Subject: Keeping Button Pressed In?
Posted by hocmin on Wed, 24 Aug 2005 14:50:07 GMT
View Forum Message <> Reply to Message

I'm new to IDL but have some experience in programming. I'm trying to recreate the MS Paint/Adobe Photoshop metaphor with my application through a series of image buttons. Basically, I want one button to stay pressed down, signifying the "mode" the application is in. Is this possible with widget_button's using the /bitmap keyword? I realize I could throw something up with radio buttons to achieve the same functionality, but I really think the pressed image buttons is a more elegant solution.

If not possible, are there any workarounds? I tried using two different images, one to make it look like it's pressed. But the widget_button has too much padding between the image and the border of the button to make it "look" like it's pressed. I couldn't figure out a way around this either.

Any help would be greatly appreciated.

Thanks

Subject: Re: Keeping Button Pressed In?
Posted by pravesh.subramanian on Mon, 29 Aug 2005 14:57:34 GMT
View Forum Message <> Reply to Message

Excellent idea, david. now thats what we call a seasoned programmer!

Subject: Re: Keeping Button Pressed In?
Posted by David Fanning on Mon, 29 Aug 2005 15:25:48 GMT
View Forum Message <> Reply to Message

pravesh.subramanian@gmail.com writes:

> Excellent idea, david. now thats what we call a seasoned programmer!

Humm. "Seasoned", huh. Doesn't that mean "old"? :-(

Cheers,

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

Subject: Re: Keeping Button Pressed In?
Posted by Mark Hadfield on Mon, 29 Aug 2005 22:13:08 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> pravesh.subramanian@gmail.com writes:
>
>
>>Excellent idea, david. now thats what we call a seasoned programmer!
>
>
> Humm. "Seasoned", huh. Doesn't that mean "old"? :-(

1. seasoned - aged or processed; "seasoned wood"
   unseasoned - not aged or processed; "unseasoned timber"
2. seasoned - having been given flavor (as by seasoning)
   flavored, flavoured, tasteful - having flavor
3. seasoned - rendered competent through trial and experience;
   "troops seasoned in combat"; "a seasoned traveler"; "veteran
   steadiness"; "a veteran officer"
   veteran experienced - having become knowledgeable or skillful
   from observation or participation

But before we discuss whether David is "rendered competent through trial and experience" or "flavored" I would like to return to the topic and offer the observation that the Itools have buttons of the type required. (Eg if you bring up an Iplot window you will see an arrow button and another next to it that represents rotation.) I have looked at the source and for the life of me I cannot work out where and how they do this.

The following (from the Itool Developer's Guide) explains where the bitmap for the arrow button comes from:

> By default, bitmap files for icons used by the iTool system
> are stored in the bitmaps subdirectory of the resource
> subdirectory of the IDL distribution. If an icon's bitmap
> file is located in this directory, specify the base name
> of the file - without the filename extension - as the value
> of the ICON property of the component. For example, to use
> the file arrow.bmp, located in the resource/bitmaps subdirectory
> of the IDL distribution, specify the value of the ICON property
> as follows:
>
> ICON = 'arrow'

But I don't know how they modify it to show it has been pressed.

Can anyone point me to the relevant code?

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Keeping Button Pressed In?
Posted by Mark Hadfield on Mon, 29 Aug 2005 22:53:50 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
>
> ... the Itools have buttons of the
> type required. (Eg if you bring up an Iplot window you will see an
> arrow button and another next to it that represents rotation.) I
> have looked at the source and for the life of me I cannot work out
> where and how they do this.

Silly me. The buttons in question are just ordinary button widgets on an exclusive base. (See the EXCLUSIVE keyword to the WIDGET_BASE function.)

The original poster said

> Basically, I want one button to stay pressed down, signifying
> the "mode" the application is in.

If the word "mode" implies a choice between several mutually exclusive choices, then button widgets on an exclusive base fit the bill exactly.

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Keeping Button Pressed In?
Posted by David Fanning on Tue, 30 Aug 2005 00:00:50 GMT
View Forum Message <> Reply to Message

Mark Hadfield writes:

> Silly me. The buttons in question are just ordinary button widgets on an
> exclusive base. (See the EXCLUSIVE keyword to the WIDGET_BASE function.)
>
> The original poster said
>
> Basically, I want one button to stay pressed down, signifying
> the "mode" the application is in.
>
> If the word "mode" implies a choice between several mutually exclusive
> choices, then button widgets on an exclusive base fit the bill exactly.

I appreciate the definitions of "seasoned". I think I'm probably
"flavored". At least after I get done playing tennis. :-)

Yes, I use these kind of exclusive base buttons all the time.
I think of them as "toolbar" buttons" but they could just as
easily be "mode" buttons, as long as the mode is exclusive.

