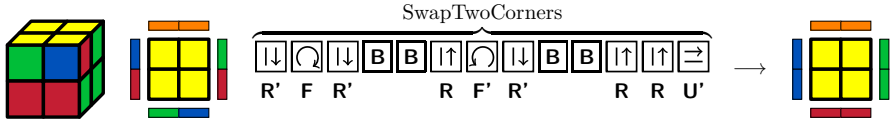
The RUBIKTWOCUBE package

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

This file describes version 5.0 (2018/02/25)
www.ctan.org/pkg/rubik

Abstract

The RUBIKTWOCUBE package provides LaTeX commands and macros for typesetting TwoCube (2x2x2) notation, configurations, and rotation sequences using the TikZ graphic language. It is part of the Rubik 'bundle'.



Contents

1 Introduction

The RUBIKTWOCUBE package (part of the RUBIK 'bundle') provides a collection of LaTeX commands and macros for typesetting Rubik 2x2x2 cube configurations using the PGF/TikZ graphic languages. This package is a minor extension of the RUBIKCUBE package, and users are therefore assumed to be familiar with both the RUBIKCUBE and RUBIKROTATION packages. For examples of use see the file `rubikexamples.pdf`.

1.1 Requirements

The RUBIKTWOCUBE package requires the TikZ package (since it makes use of the TikZ picture environment), and also the RUBIKCUBE package.

For full functionality the complementary packages RUBIKROTATION and RUBIKPATTERNS also need to be loaded. Note that the RUBIKROTATION package requires Perl to be installed. See the 'Installation' section in the RUBIKCUBE package documentation (`rubikcube.pdf`) for more details.

1.2 Copyright

2 Installation

The Rubik bundle consists of the four packages RUBIKCUBE, RUBIKROTATION, RUBIKPATTERNS and RUBIKTWOCUBE.

Here we describe only the installation of the RUBIKTWOCUBE package, which consists of the following files:

```
rubiktwocube.ins
rubiktwocube.dtx
rubiktwocube.pdf         --documentation of the rubiktwocube package
rubiktwo-doc-figA.pdf
```

Before installing the RUBIKTWOCUBE package make sure the following packages are already installed (TikZ graphics system and the RUBIKCUBE package).

2.1 `rubiktwocube.sty`

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

```
pdflatex  rubiktwocube.ins
```

2.2 rubiktwocube.pdf

The documentation file (rubiktwocube.pdf) is then generated using the following steps [1]:

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

2.3 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/rubik/
*.pdf → /usr/local/texlive/texmf-local/doc/rubik/

Finally, (depending on your system) update the TEX file database. For example, on a Linux system one uses the texhash command.

2.4 Usage

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

```
\usepackage{tikz}
\usepackage{rubikcube,rubikrotation,rubikpatterns,rubiktwocube}
```

2.5 rubikexamples.pdf

The Rubik bundle includes a 'RubikExamples' file (rubikexamples.pdf) as well as associated .sh (Linux) and .bat (Microsoft) batch files which can be used to facilitate processing the source file (rubikexamples.tex). See the 'Installation' section in the RUBIKCUBE package documentation (rubikcube.pdf) for details regarding processing the examples source file.

[1] Several pdflatex runs are required, since the documentation includes an index as well as hyperef links (the package hypdoc is used). Prior to the first run it is a good idea to delete any relevant .toc, .aux, .out files.

3 Command conventions

The examples given in the file `rubikexamples.pdf` present a good overview of the commands and how to use them.

3.1 The keywords Two and Rubik in commands

In order to try and keep commands intuitive [2] we adopt the convention that the word 'Two' in a command reflects the fact that the command relates to a 2x2x2 cube (a 'Two' cube). Similarly, commands which relate to a 3x3x3 cube (a 'Rubik' cube) —see the RUBIKCUBE package— use instead the word 'Rubik'.

It is assumed that users are familiar with the RUBIKCUBE and RUBIKROTATION packages, since virtually all RUBIKTWOCUBE commands mirror the Rubik (3x3x3) cube commands, such that the word 'Rubik' is replaced by the word 'Two' (exceptions are highlighted). For example, the commands for drawing a 2x2x2 cube and a 3x3x3 cube from a RU viewpoint are respectively `\DrawTwoCubeRU` and `\DrawRubikCubeRU`. The examples given in the file `rubikexamples.pdf` present a good overview of the commands and how to use them.

For more detailed information see (a) the 'code' section (Section 9), or (b) see the equivalent 3x3x3 commands in the RUBIKCUBE package.

4 Colour commands

The following list shows the RUBIKTWOCUBE colour commands paired (for convenience) with the equivalent 3x3x3 version from the RUBIKCUBE package. The `..` indicates that mandatory arguments are required.

RubikCube 3x3x3	TwoCube 2x2x2
`\RubikCubeSolved`	`\TwoCubeSolved`
`\RubikCubeSolvedWY`	`\TwoCubeSolvedWY`
`\RubikCubeSolvedWB`	`\TwoCubeSolvedWB`
`\RubikCubeGrey`	`\TwoCubeGrey`
`\RubikCubeGray`	`\TwoCubeGray`
`\RubikCubeGreyWY`	
`\RubikCubeGrayWY`	
`\RubikCubeGreyWB`	
`\RubikCubeGrayWB`	
`\RubikCubeGreyAll`	`\TwoCubeGreyAll`
`\RubikCubeGrayAll`	`\TwoCubeGrayAll`
`\RubikSolvedConfig..`	`\TwoSolvedConfig..`
`\RubikFaceUp..`	`\TwoFaceUp..`
`\RubikFaceDown..`	`\TwoFaceDown..`

[2] This is a tricky problem given the large number of commands, so any feedback or ideas on how to avoid ambiguity, including pruning or revising 'bad' commands, is always welcome.

```
\RubikFaceLeft..        \TwoFaceLeft..
\RubikFaceRight..       \TwoFaceRight..
\RubikFaceFront..       \TwoFaceFront..
\RubikFaceBack..        \TwoFaceBack..
\RubikFaceUpAll..       \TwoFaceUpAll..
\RubikFaceDownAll..     \TwoFaceDownAll..
\RubikFaceLeftAll..     \TwoFaceLeftAll..
\RubikFaceRightAl..     \TwoFaceRightAll..
\RubikFaceFrontAl..     \TwoFaceFrontAll..
\RubikFaceBackAll..     \TwoFaceBackAll..

\RubikSidebarWidth..    \TwoSidebarWidth..
\RubikSidebarLength..   \TwoSidebarLength..
\RubikSidebarSep..      \TwoSidebarSep..

\RubikSliceTopL..       \TwoSliceTopL..
\RubikSliceTopR..       \TwoSliceTopR..
\RubikSliceBottomL..    \TwoSliceBottomL..
\RubikSliceBottomR..    \TwoSliceBottomR..
```

5 Draw commands

The following list shows the RUBIKTWOCUBE Draw commands paired (for convenience) with the equivalent 3x3x3 version from the RUBIKCUBE package. Commands in round brackets show short-hand equivalents.

```
 RubikCube               TwoCube
 3x3x3                   2x2x2

\DrawRubikCubeRU         \DrawTwoCubeRU
\DrawRubikCubeRD         \DrawTwoCubeRD
\DrawRubikCubeLU         \DrawTwoCubeLU
\DrawRubikCubeLD         \DrawTwoCubeLD
\DrawRubikCubeF          \DrawTwoCubeF
\DrawRubikCubeSF         \DrawTwoCubeSF

\DrawRubikCubeSidebarFL..  \DrawTwoCubeSidebarFL..
\DrawRubikCubeSidebarFR..  \DrawTwoCubeSidebarFR..
\DrawRubikCubeSidebarFU..  \DrawTwoCubeSidebarFU..
\DrawRubikCubeSidebarFD..  \DrawTwoCubeSidebarFD..
\DrawRubikCubeSidebarBL..  \DrawTwoCubeSidebarBL..
\DrawRubikCubeSidebarBR..  \DrawTwoCubeSidebarBR..
\DrawRubikCubeSidebarBU..  \DrawTwoCubeSidebarBU..
\DrawRubikCubeSidebarBD..  \DrawTwoCubeSidebarBD..

\DrawRubikFaceUp         \DrawTwoFaceUp         (= \DrawTwoFaceU )
\DrawRubikFaceDown       \DrawTwoFaceDown       (= \DrawTwoFaceD )
\DrawRubikFaceLeft       \DrawTwoFaceLeft       (= \DrawTwoFaceL )
\DrawRubikFaceRight      \DrawTwoFaceRight      (= \DrawTwoFaceR )
```

```
\DrawRubikFaceFront      \DrawTwoFaceFront      (= \DrawTwoFaceF )
\DrawRubikFaceBack       \DrawTwoFaceBack       (= \DrawTwoFaceB )

\DrawRubikFaceUpSide     \DrawTwoFaceUpSide     (= \DrawTwoFaceUS )
\DrawRubikFaceDownSide   \DrawTwoFaceDownSide   (= \DrawTwoFaceDS )
\DrawRubikFaceLeftSide   \DrawTwoFaceLeftSide   (= \DrawTwoFaceLS )
\DrawRubikFaceRightSide  \DrawTwoFaceRightSide  (= \DrawTwoFaceRS )
\DrawRubikFaceFrontSide  \DrawTwoFaceFrontSide  (= \DrawTwoFaceFS )
\DrawRubikFaceBackSide   \DrawTwoFaceBackSide   (= \DrawTwoFaceBS )

\DrawRubikFlatUp..       \DrawTwoFlatUp..
\DrawRubikFlatDown..     \DrawTwoFlatDown..
\DrawRubikFlatLeft..     \DrawTwoFlatLeft..
\DrawRubikFlatRight..    \DrawTwoFlatRight..
\DrawRubikFlatFront..    \DrawTwoFlatFront..
\DrawRubikFlatBack..     \DrawTwoFlatBack..
```

6 Rotation commands

```
RubikCube           TwoCube
3x3x3               2x2x2

\RubikRotation..    \TwoRotation..
\SaveRubikState..   \SaveTwoState..
\ShowErrors         \ShowErrors
\CheckState         \CheckState
```

6.1 List of rotation commands

All the commands presented here also have a `\Two{}` equivalent form which typesets both the hieroglyph and its lettercode in a vertical format. These have been omitted here owing to the difficulty of including this form easily in the following table.

