
Subject: Re: Widget-event handling problem
Posted by [ashmall](#) on Mon, 12 Apr 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

> The problem is that whenever one of the four (exclusive)
> buttons is selected, the event handling routine is entered twice

Jason,

You might try setting the NO_RELEASE keyword in your WIDGET_BUTTON functions.
You might find it easier (in the long run) to start using the CW_BGROU
function for producing groups of normal, exclusive and non-exclusive buttons
(again supplying the NO_RELEASE keyword).

Good Luck,

Justin

In article <3711C6D8.2781@rmsb.u-bordeaux2.fr>, Jason Brookes
<jason.brookes@rmsb.u-bordeaux2.fr> wrote:

> Hi,

>

> I have been experiencing an annoying widget-event handling problem. I
> create a widget "sub-panel" on which the user is allowed to set
> (exclusively) one of four possible buttons, each button corresponding to
> a different image. I have included the relevant code from the widget
> creation and event handling functions below. The problem is that
> whenever one of the four buttons is selected, the event handling routine
> is entered twice, AS IF responding to both the previous mouse button
> release, and the current mouse button press (this may or may not be what
> is actually happening). For example, if button_3 is currently set, and
> the user sets button 1, the event handling routine will first execute
> the code corresponding to 'stim_3' in the CASE statement followed by the
> code corresponding to 'stim_1'. I would much rather it executed only the
> code corresponding to 'stim_1'.

>

> Hope my explanation is clear(ish). Can anybody enlighten me ?

>

> While I'm at it, is there an elegant way of updating the maximum value
> for a floating-point slider ?

>

> Cheers,

>

> Jason

>

>

> Jason A. Brookes,

> 33076 Bordeaux,

> France.

>

> Tel: +33 (0)5 57 57 10 83

> Fax: +33 (0)5 56 96 13 41

> E-mail: jason.brookes@rmsb.u-bordeaux2.fr

>

>

>

>

>

>

> PRO sea_base_widget_event,event

>

> COMMON sea_common,im_par,sea_widget

>

> WIDGET_CONTROL,event.id,GET_UVALUE = event_val

>

> CASE event_val OF

>

> 'stim_1':\$

> BEGIN

> im_par.overlay_id = 1

> WIDGET_CONTROL,sea_widget.slider[1],set_slider_max = \$

> MAX(stat_map[*,*,im_par.dz,im_par.overlay_id-1])

> update_image_window

> END

>

> 'stim_2':\$

> BEGIN

> im_par.overlay_id = 2

> WIDGET_CONTROL,sea_widget.slider[1],set_slider_max = \$

> MAX(stat_map[*,*,im_par.dz,im_par.overlay_id-1])

> update_image_window

> END

>

> 'stim_3':\$

> BEGIN

> im_par.overlay_id = 3

> WIDGET_CONTROL,sea_widget.slider[1],set_slider_max = \$

> MAX(stat_map[*,*,im_par.dz,im_par.overlay_id-1])

> update_image_window

> END

>

```

> 'stim_4':$
> BEGIN
>   im_par.overlay_id = 4
>   WIDGET_CONTROL,sea_widget.slider[1],set_slider_max = $
>     MAX(stat_map[*,*,im_par.dz,im_par.overlay_id-1])
>   update_image_window
> END
>
> PRO ENDCASE
>
> END
>
>
> sea_base_widget
>
> sea_widget = {root:0L,slider:LONARR(2)}
>
> ....
> ....
>
> base = WIDGET_BASE(/COLUMN)
> sub_base = WIDGET_BASE(base,/ROW,/FRAME,/EXCLUSIVE)
>
> slider = CW_FSLIDER(base,TITLE = 'threshold',UVALUE = 'f_stat_sld',$
>   MINIMUM = 0.0,MAXIMUM = 50.0,VALUE = 25.0,FORMAT = '(F4.1)',$
>   XSIZE = 256)
>
> button_1 = WIDGET_BUTTON(sub_base,UVALUE = 'stim_1',VALUE = '2.5 Hz')
> button_2 = WIDGET_BUTTON(sub_base,UVALUE = 'stim_2',VALUE = '5.0 Hz')
> button_3 = WIDGET_BUTTON(sub_base,UVALUE = 'stim_3',VALUE = '7.5 Hz')
> button_4 = WIDGET_BUTTON(sub_base,UVALUE = 'stim_4',VALUE = 'Full F')
> CASE im_par.overlay_id OF
>   1: WIDGET_CONTROL,button_1,/SET_BUTTON
>   2: WIDGET_CONTROL,button_2,/SET_BUTTON
>   3: WIDGET_CONTROL,button_3,/SET_BUTTON
>   4: WIDGET_CONTROL,button_4,/SET_BUTTON
> ENDCASE
>
> sea_widget.root = base
> sea_widget.slider[1] = slider
>
> ....
> ....
>
> XMANAGER,'sea_base_widget',sea_widget.root
>
> END

```

Subject: Re: Widget-event handling problem
Posted by [steinhh](#) on Mon, 12 Apr 1999 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Jason Brookes wrote:

- > For example, if button_3 is currently set, and
- > the user sets button 1, the event handling routine will first execute
- > the code corresponding to 'stim_3' in the CASE statement followed by the
- > code corresponding to 'stim_1'.

An extra event is generated to signal the **release** of button 3.
See the keyword /no_release to widget_button.

Regards

Stein Vidar
