The 'pst-ob3d' package
A PSTricks package
for three dimensional basic objects

Denis Girou*   Herbert Voß†

Version 0.21
August 22, 2007

Abstract

This package allow to draw basic three dimensional objects. Up to now only cubes (which can be deformed to rectangular parallelipiped ones) and dies (which are only a special kind of cubes) are defined.

Contents

*CNRS/IDRIS — Centre National de la Recherche Scientifique / Institut du Développement et des Ressources en Informatique Scientifique — Orsay — France — `Denis.Girou@idris.fr`.

†`hvoss@tug.org` for the CTAN Version

1   Introduction

'pst-ob3d' define basic three dimensional objects. Up to now only cubes of several kinds are defined (as this is the easiest thing to do!), so the interest of this package is still limited...

As usual, all the relevant PSTricks parameters can be use and few ones are added specially for these 3d objects.

You must take care that these objects are *pure graphics* ones, that is to say that they have no dimension (in any case, it would be very difficult to compute their bounding boxes accurately, according to the user point of vue chosen — the *viewpoint*). So, we have to compute their sizes and to put these objects in a `pspicture` environment by ourselves.
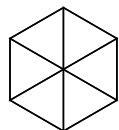
2   Usage

2.1   Macros

Two macros are currently defined: `\PstCube` to draw a cube and `\PstDie` to draw a die (which is only a cube with dots on it faces).
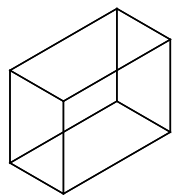
2.1.1   Cubes

`\PstCube` has three required parameters, respectively for the X, Y and Z lengths, as we can distord the cubes to parallelipiped ones.
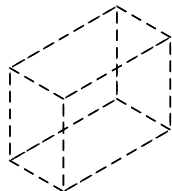
```
\PstCube[optional parameters]{X length}{Y length}{Z length}
```
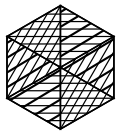


```
1 \PstCube{1}{1}{1}
```
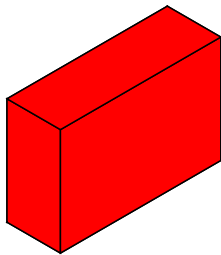


```
1 \PstCube{1}{2}{1.5}
```



```
1 \PstCube[linestyle=dashed]{1}{2}{1.5}
```
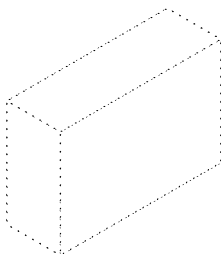
2

```
1  \PstCube[fillstyle=hlines]{1}{1}{1}
```

As special care is made to draw the faces in the right order, according to the user point of view, to draw last the visible faces, we can use a solid color for the faces and still got a correct drawing of the borders.



```
1  \PstCube[fillstyle=solid,fillcolor=red]{1}{3}{2}
```



```
1  \PstCube[linestyle=dotted,fillstyle=solid,fillcolor=white]
2       {1}{3}{2}
```

The unit parameter can still be used and has for effect to multiply each of the three dimensions by it. In fact the value for xunit is used both for the X and Y directions, and the yunit one for the Z direction.

```
1  \rput(0,0){\PstCube[fillstyle=solid,fillcolor=white]{2}{2}{2}}
2  \rput(4,0){\PstCube[unit=2,fillstyle=solid,fillcolor=white]{1}{1}{1}}
```
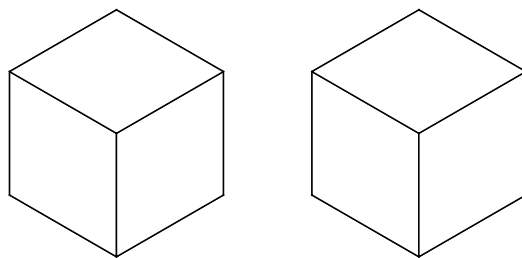


```
1  \rput(0,0){\PstCube[fillstyle=solid,fillcolor=white]{2}{2}{1}}
2  \rput(4,0){\PstCube[xunit=2,fillstyle=solid,fillcolor=white]{1}{1}{1}}
```
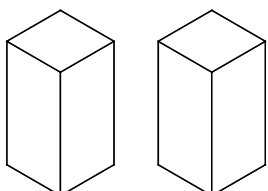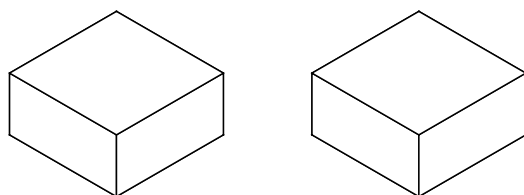
```
1  \PstCube[fillstyle=solid,fillcolor=white]{1}{1}{2}
2  \rput(2,0){%
3    \PstCube[yunit=2,fillstyle=solid,fillcolor=white]
4            {1}{1}{1}}
```
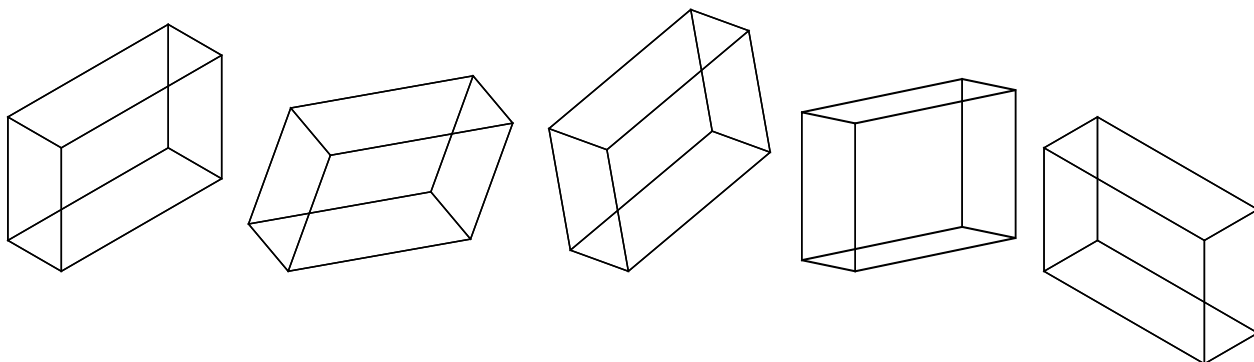
We can of course use the `viewpoint` and `viewangle` parameters of the 'pst-3d' package to change the user point of vue.

```
1  \rput(0,0){\PstCube{1}{3}{2}}
2  \rput(3,0){\PstCube[viewangle=20]{1}{3}{2}}
3  \rput(7.5,0){\PstCube[viewangle=-10]{1}{3}{2}}
4  \rput(10.5,0){\PstCube[viewpoint=1 -1 0.3]{1}{3}{2}}
5  \rput(13,0){\PstCube[viewpoint=1 1 1]{1}{3}{2}}
```


