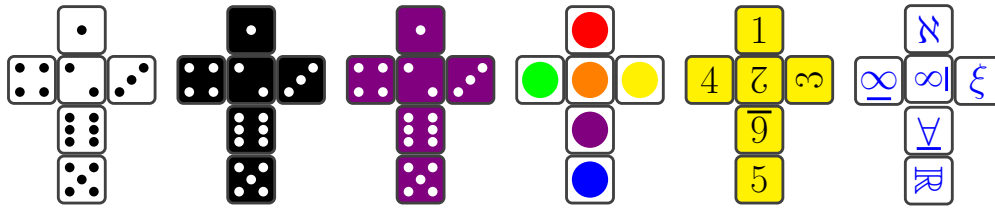


customdice v1.1

Peter Rowlett

August 10, 2022



`customdice` is a package for \LaTeX \LuaTeX and \XeTeX that provides functionality for drawing dice. The aim is to provide highly-customisable but simple-to-use commands, allowing:

- adding custom text to dice faces;
- control over colouring;
- control over sizing.

The package is licenced under Creative Commons Attribution-ShareAlike 4.0 International (CC BY-SA 4.0). To view a copy of this license, visit <http://creativecommons.org/licenses/by-sa/4.0/>

Contents

1	Motivation	2
2	Basic usage	2
2.1	Standard dice	2
2.2	Big dot dice	2
2.3	Text on dice	2
2.4	Layout dice	3
2.5	Colour	3
2.6	Size	4
3	Advanced usage notes	6
3.1	Dependencies	6
3.2	Rotated text	6
3.3	Size	6
3.3.1	Size and inline display	6
3.3.2	Changing the basic size	7
3.3.3	Changing all the sizes	8
3.3.4	Text doesn't fit?	9
3.4	Border colour	9
3.5	Arbitrary drawing on dice faces	9
4	Is it 'die' or 'dice'?	10

1 Motivation

I've long struggled to find a dice package I fully like. I have been using `epsdice`, but it only offers white or black standard dice and I found I wanted to draw dice with other text on the faces so I wrote this package which draws the dice in TikZ and offers lots of customisation around what is displayed on the dice face, colour and size.

2 Basic usage

2.1 Standard dice

Standard dice faces are created using `\dice{num}` where $\text{num} \in \{1, 2, 3, 4, 5, 6\}$.

For example

```
\dice{1} \dice{2} \dice{3} \dice{4} \dice{5} \dice{6}
```

produces output like this:



Apparently there are occasions where people desire dice faces as if from a six-sided cube that show 7, 8 or 9 dots. This is incorporated into `\dice` also.

For example

```
\dice{7} \dice{8} \dice{9}
```

produces output like this:



Also, inputting `\dice{0}` produces a blank face, like this:



2.2 Big dot dice

The command `\bigdotdice` produces a dice face with one big dot in the centre.

For example

```
\bigdotdice
```

produces output like this:



2.3 Text on dice

The command `\textdice{text}` produces a dice face with `text` on it.

For example

```
\textdice{7}  
\textdice{P}  
\textdice{\(\aleph\)}
```

produces output like this:



Sometimes when drawing text on dice, there may be ambiguity about which way up a character goes. For this reason a second command `\textdicebot{text}` is provided which draws a small line under `text` ('bot' is to indicate 'bottom line').


For example

```
\textdicebot{\(\cup\)}
\textdicebot{\(\cap\)}
\textdicebot{\sffamily W}
```

produces output like this:



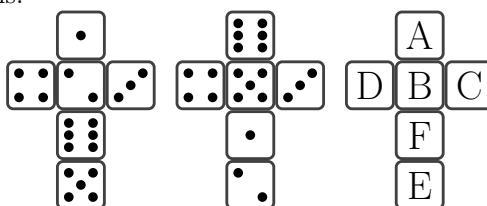
2.4 Layout dice

A command `\layoutdice{face1}{face2}{face3}{face4}{face5}{face6}` takes six inputs and places them on an expanded net of a dice cube. Important: this command takes six inputs in numerical order and displays those inputs where they would sit on a standard die with  oriented at the top of the diagram.

