The MaxGUI Beginner Tutorial Series - Tutorial 5: Panel and Mouse Events
(c) Assari Dec 23 2005

In the previous tutorial we covered panels and the various styles except for the PANEL_ACTIVE style. In this tutorial we are going to cover ACTIVE_PANEL and introduce ourselves to mouse events.

To ensure this program runs as shown with the proper images, download the 3 images below to your hard-disk and then specify the correct location in the code below.

Let's first create a panel with an image background to simulate a button image

```
SuperStrict

Local MyWindow:TGadget=CreateWindow("Panel Example", 200,200,320,240)
Local MyPanel:TGadget=CreatePanel(110,30,80,80, MyWindow)
Local image:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials
\Button_1.jpg")

Repeat
  WaitEvent()
  Select EventID()
  Case EVENT_WINDOWCLOSE
    End
  End Select
  SetPanelPixmap MyPanel, image, PANELPIXMAP_CENTER
Forever
```

We now have a button in our window but it does not do anything for the moment

Let's put some more bells and whistle into our code (items in bold), it will look a bit hairy but bear with me and I will explain the code line by line later.

```
SuperStrict

Local MyWindow:TGadget=CreateWindow("Panel Example", 200,200,320,240)
Local MyPanel:TGadget=CreatePanel(110,30,80,80, MyWindow, PANEL_ACTIVE)
```

```
Local image1:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\Button_1.jpg")
Local image2:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials
\Button_2.jpg")
Local image:TPixmap = image1

Repeat
  WaitEvent()
  Select EventID()
  Case EVENT_WINDOWCLOSE
    End
  Case EVENT_MOUSEENTER
    image=image2
  Case EVENT_MOUSELEAVE
    image=image1
  End Select
  SetPanelPixmap MyPanel, image, PANELPIXMAP_CENTER
Forever
```

Now if you were to execute the above code and move the mouse in and out of the button you will see the rollover effect

Now let us see how this effect was achieved.

```
SuperStrict

Local MyWindow:TGadget=CreateWindow("Panel Example", 200,200,320,240)
Local MyPanel:TGadget=CreatePanel(110,30,80,80, MyWindow, PANEL_ACTIVE)
```

Hopefully by now you are familiar with the first two lines of code above.

The 3rd line is the one of interest specifically the PANEL_ACTIVE flag which must be set. The rest are the normally required **CreatePanel** parameters.

```
Local image1:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\Button_1.jpg")
Local image2:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\Button_2.jpg")
```

The two lines above loads the image files that we want to use as our buttons. This is done using the Blitzmax function **LoadPixmap**. This function will store our images as Pixmap objects.

The LoadPixmap function require us to specify where the image files are stored on the hard-disk so we need to be very specific here.

```
Local image:TPixmap = image1
```

The above line define a variable called **image** of type Pixmap to allow us to use in our **SetPanelPixmap** function later.

```
Select EventID()
Case EVENT_WINDOWCLOSE
  End
Case EVENT_MOUSEENTER
  image=image2
Case EVENT_MOUSELEAVE
  image=image1
End Select
```

Our Select EventID() statement now selects for 2 types of mouse events, EVENT_MOUSEENTER and EVENT_MOUSELEAVE.

These events are emitted by MaxGUI when the mouse enters or leave our panel. The case statements then checks for which event has occured and assign the relevant button imagery to our **image** variable.

And finally the **SetPanelPixmap** function displays the image on our panel. This produces the rollover effect.

```
SetPanelPixmap MyPanel, image, PANELPIXMAP_CENTER
```

Note that you will need to use the PANELPIXMAP_CENTER style rather than the other styles to achieve the desired look.

Now to complete this tutorial let us add some more code (see bold items) to achieve the effect of mouse-clicks.

```
SuperStrict

Local MyWindow:TGadget=CreateWindow("Panel Example", 200,200,320,240)
Local MyPanel:TGadget=CreatePanel(110,30,80,80, MyWindow, PANEL_ACTIVE)
Local image1:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\Button_1.jpg")
Local image2:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials\Button_2.jpg")
Local image3:TPixmap=LoadPixmap("D:\My Documents on E\_Tutorials
\Button_3.jpg")
Local image:TPixmap = image1

Repeat
  WaitEvent()
  Select EventID()
  Case EVENT_WINDOWCLOSE
    End
  Case EVENT_MOUSEENTER, EVENT_MOUSEUP
    image=image2
```

```
      Case EVENT_MOUSELEAVE
        image=image1
      Case EVENT_MOUSEDOWN
        image=image3
      End Select
      SetPanelPixmap MyPanel, image, PANELPIXMAP_CENTER
    Forever
```

Cut and Paste the above program, run it and go ahead and click on the button. You should now see the mouse click effect on the button. In the code above you will notice that we have added a third image (of the clicked button) to our set.

We then added the checks for the MOUSEUP and MOUSEDOWN events and assign the relevant image to our image variable and voila we have our desired effects.

The PANEL_ACTIVE flag in our **CreatePanel** function allows our panel to detect the following events

EVENT_MOUSEDOWN	Mouse button pressed. Event data contains mouse button code
EVENT_MOUSEUP	Mouse button released. Event data contains mouse button code
EVENT_MOUSEMOVE	Mouse moved. Event x and y contain mouse coordinates
EVENT_MOUSEWHEEL	Mouse wheel spun. Event data contains delta clicks
EVENT_MOUSEENTER	Mouse entered gadget area
EVENT_MOUSELEAVE	Mouse left gadget area
EVENT_KEYDOWN	Key pressed. Event data contains keycode
EVENT_KEYUP	Key released. Event data contains keycode
EVENT_KEYCHAR	Key character. Event data contains unicode value

For this tutorial we have only covered the mouse down, mouse up, mouse leave and mouse enter events. The others we will cover in due course in future tutorials.

Summary

To re-cap what we have learnt in this tutorial

- Setting the PANEL_ACTIVE flag allows MaxGUI to monitor events generated by our panel.
- Together with the SetPanelPixmap functions and Blitzmax's LoadPixmap function, we can use image buttons in our MaxGUI programs, functionality missing from the MaxGUI button repertoire.