Subject: Re: POLYFILL erases my tick values - solution?
Posted by davidf on Wed, 27 Sep 2000 07:00:00 GMT
View Forum Message <> Reply to Message

J.D. Smith (jdsmith@astro.cornell.edu) writes:

> This brings up an interesting side problem: I have a direct graphics widget
> application which undershades various parts of a plot. These shaded regions can
> be moved with the mouse or arrow keys after selection with a mouse click. Since
> they *underly* the plot, the only reliable way of moving them I've found is
> re-shading and re-plotting at each step of the move. As you can imagine, this
> causes the updates to be somewhat... unappealing. A pixmap in the normal usage
> won't seem to do the job, since usually the trick is to restore some portion of
> the window and then overplot some changing feature (like a rubber-band selection
> outline). There is no equivalent way for putting things *under* the restored
> area. One solution I've thought of is the SET_GRAPHICS_FUNCTION keyword of
> device with GXor, but I can't seem to make that work (since it doesn't stay
> "inside" of IDL's color table but runs the full gamut of the device's color
> table). I could also use tvrd() to store the plot and the shadings without plot
> separately, and OR them by hand. This saves all the replotting but at the
> expense of tvrd() and tv()'ing (which may make updates just as bad or worse).
>
> Any thoughts?

I think you may be stuck with this unappealing, but workable,
solution.

It is a common misconception of object graphics that when something
changes in a scene the whole thing doesn't need to be re-rendered.
It does. (I was under this misconception too when I first started
working with object graphics. There must be something in the
documentation that leads people to believe this, but I've forgotten
now what it was.)

You have a few more tricks available to you in object graphics,
of course. For example, it is possible, with some trickery and
judicious use of model objects, to take a "snap-shot" of all the
non-moving objects in a scene (this is called an "instance") and
render that over and over along with the object that is changing.
And what is "under" something else is easier to keep track of
(assuming you have a decent OpenGL compliant graphics card, of
course). But you still have to render the scene each time
something changes, and this can be fast or slow depending upon
factors that are not always in your control.

> Is this something which 3 lines of Object Graphics code could do
> in a snap?

Three lines!? No, probably not three lines. Thirty, maybe. And
you better hope it is a program that is worth the effort. :-)

Cheers,

David

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