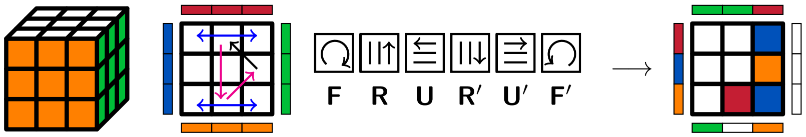
The rubikcube package

RWD Nickalls (dick@nickalls.org)
A Syropoulos (asyropoulos@yahoo.com)

This file describes version 2.0, last revised 2014/01/20

Abstract

The rubikcube package provides LaTeX commands and macros for typesetting Rubik cube (3x3x3) notation, configurations, and rotation sequences using the TikZ graphic language.



Contents

1 Introduction

The rubikcube package provides a collection of LaTeX commands and macros for typesetting Rubik cube configurations and layer-rotation instructions using the PGF/TikZ graphic languages—see examples below. We have extended slightly the layer-rotation hieroglyphic notation, originally developed by Garfath-Cox (1981), and improved by Duvoid (2010, 2011).

The rubikcube package has been road-tested on a Microsoft platform (with MikTeX and Strawberry Perl) and on a Linux platform (TEXLive and Mandriva).

Useful web sites for solvers are the Speedsolving website, and those maintained by Duvoid, by Fridrich, by Jelinek, by Reid, and by Vandenburgh (see References). For the mathematics and group theory associated with Rubik's cube see Chen (2004), Joyner (2008), Hutchings (2011), Rokicki *et al.* (2013).

1.1 Requirements

The rubikcube package requires the TikZ package.

1.2 Supporting tool—the **rubikrotation** package

The rubikrotation package, is a dynamic extension to the rubikcube package. It consists of the Perl program `rubikrotation.pl` and style option `rubikrotation.sty`. The rubikrotation package implements rotation sequences and random scrambling of the 3x3x3 Rubik cube on-the-fly using a `\RubikRotation{`⟨*rotation-sequence*⟩`}` command. It returns the new state in a form which is then used by the rubikcube package.

Since the `\RubikRotation` command works by CALLing the `rubikrotation.pl` program, it follows that the rubikrotation package requires (a) Perl to be installed, and (b) LaTeX needs to be run using the `--shell-escape` commandline option. See the rubikrotation documentation for details. The rubikrotation package has been road-tested on a Microsoft platform (with MikTeX and Strawberry Perl) and on a Linux platform (TEXLive and Mandriva).

1.3 Copyright

2 Installation

2.1 Generating the files

Place the file `rubikcube.zip` into a temporary directory, and unzip it. This will generate the following files:

```
rubikcube.ins
rubikcube.dtx
rubikcube.sty
rubikcube.pdf
example-cube.tex
example-cube.pdf
Rubik_doc_figA.pdf
Rubik_doc_figB.pdf
Rubik_doc_figC.pdf
Rubik_doc_figD.pdf
Rubik_doc_figE.pdf
```

The style option `rubikcube.sty` is generated by running (pdf)LaTeX on the file `rubikcube.ins` as follows:

```
  pdflatex  rubikcube.ins
```

The documentation file (`rubikcube.pdf`) was generated using the following steps:

```
 pdflatex   rubikcube.dtx
 pdflatex   rubikcube.dtx
 makeindex -s gind.ist  rubikcube
 makeindex -s gglo.ist -o rubikcube.gls  rubikcube.glo
 pdflatex   rubikcube.dtx
```

2.2 Placing the files

Place the files either in the local working directory, or where your system will find them. For a Linux system with a standard TeX Directory Structure (TDS), then:

*.sty → `/usr/local/texlive/texmf-local/tex/latex/local/rubikcube/`
*.pdf → `/usr/local/texlive/texmf-local/doc/rubikcube/`

Finally, (depending on your system) update the TeX file database using the `texhash` command.

2.3 Usage

Load the package by using the command `\usepackage{rubikcube}`. Note that the rubikcube package requires the TikZ package, and so always load TikZ before rubikcube as follows:

```
\usepackage{tikz}
\usepackage{rubikcube}
```

However, the rubikcube package does check for the presence of TikZ, and will load it if TikZ is not already loaded.

3 TikZ picture environment

For a basic introduction to the use of TikZ see the following manuals (from CTAN or from `http://altermundus.com/`).

- `pgfmanual.pdf`, version 2.10 (2012) (900 pages)
- tkz-base-screen.pdf

An example of the TikZ picture environment for use with rubikcube is as follows If no scale is used (default scale = 1), then each of the small cubie sides will have a length of 1 cm.

```
\begin{tikzpicture}[scale=0.5]
....
\end{tikzpicture}
```

A very useful feature of TikZ is that it automatically minimises the surrounding white-space, and consequently this is mostly quite sufficient.

However, when making complicated side-by-side figures it is sometimes helpful to place each sub-figure in a surrounding minipage. In this case, a useful approach is to first adjust the 'scale' parameter to obtain the appropriate size, and then adjust the minipage width as necessary.

Note that sometimes a temporary `\fbox{}` can be a useful aid when trying to visualise the full extent of the minipage (and its whitespace). In this case remember to include a trailing % after the first brace { of the fbox command in order not to collect any extra 'space', as follows:

```
\fbox{%
   \begin{minipage}{2cm}
   \centering
   \begin{tikzpicture}[scale=0.5]
   ....
   \end{tikzpicture}
   \end{minipage}
}
```

Unexpected spacing between two adjacent images, or between an image and adjacent text, is usually related to inappropriate (and hidden) white-space associated with the image or its associated `\minipage`.

3.1 rubikcube commands

Although the rubikcube package has been designed with TikZ in mind, it is important to appreciate that of all the various rubikcube commands only the 'Draw...' commands actually have to be used within a TikZ picture environment.

Indeed, using rubikcube commands outside the TikZ picture environment can make for useful flexibility when a documant is generating more than one figure or image. This is because the scope of a command when used inside a TikZ environment is constrained to be ‘local’ to that particular environment, and hence its influence is not accessable outside the environment.

For example,

3.2 \draw error message

If a \Draw... command is used *outside* a TikZ picture environment, then LaTeX issues a \draw “Undefined control sequence” error message, indicating that it is trying to draw something using an undefined TikZ \draw command.

For example, if we try to run the rubikcube command \DrawRubikFlat without a surrounding TikZ picture environment you will see something like the following error message.

```
! Undefined control sequence.
\FlatUp ... }{#1}\pgfmathsetmacro {\uy }{#2}\draw
                                                  [line join=round,...
l.56 \DrawRubikFlat
```

4 Command conventions

\rubikcube This command generates the logo rubikcube.

4.1 Capital letters

Each ‘word’ in a command (except the word ‘text’) starts with a capital letter. For example, \DrawRubikCubeRU, \DrawCubieRU. However, in line with LaTeX, ‘text’ commands start with a lowercase ‘t’; for example \textCubieRU. Letters for colours (R, O, Y, G, B, W, X) are always written in uppercase. How the six face letters are written depends on the command; sometimes in uppercase (eg \rrD, \rrDs, \rrDa), and sometimes in lowercase (eg \rrSu, \rrd, \rrhf).

4.2 XYZ argument ordering

Many commands have an appended two (XY) or three (XYZ) ordered letter code which is used to for specifying some feature of the command; perhaps either face or cubie colours or a viewpoint direction.

The convention is that the letter codes are ordered in the XYZ order; ie the first code relates to an X-related parameter; for example L (Left) or R (Right); the second relates to a Y-related parameter; for example U (Up) or D (Down); the third (if required) relates to a Z-related parameter; for example F (Front) or B (Back).

EXAMPLE: \DrawCubieRU{G}{Y}{O} draws a cubie from the RightUp viewpoint. The sequence of colour codes for the three visible faces are XYZ ordered, and hence result in the cube having a Green Right face, Yellow Up face and Orange Front face.

4.3 Trailing % on end of rubikcube commands

It is important to include a trailing % on the end of rubikcube commands when used *outside* a TikZ picture environment. This is to prevent unwanted 'space' characters appearing in the graphics. Note that a useful feature of the TikZ environment is that it automatically minimises any such space.

5 Rubik rotation commands and notation

We use the standard Rubik cube notation of WCA (2012)—see article 12—and also the 's' (Slice) and 'a' (anti-Slice) notation described in the 'Notation and terminology' section in the 'Pretty patterns' page on the website of Fridrich (`http://www.ws.binghamton.edu/fridrich/ptrns.html`). By the term rubikcube we denote only the familiar 3x3x3 puzzle cube. We denote similar puzzle cubes of a different size by the term NCubes (NxNxN).

It is recommended that commas are used to separate sequestial Rubik moves or commands to avoid ambiguity, especially when using just lettercodes on their own. For example, in the following sequence the commas usefully remove any ambiguity: **U**,**Lw**2,**Us**′,**Da** (`\rrU,\rrLw2,\rrUsp,\rrDa`).

5.1 Overview

The rubikcube notation comprises a range of commands for moves or rotations (e.g., **R**, **y**, **Bw**) and their equivalent hieroglyphs (e.g., $\boxed{||\uparrow}$, [**y**] , $\boxed{\mathsf{Bw}}$), as well as commands for drawing 3x3x3 cubes and single cubies.

Note that there are a few rotation commands which do not have arrow hieroglyphs—their their rotation is not visible from the FRONT face and hence cannot easily be rendered as an arrow hieroglyph. Consequently these rotations have a simple 'letter' hieroglyph in the form of the letter-code in a square; for example $\boxed{\mathsf{Bw}}$, $\boxed{\mathsf{Sb}}$.

`\rr..` The letter-code of a rotation is typeset (as in text) using the rubik-rotation `\rr..` stem: ie **R** is typeset using the command `\rrR`. The hieroglyph of a rotation

`\rrh..` command is generated (in text) by using instead the stem `\rrh..`. Thus the command `\rrhR` generates $\boxed{||\uparrow}$ which is the hieroglyph associated with **R**.

`\Rubik..` A vertically combined lettercode and its hieroglyph is generated using the command stem `\Rubik..`. For example, $\underset{\mathsf{R}}{\boxed{||\uparrow}}$ is generated by the command `\RubikR`, with the square hieroglyph sitting on the baseline. For some hieroglyphs, eg the [**x**] , [**y**] , [**z**] denoting 90 degree axis rotations, the only difference between the `\rrh..` and `\Rubik..` forms is that the `\Rubik..` form is eleveted to sit on the baseline just like the other `\Rubik..` hieroglyphs. For example `\rrhyp` generates [**y**′] , while `\Rubikyp` generates $^{[\mathbf{y}']}$.

`\textRubik..` A horizontally combined lettercode and its hieroglyph (in sequence as in text) is generated using the command stem `\textRubik..`. For example, **R**$\boxed{||\uparrow}$ is typeset using the command \textRubikR. A list of all commands and their associated hieroglyphs is given in Section 5.7.

5.2 Face rotations

\rrU
\rrD
\rrL
\rrR
\rrF
\rrB
\rrUp
\rrDp
\rrLp
\rrRp
\rrFp
\rrBp

The six main faces of the cube are denoted as FRONT (towards the observer), BACK, LEFT, RIGHT, UP, DOWN. The uppercase initial letter of each face-name (**F**, **B**, **L**, **R**, **U**, **D**) denotes a clockwise 90-degree rotation of the face as shown in Figure 1. For example, **D** is generated by the 'rubik rotation' command \rrD.

An appended prime ′ indicates an anticlockwise rotation; eg **F**′. This is sometimes written as $\mathbf{F}^{-1}$. The 'prime' notation is achieved by appending a lowercase 'p' to the face rotation command. For example, **R**′ is generated by \rrRp.

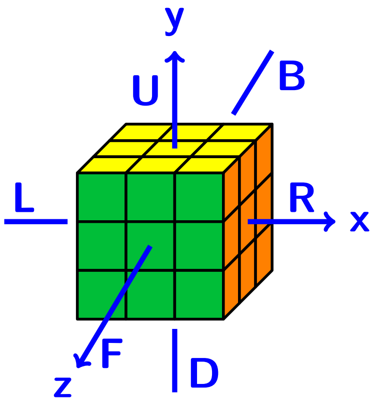


Figure 1:

5.3 Slice rotations

\rrSu
\rrSd
\rrSl
\rrSr
\rrSf
\rrSb
\rrSup
\rrSdp
\rrSlp
\rrSrp
\rrSfp
\rrSbp

The Rubik cube (3x3x3) has three orthogonal so-called 'inner' slices, whose +ve rotation direction follows that of a named face. For example the inner slice rotation between the RIGHT and LEFT faces whose rotation direction follows the rotation **R** (rotation is isomorphic to **R**) is denoted as **Sr**, which is typeset using the command \rrSr. Note that in these cases the trailing r in the command is lowercase.

Each of these slice rotations (S rotations) has a reversed (primed) p-form, the command for which is generated by appending the suffix 'p'. For example the inner slice rotations **Sl**′ (\rrSlp) and **Sr** (\rrSr) are identical. The equivalence is more obvious when we see their respective hieroglyphs. For example, in this case **Sl**′ (\rrSlp) ≡ $\boxed{|\uparrow|}$ (\rrhSlP), and **Sr** (\rrSr) ≡ $\boxed{|\uparrow|}$ (\rrhSr).

MES slice notation

`\rrM` `\rrE` `\rrS` `\rrMp` `\rrEp` `\rrSp`

An alternative and somewhat confusing (and hence is non-standard) slice notation which is sometimes used is the following so-called MES notation, as used in the Waterman algorithm (Treep and Waterman 1987).

M (MIDDLE slice, between the LEFT and RIGHT faces; direction follows **L**),

E (EQUATOR slice, between the UP and DOWN faces; direction follows **D**),

S (STANDING slice, between the FRONT and BACK faces; direction follows **F**).

Each of these also has a reversed (prime) version. The equivalent S notations are therefore as follows: **E** $\equiv$ **Sd**, **E′** $\equiv$ **Su**, **M** $\equiv$ **Sl**, **M′** $\equiv$ **Sr**, **S** $\equiv$ **Sf**, **S′** $\equiv$ **Sb**.

Singmaster slice notation

`\rrUs` `\rrDs` `\rrLs` `\rrRs` `\rrFs` `\rrBs`

These are an alternative (but somewhat less intuitive) form of slice notation which can be thought of as complementing the inner slice rotations. These were originally described by Singmaster (Frey and Singmaster, 1982). See the link to ‘notation’ on the ‘Pretty patterns’ page of the Fridrich website.

Each of these commands denotes a rotation of two opposite faces in the same direction. For example, **Us**$\boxed{\begin{smallmatrix}\leftarrow\\ -\\ \leftarrow\end{smallmatrix}}$ $\equiv$ **U**$\boxed{\begin{smallmatrix}\leftarrow\\ -\\ -\end{smallmatrix}}$ $+$ **D′**$\boxed{\begin{smallmatrix}-\\ -\\ \leftarrow\end{smallmatrix}}$, which is typeset as: `\textRubikUs\ $\equiv$ \textRubikU\ + \textRubikDp`, i.e., for **Us** both face-rotations are in the *same* direction as **U**.

Anti-slice notation

`\rrUa` `\rrDa` `\rrLa` `\rrRa` `\rrFa` `\rrBa`

Each of these commands denotes a rotation of two opposite faces in *opposite* directions. For example, **Ua**$\boxed{\begin{smallmatrix}\leftarrow\\ -\\ \rightarrow\end{smallmatrix}}$ $\equiv$ **U**$\boxed{\begin{smallmatrix}\leftarrow\\ -\\ -\end{smallmatrix}}$ $+$ **D**$\boxed{\begin{smallmatrix}-\\ -\\ \rightarrow\end{smallmatrix}}$, which is typeset as: `\textRubikUa\ $\equiv$ \textRubikU\ + \textRubikD`. See the link to ‘notation’ on the ‘Pretty patterns’ page of the Fridrich website.

5.4 Wide rotations

`\rrUw` `\rrDw` `\rrLw` `\rrRw` `\rrFw` `\rrBw`

The clockwise *combined* rotation of an outer face AND its inner slice (officially known as a ‘double outer slice’ rotation) is denoted by appending a lowercase **w** (denoting ‘wide’) to a face rotation command. For example, a RIGHT double outer slice rotation is denoted as **Rw**. Similarly, the prime ′ version **Lw′** is generated by `\rrLwp`.

The superscript ^{2}, or sometimes just an ordinary 2, indicates that the rotation is applied twice. For example **R**^{2} or **R2** denote *two* successive 90 degree clockwise rotations of the RIGHT face.

5.5 Axis rotations

`\rrx` `\rry` `\rrz` `\rrhx` `\rrhy` `\rrhz`

Whole-cube clockwise rotations of 90-degrees about about the orthogonal axes centred on the RIGHT, UP, FRONT faces are denoted as **x**, **y**, **z** (the `\rr..` forms) respectively, with their hieroglyphs (the `\rrh..` forms) being denoted as [**x**], [**y**], [**z**] in order to distinguish them from square layer-rotation hieroglyphs. Note that **x**, **y**, **z** rotations are always expressed in lowercase, and hence this practice is extended also to the commands.

An **x2** rotation (two **x** rotations one after the other, ie [**x**] [**x**]) denotes rotating the cube 180 degrees about its x axis so as to bring the DOWN face into the UP position.

An appended prime ′ indicates an anticlockwise rotation; for example, **x**′ (which is generated by appending a 'p' to the end of the command, ie `\rrxp`).

`\Rubikx` `\Rubiky` `\Rubikz` These `\Rubik..` forms (and their prime 'p' versions) generate the same hieroglyphs as their `\rrh..` versions, except that their spacing is similar to that associated with the 'square box' `\Rubik..` hieroglyphs. Consequently when typesetting an axis command in a sequence of 'square-box' `\Rubik..` commands, it is better to use the `\Rubik..` form rather than the equivalent `\rrh..` form (see the examples in Section 5.6). There are no `\textRubik..` forms for the axis commands (since they are not necessary).

The u, d, l, r, f, b notation

`\rru` A commonly used alternative for the **x**, **y**, **z** notation (and endorsed by the WCA)
`\rrd` uses the lowercase face letter to denote a 90 degree whole-cube rotation in the
`\rrl` same sense as that of the standard face rotations.
`\rrr` Thus **u** ≡ **y**, **d** ≡ **y**′, **l** ≡ **x**′, **r** ≡ **x**, **f** ≡ **z**, **b** ≡ **z**′,
`\rrf` For example, **d** is generated by the command `\rrd`.
`\rrb` Note that these moves do not have prime ′ versions since **u** is the opposite of **d**, **l** is the opposite of **r**, and **f** is the opposite of **b**.

`\rrh..` `\Rubik..` As with the [**x**], [**y**], [**z**] forms (described above) there also equivalent `\rrh..` and `\Rubik..` forms. For example, [**d**] is generated by the command `\rrhd`.

5.6 Examples

R is generated by the 'rubik rotation' command `\rrR`

Fw is generated by the 'rubik rotation' command `\rrFw`

L^2 is generated by `\rrL$^2$`

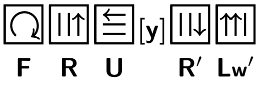
R′ is generated by `\rrRp`

Fw′ is generated by `\rrFwp`

x and [**y**] and [**z**′] are generated by `\rrx` and `\rrhy` and `\Rubikzp`

f and [**b**] are generated by `\rrf` and `\rrhb`

The sequence **UURR** is generated by `\rrU\rrU\rrR\rrR`



`\RubikF\RubikR\RubikU\Rubiky\RubikRp\RubikLwp`

The equivalent 'text' version of the first three rotations above (and their commands) are as follows:

F$\boxed{\circlearrowright}$ **R**$\boxed{|\,|\uparrow}$ **U**$\boxed{\substack{\leftarrow\\-\\-}}$ `\textRubikF\ \textRubikR\ \textRubikU`

Writing the first sequence using only letter-codes or only hieroglyphs, we have:

FRUyR′ `\rrF\rrR\rrU\rry\rrRp`

$\boxed{\circlearrowright}\boxed{|\,|\uparrow}\boxed{\substack{\leftarrow\\-\\-}}$[**y**] $\boxed{|\,|\downarrow}$ `\rrhF\rrhR\rrhU\rrhy\rrhRp\`

The following shows the importance of using commas to avoid ambiguity:

D,**U**2,**F**2,**Ds**2,**B**, `\rrD,\rrU2,\rrF2,\rrDs2,\rrB,`

$\boxed{\substack{-\\-\\\rightarrow}}$,$\boxed{\substack{\leftarrow\\-\\-}}$2,$\boxed{\circlearrowright}$2,$\boxed{\substack{\rightarrow\\-\\\rightarrow}}$2,$\boxed{\textbf{B}}$, `\rrhD,\rrhU2,\rrhF2,\rrhDs2,\rrhB,`

5.7 Listing of all rotation commands

Note that all these commands also have a `\Rubik..` equivalent form which typesets both the hieroglyph and its lettercode in a vertical format, as shown in the above 'Examples' section. These have been ommitted here owing to the difficulty of including this form easily in the following table.

Note also that some `\rrh..` commands (eg the `\rrhB..` commands) show only the lettercode in a square box. This is because these rotations, which do not have a 'true' visual representation as seen from the FRONT face, are somewhat ambiguous unless typeset with the associated lettercode (ie using the `\Rtbik..` or `\textRubik..` commands).

