Subject: Re: Help with pick menu widget code
Posted by davidf on Wed, 16 Sep 1998 07:00:00 GMT
View Forum Message <> Reply to Message

Doug Larson (djl@REMOVECAPSloki.srl.caltech.edu) writes:

> Well, I did what David told me, and of course it works :)

Well, you're lucky. I figured your chances were better than average at most when I fired the answer off at 11:00 last night. :-)

> The second part with the "Widget_Info" still confuses me as to why it
> works.

Here is the code:

```
  IF (Widget_Info(event.top, /Valid_ID) THEN $
     WIDGET_CONTROL, event.top, SET_UVALUE= pickPtr, /NO_COPY
```

You are probably confused because I did the usual advanced IDL thing of combining two commands in one. Widget_Info is a function that returns (in this case) either a zero or one based on whether the event.top widget is still a valid widget or not. I'm using the return value as the test in the IF statement. If the widget ID is valid, I put something in it's user value. If the widget ID is not valid, I just skip this statement.

> With the destruction of the top-level base preceding the SET_UVALUE, how
> does PickFinish ever know what was done to the values in the pickPtr? I
> guess I still have not grasped this whole event driven style of code yet.

To tell you the truth, PickFinish doesn't even care. :-)
Each time you go through *any* of your event handlers and do something to the data pointed to by pickPtr, you are changing the thing pointed to, not the pointer itself.

And the thing pointed to exists someplace else. (It exists on the HEAP, which is where all pointers and objects live.) What you have in this program is a *reference* to that space. When you destroy that reference, you do not destroy the things that the reference points to. This is both good news and bad news, although I would say it is mostly bad news for your program at the moment.

I say this because it looks like to me that when you destroy that top-level base you are destroying the only reference you have to that stuff stored in the pointer.

This results in a condition we call "leaking memory". In other words, each time you run this program some of your memory gets allocated and then leaks away as you destroy the only way you have to get to it. Over time, you will find that you have no more memory to do anything useful. (And Rose talks about cryptic error messages. You should see this one!).

To clean this pointer up, I recommend you use a CLEANUP routine. The purpose of the cleanup routine is to destroy pointer and object references, delete pixmaps, and generally do the sorts of things that need to be done when a program exits. This routine can be assigned to the top-level base at the time you call XMANAGER with the CLEANUP keyword:

```
   XMANAGER, 'picked', pickTLB, GROUP_LEADER = pickTLB, $
 CLEANUP='PICKMENU_CLEANUP'
```

The routine (and please place *all* your event handler and other routines IN FRONT OF your PickMenu routine. That way your program will work correctly when it is called and you won't have to individually compile it before you use it.) is written like this:

```
   PRO PICKMENU_CLEANUP, pickTLB
   Widget_Control, pickTLB, Get_UValue=pickPtr
   Ptr_Free, pickPtr
   END
```

When the TLB dies, this routine is called. It's sort of like calling a priest to issue last rites. The TLB is "mostly dead" to use one of my favorite phrases from the movie "The Prince's Bride". All the TLB can do is whisper the name of the thing in its user value. :-) You use this name to your own advantage, in this case freeing both the pointer and the data the pointer points to, thereby avoiding memory leakage.

> Would it be better to have the destruction "percolate up" to
> a handler for the pickTLB? If so, how do I do this?

Better, I think, to have a separate event handler for the Done button. I would assign it when I created the button, like this:

```
   done   = WIDGET_BUTTON(finishbase, VALUE = 'DONE', $
      Event_Pro='Out_of_Here')
```

It would be written like this (provided I had written the
Cleanup routine as I recommended above):

```
   PRO Out_Of_Here, event
   Widget_Control, event.top, /Destroy
   END
```

Cheers,

David

P.S. I would *really* give that event handler the name
"PICKMENU_OUT_OF_HERE" if I was trying to write code that
works all of the time. :-)

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438, Toll-Free Book Orders: 1-888-461-0155
Coyote's Guide to IDL Programming: http://www.dfanning.com/