2x2x2 CHANGES: Note that all these command names mirror their 3x3x3 equivalents in the RUBIKCUBE package; the changes in the command prefixes are as follows:

`\tr` ← `\rr`
`\trh` ← `\rrh`
`\Two` ← `\Rubik`
`\textTwo` ← `\textRubik`

6.1.1 Face rotations

U \tr{U}	 \trh{U}	**U** \textTwo{U}
U' \tr{Up}	 \trh{Up}	**U'** \textTwo{Up}
D \tr{D}	 \trh{D}	**D** \textTwo{D}
D' \tr{Dp}	 \trh{Dp}	**D'** \textTwo{Dp}
L \tr{L}	 \trh{L}	**L** \textTwo{L}
L' \tr{Lp}	 \trh{Lp}	**L'** \textTwo{Lp}
R \tr{R}	 \trh{R}	**R** \textTwo{R}
R' \tr{Rp}	 \trh{Rp}	**R'** \textTwo{Rp}
F \tr{F}	 \trh{F}	**F** \textTwo{F}
F' \tr{Fp}	 \trh{Fp}	**F'** \textTwo{Fp}
B \tr{B}	 \trh{B}	\textTwo{B}
B' \tr{Bp}	 \trh{Bp}	\textTwo{Bp}

6.1.2 Axis rotations

x \tr{x}	[**x**] \trh{x}	[**x**] \Two{x}
x' \tr{xp}	[**x'**] \trh{xp}	[**x'**] \Two{xp}
y \tr{y}	[**y**] \trh{y}	[**y**] \Two{y}
y' \tr{yp}	[**y'**] \trh{yp}	[**y'**] \Two{yp}
z \tr{z}	[**z**] \trh{z}	[**z**] \Two{z}
z' \tr{zp}	[**z'**] \trh{zp}	[**z'**] \Two{zp}
u \tr{u}	[**u**] \trh{u}	[**u**] \Two{u}
u' \tr{up}	[**u'**] \trh{up}	[**u'**] \Two{up}
d \tr{d}	[**d**] \trh{d}	[**d**] \Two{d}
d' \tr{dp}	[**d'**] \trh{dp}	[**d'**] \Two{dp}

l \tr{l}	[**l**] \trh{l}	[**l**] \Two{l}
l' \tr{lp}	[**l'**] \trh{lp}	[**l'**] \Two{lp}
r \tr{r}	[**r**] \trh{r}	[**r**] \Two{r}
r' \tr{rp}	[**r'**] \trh{rp}	[**r'**] \Two{rp}
f \tr{f}	[**f**] \trh{f}	[**f**] \Two{f}
f' \tr{fp}	[**f'**] \trh{fp}	[**f'**] \Two{fp}
b \tr{b}	[**b**] \trh{b}	[**b**] \Two{b}
b' \tr{bp}	[**b'**] \trh{bp}	[**b'**] \Two{bp}
Uc \tr{Uc}	[**Uc**] \trh{Uc}	[**Uc**] \Two{Uc}
Uc' \tr{Ucp}	[**Uc'**] \trh{Ucp}	[**Uc'**] \Two{Ucp}
Dc \tr{Dc}	[**Dc**] \trh{Dc}	[**Dc**] \Two{Dc}
Dc' \tr{Dcp}	[**Dc'**] \trh{Dcp}	[**Dc'**] \Two{Dcp}
Lc \tr{Lc}	[**Lc**] \trh{Lc}	[**Lc**] \Two{Lc}
Lc' \tr{Lcp}	[**Lc'**] \trh{Lcp}	[**Lc'**] \Two{Lcp}
Rc \tr{Rc}	[**Rc**] \trh{Rc}	[**Rc**] \Two{Rc}
Rc' \tr{Rcp}	[**Rc'**] \trh{Rcp}	[**Rc'**] \Two{Rcp}
Fc \tr{Fc}	[**Fc**] \trh{Fc}	[**Fc**] \Two{Fc}
Fc' \tr{Fcp}	[**Fc'**] \trh{Fcp}	[**Fc'**] \Two{Fcp}
Bc \tr{Bc}	[**Bc**] \trh{Bc}	[**Bc**] \Two{Bc}
Bc' \tr{Bcp}	[**Bc'**] \trh{Bcp}	[**Bc'**] \Two{Bcp}
CR \tr{CR}	[**CR**] \trh{CR}	[**CR**] \Two{CR}
CR' \tr{CRp}	[**CR'**] \trh{CRp}	[**CR'**] \Two{CRp}
CL \tr{CL}	[**CL**] \trh{CL}	[**CL**] \Two{CL}
CL' \tr{CLp}	[**CL'**] \trh{CLp}	[**CL'**] \Two{CLp}
CU \tr{CU}	[**CU**] \trh{CU}	[**CU**] \Two{CU}

CU' \tr{CUp}	[**CU'**] \trh{CUp}	[**CU'**] \Two{CUp}
CD \tr{CD}	[**CD**] \trh{CD}	[**CD**] \Two{CD}
CD' \tr{CDp}	[**CD'**] \trh{CDp}	[**CD'**] \Two{CDp}
CF \tr{CF}	[**CF**] \trh{CF}	[**CF**] \Two{CF}
CF' \tr{CFp}	[**CF'**] \trh{CFp}	[**CF'**] \Two{CFp}
CB \tr{CB}	[**CB**] \trh{CB}	[**CB**] \Two{CB}
CB' \tr{CBp}	[**CB'**] \trh{CBp}	[**CB'**] \Two{CBp}

7 References

See the RUBIKCUBE package documentation for a full list of references.

8 Change history

- Version 5.0 (February 2018)
  —First release.

9 The code

All the 2x2x2 code here is essentially a cut-down version of the 3x3x3 code (RUBIKCUBE package); i.e., we have mostly just removed the 3x3x3 code relating to middle columns and rows, exchanged the word 'Rubik' for the word 'Two' in command names, and refashioned some of the commands involved in writing the temporary file `rubikstate.dat`. We assume that users are familiar with the RUBIKCUBE and RUBIKROTATION package documentation.

In order to avoid much repetition, we describe here only the essential details for understanding the relatively minor changes made in order to transform the earlier 3x3x3 RUBIKCUBE package code into working 2x2x2 code. In the following, the various instances of the heading 'CHANGES:' imply that more extensive details will be found with the equivalent 'Rubik' commands in the RUBIKCUBE or RUBIKROTATION package documentation.

Relatively few 2x2x2 square hieroglyphs are required; some needed reformulating from their 3x3x3 cousins, ie those associated with `L, Lp, R, Rp, U, Up, D, Dp`. The axis rotations and the rotations `F, Fp, B, Bp` simply required renaming; for example, as a 'TwoRotationHieroglyph' (`\trh..`) instead of the 3x3x3 'RubikRotationHieroglyph (`\rrh..`).

9.1 Package heading

The 'RTC' in the following refers to the package name RubikTwoCube.

```
⟨*rubiktwocube⟩
\def\RTCfileversion{5.0}%
\def\RTCfiledate{2018/02/25}% February 25, 2018
\NeedsTeXFormat{LaTeX2e}
\ProvidesPackage{rubiktwocube}[\RTCfiledate\space (v\RTCfileversion)]
```

The package requires TikZ (we use the pgfmathsetmacro command) —so we load it if not already loaded.

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

The package requires `rubikcube.sty`. However `rubikcube.sty` is not automatically loaded (for the moment at least) since this makes it difficult to errorcheck new versions, so we just write a message.

```
\@ifpackageloaded{rubikcube}{}{%
   \typeout{---rubiktwocube requires the rubikcube package.}%
   }%
\@ifpackageloaded{rubikrotation}{}{%
   \typeout{---rubiktwocube requires the rubikrotation package.}%
   }%
```

`\rubiktwocube` First we create a suitable logo

```
\newcommand{\rubiktwocube}{\textsc{rubiktwocube}}%
```

9.2 Saving the Two-cube state

Note that this package writes this state data to the same 'output' file (`rubikstate.dat`) as used by the 3x3x3 RUBIKROTATION package, since there is no need to change this (since the TwoCube corners will be processed in exactly the same way as for 3x3x3 cube corners).

`\@printTWOstate` This internal command writes the TwoCube state data to the 'output' file `rubikstate.dat`, and is used by the `\TwoRotation` command (see also RUBIKROTATION package documentation Sections on *save rubikstate* and *general overview* for further details). The file `rubikstate.dat` is read by the Perl script, and represents the state on which the `\TwoRotation` command acts.

CHANGES: Since this is a TwoCube all the non-corner facelets (ie those in middle rows & columns) are filled with X (grey). We have also introduced a new line in the output file (`rubikstate.dat`) namely `cubesize,two` which is used to inform the Perl program that we are dealing with a TwoCube.

```
\newcommand{\@printTWOstate}{%
   \@print{cubesize,two}%
   \@print{\space \space up,\Ult,\Umt,\Urt,\Ulm,\Umm,\Urm,\Ulb,\Umb,\Urb}%
   \@print{down,\Dlt,\Dmt,\Drt,\Dlm,\Dmm,\Drm,\Dlb,\Dmb,\Drb}%
   \@print{left,\Llt,\Lmt,\Lrt,\Llm,\Lmm,\Lrm,\Llb,\Lmb,\Lrb}%
```

```
    \@print{right,\Rlt,\Rmt,\Rrt,\Rlm,\Rmm,\Rrm,\Rlb,\Rmb,\Rrb}%
    \@print{front,\Flt,\Fmt,\Frt,\Flm,\Fmm,\Frm,\Flb,\Fmb,\Frb}%
    \@print{back,\Blt,\Bmt,\Brt,\Blm,\Bmm,\Brm,\Blb,\Bmb,\Brb}%
}
```

9.3 SaveTwoState command

\SaveTwoState

We create a TwoCube version of the existing \SaveRubikState command (RUBIKROTATION package), simply for symmetry and convenience. This command is identical to the 'Rubik' version, and will require the RUBIKROTATION package to be loaded already (as does the following \TwoRotation command.

```
\newcommand{\SaveTwoState}{\SaveRubikState}
```

9.4 TwoRotation command

Note that this command writes the data to the same file (rubikstate.dat) as that output by the equivalent 3x3x3 \RubikRotation command, since there is no need to change this.

