The 'pst-lens' package
A PSTricks package for lens

Denis Girou*
and
Manuel Luque†

Version 1.0
February 19, 2001
Documentation revised February 19, 2001

Abstract

This package define a lens which can be used in various contexts to simulate the effect of a lens, using the unique macro `\PstLens`, with some customization parameters.

It is also a good example of the great power and flexibility of PSTricks, as in fact it is a very short program (it body, without considering the various customizations, is only 7 lines long!) but nevertheless powerful.

And last, it is also a good pedagogical example of how to design and program *high level graphic objects* above PSTricks own ones.

Contents

1 Introduction

'pst-lens' offer a unique macro with some parameters to interact on it.

The syntax is simply: `\PstLens[optional_parameters](x,y){Object}`

`(x,y)` is a PSTricks coordinate, which as usual is taken as `(0,0)` if it is not defined.

To use the lens, we must define a `pspicture` environment, optionally draw the object and then call the `\PstLens` macro on it.

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

†<Mluque5130@aol.com>. The original idea and the first version of the lens were from Manuel Luque.

2 Usage

We will use the following textual object to illustrate our examples (note that we must define the reference point at the left bottom corner, as it is the normal behavior of PSTricks):

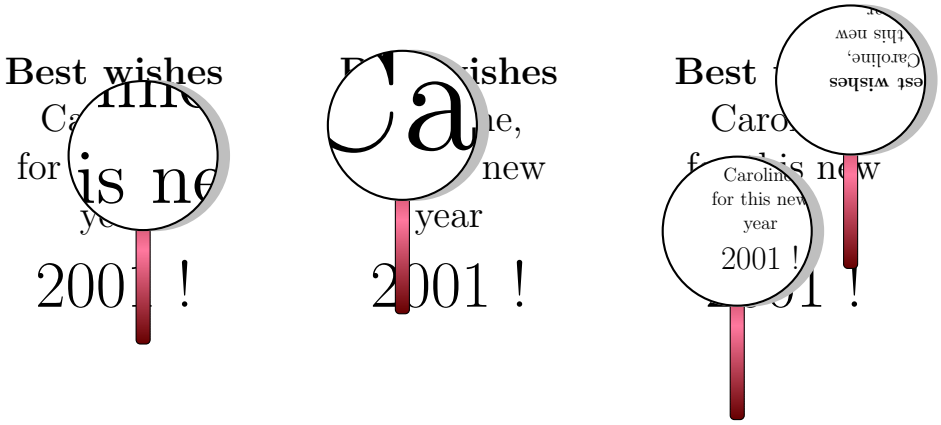

```
\def\Wishes{{%
\rput[lb](0,0){%
  \Large
  \begin{minipage}{3cm}
    \centering
    \textbf{Best wishes}\\
    Caroline,\\
    for this new year\\
    \Huge 2001 !
  \end{minipage}}}}

\Wishes
```

2.1 Parameters

There are **9** specific parameters defined to change the way the lens works:

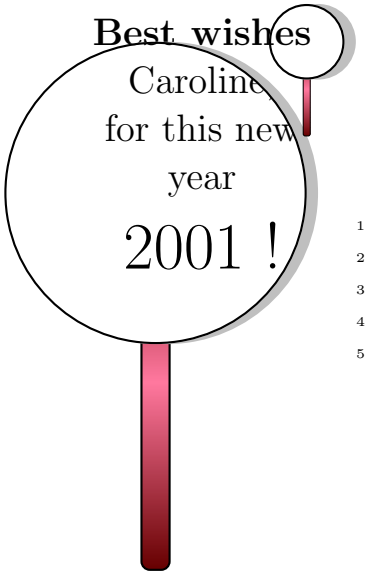
LensMagnification (real) : magnification to apply for the lens (*Default: 1* — no magnification).

```
\begin{pspicture}(0,-1.5)(3,4)
  \Wishes
  \PstLens[LensMagnification=2](2,2){\Wishes}
\end{pspicture}
\hfill
\begin{pspicture}(0,-1.5)(3,4)
  \Wishes
  \PstLens[LensMagnification=4](1,2.4){\Wishes}
\end{pspicture}
\hfill
\begin{pspicture}(0,-1.5)(3.5,4)
  \Wishes
  \PstLens[LensMagnification=0.5](1,1){\Wishes}
  \PstLens[LensMagnification=-0.5](2.5,3){\Wishes}
\end{pspicture}
```



LensSize (real or length) : value of the radius of the glass of the lens (*Default: 1*).

