Subject: Re: Pointers
Posted by J.D. Smith on Sat, 14 Aug 1999 07:00:00 GMT
View Forum Message <> Reply to Message

L
> (2) When you wish to return information from a dying widget. If you want
> a widget program to return an item of information, then it must be
> passed in a variable which has global scope. A pointer is the most
> convenient way to pass this kind of information. The method is
> - Create the information structure,
> - Store the information structure using a pointer,
> - Store the pointer in the user value of the widget top level base,

I would expand this step to include storing a copy of the pointer elsewhere, to eliminate the problem of dissappearing uvalues so common in non-blocking widget applications.

The best thing about pointers (and objects, for that matter) is that the heap data they reference is global and persistent, but doesn't suffer from the limitations of the common block. In fact, the treadworn common block had new life breathed into by pointers: finally something to stick into it which doesn't bloat your code and cause redefinition headaches.

One feature I use all the time with pointers is as follows: when saving a data snapshot (actually, an object!) of interest, I can "detach" those fragents of member data I don't need or want to save. Simply do something like:

```
saved_ptr=self.BigAndUselessDataPtr ; detach
self.BigAndUselessDataPtr=ptr_new() ; a null pointer
save, self
self.BigAndUselessDataPtr=saved_ptr  ; reattach
```

I actually wrap the whole thing in some error catching code to avoid losing data but you get the point. This way, I can avoid saving class definitions and member data of subsidiary objects and data structures I don't want or need, and can keep my saved files relevant. And what could be easier for saving a giant nested structure of structures of pointers to arrays of structures containing pointer arrays (etc.) -- certainly I don't want to follow that mess along myself and save it bit by bit.

(By the way, if you actually want to use this technique, please search the archives for a thread entitled "Restoring Objects from file" or some such.... there are a few issues to deal with -- not all of which David has documented on his website. But nothing is more exciting than replacing an object's "self" from one restored in place from file. It's like reincarnation or identity transmogrification. Very kharmic.)

I could go on and on, but basically, the more you use pointers, the
happier you'll be (reminding me of a camp song there...)

Good Luck,

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|