For example

```
\layoutdice{\dice{1}}{\dice{2}}{\dice{3}}{\dice{4}}{\dice{5}}{\dice{6}}
\layoutdice{\dice{6}}{\dice{5}}{\dice{3}}{\dice{4}}{\dice{2}}{\dice{1}}
\layoutdice{\textdice{A}}{\textdice{B}}{\textdice{C}}{\textdice{D}}{\textdice{E}}{\textdice{F}}
```

produces output like this:



2.5 Colour

By default, dice are black on white. This can be changed by passing a pair of colour names `[background,foreground]` to any of the commands above.

For example

```
\dice[black,white]{3}
\dice[violet,white]{5}
\dice[yellow,black]{6}
\bigdotdice[white,blue]
\textdice[gray,green]{\(\aleph\)}
\textdicebot[magenta,blue]{\sffamily W}
```

produces output like this:



Any defined colour name can be used, including ones you defined yourself. For example,

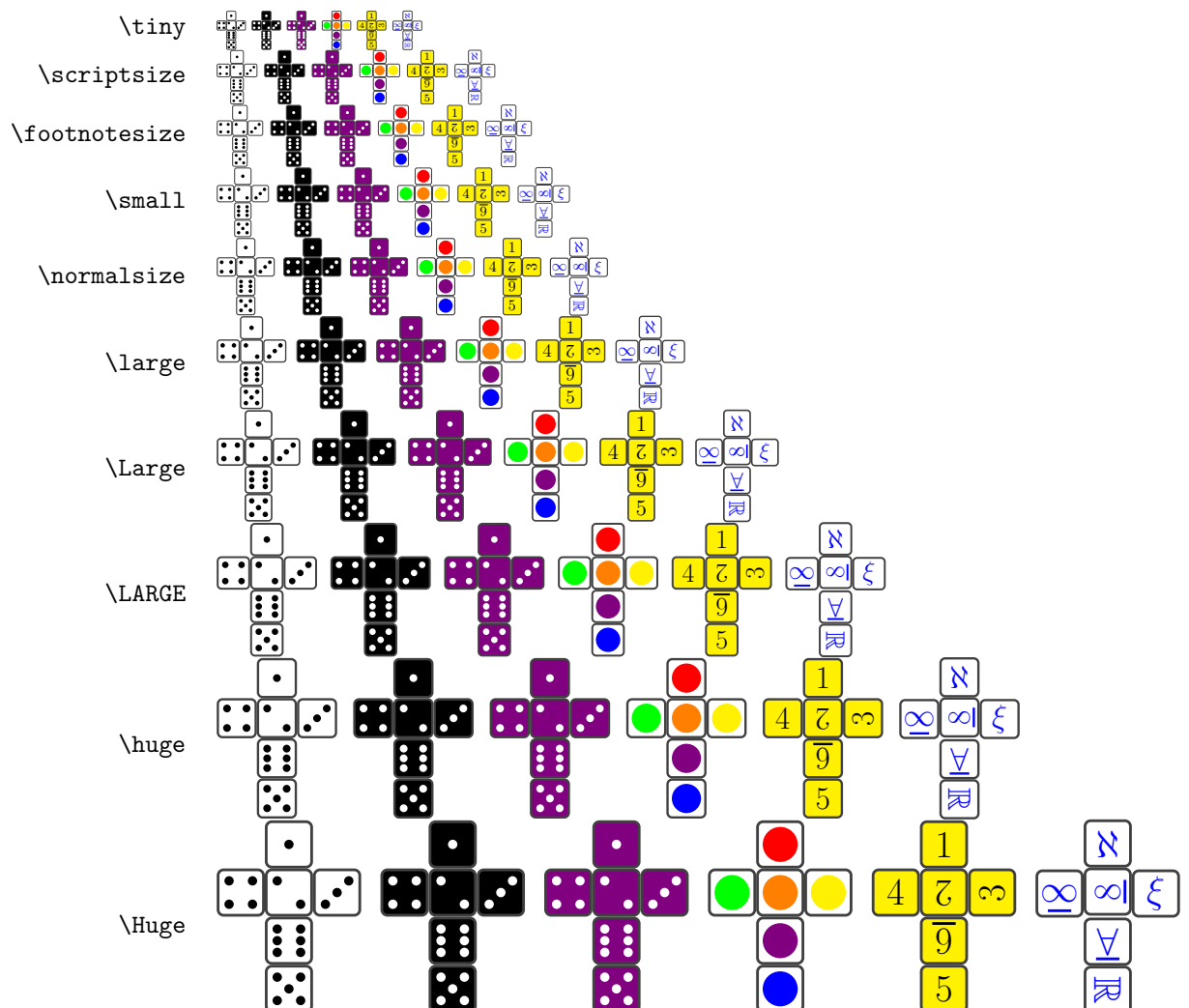
```
\usepackage{xcolor}
\definecolor{airforceblue}{rgb}{0.36, 0.54, 0.66} % latexcolor.com
\definecolor{papayawhip}{rgb}{1.0, 0.94, 0.84} % latexcolor.com
\dice[airforceblue,papayawhip]{5}
```

produces output like this:











