2.1.2 Dies

`\PstDie` has no required parameters, as it must be a cube, so with the same lengths in each direction. The `unit` parameter is enough to change it size. Of course, only the visibles faces, according to the user point of vue, are shown.

```
\PstDie[optional parameters]
```
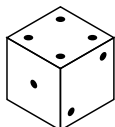


```
1  \PstDie
```

```
1 \PstDie[unit=1.5,dotscale=3,viewangle=45]
```
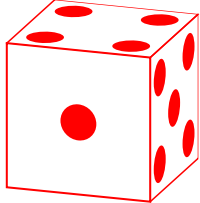


```
1 \PstDie[unit=2,dotsize=0.25,linecolor=red,viewpoint=1 3 -1]
```
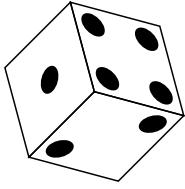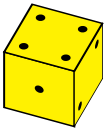


```
1 \PstDie[fillstyle=solid,fillcolor=yellow,
2         viewpoint=0.4 -1 1]
```
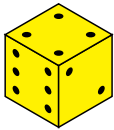


```
1 \PstDie[fillstyle=solid,fillcolor=yellow,
2         viewpoint=-1 -1 -1]
```
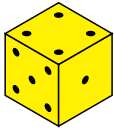


```
1 \PstDie[fillstyle=solid,fillcolor=yellow,viewpoint=-1 -1 1]
```



```
1 \PstDie[fillstyle=solid,fillcolor=yellow,viewpoint=-1 1 1]
```
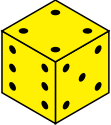
```
1 \psset{fillstyle=solid,fillcolor=yellow}
2 \multido{\iPos=0+2,\nViewPointZ=-2.1+0.5}{8}{%
3   \rput(\iPos,0){\PstDie[viewpoint=1 -1 \nViewPointZ]}}
```

5
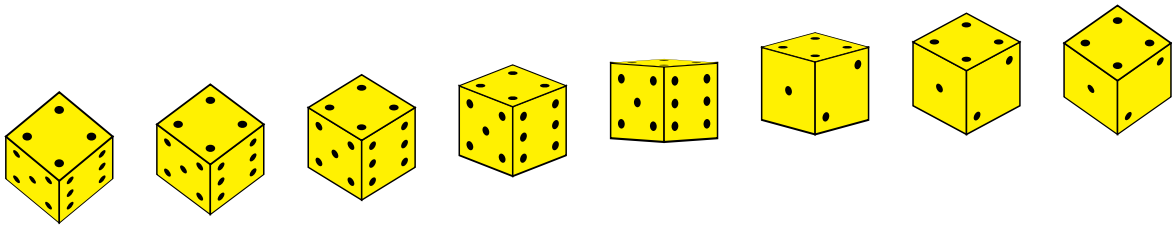
```
1  \psset{fillstyle=solid,fillcolor=yellow}
2  \rput(0,0){\PstDie[viewpoint=-5 -1 1]}
3  \rput(2,0){\PstDie[viewpoint=-3 -1 1]}
4  \rput(4,0){\PstDie[viewpoint=-1 -1 1]}
5  \rput(6,0){\PstDie[viewpoint=-0.3 -1 1]}
6  \rput(8,0){\PstDie[viewpoint=0.3 -1 1]}
7  \rput(10,0){\PstDie[viewpoint=1 -1 1]}
8  \rput(12,0){\PstDie[viewpoint=3 -1 1]}
9  \rput(14,0){\PstDie[viewpoint=5 -1 1]}
```
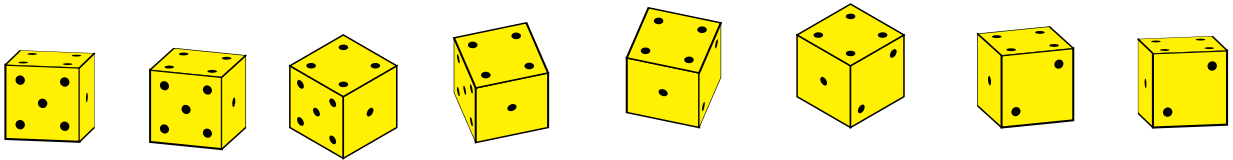


```
1  \psset{fillstyle=solid,fillcolor=yellow}
2  \rput(0,0){\PstDie[viewpoint=1 -5 1]}
3  \rput(2,0){\PstDie[viewpoint=1 -3 1]}
4  \rput(4,0){\PstDie[viewpoint=1 -1 1]}
5  \rput(6,0){\PstDie[viewpoint=1 -0.3 1]}
6  \rput(8,0){\PstDie[viewpoint=1 0.3 1]}
7  \rput(10,0){\PstDie[viewpoint=1 1 1]}
8  \rput(12,0){\PstDie[viewpoint=1 3 1]}
9  \rput(14,0){\PstDie[viewpoint=1 5 1]}
```
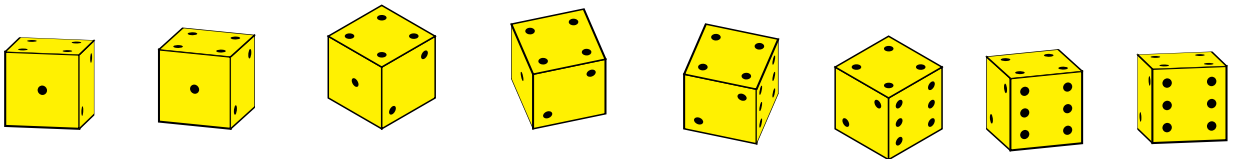
2.2 Parameters