Note that although the system works perfectly well even if we just continue to use the 3x3x3 \RubikRotation command, it was felt appropriate to implement a special TwoCube version of this command, since this allows the Perl script to be aware (via the cubesize,two line written to the rubikstate.dat file) which sort of cube it is dealing with, and hence allows the option for the program to adjust its action accordingly (for example, with regard to the randomisation procedure which is different for different cubes).

\TwoRotation

The \TwoRotation[⟨*integer*⟩]{⟨*comma separated sequence*⟩} command (a) writes the current TwoCube state to the file rubikstate.dat, (b) writes the rotation sequence (either once or multiple times depending on the value of the optional integer argument), and then (c) CALLs the Perl script rubikrotation.pl. It also writes comments to the data file and also to the log file.

The way we allow the user to (optionally) process the main argument multiple times is simply by writing the associated output command multiple times to the output data-file. Consequently, we require the \TwoRotation command to allow a square-bracket optional argument (a non-negative integer) to specify the number of such repeats.

2x2x2 CHANGES: (1) We have replaced 'Rubik' by 'Two' in the command-name (2) we use the command \@printTWOstate (see above) to write the current state data, (3) the RTC in the fileversion and filedate names denotes 'RubikTwoCube'.

```
\newcommand{\TwoRotation}[2][1]{%
    \typeout{---TeX process}%
    \typeout{---script = TwoRotation cmd (rubiktwocube.sty)%
                          v\RTCfileversion\space (\RTCfiledate)}%
    \typeout{---NEW rotation command}%
    \typeout{---command = TwoRotation[#1]{#2}}%
    \typeout{---writing current TWOcube state to file rubikstate.dat}%
```

```
  \@openstatefile% open data file
  \@print{\@comment filename: rubikstate.dat}%
  \@print{\@comment written by TwoRotation cmd (rubiktwocube.sty)%
                              v\RTCfileversion\space (\RTCfiledate)}%
  \@printTWOstate%
  %% countingloop code from Feuersaenger (2015)
  \newcount\ourRRcounter%
  \@countingloop{\ourRRcounter} in 1:{#1}{%
        \immediate\write\outfile{rotation,#2}}%
  \@closestatefile% close data file
  \typeout{---CALLing Perl script (rubikrotation.pl)}%
  \immediate\write18{\rubikperlcmd}%
  \typeout{---inputting NEW datafile (data written by Perl script)}%
  \input{rubikstateNEW.dat}%
  \typeout{----------------------------------------}%
}
```

As usual we require the `--shell-escape` command-line option to be used. This is provided by the shellesc package, and is equivalent to `\immediate\write18`. In the future we may need to replace the `\immediate\write18` with `\ShellEscape` —see the shellesc package documentation.

9.5 TwoFaceX macros

Allocate the four facelet colours to each face (only four facelets now).

```
\newcommand{\TwoFaceUp}[4]{%
   \def\Ult{#1}\def\Urt{#2}\def\Ulb{#3}\def\Urb{#4}}
\newcommand{\TwoFaceFront}[4]{%
   \def\Flt{#1}\def\Frt{#2}\def\Flb{#3}\def\Frb{#4}}
\newcommand{\TwoFaceRight}[4]{%
   \def\Rlt{#1}\def\Rrt{#2}\def\Rlb{#3}\def\Rrb{#4}}
\newcommand{\TwoFaceDown}[4]{%
   \def\Dlt{#1}\def\Drt{#2}\def\Dlb{#3}\def\Drb{#4}}
\newcommand{\TwoFaceLeft}[4]{%
   \def\Llt{#1}\def\Lrt{#2}\def\Llb{#3}\def\Lrb{#4}}
\newcommand{\TwoFaceBack}[4]{%
   \def\Blt{#1}\def\Brt{#2}\def\Blb{#3}\def\Brb{#4}}
\newcommand{\TwoFaceUpAll}[1]{%
   \def\Ult{#1}\def\Urt{#1}\def\Ulb{#1}\def\Urb{#1}}
\newcommand{\TwoFaceFrontAll}[1]{%
   \def\Flt{#1}\def\Frt{#1}\def\Flb{#1}\def\Frb{#1}}
\newcommand{\TwoFaceRightAll}[1]{%
   \def\Rlt{#1}\def\Rrt{#1}\def\Rlb{#1}\def\Rrb{#1}}
\newcommand{\TwoFaceLeftAll}[1]{%
   \def\Llt{#1}\def\Lrt{#1}\def\Llb{#1}\def\Lrb{#1}}
\newcommand{\TwoFaceDownAll}[1]{%
   \def\Dlt{#1}\def\Drt{#1}\def\Dlb{#1}\def\Drb{#1}}
\newcommand{\TwoFaceBackAll}[1]{%
   \def\Blt{#1}\def\Brt{#1}\def\Blb{#1}\def\Brb{#1}}
```

set the default colour = grey = X

```
\TwoFaceUpAll{X}%
\TwoFaceDownAll{X}%
\TwoFaceLeftAll{X}%
\TwoFaceRightAll{X}%
\TwoFaceFrontAll{X}%
\TwoFaceBackAll{X}%
```

```
\newcommand{\TwoSolvedConfig}[6]{%
  \TwoFaceRightAll{#1}%
  \TwoFaceLeftAll{#2}%
  \TwoFaceUpAll{#3}%
  \TwoFaceDownAll{#4}%
  \TwoFaceFrontAll{#5}%
  \TwoFaceBackAll{#6}%
}
```

9.6 Grey cube

\TwoCubeGrey
\TwoCubeGreyAll

This command sets up an all-grey Twocube. We accommodate both spellings 'grey' and 'gray' (as used by TikZ). We include \TwoCubeGreyAll (exactly the same) to complement the 3x3x3 version just for convenience.

Note that we include the \RubikCubeGreyAll command immediately before the \TwoSolvedConfig in order to first initialise all facelets to grey (X), (since the \.Config.. command only sets the corner cubies)

```
\newcommand{\TwoCubeGrey}{\RubikCubeGreyAll\TwoSolvedConfig{X}{X}{X}{X}{X}{X}}%
\newcommand{\TwoCubeGreyAll}{\TwoCubeGrey}
\newcommand{\TwoCubeGray}{\TwoCubeGrey}
\newcommand{\TwoCubeGrayAll}{\TwoCubeGrey}
```

Note that we include the \RubikCubeGreyAll command immediately before the \TwoSolvedConfig in order to first initialise all facelets to grey (X), (since the \.Config.. command only sets the corner cubies)

```
\newcommand{\TwoCubeSolvedWY}{\RubikCubeGreyAll\TwoSolvedConfig{G}{B}{W}{Y}{O}{R}}%
\newcommand{\TwoCubeSolved}{\TwoCubeSolvedWY}%
\newcommand{\TwoCubeSolvedWB}{\RubikCubeGreyAll\TwoSolvedConfig{R}{O}{W}{B}{G}{Y}}%
```

9.7 Slice macros

Only top and bottom horizontal slices, as viewed from TopR, TopL,BottomR,BottomL.

```
\newcommand{\TwoSliceTopR}[4]{%
   \def\Flt{#1}\def\Frt{#2}\def\Rlt{#3}\def\Rrt{#4}}
\newcommand{\TwoSliceTopL}[4]{%
   \def\Llt{#1}\def\Lrt{#2}\def\Flt{#3}\def\Frt{#4}}
\newcommand{\TwoSliceBottomR}[4]{%
   \def\Flb{#1}\def\Frb{#2}\def\Rlb{#3}\def\Rrb{#4}}
\newcommand{\TwoSliceBottomL}[4]{%
   \def\Llb{#1}\def\Lrb{#2}\def\Flb{#3}\def\Frb{#4}}
```

```
%%--------------------
\newcommand{\DrawTwoCubeFrontFace}{%
\draw[line join=round,line cap=round,ultra thick,fill=\Flt]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Frt]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- cycle;
%%
\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=\Frb]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
}
```

9.8 DrawTwoCube.. macros

```
\newcommand{\DrawTwoCubeRU}{%
%%-----------Front face----------
\DrawTwoCubeFrontFace %% frontface
%%-----------Up face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Ult]%
(0.33,2.33) -- (0.66,2.66) -- (1.66,2.66) -- (1.33,2.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urt]%
(1.33,2.33) -- (1.66,2.66) -- (2.66,2.66) -- (2.33,2.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulb]%
(0,2) -- (0.33,2.33) -- (1.33,2.33) -- (1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urb]%
(1,2) -- (1.33,2.33) -- (2.33,2.33) -- (2,2)  -- cycle;
%%-----------Right face----------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(2,1) -- (2, 2) -- (2.33,2.33) -- (2.33,1.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(2.33,1.33) -- (2.33, 2.33) -- (2.66,2.66) -- (2.66,1.66)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(2,0) -- (2, 1) -- (2.33,1.33) -- (2.33,0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(2.33,0.33) -- (2.33, 1.33) -- (2.66,1.66) -- (2.66,0.66)  -- cycle;
}
%%
\newcommand{\DrawTwoCube}{\DrawTwoCubeRU}
%%
\newcommand{\DrawTwoCubeRD}{%
\DrawTwoCubeFrontFace %% frontface
%%----------Right face--------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(2,1) -- (2, 2) -- (2.33,1.66) -- (2.33,0.66)  -- cycle;
```

```
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(2.33,0.66) -- (2.33, 1.66) -- (2.66,1.33) -- (2.66,0.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(2,0) -- (2, 1) -- (2.33,0.66) -- (2.33,-0.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(2.33,-0.33) -- (2.33, 0.66) -- (2.66,0.33) -- (2.66,-0.66)  -- 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=\Drt]%
(1.33,-0.33) -- (1, 0) -- (2,0) -- (2.33,-0.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb]%
(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=\Drb]%
(1.66,-0.66) -- (1.33, -0.33) -- (2.33,-0.33) -- (2.66,-0.66)  -- cycle;
}
%%
\newcommand{\DrawTwoCubeLD}{%
\DrawTwoCubeFrontFace %% frontface
%%-----------Left face--------
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(-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=\Lrt]%
(-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]%
(-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=\Drt]%
(0.66,-0.33) -- (1, 0) -- (2,0) -- (1.66,-0.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Dlb]%
(-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=\Drb]%
(0.33,-0.66) -- (0.66, -0.33) -- (1.66,-0.33) -- (1.33,-0.66)  -- cycle;
}
%%
\newcommand{\DrawTwoCubeLU}{%
\DrawTwoCubeFrontFace %% frontface
%%-----------Left face----------
%%------top row
```

```
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(-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=\Lrt]%
(-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]%
(-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---------
\draw[line join=round,line cap=round,ultra thick,fill=\Ult]%
(-0.33,2.33) -- (-0.66, 2.66) -- (0.33,2.66) -- (0.66,2.33)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urt]%
(0.66,2.33) -- (0.33, 2.66) -- (1.33,2.66) -- (1.66,2.33)  -- cycle;
%%---bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Ulb]%
(0,2) -- (-0.33, 2.33) -- (0.66,2.33) -- (1,2)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Urb]%
(1,2) -- (0.66, 2.33) -- (1.66,2.33) -- (2,2)  -- cycle;
}
```

