Subject: Re: Object graphic 3d Scatterplot
Posted by Ben Tupper on Wed, 22 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

David Fanning wrote:

> Ben Tupper (tupper@seadas.bigelow.org) writes:
>
>> That's ok for a small dataset; don't waste your time for
>> a large set.
>
> Humm. That's what I was thinking too. But I'm not clear
> by what I mean by "large". Can you share a bit more of
> your experience in this area, Ben?
>
>

Hello,

Well, I found that I couldn't manage more than 10-100 points without seeing performance drop. For each point, I defined a symbol obect and placed them all in an array. A lot depended on whether or not these points were draped across a surface with some kind of image texture. A lot depends on the platform I suppose. After posting this, I'llprobably find out it is the programmer. The data sets I am using are numbering 1000s of data points. To be sure, a visualization with 5000-1000 points can begin to obscure the different class distributions if the are more than just a few data classes (colors).

It seems like such a simple thing (and wicked, as we DownEasters say, vital to datah vizulizin'). In direct graphics, I use colored (sized, etc.) data symbols all of the time to communicate some extra dimensionality to the data.

Here's what all I know about the relationship between this attribute object (IDLgrSymbol) and its parent (IDLgrPlot, IDLgrPolyLine,...)

SYMBOL (Get, Set)
Set this keyword to a vector containing instances of the IDLgrSymbol object class. Each symbol in the vector will be drawn at the corresponding plotted point. If there are more points than elements in SYMBOL, the elements of the SYMBOL vector are cyclically repeated. By default, no symbols are drawn. To remove symbols from a plot, set the SYMBOL property equal to a null object reference.

So the plot object would have to perform umpteen GetProperty calls to umpteen symbol objects. (Which, by the way, the documentation doesn't make it very clear that this field can be set to an array of symbol objects.) This is, I suppose, the correct way to define the realtionship between attribute objects and parent objects as I read in Object-Oriented Design Heuristics by Arthur J. Riel (good advice on that David). But, I seems to me that it might be equally right to make one GetProperty call to the attribute object for something like COLOR and then cycle through those colors.

Since I'm still an object newbie, maybe you could shed some light on this?

Ben

--
Ben Tupper

Bigelow Laboratory for Ocean Science
tupper@seadas.bigelow.org

pemaquidriver@tidewater.net