
Subject: IDLgrPoly[line|gon] and cv_coord
Posted by [paul wisehart](#) on Tue, 07 Jan 2003 22:35:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi,
Its me again w/the 3D spheres :)
I'm having trouble understanding the coordinate system
w/regards to the IDLgrPolyline (or Polygon) methods and 'orb's.

Here's an example:

```
;-----  
my_image = obj_new('IDLgrImage', earthImage)  
my_orb = obj_new('orb', COLOR=[0, 255, 0], RADIUS=0.8, $  
    DENSITY=2, /TEX_COORDS, TEXTURE_MAP=oimage1,style=1,hide=0,/zero)  
  
pt_radius = 1  
pt1 = [0,0,pt_radius]  
pt2 = [15,0,pt_radius]  
  
sph_coord = [pt1,pt2]  
rect_coord = CV_COORD(FROM_sphere=sph_coord,/to_rect)  
  
my_poly = obj_new('idlgrpolyline',rect_coord,$  
    thick=10,color=[255,0,0])  
  
my_model -> add, my_poly  
my_model -> add, my_orb  
  
xobjview, my_model  
;-----
```

My 'pt1' and 'pt2' are supposed to be lat/lon coordinates.(spherical)
I convert them to rectangular coordinates.

I've tried pt_radius values .4 -> 1.4 (or so)

I'm trying to draw a line on the surface of the sphere.
The lines I am getting are outside the sphere or not there.
I cannot figure out how the coordinate system works.
I am assuming that by leaving the radius constant my lines will
follow the curve of a sphere.

Whats the default coordinate range of an 'orb' object?
Where's [0,0,0]? At the center of the sphere?
Is the sphere in a 'frame' where [0,0,0] is a vertex?

I can't find any info on the coordinate systems used in object
graphics in the manual either. Maybe someone know's where to look?

(IDL 5.4 by the way)

THANKS!!!!

I'll try to stop bugging you after this one.

--

paul wisehart
wisehart <at> runbox <dot> com

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by [David Fanning](#) on Wed, 08 Jan 2003 21:49:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

Rick Towler (rtowler@u.washington.edu) writes:

> I can't argue that there doesn't exist a graphics library whose coordinate
> system is not rectangular, but OpenGL's is.

Yes, this is probably true. In any case, it makes sense. :-)

> Try defining an IDLgrPolygon
> with [angle, radius, z] values instead of [x,y,z]. I don't think you'll get
> what you were expecting.

No, I'm sure not. But what I had in mind was defining an FSC_Polygon.
I'm pretty sure I could make one of those that allowed you
to specify angle, radius and z value. But you are probably right
that I would eventually have to translate it back to OpenGL's
rectangular coordinate system.

I guess rotations would be the same thing. In *my* odd coordinate
system you could appear to rotate about (3.459, 78.452), but to
actually implement it in my code I'd have to translate it
to the origin to take advantage of OpenGL, then translate it
back to where it was suppose to be. Lot's of bother, to be sure.
No one would volunteer to do it, I'm sure of that. :-)

Cheers,

David

--

David W. Fanning, Ph.D.
Fanning Software Consulting, Inc.
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: <http://www.dfanning.com/>

Toll-Free IDL Book Orders: 1-888-461-0155
