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 parallelepiped ones) and dies (which are only a special kind of
cubes) are defined.

Contents

1 Introduction 1

2 Usage 1
  2.1 Macros ................................................................. 1
  2.1.1 Cubes ............................................................. 2
  2.1.2 Dies ............................................................... 4
  2.2 Parameters ......................................................... 6
  2.3 Hooks ............................................................... 11

3 Examples 12

4 Driver file 13

5 ‘pst-ob3d’ \LaTeX{} wrapper 15

6 ‘pst-ob3d’ code 15
  6.1 Preamble .......................................................... 15
  6.2 Definition of the parameters ................................... 15
  6.3 Cube definition .................................................. 16
  6.4 Die definition .................................................... 18

*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

‘\texttt{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 \texttt{PSTricks} parameters can be use and few ones are added specially for these 3d objects.

You must take care that these objects are \textit{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 \textit{viewpoint}). So, we have to compute their sizes and to put these objects in a \texttt{pspicture} environment by ourselves.

2 Usage

2.1 Macros

Two macros are currently defined: \texttt{\textbackslash PstCube} to draw a cube and \texttt{\textbackslash PstDie} to draw a die (which is only a cube with dots on it faces).

2.1.1 Cubes

\texttt{\textbackslash PstCube} has three required parameters, respectively for the X, Y and Z lengths, as we can distord the cubes to parallelipiped ones.

\begin{verbatim}
\begin{verbatim}
\PstCube[optional\ parameters]{X\ length}{Y\ length}{Z\ length}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\PstCube\{1\}{1\}{1}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\PstCube\{2\}{1.5}
\end{verbatim}
\end{verbatim}

\begin{verbatim}
\begin{verbatim}
\PstCube[linestyle=dashed]\{1\}{2\}{1.5}
\end{verbatim}
\end{verbatim}
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 get a correct drawing of the borders.

The \texttt{unit} parameter can still be used and has for effect to multiply each of the three dimensions by it. In fact the value for \texttt{xunit} is used both for the X and Y directions, and the \texttt{yunit} one for the Z direction.
We can of course use the `viewpoint` and `viewangle` parameters of the `pst-3d` package to change the user point of vue.

\begin{verbatim}
\begin{verbatim}
\rput(0,0){\PstCube{1}{3}{2}}
\rput(2,0){\PstCube[viewangle=20]{1}{3}{2}}
\rput(3,0){\PstCube[viewangle=-10]{1}{3}{2}}
\rput(7.5,0){\PstCube[viewpoint=1 -1 0.3]{1}{3}{2}}
\rput(10.5,0){\PstCube[viewpoint=1 1 1]{1}{3}{2}}
\end{verbatim}
\end{verbatim}
\end{verbatim}

\subsection*{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 visible faces, according to the user point of vue, are shown.

\begin{verbatim}
\PstDie[optional parameters]
\end{verbatim}
\begin{verbatim}
\psset{fillstyle=solid,fillcolor=yellow}
\multido{\iPos=0+2,\nViewPointZ=-2.1+0.5}{8}{{\rput(\iPos,0){\PstDie[viewpoint=1 -1 \nViewPointZ]}}}
\end{verbatim}
2.2 Parameters

\textbf{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 view. It can take the values 0 (no debug) or 1. (Default: 0 — no debugging informations).

\begin{itemize}
  \item \texttt{\textbackslash PstCube[PstDebug=1]} \{0.5\} \{3\} \{1\}
  \item \texttt{\textbackslash PstCube[PstDebug=1,viewpoint=-0.8 \ -1 \ 1]} \{0.5\} \{3\} \{1\}
  \item \texttt{\textbackslash PstCube[PstDebug=1,viewpoint=-0.8 \ 1 \ -1]} \{0.5\} \{3\} \{1\}
\end{itemize}

\textbf{OnlyVisibleFaces (boolean)} : to draw only the (three) visible faces, according to the user point of view (Default: false).

\begin{itemize}
  \item \texttt{\textbackslash PstCube[OnlyVisibleFaces=true]} \{1\} \{1\} \{1\}
  \item \texttt{\textbackslash PstCube[PstDebug=1,OnlyVisibleFaces=true, viewpoint=0.8 \ -1 \ 1]} \{1\} \{3\} \{2\}
\end{itemize}
\begin{verbatim}
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
  viewpoint=-0.8 -1 1]{{1}{3}{2}}
\end{verbatim}

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

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

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

\begin{verbatim}
\PstCube[PstDebug=1,OnlyVisibleFaces=true,
  viewpoint=-0.8 1 -1]{{1}{3}{2}}
\end{verbatim}
Corners (boolean) : to mark corners. This is mostly an aesthetics effect, in fact mainly pleasant for dies\textsuperscript{1}. (Default: false).

