

Entry-level MetaPost 3: Color

Mari Voipio

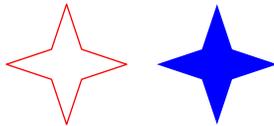
This installment of the entry-level MetaPost series discusses color; previous topics include the basic grid (*TUGboat* 34:1) and image transformations (34:2). For basic information on running MetaPost, either standalone or within a ConTeXt document, see <http://tug.org/metapost/runningmp.html>.

As we have seen in the earlier tutorials, MetaPost recognizes the colors *Red*, *Green* and *Blue* (plus black and white). We have also used color for both outlines and fills. Here's an example star:

```
numeric u; u:= 2mm; % defining the unit
path star; % defining a star shape
star := (3u,3u) -- (0u,4u) -- (3u,5u)
-- (4u,8u) -- (5u,5u) -- (8u,4u)
-- (5u,3u) -- (4u,0u) -- cycle;
```

```
% outlined red star:
draw star withcolor red;
```

```
% filled blue star, a bit to the right:
fill star shifted (10u,0u) withcolor blue;
```



These three basic colors can be modified by “removing” some color, thus letting more black in, or by mixing in another color, e.g. white. The amount of color is given by adding a number between 0 and 1 to the color name. I think about these as percentages; I read `.25red` as “take 25% red and the rest black”.

On the other hand, we can add in a color other than black. For instance, `.25[red,white]` gives a color that is 75% red and 25% white—a quarter “of the way” between red and white, thus closer to red.

Here are some examples:

```
numeric u; u := 5mm;
% all black:
fill unitsquare scaled 1u withcolor 0red;

fill unitsquare scaled 1u shifted (1u,0)
withcolor .25red;
fill unitsquare scaled 1u shifted (2u,0)
withcolor .5red;
fill unitsquare scaled 1u shifted (3u,0)
withcolor .75red;
fill unitsquare scaled 1u shifted (4u,0)
withcolor red; % basic red
fill unitsquare scaled 1u shifted (5u,0)
withcolor .25[red,white];
fill unitsquare scaled 1u shifted (6u,0)
withcolor .5[red,white];
```

```
fill unitsquare scaled 1u shifted (7u,0)
withcolor .75[red,white];
```

```
% all white (not visible):
fill unitsquare scaled 1u shifted (8u,0)
withcolor 1[red,white];
```



(Only eight squares are evident, although nine were specified, because a white square on white paper, or a white background, is “invisible”.)

Once we have found a pleasing shade, we can name it in the same way as a path:

```
% defining a color:
color lightpink; lightpink := .8[red,white];
```

```
% a light pink star:
fill star withcolor lightpink;
```



However, this *is* a star and those are conventionally drawn in yellow. Not red, not green, not blue—yellow. How to achieve that?

The answer lies in RGB values, RGB being the color scheme used by MetaPost. Each RGB color is expressed with three values in the range 0–255, the numbers telling what proportions of red, green and blue to mix to achieve a certain color. E.g., the RGB code for the color “gold” is 255–215–0. To make MetaPost understand the code, the color proportions are given as fractions:

```
% fill star with gold (255–215–0):
fill star withcolor (255/255, 215/255, 0/255);
```



RGB charts are widely available online, so we don't have to guess the color codes (although that can be quite fun). I use “The Other RGB Chart” (<http://www.tayloredmktg.com/rgb>) because it is organized by color and includes color names that I like to use in my code.

```
numeric u; u := 10mm;
% color "dark orchid", 153–50–204
color darkorchid;
darkorchid := (153/255, 50/255, 204/255);
% fill a square:
fill unitsquare scaled 1u withcolor darkorchid;
```



The RGB color scheme is primarily for screens, so what you see is not necessarily exactly what prints. Screens are different, too, so what you see might not be the same as what I see. However, considering that MetaPost is mostly used to draw schematic graphics, the RGB color scheme is adequate for the job.

Bonus: Shading

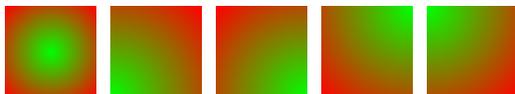
If you use ConTeXt, you can easily use shading from one color to another. The feature utilizes both Metafun and some advanced PDF trickery and thus is only available in ConTeXt. However, there is a cheat that gives MetaPost users limited access to a similar feature, see below.

Shading in ConTeXt. Shading can be either *circular* or *linear*. Circular shading can be either centered (option number 0, the default) or off-centered to one of the corners (options 1–4). Linear shading can be either diagonal (options 1–4) or vertical (options 5 and 7) or horizontal (options 6 and 8). Linear option 0 defaults to option 5, left-to-right shading.

For shading we have to specify what is shaded, how it is shaded (the option number) and the colors to shade from and to. Here are examples of all the circular shading options:

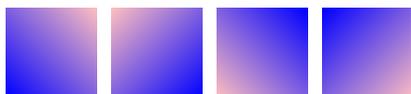
```
numeric u; u := 12mm;
path sq; sq := unitsquare scaled 1u;

% Circular shading (Metafun):
circular_shade (sq, 0, green, red);
circular_shade (sq shifted ((1u+2mm),0), 1,
  green, red);
circular_shade (sq shifted ((2u+2*2mm),0), 2,
  green, red);
circular_shade (sq shifted ((3u+3*2mm),0), 3,
  green, red);
circular_shade (sq shifted ((4u+4*2mm),0), 4,
  green, red);
```