U	`\rrU`	$\boxed{\substack{\leftarrow\\-\\-}}$	`\rrhU`	**U**$\boxed{\substack{\leftarrow\\-\\-}}$	`\textRubikU`
U′	`\rrUp`	$\boxed{\substack{\rightarrow\\-\\-}}$	`\rrhUp`	**U′**$\boxed{\substack{\rightarrow\\-\\-}}$	`\textRubikUp`
Uw	`\rrUw`	$\boxed{\substack{\leftarrow\\\leftarrow\\-}}$	`\rrhUw`	**Uw**$\boxed{\substack{\leftarrow\\\leftarrow\\-}}$	`\textRubikUw`
Uw′	`\rrUwp`	$\boxed{\substack{\rightarrow\\\rightarrow\\-}}$	`\rrhUwp`	**Uw′**$\boxed{\substack{\rightarrow\\\rightarrow\\-}}$	`\textRubikUwp`
Us	`\rrUs`	$\boxed{\substack{\leftarrow\\-\\\leftarrow}}$	`\rrhUs`	**Us**$\boxed{\substack{\leftarrow\\-\\\leftarrow}}$	`\textRubikUs`
Us′	`\rrUsp`	$\boxed{\substack{\rightarrow\\-\\\rightarrow}}$	`\rrhUsp`	**Us′**$\boxed{\substack{\rightarrow\\-\\\rightarrow}}$	`\textRubikUsp`
Ua	`\rrUa`	$\boxed{\substack{\leftarrow\\-\\\rightarrow}}$	`\rrhUa`	**Ua**$\boxed{\substack{\leftarrow\\-\\\rightarrow}}$	`\textRubikUa`
Ua′	`\rrUap`	$\boxed{\substack{\rightarrow\\-\\\leftarrow}}$	`\rrhUap`	**Ua′**$\boxed{\substack{\rightarrow\\-\\\leftarrow}}$	`\textRubikUap`

D	\rrD	$\boxed{\begin{smallmatrix}-\\-\\\rightarrow\end{smallmatrix}}$	\rrhD	D $\boxed{\begin{smallmatrix}-\\-\\\rightarrow\end{smallmatrix}}$	\textRubikD
D′	\rrDp	$\boxed{\begin{smallmatrix}-\\-\\\leftarrow\end{smallmatrix}}$	\rrhDp	D′ $\boxed{\begin{smallmatrix}-\\-\\\leftarrow\end{smallmatrix}}$	\textRubikDp
Dw	\rrDw	$\boxed{\begin{smallmatrix}-\\\rightarrow\\\rightarrow\end{smallmatrix}}$	\rrhDw	Dw $\boxed{\begin{smallmatrix}-\\\rightarrow\\\rightarrow\end{smallmatrix}}$	\textRubikDw
Dw′	\rrDwp	$\boxed{\begin{smallmatrix}-\\\leftarrow\\\leftarrow\end{smallmatrix}}$	\rrhDwp	Dw′ $\boxed{\begin{smallmatrix}-\\\leftarrow\\\leftarrow\end{smallmatrix}}$	\textRubikDwp
Ds	\rrDs	$\boxed{\begin{smallmatrix}\rightarrow\\-\\\rightarrow\end{smallmatrix}}$	\rrhDs	Ds $\boxed{\begin{smallmatrix}\rightarrow\\-\\\rightarrow\end{smallmatrix}}$	\textRubikDs
Ds′	\rrDsp	$\boxed{\begin{smallmatrix}\leftarrow\\-\\\leftarrow\end{smallmatrix}}$	\rrhDsp	Ds′ $\boxed{\begin{smallmatrix}\leftarrow\\-\\\leftarrow\end{smallmatrix}}$	\textRubikDsp
Da	\rrDa	$\boxed{\begin{smallmatrix}\leftarrow\\-\\\rightarrow\end{smallmatrix}}$	\rrhDa	Da $\boxed{\begin{smallmatrix}\leftarrow\\-\\\rightarrow\end{smallmatrix}}$	\textRubikDa
Da′	\rrDap	$\boxed{\begin{smallmatrix}\rightarrow\\-\\\leftarrow\end{smallmatrix}}$	\rrhDap	Da′ $\boxed{\begin{smallmatrix}\rightarrow\\-\\\leftarrow\end{smallmatrix}}$	\textRubikDap
L	\rrL	$\boxed{\downarrow\|\|}$	\rrhL	L $\boxed{\downarrow\|\|}$	\textRubikL
L′	\rrLp	$\boxed{\uparrow\|\|}$	\rrhLp	L′ $\boxed{\uparrow\|\|}$	\textRubikLp
Lw	\rrLw	$\boxed{\downarrow\downarrow\|}$	\rrhLw	Lw $\boxed{\downarrow\downarrow\|}$	\textRubikLw
Lw′	\rrLwp	$\boxed{\uparrow\uparrow\|}$	\rrhLwp	Lw′ $\boxed{\uparrow\uparrow\|}$	\textRubikLwp
Ls	\rrLs	$\boxed{\downarrow\|\downarrow}$	\rrhLs	Ls $\boxed{\downarrow\|\downarrow}$	\textRubikLs
Ls′	\rrLsp	$\boxed{\uparrow\|\uparrow}$	\rrhLsp	Ls′ $\boxed{\uparrow\|\uparrow}$	\textRubikLsp
La	\rrLa	$\boxed{\downarrow\|\uparrow}$	\rrhLa	La $\boxed{\downarrow\|\uparrow}$	\textRubikLa
La′	\rrLap	$\boxed{\uparrow\|\downarrow}$	\rrhLap	La′ $\boxed{\uparrow\|\downarrow}$	\textRubikLap
R	\rrR	$\boxed{\|\|\uparrow}$	\rrhR	R $\boxed{\|\|\uparrow}$	\textRubikR
R′	\rrRp	$\boxed{\|\|\downarrow}$	\rrhRp	R′ $\boxed{\|\|\downarrow}$	\textRubikRp
Rw	\rrRw	$\boxed{\|\uparrow\uparrow}$	\rrhRw	Rw $\boxed{\|\uparrow\uparrow}$	\textRubikRw
Rw′	\rrRwp	$\boxed{\|\downarrow\downarrow}$	\rrhRwp	Rw′ $\boxed{\|\downarrow\downarrow}$	\textRubikRwp
Rs	\rrRs	$\boxed{\uparrow\|\uparrow}$	\rrhRs	Rs $\boxed{\uparrow\|\uparrow}$	\textRubikRs
Rs′	\rrRsp	$\boxed{\downarrow\|\downarrow}$	\rrhRsp	Rs′ $\boxed{\downarrow\|\downarrow}$	\textRubikRsp
Ra	\rrRa	$\boxed{\downarrow\|\uparrow}$	\rrhRa	Ra $\boxed{\downarrow\|\uparrow}$	\textRubikRa

Ra′ `\rrRap`	↑|↓ `\rrhRap`	Ra′ ↑|↓ `\textRubikRap`
F `\rrF`	↻ `\rrhF`	F ↻ `\textRubikF`
F′ `\rrFp`	↺ `\rrhFp`	F′ ↺ `\textRubikFp`
Fw `\rrFw`	⟳ `\rrhFw`	Fw ⟳ `\textRubikFw`
Fw′ `\rrFwp`	⟲ `\rrhFwp`	Fw′ ⟲ `\textRubikFwp`
Fs `\rrFs`	$\boxed{\text{Fs}}$ `\rrhFs`	$\boxed{\text{Fs}}$ `\textRubikFs`
Fs′ `\rrFsp`	$\boxed{\text{Fs}'}$ `\rrhFsp`	$\boxed{\text{Fs}'}$ `\textRubikFsp`
Fa `\rrFa`	$\boxed{\text{Fa}}$ `\rrhFa`	$\boxed{\text{Fa}}$ `\textRubikFa`
Fa′ `\rrFap`	$\boxed{\text{Fa}'}$ `\rrhFap`	$\boxed{\text{Fa}'}$ `\textRubikFap`
B `\rrB`	$\boxed{\text{B}}$ `\rrhB`	$\boxed{\text{B}}$ `\textRubikB`
B′ `\rrBp`	$\boxed{\text{B}'}$ `\rrhBp`	$\boxed{\text{B}'}$ `\textRubikBp`
Bw `\rrBw`	$\boxed{\text{Bw}}$ `\rrhBw`	$\boxed{\text{Bw}}$ `\textRubikBw`
Bw′ `\rrBwp`	$\boxed{\text{Bw}'}$ `\rrhBwp`	$\boxed{\text{Bw}'}$ `\textRubikBwp`
Bs `\rrBs`	$\boxed{\text{Bs}}$ `\rrhBs`	$\boxed{\text{Bs}}$ `\textRubikBs`
Bs′ `\rrBsp`	$\boxed{\text{Bs}'}$ `\rrhBsp`	$\boxed{\text{Bs}'}$ `\textRubikBsp`
Ba `\rrBa`	$\boxed{\text{Ba}}$ `\rrhBa`	$\boxed{\text{Ba}}$ `\textRubikBa`
Ba′ `\rrBap`	$\boxed{\text{Ba}'}$ `\rrhBap`	$\boxed{\text{Ba}'}$ `\textRubikBap`
Su `\rrSu`	⇇ `\rrhSu`	Su ⇇ `\textRubikSu`
Su′ `\rrSup`	⇉ `\rrhSup`	Su′ ⇉ `\textRubikSup`
Sd `\rrSd`	⇉ `\rrhSd`	Sd ⇉ `\textRubikSd`
Sd′ `\rrSdp`	⇇ `\rrhSdp`	Sd′ ⇇ `\textRubikSdp`
Sl `\rrSl`	|↓| `\rrhSl`	Sl |↓| `\textRubikSl`
Sl′ `\rrSlp`	|↑| `\rrhSlp`	Sl′ |↑| `\textRubikSlp`
Sr `\rrSr`	|↑| `\rrhSr`	Sr |↑| `\textRubikSr`

Sr′	`\rrSrp`	$\boxed{\mid\downarrow\mid}$	`\rrhSrp`	Sr′ $\boxed{\mid\downarrow\mid}$	`\textRubikSrp`
Sf	`\rrSf`	$\boxed{\text{Sf}}$	`\rrhSf`	$\boxed{\text{Sf}}$	`\textRubikSf`
Sf′	`\rrSfp`	$\boxed{\text{Sf}'}$	`\rrhSfp`	$\boxed{\text{Sf}'}$	`\textRubikSfp`
Sb	`\rrSb`	$\boxed{\text{Sb}}$	`\rrhSb`	$\boxed{\text{Sb}}$	`\textRubikSb`
Sb′	`\rrSbp`	$\boxed{\text{Sb}'}$	`\rrhSbp`	$\boxed{\text{Sb}'}$	`\textRubikSbp`
E	`\rrE`	$\boxed{\overline{\underline{\rightarrow}}}$	`\rrhE`	E $\boxed{\overline{\underline{\rightarrow}}}$	`\textRubikE`
E′	`\rrEp`	$\boxed{\overline{\underline{\leftarrow}}}$	`\rrhEp`	E′ $\boxed{\overline{\underline{\leftarrow}}}$	`\textRubikEp`
M	`\rrM`	$\boxed{\mid\downarrow\mid}$	`\rrhM`	M $\boxed{\mid\downarrow\mid}$	`\textRubikM`
M′	`\rrMp`	$\boxed{\mid\uparrow\mid}$	`\rrhMp`	M′ $\boxed{\mid\uparrow\mid}$	`\textRubikMp`
S	`\rrS`	$\boxed{\text{S}}$	`\rrhS`	$\boxed{\text{S}}$	`\textRubikS`
S′	`\rrSp`	$\boxed{\text{S}'}$	`\rrhSp`	$\boxed{\text{S}'}$	`\textRubikSp`
x	`\rrx`	[x]	`\rrhx`	[x]	`\Rubikx`
x′	`\rrxp`	[x′]	`\rrhxp`	[x′]	`\Rubikxp`
y	`\rry`	[y]	`\rrhy`	[y]	`\Rubiky`
y′	`\rryp`	[y′]	`\rrhyp`	[y′]	`\Rubikyp`
z	`\rrz`	[z]	`\rrhz`	[z]	`\Rubikz`
z′	`\rrzp`	[z′]	`\rrhzp`	[z′]	`\Rubikzp`
u	`\rru`	[u]	`\rrhu`	[u]	`\Rubiku`
d	`\rrd`	[d]	`\rrhd`	[d]	`\Rubikd`
l	`\rrl`	[l]	`\rrhl`	[l]	`\Rubikl`
r	`\rrr`	[r]	`\rrhr`	[r]	`\Rubikr`
f	`\rrf`	[f]	`\rrhf`	[f]	`\Rubikf`
b	`\rrb`	[b]	`\rrhb`	[b]	`\Rubikb`

6 Colours

The rubikcube package defines seven colours which are defined as follows: red (R), orange (O), yellow (Y), green (G), blue (B), white (W), and grey (X). Now according to the following webpage[1]

`http://The-Rubiks-Cube.deviantart.com/journal/Using-Official-Rubik-s-Cube-Colors-268760351`

the official Rubik cube colours are defined as

```
.... colours which are red (PMS 200C*), green (PMS 347C*),
blue (PMS 293C*), orange (PMS 021C*), yellow (PMS 012C*)
and white.
.....
Pantone colors can not be accurately converted to RGB colors,
the colors the web runs on. But they can be approximated.
Through some research, I have found some estimations which
may help you which I have listed below. Remember, these are
just approximate RGB equivalents to the official Rubik's Cube
 colors.

Red: 200C #C41E3A  (www.perbang.dk/rgb/c41e3a/)
Green: 347C #009E60 (www.perbang.dk/rgb/009e60/)
Blue: 293C #0051BA (www.perbang.dk/rgb/0051ba/)
Orange: 021C "Pantone Orange" #FF5800 (www.perbang.dk/rgb/ff5800/)
Yellow: 012C "Pantone Yellow" #FFD500 (www.perbang.dk/rgb/ffd500/)
White: N/A #FFFFFF

Red    {HTML}{C41E3A}
green  {HTML}{009E60}
Blue   {HTML}{0051BA}
Yellow {HTML}{FFD500}
Orange {HTML}{FF5800}
White  {HTML}{FFFFFF}
```

However, we have optimised these prescribed colours very slightly for screen & print use (for example, the yellow was made very slightly brighter), and so the actual colours as used by the rubikcube package are as follows:

```
\definecolor{R}{HTML}{C41E33}
\definecolor{G}{HTML}{00BE38}
\definecolor{B}{HTML}{0051BA}
\definecolor{Y}{HTML}{FFFF00}
\colorlet{O}{orange}
\colorlet{W}{white}
\colorlet{X}{black!30}%
```

Different colours can be allocated to the ROYGBWX letters (using the `\colorlet` command) as required. For example, the standard 'red' colour could be allocated to the letter R using the command

[1] I thank Peter Bartal for pointing this out.

```
\colorlet{R}{red}
```

However, it is important to appreciate that the letter codes ROYGBWX are 'hard-wired' into many of the macros in the `rubikcube` package, so don't change these.

7 Rubik face coordinates

The coordinate origin of each view of the Rubik cube is located at the bottom left corner of the FRONT face, as shown in Figure 2. Note also that the bottom left corner of the cube itself is at $(-1, -1)$, and hence the default height and width of the cube is 4cm.

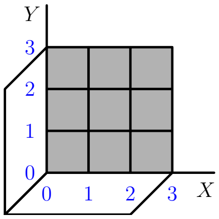


Figure 2: Origin of coordinates is at the bottom left corner of the grey FRONT face. The bottom left corner of the cube itself is at $(-1, -1)$, and hence the default height and width of the 3D-cube is 4cm.

This is an important feature since knowing the location of the origin enables one to easily use any of the TikZ commands (eg `\draw` and `\node` commands) to superimpose lines, arrows and text etc onto the Rubik cube (see Section 10).

7.1 Size of cube `\minipage`

Since the the default height and width of the 3D-cube is 4cm (see above), it follows that the width of the `\minipage` required for a cube in a TikZ picture environment can be easily calculated. For example, if the TikZ scale used is 0.5, then the appropriate width of the required minipage is $0.5 \times 4\text{cm} = 2\text{cm}$.

The default width of the semi-flat cube representation is 10cm, and that of the flat cube is 12cm.

8 Commands

All `rubikcube` commands assume a 3x3x3 cube by default. There are three primary command categories: `\Draw..` commands (which must be used *inside* a TikZ picture environment), 'parameter-allocation' commands (either inside or outside a TikZ environment), and commands which can be used in ordinary text (eg rotation commands).

A `\Draw..` command typesets either a Rubik cube, cubie or a layer using parameters set or defined via previous parameter-allocation commands (eg face colours, dimensions etc).

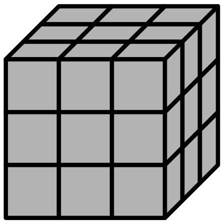
8.1 Draw commands

`\DrawRubikCubeXY`
`\DrawCubieXYxyz`

These commands draw Rubik cubes and single cubies in one of four orientations as denoted by the terminal XY viewing-direction code RU (RightUp), RD (RightDown), LU (LeftUp), LD (LeftDown). For example, the command

```
\DrawRubikCubeRU
```

will draw a Rubik cube as viewed from the RightUp direction, as shown in the following figure.



```
\begin{tikzpicture}[scale=0.7]
   \DrawRubikCubeRU
\end{tikzpicture}
```

Note that in the above example all the cubies are displayed in the default grey colour, since no Face colours have been specified.

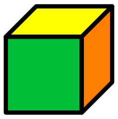
Since LaTeX commands have a maximum limit of only 9 parameters, it is necessary to use separate commands ('Face' and 'Slice' commands—see below) in order to accommodate all 27 visible colours of a 3D Rubik cube.

Since a single cubie has only three visible faces we can include colour parameters in `\DrawCubie` commands. Consequently `\DrawCubie` commands have the format

```
\DrawCubieXY{x}{y}{z}
```

where the XY pair denote the viewing direction as before, and the xyz denote the face colours associated with each of the three axes.

For example, the command `\DrawCubieRU{O}{Y}{G}` draws a single cubie as viewed from the RightUp direction, with face colours Orange (x-axis), Yellow (y-axis), Green (z-axis), as follows.



```
\begin{tikzpicture}[scale=1]
   \DrawCubieRU{O}{Y}{G}
\end{tikzpicture}
```

`\Cubiedy`
`\Cubiedx`

Minor cubie configuration changes can be effected (within a TikZ picture environment) by adjusting the dy and dx values (> 0; no units) shown in Figure 3 via the two commands

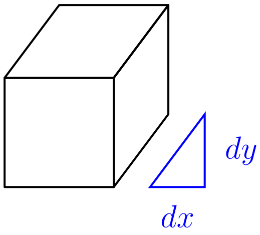
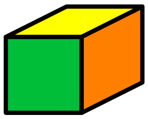


Figure 3: Cubie dy dx parameters

```
\Cubiedy{}
\Cubiedx{}
```

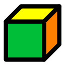
as shown in the folowing example.

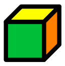


```
\begin{tikzpicture}[scale=1]
  \Cubiedy{0.4}
  \Cubiedx{0.8}
  \DrawCubieRU{O}{Y}{G}
\end{tikzpicture}
```

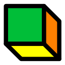
Note that the FRONT of the cubie is a unit square, and the graphic origin of the cubie image is at the bottom left corner of the FRONT (see also the section on Arrows). The default values of dy and dx are 0.4.

`\textCubieRU`
`\textCubieRD`
`\textCubieLU`
`\textCubieLD`

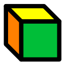
For convenience, there are also four (smaller) 'text' versions  of the four `\DrawCubie` commands for use in ordinary text, as follows:



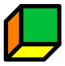
`\textCubieRU{O}{Y}{G}`



`\textCubieRD{O}{Y}{G}`



`\textCubieLU{O}{Y}{G}`



`\textCubieLD{O}{Y}{G}`

Note that the `\textCubieXY` commands are not influenced by the `\Cubiedy`, `\Cubiedx` commands as their size is preset for text use.

Other `\Draw..` commands which we shall meet later are `DrawRubikLayerFace..`, `DrawRubikLayerSide..`, `DrawNCube..`, `DrawRubikFlat` and `DrawRubikCubeFlat`.

8.1.1 `\draw` error message

See also section 3.2 regarding the error message associated with using a `\Draw...` command *outside* a TikZ picture environment.

