Subject: Propagating properties
Posted by Struan Gray on Fri, 16 Oct 1998 07:00:00 GMT
View Forum Message <> Reply to Message

I've decided to set myself a programming excercise in order to really get to grips with objects - as opposed to reading the manual and fooling myself that I understand it. I am concentrating on a program to draw crystal and surface atomic structures, since the structure definitions are nicely hierarchical and can easily be objectified, and because I need good 3D plotting so I am forced to use object graphics.

I've been playing with generating arrays of spheres using multiple instances of the example 'orb' object. This works nicely and simply but I've hit a conundrum. I would like to create an ur-orb which represents an element, say Si, with properties such as size and colour which are adopted by all the instances of Si in the crystal. A problem crops up when after creating the crystal I decide that all the Si atoms should be green rather than blue, or that they should be a different size. I would like to be able to just edit the ur-orb and have the changes propagate automatically, without having to re-create the whole crystal from scratch (which takes some time on my computer).

I can think of various ways to deal with the problem. The the two most promising are for the ur-orb to maintain a list of dependent objects (which seems ugly and slightly defeats the point of object programming), or I can subclass 'orb' to create an 'atom' class that checks with a specific ur-orb parent every time it draws itself (which has a performance hit, and suffers from the lack of a 'copy object' or 'copy properties' command discussed here previously). Does anyone have a better idea?

Also, if anyone knows of a library of other 3D graphics primitives (cylinders, cones etc) along the lines of 'orb' I would love to know about it.

Struan