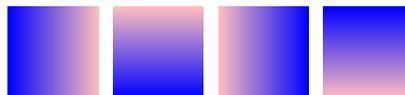
Diagonal linear shading (1–4) examples:

```
% Diagonal linear shading:
linear_shade (sq, 1, blue, .75[red,white]);
linear_shade (sq shifted ((1u+2mm),0), 2,
  blue, .75[red,white]);
linear_shade (sq shifted ((2u+2*2mm),0), 3,
  blue, .75[red,white]);
linear_shade (sq shifted ((3u+3*2mm),0), 4,
  blue, .75[red,white]);
```



Vertical/horizontal linear shading (5–8) examples:

```
% Vertical/horizontal linear shading:
linear_shade (sq, 5, blue, .75[red,white]);
linear_shade (sq shifted ((1u+2mm),0), 6,
  blue, .75[red,white]);
linear_shade (sq shifted ((2u+2*2mm),0), 7,
  blue, .75[red,white]);
linear_shade (sq shifted ((3u+3*2mm),0), 8,
  blue, .75[red,white]);
```



One of my first MetaPost exercises was a graphic that imitates a line drawn with pen and ink; I needed such a line to create a document that looks like a 17th century manuscript. As I use ConTeXt (which incorporates Metafun), I didn't realize at the time that I was using a very special feature—I was happy enough to have finally found something that did the job (although getting the line *where* I wanted it wasn't that easy...).

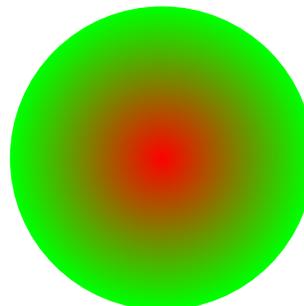
```
path cgl; % defining the line
cgl := (2mm,0) -- (70mm,0) -- (68mm,1mm)
  -- (0,2mm) -- cycle;
```

```
% filling the line with shading:
linear_shade(cgl, 0,
  black, 0.7[black,white]);
```



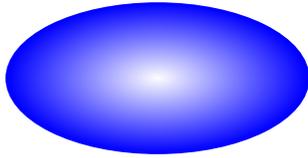
Shading in plain MetaPost. In plain MetaPost we can achieve a shaded circle by drawing multiple concentric rings, changing color at each step. For this we use a loop. Here is a complete example:

```
outputtemplate := "%j-%c.mps";
beginfig (1);
for i = 1 step 1 until 100 :
  a := i/100;
  draw fullcircle scaled (i*.4mm)
    withcolor a[red,green]
    withpen pencircle scaled .4mm;
endfor;
endfig;
end.
```



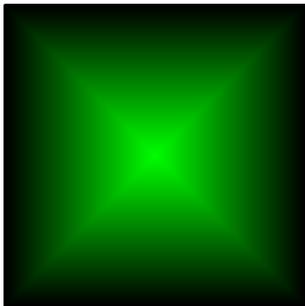
With Metafun, we can add `xscaled` or `yscaled` to create a shaded ellipse:

```
for i = 1 step 1 until 100 :
  a := i/100;
  draw fullcircle scaled (i*.2mm)
    withcolor a[white,blue]
    withpen pencircle scaled .2mm;
endfor;
currentpicture := currentpicture xscaled 2;
```



Also with Metafun, a shaded square/rectangle is possible:

```
for i = 1 step 1 until 100 :
  a := i/100;
  draw fullsquare scaled (i*.4mm)
    withcolor a[green,black]
    withpen pencircle scaled .4mm;
endfor;
```



This approach only works with PostScript output (`.mps/.eps/.ps`), not with SVG output.

Filling and drawing at the same time

We can also both draw and fill a shape. However, we need to fill first and then draw. It is a bit counterintuitive at first; I think of this as coloring the shape, then inking the outline on top.

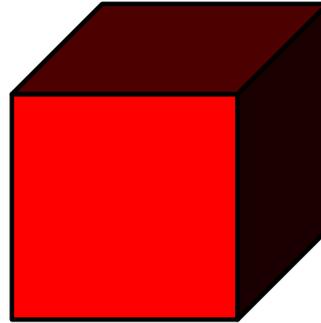
When I was a little girl, my father entertained me by teaching me to draw 3D shapes, including a cube, so we'll make that our example.

```
% draw 3d cube
numeric u; u := 6mm;
path cfront; cfront := (0,0) -- (5u,0)
  -- (5u,5u) -- (0,5u) -- cycle;
path ctop; ctop := (0,5u) -- (5u,5u)
  -- (7u,7u) -- (2u,7u) -- cycle;
path cside; cside := (5u,0) -- (7u,2u)
  -- (7u,7u) -- (5u,5u) -- cycle;
```

```
fill cfront withcolor red;
draw cfront withpen pencircle scaled .1u;
% (black is the default color)
```

```
fill ctop withcolor .3red;
draw ctop withpen pencircle scaled .1u;
```

```
fill cside withcolor .1red;
draw cside withpen pencircle scaled .1u;
```



Credits

My thanks to Hans Hagen for the plain MetaPost shading trick and for patiently answering my questions about color in MetaPost.

References

Running MetaPost and Metafun:

<http://tug.org/metapost/runningmp.html>

MetaPost manual:

<http://tug.org/docs/metapost/mpman.pdf>

MetaFun manual:

<http://www.pragma-ade.com/general/manuals/metafun-p.pdf>

The Other RGB Color Chart:

<http://www.tayloredmktg.com/rgb/>

◇ Mari Voipio

mari dot voipio (at) lucet dot fi

<http://www.lucet.fi>

[Editor's note: This was not a submitted presentation at the TUG'13 meeting; it is included in this issue as a bonus article.]