Note that the size of the handle will change accordingly.



```
\begin{pspicture}(0,-4)(3,3.5)
  \Wishes
  \PstLens[LensSize=2](1,1){\Wishes}
  \PstLens[LensSize=0.5](3,3){\Wishes}
\end{pspicture}
```

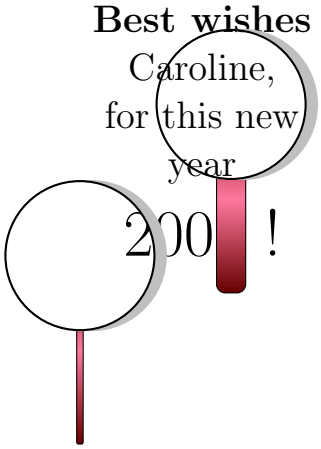
LensRotation (real) : rotation angle applied to the lens (*Default: 0* — no rotation).

```
\begin{pspicture}(0,-1)(3,3.8)
  \Wishes
  \PstLens[LensRotation=80]{\Wishes}
  \PstLens[LensRotation=-108.5](2,2){\Wishes}
\end{pspicture}
```

LensHandle (boolean) : boolean value to choose between to draw a handle for the lens or not. (*Default: true* — handle).


```
\begin{pspicture}(3,3.5)
  \Wishes
  \PstLens[LensHandle=false](2,2){\Wishes}
\end{pspicture}
```

LensHandleWidth (real or length) : width of the handle (*Default: 0.2 for* `LensSize=1`).




```
\begin{pspicture}(0,-2.5)(3,3.5)
  \Wishes
  \PstLens[LensHandleWidth=0.1]{\Wishes}
  \PstLens[LensHandleWidth=4mm](2,2){\Wishes}
\end{pspicture}
```

LensHandleHeight (real or length) : height of the handle (*Default: 2.5 for* `LensSize=1`).

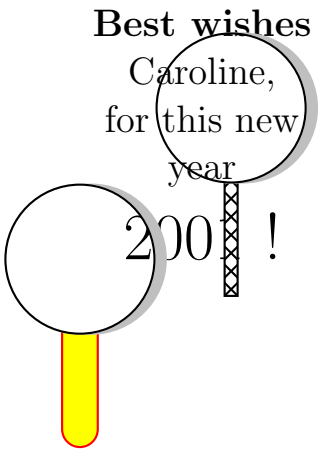
Take care that this length is between the *center* of the glass and the bottom of the handle.


```
\begin{pspicture}(0,-2)(3,3.5)
  \Wishes
  \PstLens[LensHandleHeight=15mm]{\Wishes}
  \PstLens[LensHandleHeight=4](2,2){\Wishes}
\end{pspicture}
```

LensStyleHandle (style) : name of the PSTricks style for the handle. (*Default: LensStyleHandle*).

Its default value is:

```
\newpsstyle{LensStyleHandle}{%
  fillstyle=gradient,framearc=0.6,linewidth=0.5\pslinewidth,
  gradmidpoint=0.5,gradangle=\PstLens@Rotation,
  gradbegin=Brown,gradend=Salmon}
```




```
\begin{pspicture}(0,-2.5)(3,3.5)
  \Wishes
  \newpsstyle{HandleYellow}{%
    linecolor=red,framearc=1,
    fillstyle=solid,fillcolor=yellow}
  \PstLens[LensHandleWidth=0.5,
           LensStyleHandle=HandleYellow]
          {\Wishes}
  \newpsstyle{HandleCrosshatch}{%
    fillstyle=crosshatch*,fillcolor=white}
  \PstLens[LensStyleHandle=HandleCrosshatch]
          (2,2){\Wishes}
\end{pspicture}
```

LensShadow (boolean) : boolean value to choose between to draw a shadow for the glass of the lens or not. (*Default: true* — shadow).