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...).

\textsuperscript{1}I only follow here an idea first implemented by Manuel Luque in his own PSTricks macro for dies.
CornersColor (color) : the color of the corners (Default: black).

\begin{verbatim}
\psset{fillstyle=solid,fillcolor=yellow,Corners=true,CornersColor=red}
\multido{\iPos=0+2,\nCornersLength=0+0.1}{6}{%
  \rput(\iPos,0){\PstCube[\nCornersLength=\nCornersLength]{1}{1}{1}}}
\end{verbatim}
RandomFaces (boolean) : to generate random faces. This has only a meaning for dice, 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).

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 \PstObjectsThreeDFaceLetter and \PstObjectsThreeDFaceCenterLetter (with Letter being a letter between A and F).
3 Examples

We give here some more advanced examples.
\psset{unit=2,fillstyle=solid}
\def\PstObjectsThreeDFaceCenterA{\Huge\textbf{D}}%
\def\PstObjectsThreeDFaceCenterB{\Huge\textbf{L}}%
\PstCube[fillcolor=magenta]{1}{1}{1}
\def\PstObjectsThreeDFaceCenterA{\Huge\textbf{O}}%
\def\PstObjectsThreeDFaceCenterB{\Huge\textbf{R}}%
\PstCube[fillcolor=yellow](0,0,1){1}{1}{1}
\def\PstObjectsThreeDFaceCenterA{\Huge\textbf{O}}%
\def\PstObjectsThreeDFaceCenterB{\Huge\textbf{I}}%
\PstCube[fillcolor=cyan](0,0,2){1}{1}{1}
\def\PstObjectsThreeDFaceCenterA{\Huge\textbf{G}}%
\let\PstObjectsThreeDFaceCenterB\PstObjectsThreeDFaceCenterA
\PstCube[fillcolor=blue](0,0,3){1}{1}{1}
\definecolor{PaleGreen}{rgb}{0.6,0.98,0.6}
\psset{unit=2}
\ThreeDput{\psframe[fillstyle=solid,fillcolor=PaleGreen](6,6)\psgrid[subgriddiv=0,gridlabels=0,griddots=5](6,6)\psset{fillstyle=solid,dotscale=3.5,RandomFaces=true,Corners=true}\randomi=123817\PstDie[fillcolor=yellow](1,3,0)\randomi=271354\PstDie[fillcolor= cyan,viewpoint=1.0 0.3 1,CornersColor=blue](0.3,1.5,0)\randomi=93850516\PstDie[fillcolor=blue,linecolor=white,viewpoint=1 -0.5 1,CornersColor=cyan](3,3,0)\randomi=8873165\PstDie[fillcolor=red,linecolor=white,viewpoint=1 -0.2 1,CornersColor=yellow](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.

\begin{verbatim}
\documentclass{ltxdoc}
\GetFileInfo{pst-ob3d.dtx}
\usepackage{fancyvrb,url}
\usepackage[dvips]{geometry}
\usepackage{pstricks}
\usepackage{pst-ob3d}
\usepackage{multido}
\EnableCrossrefs
\CodelineIndex
\RecordChanges
\OnlyDescription % Comment it for implementation details
\hbadness=7000 % Over and under full box warnings
\hfuzz=3pt
\%
% To redefine the format used to print the macro names,
\end{verbatim}
% which was not well adapted to very long names like the ones we used
\makeatletter
\long\def\m@cro@#1#2{\endgroup \topsep MacroTopsep \trivlist
\edef\saved@macroname{\string#2}\
% D.G. modification begin - Dec. 15, 2000
% \def\makelabel##1{\llap{##1}}%
\def\makelabel##1{\hspace{-2cm}##1}%
% D.G. modification end
\if@inlabel
\let\@tempa\@empty \count@macro\cnt
\loop \ifnum\count@>\z@
% \edef\@tempa{\@tempa\hbox{\strut}{\advance\count@\m@ne }\repeat
\edef\makelabel##1{\llap{\vtop to\baselineskip
\{\@tempa\hbox{##1}\vss}}}%
\advance \macro\cnt \@ne
\else \macro\cnt \@ne \fi
\edef\@tempa{\noexpand\item[\string#2]}%
\@tempa
\global\advance\c@CodelineNo \@ne
\noexpand\PrintMacroName
\else

