

---

Subject: Re: Need an "atom" object  
Posted by [steinhh](#) on Thu, 01 Apr 1999 08:00:00 GMT  
[View Forum Message](#) <> [Reply to Message](#)

---

David Fanning wrote:

> I'm trying to save myself a bit of time and help out a friend  
> at the same time. We are looking for some kind of "atom"  
> object that would be a sphere with a particular color  
> and radius. It might have its own light source associated  
> with it. The idea would be to clump these atoms together  
> into a rotatable "molecule", if you like.

I think Richard Adams' suggestion of using the "orb" data type would probably fill your needs, David, but just in case you'd like a small, quick'n dirty thing to build on, I made the following implementation of a "ball" object the other day.

Like the "orb" object, it's subclassed from the IDLgrModel class, and contains a polygon object. The surface is built up by a square array, equally spaced in [theta,phi] "space". Input radius r and number of points n.

The polygon vertex list forms a "quad strip" to take advantage of polygon mesh optimization.

I see the "orb" class has an inherent "position", personally I like to implement this through a translation of the "ball" model itself.

Regars,

Stein Vidar

-----  
PRO ball::cleanup  
 self->idlgrmodel::cleanup  
END

FUNCTION ball::init,r,n,color=color  
  
 if n\_elements(color) eq 0 then color=[255,255,255]  
  
 dummy = self->idlgrmodel::init()

```

x = rebin(reform(findgen(n)/n,n,1,/overwrite),n,n,/sample)
y = rebin(reform(findgen(n)/(n-1),1,n,/overwrite),n,n,/sample)

theta = x*(360.0*!dtor)
phi = (y-0.5)*(180.0*!dtor)

sinphi = sin(phi)
cosphi = cos(temporary(phi))

costheta = cos(theta)
sintheta = sin(temporary(theta))

xp = r* temporary(costheta)*cosphi
yp = r* temporary(sintheta)*temporary(cosphi)
zp = r* temporary(sinphi)

gons = lindgen(5,n*(n-1))

FOR yi=0,n-2 DO BEGIN
  yyi = [yi,yi,yi+1,yi+1]
  FOR xi=0,n-1 DO BEGIN
    xxi = [xi,(xi+1) MOD n,(xi+1) MOD n,xi]
    gons(0,yi*n+xi) = [4,xxi + yyi*n]
  END
END
stop

xp = reform(xp,n*n,/overwrite)
yp = reform(yp,n*n,/overwrite)
zp = reform(zp,n*n,/overwrite)

ball = obj_new('idlgrpolygon',xp,yp,zp,polygons=gons,color=color)

self->add,ball
self->translate,100,0,2*r
return,1
END

PRO ball__define
  dummy = {BALL,$
    INHERITS idlgrmodel $
  }
END

```

---