Note that if we redefine the `LensStyleGlass` parameter without explicitely require a shadow, there will be none even if `LensShadow` will have the `true` value.


```
\begin{pspicture}(0,-0.5)(3,3.5)
  \Wishes
  \PstLens[LensShadow=false](2,2){\Wishes}
\end{pspicture}
```

LensStyleGlass (style) : name of the PSTricks style for the glass. (*Default: LensStyleGlass*).

It allow to change the appearance of the glass, but its main utility is probably to be able to define the style of the shadow of the glass. Default definition is:

```
\newpsstyle{LensStyleGlass}{%
  fillstyle=solid,fillcolor=white,
  shadow=true,shadowcolor=lightgray,shadowsize=0.15,
  shadowangle=\PstLens@Rotation}
```

Take care that if we will use later the `LensRotation` parameter with `LensShadow` positioned, we must set the value of the `shadowangle` parameter to `\PstLens@Rotation` to have the shadow rotate accordingly.

And for better shadow effects, you must look at the 'pst-blur' package from Martin Giese.


```
\begin{pspicture}(3,4)
  \Wishes
  \makeatletter
  \newpsstyle{DarkShadow}{%
    fillstyle=solid,fillcolor=white,
    shadow=true,shadowcolor=darkgray,
    shadowsize=0.2,
    shadowangle=\PstLens@Rotation}
  \makeatother
  \PstLens[LensRotation=230,
           LensStyleGlass=DarkShadow](2,2)
          {\Wishes}
\end{pspicture}
```

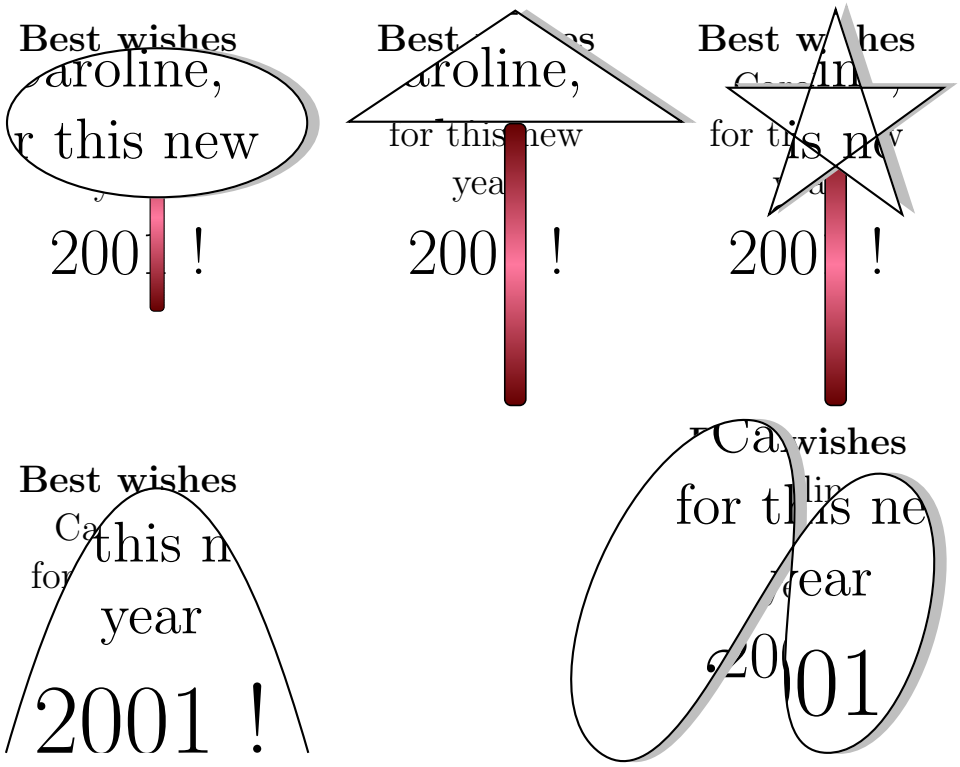

```
\begin{pspicture}(0,-0.5)(3,3.5)
  \Wishes
  \newpsstyle{YellowGlass}{%
    linecolor=red,linewidth=0.1,
    fillstyle=solid,fillcolor=yellow}
  \PstLens[LensStyleGlass=YellowGlass](2,2)
          {\Wishes}
\end{pspicture}
```