PstDebug (integer) : to obtain some internal debugging informations — here, a letter is printed at the center of each face, to help to locate the various faces according to the user point of vue. It can take the values 0 (no debug) or 1. (*Default: 0 — no debugging informations*).
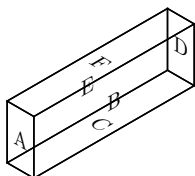
```
1 \PstCube[PstDebug=1]{0.5}{3}{1}
```
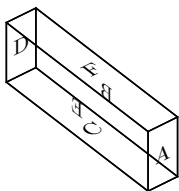
```
1 \PstCube[PstDebug=1,viewpoint=-0.8 -1 1]{0.5}{3}{1}
```
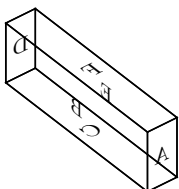
```
1 \PstCube[PstDebug=1,viewpoint=-0.8 1 -1]{0.5}{3}{1}
```

OnlyVisibleFaces (boolean) : to draw only the (three) visible faces, according to the user point of view (*Default: false*).
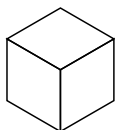
```
1 \PstCube[OnlyVisibleFaces=true]{1}{1}{1}
```
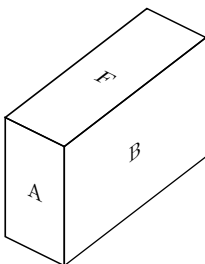
```
1 \PstCube[PstDebug=1,OnlyVisibleFaces=true,
2         viewpoint=0.8 -1 1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=-0.8 -1 1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=0.8 1 1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=-0.8 -1 1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=0.8 1 -1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=-0.8 1 -1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=0.8 -1 -1]{1}{3}{2}
```

```
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
        viewpoint=-0.8 -1 -1]{1}{3}{2}
```

Corners (boolean) : to mark corners. This is mostly an aesthetics effect, in fact mainly pleasant for dies[1]. (*Default: false*).
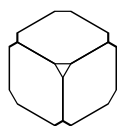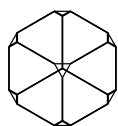
```
\PstCube[Corners=true]{1}{1}{1}
\rput(2,0){%
  \PstCube[OnlyVisibleFaces=true,Corners=true]{1}{1}{1}}
```
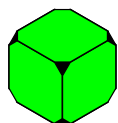
```
\PstCube[fillstyle=solid,fillcolor=green,Corners=true]
        {1}{1}{1}
```
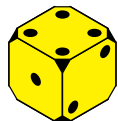
```
\PstDie[dotscale=1.5,fillstyle=solid,fillcolor=yellow,
        Corners=true]
```

CornersLength (real) : the length of the corners, when they are shown. This value is multiplied by the unit values in each direction. (*Default: 0.15* — it must rather be a number between 0 and 0.5, otherwise there will not be any error but the results will look strange...).
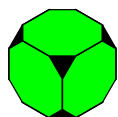
```
\PstCube[fillstyle=solid,fillcolor=green,
        Corners=true,CornersLength=0.25]{1}{1}{1}
```

---

[1]I only follow here an idea first implemented by Manuel LUQUE in his own PSTricks macro for dies.
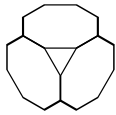
```
1  \PstCube[OnlyVisibleFaces=true,Corners=true,
2          CornersLength=0.3]{1}{1}{1}
```
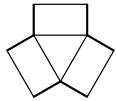


```
1  \PstCube[OnlyVisibleFaces=true,Corners=true,
2          CornersLength=0.5]{1}{1}{1}
```

CornersColor (color) : the color of the corners (*Default: black*).
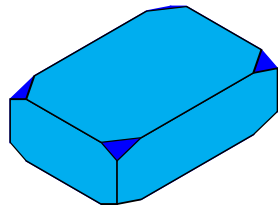

```
1  \PstCube[fillstyle=solid,fillcolor=cyan,Corners=true,
2          CornersColor=blue]{2}{3}{1}
```
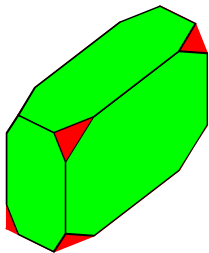


```
1  \PstCube[viewpoint=-0.8 -1 -1,
2          Corners=true,CornersColor=red,CornersLength=0.2,
3          fillstyle=solid,fillcolor=green]{1}{3}{2}
```
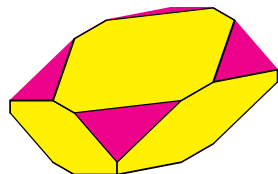


```
1  \PstCube[fillstyle=solid,fillcolor=yellow,
2          Corners=true,CornersColor=magenta,
3          CornersLength=0.4]{2}{3}{1}
```

```
1  \psset{fillstyle=solid,fillcolor=yellow,Corners=true,CornersColor=red}
2  \multido{\iPos=0+2,\nCornersLength=0+0.1}{6}{%
3    \rput(\iPos,0){\PstCube[CornersLength=\nCornersLength]{1}{1}{1}}}
```
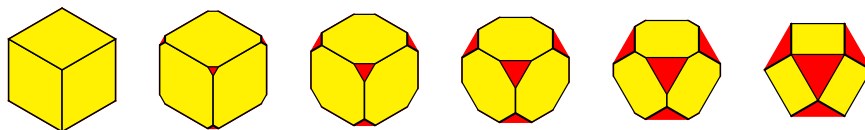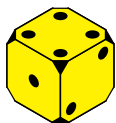
```
1 \PstDie[dotscale=1.5,fillstyle=solid,fillcolor=yellow,
2        Corners=true]
```
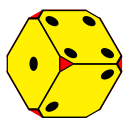
```
1 \PstDie[dotscale=1.5,fillstyle=solid,fillcolor=yellow,
2        viewangle=30,Corners=true,CornersColor=red]
```
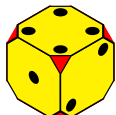
```
1 \PstDie[dotscale=1.5,fillstyle=solid,fillcolor=yellow,
2        Corners=true,CornersColor=red,CornersLength=0.2]
```

RandomFaces (boolean) : to generate random faces. This has only a meaning for dies, to simulate a throw of them. The hazard is managed by the 'random' package from Donald ARSENEAU. The random seed is set by using the time when the compilation occur. But take care that TEX allow to access only to minutes and not to microseconds, so several consecutive usages of this parameter will give the same behavior, and no hazard at all... To force it, we must use the \randomi macro, to initialize the random seed to an arbitrary integer value (see the documentation of the 'random' package). (*Default: false*).

```
1 \PstDie[dotscale=2,fillstyle=solid,fillcolor=cyan,
2        RandomFaces=true]
```
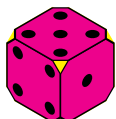
```
1 \randomi=12345
2 \PstDie[dotscale=1.5,fillstyle=solid,fillcolor=magenta,
3        Corners=true,CornersColor=yellow,RandomFaces=true]
```

2.3   Hooks

Twelve hooks can be used, to put arbitrary stuff on the faces. Six hooks allow to put some material on a specified position (by default in the point (0,0) of the face) and six other put it on the center of the faces, according of the lengths in each direction.