\noexpand\PrintEnvName
\fi
\{:\string#2}]}%
\@tempa
\global\advance\c@CodelineNo \@ne
\SpecialMainIndex{\string#2}\nobreak
\DoNotIndex{\string#2}\
% \fi
\SpecialMainEnvIndex{\string#2}\nobreak
\fi
\global\advance\c@CodelineNo \@ne
\ignorespaces}
\makeatother
\begin{document}
\DocInput{pst-ob3d.dtx}
\end{document}
⟨ /driver ⟩

\section{\texttt{pst-ob3d}' \LaTeX{} wrapper}
\begin{verbatim}
text -- \texttt{latex \textminus \ wrapper}
\RequirePackage{pstricks}
\ProvidesPackage{pst-ob3d}[2006/11/25 package wrapper for
\texttt{pst-ob3d}.tex \ (hv)]
\input{pst-ob3d.tex}
\ProvidesFile{pst-ob3d.tex}
[\filedate  \space v\fileversion\space \texttt{PST-ob3d} \ (dg,hv)]
\end{verbatim}
6 ‘pst-ob3d’ code

6.1 Preambule

Who we are.

\def\fileversion{0.21}
\def\filedate{2007/08/22}
\message{Pst-Objects3d \fileversion, \filedate (DG,hv)}
\csname PstObjectsThreeDLoaded\endcsname
\let\PstObjectsThreeDLoaded\endinput

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

Donald ARSENEAU package for random number generation.
\input random.tex

Catcodes changes.
\edef\PstAtCode{\the\catcode'\@}
\catcode'\@=11\relax

6.2 Definition of the parameters

Add the family name to the key list.
\pst@addfams{pst-ob3d}

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

\OnlyVisibleFaces to show only the visible faces.
\define@boolkey[psset]{pst-ob3d}{PstObjectsThreeD@}{OnlyVisibleFaces}{true}{}

\RandomFaces to define randomly the faces of a die, using the ‘random’ package.
\define@boolkey[psset]{pst-ob3d}{PstObjectsThreeD@}{RandomFaces}{true}{}

\Corners to draw corners.
\define@boolkey[psset]{pst-ob3d}{PstObjectsThreeD@}{Corners}{true}{}

\CornersColor to choose the color of the corners.
\define@key[psset]{pst-ob3d}{CornersColor}{\def\PstObjectsThreeD@CornersColor{#1}}

\CornersLength to choose the length of the corners.
\define@key[psset]{pst-ob3d}{CornersLength}{\def\PstObjectsThreeD@CornersLength{#1}}
\def\PstObjectsThreeD@CornersLengthTmpA{#1}
\def\PstObjectsThreeD@CornersLengthTmpB{#1}
Next, we set the default values for all these new parameters.

\begin{verbatim}
\psset[pst-ob3d]{OnlyVisibleFaces=false,RandomFaces=false, 
Corners=false,CornersColor=black,CornersLength=0.15}
\end{verbatim}

\subsection*{6.3 Cube definition}

\begin{verbatim}
\PstCube
\end{verbatim}

\begin{verbatim}
\PstCube\@ifnextchar[{{\PstCube@i}{\PstCube@i[]}}
\end{verbatim}

\begin{verbatim}
\PstCube@i
\end{verbatim}

\begin{verbatim}
\def\PstCube@i[#1]{\@ifnextchar({\PstCube@ii[#1]}{\PstCube@ii[#1](0,0,0)}}
\end{verbatim}

\begin{verbatim}
\PstCube@ii
\end{verbatim}

\begin{verbatim}
\def\PstCube@ii[#1](#2,#3,#4)#5#6#7{{%
    We force “\texttt{dimen=middle}”.
    Then we set the local changes of parameters.
    \psset{\texttt{\#1}}%
    For debugging purposes, we can require to print letters centered on the faces.
\end{verbatim}

\begin{verbatim}
\ifnum\Pst@Debug=\@ne
\def\PstObjectsThreeDFaceCenterA{A}%
\def\PstObjectsThreeDFaceCenterB{B}%
\def\PstObjectsThreeDFaceCenterC{C}%
\def\PstObjectsThreeDFaceCenterD{D}%
\def\PstObjectsThreeDFaceCenterE{E}%
\def\PstObjectsThreeDFaceCenterF{F}%
\fi
\end{verbatim}