8.2 Face commands

\RubikFaceUp
\RubikFaceDown
\RubikFaceLeft
\RubikFaceRight
\RubikFaceFront
\RubikFaceBack

These commands take nine colour arguments and allocate colours to the individual cubies of a Rubik cube face. The ordering is isomorphic to the sequence 1–9, i.e., numbering the small squares 1-3 (top row, left to right), 4-6 (middle row, left to right), 7-9 (bottom row, left to right), as follows:

#1	#2	#3
#4	#5	#6
#7	#8	#9

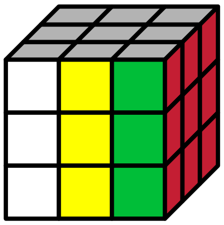
Conveniently, LaTeX allows the colour arguments to be separated by spaces (e.g., separated in groups of three), or even spread across several lines (e.g., in a square block to resemble a 9-face) in order to make the command more visually intuitive, as in the following examples.

```
\RubikFaceUp{G}{B}{G}   {G}{W}{O}   {G}{O}{G}

\RubikFaceFront{O}{W}{R}
               {W}{W}{W}
               {G}{W}{G}
```

\RubikFaceUpAll
\RubikFaceDownAll
\RubikFaceLeftAll
\RubikFaceRightAll
\RubikFaceFrontAll
\RubikFaceBackAll

For convenience, each of these commands has an associated 'ALL' command which allocates the same colour to all the cubies on a 9-face, as follows (only a single colour argument is required):

If you want a particular face to be ALL grey, then one can either omit the particular 'Face' command (since the default colour is grey), or use a 'Face' command specifying the colour-code X; for example, `\RubikFaceUpAll{X}`. However, if you do use a Face command, then all of the command's colour arguments must be used, as otherwise you will generate a 'missing parameter' error, and no colour will be allocated (i.e., you will see a black-hole). Use of these commands is shown in the following figure.



```
\begin{tikzpicture}[scale=0.7]
   \RubikFaceUpAll{X}
   \RubikFaceRightAll{R}
   \RubikFaceFront{W}{Y}{G}
                  {W}{Y}{G}
                  {W}{Y}{G}
   \DrawRubikCubeRU
\end{tikzpicture}
```

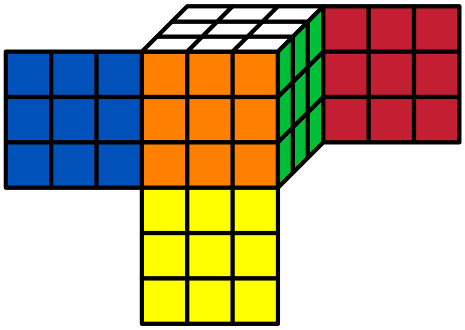
8.3 RubikCubeSolved command

\RubikCubeSolved

This command sets all the face colours to that of a standard 'solved' cube as shown in the following semi-flat image. This command is equivalent to the following set of face commands:

```
\RubikFaceUpAll{W}%
```

```
\RubikFaceDownAll{Y}%
\RubikFaceLeftAll{B}%
\RubikFaceRightAll{G}%
\RubikFaceFrontAll{O}%
\RubikFaceBackAll{R}%
```



Users can easily set their own alternative 'solved' face/colour configuration by 'renewing' this command as follows—remember to include the trailing % (note that the colours ROYGBWX are currently hardwired so don't use a different set of uppercase letters).

```
\renewcommand{\RubikCubeSolved}{%
   \RubikFaceUpAll{..}%
   \RubikFaceDownAll{..}%
   \RubikFaceLeftAll{..}%
   \RubikFaceRightAll{..}%
   \RubikFaceFrontAll{..}%
   \RubikFaceBackAll{..}%
}
```

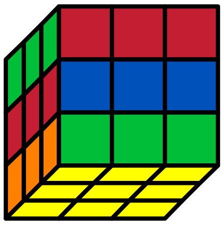
8.4 Slice commands

`\RubikSliceTopX`
`\RubikSliceEquatorX`
`\RubikSliceBottomX`

These three commands allocate the six visible cubie colours associated with a *horizontal* slice of a Rubik cube. There are three pairs of Slice commands; one pair (Left view & Right view) for each of the horizontal slices Top, Equator, Bottom. The six colour arguments associated with a given slice run in sequence from left to right irrespective of the viewpoint, eg #1 #2 #3 #4 #5 #6.

Since the viewpoint of the Rubik cube (from the Right or from the Left) influences which face the colours are associated with, it is necessary to have the view (R or L) represented in the command name.

The format of the 'slice' command is shown in the following example. The Rubik cube is shown from the LeftDown (LD) view and consequently each of the 'slice' commands in this particular example ends in L, consistent with the final `\DrawRubikCubeLD` command.



```
\begin{tikzpicture}[scale=0.7]
   \RubikFaceDownAll{Y}
   \RubikSliceTopL     {G}{G}{G} {R}{R}{R}
   \RubikSliceEquatorL {R}{R}{R} {B}{B}{B}
   \RubikSliceBottomL  {O}{O}{O} {G}{G}{G}
   \DrawRubikCubeLD
\end{tikzpicture}
```

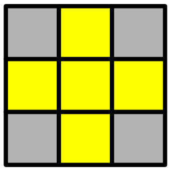
8.5 LayerFace & LayerSide commands

These commands are mainly intended for drawing final layer configurations (ie typically using the yellow face in the UP position) with or without the associated side faces of the cubies of the top layer.

8.5.1 LayerFace

`\DrawRubikLayerFace`

This LayerFace command draws a simple Rubik cube 3x3 face and allocates colours to the 9 cubies. The command takes nine ordered colour arguments. Their use is illustrated in the following example, which shows a yellow cross configuration.



```
\begin{tikzpicture}[scale=0.7]
   \DrawRubikLayerFace{X}{Y}{X}
                      {Y}{Y}{Y}
                      {X}{Y}{X}
\end{tikzpicture}
```

8.5.2 `\draw` error message

See also section 3.2 regarding the error message associated with using a `\Draw...` command *outside* a TikZ picture environment.

8.5.3 LayerSide

`\DrawLayerSideXYp`

LayerSide commands draw the associated side colours of the top layer as small rectangular sidebars. The LayerSide commands adopt a three-letter XYp **position** notation where the XY pair define the location (X:[Left | Middle | Right]; Y:[Top | Equator | Bottom]) of a particular cubie in the layer face.

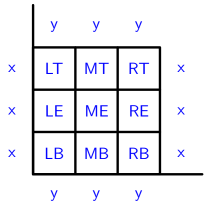


Figure 4: LayerSide letter-codes

Since corner cubies have two side faces, the ‘p’ parameter (p:[x|y]) is required to denote the directional ‘position’ of the cubie side-face relative to the cubicle XY position (x indicates adjacent along the x-axis, and y indicates adjacent along the y-axis). Since edge cubies have only one side face, the ‘p’ parameter is optional (for consistency), since it is not strictly necessary.

Commands: Various different LayerSide commands are available: those for drawing a single colour side bar, and others for facilitating drawing all 12 side bars. For example, *either* of the following commands

```
\DrawRubikLayerSideLE{G}
\DrawRubikLayerSideLEx{G}
```

draws a single small vertical green rectangle just to the left of the Left Equator (LE) square of the 9-face. The following commands draw all three side bars of a given side (Top, Bottom, Left, Right)

```
\DrawRubikLayerSideT{}{}{}
\DrawRubikLayerSideB{}{}{}
\DrawRubikLayerSideL{}{}{}
\DrawRubikLayerSideR{}{}{}
```

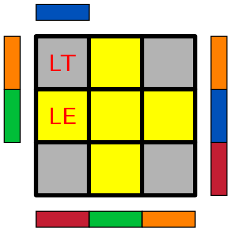
where the T and B forms require the colour parameters to be in left-right horizontal order (eg, left, middle, right), while the L and R forms require the colour parameters to be in top-down vertical order (eg, top, equator, bottom). Note that this means that the colour parameters of the L and R commands can (if required) then be positioned intuitively (vertically) as follows:

```
\DrawRubikLayerSideL{}
                    {}
                    {}
```

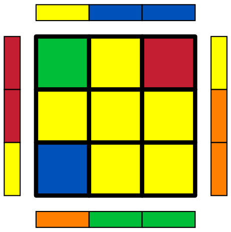
Extending this idea, the L and R forms are combined in the LR command, which takes six colour arguments ordered in left-right pairs,

```
\DrawRubikLayerSideLR{}{} {}{} {}{}
```

so that they can also be written vertically as left-right pairs. Use of these commands is shown in the following two examples.



```
\begin{tikzpicture}[scale=0.7]
   \DrawRubikLayerFace{X}{Y}{X}
                      {Y}{Y}{Y}
                      {X}{Y}{X}

   \DrawRubikLayerSideLTy{B}
   \DrawRubikLayerSideLTx{O}
   \DrawRubikLayerSideLE{G}

   \DrawRubikLayerSideR{O}{B}{R}
   \DrawRubikLayerSideB{R}{G}{O}

   \node (LT) at (0.5, 2.5)
              [red]{\small\textsf{LT}};
   \node (LE) at (0.5, 1.5)
              [red]{\small\textsf{LE}};
\end{tikzpicture}
```

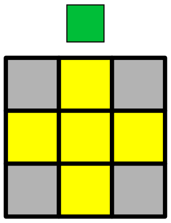


```
\begin{tikzpicture}[scale=0.7]
   \DrawRubikLayerFace{G}{Y}{R}
                      {Y}{Y}{Y}
                      {B}{Y}{Y}

   \DrawRubikLayerSideT {Y}{B}{B}
   \DrawRubikLayerSideLR{R}   {Y}
                        {R}   {O}
                        {Y}   {O}
   \DrawRubikLayerSideB {O}{G}{G}
\end{tikzpicture}
```

The default values (size) of the sidebars are as follows: width (0.3), length(1) and separation from the square face (0.3). Note that the default value of the length of a cubie side is 1. These sidebar values (decimal values ≥ 0; no units) can be changed from their default values using the three commands.

```
\RubikSideBarWidth{}
\RubikSideBarLength{}
\RubikSideBarSep{}
```

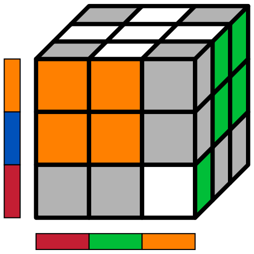
Values set in the document preamble will apply globally, while values set within a TikZ picture environment will apply only locally to that particular environment, as shown in the following example where we have set both the sidebar width and length to 0.7.



```
\begin{tikzpicture}[scale=0.7]
   \DrawRubikLayerFace{X}{Y}{X}
                      {Y}{Y}{Y}
                      {X}{Y}{X}
   \RubikSideBarWidth{0.7}
   \RubikSideBarLength{0.7}
   \DrawRubikLayerSideMTy{G}
\end{tikzpicture}
```

Note also that changing the sidebar-width or sidebar-separation values may well also change the surrounding white-space (use `\fbox` to visualise this) and may therefore require some fine-tuning of the minipage width setting in order to optimise appearance.

Since the FRONT face drawn using the `\DrawRubikCube` command is identical with that drawn using the `\DrawLayerFace` command it follows that LayerSide commands can also be used in conjunction with the FRONT face drawn using `\DrawRubikCube` command, as shown in the following example.



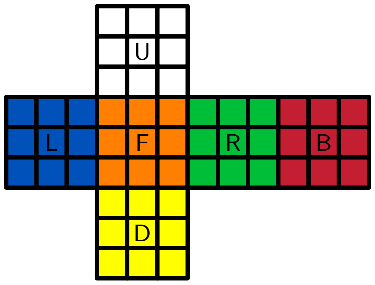
```
\begin{tikzpicture}[scale=0.7]
   \RubikFaceUp    {X}{W}{X}%
                   {W}{W}{W}%
                   {X}{W}{X}%

   \RubikFaceFront{O}{O}{X}%
                   {O}{O}{X}%
                   {X}{X}{W}%

   \RubikFaceRight{X}{G}{G}%
                   {X}{G}{G}%
                   {G}{X}{X}%
   \DrawRubikCubeRU

   \DrawRubikLayerSideL{O}{B}{R}
   \DrawRubikLayerSideB{R}{G}{O}
\end{tikzpicture}
```

Note that since using a left or a right `\DrawRubikLayerSide` command in conjunction with a `\DrawRubikCube` command will necessarily increase the width of the image, one may also have to adjust the width of the associated minipage.

8.6 Flat commands

`\DrawRubikFlat`
`\DrawRubikCubeFlat`

Flat commands are used to draw 'flat' representations of the cube. The most standard format, which is generated by the command `\DrawRubikFlat`, is shown in the following example.



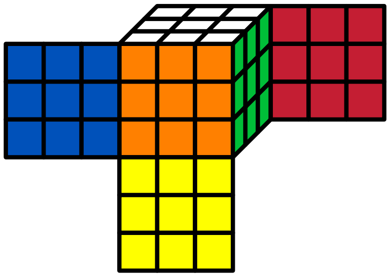
\DrawRubikFlat

```
\begin{tikzpicture}[scale=0.4]
   \RubikCubeSolved
   \DrawRubikFlat
\end{tikzpicture}
```

The addition of text (numbers or letters) in the faces is straightforward—the origin of the 1-unit grid is located at the bottom left corner of the front face (green here). The letters were placed using the following TikZ code:

```
\node (U) at (1.5, 4.5)   [black]{\small\textsf{U}};
\node (D) at (1.5, -1.5)  [black]{\small\textsf{D}};
\node (L) at (-1.5, 1.5)  [black]{\small\textsf{L}};
\node (R) at (4.5, 1.5)   [black]{\small\textsf{R}};
\node (F) at (1.5, 1.5)   [black]{\small\textsf{F}};
\node (B) at (7.5, 1.5)   [black]{\small\textsf{B}};
```

An interesting ‘semi-flat’ alternative format uses the standard view of the cube and appends the three hidden sides (see Rokicki *etal.*, 2013), which is generated by the command `\DrawRubikCubeFlat` as follows.



```
\begin{tikzpicture}[scale=0.5]
   \RubikCubeSolved
   \DrawRubikCubeFlat
\end{tikzpicture}
```

8.6.1 `\draw` error message

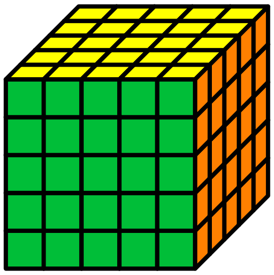
See also section 3.2 regarding the error message associated with using a `\Draw...` command *outside* a TikZ picture environment.

9 NCube (NxNxN)

`\DrawNCubeAll` An ‘NCube’ is a solved NxNxN cube drawn from the RU direction; (ie. only shows faces UP, FRONT, RIGHT). The cubie colours of each face are All the same.

```
\DrawNCubeAll{N}{Xcolour}{Ycolour}{Zcolour}.
```

This command takes four ordered parameters (N, X, Y, Z)—the number (integer; $N > 0$) of cubies along an edge, followed by three face colours in XYZ order. Since the viewpoint is only from the RU direction, the three colour parameters are: X(Right), Y(Up), Z(Front).



```
\begin{tikzpicture}[scale=0.5]
   \DrawNCubeAll{5}{O}{Y}{G}
\end{tikzpicture}
```

9.0.2 `\draw` error message

See also section 3.2 regarding the error message associated with using a `\Draw...` command *outside* a TikZ picture environment.

10 Arrows

The rubikcube package does not offer any special commands for drawing arrows since it is straightforward just to include the appropriate TikZ ‘draw’ commands in the `tikzpicture` environment.

In order to facilitate using the standard TikZ ‘draw’ commands both the RubikCubeFaceFront and LayerFace commands have the coordinate origin at the bottom left corner, and draw ‘faces’ consisting of 9 unit-squares in a 3x3 grid, as shown in Figure 5. Consequently the start and finish coordinates for any arrow or line are easy to determine.

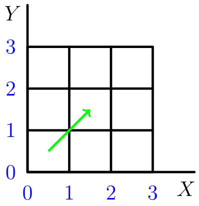


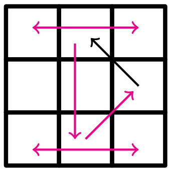
Figure 5: RubikCubeFaceFront and RubikLayerFace coordinates

Thus, to draw an ‘ultra thick’ green arrow from the centre of cubie LB (say point A) to the centre of cubie ME (say point B)—see Figure 5—we first determine the coordinates of A & B, namely $A(0.5, 0.5)$, $B(1.5, 1.5)$. Now, to draw the arrow from A to B we just include the following TikZ command in the `tikzpicture` environment.

```
\draw[->,ultra thick,color=green] (0.5,0.5) -- (1.5, 1.5);
```

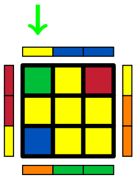
The following example shows the cubie changes in the UP face generated by the rotation sequence **FRUR′U′F′**. The magenta arrows indicate movement *with* cubie rotations, while the black arrow indicates movement *without* rotation.

This example also highlights the fact that when there are several arrows, the start and end positions often need to be offset slightly away from cubie centres.



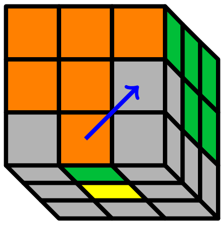
```
\begin{tikzpicture}[scale=0.7]
\DrawRubikLayerFaceAll{W}
\draw[->,thick,color=magenta] (1.5,0.5) -- (2.4, 1.4);
\draw[->,thick] (2.5,1.5) -- (1.6, 2.4);
\draw[->,thick,color=magenta] (1.3, 2.3) -- (1.3, 0.5);
\draw[<->,thick,color=magenta] (0.5,2.6) -- (2.5, 2.6);
\draw[<->,thick,color=magenta] (0.5,0.3) -- (2.5, 0.3);
\end{tikzpicture}
```

Since the coordinates shown in Figure 5 extend outwards in all directions, they can also be used as a guide for drawing arrows (or other structures) outside this 3x3 'face' square. This approach is shown in the following example, where we have changed the TikZ scale to 0.4 in order to generate a small figure, and also reduced the minipage width to minimise surrounding white-space, in order to facilitate placing the figure and the code side-by-side.



```
\begin{minipage}{1.8cm}
\centering
\begin{tikzpicture}[scale=0.4]
   \DrawRubikLayerFace{G}{Y}{R}
                      {Y}{Y}{Y}
                      {B}{Y}{Y}

   \DrawRubikLayerSideT {Y}{B}{B}
   \DrawRubikLayerSideLR{R}   {Y}
                        {R}   {O}
                        {Y}   {O}
   \DrawRubikLayerSideB {O}{G}{G}
   \draw[->,ultra thick,color=green] (0.5,5) -- (0.5, 4);
\end{tikzpicture}
\end{minipage}
```

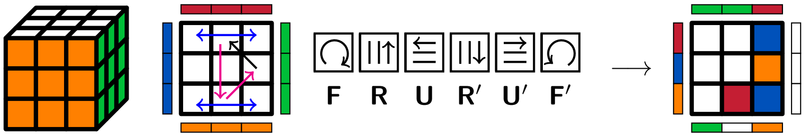
The following example shows an arrow on the Rubik cube. The origin of coordinates is at the bottom left corner of the FRONT face (see Section 7).



```
  \begin{tikzpicture}[scale=0.7]

  \RubikFaceFront{O}{O}{O}
                 {O}{O}{X}
                 {X}{O}{X}

  \RubikFaceRight{G}{G}{G}
                 {X}{G}{G}
                 {X}{X}{X}

  \RubikFaceDown {X}{G}{X}
                 {X}{Y}{X}
                 {X}{X}{X}

  \DrawRubikCubeRD
  \draw[ultra thick,->,color=blue]
              (1.5,0.5) -- (2.5, 1.5);
\end{tikzpicture}
```

11 Final example

We now present, as a final example, the code used to draw the front page figure:



```
\hfil
\begin{minipage}{2cm}
  \centering
  \begin{tikzpicture}[scale=0.4]
    \RubikCubeSolved
    \DrawRubikCubeRU
  \end{tikzpicture}
\end{minipage}
\begin{minipage}{2cm}
  \centering
  \begin{tikzpicture}[scale=0.4]
    \DrawRubikLayerFaceAll{W}
    \DrawRubikLayerSideT {R}{R}{R}
    \DrawRubikLayerSideLR{B}   {G}
                         {B}   {G}
                         {B}   {G}
    \DrawRubikLayerSideB {O}{O}{O}
    \draw[thick,->,color=magenta] (1.5,0.5) -- (2.4, 1.4);
    \draw[thick,->] (2.5,1.5) -- (1.6, 2.4);
    \draw[thick,->,color=magenta] (1.3, 2.3) -- (1.3, 0.5);
    \draw[thick,<->,  color=blue] (0.5,2.6) -- (2.5, 2.6);
    \draw[thick,<->,  color=blue] (0.5,0.3) -- (2.5, 0.3);
  \end{tikzpicture}
\end{minipage}
\RubikF\RubikR\RubikU\RubikRp\RubikUp\RubikFp\
\ \ $\longrightarrow$
\begin{minipage}{2cm}
  \centering
  \begin{tikzpicture}[scale=0.4]
    \DrawRubikLayerFace{W}{W}{B}
                       {W}{W}{O}
                       {W}{R}{B}
    \DrawRubikLayerSideT {G}{G}{R}
    \DrawRubikLayerSideLR{R}   {W}
                         {B}   {W}
                         {O}   {W}
    \DrawRubikLayerSideB {G}{W}{O}
  \end{tikzpicture}
\end{minipage}
\hfil
```

12 Known issues & shortcomings

Please contact the authors regarding any errors, shortcomings and ideas etc.

- The rotation hieroglyphs are optimised for a 10pt font, and do not scale with document font size. This may be implemented in the future, and will require careful mapping of the TikZ scale factor to font point size, or just replacing all length parameters by a multiple of `\em` or `\en` etc. At the moment it seems to work reasonably well with both 11pt and 12pt fonts.

- The sidebars can only be drawn in relation to either a Front face of a cube, or a LayerFace.