2.6 Size

The dice size responds to the current font size, as demonstrated below. (Actually, most of the example output in this document is done within `\Huge` for visibility.)



More detail on customising sizes is given in the advanced usage section [3.3](#).

For convenience, a series of commands are defined for each dice type based on the standard LaTeX size names. While e.g. `\dice{num}` uses the current font size, e.g. `\Largedice{num}` is always `\Large`.

Command	Example output
<code>\tinydice</code>	
<code>\scriptsize dice</code>	
<code>\footnotesize dice</code>	
<code>\small dice</code>	
<code>\normalsize dice</code>	
<code>\large dice</code>	
<code>\Largedice</code>	
<code>\LARGE dice</code>	
<code>\huge dice</code>	
<code>\Huge dice</code>	
<code>\tinybigdotdice</code>	
<code>\scriptsizebigdotdice</code>	
<code>\footnotesizebigdotdice</code>	
<code>\smallbigdotdice</code>	
<code>\normalsizebigdotdice</code>	
<code>\largebigdotdice</code>	
<code>\Largebigdotdice</code>	
<code>\LARGEbigdotdice</code>	
<code>\hugebigdotdice</code>	
<code>\Hugebigdotdice</code>	
<code>\tinytextdice</code>	
<code>\scriptsize textdice</code>	
<code>\footnotesize textdice</code>	
<code>\small textdice</code>	
<code>\normalsize textdice</code>	
<code>\large textdice</code>	
<code>\Largetextdice</code>	
<code>\LARGETextdice</code>	
<code>\hugetextdice</code>	
<code>\Hugetextdice</code>	
<code>\tinytextdicebot</code>	
<code>\scriptsize textdicebot</code>	
<code>\footnotesize textdicebot</code>	
<code>\small textdicebot</code>	
<code>\normalsize textdicebot</code>	
<code>\large textdicebot</code>	
<code>\Largetextdicebot</code>	
<code>\LARGETextdicebot</code>	
<code>\hugetextdicebot</code>	
<code>\Hugetextdicebot</code>	

3 Advanced usage notes

3.1 Dependencies

customdice uses TikZ and etoolbox.

3.2 Rotated text

You can rotate what is displayed on the face using `\rotatebox{}` from the `rotating` package.

For example

```
\usepackage{rotating}
\textdice[yellow,black]{\rotatebox{180}{2}}
\textdice[white,blue]{\rotatebox{90}{\(\mathbb{R}\)}}
```

produces output like this:



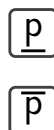
Sometimes you might need to rotate the whole box, for example when using `\dicetextbot`.

For example compare

```
\usepackage{rotating}
\textdicebot{\rotatebox{180}{\sffamily d}}

\rotatebox{180}{\textdicebot{\sffamily d}}
```

which produces output like this:



The first has rotated the ‘d’ but not the underline, so it appears to be a ‘P’.

3.3 Size

3.3.1 Size and inline display

The size of the die face is 1.65ex square, which is about the height of capital letters in whatever font size is currently being used. This should mean inline use works well, as demonstrated below.

<code>\tiny</code>	ONE	ONE
<code>\scriptsize</code>	TWO	TWO
<code>\footnotesize</code>	THREE	THREE
<code>\small</code>	FOUR	FOUR
<code>\normalsize</code>	FIVE	FIVE
<code>\large</code>	SIX	SIX
<code>\Large</code>	EIGHT	EIGHT
<code>\LARGE</code>	TEN	TEN
<code>\huge</code>	THREE	THREE
<code>\Huge</code>	FOUR	FOUR

You may find that you prefer your dice larger than the surrounding text, like this \boxed{A} is done using `\Largetextdice{A}`. The dice are designed to sit on the baseline of the current text size via a parameter called `\customdicebaseline` which is by default 0.02 and is passed to TikZ in `\begin{tikzpicture}[baseline=\customdicebaseline ex]`. If you enlarge the size of the dice to be bigger than the surrounding text you may find they appear to sit above the line. If

you prefer them to be centred vertically with the surrounding text you will need to adjust this parameter (by trial and error). Do this using `\setdicebaseline{num}`. For example, setting `\setdicebaseline{0.35}` should make `\Largetextdice{A}` appear centred vertically on the line of text again, like this A now does.

