`image y padding`	number	`0`	Additionally hide blocks (y) around the image.
`l color`	color	`purple`	Set the top left gradient color.
`left color`	color		Alias for `l color`.
`level`	L/M/Q/H	`M`	`qrcode` option affecting error correction (low, medium, quartile, high).
`padding`	flag		`qrcode` option adding sufficient additional space around the qr-code.
`r color`	color	`teal`	Set the bottom right gradient color.
`random color`	colors		Allow to set a random color pool to pick from.
`right color`	color		Alias for `r color`.
`size`	length		Alias for `qrcode`'s `height` option.
`tight`	flag		`qrcode` option adding no additional space around the qr-code.
`version`	$[0..40] \in \mathbb{N}$	`0`	`qrcode` option affecting the size (tries to be as small as possible).
`width`	length		Alias for `qrcode`'s `height` option.

Table 1: Overview of special qr-options.

[2]The package will automatically calculate the required `\FancyQrDoNotPrintSquare` (you have to make sure that the qr-code still has enough information to be readable). Therefore, the image will not scale with the qr-code.