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.

\begin{verbatim}
\pst@expandafter\psget@@viewpoint\psk@viewpoint {} {} {} \@nil
\end{verbatim}

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

\begin{verbatim}
\ifPstObjectsThreeD@Corners
\let\PstObjectsThreeD@Shape\PstObjectsThreeD@Octogon
\else
\let\PstObjectsThreeD@Shape\psframe
\fi
\end{verbatim}

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

\begin{verbatim}
\iftikzdim\Pst@dimc>\z@
\iftikzdim\Pst@dimc>\z@
\iftikzdim\Pst@dimc>\z@
\end{verbatim}
Case where $x > 0; y > 0; z > 0$

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

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

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

\ifdim\pst@dimb<\z@

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

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

\ifdim\pst@dima<\z@

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

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

\ifdim\pst@dimb<\z@

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

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

\ifdim\pst@dima<\z@

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

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

\ifdim\pst@dimb<\z@

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

\PstCube\DrawFaces{A}{C}{D}{E}{B}{F}{(#2,#3,#4)}{(#5,#6,#7)}%
\fi
\fi
\fi
\fi
\fi}
6.4 Die definition

\PstDie
\def\PstDie{\ifnextchar[{\PstDie@i}{\PstDie@i[]}}
\PstDie@i
\def\PstDie@i[#1]{\ifnextchar[{{\PstDie@ii[#1]}\else\PstDie@ii[#1](0,0,0)}
\PstDie@ii
\def\PstDie@ii[#1](#2,#3,#4){% 
  \psset{dimen=middle}%
  We set the local changes of parameters.
  \psset{\#1}%
  A die is only a cube where the six faces have dots between 1 and 6.
  \ifPstObjectsThreeD@RandomFaces
  First the case where the faces are defined randomly.
  \setrannum{\pst@cnth}{1}{6}%
  \PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.5,.5)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.2,.2)(.8,.8)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.2,.2)(.5,.5)(.8,.8)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.2,.2)(.2,.5)(.2,.8)(.8,.2)(.8,.5)(.8,.8)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.2,.2)(.2,.8)(.5,.5)(.8,.2)(.8,.8)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \expandafter\def\csname PstObjectsThreeDFace\@tempa\endcsname{%
    \psdots(.2,.2)(.8,.2)(.8,.8)%
  }%\PstDie@Letter{\pst@cnth}{\@tempa}%
  \else
  Then the case where the faces are defined fixedly (the face “4” will be above).
  \def\PstObjectsThreeDFaceA{\psdots(.5,.5)}%
  \def\PstObjectsThreeDFaceB{\psdots(.2,.2)(.8,.8)}%
  \def\PstObjectsThreeDFaceC{\psdots(.2,.2)(.5,.5)(.8,.8)}%
  \def\PstObjectsThreeDFaceD{\psdots(.2,.2)(.2,.5)(.2,.8)(.8,.2)(.8,.5)(.8,.8)}%
  \def\PstObjectsThreeDFaceE{\psdots(.2,.2)(.2,.8)(.5,.5)(.8,.2)(.8,.8)}%
  \def\PstObjectsThreeDFaceF{\psdots(.2,.2)(.2,.8)(.8,.2)(.8,.8)}%
Now that the faces are defined, we can draw the cube, with only the visible faces if corners are not to be drawn.

\begin{verbatim}
\ifPstObjectsThreeD@Corners
\else
\PstObjectsThreeD@OnlyVisibleFacestrue
\fi
\PstCube(#2,#3,#4){1}{1}{1}}
\end{verbatim}

\section{Internal functions to draw the faces}

To draw successively the six faces in the right order.