The name of these hooks are \PstObjectsThreeDFace*Letter* and \PstObjectsThreeDFaceCenter*Letter* (with *Letter* being a letter between **A** and **F**).
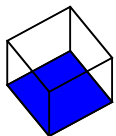
```
1 \def\PstObjectsThreeDFaceC{%
2   \psframe[fillstyle=solid,fillcolor=blue](1,1)}
3 \PstCube[viewpoint=0.6 -0.4 1]{1}{1}{1}
```
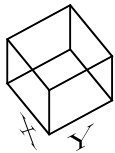
```
1 \def\PstObjectsThreeDFaceA{\rput(0.5,-0.5){\LARGE X}}
2 \def\PstObjectsThreeDFaceB{\rput(0.5,-0.5){\LARGE Y}}
3 \PstCube[viewpoint=0.6 -0.4 1]{1}{1}{1}
```

```
1 \psset{viewpoint=0.6 -0.4 1}
2 \def\PstObjectsThreeDFaceA{\LARGE X}
3 \def\PstObjectsThreeDFaceB{\rput(0.5,0.5){\LARGE Y}}
4 \rput(0,0){\PstCube{1}{1}{1}}
5 \def\PstObjectsThreeDFaceCenterC{\LARGE\textcolor{red}{\textbf{1}}}
6 \rput(3,0){\PstCube{3}{1}{1}}
7 \rput(6,0){\PstCube[viewpoint=0.8 1 1]{3}{1}{1}}
```
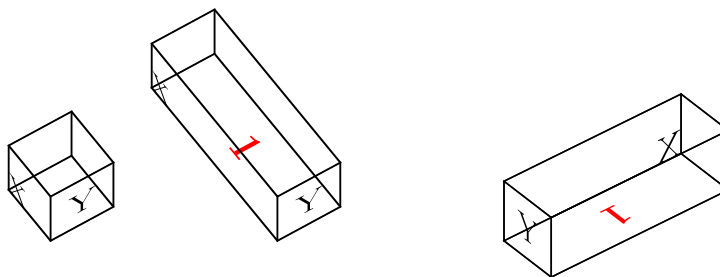
3   Examples

We give here some more advanced examples.
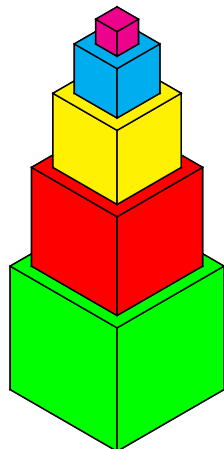
```
1 \psset{fillstyle=solid}
2 \PstCube[fillcolor=green]{2}{2}{2}
3 \PstCube[fillcolor=red](-0.2,0.2,2){1.6}{1.6}{1.6}
4 \PstCube[fillcolor=yellow](-0.4,0.4,3.6){1.2}{1.2}{1.2}
5 \PstCube[fillcolor=cyan](-0.6,0.6,4.8){0.8}{0.8}{0.8}
6 \PstCube[fillcolor=magenta](-0.8,0.8,5.6){0.4}{0.4}{0.4}
```
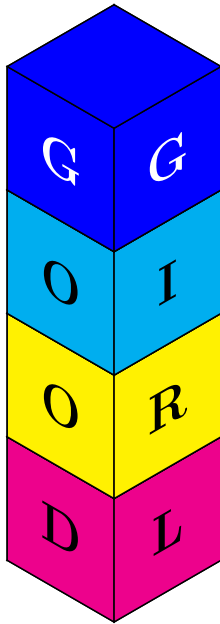
```
1  \psset{unit=2,fillstyle=solid}
2  \def\PstObjectsThreeDFaceCenterA{\Huge\textbf{D}}%
3  \def\PstObjectsThreeDFaceCenterB{\Huge\textbf{L}}%
4  \PstCube[fillcolor=magenta]{1}{1}{1}
5  \def\PstObjectsThreeDFaceCenterA{\Huge\textbf{O}}%
6  \def\PstObjectsThreeDFaceCenterB{\Huge\textbf{R}}%
7  \PstCube[fillcolor=yellow](0,0,1){1}{1}{1}
8  \def\PstObjectsThreeDFaceCenterA{\Huge\textbf{O}}%
9  \def\PstObjectsThreeDFaceCenterB{\Huge\textbf{I}}%
10 \PstCube[fillcolor=cyan](0,0,2){1}{1}{1}
11 \def\PstObjectsThreeDFaceCenterA{%
12   \Huge\textcolor{white}{\textbf{G}}}%
13 \let\PstObjectsThreeDFaceCenterB
14      \PstObjectsThreeDFaceCenterA
15 \PstCube[fillcolor=blue](0,0,3){1}{1}{1}
```

```
1  \definecolor{PaleGreen}{rgb}{0.6,0.98,0.6}
2  \psset{unit=2}
3  \ThreeDput{%
4    \psframe[fillstyle=solid,fillcolor=PaleGreen](6,6)
5    \psgrid[subgriddiv=0,gridlabels=0,griddots=5](6,6)}
6  \psset{fillstyle=solid,dotscale=3.5,RandomFaces=true,Corners=true}
7  \randomi=123817
8  \PstDie[fillcolor=yellow](1,3,0)
9  \randomi=271354
10 \PstDie[fillcolor=cyan,viewpoint=1 0.3 1,CornersColor=blue](0.3,1.5,0)
11 \randomi=93850516
12 \PstDie[fillcolor=blue,linecolor=white,viewpoint=1 -0.5 1,CornersColor=cyan]
13        (3,3,0)
14 \randomi=8873165
15 \PstDie[fillcolor=red,linecolor=white,viewpoint=1 -0.2 1,CornersColor=yellow]
16        (2,5,0)
```
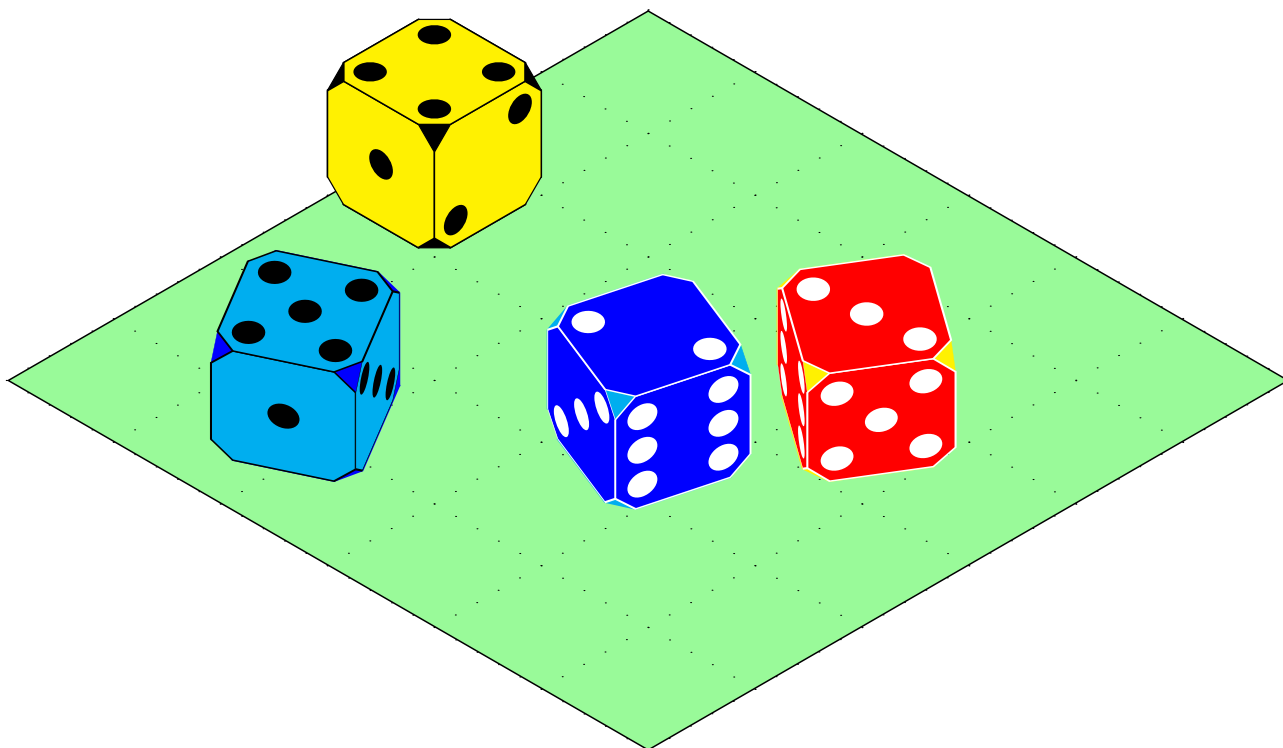
4 Driver file