2.2 Shape of the glass

The `\PstLensShape` macro define the shape of the glass. It default value is a circle, as in real life, but we can redefine it for various effects...

```
\psset{LensMagnification=1.5}
\begin{pspicture}(0,-1.8)(4,3.8)
  \Wishes
  \renewcommand{\PstLensShape}{\psellipse(2,1)}
  \PstLens(2,2){\Wishes}
\end{pspicture}
\hfill
\begin{pspicture}(-0.5,-1.8)(4,3.8)
  \Wishes
  \renewcommand{\PstLensShape}{\pstriangle(3,1)}
  \PstLens[LensSize=1.5](2,2){\Wishes}
\end{pspicture}
\hfill
\begin{pspicture}(0,-1.8)(3.5,3.8)
  \Wishes
  \renewcommand{\PstLensShape}{%
    \rput{18}{\pspolygon(1;0)(1;144)(1;288)(1;72)(1;216)}}
  \PstLens[LensSize=1.5](2,2){\Wishes}
```

```
\end{pspicture}

\begin{pspicture}(0,-0.5)(4,3.5)
  \renewcommand{\PstLensShape}{%
     \parabola[fillstyle=solid,fillcolor=white](-1,-1.5)(1,2)}
  \Wishes
  \PstLens[LensShadow=false,LensHandle=false](1,1){\Wishes}
\end{pspicture}
\hfill
\begin{pspicture}(-1.5,-1)(3.5,3.5)
  \renewcommand{\PstLensShape}{%
    \psccurve(-1,-1)(0,1.2)(0.5,-1)(1,0.8)}
  \Wishes
  \PstLens[LensSize=2,LensHandle=false](1,1){\Wishes}
\end{pspicture}
```



