MaxGUI Tutorial: Scrolling image on Canvas

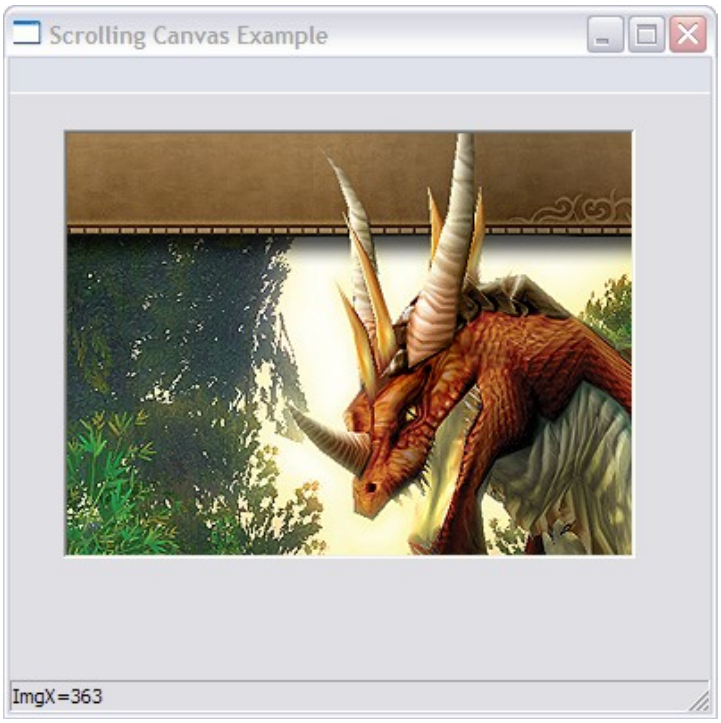
We need to understand 2 concepts if we want to understand how to implement scrolling images on canvases.

Concept 1: Scrolling using **SetGadgetShape**

Cut and Paste the program below and run it. We also need an image to load into this example.

```
SuperStrict

Local Pix:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\dragon-800x.jpg")
Local MyWindow:TGadget=CreateWindow("Scrolling Canvas Example",  100, 10,
400,400)
Local MyPanel:TGadget=CreatePanel(30, 20, 320, 240, MyWindow,
PANEL_ACTIVE|PANEL_BORDER)
Local MyCanvas:TGadget=CreateCanvas(0, 0, 800,600, MyPanel)
SetGraphics CanvasGraphics(MyCanvas)
DrawPixmap Pix,0,0

Local ImgX:Int=0
Local ImgY:Int=0
Local Timer:TTimer=CreateTimer(60)

Repeat
  WaitEvent()
  Select EventID()
  Case EVENT_WINDOWCLOSE
     End

  Case EVENT_TIMERTICK
     imgX=imgX+1
     If ImgX>=GadgetWidth(MyCanvas) Then ImgX=0
     RedrawGadget(MyCanvas)

  Case EVENT_GADGETPAINT
    SetGraphics CanvasGraphics (MyCanvas)
    SetGadgetShape MyCanvas,-ImgX,
-ImgY,GadgetWidth(MyCanvas),GadgetHeight(MyCanvas)
     Flip
   End Select
   SetStatusText MyWindow, "ImgX="+ImgX
Forever
```

We should now be able to see an image scrolling across our panel.



Now let us go through the code line by line.

First we need to load the image

```
Local Pix:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\dragon-800x.jpg")
```

Then we need to create a window, a panel and a canvas. The Canvas needs to be grouped to the panel.
The job of the panel is to clipped the image on the canvas to within its frame. Note that the panel is smaller than the canvas.

```
Local MyWindow:TGadget=CreateWindow("Scrolling Canvas Example",  100, 10, 400,400)
Local MyPanel:TGadget=CreatePanel(30, 20, 320, 240, MyWindow, PANEL_ACTIVE|PANEL_BORDER)
Local MyCanvas:TGadget=CreateCanvas(0, 0, 800, 600, MyPanel)
```

We then draw our loaded image onto the canvas

```
SetGraphics CanvasGraphics(MyCanvas)
DrawPixmap Pix,0,0
```

We also need a timer for the animation and location variables as well

```
Local ImgX:Int=0
Local ImgY:Int=0
Local Timer:TTimer=CreateTimer(60)
```

When the timer event happens, we update our image X location. Also we need to check that if it runs over the edge we reset the location back to the start.

```
    Case EVENT_TIMERTICK
      imgX=imgX+1
      If ImgX>=GadgetWidth(MyCanvas) Then ImgX=0
      RedrawGadget(MyCanvas)
```

The Paint Canvas Event will be generated when we call the above **RedrawGadget** function, The **SetGraphics** and **Flip** statements do the usual stuff, ie displays our canvas drawing to the screen.

```
    Case EVENT_GADGETPAINT
      SetGraphics CanvasGraphics (MyCanvas)
      SetGadgetShape MyCanvas,-ImgX,
  -ImgY,GadgetWidth(MyCanvas),GadgetHeight(MyCanvas)
      Flip
    End Select
```

The key function here is the **SetGadgetShape** function. The syntax of this function is as follows:-

```
  Function SetGadgetShape( gadget:TGadget,x,y,w,h )
```

According to the manual, this function "Sets a gadget's position and size **relative** to it's group". The group in our example being the panel gadget.