\begin{verbatim}
\PstCube@DrawFaces
\def\PstCube@DrawFaces#1#2#3#4#5#6#7#8{%
\ifPstObjectsThreeD@OnlyVisibleFaces
\else
\bgroup
\ifPstObjectsThreeD@Corners
\psset{linecolor=\PstObjectsThreeD@CornersColor, fillcolor=\PstObjectsThreeD@CornersColor}%
\fi
\bgroup
\ifPstObjectsThreeD@Corners
\def\PstObjectsThreeD@CornersLengthTmpA{0}%
\fi
\expandafter\csname PstCube@DrawFace#1\endcsname#7#8
\expandafter\csname PstCube@DrawFace#2\endcsname#7#8
\egroup
\ifPstObjectsThreeD@Corners
\def\PstObjectsThreeD@CornersLengthTmpB{0}%
\fi
\expandafter\csname PstCube@DrawFace#3\endcsname#7#8
\expandafter\csname PstCube@DrawFace#4\endcsname#7#8
\expandafter\csname PstCube@DrawFace#5\endcsname#7#8
\expandafter\csname PstCube@DrawFace#6\endcsname#7#8
\egroup
\ifPstObjectsThreeD@Corners
\def\PstObjectsThreeD@CornersLengthTmpA{0}%
\fi
\expandafter\csname PstCube@DrawFace#1\endcsname#7#8
\expandafter\csname PstCube@DrawFace#2\endcsname#7#8
\expandafter\csname PstCube@DrawFace#3\endcsname#7#8
\expandafter\csname PstCube@DrawFace#4\endcsname#7#8
\expandafter\csname PstCube@DrawFace#5\endcsname#7#8
\expandafter\csname PstCube@DrawFace#6\endcsname#7#8
\egroup
\expandafter\csname PstCube@DrawFace#1\endcsname#7#8
\expandafter\csname PstCube@DrawFace#2\endcsname#7#8
\expandafter\csname PstCube@DrawFace#3\endcsname#7#8
\expandafter\csname PstCube@DrawFace#4\endcsname#7#8
\expandafter\csname PstCube@DrawFace#5\endcsname#7#8
\expandafter\csname PstCube@DrawFace#6\endcsname#7#8
\end{verbatim}

To draw the face A.

\begin{verbatim}
\PstCube@DrawFaceA
\def\PstCube@DrawFaceA(#1,#2,#3)(#4,#5,#6){%
\psdimaa=-#4\psxunit
\divide\psdimaa\tw@
\psdimaa=-#5\psxunit
\divide\psdimaa\tw@
\psdimaa=-#6\psyunit
\divide\psdimaa\tw@
\end{verbatim}
To draw the face B.

\texttt{\textbackslash PstCube@DrawFaceB}

\begin{verbatim}
\def\PstCube@DrawFaceB(#1,#2,#3)(#4,#5,#6){
\pst@dimg=-#4\psxunit
\advance \pst@dimg by #1\psxunit
\pst@dima=#4\psxunit
\divide\pst@dima\tw@
\pst@dimb=#5\psxunit
\divide\pst@dimb\tw@
\ThreeDput[\normal=0 0 1](\pst@dimg,#2,#3){%
\psyunit=\psxunit
\PstObjectsThreeD@Shape(#4,#5)
\ifx\PstObjectsThreeDFaceC\@undefined
\else
\rput[bl](0,0){\PstObjectsThreeDFaceC}
\fi
\ifx\PstObjectsThreeDFaceCenterC\@undefined
\else
\rput[c](\pst@dima,\pst@dimb){\PstObjectsThreeDFaceCenterC}
\fi}}
\end{verbatim}

To draw the face C.

\texttt{\textbackslash PstCube@DrawFaceC}

\begin{verbatim}
\def\PstCube@DrawFaceC(#1,#2,#3)(#4,#5,#6){
\pst@dimb=\pst@dima=\pst@dimc=\psxunit
\advance \pst@dimb by \pst@dima\psxunit
\divide\pst@dimb\tw@
\divide\pst@dima\tw@
\divide\pst@dimc\tw@
\ThreeDput[\normal=0 -1 0](#1,#2,#3){%
\PstObjectsThreeD@Shape(-#4,#6)
\ifx\PstObjectsThreeDFaceA\@undefined
\else
\rput[bl](-#4,0){\PstObjectsThreeDFaceA}
\fi
\ifx\PstObjectsThreeDFaceCenterA\@undefined
\else
\rput[c](\pst@dima,\pst@dimc){\PstObjectsThreeDFaceCenterA}
\fi}
\end{verbatim}
To draw the face D.

