Creating Shaded Rectangles with PostScript

David Salomon
California State University, Northridge
Computer Science Department
Northridge, CA 91330 USA
Phone (818) 885-4954; FAX (818) 885-2140
Internet: `vacsc0rf@vax.csun.edu`

Abstract

One of the most common graphics used in documents is text with a shaded background. This is hard to do with TEX but easy with PostScript. Simple PostScript code is presented here to create shaded rectangles, and a macro is developed to combine such a rectangle with text.

Introduction

It is well known that TEX lacks facilities for graphics. Certain drawings, such as logos, may be developed in METAFONT, but for any non-trivial graphics this is usually too time consuming. A ruled box, $\boxed{\text{like this}}$, is the closest TEX can get to anything resembling a drawing. Slanted lines can be handled by typesetting a dot and moving it in small steps. Hendrickson (1985), Cameron (1985) and Salomon (1989) use this idea to develop simple macros for slanted lines. The same principle can be used to typeset curves, as done by `Bezier.sty` in LATEX, and by the PICTEX macro package (Wichura, 1986). For more complex graphics, the `\special` command can be used to include graphics in the final document. Rahtz (1989) is an excellent survey of the different methods used to combine TEX and graphics.

In my work I have often felt the need for enclosing text in ruled boxes, either rectangular or with rounded corners, with backgrounds of shaded gray, and sometimes with a thick stroke around them. Glendown (1989) is an (unsatisfactory) attempt to draw similar boxes using the arcs provided in the LATEX circle fonts. As a heavy Macintosh user, it seemed to me that my best choice was to use PostScript to achieve such effects. PostScript printers are widely available, and PostScript programs are supported, via `\special`, by more and more TEX implementations. The program used here has the additional advantage of being small and fast.

The `\shade` Macro

The result of my efforts is the macro `\shade` below. It creates a rectangular box with either sharp or rounded corners around text. The text can be placed, by the user, in either an `\hbox` or a `\vbox`, or it can be written explicitly as the macro argument (see third example below). The rectangle is filled with the desired shade of gray, and is optionally surrounded by a stroke of any desired thickness. It is also possible to make the shaded area larger than the text by any amount. The macro has five parameters:

- A decimal number specifying the proportion of white in the shaded background.
$\boxed{\text{This is a .97 background.}}$
- A dimension specifying the
$\boxed{\text{extra size of shaded area (12pt on each side).}}$
- A count specifying the width of the stroke in points. A zero or empty argument creates no stroke.
- A count—$\boxed{\text{the radius of the rounded corners,}}$ in points. A zero or empty argument produces square corners.
- The text to be boxed. It is either straight text (if it fits on one line) or is enclosed, by the user, in an `\hbox` or `\vbox`.

The arguments are separated by commas, and the last one is delimited by '`\\`'.

Here is the listing of `\shade`.

```
\newdimen\tmp \newdimen\tmpw
\newdimen\tmpk \newdimen\tmph
\newcount\heit \newcount\widf
\newcount\strk \newcount\radius
\def\shade#1,#2,#3,#4,#5\\{%
\setbox1=\hbox{#5}%
\def\inner{#3}\ifx\inner\empty \strk=0
 \else\strk=#3\fi
\def\inner{#4}\ifx\inner\empty \radius=0
\else \radius=#4\fi
\tmp=#2 \tmp=2\tmp \advance\tmp\wd1
 \widf=\tmp \divide\widf65536
\tmp=#2 \tmp=2\tmp \advance\tmp\ht1
 \heit=\tmp \divide\heit65536
\setshadebox{#1}{\the\heit}{\the\widf}
 {\the\strk}{\the\radius}%
\tmp=#2
\tmpw=#2 \tmpw=2\tmpw
 \advance\tmpw by\the\strk pt
 \advance\tmpw\wd1
\tmpk=\the\strk pt \divide\tmpk by2
\tmph=#2 \tmph=2\tmph
 \advance\tmph by\the\strk pt
 \advance\tmph\ht1
\vbox to\tmph{\vss
\hbox to\tmpw{\kern\tmpk\lower\tmp\box0%
 \kern-\tmpk\hfil\box1\hfil}\vss}}
% The ps commands are placed, as the argument
% of a \special, at the bottom of
% \box0 whose width is set to zero.
\def\setshadebox#1#2#3#4#5{%
\setbox0=\vbox to\tmp{\hsize=0pt\vfil
 \special{postscript
/fpops{4 {pop} repeat} def
#3 0 moveto 0 0 0 #2 #5 arcto
newpath % start the rounded-corner rectangle
moveto % the coordinates are in stack
0 #2 #3 #2 #5 arcto fpops
#3 #2 #3 0 #5 arcto fpops
#3 0 0 0 #5 arcto fpops
0 0 0 #2 #5 arcto fpops closepath
#4 0 ne
 {gsave #4 setlinewidth stroke grestore} if
% stroke width of zero not recommended
#1 setgray fill}}}
```

