The `roundrect` Macros, v1.0

Donald P. Goodman III

July 29, 2015

Abstract

The `roundrect` macros for METAPOST provide extremely configurable, extremely versatile rectangles (including rounded corners), intended primarily for inclusion in documents produced by TEX and friends. The idea was to provide a METAPOST-based replacement for the incredibly versatile `tcolorbox` package; the macros are far from achieving that goal. But they are nevertheless extremely useful.

Contents

1 Introduction

While Ti*k*Z and its many accompanying packages, particularly `tcolorbox`, are wonderful and powerful tools, whenever using them I inevitably feel completely lost, and I exert great effort doing comparatively simple things. Contrariwise, thanks to my experience with the `drm` and `dozenal` packages, writing in METAPOST is quite straightforward for me. So I decided to try to write some generalized macros to provide functionality similar to that of `tcolorbox`. It's not even close to that kind of flexibility or power, but it's still quite useful and versatile, so I make it available for anyone who might be interested.

This document was typeset in accordance with the `docstrip` utility, which allows the automatic extraction of code and documentation from the same document.

2 Prerequisites and Conventions

Some prerequesites for using this package are METAPOST itself (obviously). If you're using the package with LaTeX, the `gmp` package would probably be helpful; be sure to use the `latex` package option. Finally, the package internally calls `TEX.mp`, so that is also required. All of these should be packaged in any reasonably modern LaTeX system, such as TeXLive or MikTeX.

This documentation assumes nothing about your personal TeX or METAPOST environment. ConTeXt and the various forms of LuaTeX have METAPOST built-in; with pdfLaTeX, the author's choice, one can use the `gmp` package to include the source directly in one's document (that's what's been done in this documentation) or develop a simple script to compile them afterwards and include them in the source via `\includegraphics` (probably the quickest option, since compilation is done in advance). Here, we simply post the plain vanilla METAPOST code, and let you work out those details however you prefer.

3 Basic Usage

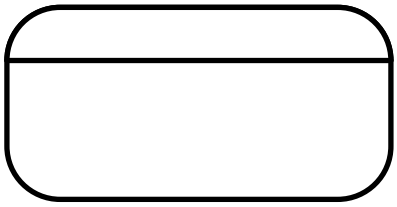
The core of all the action is the `roundrect` macro; this will set up your rounded rectangle in the plainest way possible. The first argument is the box's height, the second its width, and the third its name, by which you will draw it later:

```
roundrect(1in,2in)(rectangle);
draw rectangle;
```

We have some options with this. If you want a title bar in your box, say `rrtitlebar := true;`:

`rrtitlebar`

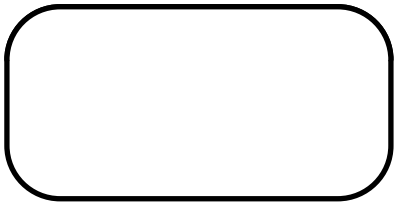
```
rrtitlebar := true;
roundrect(1in,2in)(rectangle);
draw rectangle;
```



By default, `roundrect` displays a lower border for the title bar; if you don't like this, say `rrtitlebotborder := false;`:

`rrtitlebotborder`

```
rrtitlebar := true;
rrtitlebotborder := false;
roundrect(1in,2in)(rectangle);
draw rectangle;
```



The command is called `rrtitlebotborder` because it's usually on the bottom of the title bar; however, it retains this name even when the title bar is elsewhere.

Although it's impossible to tell whether this box still has a title bar, we can verify it (if we don't believe it) by changing the background color with `rrtitlebgcolor`:

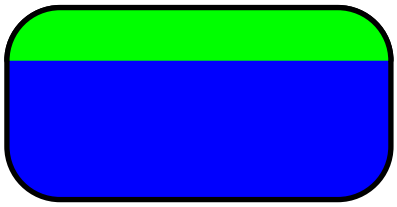
`rrtitlebgcolor`

```
rrtitlebar := true;
rrtitlebotborder := false;
rrtitlebgcolor := green;
roundrect(1in,2in)(rectangle);
draw rectangle;
```

And while we're at it, we may as well color the background of the main part of the box with `rrinnercolor`:

`rrinnercolor`

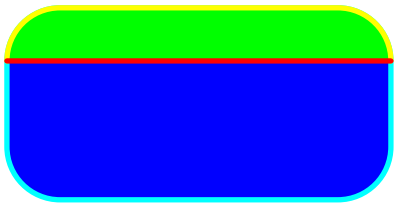
```
rrtitlebar := true;
rrtitlebotborder := false;
rrinnercolor := blue;
rrtitlebgcolor := green;
roundrect(1in,2in)(rectangle);
draw rectangle;
```