\PstCube@DrawFaceD
\def\PstCube@DrawFaceD(#1,#2,#3)(#4,#5,#6){%
    \pst@dimg=#5\psxunit
    \advance \pst@dimg by #2\psyunit
    \pst@dimd=-#4\psxunit
    \divide \pst@dima \tw@
    \pst@dimc=#6\psyunit
    \divide \pst@dimc \tw@
    \ThreeDput[normal=0 -1 0](#1,\pst@dimg,#3){%
        \PstObjectsThreeD@Shape(-#4,#6)
        \ifx\PstObjectsThreeDFaceD\@undefined
            \else
                \rput[bl]{(-4,0)\{\PstObjectsThreeDFaceD}
        \fi
    \ifx\PstObjectsThreeDFaceCenterD\@undefined
        \else
            \rput[c]({\pst@dima,\pst@dimc}\{\PstObjectsThreeDFaceCenterD}
    \fi}
}

To draw the face E.

\PstCube@DrawFaceE
\def\PstCube@DrawFaceE(#1,#2,#3)(#4,#5,#6){%
    \pst@dimg=-#4\psxunit
    \advance \pst@dimg by #1\psxunit
    \pst@dimb=#5\psxunit
    \divide \pst@dimb \tw@
    \pst@dimc=#6\psyunit
    \divide \pst@dimc \tw@
    \ThreeDput[normal=1 0 0](\pst@dimg,#2,#3){%
        \PstObjectsThreeD@Shape(#5,#6)
        \ifx\PstObjectsThreeDFaceE\@undefined
            \else
                \rput[bl]{(0,0)\{\PstObjectsThreeDFaceE}
        \fi
    \ifx\PstObjectsThreeDFaceCenterE\@undefined
        \else
            \rput[c]({\pst@dimb,\pst@dimc}\{\PstObjectsThreeDFaceCenterE}
    \fi}
}

To draw the face F.

\PstCube@DrawFaceF
\def\PstCube@DrawFaceF(#1,#2,#3)(#4,#5,#6){%
    \pst@dimg=-#4\psxunit
    \advance \pst@dimg by #1\psxunit
    \pst@dimh=#6\psyunit
    \advance \pst@dimh by #3\psyunit
    \ifx\PstObjectsThreeDFaceF\@undefined
        \else
            \rput[c]({\pst@dimh,\pst@dimc}\{\PstObjectsThreeDFaceF}
    \fi}
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

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

\PstDie@Letter
\def\PstDie@Letter#1#2{\% 
\advance#1\@ne
\ifnum#1>6
#1=\@ne
\fi
\def#2{\ifcase#1\or A\or B\or C\or D\or E\or F\fi}}

6.7 Closing
Catcodes restoration.
\catcode\@=\PstAtCode\relax
(/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.
<table>
<thead>
<tr>
<th>Package</th>
<th>Pages</th>
</tr>
</thead>
<tbody>
<tr>
<td>\PstCube@ii</td>
<td>88, 89</td>
</tr>
<tr>
<td>\PstDie</td>
<td>135</td>
</tr>
<tr>
<td>\PstDie@i</td>
<td>135, 136</td>
</tr>
<tr>
<td>\PstDie@ii</td>
<td>136, 137</td>
</tr>
<tr>
<td>\PstDie@Letter</td>
<td>142, 145, 148, 151, 154, 157, 337</td>
</tr>
<tr>
<td>\PstObjectsThreeD@CornersColor</td>
<td>80, 179, 180</td>
</tr>
<tr>
<td>\PstObjectsThreeD@CornersLength</td>
<td>82, 307, 314, 321, 328</td>
</tr>
<tr>
<td>\PstObjectsThreeD@CornersLengthTmpA</td>
<td>83, 184, 318, 324</td>
</tr>
<tr>
<td>\PstObjectsThreeD@CornersLengthTmpB</td>
<td>84, 191, 310, 332</td>
</tr>
<tr>
<td>\PstObjectsThreeD@Octogon</td>
<td>102, 306</td>
</tr>
<tr>
<td>\PstObjectsThreeD@OnlyVisibleFacetrue</td>
<td>171</td>
</tr>
<tr>
<td>\PstObjectsThreeD@Shape</td>
<td>102, 104, 206, 221, 239, 256, 273, 293</td>
</tr>
</tbody>
</table>

## Change History

<table>
<thead>
<tr>
<th>Version</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>v0.1 Beta</td>
<td>General: First beta version in 1994 (unpublished)</td>
</tr>
<tr>
<td>v0.11 Beta</td>
<td>General: Fourth packaged release.</td>
</tr>
<tr>
<td>v0.20</td>
<td>General: First CTAN release</td>
</tr>
<tr>
<td>v0.21</td>
<td>General: Updated the style file</td>
</tr>
</tbody>
</table>