The next bit of code contains the documentation driver file for TEX, i.e., the file that will produce the documentation you are currently reading. It will be extracted from this file by the `docstrip` program.

```
 1 ⟨∗driver⟩
 2 \documentclass{ltxdoc}
 3 \GetFileInfo{pst-ob3d.dtx}
 4 \usepackage{fancyvrb,url}
 5 \usepackage[dvips]{geometry}
 6 \usepackage{pstricks}
 7 \usepackage{pst-ob3d}
 8 \usepackage{multido}
 9 \EnableCrossrefs
10 \CodelineIndex
11 \RecordChanges
12 %\OnlyDescription              % Comment it for implementation details
13 \hbadness=7000                 % Over and under full box warnings
14 \hfuzz=3pt
15 %
16 % To redefine the format used to print the macro names,
```

```
17 % which was not well adapted to very long names like the ones we used
18 \makeatletter
19 \long\def\m@cro@#1#2{\endgroup \topsep\MacroTopsep \trivlist
20   \edef\saved@macroname{\string#2}%
21 % D.G. modification begin - Dec. 15, 2000
22 %  \def\makelabel##1{\llap{##1}}%
23   \def\makelabel##1{\hspace{-2cm}##1}%
24 % D.G. modification end
25   \if@inlabel
26     \let\@tempa\@empty \count@\macro@cnt
27     \loop \ifnum\count@>\z@
28       \edef\@tempa{\@tempa\hbox{\strut}}\advance\count@\m@ne \repeat
29     \edef\makelabel##1{\llap{\vtop to\baselineskip
30                              {\@tempa\hbox{##1}\vss}}}%
31     \advance \macro@cnt \@ne
32   \else  \macro@cnt\@ne  \fi
33   \edef\@tempa{\noexpand\item[%
34     #1%
35       \noexpand\PrintMacroName
36     \else
37       \noexpand\PrintEnvName
38     \fi
39     {\string#2}]}%
40   \@tempa
41   \global\advance\c@CodelineNo\@ne
42   #1%
43     \SpecialMainIndex{#2}\nobreak
44     \DoNotIndex{#2}%
45   \else
46     \SpecialMainEnvIndex{#2}\nobreak
47   \fi
48   \global\advance\c@CodelineNo\m@ne
49   \ignorespaces}
50 \makeatother
51 %
52 \begin{document}
53   \DocInput{pst-ob3d.dtx}
54 \end{document}
55 ⟨/driver⟩
```

5 'pst-ob3d' LATEX wrapper

```
56 ⟨∗latex − wrapper⟩
57 \RequirePackage{pstricks}
58 \ProvidesPackage{pst-ob3d}[2006/11/25 package wrapper for
59   pst-ob3d.tex (hv)]
60 \input{pst-ob3d.tex}
61 \ProvidesFile{pst-ob3d.tex}
62   [\filedate\space v\fileversion\space 'PST-ob3d' (dg,hv)]
```

63 ⟨/latex − wrapper⟩

6 'pst-ob3d' code

64 ⟨∗pst − ob3d⟩

6.1 Preambule

Who we are.

```
65 \def\fileversion{0.21}
66 \def\filedate{2007/08/22}
67 \message{'Pst-Objects3d' v\fileversion, \filedate\space (DG,hv)}
68 \csname PstObjectsThreeDLoaded\endcsname
69 \let\PstObjectsThreeDLoaded\endinput
```