We can also change the colors of the border around the box (`rrbordercolor`), the colors of the title bar's border (`rrtitlebordercolor`), and the line between the title bar and the body, if any (`rrtitlebotbordercolor`):

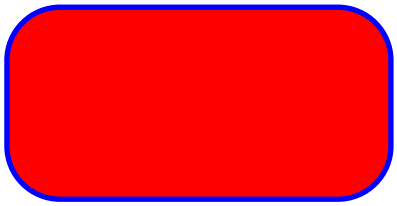
`rrbordercolor`
`rrtitlebordercolor`
`rrtitlebotbordercolor`

```
rrtitlebar := true;
rrtitlebotborder := false;
rrinnercolor := blue;
rrbordercolor := (0,1,1);
rrtitlebordercolor := (1,1,0);
rrtitlebotbordercolor := red;
rrtitlebotbordercolor := red;
rrtitlebgcolor := green;
roundrect(1in,2in)(rectangle);
draw rectangle;
```



If there is no title bar, then `rrinnercolor` and `rrbordercolor` will affect the entire box:

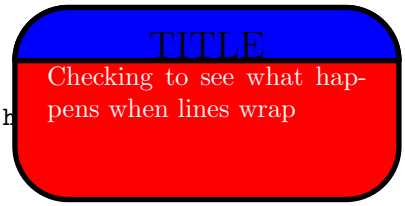
```
rrinnercolor := red;
rrbordercolor := blue;
roundrect(1in,2in)(rectangle);
draw rectangle;
```



`rrtitletext` `rrtitlecolor` `rrbodytext` `rrtextcolor`

We can put text in the title bar with `rrtitletext` (which is colored by `rrtitlecolor`) and `rrbodytext` (which is colored by `rrbodytextcolor`):

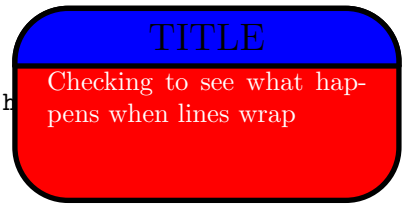
```
rrinnercolor := red;
rrtitlebgcolor := blue;
rrtitlecolor := black;
rrtextcolor := white;
rrbodytext := "Checking to see what h
rrtitletext := "TITLE";
roundrect(1in,2in)(rectangle);
draw rectangle;
```



`rrtitleht`

We see we should raise the title text; the default value is $\frac{1}{6}$ of the box's height, which is too large for this small box. So we adjust it with `rrtitleht`; this is expressed as a distance up from the bottom border of the title bar:

```
rrinnercolor := red;
rrtitlebgcolor := blue;
rrtitlecolor := black;
rrtextcolor := white;
rrbodytext := "Checking to see what h
rrtitletext := "TITLE";
rrtitleht := 0.3in;
roundrect(1in,2in)(rectangle);
draw rectangle;
```

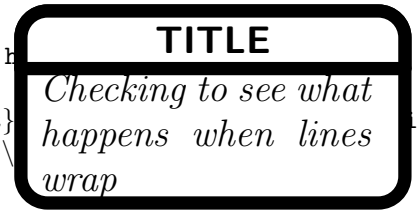


Plainly, this is terrible style; but the principles are accurately shown here.

We can easily change the font separately for both title and body with `rrtitlefont` and , with the usual font-changing commands that we would use in our text. We've also been using rectangles with extremely rounded borders; this reflects the default "border radius" of `roundrect`, which is `40pt`. We can change how rounded our corners are with `rrborderrad`, and how wide our border is with `rrborderwd`:

`rrtitlefont`
`rrtextfont`
`rrborderrad`
`rrborderwd`

```
rrtitlebar := true;
rrborderrad := 20pt;
rrborderwd := 5pt;
rrbodytext := "Checking to see what h
rrtitletext := "TITLE";
rrtitlefont := "\fontsize{14pt}{17pt}
rrtextfont := "\fontsize{14pt}{17pt}\
rrtitleht := 0.3in;
roundrect(1in,2in)(rectangle);
draw rectangle;
```

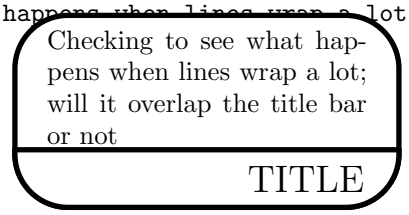


Be sure, if you are calling METAPOST from LaTeX via `gmp` or some similar package, that you wrap any LaTeX commands in an `\unexpanded` environment; otherwise LaTeX will expand them in a way that METAPOST won't be able to process.

