Subject: Re: The proper way of catching mouse button events from a draw widget?
Posted by Braedley on Wed, 18 Oct 2006 14:01:28 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Braedley writes:
>
>> 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.
>
> Well, any problems you have noticed yet. :-)
>
> Typically the reported cursor location is 2-3 pixels off from
> where you *think* you clicked. Not a big deal most of the time,
> but a HUGE annoyance when you are trying to do things like
> select a resizing square or something else that requires
> precision pointing.
>
>> 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?
>
> Uh, I'm not sure you fully understand how widget event
> handlers work. There is no "returning to the proper
> point" in an event handler. Events are one-shot deals.
> An event happens, it is handled. If another event occurs
> while the first event is being handled, the event gets
> queued up so it can be handled when the first event
> handler is finished, etc. Events are processed one after
> the other, consecutively.
>
> To see how widget events might be used to select the
> boundary of a plot, have a look at ZPLOT:
>
>   http://www.dfanning.com/programs/zplot.pro
>
> Here the boundaries are chosen by the user clicking
> and dragging on the plot. To restore the plot to its
> original boundaries, just click and release somewhere
> inside the plot.
>
> Cheers,
>
> David
> --

> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

I'm well aware of how event driven programming works, so I probably should have been a little more precise. What I should have asked is how would I go about creating a function or procedure that would perform the same action as a call to cursor, but follow the guidelines for draw widgets.

In any case, the example you provided, as well the example provided by Allan are good starting points. It's a bit unfortunate that the example I provided is just that: an example. It isn't indicative of the scope of the entire program in question. Due to the flexible nature of the program, a user may include procedures that need cursor like calls that would exceed the scope of the basic program.

Thanks anyways

Braedley