The `\special` with the PostScript commands is placed at the bottom of `\box0`, which is set to width zero. The text is placed in `\box1`. Both boxes are then placed, side by side, in an `\hbox` set to the right width (the width of the text, plus the extra width of the shaded area and the stroke).

The macro is normally used in vertical mode, but also works well in horizontal mode, not introducing any spurious spaces (the percent signs at the end of certain lines should not be removed). It has also been tested in both math modes. Note that PostScript code should normally be surrounded by a `gsave`, `grestore` pair. However, I use TEXtures, which automatically supplies these commands.

Examples

The following examples show samples of input and their respective output:

```
\shade0.90,16pt,5,10,\vbox{\hsize=100pt
\noindent A test in vertical mode.
Filled with 90\% white \&
a 5pt stroke}\\
```

A test in vertical mode.
Filled with 90% white
& a 5pt stroke

```
\shade0.97,5pt,1,3,%
straight text, not pre-boxed\\
```

straight text, not pre-boxed

```
\shade0.90,5pt,,3,%
\hbox{No stroke at all}\\
```

No stroke at all

```
\noindent A\lower30pt\hbox{%
\shade0.98,10pt,1,5,
\vbox{\hsize=100pt
 A test in horizontal mode. There are
no spaces between this text and the
‘A’ or the ‘D’ (98\% white)}\\}D
```

A very heavy stroke

```
\shade0.99,8pt,40,,%
\hbox{A very heavy stroke}\\
```

A A test in horizontal mode. There are no spaces between this text and the ‘A’ or the ‘D’ (98% white) D

```
\shade0.5,5pt,1,5,\hbox{\shade.99,5pt,1,3,%
 A Nested Expansion\\}\\
```

A Nested Expansion

Using the Macro in Practice

In practice, one usually uses shaded boxes of the same type throughout a document. This suggests using macros such as

```
\def\shadedbox#1{\shade0.95,1em,1,,#1\\}
```

```
\def\shadedRbox#1{\shade0.95,1em,1,5,#1\\}
```

instead of using of `\shade` directly.

The following examples below combine `\shade` with rules and leaders.

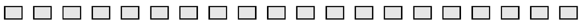
```
\vbox{\offinterlineskip
\shade0.97,4pt,1,,Section 3:\\
\hrule height4pt}
```

Section 3:

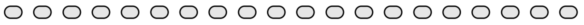
```
\hbox{\vrule width4pt%
 \shade0.95,4pt,,,Note!\\}
```

Note!

```
\line{\leaders\hbox{%
 \shade0.9,0pt,1,,\vrule height4pt width0pt%
 \kern6pt\\\kern4pt}\hfil}
```



```
\line{\leaders\hbox{%
 \shade0.9,0pt,1,2,\vrule height4pt width0pt%
 \kern6pt\\\kern4pt}\hfil}
```



References

Cameron, A.G.W. “Wiggly Lines.” *TUGboat* **6**(3), pages 155 – 156, 1985.

Glendown, G. “Rounded Boxes for `plain` TEX.” *TUGboat* **10**(3), pages 385 – 386, 1989.

Hendrickson, A. “Some Diagonal Line Hacks.” TUGboat **6**(2), pages 83 – 86, July 1985.

Rahtz, S. *A Survey of TEX and Graphics*. University of Southampton, Tech. Rep. CSTR 89-7, 1989.

Salomon, D. “DDA Methods in TEX.” *TUGboat* **10**(2), pages 207 – 216, 1989.

Wichura, M.J. *The PICTEX Manual*. Number 6 in the *TEXniques* series. Providence: TEX Users Group, 1986.