We can even tell `roundrect` that we want our title bar on the bottom, with `rrtitleside`, or aligned right or left rather than centered, with `rrtitlepos`:

`rrtitleside`
`rrtitlepos`

```
rrtitlebar := true;
rrbodytext := "Checking to see what happens when lines wrap a lot; will it overlap the t
rrtitletext := "TITLE";
rrtitletext := "TITLE";
rrtitleht := 0.3in;
rrtitleside := "bot";
rrtitlepos := "right";
roundrect(1in,2in)(rectangle);
draw rectangle;
```



At present, the only options for `rrtitleside` are "top" and "bot"; "right" and "left" should be added soon, once the code is refactored to make it easier. The options for `rrtitlepos` are, predictably, "left", "right", and "center".

`rrrestorevals` To restore all these values to the defaults, simply issue the directive `rrrestorevals;`; this macro does nothing but replace the default values in all these parameters.

And those are the `roundrect` macros; I hope they are useful for someone. Happy METAPOSTing!

4 Further Work Needed

This really boils down to just one thing: code refactoring. There's too much duplication here; that's what made it hard to add in different values for `rrtitleside`, where it should have been quite easy. At the moment, I don't really have time, so I just put this together as it was and sent it out; but once I move some other projects off my plate, this will get done. But for those who look at the code here and are disgusted by the unnecessary duplication: I know, and I want to fix it.

