Subject: Re: selection box in widget program
Posted by David Fanning on Wed, 26 Sep 2007 16:59:10 GMT
View Forum Message <> Reply to Message

markb77@gmail.com writes:

> I'm writing a program that needs to ask the user to select a region of
> an image. I'd like to have a routine that would allow the user to
> click and drag a selection box (a rubber band box), returning the
> coordinates of the box to the program once the user has made a
> selection.
>
> This is a widget program using object graphics.
>
> The draw window displaying the widget has its own event handler,
> handling keyboard, motion, and button click events.
>
> What I want to do, essentially, is temporarily have all of the motion
> and button click events handled by the drawbox routine. This way I
> can loop on a call to widget_event, obtaining all of the mouse events,
> and continually redrawing the selection box, until the user is done.
> So, what I am going to try to do is UNSET the EVENT_PRO keyword to the
> draw widget, loop on widget_event as described, and then re-set the
> EVENT_PRO keyword to the original event handler.
>
> Is this how people usually go about handling these situations?

No. This is how people who don't fully understand widget programming
yet go about it. :-)

> It seems like clicking and dragging a selection box would be the world's
> most common routine, but I haven't found many examples of it when
> using object graphics and widget programs with event handlers.. etc.

Well, the actual mechanics of responding to events in your event
handler is pretty much the same, and can be gleaned from this article:

http://www.dfanning.com/widget_tips/rubberband_widget.html

What is different in object graphics is how you display the actual
graphics. Typically, on the DOWN event you take a "snapshot" of what
you want to be the "background" of your image", and then during the
MOTION events, you simple display your "box" on top of the background.
You can find the essentials in this article:

http://www.dfanning.com/ographics_tips/objectbox.html

In any case, you will find a program there (Zoombox) that
will illustrate how it can be done.

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.")

Subject: Re: selection box in widget program
Posted by David Fanning on Wed, 26 Sep 2007 17:07:21 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>> What I want to do, essentially, is temporarily have all of the motion
>> and button click events handled by the drawbox routine. This way I
>> can loop on a call to widget_event, obtaining all of the mouse events,
>> and continually redrawing the selection box, until the user is done.
>> So, what I am going to try to do is UNSET the EVENT_PRO keyword to the
>> draw widget, loop on widget_event as described, and then re-set the
>> EVENT_PRO keyword to the original event handler.
>>
>> Is this how people usually go about handling these situations?
>
> No. This is how people who don't fully understand widget programming
> yet go about it. :-)

On the other hand, this is how David Stern used to write widget
programs, which drove me to distraction. Maybe this is how people
on either end of the widget programming spectrum write widget
programs. But the vast majority of us work out something a whole
lot easier. :-)

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.")

Subject: Re: selection box in widget program
Posted by markb77 on Wed, 26 Sep 2007 17:20:09 GMT
View Forum Message <> Reply to Message

Ok, here's what I don't get about the way you propose. Let's say I have a button on my widget application that says "calculate image background". You click it, and the event handler sends you off to a routine that is going to ask you to select a region of the image, do the processing, update the program state, and return.

If I need to let the regular draw widget event handler handle the click and drag events when drawing the box, then I need to have a flag variable to tell the system that the user had previously clicked "calculate image background", and another flag to say that a click-and-drag is in progress, and that the program should ignore other events while this is going on. Moreover, there are many possible situations in which the user might need to click and drag a box, so each of those possible situations would also need their own flag variable, and all of those flags would have to be checked by the event handler whenever a button press or motion event happened, so that it would know what to do.

Sounds too complicated, for something so simple? No?

Mark

Subject: Re: selection box in widget program
Posted by David Fanning on Wed, 26 Sep 2007 18:38:53 GMT
View Forum Message <> Reply to Message

markb77@gmail.com writes:

> Ok, here's what I don't get about the way you propose. Let's say I
> have a button on my widget application that says "calculate image
> background". You click it, and the event handler sends you off to a
> routine that is going to ask you to select a region of the image, do
> the processing, update the program state, and return.
>
> If I need to let the regular draw widget event handler handle the
> click and drag events when drawing the box, then I need to have a flag
> variable to tell the system that the user had previously clicked
> "calculate image background", and another flag to say that a click-and-
> drag is in progress, and that the program should ignore other events
> while this is going on. Moreover, there are many possible situations
> in which the user might need to click and drag a box, so each of those
> possible situations would also need their own flag variable, and all
> of those flags would have to be checked by the event handler whenever

> a button press or motion event happened, so that it would know what to
> do.
>
> Sounds too complicated, for something so simple? No?

Yeah, pretty complicated. Reminds of some of the code I see coming out of ITTVIS. :-)

Here is what I would do. There are lots of things that can go on in draw widgets, obviously, so sometimes instead of writing a huge big event handler to handle all the possibilities, with the unavoidable nested IF-THEN-ELSE code that is inevitably required, I think about what "mode" a draw widget is in currently.

You might have a mode for drawing a box, another for drawing a circle, another for calculating the background intensity, etc. Normally, you have to give the user the opportunity to see what mode they are in, and this is usually accomplished by a set of exclusive buttons with pictures on them next to the draw widget. If the button with the circle is pushed in, you are in draw circle mode, etc. The button event handler is easy to write: when the button is selected, the "mode" variable is changed. Sometimes a mode change can be announced to the user by changing the shape of the cursor in the draw widget window.

Now the event handler for the draw widget becomes more manageable. When an event comes in, it looks to see what "mode" the draw widget is in, and dispatches that event and the info pointer (usually) to a module that handles just those events for that particular mode.

This makes it very easy to add a new "mode" to your draw widget. Just add a new line to the draw widget event handler in the mode case statement, and write the module that handles the events that are expected from the draw widget in that mode. This goes a LONG way toward keeping your code from breaking every time you make a change to it.

It also has the salubrious effect of gently pushing you toward an object-oriented programming style and increased awareness of why you want ALL your widget programs to be object programs, but that's a discussion for another day. :-)

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.