- The FRONT face as drawn using the `\DrawNCubeAll` macro does not have its origin at the bottom left corner. Not regarded as a significant problem since this macros is really just a cosmetic fix/tool for drawing a neat NxNxN cube.

13 Acknowledgements

We would like to thank Peter Bartal and Peter Grill for useful ideas and suggestions. We have built on some of the initial ideas of Bartal (2011), and have cited these instances in the code documentation.

14 Future supporting packages

Since Rubik-type cubes currently exist in a variety of sizes (from 2x2x2 to 11x11x11) it is possible that similar packages dealing with cubes of other sizes may be made in the future. Supporting tools may also be made using various programming languages, eg Lua etc. Consequently the naming of such packages and tools, and even the associated CTAN directories, needs to be given some thought in order to prevent possible confusion. We suggest, therefore, the following directory structure for CTAN might be appropriate.

$$\textsc{rubik}\left\{\begin{array}{l}\text{rubiktools}\\ \text{rubik2x}\\ \text{rubik3x}\\ \dots\\ \text{rubik11x}\end{array}\right.$$

15 References

- Bartal P (2011)

  `http://tex.stackexchange.com/questions/34482/`

- Chen JJ (2004). Group theory and the Rubik’s cube. `http://www.math.harvard.edu/~jjchen/docs/Group_Theory_and_the_Rubik’s_Cube.pdf`

- Demaine ED, Demaine ML, Eisenstat S, Lubiw A and Winslow A (2011). Algorithms for solving Rubik’s cubes. `http://www.arxiv.org/abs/1106.5736/`

- Garfath-Cox, A (1981). *The cube*, (Bolden Publishing Co., East Molesey, Surrey) pp.32. [copy in British Library]

- Duvoid T (2010). Méthode simple pour remonter le Rubik’s cube.
  `http://duvoid.fr/rubik/rubik-debutant-couleurs.pdf`
  `http://duvoid.fr/rubik/sources/notation_en.eps`
  `http://duvoid.fr/rubik/sources/rubik-debutant-couleurs.tex`

- Duvoid T (2011). Méthode avancée pour remonter le Rubik’s cube.
  `http://duvoid.fr/rubik/rubik-friddrich-couleurs.pdf`
  `http://duvoid.fr/rubik/sources/rubik-friddrich-couleurs.tex`

- Fridrich J. `http://www.ws.binghamton.edu/fridrich/`. See the useful ‘notation’ section on the ‘Pretty patterns’ webpage at `http://www.ws.binghamton.edu/fridrich/ptrns.html`.

- Golomb SW (1982). Rubik’s cube and quarks. *American Scientist*; 70, pp. 257–259.

- Gymrek M (2009). The mathematics of toys and games.
  `http://wen.mit.edu/sp.268/www/rubik.pdf`

- Hofstadter D (1981). Rubik cube. *Scientific American*; March issue.
- Hutchings M (2011). The mathematics of Rubik's cube (slide presentation). `http://math.berkeley.edu/~hutching/`
- Jelinek website (Jelinek J). Rubik's cube solution method. `http://rubikscube.info/`
- Joyner D (2008). *Adventures in group theory: Rubik's cube, Merlin's machine and other mathematical toys.* `http://mike.verdone.ca/media/rubiks.pdf`
- Kriz J and Siegel P (2008). Rubik's cube-inspired puzzles demonstrate math's simple groups. *Scientific American*; [? July 2008]
- Learntofish (2010). `http://learntofish.wordpress.com/2010/10/08/20-moves-suffice-to-solve-the-rubiks-cube/`
- Randelhofer website (Randelhofer W). Pretty patterns. `http://www.randelhofer.ch/rubik/patterns/U080.01.html`
- Reid M. `http://www.math.ucf.edu/~reid/Rubik/`
- Rokicki T, Kociemba H, Davidson M and Dethridge J (2013). The diameter of the Rubik's cube is twenty. *SIAM. J. Discrete Math.*, 27, 1082–1105. (`http://tomas.rokicki.com/rubik20.pdf`)
- Rubik's cube. See Section on notation. `http://en.wikipedia.org/wiki/Rubik's_Cube`
- Speedsolving website. `www.speedsolving.com/`
- Tran R (2005). A mathematical approach to solving Rubik's cube. `http://www.math.ubc.ca/~cass/courses/m308/projects/rtran/rtran.pdf`
- Treep A and Waterman M (1987). Marc Waterman's Algorithm, Part 2. *Cubism For Fun 15*, p. 10 (Nederlandse Kubus Club) [cited from *Wikipedia* (Rubik's cube)]
- Vandenbergh L. CUBEZONE (`http://www.cubezone.be`).
- WCA (2012). World Cube Association Regulations. See Section 12 for notation. `http://www.worldcubeassociation.org/regulations.htm`

16 The code

The conventions we adopt regarding capital letters and the XYZ argument ordering are detailed in Section 4.

16.1 Package heading

```
⟨*rubikcube⟩
\def\RCfileversion{2.0}%
\def\RCfiledate{2014/01/20}%
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{rubikcube}[\RCfiledate\space (v\RCfileversion)]
```

The package requires TikZ

```
\@ifpackageloaded{tikz}{}{%
  \typeout{---rubikcube requires the TikZ package.}%
  \RequirePackage{tikz}}%
```

\rubikcube

First we create a suitable logo

```
\newcommand{\rubikcube}{\textsf{rubikcube}}%
```

16.2 Some useful internal commands

\@rr
\@rrp
\@rrw
\@rrwp
\@rrs
\@rrsp
\@rra
\@rrap
\@xyzh
\@xyzhp
\@xyzRubik
\@xyzRubikp
\@SquareLetter
\@hRubik

Internal commands. These are used to generate the prime, w, w-prime, s, s-prime, a, a-prime rotation commands. The `\@xyz..` commands are used to generate the x,y,z,u,d,l,r,f,b, and their prime rotation commands. The `\@SquareLetter` command is used to form the separate square hieroglyph form used for rotations with no FRONT representation (eg B.., Fs, Fsp, Fa, Fap, S, Sp, Sf, Sfp, Sb, Sbp). The `\@hRubik` is the vertical shift used to raise the box carying the rotation letter-code in `\Rubik..` commands not visible from the front.

The idea is that by using these internal tools we will be able to more easily standardise the size and position of all the various glyphs. However, these internal tools are currently only partially implemented (= work in progress).

```
\newcommand{\@rr}[1]{\textbf{\textsf{#1}}}
\newcommand{\@rrp}[1]{\textbf{\textsf{#1}$^\prime$}}
\newcommand{\@rrw}[1]{\textbf{\textsf{#1\footnotesize{w}}}}
\newcommand{\@rrwp}[1]{\textbf{\textsf{#1\footnotesize{w}}$^\prime$}}
\newcommand{\@rrs}[1]{\textbf{\textsf{#1\footnotesize{s}}}}
\newcommand{\@rrsp}[1]{\textbf{\textsf{#1\footnotesize{s}}$^\prime$}}
\newcommand{\@rra}[1]{\textbf{\textsf{#1\footnotesize{a}}}}
\newcommand{\@rrap}[1]{\textbf{\textsf{#1\footnotesize{a}}$^\prime$}}
\newcommand{\@xyzh}[1]{\textbf{[\textsf{#1}]}\,}
\newcommand{\@xyzhp}[1]{\textbf{[\textsf{#1}$^\prime$]}\,}
\newcommand{\@xyzRubik}[1]{\raisebox{3.45pt}{\textbf{[\textsf{#1}]}}}
\newcommand{\@xyzRubikp}[1]{\raisebox{3.45pt}{\textbf{[\textsf{#1}$^\prime$]}}}
\newcommand{\@SquareLetter}[1]{\setlength\fboxsep{2.5pt}\fboxrule=0.8pt%
   \fbox{\rule[-1pt]{0pt}{8.5pt}\raisebox{-0.5pt}{#1}}}
\newlength\@hRubik%
\setlength{\@hRubik}{0.185cm}%
```

16.3 Colours

These colour allocations were initially defined by Peter Bartal (2011). Now modified so that grey is defined as black!30.

```
\definecolor{R}{HTML}{C41E33}%
\definecolor{G}{HTML}{00BE38}%
\definecolor{B}{HTML}{0051BA}%
\definecolor{Y}{HTML}{FFFF00}%
\colorlet{X}{black!30}% grey
\colorlet{O}{orange}%
\colorlet{W}{white}%
```

16.4 Face commands

Cubeface notation = U,D,L,R,F,B (Singmaster)
Cubiesquare notation = t,b,l,r = top,bottom,left,right. Need to use t,b for cubiesquares to avoid confusion with cubeFace notation. Number the cubiesquares on a face 1-9, starting topleft, ending bottom right, as follows:

Up row (1,2,3) = tl, tm, tr
middle row (4,5,6) = ml, mm, mr
bottom row (7,8,9) = bl, bm, br

`\RubikFaceUp`
`\RubikFaceDown`
`\RubikFaceLeft`
`\RubikFaceRight`
`\RubikFaceFront`
`\RubikFaceBack`

These 5 commands allocate a colour to each of the 9 cubiesquares in the specified face (Up, Down, Left, Right, Front, Back). Each command takes 9 arguments (colour codes) in the order 1–9 as specified above.
For example, `\RubikFaceUp{R}{O}{Y} {G}{B}{W} {X}{R}{G}`
Each of the 9 `\def{}` commands below allocates one colour to a specific cubiesquare, using a simple three-letter encoding. Each letter is an initial letter of the words Up, Down, Left, Right, Front, Back, left, middle, right, top, middle, bottom. For example in the command `\Urt{#1}` the U stands for the Up face of the cube, while the `rt` stands for the "right-top" cubiesquare on this face. Note that the order of the two lowercase letters (in this case `rt`) are written in the x, y order; ie the first of the two lowercase letters relates to the x direction (either left, middle, or right), while the second lowercase letter relates to the y direction (either top, middle, or bottom)

```
\newcommand{\RubikFaceUp}[9]{%
\def\Ult{#1}\def\Umt{#2}\def\Urt{#3}%
\def\Ulm{#4}\def\Umm{#5}\def\Urm{#6}%
\def\Ulb{#7}\def\Umb{#8}\def\Urb{#9}%
}
\newcommand{\RubikFaceFront}[9]{%
\def\Flt{#1}\def\Fmt{#2}\def\Frt{#3}%
\def\Flm{#4}\def\Fmm{#5}\def\Frm{#6}%
\def\Flb{#7}\def\Fmb{#8}\def\Frb{#9}%
}
\newcommand{\RubikFaceRight}[9]{%
\def\Rlt{#1}\def\Rmt{#2}\def\Rrt{#3}%
\def\Rlm{#4}\def\Rmm{#5}\def\Rrm{#6}%
\def\Rlb{#7}\def\Rmb{#8}\def\Rrb{#9}%
}
\newcommand{\RubikFaceDown}[9]{%
\def\Dlt{#1}\def\Dmt{#2}\def\Drt{#3}%
\def\Dlm{#4}\def\Dmm{#5}\def\Drm{#6}%
\def\Dlb{#7}\def\Dmb{#8}\def\Drb{#9}%
}
\newcommand{\RubikFaceLeft}[9]{%
\def\Llt{#1}\def\Lmt{#2}\def\Lrt{#3}%
\def\Llm{#4}\def\Lmm{#5}\def\Lrm{#6}%
\def\Llb{#7}\def\Lmb{#8}\def\Lrb{#9}%
}
\newcommand{\RubikFaceBack}[9]{%
\def\Blt{#1}\def\Bmt{#2}\def\Brt{#3}%
\def\Blm{#4}\def\Bmm{#5}\def\Brm{#6}%
\def\Blb{#7}\def\Bmb{#8}\def\Brb{#9}%
}
```

\RubikFaceUpAll
\RubikFaceDownAll
\RubikFaceLeftAll
\RubikFaceRightAll
\RubikFaceFrontAll
\RubikFaceBackAll

These 5 commands allocate the same colour to all 9 cubiesquares in the specified face (Up, Down, Left, Right, Front). Each command therefore takes only 1 argument (one of the colour codes).
For example, `\RubikFaceUpAll{R}`

```
\newcommand{\RubikFaceUpAll}[1]{%
\def\Ult{#1}\def\Umt{#1}\def\Urt{#1}%
\def\Ulm{#1}\def\Umm{#1}\def\Urm{#1}%
\def\Ulb{#1}\def\Umb{#1}\def\Urb{#1}%
}
\newcommand{\RubikFaceFrontAll}[1]{%
\def\Flt{#1}\def\Fmt{#1}\def\Frt{#1}%
\def\Flm{#1}\def\Fmm{#1}\def\Frm{#1}%
\def\Flb{#1}\def\Fmb{#1}\def\Frb{#1}%
}
\newcommand{\RubikFaceRightAll}[1]{%
\def\Rlt{#1}\def\Rmt{#1}\def\Rrt{#1}%
\def\Rlm{#1}\def\Rmm{#1}\def\Rrm{#1}%
\def\Rlb{#1}\def\Rmb{#1}\def\Rrb{#1}%
}
\newcommand{\RubikFaceLeftAll}[1]{%
\def\Llt{#1}\def\Lmt{#1}\def\Lrt{#1}%
\def\Llm{#1}\def\Lmm{#1}\def\Lrm{#1}%
\def\Llb{#1}\def\Lmb{#1}\def\Lrb{#1}%
}
\newcommand{\RubikFaceDownAll}[1]{%
\def\Dlt{#1}\def\Dmt{#1}\def\Drt{#1}%
\def\Dlm{#1}\def\Dmm{#1}\def\Drm{#1}%
\def\Dlb{#1}\def\Dmb{#1}\def\Drb{#1}%
}
\newcommand{\RubikFaceBackAll}[1]{%
\def\Blt{#1}\def\Bmt{#1}\def\Brt{#1}%
\def\Blm{#1}\def\Bmm{#1}\def\Brm{#1}%
\def\Blb{#1}\def\Bmb{#1}\def\Brb{#1}%
}
```

We now use these commands to initialise all visible faces to default colour grey (X)

```
\RubikFaceUpAll{X}%
\RubikFaceDownAll{X}%
\RubikFaceLeftAll{X}%
\RubikFaceRightAll{X}%
\RubikFaceFrontAll{X}%
\RubikFaceBackAll{X}%
```

16.5 RubikCubeSolved command

\RubikCubeSolved

This command sets the face/colour configuration (state) of a typical solved Rubik cube.

```
\newcommand{\RubikCubeSolved}{%
  \RubikFaceUpAll{W}%
  \RubikFaceDownAll{Y}%
  \RubikFaceLeftAll{B}%
  \RubikFaceRightAll{G}%
  \RubikFaceFrontAll{O}%
  \RubikFaceBackAll{R}%
}
```

16.6 Slice macros

\RubikSliceTopR
\RubikSliceTopL
\RubikSliceEquatorR
\RubikSliceEquatorL
\RubikSliceBottomR
\RubikSliceBottomL

These 6 commands allocate colours for the 6 visible ordered cubiesquares along the side of a horizontal slice ordered 1–6 from left to right. We require both L view and R view versions.

```
\newcommand{\RubikSliceTopR}[6]{%
\def\Flt{#1}\def\Fmt{#2}\def\Frt{#3}%
\def\Rlt{#4}\def\Rmt{#5}\def\Rrt{#6}%
}
\newcommand{\RubikSliceTopL}[6]{%
\def\Llt{#1}\def\Lmt{#2}\def\Lrt{#3}%
\def\Flt{#4}\def\Fmt{#5}\def\Frt{#6}%
}
\newcommand{\RubikSliceEquatorR}[6]{%
\def\Flm{#1}\def\Fmm{#2}\def\Frm{#3}%
\def\Rlm{#4}\def\Rmm{#5}\def\Rrm{#6}%
}
\newcommand{\RubikSliceEquatorL}[6]{%
\def\Llm{#1}\def\Lmm{#2}\def\Lrm{#3}%
\def\Flm{#4}\def\Fmm{#5}\def\Frm{#6}%
}
\newcommand{\RubikSliceBottomR}[6]{%
\def\Flb{#1}\def\Fmb{#2}\def\Frb{#3}%
\def\Rlb{#4}\def\Rmb{#5}\def\Rrb{#6}%
}
\newcommand{\RubikSliceBottomL}[6]{%
\def\Llb{#1}\def\Lmb{#2}\def\Lrb{#3}%
\def\Flb{#4}\def\Fmb{#5}\def\Frb{#6}%
}
```

16.7 Cube drawing macros

Since the three visible sides of a Rubik cube have up to 27 non-grey colours, and TeX has only 9 macro parameters available, we are forced to draw Rubik cubes by first specifying the colours on each of the three faces, and then using a 'DrawRubikCubeXY' command, where the trailing XY code defines the view direction (X = either R, L; Y = either U, D). The order of the XY code is important: X first, Y second.

On each face the cubies are drawn in the following order: Top row (left to right), Middle row (left to right), Bottom row (left to right).

The TikZ draw cycle for each cubie square on a Rubik cube face cycles through the four corners of the cubie in the following order: lb → lt → rt → rb; the code being lb (LeftBottom), lt (LeftTop), rt (RightTop), rb (RightBottom) (only need four coords) origin at bottom left of front face.

`\DrawRubikCubeFrontFace` This 'FrontFace' command is an 'internal' command which draws and paints all the cubiesquares on the front face of a cube. It is used by all of the cube drawing macros which display the front face. The 9 colours are allocated by an earlier `\RubikFaceFront` command. These Face macros are based, in part, on those of Peter Bartal (2011).

```
\newcommand{\DrawRubikCubeFrontFace}{%
%  ---top row left to right
\draw[line join=round,line cap=round,ultra thick,fill=\Flt]%
(0,2) -- (0, 3) -- (1,3) -- (1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Fmt]%
(1,2) -- (1, 3) -- (2,3) -- (2,2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frt]%
(2,2) -- (2, 3) -- (3,3) -- (3,2) -- cycle;
% -----middle row left to right
\draw[line join=round,line cap=round,ultra thick,fill=\Flm]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Fmm]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frm]%
(2,1) -- (2, 2) -- (3,2) -- (3,1) -- cycle;
% ----bottom row left to right
\draw[line join=round,line cap=round,ultra thick,fill=\Flb]%
(0,0) -- (0, 1) -- (1,1) -- (1,0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Fmb]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frb]%
(2,0) -- (2, 1) -- (3,1) -- (3,0) -- cycle;
}
```

16.7.1 Viewing direction

The command ‘DrawRubikCubeXY’ command uses a trailing XY code to specify the view direction (X = either R, L; Y = either U, D). The order of the XY code is important: X first, Y second (so its easy to remember).