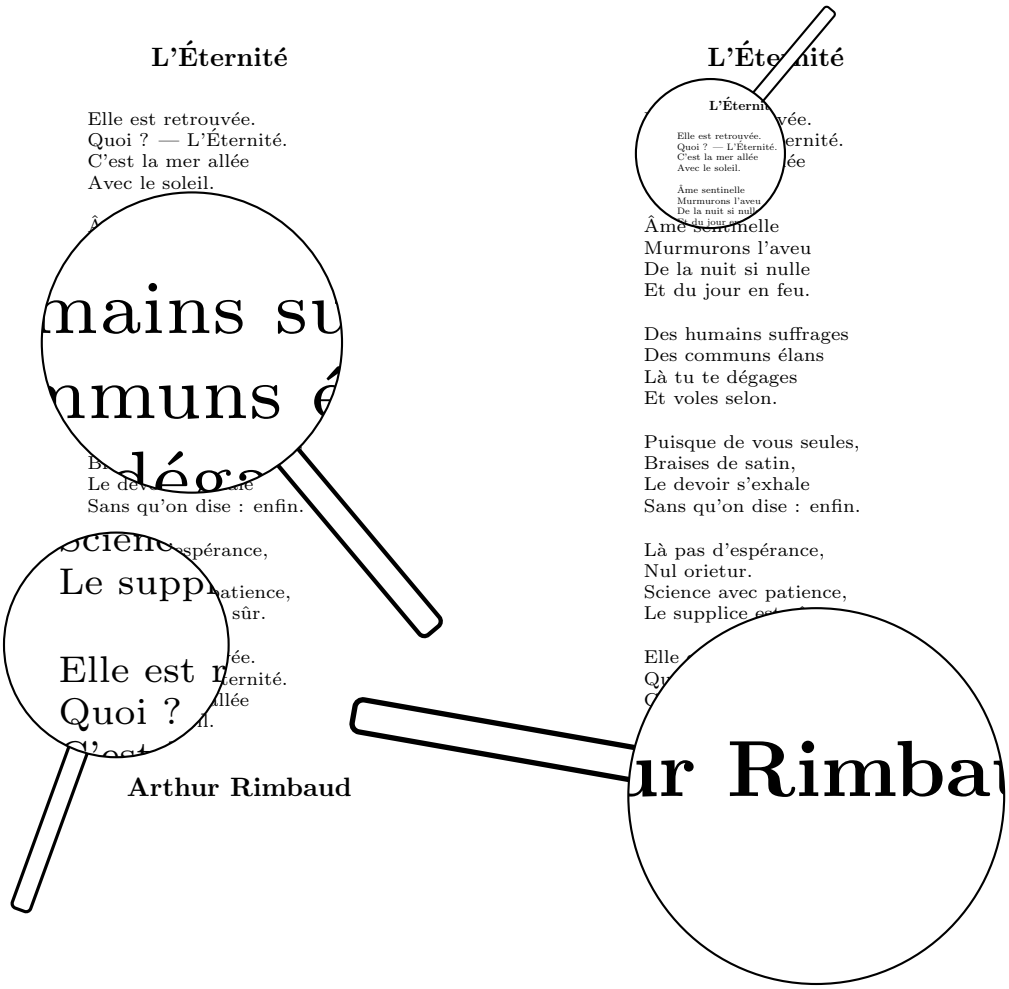
2.3 Examples

We can use the lens for all textual objects and for all PSTricks graphic objects (we use here some versions of tilings and fractals, but only basic ones to avoid requiring too much memory from old TEX systems, to compile the file).

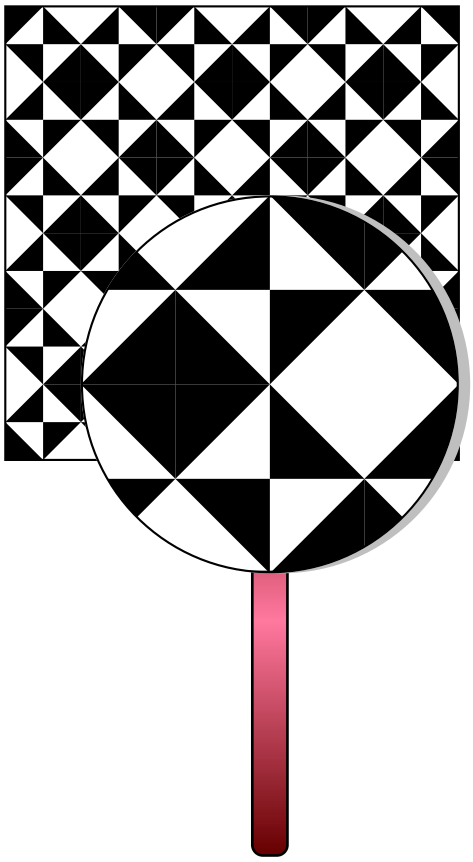
And specially take care to explicitely position the reference point at the left bottom corner and to compute the correct dimensions for the `pspicture` environment (in our examples, we choose most of the time to include the lens inside the bounding boxes, but we can choose to define them just for the objects).

```
\def\TheEternity{{%
\rput[lb](0,0){%
  \scriptsize
  \begin{minipage}{3.5cm}
    \centerline{\normalsize\textbf{L'\'Eternit\'e}}
...
\def\TruchetTiling#1#2{{%
\rput[lb](0,0){%
...
\def\PstSierpinskiTriangle#1{{%
\rput[lb](0,0){%
...
\def\PstVonKochCurve#1{{%
\rput[lb](0,0){%
...
```

```
\newpsstyle{SimpleGlass}{fillstyle=solid,fillcolor=white}
\newpsstyle{SimpleHandle}{fillstyle=solid,fillcolor=white,
                          framearc=0.5}
\psset{LensStyleGlass=SimpleGlass,LensStyleHandle=SimpleHandle}

\begin{pspicture}(-1,-2.5)(5,10.5)
  \TheEternity
  \PstLens[LensSize=2,LensMagnification=4,LensRotation=40]
          (1.5,6){\TheEternity}
  \PstLens[LensSize=1.5,LensMagnification=2,LensRotation=-20]
          (0.5,2){\TheEternity}
\end{pspicture}
\hfill
\begin{pspicture}(-2,-2.5)(4,10.5)
  \TheEternity
  \PstLens[LensMagnification=0.5,LensRotation=140]
          (1,8.5){\TheEternity}
  \PstLens[LensSize=2.5,LensMagnification=3,LensRotation=-100]
          (2.4,0){\TheEternity}
\end{pspicture}
```

