Subject: Re: Object graphics questions and comment
Posted by davidf on Mon, 11 Aug 1997 07:00:00 GMT
View Forum Message <> Reply to Message

Paul van Delst is trying to draw a zoom-box using object graphics.
He writes:

> Anyway, I create a zoom box on a left button down event, add it to my
> plot_model and draw the resulting view.

[Details deleted]

> Now all this works great. I get a lovely zoom box except IT IS SOOOOO
> SLOW? I test my code on a 1000 element data set and the zoom box can't
> keep up with the mouse motion events. Increase the number of points and
> it's unusable.
>
> MY QUESTIONS: Is the snail pace response implicit of IDLs of object
> graphics (another e.g. IDLs Insight. Good lord) or is there another way
> to do this without having to do a
> (*info_ptr).window -> draw, (*info_ptr).view
> every time? In Direct Graphics you simply erase ONLY the box and then
> redraw ONLY the new box, you don't redraw the entire view. Can this be
> done using Object Graphics? The documentation is obtuse and unhelpful.

Paul, I am VERY sympathetic. I even thought seriously about dropping everything and working on an example for you, but I have promised my wife that I would work EXCLUSIVELY on my IDL book this month in the hope that some income might be generated...soon! :-)

But, IF I were going to do this, I would try it in a slightly different way. Rendering graphics objects is inherently slower than rendering direct objects. Techniques that worked with direct graphics will have to be modified or completely re-invented to work with object graphics. I'm not sure anyone has all the answers right now.

In direct graphics, the zoom box was drawn and erased, over and over, by using a pixmap and the DEVICE COPY technique (usually). I think the preferred technique in object graphics will be to have the unchanging parts of the graphics window in one model and the changing parts (e.g. the zoom box) in another. Then you can make an INSTANCE of the unchanging model. You can think of this as a snapshot of the unchanging part of the scene. (Kind of like the role of the pixmap, previously.) This snapshot can be rendered all at once, without having to render each piece of the model individually, which I think is what is slowing you down so much.

This should prove much faster, but as I say, I haven't had the chance to try it yet and I don't have any notion of how it compares to the

old pixmap method, except to expect that it will be slower, necessarily.
I look forward to seeing your example when you are finished with it. :-)

> MY COMMENT: IMO, IDLs Object Graphics is necessary for easier
> programming of user friendly data visualisation tools. However, the
> speed of Object Graphics verges on being laughable (my perhaps poorly
> programmed example above notwithstanding). It is a step backwards and
> does not encourage the average IDL programmer (me) to utilise this
> powerful tool. My (hopeful) assumption is that future incremental
> releases of IDLv5 will contain Object Graphics that produce graphics as
> fast or faster than Direct Graphics.

I have found object graphics slower, yes, but not verging on laughable.
At least not for tightly focused object programs like those on my web
page. Insight is probably too slow for real work, but I think
Insight is trying to be all things to all people, which makes it
difficult to write efficiently. I would say it's too early to tell
if object graphics are too slow. We need more experience using them
and writing object programs. I agree that there is a steep learning
curve and that RSI has not helped much with their object graphics
documentation. But in my line of work, that is what is known as an
"exploitable niche".

I'm going back to work on my book... :-)

Cheers,

David

-- -

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