9.9 DrawTwoFlat.. macros

These 'Flat' macros draw a specified face with its origin (left bottom corner of the face) at a specified (x, y) coordinate. They allow USERS to place the image of a face at a specific location.

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

```
\draw[line join=round,line cap=round,ultra thick,fill=\Dlt]%
(\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=\Drt]%
(\ddx + 1,\ddy + 1) -- (\ddx + 1,\ddy + 2) -- (\ddx + 2,\ddy + 2)%
 -- (\ddx + 2,\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=\Drb]%
(\ddx + 1,\ddy + 0) -- (\ddx + 1,\ddy + 1) -- (\ddx + 2,\ddy + 1)%
 -- (\ddx + 2,\ddy + 0) -- cycle;
}
%%-------------------------
\newcommand{\DrawTwoFlatLeft}[2]{%
\pgfmathsetmacro{\lx}{#1}%
\pgfmathsetmacro{\ly}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Llt]%
(\lx + 0, \ly + 1) -- (\lx + 0, \ly +  2) -- (\lx + 1, \ly + 2)%
 -- (\lx + 1, \ly + 1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrt]%
(\lx + 1, \ly + 1) -- (\lx + 1, \ly + 2) -- (\lx + 2, \ly + 2)%
 -- (\lx + 2, \ly + 1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Llb]%
(\lx + 0, \ly + 0) -- (\lx + 0, \ly + 1) -- (\lx + 1, \ly + 1)%
 -- (\lx + 1, \ly + 0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Lrb]%
(\lx + 1, \ly + 0) -- (\lx + 1, \ly + 1) -- (\lx + 2, \ly + 1)%
 -- (\lx + 2, \ly + 0) -- cycle;
}
%%-------------------------
\newcommand{\DrawTwoFlatRight}[2]{%
\pgfmathsetmacro{\rx}{#1}%
\pgfmathsetmacro{\ry}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlt]%
(\rx + 0, \ry + 1) -- (\rx + 0, \ry + 2) -- (\rx + 1, \ry + 2)%
 -- (\rx + 1, \ry + 1)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrt]%
(\rx + 1, \ry + 1) -- (\rx + 1, \ry + 2) -- (\rx + 2, \ry + 2)%
 -- (\rx + 2, \ry + 1) -- cycle;
%%----bottom row
\draw[line join=round,line cap=round,ultra thick,fill=\Rlb]%
(\rx + 0, \ry + 0) -- (\rx + 0, \ry + 1) -- (\rx + 1, \ry + 1)%
 -- (\rx + 1, \ry + 0)  -- cycle;
\draw[line join=round,line cap=round,ultra thick,fill=\Rrb]%
(\rx + 1, \ry + 0) -- (\rx + 1, \ry + 1) -- (\rx + 2, \ry + 1)%
```

```
 -- (\rx + 2, \ry + 0) -- cycle;
}
%%--------------
\newcommand{\DrawTwoFlatFront}{%
%% This command is used /only/ by  the \cmd{\DrawRubikFlat} command.
%% NOTE: x, y variables not implemented as not required here
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Flt]%
(0,1) -- (0, 2) -- (1,2) -- (1,1)  -- cycle;
%%
\draw[line join=round,line cap=round,ultra thick,fill=\Frt]%
(1,1) -- (1, 2) -- (2,2) -- (2,1) -- 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=\Frb]%
(1,0) -- (1, 1) -- (2,1) -- (2,0) -- cycle;
}
%%--------------------------
\newcommand{\DrawTwoFlatBack}[2]{%
\pgfmathsetmacro{\bx}{#1}%
\pgfmathsetmacro{\by}{#2}%
%%---top row
\draw[line join=round,line cap=round,ultra thick,fill=\Blt]%
(\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=\Brt]%
(\bx + 1,\by + 1) -- (\bx + 1,\by + 2) -- (\bx + 2,\by + 2)%
 -- (\bx + 2,\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=\Brb]%
(\bx + 1,\by + 0) -- (\bx + 1,\by + 1) -- (\bx + 2,\by + 1)%
 -- (\bx + 2,\by + 0) -- cycle;
}
%%-----------------------------
\newcommand{\DrawTwoCubeF}{%
  \DrawTwoFlatUp{0}{2}%
  \DrawTwoFlatDown{0}{-2}%
  \DrawTwoFlatLeft{-2}{0}%
  \DrawTwoFlatFront%
  \DrawTwoFlatRight{2}{0}%
  \DrawTwoFlatBack{4}{0}%
}
%%
\newcommand{\DrawTwoCubeSF}{%
  \DrawTwoCubeRU% RU
```

```
\DrawTwoFlatDown{0}{-2}%
\DrawTwoFlatLeft{-2}{0}%
\DrawTwoFlatBack{2.666}{0.66}%
}
```

9.10 Sidebars (Face)

Making sidebar macros for a TwoCube (converting the 3x3x3x versions to the 2x2x2 versions).
(1) change name → `TWOside@barX`
(2) change the value `(3 + \bs)` → `(2 + \tbs)` (as only two squares on a side).
(3) change the Sidebar length parameter names by adding a `\t` prefix to distinguish the TwoCube parameters from those of the RubikCube. Thus we change the Rubik names (`\dx`, `\dy`, `\bw`, `\bl`, `\blh`, `\bs`) to their equivalent Two names (`\tdx`, `\tdy`, `\tbw`, `\tbl`, `\tblh`, `\tbs`).

The coordinates of the bottom left corner of a TwoSidebar are (`\tdx`, `\tdy`). The other parameters are width (`\tbw`), length (`\tbl`), half length (`\tblh`), separation (`\tbs`).

`\TwoSidebarWidth` `\TwoSidebarLength` `\TwoSidebarSep`

These commands set the width, length and separation of the Sidebars. Each takes a single scalar argument (no units).
USAGE: `\TwoSidebarWidth{`⟨*0.5*⟩`}`

```
\newcommand{\TwoSidebarWidth}[1]{\pgfmathsetmacro{\tbw}{#1}}
\newcommand{\TwoSidebarLength}[1]{\pgfmathsetmacro{\tbl}{#1}}
\newcommand{\TwoSidebarSep}[1]{\pgfmathsetmacro{\tbs}{#1}}
```

We first set some default values. We have set the Sidebar width and separation to 2/3 those of the Rubik 3x3x3 values (so that when a 3x3x3 cube and a 2x2x2 cube are scaled to be the same size, then the Sidebar width and sep will be the same) These also seem to look good generally as well. Of course, users can adjust these as they wish anyway. Any new values will will of course act globally unless constrained (either by using curley brackets, or by writing them into a TEX environment).

```
\TwoSidebarWidth{0.2}
\TwoSidebarLength{1.0}
\TwoSidebarSep{0.2}
```

`\NoSidebar`

This command `\NoSidebar{`⟨*colour code*⟩`}` (defined in the RUBIKCUBE package) defines the colour for which sidebars should *not* be drawn (particularly useful when drawing OLL configurations). This idea was suggested by Robert Mařík (May 2017).
The principle is that we let the command `\NoSidebar` define a face colour, and then we use the `\ifthenelse{\equal{#2}{\no@sidebar}}{}{...}` structure to either (a) draw all sidebars as usual (if `\NoSidebar` is undefined), or (b) draw all sidebars *except* those having the `\NoSidebar` colour (if `\NoSidebar` colour = `#2`).
USAGE: `\NoSidebar{X}` If this command in *not* inside an environment, then its action will continue until it is cancelled (undefined) as follows: `\NoSidebar{}`.

\TWOside@barL
\TWOside@barR
\TWOside@barT
\TWOside@barB

These commands \TWOside@barX{⟨*position no*⟩}{⟨*facelet location*⟩} (where X is one of L (Left), R (Right), T (Top), B (Bottom), which denote the side of the 2x2 square representing a face), draw a single small bar in a position (1, or 2), having the colour of a specified facelet. The integers 1, 2 denote the facelet number (measured from the grid origin = bottom left corner of the face) adjacent to which the bar is positioned.

These macros are used in the various 'Sidebar' commands which draw pairs of these small bars adjacent to specified faces.

CHANGES: Rubik → Two, \bw → \tbw, \bl → \tbl, \bs → \tbs, \dx → \tdx, \dy → \tdy (see Section **??** for details of these variables).