The buttons I was referring to that use Draw Widgets are
actually buttons I use in non-exclusive situations, where
several possible buttons can be selected at any particular
time, and I want a "selected" property to appear differently.

I came on the idea by peering closely at those buttons in
the exclusive bases and seeing how it was *they* appeared
to be selected. It was simply a matter of reversing the button
highlights. Or, another example of smoke and mirrors, depending
upon your philosophical leanings.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/

Subject: Re: Keeping Button Pressed In?
Posted by Mark Hadfield on Tue, 30 Aug 2005 00:31:56 GMT
View Forum Message <> Reply to Message

David Fanning wrote:
> Yes, I use these kind of exclusive base buttons all the time.
> I think of them as "toolbar" buttons" but they could just as
> easily be "mode" buttons, as long as the mode is exclusive.

>
> The buttons I was referring to that use Draw Widgets are
> actually buttons I use in non-exclusive situations, where
> several possible buttons can be selected at any particular
> time, and I want a "selected" property to appear differently.

I think then you can use a NONEXCLUSIVE base, which the documentation
describes thus:

NONEXCLUSIVE

Set this keyword to specify that the base can only have
button widget children. These buttons, unlike normal button
widgets, have two states-set and unset. Non-exclusive bases
allow any number of the toggle buttons to be set at one time.

(So setting NONEXCLUSIVE is not the same as not setting EXCLUSIVE.)

--
Mark Hadfield "Kei puwaha te tai nei, Hoea tahi tatou"
m.hadfield@niwa.co.nz
National Institute for Water and Atmospheric Research (NIWA)

Subject: Re: Keeping Button Pressed In?
Posted by Benjamin Hornberger on Fri, 21 Oct 2005 17:57:11 GMT
View Forum Message <> Reply to Message

Mark Hadfield wrote:
> Mark Hadfield wrote:
>
>>
>> ... the Itools have buttons of the
>> type required. (Eg if you bring up an Iplot window you will see an
>> arrow button and another next to it that represents rotation.) I
>> have looked at the source and for the life of me I cannot work out
>> where and how they do this.
>
>
> Silly me. The buttons in question are just ordinary button widgets on an
> exclusive base. (See the EXCLUSIVE keyword to the WIDGET_BASE function.)
>

Interesting. I find it inconsistent though that bitmap buttons behave
like that, while buttons with a text value (label) turn into radio
buttons in an exclusive base. What if I want a "depressed" button with a
text value?

Benjamin

Subject: Re: Keeping Button Pressed In?
Posted by Dick Jackson on Mon, 24 Oct 2005 04:39:53 GMT
View Forum Message <> Reply to Message

Hi Benjamin:

"Benjamin Hornberger" <benjamin.hornberger@stonybrook.edu> wrote in message
news:43592bf7$1_1@marge.ic.sunysb.edu...
>
> Interesting. I find it inconsistent though that bitmap buttons behave like
> that, while buttons with a text value (label) turn into radio buttons in
> an exclusive base. What if I want a "depressed" button with a text value?
>
> Benjamin

I've found this to work pretty well, a function that takes the text value as
a string and converts it to the RGB byte array for your Widget_Button. I
realize it's not perfect about aligning one button with no descenders (e.g.,
'GJPY') and another with descenders (e.g., 'gjpy'), but I don't have time to
make that very doable fix right now.

Hope this is of help to someone:

=====

```
FUNCTION BitmapForButtonText, str

;; Return an RGB byte array (w, h, 3) suitable for use as the Value of a
;; Widget_Button to display the text given in parameter 'str'.

;; Example:
; wTLB0=Widget_Base(/Row,/Exclusive)
; wBtn1=Widget_Button(wTLB0,Value=BitmapForButtonText('1'))
; wBtn2=Widget_Button(wTLB0,Value=BitmapForButtonText('2'))
; Widget_Control,wTLB0,/Realize

;; Dick Jackson  /  D-Jackson Software Consulting  /  dick@d-jackson.com

wTLB = Widget_Base()
wBtn = Widget_Button(wTLB)
font = Widget_Info(wBtn, /FontName)
sysColors = Widget_Info(wBtn, /System_Colors)
xs = 200                        ; Maximum width of button text
ys = 40
```

```
x0 = 3
y0 = 6
border = 3
Window, XSize=xs, YSize=ys, /Pixmap, /Free
Erase, Color=Total(sysColors.Face_3D * [1, 256, 65536L])
blankRGB = TVRD(True=3)
Device, Set_Font=font
XYOutS, x0, y0, str, /Device, Font=0, $
      Color=Total(sysColors.Button_Text * [1, 256, 65536L])
textRGB = TVRD(True=3)
WDelete, !D.Window
text2D = Total(textRGB NE blankRGB, 3)
whereX = Where(Total(text2D, 2) NE 0, nWhereX)
whereY = Where(Total(text2D, 1) NE 0, nWhereY)

IF nWhereX * nWhereY EQ 0 THEN result = blankRGB[0, 0, *] $
ELSE result = textRGB[(whereX[0]-border): $
                  (whereX[nWhereX-1]+border) < (xs-1), $
                  (whereY[0]-border): $
                  (whereY[nWhereY-1]+border), *]

IF !Order EQ 1 THEN result = Reverse(result, 2)

Return, result

END
```

