phys2new wrote:
> I'm writing a little program to make an animated diagram using object
> graphics. I'm building simple composite objects e.g. arrows = cylinder
> + cone built out of polygons. During the animation I want the color of
> each arrow to change as I rotate it in 3-D (this is color-coding, I'm
> not trying to re-invent lighting etc!). Is there a way to do this to
> the composite model, or
> do I need to adjust the color of each polygon individually?

I'm sure someone has posted a response to this as my news server seems
to only serve up about half of the posts here but at some point you'll
need to change the color of each IDLgrPolygon object. Usually you would
create a new "arrow" class containing your cone and cylinder
IDLgrPolygon objects by subclassing IDLgrModel. Then in the SetProperty
method of this new class you would handle these details automagically
using keyword inheritance. Assuming you had some specific property
"size" that determined shaft length and cone radius, you would use the
_extra keyword inheritance mechanism to pass on the IDLgrPolygon and
IDLgrModel properties:


```
pro myArrow::SetProperty,  size=size, $
                _extra=extra

    <Do arrow specific stuff here>

    self.head -> SetProperty, _EXTRA=extra
    self.shaft -> SetProperty, _EXTRA=extra
    self->IDLgrModel::SetProperty, _EXTRA=extra

end
```


Properties specific to a class will be passed on (like color for
IDLgrPolygon) and quietly ignored by classes that do not possess that
property.

Or you can just set the head and shaft properties individually.

-Rick