USAGE: \TWOside@barL{2}{\Lrt}

```
\newcommand{\TWOside@barL}[2]{%
  %% #1 = cubie possn no, #2 = colour
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{0 - \tbs - \tbw}%
  \pgfmathsetmacro{\tdy}{#1-1+0.5-\tblh}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;
}}
%% changed  Rubik value  (3 + \bs)  --> (2 + \tbs) (as only TWO squares)
\newcommand{\TWOside@barR}[2]{%
  %% #1 = cubie possn no, #2 = colour
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{2 + \tbs}%
  \pgfmathsetmacro{\tdy}{#1 -1+0.5-\tblh}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;
}}
%%  changed Rubik value  (3 + \bs)  --> (2 + \tbs) (as only TWO squares)
\newcommand{\TWOside@barT}[2]{%
  %% #1 = cubie possn no, #2 = colour
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh}%
  \pgfmathsetmacro{\tdy}{2 +\tbs}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
}}
\newcommand{\TWOside@barB}[2]{%
  %% #1 = cubie possn no, #2 = colour
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh}%
  \pgfmathsetmacro{\tdy}{0 -\tbs-\tbw}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
```

```
}}
```

9.10.1 DrawTwoFaceXSide macros

Only 2 bars on each side for a TwoCube. Change from 3x3x3: we remove the middle cols & row sections change name DrawRubikLayerSide –¿ DrawTwoLayerSide

RWDN 16 Feb 2018 v5: removed many (duplicated and unnecessary) macros, and replaced them with these TwoFace.. macros using just the basic `\TWOside@bar..` macros for drawing small single bars.

These new macros draw a specified face (using the ..Flat.. commands) as well as all the associated sidebars. Note we continue to use the key-word 'Side' here to indicate we are drawing all the sidebars, since we are drawing a face. (only for 3D cubes do we use the word 'SidebarXX' for denoting a particular Sidebar to be drawn etc.)

```
\newcommand{\DrawTwoFaceUpSide}{%
 \DrawTwoFlatUp{0}{0}%
 \TWOside@barT{1}{\Brt}%
 \TWOside@barT{2}{\Blt}%
 \TWOside@barL{2}{\Llt}%
 \TWOside@barL{1}{\Lrt}%
 \TWOside@barR{2}{\Rrt}%
 \TWOside@barR{1}{\Rlt}%
 \TWOside@barB{1}{\Flt}%
 \TWOside@barB{2}{\Frt}%
}
\newcommand{\DrawTwoFaceFrontSide}{%
 \DrawTwoFlatFront{0}{0}%
 \TWOside@barT{1}{\Ulb}%
 \TWOside@barT{2}{\Urb}%
 \TWOside@barL{2}{\Lrt}%
 \TWOside@barL{1}{\Lrb}%
 \TWOside@barR{2}{\Rlt}%
 \TWOside@barR{1}{\Rlb}%
 \TWOside@barB{1}{\Dlt}%
 \TWOside@barB{2}{\Drt}%
}
\newcommand{\DrawTwoFaceRightSide}{%
 \DrawTwoFlatRight{0}{0}%
 \TWOside@barT{1}{\Urb}%
 \TWOside@barT{2}{\Urt}%
 \TWOside@barL{2}{\Frt}%
 \TWOside@barL{1}{\Frb}%
 \TWOside@barR{2}{\Blt}%
 \TWOside@barR{1}{\Blb}%
 \TWOside@barB{1}{\Drt}%
 \TWOside@barB{2}{\Drb}%
}
\newcommand{\DrawTwoFaceLeftSide}{%
```

```
 \DrawTwoFlatLeft{0}{0}%
 \TWOside@barT{1}{\Ult}%
 \TWOside@barT{2}{\Ulb}%
 \TWOside@barL{2}{\Brt}%
 \TWOside@barL{1}{\Brb}%
 \TWOside@barR{2}{\Flt}%
 \TWOside@barR{1}{\Flb}%
 \TWOside@barB{1}{\Dlb}%
 \TWOside@barB{2}{\Dlt}%
}
\newcommand{\DrawTwoFaceBackSide}{%
 \DrawTwoFlatBack{0}{0}%
 \TWOside@barT{1}{\Urt}%
 \TWOside@barT{2}{\Ult}%
 \TWOside@barL{2}{\Rrt}%
 \TWOside@barL{1}{\Rrb}%
 \TWOside@barR{2}{\Llt}%
 \TWOside@barR{1}{\Llb}%
 \TWOside@barB{1}{\Drb}%
 \TWOside@barB{2}{\Dlb}%
}
\newcommand{\DrawTwoFaceDownSide}{%
 \DrawTwoFlatDown{0}{0}%
 \TWOside@barT{1}{\Flb}%
 \TWOside@barT{2}{\Frb}%
 \TWOside@barL{2}{\Lrb}%
 \TWOside@barL{1}{\Llb}%
 \TWOside@barR{2}{\Rlb}%
 \TWOside@barR{1}{\Rrb}%
 \TWOside@barB{1}{\Brb}%
 \TWOside@barB{2}{\Blb}%
}
%%
%% v5: made Face versions (for USER) without the (x,y) coordinates
\newcommand{\DrawTwoFaceUp}{\DrawTwoFlatUp{0}{0}}
\newcommand{\DrawTwoFaceDown}{\DrawTwoFlatDown{0}{0}}
\newcommand{\DrawTwoFaceLeft}{\DrawTwoFlatLeft{0}{0}}
\newcommand{\DrawTwoFaceRight}{\DrawTwoFlatRight{0}{0}}
\newcommand{\DrawTwoFaceFront}{\DrawTwoFlatFront{0}{0}}
\newcommand{\DrawTwoFaceBack}{\DrawTwoFlatBack{0}{0}}
%%
%% v5: made short forms --> abbreviations
\newcommand{\DrawTwoFaceU}{\DrawTwoFlatUp{0}{0}}
\newcommand{\DrawTwoFaceD}{\DrawTwoFlatDown{0}{0}}
\newcommand{\DrawTwoFaceL}{\DrawTwoFlatLeft{0}{0}}
\newcommand{\DrawTwoFaceR}{\DrawTwoFlatRight{0}{0}}
\newcommand{\DrawTwoFaceF}{\DrawTwoFlatFront{0}{0}}
\newcommand{\DrawTwoFaceB}{\DrawTwoFlatBack{0}{0}}
%%
%% v5: made short forms --> abbreviations
```

```
\newcommand{\DrawTwoFaceUS}{\DrawTwoFaceUpSide}
\newcommand{\DrawTwoFaceDS}{\DrawTwoFaceDownSide}
\newcommand{\DrawTwoFaceLS}{\DrawTwoFaceLeftSide}
\newcommand{\DrawTwoFaceRS}{\DrawTwoFaceRightSide}
\newcommand{\DrawTwoFaceFS}{\DrawTwoFaceFrontSide}
\newcommand{\DrawTwoFaceBS}{\DrawTwoFaceBackSide}
```

9.11 Sidebars (Cube)

In order to position sidebars adjacent to a TwoCube (ie in 3D) requires that we first make some new `\TWOside@bar..` commands which draw sidebars adjacent to the BACK face (we have already made the macros for the front face sidebars—see Section 9.10). Furthermore, these new macros need to be tailored to each of the four standard cube viewing directions RU, LU, RD, LD.

Finally, the USER commands for drawing these sidebars need to accommodate (a) some code for identifying each set of sidebars, and (b) the viewing direction. So, for example, a USER command for drawing the sidebars associated with the cube edge formed by the RIGHT face and the BACK face (lets define this as the RB sidebar) as viewed from the RU direction, might be something like `\DrawTwoCubeSidebarRBRU`. Since this is not particularly user-friendly, we can improve on this slightly for the USER by (a) defining the sidebar as `SidebarRB`, and (b) appending the view direction in a curly bracket, say as `{RU}`. This allows a more intuitive command structure for the USER, as follows: `\DrawTwoCubeSidebarRB{RU}`. We then use the `\@join` command to append the string `RU` to the string `DrawTwoCubeSidebarRB` forming the (internal) command `\DrawTwoCubeSidebarRBRU`.

In the following we will group the development according to to the view direction.

9.11.1 Sidebars: RU view

Right-Back vert sidebar (RU view)

Need to write a new `\TWOside@barR..` command (see Section 9.10). This draws only a single small bar. Each of the two small bars has a numbered position (1,2); (dx,dy) = bottom Left corner of single bar

```
\newcommand{\TWOside@barRbackRU}[2]{%
  %% #1 = cubie possn no, #2 = colour
  %% tdx --> tdx + (2/3)
  %% tdy --> tdy + (2/3)
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{2 + \tbs        +0.666}%
  \pgfmathsetmacro{\tdy}{#1 -1+0.5-\tblh +0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;%
}}
```

Now make the RB (RightBack) vertical sidebar command; ie bar 1 is at the bottom; bar 2 is at the top.

```
\newcommand{\DrawTwoCubeSidebarRBRU}{%
 \TWOside@barRbackRU{2}{\Blt}%
 \TWOside@barRbackRU{1}{\Blb}%
}
```

Now make the reverse command (BR) = RB

```
\newcommand{\DrawTwoCubeSidebarBRRU}{\DrawTwoCubeSidebarRBRU}
```

Finally, make the join commands

```
\newcommand{\DrawTwoCubeSidebarRB}[1]{\@join{\DrawTwoCubeSidebarRB}{#1}}
\newcommand{\DrawTwoCubeSidebarBR}[1]{\@join{\DrawTwoCubeSidebarBR}{#1}}
```

Up-Back horiz sidebar (RU view)

```
\newcommand{\TWOside@barTbackRU}[2]{%
  %% #1 = cubie possn no; #2 = colour
  %% tdx --> tdx + (2/3)
  %% tdy --> tdy + (2/3)
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh +0.666}%
  \pgfmathsetmacro{\tdy}{2 +\tbs         +0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
}}
```

Now make the UB (Up-Back) horizontal sidebar command; ie bar 1 is on the left, bar 2 is on the right (as we look at the cube).

```
\newcommand{\DrawTwoCubeSidebarUBRU}{%
 \TWOside@barTbackRU{1}{\Brt}%
 \TWOside@barTbackRU{2}{\Blt}%
}
```

Now make the reverse command (BU) = UB

```
\newcommand{\DrawTwoCubeSidebarBURU}{\DrawTwoCubeSidebarUBRU}
```

Now make the join commands

```
\newcommand{\DrawTwoCubeSidebarUB}[1]{\@join{\DrawTwoCubeSidebarUB}{#1}}
\newcommand{\DrawTwoCubeSidebarBU}[1]{\@join{\DrawTwoCubeSidebarBU}{#1}}
```