`\DrawRubikCubeRU` This command draws and paints a Rubik cube as viewed from the Right Upper (RU) viewpoint. It starts by using the internal command `\DrawRubikCubeFrontFace` to draw the Front face, and then draws the Up face followed by the Right face. The colours are allocated by the last `\RubikFaceUp` and `\RubikFaceRight` commands.

```
\newcommand{\DrawRubikCubeRU}{%
\DrawRubikCubeFrontFace %% frontface
%%-----------Up face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Ult]%
(0.66,3.66) -- (1,4) -- (2,4) -- (1.66,3.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umt]%
(1.66,3.66) -- (2,4) -- (3,4) -- (2.66,3.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urt]%
(2.66,3.66) -- (3,4) -- (4,4) -- (3.66,3.66)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulm]%
(0.33,3.33) -- (0.66,3.66) -- (1.66,3.66) -- (1.33,3.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umm]%
(1.33,3.33) -- (1.66,3.66) -- (2.66,3.66) -- (2.33,3.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urm]%
(2.33,3.33) -- (2.66,3.66) -- (3.66,3.66) -- (3.33,3.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulb]%
(0,3) -- (0.33,3.33) -- (1.33,3.33) -- (1,3)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umb]%
(1,3) -- (1.33,3.33) -- (2.33,3.33) -- (2,3)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urb]%
(2,3) -- (2.33,3.33) -- (3.33,3.33) -- (3,3)  -- cycle;
%%-----------Right face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(3,2) -- (3, 3) -- (3.33,3.33) -- (3.33,2.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmt]%
(3.33,2.33) -- (3.33, 3.33) -- (3.66,3.66) -- (3.66,2.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(3.66,2.66) -- (3.66, 3.66) -- (4,4) -- (4,3)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlm]%
(3,1) -- (3, 2) -- (3.33,2.33) -- (3.33,1.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmm]%
(3.33,1.33) -- (3.33, 2.33) -- (3.66,2.66) -- (3.66,1.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrm]%
(3.66,1.66) -- (3.66, 2.66) -- (4,3) -- (4,2)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(3,0) -- (3, 1) -- (3.33,1.33) -- (3.33,0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmb]%
(3.33,0.33) -- (3.33, 1.33) -- (3.66,1.66) -- (3.66,0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(3.66,0.66) -- (3.66, 1.66) -- (4,2) -- (4,1)  -- cycle;
}
```

\DrawRubikCube This command is equivalent to the previous \DrawRubikCubeRU and hence is the default form (ie if one omits the trailing XY viewpoint code).

```
\newcommand{\DrawRubikCube}{\DrawRubikCubeRU}
```

\DrawRubikCubeRD This command draws and paints a Rubik cube as viewed from the Right Down (RD) viewpoint.

```
\newcommand{\DrawRubikCubeRD}{%
\DrawRubikCubeFrontFace %% frontface
%%----------Right face--------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(3,2) -- (3, 3) -- (3.33,2.66) -- (3.33,1.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmt]%
(3.33,1.66) -- (3.33, 2.66) -- (3.66,2.33) -- (3.66,1.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(3.66,1.33) -- (3.66, 2.33) -- (4,2) -- (4,1)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlm]%
(3,1) -- (3, 2) -- (3.33,1.66) -- (3.33,0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmm]%
(3.33,0.66) -- (3.33, 1.66) -- (3.66,1.33) -- (3.66,0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrm]%
(3.66,0.33) -- (3.66, 1.33) -- (4,1) -- (4,0)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(3,0) -- (3, 1) -- (3.33,0.66) -- (3.33,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmb]%
(3.33,-0.33) -- (3.33, 0.66) -- (3.66,0.33) -- (3.66,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(3.66,-0.66) -- (3.66, 0.33) -- (4,0) -- (4,-1)  -- cycle;
%%-----------Down face---------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlt]%
(0.33,-0.33) -- (0, 0) -- (1,0) -- (1.33,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmt]%
(1.33,-0.33) -- (1, 0) -- (2,0) -- (2.33,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drt]%
(2.33,-0.33) -- (2, 0) -- (3,0) -- (3.33,-0.33)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlm]%
(0.66,-0.66) -- (0.33, -0.33) -- (1.33,-0.33) -- (1.66,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmm]%
(1.66,-0.66) -- (1.33, -0.33) -- (2.33,-0.33) -- (2.66,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drm]%
(2.66,-0.66) -- (2.33, -0.33) -- (3.33,-0.33) -- (3.66,-0.66)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb]%
(1,-1) -- (0.66, -0.66) -- (1.66,-0.66) -- (2,-1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmb]%
(2,-1) -- (1.66, -0.66) -- (2.66,-0.66) -- (3,-1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drb]%
(3,-1) -- (2.66, -0.66) -- (3.66,-0.66) -- (4,-1)  -- cycle;
}
```

`\DrawRubikCubeLD` This command draws and paints a Rubik cube as viewed from the Left Down (LD) viewpoint.

```
\newcommand{\DrawRubikCubeLD}{%
\DrawRubikCubeFrontFace %% frontface
%%-----------Left face--------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(-1,1) -- (-1, 2) -- (-0.66,2.33) -- (-0.66,1.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmt]%
(-0.66,1.33) -- (-0.66, 2.33) -- (-0.33,2.66) -- (-0.33,1.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrt]%
(-0.33,1.66) -- (-0.33, 2.66) -- (0,3) -- (0,2)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Llm]%
(-1,0) -- (-1, 1) -- (-0.66,1.33) -- (-0.66,0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmm]%
(-0.66,0.33) -- (-0.66, 1.33) -- (-0.33,1.66) -- (-0.33,0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrm]%
(-0.33,0.66) -- (-0.33, 1.66) -- (0,2) -- (0,1)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Llb]%
(-1,-1) -- (-1, 0) -- (-0.66,0.33) -- (-0.66,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmb]%
(-0.66,-0.66) -- (-0.66, 0.33) -- (-0.33,0.66) -- (-0.33,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrb]%
(-0.33,-0.33) -- (-0.33, 0.66) -- (0,1) -- (0,0)  -- cycle;
%%-----------Down face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlt]%
(-0.33,-0.33) -- (0, 0) -- (1,0) -- (0.66,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmt]%
(0.66,-0.33) -- (1, 0) -- (2,0) -- (1.66,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drt]%
(1.66,-0.33) -- (2, 0) -- (3,0) -- (2.66,-0.33)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlm]%
(-0.66,-0.66) -- (-0.33, -0.33) -- (0.66,-0.33) -- (0.33,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmm]%
(0.33,-0.66) -- (0.66, -0.33) -- (1.66,-0.33) -- (1.33,-0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drm]%
(1.33,-0.66) -- (1.66, -0.33) -- (2.66,-0.33) -- (2.33,-0.66)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb]%
(-1,-1) -- (-0.66, -0.66) -- (0.33,-0.66) -- (0,-1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmb]%
(0,-1) -- (0.33, -0.66) -- (1.33,-0.66) -- (1,-1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drb]%
(1,-1) -- (1.33, -0.66) -- (2.33,-0.66) -- (2,-1)  -- cycle;
}
```

\DrawRubikCubeLU This command draws and paints a Rubik cube as viewed from the Left Up (LU) viewpoint.

```
\newcommand{\DrawRubikCubeLU}{%
\DrawRubikCubeFrontFace %% frontface
%%-----------Left face-----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(-1,3) -- (-1, 4) -- (-0.66,3.66) -- (-0.66,2.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmt]%
(-0.66,2.66) -- (-0.66, 3.66) -- (-0.33,3.33) -- (-0.33,2.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrt]%
(-0.33,2.33) -- (-0.33, 3.33) -- (0,3) -- (0,2)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Llm]%
(-1,2) -- (-1, 3) -- (-0.66,2.66) -- (-0.66,1.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmm]%
(-0.66,1.66) -- (-0.66, 2.66) -- (-0.33,2.33) -- (-0.33,1.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrm]%
(-0.33,1.33) -- (-0.33, 2.33) -- (0,2) -- (0,1)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Llb]%
(-1,1) -- (-1, 2) -- (-0.66,1.66) -- (-0.66,0.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmb]%
(-0.66,0.66) -- (-0.66, 1.66) -- (-0.33,1.33) -- (-0.33,0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrb]%
(-0.33,0.33) -- (-0.33, 1.33) -- (0,1) -- (0,0)  -- cycle;
%%-----------Up face---------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Ult]%
(-0.66,3.66) -- (-1, 4) -- (0,4) -- (0.33,3.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umt]%
(0.33,3.66) -- (0, 4) -- (1,4) -- (1.33,3.66)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urt]%
(1.33,3.66) -- (1, 4) -- (2,4) -- (2.33,3.66)  -- cycle;
%%---middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulm]%
(-0.33,3.33) -- (-0.66, 3.66) -- (0.33,3.66) -- (0.66,3.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umm]%
(0.66,3.33) -- (0.33, 3.66) -- (1.33,3.66) -- (1.66,3.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urm]%
(1.66,3.33) -- (1.33, 3.66) -- (2.33,3.66) -- (2.66,3.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulb]%
(0,3) -- (-0.33, 3.33) -- (0.66,3.33) -- (1,3)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umb]%
(1,3) -- (0.66, 3.33) -- (1.66,3.33) -- (2,3)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urb]%
(2,3) -- (1.66, 3.33) -- (2.66,3.33) -- (3,3)  -- cycle;%
\ %%trailing space
}
```

16.8 LayerFace commands

\DrawRubikLayerFace
\DrawRubikLayerFaceAll

[Nov 2, 2013] These two LayerFace commands draw and paint a single 9-cubie face. The first command takes 9 ordered colour parameters, (ordered in layers from top left to bottom right, so #1 is the placeholder for the colour of the TopLeft cubie etc.) The second takes only one colour parameter (since all the colours are the same). The drawing origin (0,0) = bottom left corner. Cubies are drawn from left to right. NOTE: this macro is SAME as the internal command \DrawRubikCubeFrontFace which is used for drawing the front face of a cube.

```
\newcommand{\DrawRubikLayerFace}[9]{%
%%----------FRONT face---------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(0,2) -- (0, 3) -- (1,3) -- (1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#2]%
(1,2) -- (1, 3) -- (2,3) -- (2,2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#3]%
(2,2) -- (2, 3) -- (3,3) -- (3,2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=#4]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#5]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#6]%
(2,1) -- (2, 2) -- (3,2) -- (3,1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=#7]%
(0,0) -- (0, 1) -- (1,1) -- (1,0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#8]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#9]%
(2,0) -- (2, 1) -- (3,1) -- (3,0) -- cycle;
}
\newcommand{\DrawRubikLayerFaceAll}[1]{%
%%---------FRONT face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(0,2) -- (0, 3) -- (1,3) -- (1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(1,2) -- (1, 3) -- (2,3) -- (2,2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(2,2) -- (2, 3) -- (3,3) -- (3,2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(2,1) -- (2, 2) -- (3,2) -- (3,1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(0,0) -- (0, 1) -- (1,1) -- (1,0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
(2,0) -- (2, 1) -- (3,1) -- (3,0) -- cycle;
}
```

16.9 Flat commands

`\FlatUp` `\FlatDown` `\FlatLeft` `\FlatRight` `\FlatFront` `\FlatBack`

Each of these six internal 'Flat' commands draws a separate face. Each command (except `\FlatFront`) takes two arguments, namely an X-coordinate and Y-coordinate of the bottom left corner of the face. This (X,Y) pair allows the user to position the face. Note that the `\FlatFront` command takes no arguments, since by definition the bottom left corner of this face is (0,0).

EXAMPLE: The following command positions the Up face so that its bottom left corner is located at (0,3):

```
\FlatUp{0}{3}
```

These internal commands are used by the commands `\DrawRubikFlat` and `\DrawRubikCubeFlat` to draw various flat representations of a Rubik's cube.

```
\newcommand{\FlatUp}[2]{%
\pgfmathsetmacro{\ux}{#1}%
\pgfmathsetmacro{\uy}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Ult]%
(\ux + 0,\uy + 2) -- (\ux + 0,\uy + 3) -- (\ux + 1,\uy + 3)%
 -- (\ux + 1,\uy + 2)  -- cycle;
```

```
\draw[line join=round,line cap=round,ultra thick,fill=\Umt]%
(\ux + 1,\uy + 2) -- (\ux + 1,\uy + 3) -- (\ux + 2,\uy + 3)%
 -- (\ux + 2,\uy + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urt]%
(\ux + 2,\uy + 2) -- (\ux + 2,\uy + 3) -- (\ux + 3,\uy + 3)%
 -- (\ux + 3,\uy + 2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulm]%
(\ux + 0,\uy + 1) -- (\ux + 0,\uy + 2) -- (\ux + 1,\uy + 2)%
 -- (\ux + 1,\uy + 1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umm]%
(\ux + 1,\uy + 1) -- (\ux + 1,\uy + 2) -- (\ux + 2,\uy + 2)%
 -- (\ux + 2,\uy + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urm]%
(\ux + 2,\uy + 1) -- (\ux + 2,\uy + 2) -- (\ux + 3,\uy + 2)%
 -- (\ux + 3,\uy + 1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulb]%
(\ux + 0,\uy + 0) -- (\ux + 0,\uy + 1) -- (\ux + 1,\uy + 1)%
 -- (\ux + 1,\uy + 0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Umb]%
(\ux + 1,\uy + 0) -- (\ux + 1,\uy + 1) -- (\ux + 2,\uy + 1)%
 -- (\ux + 2,\uy + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urb]%
(\ux + 2,\uy + 0) -- (\ux + 2,\uy + 1) -- (\ux + 3,\uy + 1)%
 -- (\ux + 3,\uy + 0) -- cycle;
}
%%------------------------
\newcommand{\FlatDown}[2]{%
\pgfmathsetmacro{\ddx}{#1}%
\pgfmathsetmacro{\ddy}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlt]%
(\ddx + 0,\ddy + 2) -- (\ddx + 0,\ddy + 3) -- (\ddx + 1,\ddy + 3)%
 -- (\ddx + 1,\ddy + 2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmt]%
(\ddx + 1,\ddy + 2) -- (\ddx + 1,\ddy + 3) -- (\ddx + 2,\ddy + 3)%
 -- (\ddx + 2,\ddy + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drt]%
(\ddx + 2,\ddy + 2) -- (\ddx + 2,\ddy + 3) -- (\ddx + 3,\ddy + 3)%
 -- (\ddx + 3,\ddy + 2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlm]%
(\ddx + 0,\ddy + 1) -- (\ddx + 0,\ddy + 2) -- (\ddx + 1,\ddy + 2)%
 -- (\ddx + 1,\ddy + 1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmm]%
(\ddx + 1,\ddy + 1) -- (\ddx + 1,\ddy + 2) -- (\ddx + 2,\ddy + 2)%
 -- (\ddx + 2,\ddy + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drm]%
(\ddx + 2,\ddy + 1) -- (\ddx + 2,\ddy + 2) -- (\ddx + 3,\ddy + 2)%
```

```
 -- (\ddx + 3,\ddy + 1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb]%
(\ddx + 0,\ddy + 0) -- (\ddx + 0,\ddy + 1) -- (\ddx + 1,\ddy + 1)%
 -- (\ddx + 1,\ddy + 0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Dmb]%
(\ddx + 1,\ddy + 0) -- (\ddx + 1,\ddy + 1) -- (\ddx + 2,\ddy + 1)%
 -- (\ddx + 2,\ddy + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Drb]%
(\ddx + 2,\ddy + 0) -- (\ddx + 2,\ddy + 1) -- (\ddx + 3,\ddy + 1)%
 -- (\ddx + 3,\ddy + 0) -- cycle;
}
%%-----------------------
\newcommand{\FlatFront}{%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Flt]%
(0,2) -- (0, 3) -- (1,3) -- (1,2)  -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Fmt]%
(1,2) -- (1, 3) -- (2,3) -- (2,2) -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Frt]%
(2,2) -- (2, 3) -- (3,3) -- (3,2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Flm]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Fmm]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Frm]%
(2,1) -- (2, 2) -- (3,2) -- (3,1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Flb]%
(0,0) -- (0, 1) -- (1,1) -- (1,0)  -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Fmb]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Frb]%
(2,0) -- (2, 1) -- (3,1) -- (3,0) -- cycle;
}
%%-------------------------
\newcommand{\FlatBack}[2]{%
\pgfmathsetmacro{\bx}{#1}%
\pgfmathsetmacro{\by}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Blt]%
(\bx + 0,\by + 2) -- (\bx + 0,\by + 3) -- (\bx + 1,\by + 3)%
 -- (\bx + 1,\by + 2)  -- cycle;
```

```
\draw[line join=round,line cap=round,ultra thick,fill=\Bmt]%
(\bx + 1,\by + 2) -- (\bx + 1,\by + 3) -- (\bx + 2,\by + 3)%
 -- (\bx + 2,\by + 2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Brt]%
(\bx + 2,\by + 2) -- (\bx + 2,\by + 3) -- (\bx + 3,\by + 3)%
 -- (\bx + 3,\by + 2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Blm]%
(\bx + 0,\by + 1) -- (\bx + 0,\by + 2) -- (\bx + 1,\by + 2)%
 -- (\bx + 1,\by + 1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Bmm]%
(\bx + 1,\by + 1) -- (\bx + 1,\by + 2) -- (\bx + 2,\by + 2)%
 -- (\bx + 2,\by + 1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Brm]%
(\bx + 2,\by + 1) -- (\bx + 2,\by + 2) -- (\bx + 3,\by + 2)%
 -- (\bx + 3,\by + 1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Blb]%
(\bx + 0,\by + 0) -- (\bx + 0,\by + 1) -- (\bx + 1,\by + 1)%
 -- (\bx + 1,\by + 0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Bmb]%
(\bx + 1,\by + 0) -- (\bx + 1,\by + 1) -- (\bx + 2,\by + 1)%
 -- (\bx + 2,\by + 0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Brb]%
(\bx + 2,\by + 0) -- (\bx + 2,\by + 1) -- (\bx + 3,\by + 1)%
 -- (\bx + 3,\by + 0) -- cycle;
}
%%-------------------------
\newcommand{\FlatLeft}[2]{%
\pgfmathsetmacro{\lx}{#1}%
\pgfmathsetmacro{\ly}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(\lx + 0,2) -- (\lx + 0, 3) -- (\lx + 1,3) -- (\lx + 1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmt]%
(\lx + 1,2) -- (\lx + 1, 3) -- (\lx + 2,3) -- (\lx + 2,2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrt]%
(\lx + 2,2) -- (\lx + 2, 3) -- (\lx + 3,3) -- (\lx + 3,2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Llm]%
(\lx + 0,1) -- (\lx + 0, 2) -- (\lx + 1,2) -- (\lx + 1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmm]%
(\lx + 1,1) -- (\lx + 1, 2) -- (\lx + 2,2) -- (\lx + 2,1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrm]%
(\lx + 2,1) -- (\lx + 2, 2) -- (\lx + 3,2) -- (\lx + 3,1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Llb]%
(\lx + 0,0) -- (\lx + 0, 1) -- (\lx + 1,1) -- (\lx + 1,0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lmb]%
(\lx + 1,0) -- (\lx + 1, 1) -- (\lx + 2,1) -- (\lx + 2,0) -- cycle;
```

```
\draw[line join=round,line cap=round,ultra thick,fill=\Lrb]%
(\lx + 2,0) -- (\lx + 2, 1) -- (\lx + 3,1) -- (\lx + 3,0) -- cycle;
}
%%--------------------------
\newcommand{\FlatRight}[2]{%
\pgfmathsetmacro{\rx}{#1}% %3
\pgfmathsetmacro{\ry}{#2}% %0
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(\rx + 0,2) -- (\rx + 0, 3) -- (\rx + 1,3) -- (\rx + 1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmt]%
(\rx + 1,2) -- (\rx + 1, 3) -- (\rx + 2,3) -- (\rx + 2,2) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(\rx + 2,2) -- (\rx + 2, 3) -- (\rx + 3,3) -- (\rx + 3,2) -- cycle;
%%-----middle row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlm]%
(\rx + 0,1) -- (\rx + 0, 2) -- (\rx + 1,2) -- (\rx + 1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmm]%
(\rx + 1,1) -- (\rx + 1, 2) -- (\rx + 2,2) -- (\rx + 2,1) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrm]%
(\rx + 2,1) -- (\rx + 2, 2) -- (\rx + 3,2) -- (\rx + 3,1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(\rx + 0,0) -- (\rx + 0, 1) -- (\rx + 1,1) -- (\rx + 1,0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rmb]%
(\rx + 1,0) -- (\rx + 1, 1) -- (\rx + 2,1) -- (\rx + 2,0) -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(\rx + 2,0) -- (\rx + 2, 1) -- (\rx + 3,1) -- (\rx + 3,0) -- cycle;
}
```

`\DrawRubikFlat` Draws a standard flat representation of the Rubik's cube (colours only)

```
\newcommand{\DrawRubikFlat}{%
  \FlatUp{0}{3}%
  \FlatDown{0}{-3}%
  \FlatLeft{-3}{0}%
  \FlatFront%
  \FlatRight{3}{0}%
  \FlatBack{6}{0}%
}
```

`\DrawRubikCubeFlat` Draws a Rubik's cube together with the three hidden faces (colours only)

```
\newcommand{\DrawRubikCubeFlat}{%
  \DrawRubikCube%
  \FlatDown{0}{-3}%
  \FlatLeft{-3}{0}%
  \FlatBack{4}{1}%
}
```

16.10 SideBar commands

SideBar commands draw narrow bars of colour indicating the side colours of each of the cubies forming the side of a given layer (face).

`\RubikSideBarWidth`
`\RubikSideBarLength`
`\RubikSideBarSep`

These three commands allow the user to set the Width, Length and Separation parameters for the sidebar (in decimal values, where 1 is equivalent to the length of the side of a cubie).

```
\newcommand{\RubikSideBarWidth}[1]{\pgfmathsetmacro{\bw}{#1}}
\newcommand{\RubikSideBarLength}[1]{\pgfmathsetmacro{\bl}{#1}}
\newcommand{\RubikSideBarSep}[1]{\pgfmathsetmacro{\bs}{#1}}
```

We first set some default values

```
\RubikSideBarWidth{0.3}%
\RubikSideBarLength{1}%
\RubikSideBarSep{0.3}%
```

16.10.1 Allocating a colour to a single cubie sidebar

`\side@barT`
`\side@barB`
`\side@barL`
`\side@barR`

Internal commands. Full length face SideBars are really multiple instances of single cubie bars, each of which is drawn using one of four internal SideBar commands—one for each of the sides which we shall call Top, Bottom, Left, Right. Each SideBar command takes two arguments: one for cubie position {$\langle 1 \mid 2 \mid 3\rangle$} and one for the colour-code {$\langle R \mid O \mid Y \mid G \mid B \mid W \mid X\rangle$}.