Require the PSTricks, 'pst-3d', and 'pst-xkey' packages.

```
70 \ifx\PSTricksLoaded\endinput\else\input pstricks.tex\fi
71 \ifx\PSTthreeDLoaded\endinput\else\input pst-3d.tex\fi
72 \ifx\PSTXKeyLoaded\endinput\else\input pst-xkey \fi
```

Donald ARSENEAU package for random number generation.

```
73 \input random.tex
```

Catcodes changes.

```
74 \edef\PstAtCode{\the\catcode'\@}
75 \catcode'\@=11\relax
```

6.2 Definition of the parameters

Add the family name to the key list.

```
76 \pst@addfams{pst-ob3d}
```

PstDebug is for internal debugging purposes (here letters will be printed on the faces). This option is alread defined in the basic PSTricks package

OnlyVisibleFaces to show only the visible faces.

```
77 \define@boolkey[psset]{pst-ob3d}[PstObjectsThreeD@]{OnlyVisibleFaces}[true]{}
```

RandomFaces to define randomly the faces of a die, using the 'random' package.

```
78 \define@boolkey[psset]{pst-ob3d}[PstObjectsThreeD@]{RandomFaces}[true]{}
```

Corners to draw corners.

```
79 \define@boolkey[psset]{pst-ob3d}[PstObjectsThreeD@]{Corners}[true]{}
```

CornersColor to choose the color of the corners.

```
80 \define@key[psset]{pst-ob3d}{CornersColor}{\def\PstObjectsThreeD@CornersColor{#1}}
```

CornersLength to choose the length of the corners.

```
81 \define@key[psset]{pst-ob3d}{CornersLength}{%
82   \def\PstObjectsThreeD@CornersLength{#1}%
83   \def\PstObjectsThreeD@CornersLengthTmpA{#1}%
84   \def\PstObjectsThreeD@CornersLengthTmpB{#1}}
```

16

Next, we set the default values for all these new parameters.

```
85 \psset[pst-ob3d]{OnlyVisibleFaces=false,RandomFaces=false,
86                   Corners=false,CornersColor=black,CornersLength=0.15}
```

6.3   Cube definition

\PstCube

```
87 \def\PstCube{\@ifnextchar[{\PstCube@i}{\PstCube@i[]}}
```

\PstCube@i

```
88 \def\PstCube@i[#1]{\@ifnextchar({\PstCube@ii[#1]}{\PstCube@ii[#1](0,0,0)}}
```

\PstCube@ii

```
89 \def\PstCube@ii[#1](#2,#3,#4)#5#6#7{{%
```

We force "`dimen=middle`".

```
90 \psset{dimen=middle}%
```

Then we set the local changes of parameters.

```
91 \psset{#1}%
```

For debugging purposes, we can require to print letters centered on the faces.

```
92 \ifnum\Pst@Debug=\@ne
93   \def\PstObjectsThreeDFaceCenterA{A}%
94   \def\PstObjectsThreeDFaceCenterB{B}%
95   \def\PstObjectsThreeDFaceCenterC{C}%
96   \def\PstObjectsThreeDFaceCenterD{D}%
97   \def\PstObjectsThreeDFaceCenterE{E}%
98   \def\PstObjectsThreeDFaceCenterF{F}%
99 \fi
```

We get the signs of the viewpoint coordinates (which are wrong by themselves). This is necessary because the order of the drawing of the faces must change according to the viewpoint, to hidden the non visible faces.

```
100 \pst@expandafter\psget@@viewpoint\psk@viewpoint {} {} {} \@nil
```

If corners must be shown, we will draw octogons, otherwise frames.

```
101 \ifPstObjectsThreeD@Corners
102   \let\PstObjectsThreeD@Shape\PstObjectsThreeD@Octogon
103 \else
104   \let\PstObjectsThreeD@Shape\psframe
105 \fi
```

According to the viewpoint, we draw in the right order the faces (three ones when we must show only the visible ones, six otherwise).

```
106 \ifdim\pst@dimc>\z@
107   \ifdim\pst@dima>\z@
108     \ifdim\pst@dimb>\z@
```

Case where $x > 0$; $y > 0$; $z > 0$

```
109        \PstCube@DrawFaces{A}{E}{C}{B}{D}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
110      \else
```

Case where $x > 0$; $y < 0$; $z > 0$

```
111        \PstCube@DrawFaces{E}{D}{C}{A}{B}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
112      \fi
113    \else
114      \ifdim\pst@dimb>\z@
```

Case where $x < 0$; $y > 0$; $z > 0$

```
115        \PstCube@DrawFaces{A}{B}{C}{E}{D}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
116      \else
```

Case where $x < 0$; $y < 0$; $z > 0$

```
117        \PstCube@DrawFaces{D}{B}{C}{E}{A}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
118      \fi
119    \fi
120  \else
121    \ifdim\pst@dima>\z@
122      \ifdim\pst@dimb>\z@
```

Case where $x > 0$; $y > 0$; $z < 0$

```
123        \PstCube@DrawFaces{B}{D}{C}{A}{E}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
124      \else
```

Case where $x > 0$; $y < 0$; $z < 0$

```
125        \PstCube@DrawFaces{A}{B}{C}{E}{D}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
126      \fi
127    \else
128      \ifdim\pst@dimb>\z@
```

Case where $x < 0$; $y > 0$; $z < 0$

```
129        \PstCube@DrawFaces{D}{E}{C}{A}{B}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
130      \else
```

Case where $x < 0$; $y > 0$; $z < 0$

```
131        \PstCube@DrawFaces{E}{A}{C}{D}{B}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
132      \fi
133    \fi
134  \fi}}
```

6.4 Die definition

135 \def\PstDie{\@ifnextchar[{\PstDie@i}{\PstDie@i[]}}

136 \def\PstDie@i[#1]{\@ifnextchar({\PstDie@ii[#1]}{\PstDie@ii[#1](0,0,0)}}

137 \def\PstDie@ii[#1](#2,#3,#4){{%

We force "`dimen=middle`".

138 \psset{dimen=middle}%

We set the local changes of parameters.

139 \psset{#1}%

A die is only a cube where the six faces have dots between 1 and 6.