Front-Left vert sidebar (RU view)

Since this is a front-face sidebar we can use the regular `\WOside@barL..` command. Now make the FL (Front-Left) vertical sidebar command; ie bar 1 is at the bottom; bar 2 is at the top.

```
\newcommand{\DrawTwoCubeSidebarFLRU}{%
 \TWOside@barL{2}{\Lrt}%
 \TWOside@barL{1}{\Lrb}%
}
```

Now do the reverse (LF)

```
\newcommand{\DrawTwoCubeSidebarLFRU}{\DrawTwoCubeSidebarFLRU}
```

Now do the two join commands

```
\newcommand{\DrawTwoCubeSidebarFL}[1]{\@join{\DrawTwoCubeSidebarFL}{#1}}
\newcommand{\DrawTwoCubeSidebarLF}[1]{\@join{\DrawTwoCubeSidebarLF}{#1}}
```

Front-Down horiz sidebar (RU view)

Since this is a front face sidebar we can use the regular `\WOside@barL..` command.

```
\newcommand{\DrawTwoCubeSidebarFDRU}{%
 \TWOside@barB{1}{\Dlt}%
 \TWOside@barB{2}{\Drt}%
}
```

Now do the reverse (DF) = FD

```
\newcommand{\DrawTwoCubeSidebarDFRU}{\DrawTwoCubeSidebarFDRU}
```

Now do the two join commands

```
\newcommand{\DrawTwoCubeSidebarFD}[1]{\@join{\DrawTwoCubeSidebarFD}{#1}}
\newcommand{\DrawTwoCubeSidebarDF}[1]{\@join{\DrawTwoCubeSidebarDF}{#1}}
```

But FD-LU is the same as FD-RU, so we need to make copies of each.

```
\newcommand{\DrawTwoCubeSidebarDFLU}{\DrawTwoCubeSidebarDFRU}
\newcommand{\DrawTwoCubeSidebarFDLU}{\DrawTwoCubeSidebarFDRU}
```

9.11.2 Sidebars: LU view

Left-Back vert sidebar (LU view)

```
\newcommand{\TWOside@barLbackLU}[2]{%
  %% #1 = cubie possn no, #2 = colour
  %% tdx --> tdx - 2/3
  %% tdy --> tdy + 2/3
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{0 - \tbs -\tbw   -0.666}%
  \pgfmathsetmacro{\tdy}{#1 -1+0.5-\tblh +0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;
}}
```

Now make the LB (LeftBack) vertical sidebar command; bar 1 is at the bottom

```
\newcommand{\DrawTwoCubeSidebarLBLU}{%
 \TWOside@barLbackLU{2}{\Brt}%
 \TWOside@barLbackLU{1}{\Brb}%
}
```

Now do the reverse (BL) = LB

```
\newcommand{\DrawTwoCubeSidebarBLLU}{\DrawTwoCubeSidebarLBLU}
```

Now make the join commands

```
\newcommand{\DrawTwoCubeSidebarLB}[1]{\@join{\DrawTwoCubeSidebarLB}{#1}}
\newcommand{\DrawTwoCubeSidebarBL}[1]{\@join{\DrawTwoCubeSidebarBL}{#1}}
```

Up-Back horizontal sidebar (LU view)

```
\newcommand{\TWOside@barTbackLU}[2]{%
  %% #1 = cubie possn no; #2 = colour
  %% tdx --> tdx-2/3
  %% tdy --> tdy+2/3
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh  -0.666}%
  \pgfmathsetmacro{\tdy}{2 +\tbs          +0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
}}
```

Now make the UB (Up-Back) version bar 1 is at the left, 2 on the right.

```
\newcommand{\DrawTwoCubeSidebarUBLU}{%
 \TWOside@barTbackLU{1}{\Brt}%
 \TWOside@barTbackLU{2}{\Blt}%
}
```

Now do the reverse (BU) = UB

```
\newcommand{\DrawTwoCubeSidebarBULU}{\DrawTwoCubeSidebarUBLU}
```

We do NOT need to make the join commands here as the USER commands for BU and UB are the same as for the RU

Front-Right vertical sidebar (LU view)

```
\newcommand{\DrawTwoCubeSidebarFRLU}{%
 \TWOside@barR{2}{\Rlt}%
 \TWOside@barR{1}{\Rlb}%
}
```

Now do the reverse (RF)

```
\newcommand{\DrawTwoCubeSidebarRFLU}{\DrawTwoCubeSidebarFRLU}
```

Now do the two join commands

```
\newcommand{\DrawTwoCubeSidebarFR}[1]{\@join{\DrawTwoCubeSidebarFR}{#1}}
\newcommand{\DrawTwoCubeSidebarRF}[1]{\@join{\DrawTwoCubeSidebarRF}{#1}}
```

9.11.3 Sidebars: RD view

Front-Up horizontal sidebar (RD view)

```
\newcommand{\DrawTwoCubeSidebarFURD}{%
 \TWOside@barT{1}{\Ulb}%
 \TWOside@barT{2}{\Urb}%
}
```

Now do the reverse (UF) = FU

```
\newcommand{\DrawTwoCubeSidebarUFRD}{\DrawTwoCubeSidebarFURD}
```

Now do the two join commands

```
\newcommand{\DrawTwoCubeSidebarFU}[1]{\@join{\DrawTwoCubeSidebarFU}{#1}}
\newcommand{\DrawTwoCubeSidebarUF}[1]{\@join{\DrawTwoCubeSidebarUF}{#1}}
```

Front-Left vertical sidebar (RD view)

Only need to copy an earlier command here since FL, RD view = same as for RU view.

```
\newcommand{\DrawTwoCubeSidebarFLRD}{\DrawTwoCubeSidebarFLRU}
\newcommand{\DrawTwoCubeSidebarLFRD}{\DrawTwoCubeSidebarLFRU}
```

Right-Back vertical sidebar (RD view)

```
\newcommand{\TWOside@barRbackRD}[2]{%
  %% #1 = cubie possn no, #2 = colour
  %% tdx --> tdx + (2/3)
  %% tdy --> tdy - (2/3)
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{2 + \tbs        +0.666}%
  \pgfmathsetmacro{\tdy}{#1 -1+0.5-\tblh -0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;
}}
```

Now make the RB (RightBack) version bar 1 is at the bottom

```
\newcommand{\DrawTwoCubeSidebarRBRD}{%
\TWOside@barRbackRD{2}{\Blt}%
\TWOside@barRbackRD{1}{\Blb}%
}
```

now do the reverse (BR) = RB

```
\newcommand{\DrawTwoCubeSidebarBRRD}{\DrawTwoCubeSidebarRBRD}
```

Do NOT need to make the join commands (as same as for the RU view)

Down-Back horizotal sidebar (RD view)

```
\newcommand{\TWOside@barBbackRD}[2]{%
  %% #1 = cubie possn no; #2 = colour
  %% tdx --> tdx+2/3
  %% tdy --> tdy-2/3
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh +0.666}%
  \pgfmathsetmacro{\tdy}{0 -\tbs - \tbw   -0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
}}
```

Now make the DB (Down-Back) version bar 1 is at the left, 2 on the right (as we look at the cube)

```
\newcommand{\DrawTwoCubeSidebarDBRD}{%
 \TWOside@barBbackRD{1}{\Brb}%
 \TWOside@barBbackRD{2}{\Blb}%
}
```

Now do the reverse (BD) = DB

```
\newcommand{\DrawTwoCubeSidebarBDRD}{\DrawTwoCubeSidebarDBRD}
```

Now make the join commands

```
\newcommand{\DrawTwoCubeSidebarDB}[1]{\@join{\DrawTwoCubeSidebarDB}{#1}}
\newcommand{\DrawTwoCubeSidebarBD}[1]{\@join{\DrawTwoCubeSidebarBD}{#1}}
```

9.11.4 Sidebars: LD view

Front-Up horizotal sidebar (LD view)

But FU (LD view) is the same as for (RU view), (see above)

```
\newcommand{\DrawTwoCubeSidebarFULD}{\DrawTwoCubeSidebarFURD}
\newcommand{\DrawTwoCubeSidebarUFLD}{\DrawTwoCubeSidebarUFRD}
```

Front-Right vertical sidebar (LD view)

But FR (LDview) is the same as for (LU view), (see above)

```
\newcommand{\DrawTwoCubeSidebarFRLD}{\DrawTwoCubeSidebarFRLU}
\newcommand{\DrawTwoCubeSidebarRFLD}{\DrawTwoCubeSidebarRFLU}
```

Left-Back vertical sidebar (LD view)

```
\newcommand{\TWOside@barLbackLD}[2]{%
  %% #1 = cubie possn no, #2 = colour
  %% tdx --> tdx-2/3
  %% tdy --> tdy-2/3
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{0 - \tbs -\tbw   -0.666}%
  \pgfmathsetmacro{\tdy}{#1 -1+0.5-\tblh -0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbl)
  -- (\tdx+\tbw,\tdy+\tbl) -- (\tdx+\tbw,\tdy) -- cycle;
}}
```

Now make the LB (LeftBack) version bar 1 is at the bottom

```
\newcommand{\DrawTwoCubeSidebarLBLD}{%
 \TWOside@barLbackLD{2}{\Brt}%
 \TWOside@barLbackLD{1}{\Brb}%
}
```

Now do the reverse (BL) = LB

```
\newcommand{\DrawTwoCubeSidebarBLLD}{\DrawTwoCubeSidebarLBLD}
```

Do NOT need to make the join commands (same as for the LU view)

Down-Back horizontal sidebar (LD view)

```
\newcommand{\TWOside@barBbackLD}[2]{%
  %% #1 = cubie possn no; #2 = colour
  %% tdx --> tdx-2/3
  %% tdy --> tdy-2/3
  \ifthenelse{\equal{#2}{\no@sidebar}}{}{%
  \pgfmathsetmacro{\tblh}{\tbl*(0.5)}%
  \pgfmathsetmacro{\tdx}{#1 -1+0.5-\tblh -0.666}%
  \pgfmathsetmacro{\tdy}{0 -\tbs - \tbw   -0.666}%
  \draw[fill=#2] (\tdx,\tdy) -- (\tdx,\tdy + \tbw)
  -- (\tdx+\tbl,\tdy+\tbw) -- (\tdx+\tbl,\tdy) -- cycle;
}}
```

Now make the DB (Down-Back) version bar 1 is at the left, 2 on the right (as we look at the cube)

```
\newcommand{\DrawTwoCubeSidebarDBLD}{%
\TWOside@barBbackLD{1}{\Brb}%
\TWOside@barBbackLD{2}{\Blb}%
}
```

Now do the reverse (BD) = DB

```
\newcommand{\DrawTwoCubeSidebarBDLD}{\DrawTwoCubeSidebarDBLD}
```

Do NOT need to make any join commands (same as for the RD view)

9.12 Hieroglyphs

Not many changes to make (from rubikcube sty). In general we try to keep things fairly intuitive by changing Rubik → Two, and changing r → t. Note that since this package uses a lot of commands defined in the `rubikcube` package, eg the 'join' utility and the rubikfont, consequently rubikcube sty needs to be loaded when running this package.