The key word here is **relative** i.e position relative to the panel. Notice how we have put in a negative sign infront of the **ImgX** variable. The width and the height of the gadget remains the same, we only need to update the X and Y values to scroll our image. Simple, once we get the concept :)

Now that we know how to scroll an image on a canvas using the **SetGadgetShape** function, the next concept we need to understand is how to use the slider gadget to give us the correct **ImgX** and **ImgY** variable to scroll our image.

Concept 2:Using Scrollbars to set the image location

Our program is now slightly longer as we now need to add code to handle the scrollbars.

```
  SuperStrict

  Local Pix:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\dragon-800x.jpg")
  Local MyWindow:TGadget=CreateWindow("Scrolling Canvas Example",  100, 10, 400,400)
  Local MyPanel:TGadget=CreatePanel(30, 20, 320, 240, MyWindow,
  PANEL_ACTIVE|PANEL_BORDER)

  Local CanvasHeight:int=600
  Local CanvasWidth:int=800
  Local MyCanvas:TGadget=CreateCanvas(0, 0, CanvasWidth, CanvasHeight, MyPanel)
  SetGraphics CanvasGraphics(MyCanvas)
  DrawPixmap Pix,0,0

  Local ImgX:Int=0
  Local ImgY:Int=0

  Local VScroller:TGadget=CreateSlider(GadgetWidth(MyPanel)+30, 20, 20,
  GadgetHeight(MyPanel), MyWindow, SLIDER_VERTICAL)
  Local HScroller:TGadget=CreateSlider(30,GadgetHeight(MyPanel)+20,
  GadgetWidth(MyPanel), 20, MyWindow, SLIDER_HORIZONTAL)
```

```
SetSliderRange VScroller, GadgetHeight(MyPanel), CanvasHeight
SetSliderRange HScroller, GadgetWidth(MyPanel), CanvasWidth


Repeat
 WaitEvent()
 Select EventID()
 Case EVENT_WINDOWCLOSE
   End

 Case EVENT_GADGETACTION
   Select EventSource()
     Case HScroller
      ImgX=EventData()
     Case VScroller
       ImgY=EventData()
     End Select
   RedrawGadget(MyCanvas)

 Case EVENT_GADGETPAINT
   SetGraphics CanvasGraphics (MyCanvas)
   SetGadgetShape MyCanvas,-ImgX,
-ImgY,GadgetWidth(MyCanvas),GadgetHeight(MyCanvas)
   Flip
  End Select
  SetStatusText MyWindow, "ImgX="+ImgX+", ImgY="+ImgY
Forever
```

Now we see our image within a scrolling panel. Clicking, dragging the scroll bar will produce the expected effect.

The main changes from the example above are the creations of the Slider gadgets as can be seen below.

```
Local VScroller:TGadget=CreateSlider(GadgetWidth(MyPanel)+30, 20, 20,
GadgetHeight(MyPanel), MyWindow, SLIDER_VERTICAL)
Local HScroller:TGadget=CreateSlider(30,GadgetHeight(MyPanel)+20,
```

```
GadgetWidth(MyPanel), 20, MyWindow, SLIDER_HORIZONTAL)
```

We also need to set the slider range, this is important as the right slider range will give us ranges of slider values suitable for our use.

```
SetSliderRange VScroller, GadgetHeight(MyPanel), CanvasHeight
SetSliderRange HScroller, GadgetWidth(MyPanel), CanvasWidth
```

The syntax for SetSliderRange from the manual is as below:-

```
Function SetSliderRange(slider:TGadget,range0,range1)
```

I actually prefer the Blitzplus description, where range0 actually is the visible panel, hence we use the panel width and height and range1 is the total size of the canvas (which holds our image)

```
Function SetSliderRange(slider:TGadget,Visible Range,Total Range)
```

We then capture the gadget events and get the slider values from the eventdata (you can also use the **SliderValue** function instead of **EventData**() function)

```
Case EVENT_GADGETACTION
  Select EventSource()
    Case HScroller
      ImgX=EventData()
    Case VScroller
      ImgY=EventData()
    End Select
  RedrawGadget(MyCanvas)
```

The gadgetpaint event statements are the same as before

```
Case EVENT_GADGETPAINT
  SetGraphics CanvasGraphics (MyCanvas)
  SetGadgetShape MyCanvas,-ImgX,
-ImgY,GadgetWidth(MyCanvas),GadgetHeight(MyCanvas)
  Flip
 End Select
```

I hope it is clear that the sliders are nothing more than a way for the user to enter the ImgX and ImgY values for our **SetGadgetShape** function to do the hard-work of scrolling the canvas.

As you can see from this tutorial, you do not need all the esotoric hooks and OOP stuff to scroll images on canvases.

They are good concepts to learn but a step at a time. The bad news is that this technique only work for images no bigger than the maximum size of a canvas. Currently MaxGUI limits canvases to the maximum size of the desktop.

If you want to learn more about Canvas and Scrollers, you can take a look at my MaxGUI beginner tutorials, specifically Tutorial 12: Canvas, Events and 2D Graphics and Tutorial 13:

Sliders and Scrollbars.