
Subject: Re: Interfacing the functions

Posted by [idlfreak](#) on Tue, 09 Apr 2002 17:50:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

David Fanning <david@dfanning.com> wrote in message
news:<MPG.171b68cfa798863b989874@news.frii.com>...

> Jaco van Gorkom (j.c.van.gorkom@fz-juelich.de) writes:

>

>> We are all eagerly awaiting that object book, you know...

>

> Odd that you should mention this. I was just on the phone
> with someone who very likely will co-author this book with
> me. We were discussing who will "write" the book and who
> will "edit" it. Unfortunately, we are both editors. :-(

>

> I also heard from Ronn Kling this morning. His object
> book is rapidly nearly completion and should be available
> soon.

>

> And I know Mark is all over these ideas. I'm sure we
> can expect something from him real soon now. :-)

>

> Cheers,

>

> David

Dr.Fanning,

I've wrote some code to perform zoom and pan as per what i understood
from ur explanation. But i couldn't understand clearly the method of
linking the event handlers. I get some errors. Can u please look at it
and tell me if i'm proceeding in the right way.

Thank you so much for ur time.

Cheers,
Akhila.

PRO Button_Event, event

```
;IF event.select NE 1 THEN RETURN  
Widget_Control, event.top, Get_UValue=info, /No_Copy  
Widget_Control, event.id, Get_Value=buttonValue  
info.currentMode = buttonValue  
Widget_Control, event.top, Set_UValue=info, /No_Copy
```

END

PRO Draw_Widget_Event, event

```
Widget_Control, event.top, Get_UValue=info, /No_Copy
CASE info.currentMode OF
  'zoom': zoom_event, event
  'pan': pan_event, event
  'adjust': adjust_event, event
  'quit': quit_event, event
ENDCASE
Widget_Control, event.top, Set_UValue=info, /No_Copy
END
```

PRO zoom_event, event

Widget_Control, event.top, Get_UValue=info

```
possibleEvents = ['DOWN', 'UP', 'MOTION', 'SCROLL', 'EXPOSE']
possibleButtons = ['NONE', 'LEFT', 'MIDDLE', 'NONE', 'RIGHT']
thisEvent = possibleEvents(event.type)
thisButton = possibleButtons(event.press)
```

CASE thisEvent OF

'EXPOSE': info.oWindow->Draw, info.oView

'DOWN': BEGIN

print, 'MOUSE DOWN STATE'

CASE thisButton OF

'LEFT': BEGIN

info.oModel->Scale, 0.95, 0.95, 1.00

info.oWindow->Draw, info.oView

END

'RIGHT': BEGIN

info.oModel->Scale, 1.05, 1.05, 1.00

info.oWindow->Draw, info.oView

END

ELSE:BEGIN

END

ENDCASE

END

```
'UP': BEGIN
print, 'MOUSE UP STATE'
END
```

ELSE:

ENDCASE

```
Widget_Control, event.top, Set_UValue=info
END
```

PRO pan_event, event

```
Widget_Control, event.top, Get_UValue=info
```

```
possibleEvents = ['DOWN', 'UP', 'MOTION', 'SCROLL', 'EXPOSE']
possibleButtons = ['NONE', 'LEFT', 'MIDDLE', 'NONE', 'RIGHT']
thisEvent = possibleEvents(event.type)
thisButton = possibleButtons(event.press)
```

CASE thisEvent OF

```
'EXPOSE': info.oWindow->Draw, info.oView
```

```
'MOTION': BEGIN
print, 'MOUSE MOVE STATE'
    deltax = (event.x - info.xstart) / Float(info.xsize)
    deltay = (event.y - info.ystart) / Float(info.ysize)
    info.xstart = event.x
    info.ystart = event.y
```

```
    info.oModel->Translate, deltax, deltay, 0
    info.oWindow->Draw, info.oView
```

END

ELSE:

ENDCASE

```
Widget_Control, event.top, Set_UValue=info
END
```

PRO Button

```
filename = FilePath(SubDirectory = ['examples', 'data'],
'worldelv.dat')
image = BytArr(360,360)
OpenR, lun, filename, /Get_Lun
ReadU, lun, image
Free_Lun, lun

tlb = Widget_Base(/COLUMN)
menubase = Widget_Base(tlb, /ROW)
ZoomButton = Widget_Button(menubase, VALUE = 'ZOOM', UVALUE = 'zoom')
PanButton = Widget_Button(menubase, VALUE = 'PAN', UVALUE = 'pan')
AdjustButton = Widget_Button(menubase, VALUE = 'ADJUST', UVALUE =
'adjust')
QuitButton = Widget_Button(menubase, VALUE = 'QUIT', UVALUE = 'quit')
drawID = Widget_Draw(tlb, /BUTTON_EVENTS, /EXPOSE_EVENTS, RETAIN = 0,
GRAPHICS_LEVEL = 2, $
    XSIZE = 512, YSIZE = 512)

Widget_Control, tlb, /REALIZE
Widget_Control, drawID, Get_Value = oWindow

oView = Obj_New('IDLgrView', VIEWPLANE_RECT = [0,0,512,512], COLOR =
[0,0,0])
oImage = Obj_New('IDLgrImage', image)
oModel = Obj_New('IDLgrModel', NAME = 'IMAGE', SELECT_TARGET = 1)
oModel -> Add, oImage
oView -> Add, oModel
oWindow -> Draw, oView

info = {currentMode: 0, $
oWindow:oWindow, $
oView:oView, $
oModel:oModel, $
xstart:0, $
ystart:0, $
xsize:512, $
ysize:512}

Widget_Control, tlb, SET_UVALUE = info
XMANAGER, 'Button', tlb

END
```