140 \ifPstObjectsThreeD@RandomFaces

First the case where the faces are defined randomly.

```
141    \setrannum{\pst@cnth}{1}{6}%
142    \PstDie@Letter{\pst@cnth}{\@tempa}%
143    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
144      \psdots(.5,.5)}%
145    \PstDie@Letter{\pst@cnth}{\@tempa}%
146    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
147      \psdots(.2,.2)(.8,.8)}%
148    \PstDie@Letter{\pst@cnth}{\@tempa}%
149    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
150      \psdots(.2,.2)(.5,.5)(.8,.8)}%
151    \PstDie@Letter{\pst@cnth}{\@tempa}%
152    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
153      \psdots(.2,.2)(.2,.5)(.2,.8)(.8,.2)(.8,.5)(.8,.8)}%
154    \PstDie@Letter{\pst@cnth}{\@tempa}%
155    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
156      \psdots(.2,.2)(.2,.8)(.5,.5)(.8,.2)(.8,.8)}%
157    \PstDie@Letter{\pst@cnth}{\@tempa}%
158    \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
159      \psdots(.2,.2)(.2,.8)(.8,.2)(.8,.8)}%
```

160 \else

Then the case where the faces are defined fixedly (the face "4" will be above).

```
161    \def\PstObjectsThreeDFaceA{\psdots(.5,.5)}%
162    \def\PstObjectsThreeDFaceB{\psdots(.2,.2)(.8,.8)}%
163    \def\PstObjectsThreeDFaceC{\psdots(.2,.2)(.5,.5)(.8,.8)}%
164    \def\PstObjectsThreeDFaceD{%
165      \psdots(.2,.2)(.2,.5)(.2,.8)(.8,.2)(.8,.5)(.8,.8)}%
166    \def\PstObjectsThreeDFaceE{\psdots(.2,.2)(.2,.8)(.5,.5)(.8,.2)(.8,.8)}%
167    \def\PstObjectsThreeDFaceF{\psdots(.2,.2)(.2,.8)(.8,.2)(.8,.8)}%
```

19

```
168 \fi
```

Now that the faces are defined, we can draw the cube, with only the visible faces if corners are not to be drawn.

```
169 \ifPstObjectsThreeD@Corners
170 \else
171   \PstObjectsThreeD@OnlyVisibleFacestrue
172 \fi
173 \PstCube(#2,#3,#4){1}{1}{1}}}
```

6.5   Internal functions to draw the faces

To draw successively the six faces in the right order.

\PstCube@DrawFaces

```
174 \def\PstCube@DrawFaces#1#2#3#4#5#6#7#8{%
175 \ifPstObjectsThreeD@OnlyVisibleFaces
176 \else
177   \bgroup
178     \ifPstObjectsThreeD@Corners
179       \psset{linecolor=\PstObjectsThreeD@CornersColor,
180               fillcolor=\PstObjectsThreeD@CornersColor}%
181     \fi
182     \bgroup
183       \ifPstObjectsThreeD@Corners
184         \def\PstObjectsThreeD@CornersLengthTmpA{0}%
185       \fi
186       \expandafter\csname PstCube@DrawFace#1\endcsname#7#8
187       \expandafter\csname PstCube@DrawFace#2\endcsname#7#8
188     \egroup
189     \bgroup
190       \ifPstObjectsThreeD@Corners
191         \def\PstObjectsThreeD@CornersLengthTmpB{0}%
192       \fi
193       \expandafter\csname PstCube@DrawFace#3\endcsname#7#8
194     \egroup
195   \egroup
196 \fi
197 \expandafter\csname PstCube@DrawFace#4\endcsname#7#8
198 \expandafter\csname PstCube@DrawFace#5\endcsname#7#8
199 \expandafter\csname PstCube@DrawFace#6\endcsname#7#8}
```

To draw the face A.

\PstCube@DrawFaceA

```
200 \def\PstCube@DrawFaceA(#1,#2,#3)(#4,#5,#6){%
201 \pst@dima=-#4\psxunit
202 \divide\pst@dima\tw@
203 \pst@dimc=#6\psyunit
204 \divide\pst@dimc\tw@
```

```
205 \ThreeDput[normal=0 -1 0](#1,#2,#3){%
206   \PstObjectsThreeD@Shape(-#4,#6)
207   \ifx\PstObjectsThreeDFaceA\@undefined
208   \else
209     \rput[bl](-#4,0){\PstObjectsThreeDFaceA}
210   \fi
211   \ifx\PstObjectsThreeDFaceCenterA\@undefined
212   \else
213     \rput[c](\pst@dima,\pst@dimc){\PstObjectsThreeDFaceCenterA}
214   \fi}}
```

To draw the face B.

\PstCube@DrawFaceB

```
215 \def\PstCube@DrawFaceB(#1,#2,#3)(#4,#5,#6){%
216 \pst@dimb=#5\psxunit
217 \divide\pst@dimb\tw@
218 \pst@dimc=#6\psyunit
219 \divide\pst@dimc\tw@
220 \ThreeDput[normal=1 0 0](#1,#2,#3){%
221   \PstObjectsThreeD@Shape(#5,#6)
222   \ifx\PstObjectsThreeDFaceB\@undefined
223   \else
224     \rput[bl](0,0){\PstObjectsThreeDFaceB}
225   \fi
226   \ifx\PstObjectsThreeDFaceCenterB\@undefined
227   \else
228     \rput[c](\pst@dimb,\pst@dimc){\PstObjectsThreeDFaceCenterB}
229   \fi}}
```

To draw the face C.