5 Implementation

```
input TEX;
color rrinnercolor; rrinnercolor := white;
color rrbordercolor; rrbordercolor := black;
numeric rrborderwd; rrborderwd := 2pt;
numeric rrborderrad; rrborderrad := 40pt;
string rrtitlefont; rrtitlefont := "\fontsize{14pt}{17pt}\selectfont ";
string rrtextfont; rrtextfont := "\fontsize{10pt}{12pt}\selectfont ";
color rrtitlecolor; rrtitlecolor := black;
color rrtextcolor; rrtextcolor := black;
color rrtitlebgcolor; rrtitlebgcolor := white;
color rrtitlebordercolor; rrtitlebordercolor := black;
boolean rrtitlebar; rrtitlebar := false;
string rrtitleside; rrtitleside := "top";
```

```
numeric rrtitleht; rrtitleht := 0pt;
string rrtitlepos; rrtitlepos := "center";
boolean rrtitlebotborder; rrtitlebotborder := true;
color rrtitlebotbordercolor; rrtitlebotbordercolor := black;
string rrtitletext; rrtitletext := "";
string rrbodytext; rrbodytext := "";
string rrbodyalign; rrbodyalign := "justify";
string rrbodyaligncom; rrbodyaligncom := "";
def rrrestorevals =
rrinnercolor := white;
rrbordercolor := black;
rrborderwd := 2pt;
rrborderrad := 40pt;
rrtitlefont := "\fontsize{14pt}{17pt}\selectfont ";
rrtextfont := "\fontsize{10pt}{12pt}\selectfont ";
rrtitlecolor := black;
rrtextcolor := black;
rrtitlebgcolor := white;
rrtitlebordercolor := black;
rrtitlebar := false;
rrtitleside := "top";
rrtitleht := 0pt;
rrtitlepos := "center";
rrtitlebotborder := true;
rrtitlebotbordercolor := black;
rrtitletext := "";
rrbodytext := "";
rrbodyalign := "justify";
rrbodyaligncom := "";
enddef;
def roundrect(expr rrht, rrwd)(suffix name) =
TEXPRE("%&latex" & char(10) & "\documentclass{article}\begin{document}");
TEXPOST("\end{document}");
if (rrtitleht = 0pt):
rrtitleht := rrht/6;
fi
path rra; path rrb; path rrc; path rrd;
pair a; pair b; pair c; pair d;
a := (0,0) shifted (-rrwd/2,-rrht/2);
b := (0,0) shifted (rrwd/2,-rrht/2);
c := (0,0) shifted (rrwd/2,rrht/2);
d := (0,0) shifted (-rrwd/2,rrht/2);
rra := fullcircle scaled rrborderrad shifted (xpart a +
(rrborderrad/2),ypart a + (rrborderrad/2));
rrb := fullcircle scaled rrborderrad shifted (xpart b -
(rrborderrad/2),ypart b + (rrborderrad/2));
rrd := fullcircle scaled rrborderrad shifted (xpart d +
(rrborderrad/2),ypart d - (rrborderrad/2));
rrc := fullcircle scaled rrborderrad shifted (xpart c -
(rrborderrad/2),ypart c - (rrborderrad/2));
```

```
pair f; f := (a--b) intersectionpoint rra;
pair g; g := (a--b) intersectionpoint rrb;
pair h; h := (b--c) intersectionpoint rrb;
pair i; i := (b--c) intersectionpoint rrc;
pair j; j := (c--d) intersectionpoint rrc;
pair k; k := (c--d) intersectionpoint rrd;
pair l; l := (d--a) intersectionpoint rrd;
pair m; m := (d--a) intersectionpoint rra;
picture name;
picture border;
pair n; pair o; path rrtitlepath; path rrfinalline;
name := image(fill f--g{right}..{up}h--i{up}..{left}j--
k{left}..{down}l--m{down}..{right}f--cycle
withcolor rrinnercolor);
border := image(pickup pencircle scaled rrborderwd;
draw f--g{right}..{up}h--i{up}..{left}j--
k{left}..{down}l--m{down}..{right}f--cycle
withcolor rrbordercolor);
addto name also border;
pair rrtitlestation;
picture rrtitlelabel;
if (rrtitlebar):
if (rrtitleside = "top"):
n := (xpart m,ypart k - (rrtitleht));
o := (xpart i,ypart k - (rrtitleht));
defaultscale := 4;
if (ypart n > ypart l):
n := (xpart n,ypart l);
o := (xpart o,ypart l);
fi
rrtitlepath := o--i{up}..{left}j--k{left}..{down}l--n;
if (rrtitlepos = "center"):
rrtitlestation := 0.5[n,o] shifted (0,rrtitleht/2);
elseif (rrtitlepos = "left"):
rrtitlestation := n shifted (rrborderrad/4,rrtitleht/2);
elseif (rrtitlepos = "right"):
rrtitlestation := o shifted (-rrborderrad/4,rrtitleht/2);
fi
elseif (rrtitleside = "bot"):
n := (xpart m,ypart f + (rrtitleht));
o := (xpart i,ypart f + (rrtitleht));
if (ypart n < ypart m):
n := (xpart n,ypart m);
o := (xpart o,ypart m);
fi
rrtitlepath := o--h{down}..{left}g--f{left}..{up}m--n;
if (rrtitlepos = "center"):
rrtitlestation := 0.5[n,o] shifted (0,-rrtitleht/2);
elseif (rrtitlepos = "left"):
rrtitlestation := n shifted (rrborderrad/4,-rrtitleht/2);
```

```
elseif (rrtitlepos = "right"):
rrtitlestation := o shifted (-rrborderrad/4,-rrtitleht/2);
fi
fi
rrfinalline = n--o;
picture rrtitlepic;
picture rrtitlepicb;
picture rrtitlebotborderpic;
rrtitlebotborderpic := image (draw rrfinalline
withcolor rrtitlebotbordercolor);
rrtitlepicb := image(draw rrtitlepath withcolor rrtitlebordercolor);
if (rrtitlebotborder = true):
addto rrtitlepicb also rrtitlebotborderpic;
fi
rrtitlepath := buildcycle(rrtitlepath,rrfinalline);
rrtitlepic := image(fill rrtitlepath withcolor rrtitlebgcolor);
if (rrtitlepos = "center"):
rrtitlelabel := image(label(TEX(rrtitlefont&" "&rrtitletext),
rrtitlestation) withcolor rrtitlecolor);
elseif (rrtitlepos = "left"):
rrtitlelabel := image(label.rt(TEX(rrtitlefont&" "&rrtitletext),
rrtitlestation) withcolor rrtitlecolor);
elseif (rrtitlepos = "right"):
rrtitlelabel := image(label.lft(TEX(rrtitlefont&" "&rrtitletext),
rrtitlestation) withcolor rrtitlecolor);
fi
if (rrbodyalign = "center"):
rrbodyaligncom := "\centering ";
elseif (rrbodyalign = "left"):
rrbodyaligncom := "\flushleft\vskip-\baselineskip ";
elseif (rrbodyalign = "right"):
rrbodyaligncom := "\flushright\vskip-\baselineskip ";
fi
picture rrbodytextpic;
if (rrtitleside = "top"):
rrbodytextpic :=
image(label.lrt(TEX("\parbox{"&decimal(rrwd-2rrborderwd-rrborderrad/2)&"bp}{"&rrbodyaligncom&rr
elseif (rrtitleside = "bot"):
rrbodytextpic :=
image(label.urt(TEX("\parbox{"&decimal(rrwd-2rrborderwd-rrborderrad/2)&"bp}{"&rrbodyaligncom&rr
fi
addto name also rrtitlepic;
addto name also rrtitlepicb;
addto name also rrtitlelabel;
addto name also rrbodytextpic;
fi
enddef;
```