=====

--
Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Keeping Button Pressed In?
Posted by Dick Jackson on Mon, 24 Oct 2005 17:08:43 GMT
View Forum Message <> Reply to Message

Hi again,

I've improved this routine a bit, it should be even more useful now!

"Dick Jackson" <dick@d-jackson.com> wrote in message

news:tAZ6f.274433$oW2.56864@pd7tw1no...
> Hi Benjamin:
>
> "Benjamin Hornberger" <benjamin.hornberger@stonybrook.edu> wrote in
> message news:43592bf7$1_1@marge.ic.sunysb.edu...
>>
>> Interesting. I find it inconsistent though that bitmap buttons behave
>> like that, while buttons with a text value (label) turn into radio
>> buttons in an exclusive base. What if I want a "depressed" button with a
>> text value?
>>
>> Benjamin
>
> I've found this to work pretty well, a function that takes the text value
> as a string and converts it to the RGB byte array for your Widget_Button.
> I realize it's not perfect about aligning one button with no descenders
> (e.g., 'GJPY') and another with descenders (e.g., 'gjpy'), but I don't
> have time to make that very doable fix right now.

That is now fixed, and I'd like to hear how the result looks on a Unix
system. Comments are welcome!

=====

```
FUNCTION BitmapForButtonText, str

;; Return an RGB byte array (w, h, 3) suitable for use as the Value of a
;; Widget_Button to display the text given in parameter 'str'.

;; Example:
; wTLB0=Widget_Base(/Row,/Exclusive)
; wBtn1=Widget_Button(wTLB0,Value=BitmapForButtonText('1'))
; wBtn2=Widget_Button(wTLB0,Value=BitmapForButtonText('2'))
; Widget_Control,wTLB0,/Realize

;; Dick Jackson  /  D-Jackson Software Consulting  /  dick@d-jackson.com

;;    Find what font and colours to use by making a button widget

wTLB = Widget_Base()
wBtn = Widget_Button(wTLB)
font = Widget_Info(wBtn, /FontName)
sysColors = Widget_Info(wBtn, /System_Colors)
Widget_Control, wTLB, /Destroy

;;    Find how high the bitmap needs to be for ascenders and descenders
;;    (highest and lowest points of characters) in this font
```

```
Window, XSize=100, YSize=100, /Pixmap, /Free
Device, Set_Font=font
y0 = 15
XYOutS, 0, y0, 'Ay', /Device, Font=0    ; Test with high and low letters
bwWindow = TVRD()                       ; Black background, white text
WDelete, !D.Window
IF !Order EQ 1 THEN bwWindow = Reverse(bwWindow, 2) ; Handle !Order=1
whRowUsed = Where(Max(bwWindow, Dimension=1) NE 0)
minY = Min(whRowUsed, Max=maxY)

;;    Calculate sizes and starting position

border = 2                          ; Width of border around text
xSize = (Get_Screen_Size())[0]          ; Maximum width of button text
ySize = (maxY-minY+1) + border*2
x0 = border
y0 = border+(y0-minY)

;;    Make window, draw text, read back

Window, XSize=xSize, YSize=ySize, /Pixmap, /Free
Erase, Color=Total(sysColors.Face_3D * [1, 256, 65536L])
blankRGB = TVRD(True=3)
XYOutS, x0, y0, str, /Device, Font=0, $
       Color=Total(sysColors.Button_Text * [1, 256, 65536L])
textRGB = TVRD(True=3)
WDelete, !D.Window
text2D = Total(textRGB NE blankRGB, 3)
whereX = Where(Total(text2D, 2) NE 0, nWhereX)

;;    Prepare result

IF nWhereX EQ 0 THEN $                  ; Nothing visible:
   result = blankRGB[0, *, *] $         ;    Return one column
ELSE $                              ; Else return width used plus
                                    ; border (plus two extra pixels
                                    ; to make Windows button look OK)
   result = textRGB[0:(whereX[nWhereX-1]+border+2) < (xSize-1), *, *]

;;    Compensate for reversal if !Order is 1

IF !Order EQ 1 THEN result = Reverse(result, 2)

Return, result

END

=====
```