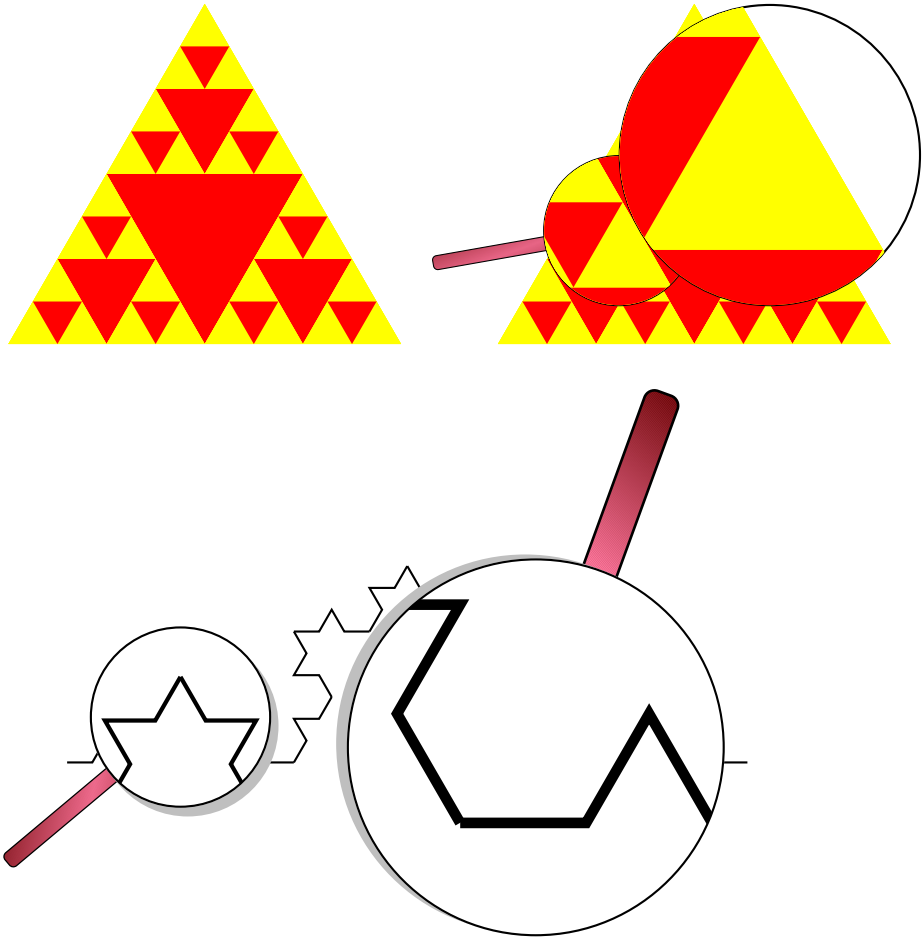



```
\begin{pspicture}(0,-6)(6,6)
  \TruchetTiling
  \PstLens[LensSize=2.5,LensMagnification=2.5](3.5,1)
          {\TruchetTiling}
\end{pspicture}
```



```
\newcommand{\PstSierpinskiInternalColor}{red}
\newcommand{\PstSierpinskiExternalColor}{yellow}

% The Sierpinski triangle is in a unit circle of radius 1,
% so we must define the "pspicture" accordingly: (-3,-2)(3,3)
\begin{pspicture}(-3,-2)(3,3)
  \PstSierpinskiTriangle{3}
\end{pspicture}
\hfill
\begin{pspicture}(-3,-2)(3,3)
  \PstSierpinskiTriangle{3}
  \psset{LensShadow=false}
  \PstLens[LensMagnification=2,LensRotation=-80](-1,0)
          {\PstSierpinskiTriangle{3}}
  \PstLens[LensSize=2,LensMagnification=5,LensRotation=100,
           LensHandle=false](1,1){\PstSierpinskiTriangle{3}}
\end{pspicture}

\begin{pspicture}(-1,-2)(11,5)
  \PstVonKochCurve{3}
```

