
Subject: draw window mouse events

Posted by [Rick Towler](#) on Thu, 27 Apr 2000 07:00:00 GMT

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

I have been trying to implement middle and third button events in my draw widget in much the same way as in the IDL demo d_objworld2.pro. I have not found it easy to understand how everything happens in this demo program.

I am interested in getting one of these buttons to utilize the TRANSLATE feature of the trackball object so I can translate my models. I did have the left button transforming the model but I have broken that in my quest to understand this whole process which obviously I don't since I can't get it working again.

I have the events getting into the correct handling loops. That is, clicks and motion trigger print statements in my loops, but I can't seem to get the updates from the trackball object. Every call to oTrack->update returns EQ 0.

Hints and/or pointers to examples that are a wee bit simpler than d_objworld2 are much appreciated.

Thanks.

-Rick Towler

Some code:

in my main program I define my trackball object "oTrack" and stick it in my main structure "state".

```
oTrack = OBJ_NEW('Trackball', [xdim/2.0, ydim/2.0], xdim/2.0)
```

The window I am interested in:

```
wDraw = widget_draw(wRightBase, GRAPHICS_LEVEL=2, $
  XSIZE=xdim, YSIZE=ydim, /BUTTON_EVENTS, $
  UVALUE='DRAW', RETAIN=0, /EXPOSE_EVENTS, $
  EVENT_PRO = 'driftanim_drawevent')
```

This is my main window event routine as referenced in EVENT_PRO:

```
PRO driftanim_drawevent, ev
```

```
WIDGET_CONTROL, ev.top, GET_UVALUE=state, /NO_COPY
```

```
possibleEvents = ['DOWN', 'UP', 'MOTION', 'SCROLL', 'EXPOSE']
```

```
possibleButtons = ['NONE', 'LEFT', 'MIDDLE', 'NONE', 'RIGHT']
```

```
thisEvent = possibleEvents(ev.type)
```

```
thisButton = possibleButtons(ev.press)
```

```
bHaveTransform = state.oTrack -> Update(ev, TRANSFORM=qmat)
```

```
;bHaveTransform2 = state.oTrack2->Update(ev, TRANSFORM=qmat2)
```

```
CASE thisEvent OF
```

```
'EXPOSE':state.oDWindow->draw, state.oView
```

```
'DOWN': BEGIN
```

```
  ;update mouse
```

```
  ;bHaveTransform = state.oTrack -> Update(ev, TRANSFORM=qmat)
```

```
  ;bHaveTransform2 = state.oTrack2->Update(ev, TRANSFORM=qmat)
```

```
  print,thisButton
```

```
  print,bHaveTransform
```

```
  IF (bHaveTransform) THEN BEGIN
```

```
    state.oModelTop -> GetProperty, TRANSFORM=t
```

```
    state.oModelTop -> SetProperty, TRANSFORM= t # qmat
```

```
  ENDIF
```

```
CASE thisButton OF
```

```
  'RIGHT': BEGIN
```

```
    ;Press with right or middle button
```

```
    print,'Right case'
```

```
    IF (bHaveTransform) THEN BEGIN
```

```
      state.oModelTop -> GetProperty, TRANSFORM=t
```

```
      state.oModelTop -> SetProperty, TRANSFORM= t # qmat
```

```
    ENDIF
```

```
    state.btndown = 1
```

```
    state.oDWindow->SetProperty, QUALITY=state.dragg
```

```
    WIDGET_CONTROL, state.wDraw, /DRAW_MOTION
```

```
  END
```

```
  'MIDDLE': BEGIN
```

```
    print,'Middle case'
```

```
    ;Middle button down
```

```
    state.btndown = 1
```

```
    state.oDWindow->SetProperty, QUALITY=state.dragg
```

```
    WIDGET_CONTROL, state.wDraw, /DRAW_MOTION
```

```
  END
```

```
'LEFT': BEGIN
  print,'Left case'
  ;Left button down
  state.btndown = 1
  state.oDWindow->SetProperty, QUALITY=state.dragg
  WIDGET_CONTROL, state.wDraw, /DRAW_MOTION
END
ENDCASE
END
'MOTION': BEGIN
  print,'Motion with btndown='+string(state.btndown)
  IF (state.btndown) AND (bHaveTransform) THEN $
state.oDWindow->Draw,state.oView
END
'UP': BEGIN
  IF (state.btndown) THEN BEGIN
    ;Release Button
    state.btndown = 0
    state.oDWindow->SetProperty, QUALITY=2
    WIDGET_CONTROL, ev.top, /HOURGLASS
    state.oDWindow->Draw, state.oView
    WIDGET_CONTROL, state.wDraw, DRAW_MOTION=0
  ENDIF
END
ENDCASE

done: WIDGET_CONTROL, ev.top, SET_UVALUE=state, /NO_COPY

END
```