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392

Subject: Re: Keeping Button Pressed In?
Posted by JD Smith on Mon, 24 Oct 2005 17:41:40 GMT
View Forum Message <> Reply to Message

On Mon, 24 Oct 2005 17:08:43 +0000, Dick Jackson wrote:

> Hi again,
>
> I've improved this routine a bit, it should be even more useful now!
>
> "Dick Jackson" <dick@d-jackson.com> wrote in message
> news:tAZ6f.274433$oW2.56864@pd7tw1no...
>> Hi Benjamin:
>>
>> "Benjamin Hornberger" <benjamin.hornberger@stonybrook.edu> wrote in
>> message news:43592bf7$1_1@marge.ic.sunysb.edu...
>>>
>>> Interesting. I find it inconsistent though that bitmap buttons behave
>>> like that, while buttons with a text value (label) turn into radio
>>> buttons in an exclusive base. What if I want a "depressed" button with a
>>> text value?
>>>
>>> Benjamin
>>
>> I've found this to work pretty well, a function that takes the text value
>> as a string and converts it to the RGB byte array for your Widget_Button.
>> I realize it's not perfect about aligning one button with no descenders
>> (e.g., 'GJPY') and another with descenders (e.g., 'gjpy'), but I don't
>> have time to make that very doable fix right now.
>
> That is now fixed, and I'd like to hear how the result looks on a Unix
> system. Comments are welcome!

I think you need to temporarily turn on decomposed color to get the
button background color correct. The bummer of Motif is that you
can't get depressed buttons in Exclusive/Non-exclusive bases, so this
routine helps us little. Instead, you always get a little diamond

next to the button, be it text or bitmap, to indicate selection
status. I've worked around this by simulating my own exclusivity and
changing the button value bitmap to look different when selected.
It's not ideal. I'm not sure if this is a Motif limitation or an IDL
limitation (buttons *do* depress when you click them).

JD

Subject: Re: Keeping Button Pressed In?
Posted by JD Smith on Tue, 25 Oct 2005 00:05:54 GMT
View Forum Message <> Reply to Message

On Mon, 24 Oct 2005 10:41:40 -0700, JD Smith wrote:

> On Mon, 24 Oct 2005 17:08:43 +0000, Dick Jackson wrote:
>
>> Hi again,
>>
>> I've improved this routine a bit, it should be even more useful now!
>>
>> "Dick Jackson" <dick@d-jackson.com> wrote in message
>> news:tAZ6f.274433$oW2.56864@pd7tw1no...
>>> Hi Benjamin:
>>>
>>> "Benjamin Hornberger" <benjamin.hornberger@stonybrook.edu> wrote in
>>> message news:43592bf7$1_1@marge.ic.sunysb.edu...
>>>>
>>>> Interesting. I find it inconsistent though that bitmap buttons behave
>>>> like that, while buttons with a text value (label) turn into radio
>>>> buttons in an exclusive base. What if I want a "depressed" button with a
>>>> text value?
>>>>
>>>> Benjamin
>>>
>>> I've found this to work pretty well, a function that takes the text value
>>> as a string and converts it to the RGB byte array for your Widget_Button.
>>> I realize it's not perfect about aligning one button with no descenders
>>> (e.g., 'GJPY') and another with descenders (e.g., 'gjpy'), but I don't
>>> have time to make that very doable fix right now.
>>
>> That is now fixed, and I'd like to hear how the result looks on a Unix
>> system. Comments are welcome!
>
>
> I think you need to temporarily turn on decomposed color to get the
> button background color correct. The bummer of Motif is that you
> can't get depressed buttons in Exclusive/Non-exclusive bases, so this

> routine helps us little. Instead, you always get a little diamond
> next to the button, be it text or bitmap, to indicate selection
> status. I've worked around this by simulating my own exclusivity and
> changing the button value bitmap to look different when selected.
> It's not ideal. I'm not sure if this is a Motif limitation or an IDL
> limitation (buttons *do* depress when you click them).

I've looked a bit deeper, noticing that iTools on Unix does indeed have depressed bitmap buttons, and found that the magic keyword required under Unix (and not Windows) is /TOOLBAR (added with v5.6). This will dispense with the radio button, and space the buttons close together, indicating selection status by "depressing" them. Interesting disparity between Windows/Motif, but I guess the Motif version is a bit more consistent.

JD