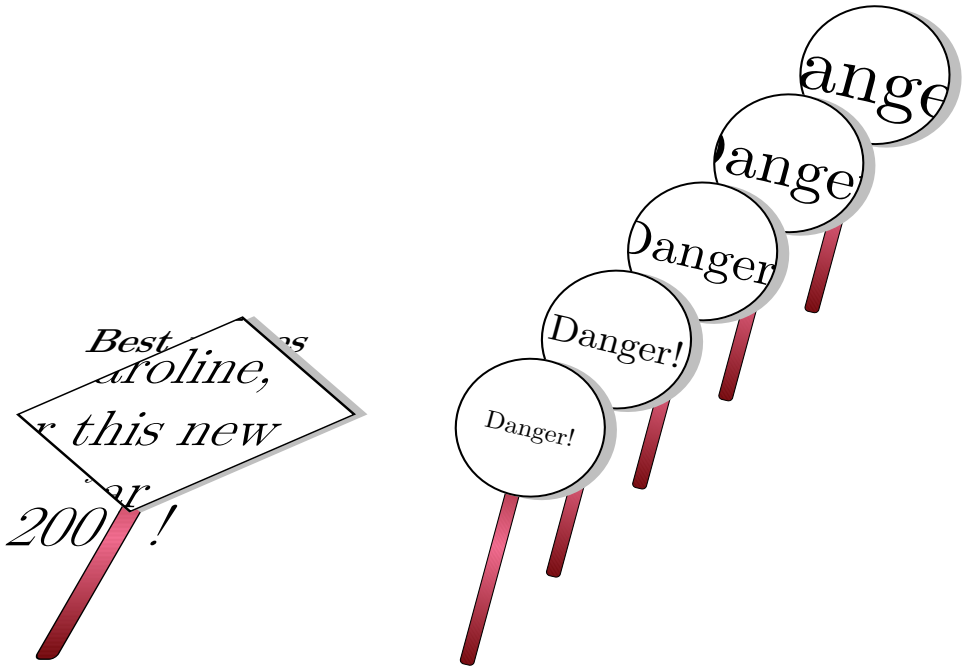
```
  \PstLens[LensSize=1.2,LensMagnification=2,LensRotation=-50]
          (1.5,0.6){\PstVonKochCurve{3}}
  \PstLens[LensSize=2.5,LensMagnification=5,LensRotation=160,
           LensHandleHeight=2](6.2,0.2){\PstVonKochCurve{3}}
\end{pspicture}
```



Of course, as for all PSTricks objects, we can apply to them some transformations. For instance, we can project them in the 3 dimensional space, with the general `\ThreeDput` macro or the simple `\pstilt` one.

```
\psset{LensMagnification=1.5}
\begin{pspicture}(0.8,-1.5)(5.3,3)
  \renewcommand{\PstLensShape}{\psdiamond(1.5,1)}
  \pstilt{60}{%
    \Wishes
    \PstLens[LensSize=1.5](2,2){\Wishes}}
\end{pspicture}
\hfill
\begin{pspicture}(-3,-0.5)(3.5,8)
```

```
  \psset{viewpoint=0.5 -2 5,LensHandleHeight=3.5}
  \multido{\nPosX=0+-0.8,\nPosY=8+-1.5,\nMag=3+-0.5}{5}{%
    \ThreeDput(\nPosX,\nPosY,0){%
      \PstLens[LensMagnification=\nMag](0.6,0.2)
              {\rput[lb](0,0){Danger!}}}}
\end{pspicture}
```



And we can also use the lens on non PSTricks graphics, as external images.

```
\newcommand{\LouiseBrooks}{%
  \rput[lb](0,0){%
    \includegraphics[width=4cm,height=5cm]{LouiseBrooks}}}
\newpsstyle{SimpleGlass}{linestyle=none}
\psset{LensStyleGlass=SimpleGlass}

\begin{pspicture}(0,-1)(4,5)
  \LouiseBrooks
\end{pspicture}
\hfill
\begin{pspicture}(-0.5,-1)(3,5)
  \PstLens[LensHandle=false,LensSize=1.8,LensMagnification=2]
          (1.2,2.3){\LouiseBrooks}
\end{pspicture}
\hfill
\newpsstyle{SimpleHandle}{fillstyle=solid,fillcolor=white,
                          framearc=0.5}
\psset{LensStyleHandle=SimpleHandle}
```

```
\begin{pspicture}(0,-1)(4,5)
  \LouiseBrooks
  \PstLens[LensSize=1.5,LensMagnification=4]
          (1.5,2.5){\LouiseBrooks}
\end{pspicture}
```