EXAMPLE: the following command allocates a colour to a single cubie sidebar on the Left of a Rubik face:

`\side@barL{`⟨*cubie-position*⟩`}{`⟨*colour-code*⟩`}`

There are three cubie positions on each of the four sides of a face, and these are numbered 1 to 3 starting from the bottom left corner (1,1). The SideBar command also implements the set (or default) Length (`\bl`), Width (`\bw`) and Separation (`\bs`) values mentioned above. `\blh` = Half `\bl` = `\bl`/2. Note that the TikZ `\pgfmathsetmacro` commands (which do the maths) must be inside the TeX sidebar command in order to work. The start point of Draw command for each bar rectangle is botton Left corner of the bar =(`\dx`,`\dy`)

```
\newcommand{\side@barL}[2]{%
%% #1 = cubie possn no, #2 = colour
\pgfmathsetmacro{\blh}{\bl*(0.5)}%
\pgfmathsetmacro{\dx}{0 - \bs - \bw}%
\pgfmathsetmacro{\dy}{#1-1+0.5-\blh}%
\draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bl)
  -- (\dx+\bw,\dy+\bl) -- (\dx+\bw,\dy) -- cycle;
}
\newcommand{\side@barR}[2]{%
%% #1 = cubie possn no, #2 = colour
\pgfmathsetmacro{\blh}{\bl*(0.5)}%
\pgfmathsetmacro{\dx}{3 + \bs}%
\pgfmathsetmacro{\dy}{#1 -1+0.5-\blh}%
\draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bl)
  -- (\dx+\bw,\dy+\bl) -- (\dx+\bw,\dy) -- cycle;
}
\newcommand{\side@barT}[2]{%
%% #1 = cubie possn no, #2 = colour
\pgfmathsetmacro{\blh}{\bl*(0.5)}%
\pgfmathsetmacro{\dx}{#1 -1+0.5-\blh}%
\pgfmathsetmacro{\dy}{3 +\bs}%
\draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bw)
  -- (\dx+\bl,\dy+\bw) -- (\dx+\bl,\dy) -- cycle;
}
\newcommand{\side@barB}[2]{%
%% #1 = cubie possn no, #2 = colour
\pgfmathsetmacro{\blh}{\bl*(0.5)}%
\pgfmathsetmacro{\dx}{#1 -1+0.5-\blh}%
\pgfmathsetmacro{\dy}{0 -\bs-\bw}%
\draw[fill=#2] (\dx,\dy) -- (\dx,\dy + \bw)
  -- (\dx+\bl,\dy+\bw) -- (\dx+\bl,\dy) -- cycle;
}
```

16.10.2 Drawing a single cubie sidebar

\DrawRubikLayerSide$X_1X_2X_3$

This command draws the sidebars using the above `\sidebar` command. The $X_1X_2X_2$ parameters refer to the options Left, Middle, Right, Top, Equator, Bottom, x, y, as follows:

> X_1 is an x parameter: either $\langle L \mid M \mid R \rangle$
> X_2 is an y parameter: either $\langle T \mid E \mid B \rangle$
> X_3 is an extra parameter: either $\langle x \mid y \rangle$, required by corner sidebars to indicate whether the sidebar was either above or below $\langle y \rangle$, or to the left or right $\langle x \rangle$ of the associated cubie. An X_3 parameter is not required for the sidebar of an edge cubie, since only one location is posible in these cases.

For example, the following command

```
\DrawRubikLayerSideLTy{G}
```

draws a Green sidebar above the Top Left cubie.

```
%%---Left side
\newcommand{\DrawRubikLayerSideLTx}[1]{\side@barL{3}{#1}}
\newcommand{\DrawRubikLayerSideLEx}[1]{\side@barL{2}{#1}}
\newcommand{\DrawRubikLayerSideLE}[1]{\side@barL{2}{#1}}
\newcommand{\DrawRubikLayerSideLBx}[1]{\side@barL{1}{#1}}
%---Right side
\newcommand{\DrawRubikLayerSideRTx}[1]{\side@barR{3}{#1}}
\newcommand{\DrawRubikLayerSideREx}[1]{\side@barR{2}{#1}}
\newcommand{\DrawRubikLayerSideRE}[1]{\side@barR{2}{#1}}
```

```
\newcommand{\DrawRubikLayerSideRBx}[1]{\side@barR{1}{#1}}
%---Top side
\newcommand{\DrawRubikLayerSideLTy}[1]{\side@barT{1}{#1}}
\newcommand{\DrawRubikLayerSideMTy}[1]{\side@barT{2}{#1}}
\newcommand{\DrawRubikLayerSideMT}[1]{\side@barT{2}{#1}}
\newcommand{\DrawRubikLayerSideRTy}[1]{\side@barT{3}{#1}}
%---Bottom side
\newcommand{\DrawRubikLayerSideLBy}[1]{\side@barB{1}{#1}}
\newcommand{\DrawRubikLayerSideMBy}[1]{\side@barB{2}{#1}}
\newcommand{\DrawRubikLayerSideMB}[1]{\side@barB{2}{#1}}
\newcommand{\DrawRubikLayerSideRBy}[1]{\side@barB{3}{#1}}
```

16.10.3 Drawing multiple cubie sidebars

`\DrawRubikLayerSideT` `\DrawRubikLayerSideB` `\DrawRubikLayerSideL` `\DrawRubikLayerSideR`

These commands allow the drawing of 3 small sidebars along one particular side (Top, Bottom, Left, Right), as indicated by the appended T, B, L, R letter code. Each command takes three ordered colour arguments, which are ordered either from left to right (the T and B forms), or from top to bottom (the L and R forms)

```
%%--Top side---
\newcommand{\DrawRubikLayerSideT}[3]{%
  \DrawRubikLayerSideLTy{#1}%
  \DrawRubikLayerSideMTy{#2}%
  \DrawRubikLayerSideRTy{#3}%
}
%%--Bottom side---
\newcommand{\DrawRubikLayerSideB}[3]{%
  \DrawRubikLayerSideLBy{#1}%
  \DrawRubikLayerSideMBy{#2}%
  \DrawRubikLayerSideRBy{#3}%
}
%%--Left side--------
%% colours run vertically DOWN
\newcommand{\DrawRubikLayerSideL}[3]{%
  \DrawRubikLayerSideLTx{#1}%
  \DrawRubikLayerSideLEx{#2}%
  \DrawRubikLayerSideLBx{#3}%
}
%%--Right side--------
%% colours run vertically DOWN
\newcommand{\DrawRubikLayerSideR}[3]{%
  \DrawRubikLayerSideRTx{#1}%
  \DrawRubikLayerSideREx{#2}%
  \DrawRubikLayerSideRBx{#3}%
}
```

`\DrawRubikLayerSideLR`

This command draws six cubie sidebars, three on each side, drawn in (L, R) pairs. The command takes six colour arguments, ordered in pairs, as shown in the following example.

```
        \DrawRubikLayerSideLR{G}   {G}
                             {R}   {B}
                             {Y}   {B}
```

```
\newcommand{\DrawRubikLayerSideLR}[6]{%
  \DrawRubikLayerSideLTx{#1}%
  \DrawRubikLayerSideRTx{#2}%
  \DrawRubikLayerSideLEx{#3}%
  \DrawRubikLayerSideREx{#4}%
  \DrawRubikLayerSideLBx{#5}%
  \DrawRubikLayerSideRBx{#6}%
}
```

16.11 NCube command

HISTORY: The essence of this command was originally developed by Peter Bartal as his command `\rubikcube` (see Bartal, 2011). We have modified it, as follows (June 2012):

(1) adjusted to use the TikZ `\pgfmathsetmacro`{}{} command,

(2) renamed to `\DrawNCubeAll`.

`\DrawNCubeAll` This command draws a solved NxNxN Rubik's cube from the RightUp viewpoint. All cubies on a given face have the same colour. The command takes four ordered arguments, as follows:

#1 = number of cubies ($n > 0$) along each side,

#2, #3, #4 = colors of the visible faces (in X,Y,Z order); X=Right face colour, Y=Up face colour, Z=Front face colour.

```
\newcommand{\DrawNCubeAll}[4]{%
   \pgfmathsetmacro{\ncubes}{#1-1}%
%% need to subtract 1 from the given number of cubies per side
%% to avoid the origin of the initial cube to be displaced
    \foreach \x in {0,...,\ncubes}{%
      \foreach \y in {0,...,\ncubes}{%
        \foreach \z in {0,...,\ncubes}{%
          \cube@dxdydz{1}{#2}{#3}{#4}{\x}{\y}{\z}%
  }}}}
```

`\cube@dxdydz` This internal command is used only by the `\DrawNCubeAll` command. The original version of this command was developed by Peter Bartal (see Bartal, 2011). It was modified (2012) by RWD Nickalls (to implement a more intuitive X, Y, Z ordering of parameters), and also by Peter Grill (see below).

The cube need not be in the origin, the distances of the Down-behind [L] corner from the origin are taken as parameters. The command takes 7 ordered arguments:

1 - length of an edge

2 - color of the X face (Right) (RWDN)

3 - color of the Y face (Up) (RWDN)

4 - color of the Z face (Front) (RWDN)

5 - x-position in space
6 - y-position in space
7 - z-position in space
USAGE: `\cube@dxdydz{1}{cX}{cY}{cZ}{x}{y}{z}`

The original code `\pgfmathparse{#1+#5}\let\dy\pgfmathresult` was changed to the more intuitive `\pgfmathsetmacro{\dx}{#1+#5}` (suggested by Peter Grill 2011).

CHANGES: RWD Nickalls (2012): (1) added the [line join=round,line cap=round] options to each of the TikZ 'draw' commands, in order to improve the line joining (first two options); (2) adjusted the `\cube@dxdydz` macro to adopt the ordered XYZ face colour notation (by reassigning #2, #3, #4).

```
\newcommand{\cube@dxdydz}[7]{%
   \pgfmathsetmacro{\dx}{#1+#5}%
%% calculates the 'displacement' (distance from the origin) of the
%% far corners of the cube along the x axis from the arguments
   \pgfmathsetmacro{\dy}{#1+#6}%
%% calculates the 'displacement' (distance from the origin) of the
%% far corners of the cube along the y axis from the arguments
   \pgfmathsetmacro{\dz}{#1+#7}%
%% calculates the 'displacement' (distance from the origin) of the
%% far corners of the cube along the z axis from the arguments
%% Draw FRONT face   (using the X colour = #4)
    \draw[line join=round,line cap=round,ultra thick,fill=#4]%
 (#5,#6,\dz) -- (\dx,#6,\dz) -- (\dx,\dy,\dz) -- (#5,\dy,\dz) -- cycle;
%% The 'rectangle' command does not work with 3D coordinates,
%% so this is the way to draw the squres with space coordinates
%% Draw UP face (using the Y colour = #3)
   \draw[line join=round,line cap=round,ultra thick,fill=#3]%
 (#5,\dy,\dz) -- (\dx,\dy,\dz) -- (\dx,\dy,#7) -- (#5,\dy,#7) -- cycle;
%% Draw RIGHT face   (using the X colour = #2)
 \draw[line join=round,line cap=round,ultra thick,fill=#2]%
 (\dx,#6,\dz) -- (\dx,#6,#7) -- (\dx,\dy,#7) -- (\dx,\dy,\dz) -- cycle;
    }
```

16.12 Drawing single cubies

`\Cubiedx`
`\Cubiedy`

These two commands set the value of the two length parameters `cx` and `cy`, and allow the user to vary the size (adjust `cy`) and horizontal viewpoint (adjust `cx`) of a single cubie (described in more detail in the rubikcube package documentation). Note that we cannot use the names `dx`, `dy` since these have been used before (see above). However we use `dx`, `dy` in the command names since these will be more readily understood by the user.

```
\newcommand{\Cubiedx}[1]{\pgfmathsetmacro{\cx}{#1}}
\newcommand{\Cubiedy}[1]{\pgfmathsetmacro{\cy}{#1}}
```

We now set the default values (cx=cy=0.4)

```
\Cubiedx{0.4}
\Cubiedy{0.4}
```

`\DrawCubieRU`
`\DrawCubieRD`
`\DrawCubieLU`
`\DrawCubieLD`

These four commands draw a single cubie from the RightUp, RightDown, LeftUp, LeftDown viewpoint. The viewpoint is specified using an appended two-letter XY ordered viewpoint code: either RU, RD, LU, LD. These commands take three arguments, namely three different XYZ ordered colour codes (R,O,Y,G,B,W,X).

FORMAT: `\DrawCubieRU{`⟨*colourX*⟩`}{`⟨*colourY*⟩`}{`⟨*colourZ*⟩`}`

USAGE: `\DrawCubieRU{G}{B}{W}`

```
\newcommand{\DrawCubieRU}[3]{%
%% Front face (z)
\draw[line join=round,line cap=round,ultra thick,fill=#3]%
 (0,0) -- (0, 1) -- (1, 1) -- (1,0) -- cycle;
%% Up face(y)
\draw[line join=round,line cap=round,ultra thick,fill=#2]%
 (0,1) -- (\cx, 1+\cy) -- (1+\cx,1+\cy) -- (1,1) -- cycle;
%% Right face(x)
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
 (1,0) -- (1,1) -- (1+\cx,1+\cy) -- (1+\cx, \cy) -- cycle;
}
\newcommand{\DrawCubieRD}[3]{%
%% Front face (z)
\draw[line join=round,line cap=round,ultra thick,fill=#3]%
 (0,0) -- (0, 1) -- (1, 1) -- (1,0) -- cycle;
%% Down face (y)
\draw[line join=round,line cap=round,ultra thick,fill=#2]%
 (\cx,-\cy) -- (0, 0) -- (1,0) -- (1+\cx,-\cy) -- cycle;
%% Right face (x)
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
 (1,0) -- (1,1) -- (1+\cx,-\cy+1) -- (1+\cx, -\cy) -- cycle;
}
\newcommand{\DrawCubieLD}[3]{%
%% Front face (z)
\draw[line join=round,line cap=round,ultra thick,fill=#3]%
 (0,0) -- (0, 1) -- (1, 1) -- (1,0) -- cycle;
%% Down face (y)
\draw[line join=round,line cap=round,ultra thick,fill=#2]%
 (-\cx,-\cy) -- (0, 0) -- (1,0) -- (1-\cx,-\cy) -- cycle;
%% Left face (x)
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
 (-\cx,-\cy) -- (-\cx,-\cy+1) -- (0,1) -- (0,0) -- cycle;
}
\newcommand{\DrawCubieLU}[3]{%
%% Front face (z)
\draw[line join=round,line cap=round,ultra thick,fill=#3]%
 (0,0) -- (0, 1) -- (1, 1) -- (1,0) -- cycle;
%% Up face (y)
\draw[line join=round,line cap=round,ultra thick,fill=#2]%
 (-\cx,1+\cy) -- (1-\cx, 1+\cy) -- (1,1) -- (0,1) -- cycle;
%% Left face (x)
\draw[line join=round,line cap=round,ultra thick,fill=#1]%
 (-\cx, \cy) -- (-\cx,\cy+1) -- (0,1) -- (0,0) -- cycle;
}
```

16.13 Text cubies

\textCubieRU
\textCubieRD
\textCubieLU
\textCubieLD

These four commands draw a single ‘text’ cubie from the RightUp, RightDown, LeftUp, LeftDown viewpoint. They are ‘text’ forms of the DrawCubie’ versions described above. Their size is scaled to be suitable for use with 10–12 point fonts.

As before, the viewpoint is specified using an appended two-letter XY ordered viewpoint code: either RU, RD, LU, LD. These commands take three arguments, namely three different XYZ ordered colour codes (R,O,Y,G,B,W,X).

FORMAT: `\textCubieRU{`⟨*colourX*⟩`}{`⟨*colourY*⟩`}{`⟨*colourZ*⟩`}`

USAGE: `\textCubieRU{G}{B}{W}`

```
\newcommand{\textCubieRU}[3]{%
\begin{minipage}{0.66cm}
\centering
\begin{tikzpicture}[scale=0.5]
\Cubiedx{0.4}\Cubiedy{0.4}
\DrawCubieRU{#1}{#2}{#3}
\end{tikzpicture}
\end{minipage}
}
\newcommand{\textCubieRD}[3]{%
\begin{minipage}{0.66cm}
\centering
\begin{tikzpicture}[scale=0.5]
\Cubiedx{0.4}\Cubiedy{0.4}
\DrawCubieRD{#1}{#2}{#3}
\end{tikzpicture}
\end{minipage}
}
\newcommand{\textCubieLD}[3]{%
\begin{minipage}{0.66cm}
\centering
\begin{tikzpicture}[scale=0.5]
\Cubiedx{0.4}\Cubiedy{0.4}
\DrawCubieLD{#1}{#2}{#3}
\end{tikzpicture}
\end{minipage}
}
\newcommand{\textCubieLU}[3]{%
\begin{minipage}{0.66cm}
\centering
\begin{tikzpicture}[scale=0.5]
\Cubiedx{0.4}\Cubiedy{0.4}
\DrawCubieLU{#1}{#2}{#3}
\end{tikzpicture}
\end{minipage}
}
```

16.14 Rotation commands — face and slice

NOTATION: We require a special notation to denote the various Rubik rotations (denoted by the stem rr). These rotations can be denoted either using a letter code say, `\rrR` for a Right face rotation, or by the equivalent hieroglyph denoted as `\rrhR` (Rubik rotation hieroglyph Right) etc. Commands such as `\RubikXXXX` typeset a pair of rr and rrh vertically. Commands such as `\textRubikXXXX` typeset the pair rr and rrh horizontally (ie a sort of 'text' mode).

The square hieroglyphs are built up in stages. We first create an internal command for drawing the square (`\DrawNotationBox`), and then draw the contents (lines, arrows, arcs of circles, and sometimes just text). (for the TikZ ARC command see TikZ pgfmanual2012 page p146 (§14.8)).

The presence of small overfilled hboxes associated with these squares were originally checked using the ltugboat.cls, and all fixed mainly by setting the associated minipages → width = 0.6cm, and using TKZ scale=0.5. NOTE: do not use a trailing `%` after each `\end{tikzpicture}` as we actually need a small space to appear when used in text.

`\DrawNotationBox` This command draws the surrounding square box of all the hieroglyphs. Note that we start at (0,0) and draw to the final point in order to make a nice corner join.

```
\newcommand{\DrawNotationBox}{%
  \draw [thick] (0,0) -- (0,1) -- (1,1) -- (1,0) -- (0,0) -- (0,1)%
}
```

We now define a number of points and line-segments inside the square (eg.; `\sd`, `\sh` ... etc.) which will be required for use in drawing the various lines and arrows. Some hieroglyphs contain either one circular arc, or two concentric arcs, and these arcs require both a center and a start point.

```
\pgfmathsetmacro{\sd}{0.25} % horiz space
\pgfmathsetmacro{\sdd}{2*\sd}
\pgfmathsetmacro{\sddd}{3*\sd}
\pgfmathsetmacro{\sh}{0.6} % height
\pgfmathsetmacro{\sb}{0.2} % base
\pgfmathsetmacro{\sbh}{\sb + \sh} % UP
\pgfmathsetmacro{\scx}{\sdd+0.2} % Start of CircleX
\pgfmathsetmacro{\scy}{\sd*2/3}  % Start of CircleY
\pgfmathsetmacro{\sqcx}{\scx-0.13} %% SQuare CenterX
\pgfmathsetmacro{\sqcy}{\scy+0.25} %% SQuare CenterY
```

16.14.1 Rotation B

`\rrB` `\SquareB` `\rrhB` `\RubikB` `\textRubikB` These commands all draw forms which denote the B rotation. Not visible from the front.

```
\newcommand{\rrB}{\@rr{B}}
\newcommand{\SquareB}{\@SquareLetter{\rrB}}
\newcommand{\rrhB}{\raisebox{-0.25mm}{\SquareB}\,}
\newcommand{\RubikB}{\raisebox{\@hRubik}{\SquareB}\,}
\newcommand{\textRubikB}{\rrhB\,}
```

16.14.2 Rotation Bp

\rrBp
\SquareBp
\rrhBp
\RubikBp
\textRubikBp

These commands all draw forms which denote the Bp rotation. Not visible from the front.

```
\newcommand{\rrBp}{\@rrp{B}}
\newcommand{\SquareBp}{\@SquareLetter{\rrBp}}
\newcommand{\rrhBp}{\raisebox{-0.25mm}{\SquareBp}\,}
\newcommand{\RubikBp}{\raisebox{\@hRubik}{\SquareBp}\,}
\newcommand{\textRubikBp}{\rrhBp\,}
```

16.14.3 Rotation Bw

\rrBw
\SquareBw
\rrhBw
\RubikBw
\textRubikBw

These commands all draw forms which denote the Bw rotation. Not visible from the front.

```
%\newcommand{\rrBw}{\textbf{\textsf{Bw}}}
\newcommand{\rrBw}{\@rrw{B}}
\newcommand{\SquareBw}{\@SquareLetter{\rrBw}}
\newcommand{\rrhBw}{\raisebox{-0.25mm}{\SquareBw}\,}
\newcommand{\RubikBw}{\raisebox{\@hRubik}{\SquareBw}\,}
\newcommand{\textRubikBw}{\rrhBw\,}
```

16.14.4 Rotation Bwp

\rrBwp
\SquareBwp
\rrhBwp
\RubikBwp
\textRubikBwp

These commands all draw forms which denote the Bwp rotation. Not visible from the front.

```
%\newcommand{\rrBwp}{\textbf{\textsf{Bwp}}}
\newcommand{\rrBwp}{\@rrwp{B}}
\newcommand{\SquareBwp}{\@SquareLetter{\rrBwp}}
\newcommand{\rrhBwp}{\raisebox{-0.25mm}{\SquareBwp}\,}
\newcommand{\RubikBwp}{\raisebox{\@hRubik}{\SquareBwp}\,}
\newcommand{\textRubikBwp}{\rrhBwp\,}
```

16.14.5 Rotation Bs

\rrBs
\SquareBs
\rrhBs
\RubikBs
\textRubikBs

These commands all draw forms which denote the Bs rotation. Not visible from the front.

```
%\newcommand{\rrBs}{\textbf{\textsf{Bs}}}
\newcommand{\rrBs}{\@rrs{B}}
\newcommand{\SquareBs}{\@SquareLetter{\rrBs}}
\newcommand{\rrhBs}{\raisebox{-0.25mm}{\SquareBs}\,}
\newcommand{\RubikBs}{\raisebox{\@hRubik}{\SquareBs}\,}
\newcommand{\textRubikBs}{\rrhBs\,}
```

