Subject: Re: The proper way of catching mouse button events from a draw widget?
Posted by Allan Whiteford on Wed, 18 Oct 2006 13:26:06 GMT
View Forum Message <> Reply to Message

Braedley,

I don't think the event driven model is setup to allow the same type of programming. I'd guess you'd need to maintain status elsewhere that you're expecting the next two events to be button clicks on the graph. You could have your draw widget ignore any button clicks made when this switch isn't set. This probably replicates what you're doing at the moment but is much less convenient to code than just an inline call to cursor when you know someone is about to click.

I don't think you can return to the proper point in the event handler. So half your code will need to be written before you're expecting the button clicks, then get the clicks and store them somewhere and finally write the last half of your code after you have both clicks. Something like:

```
pro allan_e,event
 widget_control,event.handler,get_uvalue=info

        if event.id eq info.button then begin
         info.x1=!values.f_nan
         info.x2=!values.f_nan
              info.selecting=1
  widget_control,event.handler,set_uvalue=info
              return
        endif

        if event.id eq info.plotwindow $
           and event.press eq 1 $
           and info.selecting eq 1 then begin
         widget_control,info.plotwindow,get_value=wid
              wset,wid
         !x=info.x_store
              !y=info.y_store
              xval=(convert_coord(event.x,event.y,/device,/to_data))[0]
              plots,[xval,xval],!y.crange

              if not finite(info.x1) then  $
               info.x1=xval   $
  else if not finite(info.x2) then $
               info.x2=xval

         widget_control,event.handler,set_uvalue=info
        endif
```

```
 if finite(info.x1) and finite(info.x2) then begin
         print,info.x1,info.x2
               info.selecting=0
         info.x1=!values.f_nan
         info.x2=!values.f_nan
  widget_control,event.handler,set_uvalue=info
        endif

end

pro allan_w

        xvals=findgen(100)
        yvals=sqrt(xvals)

 tlb=widget_base(title='Example',/column)
         plotwindow=widget_draw(tlb,xsize=640,ysize=480,/button_event s)
 button=widget_button(tlb,value='Push to select a range')
        widget_control,tlb,/realize

        widget_control,plotwindow,get_value=wid
        wset,wid
        plot,xvals,yvals
        x_store=!x
        y_store=!y

        info={ plotwindow:plotwindow, $
         button:button,  $
               xvals:xvals,  $
               yvals:yvals,  $
               x1:!values.f_nan, $
               x2:!values.f_nan, $
               selecting:0,  $
               x_store:x_store, $
         y_store:y_store  }
 widget_control,tlb,set_uvalue=info
        xmanager,'allan_w',tlb,event_handler='allan_e'
end
```

Thanks,

Allan

Braedley wrote:
> In many of my programs (be it command line or widgets), I often need
> the user to enter a bounds of a particular graph by clicking on it.
> With basic windows, it's just using cursor like so:
>

> cursor, x1, junk, /data
> plots, [x1, x1], !y.crange, /data, color=254 ;color=red
> cursor, x2, junk, /data
> plots, [x2, x2], !y.crange, /data, color=254 ;color=red
>
> Reading through the widget_draw documentation, they suggest that cursor
> not be used for this type of thing, but instead use mouse events. This
> is despite the fact that using cursor hasn't caused any problems at all
> in my widget programs. However, I would still like to use the proper
> implementation whenever possible. So the question becomes, how do I
> grab mouse events only when I want them, and then return to the proper
> point in the current event handler?
>
> Braedley
>