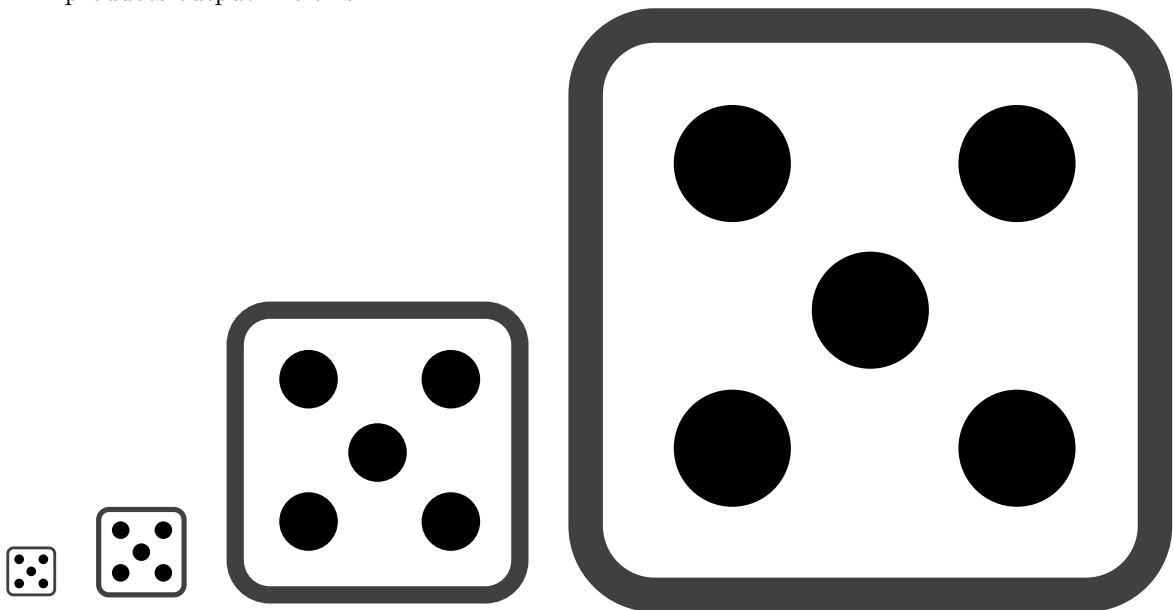
3.3.2 Changing the basic size

The size of the die face is 1.65ex square. All other sizes within the die face are defined in relation to this basic unit, and you can change it and them using `\setdicefacesize{num}` where `num` is a number that is interpreted as a number of ex.

For example,

```
\Huge
\dice{5}
\setdicefacesize{3}
\dice{5}
\setdicefacesize{10}
\dice{5}
\setdicefacesize{20}
\dice{5}
```

produces output like this:



3.3.3 Changing all the sizes

This is a list of all the sizes used by the package, set by the two commands `\setdicebaseline` and `\setdicefacesize`.

Parameter	Description	Units	Set by	Default
<code>\customdicebaseline</code>	Controls where the dice sits in relation to surrounding text.	ex	<code>\setdicebaseline</code>	0.02
<code>\customdicefacesize</code>	Side length of the dice face square.	ex	<code>\setdicefacesize</code>	1.65
<code>\customdicehalfway</code>	Half way across the dice face, distance from the edge to the centre dot on the standard dice face. Also the centre of the text in <code>\dicetext</code> and <code>\dicetextbot</code> .	ex	<code>\setdicefacesize</code>	0.825
<code>\customdicelower</code>	Distance from the edge to the ‘lower’ dots (at the bottom or the left of the standard dice face).	ex	<code>\setdicefacesize</code>	0.4
<code>\customdiceupper</code>	Distance from the edge to the ‘upper’ dots (at the top or the right of the standard dice face).	ex	<code>\setdicefacesize</code>	1.25
<code>\customdicedotsize</code>	Size of the standard dice dots.	ex	<code>\setdicefacesize</code>	0.12
<code>\customdicebigdotsize</code>	Size of the bigdot dice dots.	ex	<code>\setdicefacesize</code>	0.45
<code>\customdicecornerrounding</code>	Controls the rounding on the corners.	ex	<code>\setdicefacesize</code>	0.2
<code>\customdiceborderthickness</code>	Thickness of the dice face border.	ex	<code>\setdicefacesize</code>	0.1
<code>\customdicetextscale</code>	Scaling of the text on the dice faces in <code>\dicetext</code> and <code>\dicetextbot</code> .	none	<code>\setdicefacesize</code>	0.6