16.14.6 Rotation Bsp

\rrBsp
\SquareBsp
\rrhBsp
\RubikBsp
\textRubikBsp

These commands all draw forms which denote the Bsp rotation. Not visible from the front.

```
%\newcommand{\rrBsp}{\textbf{\textsf{Bsp}}}
```

```
\newcommand{\rrBsp}{\@rrsp{B}}
\newcommand{\SquareBsp}{\@SquareLetter{\rrBsp}}
\newcommand{\rrhBsp}{\raisebox{-0.25mm}{\SquareBsp}\,}
\newcommand{\RubikBsp}{\raisebox{\@hRubik}{\SquareBsp}\,}
\newcommand{\textRubikBsp}{\rrhBsp\,}
```

16.14.7 Rotation Ba

\rrBa
\SquareBa
\rrhBa
\RubikBa
\textRubikBa

These commands all draw forms which denote the Ba rotation. Not visible from the front.

```
%\newcommand{\rrBa}{\textbf{\textsf{Ba}}}
\newcommand{\rrBa}{\@rra{B}}
\newcommand{\SquareBa}{\@SquareLetter{\rrBa}}
\newcommand{\rrhBa}{\raisebox{-0.25mm}{\SquareBa}\,}
\newcommand{\RubikBa}{\raisebox{\@hRubik}{\SquareBa}\,}
\newcommand{\textRubikBa}{\rrhBa\,}
```

16.14.8 Rotation Bap

\rrBap
\SquareBap
\rrhBap
\RubikBap
\textRubikBap

These commands all draw forms which denote the Bap rotation. Not visible from the front.

```
%\newcommand{\rrBap}{\textbf{\textsf{Bap}}}
\newcommand{\rrBap}{\@rrap{B}}
\newcommand{\SquareBap}{\@SquareLetter{\rrBap}}
\newcommand{\rrhBap}{\raisebox{-0.25mm}{\SquareBap}\,}
\newcommand{\RubikBap}{\raisebox{\@hRubik}{\SquareBap}\,}
\newcommand{\textRubikBap}{\rrhBap\,}
```

16.14.9 Rotation D

\rrD
\SquareD
\rrhD
\RubikD
\textRubikD

These commands all draw forms which denote the D rotation.

```
\newcommand{\rrD}{\textbf{\textsf{D}}}
%%
\newcommand{\SquareD}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%%
}
\newcommand{\rrhD}{\raisebox{-0.333\height}{\SquareD}\,}
%%
\newcommand{\RubikD}{%
\begin{minipage}{0.6cm}
\centering
\SquareD\\
\rrD
\end{minipage}%
}
\newcommand{\textRubikD}{\rrD\,\rrhD}
```

16.14.10 Rotation Dp

\rrDp
\SquareDp
\rrhDp
\RubikDp
\textRubikDp

These commands all draw forms which denote the Dp rotation.

```
\newcommand{\rrDp}{\textbf{\textsf{D}$^\prime$}}
%%
\newcommand{\SquareDp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick]     (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd)  -- (\sbh, \sdd);
\draw [thick, <-] (\sb,\sd)   -- (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDp}{\raisebox{-0.333\height}{\SquareDp}\,}
%%
\newcommand{\RubikDp}{%
\begin{minipage}{0.6cm}
\centering
\SquareDp\\
\rrDp
\end{minipage}%
}
\newcommand{\textRubikDp}{\rrDp\,\rrhDp}
```

16.14.11 Rotation Dw

\rrDw
\SquareDw
\rrhDw
\RubikDw
\textRubikDw

These commands all draw forms which denote the Dw rotation.

```
\newcommand{\rrDw}{\textbf{\textsf{D\footnotesize{w}}}}
%%
\newcommand{\SquareDw}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick]     (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, ->] (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->] (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDw}{\raisebox{-0.333\height}{\SquareDw}\,}
%%
\newcommand{\RubikDw}{%
\begin{minipage}{0.6cm}
\centering
\SquareDw\\
\rrDw
\end{minipage}%
}
```

```
\newcommand{\textRubikDw}{\rrDw\,\rrhDw}
```

16.14.12 Rotation Dwp

\rrDwp \SquareDwp \rrhDwp \RubikDwp \textRubikDwp

These commands all draw forms which denote the Dwp rotation.

```
\newcommand{\rrDwp}{\textbf{\textsf{D\footnotesize{w}}$^\prime$}}
%%
\newcommand{\SquareDwp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick]     (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, <-] (\sb,\sdd)  -- (\sbh, \sdd);
\draw [thick, <-] (\sb,\sd)   -- (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDwp}{\raisebox{-0.333\height}{\SquareDwp}\,}
%%
\newcommand{\RubikDwp}{%
\begin{minipage}{0.6cm}
\centering
\SquareDwp\\
\rrDwp
\end{minipage}%
}
\newcommand{\textRubikDwp}{\rrDwp\,\rrhDwp}
```

16.14.13 Rotation Ds

\rrDs \SquareDs \rrhDs \RubikDs \textRubikDs

These commands all draw forms which denote the Ds rotation.

```
\newcommand{\rrDs}{\@rrs{D}}
%%
\newcommand{\SquareDs}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->]   (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDs}{\raisebox{-0.333\height}{\SquareDs}\,}
%%
\newcommand{\RubikDs}{%
\begin{minipage}{0.6cm}
\centering
\SquareDs\\
\rrDs
\end{minipage}%
}
\newcommand{\textRubikDs}{\rrDs\,\rrhDs}
```

16.14.14 Rotation Dsp

\rrDsp
\SquareDsp
\rrhDsp
\RubikDsp
\textRubikDsp

These commands all draw forms which denote the Dsp rotation.

```
\newcommand{\rrDsp}{\@rrsp{D}}
%%
\newcommand{\SquareDsp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, <-]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDsp}{\raisebox{-0.333\height}{\SquareDsp}\,}
%%
\newcommand{\RubikDsp}{%
\begin{minipage}{0.6cm}
\centering
\SquareDsp\\
\rrDsp
\end{minipage}%
}
\newcommand{\textRubikDsp}{\rrDsp\,\rrhDsp}
```

16.14.15 Rotation Da

\rrDa
\SquareDa
\rrhDa
\RubikDa
\textRubikDa

These commands all draw forms which denote the Da rotation.

```
\newcommand{\rrDa}{\@rra{D}}
%%
\newcommand{\SquareDa}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDa}{\raisebox{-0.333\height}{\SquareDa}\,}
%%
\newcommand{\RubikDa}{%
\begin{minipage}{0.6cm}
\centering
\SquareDa\\
\rrDa
\end{minipage}%
}
\newcommand{\textRubikDa}{\rrDa\,\rrhDa}
```

16.14.16 Rotation Dap

\rrDap
\SquareDap
\rrhDap
\RubikDap
\textRubikDap

These commands all draw forms which denote the Dap rotation.

```
\newcommand{\rrDap}{\@rrap{D}}
%%
\newcommand{\SquareDap}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, <-]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhDap}{\raisebox{-0.333\height}{\SquareDap}\,}
%%
\newcommand{\RubikDap}{%
\begin{minipage}{0.6cm}
\centering
\SquareDap\\
\rrDap
\end{minipage}%
}
\newcommand{\textRubikDap}{\rrDap\,\rrhDap}
```

16.14.17 Rotation E

\rrE
\SquareE
\rrhE
\RubikE
\textRubikE

These commands all draw forms which denote the E rotation.

```
\newcommand{\rrE}{\textbf{\textsf{E}}}
%%
\newcommand{\SquareE}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, ->]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhE}{\raisebox{-0.333\height}{\SquareE}\,}
%%
\newcommand{\RubikE}{%
\begin{minipage}{0.6cm}
\centering
\SquareE\\
\rrE
\end{minipage}%
}
\newcommand{\textRubikE}{\rrE\,\rrhE}
```

16.14.18 Rotation Ep

\rrEp
\SquareEp
\rrhEp
\RubikEp
\textRubikEp

These commands all draw forms which denote the Ep rotation.

```
\newcommand{\rrEp}{\textbf{\textsf{E}$^\prime$}}
%%
\newcommand{\SquareEp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, <-]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhEp}{\raisebox{-0.333\height}{\SquareEp}\,}
%%
\newcommand{\RubikEp}{%
\begin{minipage}{0.6cm}
\centering
\SquareEp\\
\rrEp
\end{minipage}%
}
\newcommand{\textRubikEp}{\rrEp\,\rrhEp}
```

16.14.19 Rotation F

\rrF
\SquareF
\rrhF
\RubikF
\textRubikF

These commands all draw forms which denote the F rotation.

```
\newcommand{\rrF}{\textbf{\textsf{F}}}
%%
\newcommand{\SquareF}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\scx, \scy) arc[radius=0.35, start angle=-60, delta angle=290];
\end{tikzpicture}%
}
\newcommand{\rrhF}{\raisebox{-0.333\height}{\SquareF}\,}
%%
\newcommand{\RubikF}{%
\begin{minipage}{0.6cm}
\centering
\SquareF\\
\rrF
\end{minipage}%
}
\newcommand{\textRubikF}{\rrF\,\rrhF}
```

16.14.20 Rotation Fp

These commands all draw forms which denote the Fp rotation.

\rrFp
\SquareFp
\rrhFp
\RubikFp
\textRubikFp

```
\newcommand{\rrFp}{\textbf{\textsf{F}$^\prime$}}
%%
\newcommand{\SquareFp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\scx, \scy) arc[radius=0.35, start angle=-60, delta angle=290];
\end{tikzpicture}%
}
\newcommand{\rrhFp}{\raisebox{-0.333\height}{\SquareFp}\,}
%%
\newcommand{\RubikFp}{%
\begin{minipage}{0.6cm}
\centering
\SquareFp\\
\rrFp
\end{minipage}%
}
\newcommand{\textRubikFp}{\rrFp\,\rrhFp}
```

16.14.21 Rotation Fw

\rrFw
\SquareFw
\rrhFw
\RubikFw
\textRubikFw

These commands all draw forms which denote the Fw rotation.

```
\newcommand{\rrFw}{\textbf{\textsf{F\footnotesize{w}}}}
%%
\newcommand{\SquareFw}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\scx, \scy) arc[radius=0.35, start angle=-60, delta angle=290];
\draw [thick] (\sqcx,\sqcy) arc[radius=0.1, start angle=-60, delta angle=360];
%\node (squareLab) at (0.5,0.5)  {$o$};
\end{tikzpicture}%
}
\newcommand{\rrhFw}{\raisebox{-0.333\height}{\SquareFw}\,}
%%
\newcommand{\RubikFw}{%
\begin{minipage}{0.6cm}
\centering
\SquareFw\\
\rrFw
\end{minipage}%
}
\newcommand{\textRubikFw}{\rrFw\,\rrhFw}
```

16.14.22 Rotation Fwp

\rrFwp
\SquareFwp
\rrhFwp
\RubikFwp
\textRubikFwp

These commands all draw forms which denote the Fwp rotation.

```
\newcommand{\rrFwp}{\textbf{\textsf{F\footnotesize{w}}$^\prime$}}
%%
\newcommand{\SquareFwp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\scx, \scy) arc[radius=0.35, start angle=-60, delta angle=290];
\draw [thick] (\sqcx,\sqcy) arc[radius=0.1, start angle=-60, delta angle=360];
\end{tikzpicture}%
}
\newcommand{\rrhFwp}{\raisebox{-0.333\height}{\SquareFwp}\,}
%%
\newcommand{\RubikFwp}{%
\begin{minipage}{0.6cm}
\centering
\SquareFwp\\
\rrFwp
\end{minipage}%
}
\newcommand{\textRubikFwp}{\rrFwp\,\rrhFwp}
%%
```

16.14.23 Rotation Fs

`\rrFs` `\rrhFs` `\RubikFs` `\textRubikFs`

These commands draw forms of the Singmaster Fs slice rotation. We need to just make square with Fs in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrFs}{\@rrs{F}}
\newcommand{\SquareFs}{\@SquareLetter{\rrFs}}
\newcommand{\rrhFs}{\raisebox{-0.25mm}{\SquareFs}\,}
\newcommand{\RubikFs}{\raisebox{\@hRubik}{\SquareFs}\,}
\newcommand{\textRubikFs}{\rrhFs\,}
```

16.14.24 Rotation Fsp

`\rrFsp` `\rrhFsp` `\RubikFsp` `\textRubikFsp`

These commands draw forms of the Singmaster Fsp slice rotation. We need to just make square with Fsp in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrFsp}{\@rrsp{F}}
\newcommand{\SquareFsp}{\@SquareLetter{\rrFsp}}
\newcommand{\rrhFsp}{\raisebox{-0.25mm}{\SquareFsp}\,}
\newcommand{\RubikFsp}{\raisebox{\@hRubik}{\SquareFsp}\,}
\newcommand{\textRubikFsp}{\rrhFsp\,}
```

16.14.25 Rotation Fa

`\rrFa` `\rrhFa` `\RubikFa` `\textRubikFa`

These commands draw forms of the Singmaster Fa slice rotation. We need to just make square with Fa in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrFa}{\@rra{F}}
\newcommand{\SquareFa}{\@SquareLetter{\rrFa}}
\newcommand{\rrhFa}{\raisebox{-0.25mm}{\SquareFa}\,}
\newcommand{\RubikFa}{\raisebox{\@hRubik}{\SquareFa}\,}
\newcommand{\textRubikFa}{\rrhFa\,}
```

16.14.26 Rotation Fap

`\rrFap` `\rrhFap` `\RubikFap` `\textRubikFap`

These commands draw forms of the Singmaster Fap slice rotation. We need to just make square with Fap in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrFap}{\@rrap{F}}
\newcommand{\SquareFap}{\@SquareLetter{\rrFap}}
\newcommand{\rrhFap}{\raisebox{-0.25mm}{\SquareFap}\,}
\newcommand{\RubikFap}{\raisebox{\@hRubik}{\SquareFap}\,}
\newcommand{\textRubikFap}{\rrhFap\,}
```

16.14.27 Rotation L

`\rrL` `\SquareL` `\rrhL` `\RubikL` `\textRubikL`

These commands all draw forms which denote the L rotation.

```
\newcommand{\rrL}{\textbf{\textsf{L}}}
%%
\newcommand{\SquareL}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhL}{\raisebox{-0.333\height}{\SquareL}\,}
%%
\newcommand{\RubikL}{%
\begin{minipage}{0.6cm}
\centering
\SquareL\\
\rrL
\end{minipage}%
}
\newcommand{\textRubikL}{\rrL\,\rrhL}
```

16.14.28 Rotation Lp

`\rrLp` `\SquareLp` `\rrhLp` `\RubikLp` `\textRubikLp`

These commands all draw forms which denote the Lp rotation.

```
\newcommand{\rrLp}{\textbf{\textsf{L}$^\prime$}}
%%
\newcommand{\SquareLp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLp}{\raisebox{-0.333\height}{\SquareLp}\,}
%%
\newcommand{\RubikLp}{%
\begin{minipage}{0.6cm}
\centering
\SquareLp\\
\rrLp
\end{minipage}%
}
\newcommand{\textRubikLp}{\rrLp\,\rrhLp}
```

16.14.29 Rotation Lw

\rrLw \SquareLw \rrhLw \RubikLw \textRubikLw

These commands all draw forms which denote the Lw rotation.

```
\newcommand{\rrLw}{\textbf{\textsf{L\footnotesize{w}}}}
%%
\newcommand{\SquareLw}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick, <-] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLw}{\raisebox{-0.333\height}{\SquareLw}\,}
%%
\newcommand{\RubikLw}{%
\begin{minipage}{0.6cm}
\centering
\SquareLw\\
\rrLw
\end{minipage}%
}
\newcommand{\textRubikLw}{\rrLw\,\rrhLw}
```

16.14.30 Rotation Lwp

\rrLwp \SquareLwp \rrhLwp \RubikLwp \textRubikLwp

These commands all draw forms which denote the Lwp rotation.

```
\newcommand{\rrLwp}{\textbf{\textsf{L\footnotesize{w}}$^\prime$}}
%%
\newcommand{\SquareLwp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick,->] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLwp}{\raisebox{-0.333\height}{\SquareLwp}\,}
%%
\newcommand{\RubikLwp}{%
\begin{minipage}{0.6cm}
\centering
\SquareLwp\\
\rrLwp
\end{minipage}%
}
\newcommand{\textRubikLwp}{\rrLwp\,\rrhLwp}
```

16.14.31 Rotation Ls

\rrLs
\SquareLs
\rrhLs
\RubikLs
\textRubikLs

These commands all draw forms which denote the Ls rotation.

```
\newcommand{\rrLs}{\@rrs{L}}
%%
\newcommand{\SquareLs}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, <-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLs}{\raisebox{-0.333\height}{\SquareLs}\,}
%%
\newcommand{\RubikLs}{%
\begin{minipage}{0.6cm}
\centering
\SquareLs\\
\rrLs
\end{minipage}%
}
\newcommand{\textRubikLs}{\rrLs\,\rrhLs}
```

16.14.32 Rotation Lsp

\rrLsp
\SquareLsp
\rrhLsp
\RubikLsp
\textRubikLsp

These commands all draw forms which denote the Lsp rotation.

```
\newcommand{\rrLsp}{\@rrsp{L}}
%%
\newcommand{\SquareLsp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, ->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLsp}{\raisebox{-0.333\height}{\SquareLsp}\,}
%%
\newcommand{\RubikLsp}{%
\begin{minipage}{0.6cm}
\centering
\SquareLsp\\
\rrLsp
\end{minipage}%
}
\newcommand{\textRubikLsp}{\rrLsp\,\rrhLsp}
```

16.14.33 Rotation La

\rrLa
\SquareLa
\rrhLa
\RubikLa
\textRubikLa

These commands all draw forms which denote the La rotation.

```
\newcommand{\rrLa}{\@rra{L}}
%%
\newcommand{\SquareLa}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, ->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLa}{\raisebox{-0.333\height}{\SquareLa}\,}
%%
\newcommand{\RubikLa}{%
\begin{minipage}{0.6cm}
\centering
\SquareLa\\
\rrLa
\end{minipage}%
}
\newcommand{\textRubikLa}{\rrLa\,\rrhLa}
```

16.14.34 Rotation Lap

\rrLap
\SquareLap
\rrhLap
\RubikLap
\textRubikLap

These commands all draw forms which denote the Lap rotation.

```
\newcommand{\rrLap}{\@rrap{L}}
%%
\newcommand{\SquareLap}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, <-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhLap}{\raisebox{-0.333\height}{\SquareLap}\,}
%%
\newcommand{\RubikLap}{%
\begin{minipage}{0.6cm}
\centering
\SquareLap\\
\rrLap
\end{minipage}%
}
\newcommand{\textRubikLap}{\rrLap\,\rrhLap}
```

16.14.35 Rotation M

\rrM \SquareM \rrhM \RubikM \textRubikM

These commands all draw forms which denote the M rotation.

```
\newcommand{\rrM}{\textbf{\textsf{M}}}
%%
\newcommand{\SquareM}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick, <-] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhM}{\raisebox{-0.333\height}{\SquareM}\,}
%%
\newcommand{\RubikM}{%
\begin{minipage}{0.6cm}
\centering
\SquareM\\
\rrM
\end{minipage}%
}
\newcommand{\textRubikM}{\rrM\,\rrhM}
```

16.14.36 Rotation Mp

\rrMp \SquareMp \rrhMp \RubikMp \textRubikMp

These commands all draw forms which denote the Mp rotation.

```
\newcommand{\rrMp}{\textbf{\textsf{M}$^\prime$}}
%%
\newcommand{\SquareMp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick,->] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhMp}{\raisebox{-0.333\height}{\SquareMp}\,}
%%
\newcommand{\RubikMp}{%
\begin{minipage}{0.6cm}
\centering
\SquareMp\\
\rrMp
\end{minipage}%
}
\newcommand{\textRubikMp}{\rrMp\,\rrhMp}
```

16.14.37 Rotation R

`\rrR` `\SquareR` `\rrhR` `\RubikR` `\textRubikR`

These commands all draw forms which denote the R rotation.

```
\newcommand{\rrR}{\textbf{\textsf{R}}}
%%
\newcommand{\SquareR}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
%% draw three lines in the square, one with an arrow
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, ->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhR}{\raisebox{-0.333\height}{\SquareR}\,}
%%
\newcommand{\RubikR}{%
\begin{minipage}{0.6cm}
\centering
\SquareR\\
\rrR
\end{minipage}%
}
\newcommand{\textRubikR}{\rrR\,\rrhR}
```

16.14.38 Rotation Rp

`\rrRp` `\SquareRp` `\rrhRp` `\RubikRp` `\textRubikRp`

These commands all draw forms which denote the Rp rotation.

```
\newcommand{\rrRp}{\textbf{\textsf{R}$^\prime$}}
%%
\newcommand{\SquareRp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, <-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRp}{\raisebox{-0.333\height}{\SquareRp}\,}
%%
\newcommand{\RubikRp}{%
\begin{minipage}{0.6cm}
\centering
\SquareRp\\
\rrRp
\end{minipage}%
}
\newcommand{\textRubikRp}{\rrRp\,\rrhRp}
```

