Subject: Re: The proper way of catching mouse button events from a draw widget?
Posted by Allan Whiteford on Wed, 18 Oct 2006 15:54:41 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Braedley writes:
>
>
>> 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.
>
>
> Well, presumably you heard that it can't be done. :-)
>
>

I think it perhaps could be... almost:

If we get the user subroutine to supply its own function name and a magical index to the draw widget event handler then store all it's variables before returning.

After the draw widget has collected the min and max x-range it can use a call_function to call back the original function which will take the index and essentially implement an entry point via a goto and then restore all the saved variables. It can then carry on with the min and max x-range.

I think that will pretty much give the same functionality as using an inline call to cursor. You still need to protect against someone not clicking on the draw widget twice but presumably the original cursor calls were open to that as well. I've not used cursor in this millennium so I don't know what happens if you click on a button when cursor is expecting some coordinates.

However, my 'solution' is repulsive and there is no way I'm supplying example code for the above suggested catastrophe. One day far from now in a job interview someone might put it down in front of me and ask if I actually wrote it.

I'm also of the opinion that if the above is the way to do it then it's almost indistinguishable from David's assertion that "it can't be done". However if you're in the situation where you have code you can't refactor for whatever reason and using cursor is giving you the wrong answer then maybe it's easier (but certainly not better) than re-writing

everything to function within how IDL is designed to do such things.

Thanks,

Allan