We only need to make significant changes to the following hieroglyphs: L,R,U,D,Lp,Rp,Up,Dp.

Unchanged are: axis rotations (eg Rc) and letter rotations (eg B), so these hieroglyphs just need newdefs making for them; eg trF ← rrF, etc

We need to rename some of the 'square' furniture associated with the L,R,U,D etc face rotation hieroglyphs: for example, the D heiroglyph requires the following:
change rr → tr
change SquareD → SquaretD
change RubikD → TwoD
change textRubikD → textTwoD

For a TWOcube we only need to make two lines in a square; ie we want to shift the top line down/sideways and shift the bottom line up/sideways by an amount which makes the final position = 1/3 of the square. Since the top and bottom lines (and also the left and right lines) are at 0.25 unit, then the extra distance = 0.25/3 = 0.0833; so for horiz lines we add/subtract @ty, and for vertical lines we add/subtract @tx.

`\@tx`
`\@ty`

```
\pgfmathsetmacro{\@tx}{0.0833}
\pgfmathsetmacro{\@ty}{0.0833}
```

We continue to use the rubikfont.

\@tr
\@trp

```
\newcommand{\@tr}[1]{{\@rubikfont #1}}
\newcommand{\@trp}[1]{{\@rubikfont #1\@rubikprime}}
```

We need to rename the basic 'join' commands: ie change Rubik → Two, and change r → t, as follows:

\tr
\trh
\Two
\textTwo

```
\newcommand*{\tr}[1]{\@join{\tr}{#1}}
\newcommand*{\trh}[1]{\@join{\trh}{#1}}
\newcommand*{\Two}[1]{\@join{\Two}{#1}}
\newcommand*{\textTwo}[1]{\@join{\textTwo}{#1}}
```

9.12.1 Rotation B

\trB
\trhB
\TwoB
\textTwoB

```
\newcommand{\trB}{\rrB}
\newcommand{\trBp}{\rrBp}
\newcommand{\trhB}{\rrhB}
\newcommand{\trhBp}{\rrhBp}
\newcommand{\TwoB}{\RubikB}
\newcommand{\TwoBp}{\RubikBp}
\newcommand{\textTwoB}{\textRubikB}
\newcommand{\textTwoBp}{\textRubikBp}
```

9.12.2 Rotation D

We need to rename some of the items as follows:
change rr → tr
change SquareD → SquaretD
change RubikD → TwoD
change textRubikD → textTwoD

\trD
\SquaretD
\trhD
\TwoD
\textTwoD

These commands all draw forms which denote the D rotation.

Feb 2017 (RWDN): added the `\@tlen` length to the `\trhD` command to improve the spacing between two 'arrow' square hieroglyphs; and also removed the terminal `\,` space. The same changes were made to all the 'arrow' hieroglyphs.

```
%%
\newcommand{\trD}{\@tr{D}}
%%
\newcommand{\SquaretD}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
```

```
\draw [thick] (\@sb,\@sddd - \@ty) -- (\@sbh, \@sddd - \@ty);
\draw [thick, ->]     (\@sb,\@sd + \@ty) --   (\@sbh, \@sd + \@ty);
\end{tikzpicture}%
}
\newcommand{\trhD}{\raisebox{-0.333\height}{\@tlen\SquaretD\@tlen}}
%%
\newcommand{\TwoD}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretD\\
\trD%
\end{minipage}%
}}
\newcommand{\textTwoD}{\trD\,\trhD}
```

9.12.3 Rotation Dp

\trDp
\SquaretDp
\trhDp
\TwoDp
\textTwoDp

These commands all draw forms which denote the Dp rotation.

```
\newcommand{\trDp}{\@trp{D}}
%%
\newcommand{\SquaretDp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick]     (\@sb,\@sddd - \@ty) -- (\@sbh, \@sddd - \@ty);
\draw [thick, <-] (\@sb,\@sd   + \@ty)  -- (\@sbh, \@sd + \@ty);
\end{tikzpicture}%
}
\newcommand{\trhDp}{\raisebox{-0.333\height}{\@tlen\SquaretDp\@tlen}}
%%
\newcommand{\TwoDp}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretDp\\
\trDp%
\end{minipage}%
}}
\newcommand{\textTwoDp}{\trDp\,\trhDp}
```

9.12.4 Rotation F

\trF
\trhF
\TwoF
\textTwoF

```
\newcommand{\trF}{\rrF}
\newcommand{\trFp}{\rrFp}
\newcommand{\trhF}{\rrhF}
\newcommand{\trhFp}{\rrhFp}
\newcommand{\TwoF}{\RubikF}
\newcommand{\TwoFp}{\RubikFp}
```

```
\newcommand{\textTwoF}{\textRubikF}
\newcommand{\textTwoFp}{\textRubikFp}
```

9.12.5 Rotation L

\trL
\SquaretL
\trhL
\TwoL
\textTwoL

These commands all draw forms which denote the L rotation.

```
\newcommand{\trL}{\@tr{L}}
%%
\newcommand{\SquaretL}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\@sd + \@tx, \@sb) -- (\@sd + \@tx, \@sbh);
\draw [thick] (\@sddd - \@tx, \@sb) -- (\@sddd - \@tx, \@sbh);
\end{tikzpicture}%
}
\newcommand{\trhL}{\raisebox{-0.333\height}{\@tlen\SquaretL\@tlen}}
%%
\newcommand{\TwoL}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretL\\
\trL%
\end{minipage}%
}}
\newcommand{\textTwoL}{\trL\,\trhL}
```

9.12.6 Rotation Lp

\trLp
\SquaretLp
\trhLp
\TwoLp
\textTwoLp

These commands all draw forms which denote the Lp rotation.

```
\newcommand{\trLp}{\@trp{L}}
%%
\newcommand{\SquaretLp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick,->] (\@sd + \@tx, \@sb) -- (\@sd + \@tx, \@sbh);
\draw [thick] (\@sddd - \@tx, \@sb) -- (\@sddd - \@tx, \@sbh);
\end{tikzpicture}%
}
\newcommand{\trhLp}{\raisebox{-0.333\height}{\@tlen\SquaretLp\@tlen}}
%%
\newcommand{\TwoLp}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretLp\\
\trLp%
\end{minipage}%
}}
```

```
\newcommand{\textTwoLp}{\trLp\,\trhLp}
```

9.12.7 Rotation R

\trR
\SquaretR
\trhR
\TwoR
\textTwoR

These commands all draw forms which denote the R rotation.

```
\newcommand{\trR}{\@tr{R}}
%%
\newcommand{\SquaretR}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
%% draw three lines in the square, one with an arrow
\draw [thick] (\@sd + \@tx, \@sb) -- (\@sd + \@tx, \@sbh);
\draw [thick, ->] (\@sddd - \@tx, \@sb) -- (\@sddd - \@tx, \@sbh);
\end{tikzpicture}%
}
\newcommand{\trhR}{\raisebox{-0.333\height}{\@tlen\SquaretR\@tlen}}
%%
\newcommand{\TwoR}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretR\\
\trR%
\end{minipage}%
}}
\newcommand{\textTwoR}{\trR\,\trhR}
```

9.12.8 Rotation Rp

\trRp
\SquaretRp
\trhRp
\TwoRp
\textTwoRp

These commands all draw forms which denote the Rp rotation.

```
\newcommand{\trRp}{\@trp{R}}
%%
\newcommand{\SquaretRp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick] (\@sd + \@tx, \@sb) -- (\@sd + \@tx, \@sbh);
\draw [thick, <-] (\@sddd - \@tx, \@sb) -- (\@sddd - \@tx, \@sbh);
\end{tikzpicture}%
}
\newcommand{\trhRp}{\raisebox{-0.333\height}{\@tlen\SquaretRp\@tlen}}
%%
\newcommand{\TwoRp}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretRp\\
\trRp%
\end{minipage}%
}}
```

```
\newcommand{\textTwoRp}{\trRp\,\trhRp}
```

9.12.9 Rotation U

\trU
\SquaretU
\trhU
\TwoU
\textTwoU

These commands all draw forms which denote the U rotation.

```
\newcommand{\trU}{\@tr{U}}
%%
\newcommand{\SquaretU}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, <-] (\@sb,\@sddd - \@ty) -- (\@sbh, \@sddd - \@ty);
\draw [thick]     (\@sb,\@sd + \@ty) --   (\@sbh, \@sd + \@ty);
\end{tikzpicture}%
}
\newcommand{\trhU}{\raisebox{-0.333\height}{\@tlen\SquaretU\@tlen}}
%%
\newcommand{\TwoU}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretU\\
\trU%
\end{minipage}%%
}}
\newcommand{\textTwoU}{\trU\,\trhU}
```

