Subject: Re: Object rendering with dynamic views
Posted by Mark Hadfield on Tue, 23 Jan 2001 03:53:14 GMT
View Forum Message <> Reply to Message

"Ben Tupper" <pemaquidriver@tidewater.net> wrote in message
news:3A6CED48.98337741@tidewater.net...
> Hello,
>
> I am wondering about the best way to manage objects graphics
> when the display consists of changing and unchanging
> models. I have done it three different ways in the past,
> but I never considered the merits of each (until now.)
>
> (1) The first way is that outlined in the online manual
> (also David has a nice page about this.) This method
> involves just one view. The view is first rendered with
> the static portion of the graphics exposed and the dynamic
> parts hidden using :
> myDraw->DRAW, myView, /Create_Instance
> Then the view is made transparent, the static portion is
> hidden and the dynamic portion is exposed. After that the
> view is drawn using:
> myDraw->DRAW, myView, /Draw_Instance
> Any subsequent changes in the dynamic atoms/models are
> rendered using the DRAW_INSTANCE keyword.
>
> (2) The second way is to create two overlapping views, the
> static underneath and the dynamic on top with the
> TRANSPARENT keyword set. Then draw each view using the
> CREATE_INSTANCE and DRAW_INSTANCE keywords as needed. This
> is a method discussed (a long time ago) on the newsgroup,
> but I can't find it documented anywhere.
>
> (3) Put all the atoms/models in one view and render the
> whole thing as one. Grind-grind-grind. I use this method
> when in a hurry to write code, but I really don't want to
> look at it.
>
> Can someone explain the relative merits/pitfalls of each of
> the methods (in particular the first two?)

No, but if you would like to compare them and report back to the group, that would be much appreciated :-)

I can only offer my $0.02 worth:

When I last tried to compare these different methods I wasn't clever enough to think of method 1. (Obviously I should read the manual more carefully.

Where is this described, anyway?) I tried method 2 because that seemed to be what was implied by the documentation for IDLgrWindow::Draw. Method 1 certainly seems to be more elegant than method 2, doesn't it?

But what I did find is that on my machine (a Wintel PC with ho-hum graphics hardware) method 2 was *much* slower than method 3. So I haven't tried "instancing" since. It has been suggested to me that the result might have been different had I been using the software renderer (RENDERER=1) instead of the hardware renderer.

I think this area deserves a careful examination. (Thanks for volunteering Ben.) I suspect you will find that performance is important, but that the ranking of the different methods will vary widely depending on OS, graphics hardware, IDL settings and day of the week.

Mark Hadfield
m.hadfield@niwa.cri.nz http://katipo.niwa.cri.nz/~hadfield/
National Institute for Water and Atmospheric Research
PO Box 14-901, Wellington, New Zealand