Note that `\textdicebot` draws an underline of thickness `\customdiceborderthickness` from `(\customdicelower,\customdicecornerrounding)` to `(\customdiceupper,\customdicecornerrounding)`.

These parameters are all commands, so if you have a good reason to adjust one of them you can do so using `\renewcommand`, for example `\renewcommand{\customdicehalfway}{1.1}` changes ^a to ^a. Note that calling `\setdicebaseline/\setdicefacesize` will reset the relevant commands.

3.3.4 Text doesn't fit?

There is nothing to stop text running off the side of the die. For example `\textdice{Hello}` just puts 'Hello' centred at $(\text{\customdicehalfway}, \text{\customdicehalfway})$, like this:



In some circumstances, it may be possible to squeeze some overflowing text onto the die face by adjusting `\customdicetextscale`. For example, `\renewcommand{\customdicetextscale}{0.4}` converts `\textdice{01}` to `\textdice{01}`.

It is possible to overdo this and make the text illegible, of course, like the 'Hello World' here: `\textdice{Hello World}`.

Note that calling `\setdicefacesize` will reset `\customdicetextscale` relative to the new `\customdicefacesize`.

3.4 Border colour

The background and foreground colours can be set by passing parameters to the commands as outlined above. The border of the dice face is set globally to `darkgray`, with the assumption that this will be fine with any colour combination. It is customisable though. You can set the global dice face border colour to any colour name using `\setdicefaceoutlinecol{name}` and restore the default using `\defaultdicefaceoutlinecol`.

For example,

```
\setdicefaceoutlinecol{red}\dice{5}
\setdicefaceoutlinecol{orange}\dice{5}
\setdicefaceoutlinecol{yellow}\dice{5}
\setdicefaceoutlinecol{green}\dice{5}
\setdicefaceoutlinecol{blue}\dice{5}
\setdicefaceoutlinecol{violet}\dice{5}
\defaultdicefaceoutlinecol\dice{5}
```

produces output like this:



3.5 Arbitrary drawing on dice faces

The commands `\dice`, `\bigdotdice`, `\textdice` and `\textdicedot` all start an environment called `customdiceenv` and do some drawing on it. `customdiceenv` is really a `tikzpicture` with baseline set to `\customdicebaseline ex` and the dice face outline drawn from $(0ex, 0ex)$ to $(\text{\customdicefacesize ex}, \text{\customdicefacesize ex})$. You can call `customdiceenv` directly and within this environment, any other TikZ commands should work.

For example

```
\usetikzlibrary{shapes.geometric}
\begin{customdiceenv}
\node[isosceles triangle,draw,inner sep=0.2ex] at (0.65ex,0.825ex) {};
\end{customdiceenv}
```

produces output like this:



`customdiceenv` has one optional parameter which is a pair of colours, defaulting to `white,black`. Note that what you draw isn't automatically coloured, though within `customdiceenv` you can access the foreground colour as `\customdicefg` and the background colour as `\customdicebg`.

You can also access other package parameters, such as the different sizes. To help stay within the confines of the dice face, it might be helpful to make use of `\customdiceupper`, `\customdicelower` and `\customdicehalfway`.

For example

```
\begin{customdiceenv}[pink,violet]
\draw[very thick,->,\customdicefg] (\customdiceupper ex,\customdicelower ex) --
(\customdicelower ex,\customdiceupper ex);
\end{customdiceenv}
```

produces output like this:



4 Is it ‘die’ or ‘dice’?

Gosh, I don’t know! Technically singular is ‘die’ and plural is ‘dice’, but common usage has typically drifted far from this and language evolves. I’m not sure I’ve used these terms correctly or consistently in this document. All commands use ‘dice’.