

Column: Cool Tool

Figures:

File for Subscriber Downloads: 04COOL.ZIP

A New Spin on Thermometer Bars

Whil Hentzen

We've all seen thermometer bars of various shapes and sizes – but they all have two things in common. They all take a parameter whose value changes as a process progresses, and they all demonstrate progress by displaying a horizontal (or, occasionally, vertical) bar advancing across the screen in a horizontal window.

How last century.

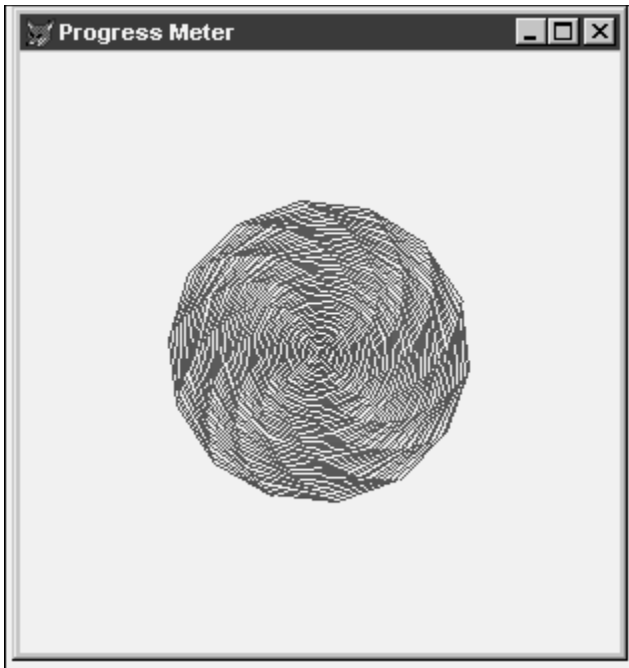
Todd Sherman of Compass Resources put together a class that performs the same function, but instead of the horizontal bar, the visual appearance mimics the aperture of a camera lens, starting with a dot in the center of the thermometer bar form, and opening into a circle whose diameter is just a hair narrower than the width (or height) of the containing form.

Yeah, that's a lot of words. How about some pictures, like in Figure 1, Figure 2 and Figure 3.



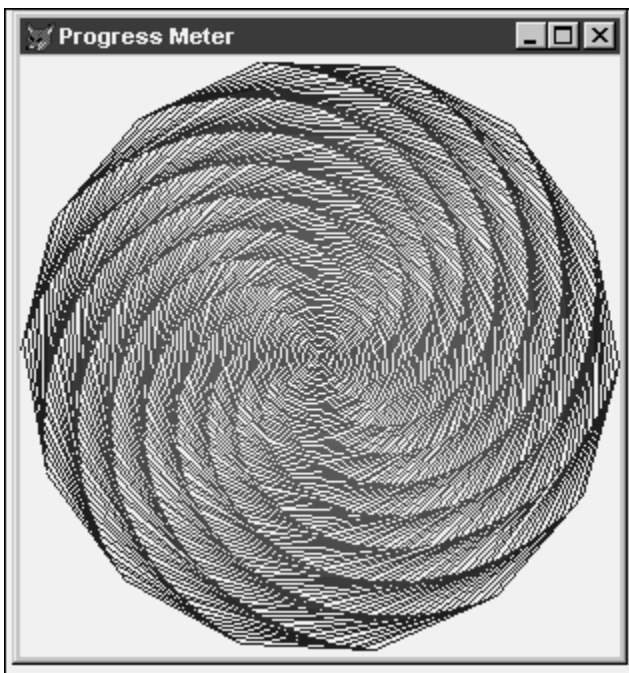
CT-01.TIF

Figure 1. Todd Sherman's Spiral Thermometer starts with a dot in the center of the form.



CT-02.TIF

Figure 2. Todd Sherman's Spiral Thermometer about one-third of the way through a process.



CT-03.TIF

Figure 3. When processing is complete, the Spiral Thermometer displays a spiral that fills the form. It looks even better in person, since the spiral is drawn in multiple colors.

The class consists of a few properties and a method that draws a line – as you'll see if you examine the figures, the spiral is really just a series of lines drawn around a central point. The line drawing method is called with a parameter that you can use to mark the progress of the process you're running. Here's the guts of the test program Todd provides to show how to call the class.

```

set proc to spiral_therm
max_x = 1000
oMeter = createobject("spiral_therm")
oMeter.show
for x= 1 to max_x - 1
  oMeter.SetSize(x/max_x)
next

```

The class itself is a bit more involved, and you may want to get your high school trig book out if you want to hack the code at all.

```

* see test_spiral.prg for usage
define class spiral_therm as form
autocenter = .t.
width = 300
height = 300
caption = 'Progress Meter'
backcolor = rgb(255, 255, 255)
name = 'form1'
* properties for the spiral itself
radius = 0
rot_progress = 500000
rot_step = 57
radius_inc = .1
* this method is the guts of the class
procedure setsize
parameter sizemult
* sizemult goes from 0.001 to .999
* do while the radius of the circle being drawn is <
* half the width of the form * the multiplier
do while thisform.radius < thisform.width/2 * sizemult
* adjust the values that make up the color of
* the line
* 0 to 255, 255 to 0, bell curve 0 to 63.75 to 0
goodblue = ;
  0+(255*(thisform.radius/(thisform.width/2)))
goodred = ;
  255-(255*(thisform.radius/(thisform.width/2)))
goodgreen = ;
  (thisform.radius/(thisform.width/2))*goodred
thisform.forecolor=rgb(goodred, goodgreen, goodblue)
* rotation progress goes from 0 to 85,500
thisform.rot_progress = thisform.rot_progress ;
  + thisform.rot_step
* create the start and end points of each line
* being drawn around the circle
startx = sin(thisform.rot_progress) * thisform.radius ;
  + thisform.width/2
starty = cos(thisform.rot_progress) * thisform.radius ;
  + thisform.width/2
endx = sin(thisform.rot_progress + thisform.rot_step) * ;
  thisform.radius + thisform.width/2
endy = cos(thisform.rot_progress + thisform.rot_step) * ;
  thisform.radius + thisform.width/2
thisform.line(startx, starty, endx, endy)
thisform.radius = thisform.radius + thisform.radius_inc
enddo
if sizemult = 1 then
  thisform.release
endif
endproc
enddefine

define class spirallabel as label
extracaption = "Complete"
enddefine

```

As with many of the Cool Tools featured in FoxTalk, this one is provided with source code, but “as is” – you may want to modify or enhance it further. I can think of a number of features you might want to tweak.

First, you might want to remove the title bar of the form, and change the backcolor of the form to match the application you’re working on. If you’re mathematically inclined, you might try tweaking the code so that the class draws from the outside to the center, giving the impression of the aperture lens closing – particularly appropriate if the process you’re metering has to do with the closing of a month end, or signifies the termination of some other event.

If you want to bother Todd, or tell him thanks, or send him a copy of your enhancements, or send him a box of Frangos, his email is Todd Sherman [sherman@compassres.com]. Ummm, on second thought, forget what I said about bothering him, OK?

Whil Hentzen is editor of FoxTalk.