9.12.10 Rotation Up

\trUp
\SquaretUp
\trhUp
\TwoUp
\textTwoUp

These commands all draw forms which denote the Up rotation.

```
\newcommand{\trUp}{\@trp{U}}
%%
\newcommand{\SquaretUp}{%
\begin{tikzpicture}[scale=0.5]
\DrawNotationBox;
\draw [thick, ->] (\@sb,\@sddd - \@ty) -- (\@sbh, \@sddd - \@ty);
\draw [thick]     (\@sb,\@sd + \@ty) --   (\@sbh, \@sd + \@ty);
\end{tikzpicture}%
}
\newcommand{\trhUp}{\raisebox{-0.333\height}{\@tlen\SquaretUp\@tlen}}
%%
\newcommand{\TwoUp}{%
{\@rubikfont%
\begin{minipage}{0.6cm}
\centering%
\SquaretUp\\
\trUp%
\end{minipage}%%
}}
\newcommand{\textTwoUp}{\trUp\,\trhUp}
```

9.13 Axis rotations

For completeness we include a \textTwo version of all the axis rotation codes (making them equal to their hieroglyphic \rrh version). Obviously this list must go at the end of this file. While these commands are perhaps strictly unnecessary, the motivation is to allow users to include them in a \ShowSequence command when using the \textTwo font argument.

```
\newcommand{\trx}{\rrx}
\newcommand{\trxp}{\rrxp}
\newcommand{\try}{\rry}
\newcommand{\tryp}{\rryp}
\newcommand{\trz}{\rrz}
\newcommand{\trzp}{\rrzp}
\newcommand{\trl}{\rrl}
  \newcommand{\trlp}{\rrlp} %%new
\newcommand{\trr}{\rrr}
  \newcommand{\trrp}{\rrrp} %%new
\newcommand{\tru}{\rru}
  \newcommand{\trup}{\rrup} %%new
\newcommand{\trd}{\rrd}
  \newcommand{\trdp}{\rrdp} %%new
\newcommand{\trf}{\rrf}
  \newcommand{\trfp}{\rrfp} %%new
\newcommand{\trb}{\rrb}
  \newcommand{\trbp}{\rrbp} %%new
%
\newcommand{\trLc}{\rrLc}
\newcommand{\trLcp}{\rrLcp}
\newcommand{\trRc}{\rrRc}
\newcommand{\trRcp}{\rrRcp}
\newcommand{\trUc}{\rrUc}
\newcommand{\trUcp}{\rrUcp}
\newcommand{\trDc}{\rrDc}
\newcommand{\trDcp}{\rrDcp}
\newcommand{\trFc}{\rrFc}
\newcommand{\trFcp}{\rrFcp}
\newcommand{\trBc}{\rrBc}
\newcommand{\trBcp}{\rrBcp}
\newcommand{\trCL}{\rrCL}
\newcommand{\trCLp}{\rrCLp}
\newcommand{\trCR}{\rrCR}
\newcommand{\trCRp}{\rrCRp}
\newcommand{\trCU}{\rrCU}
\newcommand{\trCUp}{\rrCUp}
\newcommand{\trCD}{\rrCD}
\newcommand{\trCDp}{\rrCDp}
\newcommand{\trCF}{\rrCF}
\newcommand{\trCFp}{\rrCFp}
\newcommand{\trCB}{\rrCB}
```

```
\newcommand{\trCBp}{\rrCBp}

\newcommand{\trhx}{\rrhx}
\newcommand{\trhxp}{\rrhxp}
\newcommand{\trhy}{\rrhy}
\newcommand{\trhyp}{\rrhyp}
\newcommand{\trhz}{\rrhz}
\newcommand{\trhzp}{\rrhzp}
\newcommand{\trhl}{\rrhl}
   \newcommand{\trhlp}{\rrhlp} % new
\newcommand{\trhr}{\rrhr}
   \newcommand{\trhrp}{\rrhrp} % new
\newcommand{\trhu}{\rrhu}
   \newcommand{\trhup}{\rrhup} % new
\newcommand{\trhd}{\rrhd}
   \newcommand{\trhdp}{\rrhdp} % new
\newcommand{\trhf}{\rrhf}
   \newcommand{\trhfp}{\rrhfp} % new
\newcommand{\trhb}{\rrhb}
   \newcommand{\trhbp}{\rrhbp} % new
\newcommand{\trhLc}{\rrhLc}
\newcommand{\trhLcp}{\rrhLcp}
\newcommand{\trhRc}{\rrhRc}
\newcommand{\trhRcp}{\rrhRcp}
\newcommand{\trhUc}{\rrhUc}
\newcommand{\trhUcp}{\rrhUcp}
\newcommand{\trhDc}{\rrhDc}
\newcommand{\trhDcp}{\rrhDcp}
\newcommand{\trhFc}{\rrhFc}
\newcommand{\trhFcp}{\rrhFcp}
\newcommand{\trhBc}{\rrhBc}
\newcommand{\trhBcp}{\rrhBcp}
\newcommand{\trhCL}{\rrhCL}
\newcommand{\trhCLp}{\rrhCLp}
\newcommand{\trhCR}{\rrhCR}
\newcommand{\trhCRp}{\rrhCRp}
\newcommand{\trhCU}{\rrhCU}
\newcommand{\trhCUp}{\rrhCUp}
\newcommand{\trhCD}{\rrhCD}
\newcommand{\trhCDp}{\rrhCDp}
\newcommand{\trhCF}{\rrhCF}
\newcommand{\trhCFp}{\rrhCFp}
\newcommand{\trhCB}{\rrhCB}
\newcommand{\trhCBp}{\rrhCBp}

\newcommand{\Twox}{\Rubikx}
\newcommand{\Twoxp}{\Rubikxp}
\newcommand{\Twoy}{\Rubiky}
\newcommand{\Twoyp}{\Rubikyp}
\newcommand{\Twoz}{\Rubikz}
\newcommand{\Twozp}{\Rubikzp}
```

```
\newcommand{\Twol}{\Rubikl}
   \newcommand{\Twolp}{\Rubiklp} % new
\newcommand{\Twor}{\Rubikr}
   \newcommand{\Tworp}{\Rubikrp} % new
\newcommand{\Twou}{\Rubiku}
   \newcommand{\Twoup}{\Rubikup} % new
\newcommand{\Twod}{\Rubikd}
   \newcommand{\Twodp}{\Rubikdp} % new
\newcommand{\Twof}{\Rubikf}
   \newcommand{\Twofp}{\Rubikfp} % new
\newcommand{\Twob}{\Rubikb}
   \newcommand{\Twobp}{\Rubikbp} % new
\newcommand{\TwoLc}{\RubikLc}
\newcommand{\TwoLcp}{\RubikLcp}
\newcommand{\TwoRc}{\RubikRc}
\newcommand{\TwoRcp}{\RubikRcp}
\newcommand{\TwoUc}{\RubikUc}
\newcommand{\TwoUcp}{\RubikUcp}
\newcommand{\TwoDc}{\RubikDc}
\newcommand{\TwoDcp}{\RubikDcp}
\newcommand{\TwoFc}{\RubikFc}
\newcommand{\TwoFcp}{\RubikFcp}
\newcommand{\TwoBc}{\RubikBc}
\newcommand{\TwoBcp}{\RubikBcp}
\newcommand{\TwoCL}{\RubikCL}
\newcommand{\TwoCLp}{\RubikCLp}
\newcommand{\TwoCR}{\RubikCR}
\newcommand{\TwoCRp}{\RubikCRp}
\newcommand{\TwoCU}{\RubikCU}
\newcommand{\TwoCUp}{\RubikCUp}
\newcommand{\TwoCD}{\RubikCD}
\newcommand{\TwoCDp}{\RubikCDp}
\newcommand{\TwoCF}{\RubikCF}
\newcommand{\TwoCFp}{\RubikCFp}
\newcommand{\TwoCB}{\RubikCB}
\newcommand{\TwoCBp}{\RubikCBp}

\newcommand{\textTwox}{\rrhx}
\newcommand{\textTwoxp}{\rrhxp}
\newcommand{\textTwoy}{\rrhy}
\newcommand{\textTwoyp}{\rrhyp}
\newcommand{\textTwoz}{\rrhz}
\newcommand{\textTwozp}{\rrhzp}
\newcommand{\textTwol}{\rrhl}
   \newcommand{\textTwolp}{\rrhlp} %new
\newcommand{\textTwor}{\rrhr}
   \newcommand{\textTworp}{\rrhrp} %new
\newcommand{\textTwou}{\rrhu}
   \newcommand{\textTwoup}{\rrhup} %new
\newcommand{\textTwod}{\rrhd}
```

```
    \newcommand{\textTwodp}{\rrhdp} %new
\newcommand{\textTwof}{\rrhf}
    \newcommand{\textTwofp}{\rrhfp} %new
\newcommand{\textTwob}{\rrhb}
    \newcommand{\textTwobp}{\rrhbp} %new
\newcommand{\textTwoLc}{\rrhLc}
\newcommand{\textTwoLcp}{\rrhLcp}
\newcommand{\textTwoRc}{\rrhRc}
\newcommand{\textTwoRcp}{\rrhRcp}
\newcommand{\textTwoUc}{\rrhUc}
\newcommand{\textTwoUcp}{\rrhUcp}
\newcommand{\textTwoDc}{\rrhDc}
\newcommand{\textTwoDcp}{\rrhDcp}
\newcommand{\textTwoFc}{\rrhFc}
\newcommand{\textTwoFcp}{\rrhFcp}
\newcommand{\textTwoBc}{\rrhBc}
\newcommand{\textTwoBcp}{\rrhBcp}
\newcommand{\textTwoCL}{\rrhCL}
\newcommand{\textTwoCLp}{\rrhCLp}
\newcommand{\textTwoCR}{\rrhCR}
\newcommand{\textTwoCRp}{\rrhCRp}
\newcommand{\textTwoCU}{\rrhCU}
\newcommand{\textTwoCUp}{\rrhCUp}
\newcommand{\textTwoCD}{\rrhCD}
\newcommand{\textTwoCDp}{\rrhCDp}
\newcommand{\textTwoCF}{\rrhCF}
\newcommand{\textTwoCFp}{\rrhCFp}
\newcommand{\textTwoCB}{\rrhCB}
\newcommand{\textTwoCBp}{\rrhCBp}
```

End of this package

⟨/rubiktwocube⟩

Index

Numbers written in italic refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in roman refer to the code lines where the entry is used.

E

F

S

T