Subject: Re: Object graphics - some questions
Posted by davidf on Sun, 29 Mar 1998 08:00:00 GMT
View Forum Message <> Reply to Message

Neil Conway (neil.conway@ukaea.org.uk) writes:

> (Can't find a FAQ for this group in news.answers, am I looking in the
> wrong place ?)

http://ww2.sd.cybernex.net/~mgs/idl_faq.html

> I have just found a reason to use IDL 5's object graphics system. I
> read the manual, liked the bit about objects in general, then read the
> bit about the graphics objects and then my pleasure started to
> diminish very rapidly.
>
> My worries were soon to become disbelief. Unless I have grossly
> missed the point, object graphics is a dog, and cannot replace many
> functions of direct graphics.

Having just had another look at Dan Carr's famous coins at

http://www.rmi.net/~dcarr/dollar.htm

which were produced with IDL object graphics I find the word
"dog" perhaps a little unfair. I am quite sure those coins
could not have been rendered that way in IDL direct graphics.

I am willing to concede, however, that object graphics are still
low level, are difficult to learn from the given documentation,
and have limited usefulness on 8-bit systems that do not have
3D graphics acceleration. Given that most of us, when we purchase
our next computer, WILL have 24-bit graphics cards and 3D
graphics acceleration (at least I am quite sure MINE will),
I would probably describe object graphics as "premature" on the
kinds of computers most of us are still using.

> I'd love someone to show me a way to do what I expected to be able to
> do 'out of the box' with object graphics...

There is not much you can do "out of the box" with object graphics
as it stands now. You have to build systems, almost from scratch.
This will be improving somewhat in the IDL 5.1 release with the
addition of more "live tools", but I think there is still a lot
of work to do. I am sure the folks at RSI feel similarly.

> Here are the some of the
> problems I have hit already:

>
> 1. I use an 8-bit display, and almost all the X-terminals around my
> business are also 8-bit. When one brings up an object graphics
> (henceforth OG) window it takes (by default) nearly all the available
> colours to build an RGB map. When a second window comes up, it takes
> all the rest. I have found an option to use the indexed colour model,
> but it appears that each separate OG window uses a separate colour
> map, and thus one is doomed to have either very few colours per
> window, or the dreaded colour-map switching. OK, I could get around
> this by using 24-bit displays only (not an option for me, sadly).

Object graphics are clearly designed for 24-bit graphics cards. I have had some luck writing object programs for use on 8-bit displays, but I try to control my environment carefully. I have had poor results trying to combine direct graphics and object graphics windows in the same application.

Sadly, current object graphic programs ARE dogs on X-terminals. They are so slow as to be unusable. The problem, as I understand it, is that the entire 3D pixmap must be transported to the X-terminal and there are no standard X commands that allow that to take place quickly. Extensions to the standard X libraries are apparently planned, but are not yet in place, and certainly have not been standardized.

> 2. When preparing output for a printer, incredibly, there appears to
> be no programmatic way to specify a given output device. Rather, one
> must use a dialog and ask the user to pick one. Furthermore, the
> range of output devices seems not to include the likes of CGM files,
> but just kosher print devices. Did I miss something here ?

> 3. Using (for example) the postscript output ("EPSF" output, but it's
> not encapsulated), I was horrified and bemused to discover that a
> simple sine-wave (the "simple plot" from the manual) uses up 200kB of
> disk space, because (brace yourself) it's written as a BITMAP (!?!?).
> I simply can't get my head around that one.

I don't think you missed anything here.

> Please tell me that I've been dreaming, and that all of these are
> surmountable problems... If not, then DOG will be stuck in its kennel
> for the foreseeable future in my books.

I am actually finding that many of the things that I wanted to do with object graphics (and I do NOT do much 3D stuff, where I think object graphics excels even today) I can do with direct graphics and clever object and widget programming. Techniques for writing color aware applications with resizeable

graphics windows, automatic output to all kinds of output file
formats (and even directly to the printer), and other nifty
things are described in my book and on my web page. Add some
direct graphic object oriented programs to the mix, and you
can write some awfully nice IDL programs.

It is not much of a stretch to say that object programming has
radically changed my life! Well, maybe that is a stretch. :-)
But it has certainly made writing IDL programs a whole heck of a
lot more fun and interesting than it was before.

I'm not giving up on object graphics just yet, but I am picking
and choosing carefully when I use them. In the meantime, I'm
extremely pleased with the direction IDL is heading with respect
to object programming in general and positioning the program
to perform well in the computer environment I know is coming.
I'm just planning to buy my Dell Computer stock before J.D. Smith
reports his IDL time test results. I'm going to make a killing!

Cheers,

David

David Fanning, Ph.D.
Fanning Software Consulting
E-Mail: davidf@dfanning.com
Phone: 970-221-0438
Coyote's Guide to IDL Programming: http://www.dfanning.com/