\PstCube@DrawFaceC

```
230 \def\PstCube@DrawFaceC(#1,#2,#3)(#4,#5,#6){%
231 \pst@dimg=-#4\psxunit
232 \advance \pst@dimg by #1\psxunit
233 \pst@dima=#4\psxunit
234 \divide\pst@dima\tw@
235 \pst@dimb=#5\psxunit
236 \divide\pst@dimb\tw@
237 \ThreeDput[normal=0 0 1](\pst@dimg,#2,#3){{%
238   \psyunit=\psxunit
239   \PstObjectsThreeD@Shape(#4,#5)
240   \ifx\PstObjectsThreeDFaceC\@undefined
241   \else
242     \rput[bl](0,0){\PstObjectsThreeDFaceC}
243   \fi
244   \ifx\PstObjectsThreeDFaceCenterC\@undefined
245   \else
246     \rput[c](\pst@dima,\pst@dimb){\PstObjectsThreeDFaceCenterC}
247   \fi}}}
```

To draw the face D.

```
248 \def\PstCube@DrawFaceD(#1,#2,#3)(#4,#5,#6){%
249 \pst@dimg=#5\psxunit
250 \advance \pst@dimg by #2\psyunit
251 \pst@dima=-#4\psxunit
252 \divide\pst@dima\tw@
253 \pst@dimc=#6\psyunit
254 \divide\pst@dimc\tw@
255 \ThreeDput[normal=0 -1 0](#1,\pst@dimg,#3){%
256   \PstObjectsThreeD@Shape(-#4,#6)
257   \ifx\PstObjectsThreeDFaceD\@undefined
258   \else
259     \rput[bl](-#4,0){\PstObjectsThreeDFaceD}
260   \fi
261   \ifx\PstObjectsThreeDFaceCenterD\@undefined
262   \else
263     \rput[c](\pst@dima,\pst@dimc){\PstObjectsThreeDFaceCenterD}
264   \fi}}
```

To draw the face E.

```
265 \def\PstCube@DrawFaceE(#1,#2,#3)(#4,#5,#6){%
266 \pst@dimg=-#4\psxunit
267 \advance \pst@dimg by #1\psxunit
268 \pst@dimb=#5\psxunit
269 \divide\pst@dimb\tw@
270 \pst@dimc=#6\psyunit
271 \divide\pst@dimc\tw@
272 \ThreeDput[normal=1 0 0](\pst@dimg,#2,#3){%
273   \PstObjectsThreeD@Shape(#5,#6)
274   \ifx\PstObjectsThreeDFaceE\@undefined
275   \else
276     \rput[bl](0,0){\PstObjectsThreeDFaceE}
277   \fi
278   \ifx\PstObjectsThreeDFaceCenterE\@undefined
279   \else
280     \rput[c](\pst@dimb,\pst@dimc){\PstObjectsThreeDFaceCenterE}
281   \fi}}
```

To draw the face F.

```
282 \def\PstCube@DrawFaceF(#1,#2,#3)(#4,#5,#6){%
283 \pst@dimg=-#4\psxunit
284 \advance \pst@dimg by #1\psxunit
285 \pst@dimh=#6\psyunit
286 \advance \pst@dimh by #3\psyunit
```

```
287 \pst@dima=#4\psxunit
288 \divide\pst@dima\tw@
289 \pst@dimb=#5\psxunit
290 \divide\pst@dimb\tw@
291 \ThreeDput[normal=0 0 1](\pst@dimg,#2,\pst@dimh){{%
292   \psyunit=\psxunit
293   \PstObjectsThreeD@Shape(#4,#5)
294   \ifx\PstObjectsThreeDFaceF\@undefined
295   \else
296     \rput[bl](0,0){\PstObjectsThreeDFaceF}
297   \fi
298   \ifx\PstObjectsThreeDFaceCenterF\@undefined
299   \else
300     \rput[c](\pst@dima,\pst@dimb){\PstObjectsThreeDFaceCenterF}
301   \fi}}}
```

6.6   Internal functions

The `\psget@@viewpoint` macro is derived from the `\psset@@viewpoint` one (the values will not be the good ones, but in fact we are only interested by the signs of them...)

`\psget@@viewpoint`

```
302 \def\psget@@viewpoint#1 #2 #3 #4\@nil{%
303 \pssetxlength{\pst@dima}{#1}%
304 \pssetylength{\pst@dimb}{#2}%
305 \pssetzlength{\pst@dimc}{#3}}
```

Macro to draw an octogon. This is useful to put it on each face, to let corners appear on a cube in another color. This is specially suited for dies.

`\PstObjectsThreeD@Octogon`

```
306 \def\PstObjectsThreeD@Octogon(#1,#2){%
307 \pst@dimh=\PstObjectsThreeD@CornersLength pt
308 \multiply\pst@dimh#1
309 \pst@dimtonum{\pst@dimh}{\pst@tempa}%
310 \pst@dimh=\PstObjectsThreeD@CornersLengthTmpB pt
311 \multiply\pst@dimh#1
312 \pst@dimtonum{\pst@dimh}{\pst@tempg}%
313 \pst@dimh=\@ne pt
314 \advance\pst@dimh-\PstObjectsThreeD@CornersLength pt
315 \multiply\pst@dimh#1
316 \pst@dimtonum{\pst@dimh}{\pst@tempb}%
317 \pst@dimh=\@ne pt
318 \advance\pst@dimh-\PstObjectsThreeD@CornersLengthTmpA pt
319 \multiply\pst@dimh#1
320 \pst@dimtonum{\pst@dimh}{\pst@tempe}%
321 \pst@dimh=\PstObjectsThreeD@CornersLength pt
322 \multiply\pst@dimh#2
323 \pst@dimtonum{\pst@dimh}{\pst@tempc}%
324 \pst@dimh=\PstObjectsThreeD@CornersLengthTmpA pt
```

```
325 \multiply\pst@dimh#2
326 \pst@dimtonum{\pst@dimh}{\pst@tempf}%
327 \pst@dimh=\@ne pt
328 \advance\pst@dimh-\PstObjectsThreeD@CornersLength pt
329 \multiply\pst@dimh#2
330 \pst@dimtonum{\pst@dimh}{\pst@tempd}%
331 \pst@dimh=\@ne pt
332 \advance\pst@dimh-\PstObjectsThreeD@CornersLengthTmpB pt
333 \multiply\pst@dimh#2
334 \pst@dimtonum{\pst@dimh}{\pst@temph}%
335 \pspolygon(\pst@tempa,0)(\pst@tempe,0)(#1,\pst@tempf)(#1,\pst@tempd)
336         (\pst@tempb,#2)(\pst@tempg,#2)(0,\pst@temph)(0,\pst@tempc)}
```

Macro to increment the counter (`#1`) and define (`#2`) the corresponding letter between A (for 1) and F (for 6).

`\PstDie@Letter`

```
337 \def\PstDie@Letter#1#2{%
338 \advance#1\@ne
339 \ifnum#1>6
340   #1=\@ne
341 \fi
342 \def#2{\ifcase#1\or A\or B\or C\or D\or E\or F\fi}}
```

6.7   Closing

Catcodes restoration.

```
343 \catcode'\@=\PstAtCode\relax
```

```
344 ⟨/pst − ob3d⟩
```

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.

Change History