16.14.39 Rotation Rw

\rrRw \SquareRw \rrhRw \RubikRw \textRubikRw

These commands all draw forms which denote the Rw rotation.

```
\newcommand{\rrRw}{\textbf{\textsf{R\footnotesize{w}}}}
%%
\newcommand{\SquareRw}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick, ->] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, ->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRw}{\raisebox{-0.333\height}{\SquareRw}\,}
%%
\newcommand{\RubikRw}{%
\begin{minipage}{0.6cm}
\centering
\SquareRw\\
\rrRw
\end{minipage}%
}
\newcommand{\textRubikRw}{\rrRw\,\rrhRw}
```

16.14.40 Rotation Rwp

\rrRwp \SquareRwp \rrhRwp \RubikRwp \textRubikRwp

These commands all draw forms which denote the Rwp rotation.

```
\newcommand{\rrRwp}{\textbf{\textsf{R\footnotesize{w}}$^\prime$}}
%%
\newcommand{\SquareRwp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\sd, \sb) -- (\sd, \sbh);
\draw [thick, <-] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick, <-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRwp}{\raisebox{-0.333\height}{\SquareRwp}\,}
%%
\newcommand{\RubikRwp}{%
\begin{minipage}{0.6cm}
\centering
\SquareRwp\\
\rrRwp
\end{minipage}%
}
\newcommand{\textRubikRwp}{\rrRwp\,\rrhRwp}
```

16.14.41 Rotation Rs

\rrRs
\SquareRs
\rrhRs
\RubikRs
\textRubikRs

These commands all draw forms which denote the Rs rotation.

```
\newcommand{\rrRs}{\@rrs{R}}
%%
\newcommand{\SquareRs}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick,->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRs}{\raisebox{-0.333\height}{\SquareRs}\,}
%%
\newcommand{\RubikRs}{%
\begin{minipage}{0.6cm}
\centering
\SquareRs\\
\rrRs
\end{minipage}%
}
\newcommand{\textRubikRs}{\rrRs\,\rrhRs}
```

16.14.42 Rotation Rsp

\rrRsp
\SquareRsp
\rrhRsp
\RubikRsp
\textRubikRsp

These commands all draw forms which denote the Rsp rotation.

```
\newcommand{\rrRsp}{\@rrsp{R}}
%%
\newcommand{\SquareRsp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,<-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick,<-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
```

```
\newcommand{\rrhRsp}{\raisebox{-0.333\height}{\SquareRsp}\,}
%%
\newcommand{\RubikRsp}{%
\begin{minipage}{0.6cm}
\centering
\SquareRsp\\
\rrRsp
\end{minipage}%
}
\newcommand{\textRubikRsp}{\rrRsp\,\rrhRsp}
```

16.14.43 Rotation Ra

\rrRa
\SquareRa
\rrhRa
\RubikRa
\textRubikRa

These commands all draw forms which denote the Ra rotation.

```
\newcommand{\rrRa}{\@rra{R}}
%%
\newcommand{\SquareRa}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,<-] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick,->] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRa}{\raisebox{-0.333\height}{\SquareRa}\,}
%%
\newcommand{\RubikRa}{%
\begin{minipage}{0.6cm}
\centering
\SquareRa\\
\rrRa
\end{minipage}%
}
\newcommand{\textRubikRa}{\rrRa\,\rrhRa}
```

16.14.44 Rotation Rap

\rrRap
\SquareRap
\rrhRap
\RubikRap
\textRubikRap

These commands all draw forms which denote the Rap rotation.

```
\newcommand{\rrRap}{\@rrap{R}}
%%
\newcommand{\SquareRap}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\sd, \sb) -- (\sd, \sbh);
\draw [thick] (\sdd,\sb) -- (\sdd, \sbh);
\draw [thick,<-] (\sddd, \sb) -- (\sddd, \sbh);
\end{tikzpicture}%
}
\newcommand{\rrhRap}{\raisebox{-0.333\height}{\SquareRap}\,}
```

```
%%
\newcommand{\RubikRap}{%
\begin{minipage}{0.6cm}
\centering
\SquareRap\\
\rrRap
\end{minipage}%
}
\newcommand{\textRubikRap}{\rrRap\,\rrhRap}
```

16.14.45 Rotation S

\rrS \SquareS \rrhS \RubikS \textRubikS

These commands all draw forms which denote the S rotation. Not visible from the front.

```
\newcommand{\rrS}{\@rr{S}}
\newcommand{\SquareS}{\@SquareLetter{\rrS}}
\newcommand{\rrhS}{\raisebox{-0.25mm}{\SquareS}\,}
\newcommand{\RubikS}{\raisebox{\@hRubik}{\SquareS}\,}
\newcommand{\textRubikS}{\rrhS\,}
```

16.14.46 Rotation Sp

\rrSp \SquareSp \rrhSp \RubikSp \textRubikSp

These commands all draw forms which denote the Sp rotation. Not visible from the front.

```
\newcommand{\rrSp}{\@rrp{S}}
\newcommand{\SquareSp}{\@SquareLetter{\rrSp}}
\newcommand{\rrhSp}{\raisebox{-0.25mm}{\SquareSp}\,}
\newcommand{\RubikSp}{\raisebox{\@hRubik}{\SquareSp}\,}
\newcommand{\textRubikSp}{\rrhSp\,}
```

16.14.47 Rotation Su

\rrSu \rrhSu \RubikSu \textRubikSu

These commands draw forms of the Singmaster Su slice rotation. We also need to finetune the spacing between these ‘slice’ hieroglyphs (especially Fs and Bs).

```
\newcommand{\rrSu}{\textbf{\textsf{S\footnotesize{u}}}}
\newcommand{\rrhSu}{\rrhEp}%
\newcommand{\RubikSu}{%
\begin{minipage}{0.6cm}
\centering
\SquareEp\\
\rrSu
\end{minipage}%
}
\newcommand{\textRubikSu}{\rrSu\,\rrhEp}
```

16.14.48 Rotation Sup

\rrSup \rrhSup \RubikSup \textRubikSup

These commands draw forms of the Singmaster Sup slice rotation. We also need to finetune the spacing between these 'slice' hieroglyphs (especially Fs and Bs).

```
\newcommand{\rrSup}{\textbf{\textsf{S\footnotesize{u}}$^\prime$}}
\newcommand{\rrhSup}{\rrhE}%
\newcommand{\RubikSup}{%
\begin{minipage}{0.6cm}
\centering
\SquareE\\
\rrSup
\end{minipage}%
}
\newcommand{\textRubikSup}{\rrSup\,\rrhE}
```

16.14.49 Rotation Sd

\rrSd \rrhSd \RubikSd \textRubikSd

These commands draw forms of the Singmaster Sd slice rotation.

```
\newcommand{\rrSd}{\textbf{\textsf{S\footnotesize{d}}}}
\newcommand{\rrhSd}{\rrhE}%
\newcommand{\RubikSd}{%
\begin{minipage}{0.6cm}
\centering
\SquareE\\
\rrSd
\end{minipage}%
}
\newcommand{\textRubikSd}{\rrSd\,\rrhE}
```

16.14.50 Rotation Sdp

\rrSdp \rrhSdp \RubikSdp \textRubikSdp

These commands draw forms of the Singmaster Sdp slice rotation.

```
\newcommand{\rrSdp}{\textbf{\textsf{S\footnotesize{d}}$^\prime$}}
\newcommand{\rrhSdp}{\rrhEp}%
\newcommand{\RubikSdp}{%
\begin{minipage}{0.6cm}
\centering
\SquareEp\\
\rrSdp
\end{minipage}%
}
\newcommand{\textRubikSdp}{\rrSdp\,\rrhEp}
```

16.14.51 Rotation Sl

\rrSl \rrhSl \RubikSl \textRubikSl

These commands draw forms of the Singmaster Sl slice rotation.

```
\newcommand{\rrSl}{\textbf{\textsf{S\footnotesize{l}}}}
\newcommand{\rrhSl}{\rrhM}%
\newcommand{\RubikSl}{%
\begin{minipage}{0.6cm}
\centering
\SquareM\\
\rrSl
\end{minipage}%
}
\newcommand{\textRubikSl}{\rrSl\,\rrhM}
```

16.14.52 Rotation Slp

\rrSlp \rrhSlp \RubikSlp \textRubikSlp

These commands draw forms of the Singmaster Slp slice rotation.

```
\newcommand{\rrSlp}{\textbf{\textsf{S\footnotesize{l}}$^\prime$}}
\newcommand{\rrhSlp}{\rrhMp}%
\newcommand{\RubikSlp}{%
\begin{minipage}{0.6cm}
\centering
\SquareMp\\
\rrSlp
\end{minipage}%
}
\newcommand{\textRubikSlp}{\rrSlp\,\rrhMp}
```

16.14.53 Rotation Sr

\rrSr \rrhSr \RubikSr \textRubikSr

These commands draw forms of the Singmaster Sr slice rotation.

```
\newcommand{\rrSr}{\textbf{\textsf{S\footnotesize{r}}}}
\newcommand{\rrhSr}{\rrhMp}%
\newcommand{\RubikSr}{%
\begin{minipage}{0.6cm}
\centering
\SquareMp\\
\rrSr
\end{minipage}%
}
\newcommand{\textRubikSr}{\rrSr\,\rrhMp}
```

16.14.54 Rotation Srp

\rrSrp \rrhSrp \RubikSrp \textRubikSrp

These commands draw forms of the Singmaster Srp slice rotation.

```
\newcommand{\rrSrp}{\textbf{\textsf{S\footnotesize{r}}$^\prime$}}
\newcommand{\rrhSrp}{\rrhM}%
\newcommand{\RubikSrp}{%
\begin{minipage}{0.6cm}
\centering
\SquareM\\
\rrSrp
\end{minipage}%
}
\newcommand{\textRubikSrp}{\rrSrp\,\rrhM}
```

16.14.55 Rotation Sf

\rrSf
\rrhSf
\RubikSf
\textRubikSf

These commands draw forms of the Singmaster Sf slice rotation. We need to just make square with Sf in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrSf}{\textbf{\textsf{S\footnotesize{f}}}}
\newcommand{\SquareSf}{\@SquareLetter{\rrSf}}
\newcommand{\rrhSf}{\raisebox{-0.25mm}{\SquareSf}\,}
\newcommand{\RubikSf}{\raisebox{\@hRubik}{\SquareSf}\,}
\newcommand{\textRubikSf}{\rrhSf\,}
```

16.14.56 Rotation Sfp

\rrSfp
\rrhSfp
\RubikSfp
\textRubikSfp

These commands draw forms of the Singmaster Sfp slice rotation. We need to just make square with Sfp in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrSfp}{\textbf{\textsf{S\footnotesize{f}}$^\prime$}}
\newcommand{\SquareSfp}{\@SquareLetter{\rrSfp}}
\newcommand{\rrhSfp}{\raisebox{-0.25mm}{\SquareSfp}\,}
\newcommand{\RubikSfp}{\raisebox{\@hRubik}{\SquareSfp}\,}
\newcommand{\textRubikSfp}{\rrhSfp\,}
```

16.14.57 Rotation Sb

\rrSb
\rrhSb
\RubikSb
\textRubikSb

These commands draw forms of the Singmaster Sb slice rotation. We need to just make square with Sb in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from the front.

```
\newcommand{\rrSb}{\textbf{\textsf{S\footnotesize{b}}}}
\newcommand{\SquareSb}{\@SquareLetter{\rrSb}}
\newcommand{\rrhSb}{\raisebox{-0.25mm}{\SquareSb}\,}
\newcommand{\RubikSb}{\raisebox{\@hRubik}{\SquareSb}\,}
\newcommand{\textRubikSb}{\rrhSb\,}
```

16.14.58 Rotation Sbp

\rrSbp
\rrhSbp
\RubikSbp
\textRubikSbp

These commands draw forms of the Singmaster Sbp slice rotation. We need to just make square with Sbp in square; adjust box height using a `\rule`; adjust `\fboxsep` (default=3pt); adjust `\fboxrule` (default=0.4pt); bounded by {} so no need to reset to defaults. Not visible from th front.

```
\newcommand{\rrSbp}{\textbf{\textsf{S\footnotesize{b}}$^\prime$}}
\newcommand{\SquareSbp}{\@SquareLetter{\rrSbp}}
\newcommand{\rrhSbp}{\raisebox{-0.25mm}{\SquareSbp}\,}
\newcommand{\RubikSbp}{\raisebox{\@hRubik}{\SquareSbp}\,}
\newcommand{\textRubikSbp}{\rrhSbp\,}
```

16.14.59 Rotation U

\rrU \SquareU \rrhU \RubikU \textRubikU

These commands all draw forms which denote the U rotation.

```
\newcommand{\rrU}{\textbf{\textsf{U}}}
%%
\newcommand{\SquareU}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhU}{\raisebox{-0.333\height}{\SquareU}\,}
%%
\newcommand{\RubikU}{%
\begin{minipage}{0.6cm}
\centering
\SquareU\\
\rrU
\end{minipage}%%
}
\newcommand{\textRubikU}{\rrU\,\rrhU}
```

16.14.60 Rotation Uw

\rrUw \SquareUw \rrhUw \RubikUw \textRubikUw

These commands all draw forms which denote the Uw rotation.

```
\newcommand{\rrUw}{\textbf{\textsf{U\footnotesize{w}}}}
%%
\newcommand{\SquareUw}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, <-] (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
%
\newcommand{\rrhUw}{\raisebox{-0.333\height}{\SquareUw}\,}
%%
\newcommand{\RubikUw}{%
\begin{minipage}{0.6cm}
\centering
\SquareUw\\
\rrUw
\end{minipage}%%
}
%%
\newcommand{\textRubikUw}{\rrUw\,\rrhUw}
```

16.14.61 Rotation Up

\rrUp \SquareUp \rrhUp \RubikUp \textRubikUp

These commands all draw forms which denote the Up rotation.

```
\newcommand{\rrUp}{\textbf{\textsf{U}$^\prime$}}
%%
\newcommand{\SquareUp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUp}{\raisebox{-0.333\height}{\SquareUp}\,}
%%
\newcommand{\RubikUp}{%
\begin{minipage}{0.6cm}
\centering
\SquareUp\\
\rrUp
\end{minipage}%%
}
\newcommand{\textRubikUp}{\rrUp\,\rrhUp}
```

16.14.62 Rotation Uwp

\rrUwp \SquareUwp \rrhUwp \RubikUwp \textRubikUwp

These commands all draw forms which denote the Uwp rotation.

```
\newcommand{\rrUwp}{\textbf{\textsf{U\footnotesize{w}}$^\prime$}}
%%
\newcommand{\SquareUwp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick, ->] (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUwp}{\raisebox{-0.333\height}{\SquareUwp}\,}
%%
\newcommand{\RubikUwp}{%
\begin{minipage}{0.6cm}
\centering
\SquareUwp\\
\rrUwp
\end{minipage}%%
}
\newcommand{\textRubikUwp}{\rrUwp\,\rrhUwp}
```

16.14.63 Rotation Us

\rrUs \SquareUs \rrhUs \RubikUs \textRubikUs

These commands all draw forms which denote the Us rotation.

```
\newcommand{\rrUs}{\@rrs{U}}
%%
\newcommand{\SquareUs}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, <-]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUs}{\raisebox{-0.333\height}{\SquareUs}\,}
%%
\newcommand{\RubikUs}{%
\begin{minipage}{0.6cm}
\centering
\SquareUs\\
\rrUs
\end{minipage}%
}
\newcommand{\textRubikUs}{\rrUs\,\rrhUs}
```

16.14.64 Rotation Usp

\rrUsp \SquareUsp \rrhUs \RubikUs \textRubikUsp

These commands all draw forms which denote the Usp rotation.

```
\newcommand{\rrUsp}{\@rrsp{U}}
%%
\newcommand{\SquareUsp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUsp}{\raisebox{-0.333\height}{\SquareUsp}\,}
%%
\newcommand{\RubikUsp}{%
\begin{minipage}{0.6cm}
\centering
\SquareUsp\\
\rrUsp
\end{minipage}%
}
\newcommand{\textRubikUsp}{\rrUsp\,\rrhUsp}
```

16.14.65 Rotation Ua

\rrUa
\SquareUa
\rrhUa
\RubikUa
\textRubikUa

These commands all draw forms which denote the Ua rotation.

```
\newcommand{\rrUa}{\@rra{U}}
%%
\newcommand{\SquareUa}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, ->]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUa}{\raisebox{-0.333\height}{\SquareUa}\,}
%%
\newcommand{\RubikUa}{%
\begin{minipage}{0.6cm}
\centering
\SquareUa\\
\rrUa
\end{minipage}%
}
\newcommand{\textRubikUa}{\rrUa\,\rrhUa}
```

16.14.66 Rotation Uap

\rrUap
\SquareUap
\rrhUap
\RubikUap
\textRubikUap

These commands all draw forms which denote the Uap rotation.

```
\newcommand{\rrUap}{\@rrap{U}}
%%
\newcommand{\SquareUap}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\sb,\sddd) -- (\sbh, \sddd);
\draw [thick]     (\sb,\sdd) --  (\sbh, \sdd);
\draw [thick, <-]     (\sb,\sd) --   (\sbh, \sd);
\end{tikzpicture}%
}
\newcommand{\rrhUap}{\raisebox{-0.333\height}{\SquareUap}\,}
%%
\newcommand{\RubikUap}{%
\begin{minipage}{0.6cm}
\centering
\SquareUap\\
\rrUap
\end{minipage}%
}
\newcommand{\textRubikUap}{\rrUap\,\rrhUap}
```

16.15 Rotation commands — axis rotations

16.15.1 Rotations x and xp

\rrx \rrhx \Rubikx

These commands all draw forms which denote the x rotation.

```
\newcommand{\rrx}{\textbf{\textsf{x}}}
\newcommand{\Rubikx}{\@xyzRubik{x}}
\newcommand{\rrhx}{\@xyzh{x}}
```

\rrxp \rrhxp \Rubikxp

These commands all draw forms which denote the xp rotation.

```
\newcommand{\rrxp}{\textbf{\textsf{x}$^\prime$}}
\newcommand{\Rubikxp}{\@xyzRubikp{x}}
\newcommand{\rrhxp}{\@xyzhp{x}}
```

16.15.2 Rotations y and yp

\rry \rrhy \Rubiky

These commands all draw forms which denote the y rotation.

```
\newcommand{\rry}{\textbf{\textsf{y}}}
\newcommand{\Rubiky}{\@xyzRubik{y}}
\newcommand{\rrhy}{\@xyzh{y}}
```

\rryp \rrhyp \Rubikyp

These commands all draw forms which denote the yp rotation.

```
\newcommand{\rryp}{\textbf{\textsf{y}$^\prime$}}
\newcommand{\Rubikyp}{\@xyzRubikp{y}}
\newcommand{\rrhyp}{\@xyzhp{y}}
```

16.15.3 Rotations z and zp

\rrz \rrhz \Rubikz

These commands all draw forms which denote the z rotation.

```
\newcommand{\rrz}{\textbf{\textsf{z}}}
\newcommand{\Rubikz}{\@xyzRubik{z}}
\newcommand{\rrhz}{\@xyzh{z}}
```

\rrzp \rrhzp \Rubikzp

These commands all draw forms which denote the zp rotation.

```
\newcommand{\rrzp}{\textbf{\textsf{z}$^\prime$}}
\newcommand{\Rubikzp}{\@xyzRubikp{z}}
\newcommand{\rrhzp}{\@xyzhp{z}}
```

16.15.4 Rotations u and d

\rru \rrhu \Rubiku

These commands all draw forms which denote the u rotation.

```
\newcommand{\rru}{\textbf{\textsf{u}}}
\newcommand{\Rubiku}{\@xyzRubik{u}}
\newcommand{\rrhu}{\@xyzh{u}}
```

\rrd \rrhd \Rubikd

These commands all draw forms which denote the d rotation.

```
\newcommand{\rrd}{\textbf{\textsf{d}}}
\newcommand{\Rubikd}{\@xyzRubik{d}}
\newcommand{\rrhd}{\@xyzh{d}}
```

16.15.5 Rotations l and r

\rrl \rrhl \Rubikl

These commands all draw forms which denote the l rotation.

```
\newcommand{\rrl}{\textbf{\textsf{l}}}
\newcommand{\Rubikl}{\@xyzRubik{l}}
\newcommand{\rrhl}{\@xyzh{l}}
```

\rrr \rrhr \Rubikr

These commands all draw forms which denote the r rotation.

```
\newcommand{\rrr}{\textbf{\textsf{r}}}
\newcommand{\Rubikr}{\@xyzRubik{r}}
\newcommand{\rrhr}{\@xyzh{r}}
```

16.15.6 Rotations f and b

\rrf \rrhf \Rubikf

These commands all draw forms which denote the f rotation.

```
\newcommand{\rrf}{\textbf{\textsf{f}}}
\newcommand{\Rubikf}{\@xyzRubik{f}}
\newcommand{\rrhf}{\@xyzh{f}}
```

\rrb \rrhb \Rubikb

These commands all draw forms which denote the b rotation.

```
\newcommand{\rrb}{\textbf{\textsf{b}}}
\newcommand{\Rubikb}{\@xyzRubik{b}}
\newcommand{\rrhb}{\@xyzh{b}}
```

——————— End of this package ———————

```
⟨/rubikcube⟩
```