Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by David Fanning on Tue, 07 Jan 2003 22:58:47 GMT
View Forum Message <> Reply to Message

paul wisehart (wisehart@runbox.com) writes:

> 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?

You look in the mind of the programmer that created the object. :-)

The thing about object graphics is that there *is* no
coordinate system, except the one you arbitrarily set
up with the view object. That is to say, the viewplane
rectangle, and the near and far clipping planes define
the view volume, which *is* your coordinate system.
The units of this coordinate system are completely
arbitrary. You make them up.

Of course, this is one of the huge advantages of object
graphics: you really can't do it *wrong*. :^)

Cheers,

David

P.S. But let's just say if you don't really know what
theater you are playing in, it's a bit of a trick
to get all the actors to show up at the same place
in time for the show. I'd say that's the main challenge
of object graphics, in a nutshell. :-)

--
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

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by Rick Towler on Tue, 07 Jan 2003 23:15:02 GMT
View Forum Message <> Reply to Message

"paul wisehart" <wisehart@runbox.com> wrote

Hi Paul,

You are close. Your lat/lon data isn't in the correct format so cv_coord isn't returning meaningful results.

The spherical data must be in the form [3,n] so your line:

sph_coord = [pt1,pt2]

should be:

sph_coord = [[pt1],[pt2]]

I have attached a modified version of your program below. Note that my lat/lons are in degrees and I set the degrees keyword to CV_COORD accordingly.

You can answer your questions about the orb yourself. Use the getproperty method of the orb to get the DATA property of the orb's IDLgrPolygon object. The data will be a [3,n] array of verticies that make up the orb.

```
my_model = obj_new('idlgrmodel')

my_orb = obj_new('orb', COLOR=[0, 255, 0], RADIUS=0.8, $
       DENSITY=2, style=2);,hide=0,/zero)
my_model -> add, my_orb

pt_radius = 0.801
sph_coord = FLTARR(3,361)
sph_coord[0,*] = FINDGEN(361)
sph_coord[2,*] = pt_radius

rect_coord = CV_COORD(FROM_sphere=sph_coord,/to_rect, /DEGREES)

my_poly = obj_new('idlgrpolyline',rect_coord, thick=2.0, $
   color=[255,0,0])
my_model -> add, my_poly

xobjview, my_model, /BLOCK

obj_destroy, my_model

end
```

Enjoy!

-Rick

> 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 Thomas Gutzler on Wed, 08 Jan 2003 07:16:35 GMT
View Forum Message <> Reply to Message

Hey,

David Fanning wrote:
> paul wisehart (wisehart@runbox.com) writes:
>
>> 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?
>
> You look in the mind of the programmer that created the object. :-)
>
> The thing about object graphics is that there *is* no
> coordinate system, except the one you arbitrarily set
> up with the view object. That is to say, the viewplane
> rectangle, and the near and far clipping planes define
> the view volume, which *is* your coordinate system.
> The units of this coordinate system are completely
> arbitrary. You make them up.
>
> Of course, this is one of the huge advantages of object
> graphics: you really can't do it *wrong*. :^)

I used to scale every Object to have max 2 Units in each dimension and
translating it to have 0/0/0 in its center because I discovered that the
visible area is only [-1..1] in each dimension. I tried to enlarge it
but without any success. Did I miss something or is that the way to do
it correctly?

Tom

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by David Fanning on Wed, 08 Jan 2003 13:23:37 GMT
View Forum Message <> Reply to Message

Thomas Gutzler (tgutzler@ee.uwa.edu.au) writes:

> I used to scale every Object to have max 2 Units in each dimension and
> translating it to have 0/0/0 in its center because I discovered that the
> visible area is only [-1..1] in each dimension. I tried to enlarge it
> but without any success. Did I miss something or is that the way to do
> it correctly?

I think you missed something. What you describe is
the default viewplane rectangle. It is certainly
not "wrong" to scale everything to it. It's just
that this particular coordinate system is not
always convenient.

If you are working with images of a particular
size, for example, it is sometimes convenient to
have an "image" coordinate system (0 to 512, say).
I often prefer to have a "normalized" coordinate
system of 0 to 1, since I am more familiar with
how to scale things into this coordinate system
than one that goes from -1 to 1.

The thing about object graphics is, you can do
whatever you like and whatever is convenient for
the problem at hand. But, if you are like me, I
find all of this terribly confusing, and it took
me literally weeks before I could get the components
of a line plot (axes, titles, the line plot itself)
with all the bits showing up in the proper relationship
to one another.

That's why I wrote the NORMALIZE function. That little
guy, which you see in every object graphics program I've
ever written, is magical. It just seems to *know* how
to put things where they need to go. I'm sure I couldn't
write an object graphics program without it, Ronn Kling's
excellent book on the subject not withstanding. :-)

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

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by paul wisehart on Wed, 08 Jan 2003 13:58:17 GMT
View Forum Message <> Reply to Message

Thanks a lot for the help.

Its slowly started to make sense to me.

--
paul wisehart
wisehart <at> runbox <dot> com

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by Rick Towler on Wed, 08 Jan 2003 17:55:56 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote
> paul wisehart (wisehart@runbox.com) writes:
>
>> 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?
>
> You look in the mind of the programmer that created the object. :-)
>
> The thing about object graphics is that there *is* no
> coordinate system, except the one you arbitrarily set
> up with the view object.

I think you should be careful how you are phrasing this. Object graphics
uses a right-handed rectangular coordinate system. The units and view
volume may be arbitrary but you always have a point [0,0,0] and 3 mutually
perpendicular axes with regular intervals passing thru it which define
[x,y,z] values.

-Rick

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by David Fanning on Wed, 08 Jan 2003 18:37:42 GMT
View Forum Message <> Reply to Message

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

> I think you should be careful how you are phrasing this. Object graphics
> uses a right-handed rectangular coordinate system. The units and view
> volume may be arbitrary but you always have a point [0,0,0] and 3 mutually

> perpendicular axes with regular intervals passing thru it which define
> [x,y,z] values.

Humm. I wonder? While it's true that I have never
used anything *but* a rectangular coordinate system
for a program, it is not immediately obvious to me
that this is the only possible coordinate system that
can be set up. I'm almost certain a (0,0,0) point is
not required. What makes you think this is the case?

I wonder is the following quote would apply here? This
was on the signature of an e-mail a friend sent to me
the other day:

"As often happens in science, the paradox was resolved
as soon as the obvious was abandoned in the face of
experimental evidence." {from a recent review in Science}

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

Subject: Re: IDLgrPoly[line|gon] and cv_coord
Posted by Rick Towler on Wed, 08 Jan 2003 21:04:00 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote:
> Rick Towler (rtowler@u.washington.edu) writes:
>
>> I think you should be careful how you are phrasing this. Object
graphics
>> uses a right-handed rectangular coordinate system. The units and view
>> volume may be arbitrary but you always have a point [0,0,0] and 3
mutually
>> perpendicular axes with regular intervals passing thru it which define
>> [x,y,z] values.
>
> Humm. I wonder? While it's true that I have never
> used anything *but* a rectangular coordinate system
> for a program, it is not immediately obvious to me
> that this is the only possible coordinate system that

> can be set up. I'm almost certain a (0,0,0) point is
> not required. What makes you think this is the case?

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

I didn't mean to imply that you need [0,0,0] in your view. What I meant was that whether you choose to include [0,0,0] in your view volume or not the origin is out there, somewhere. You can ignore this until you start rotating things. Models are rotated about a defined axis which passes through [0,0,0] regardless of your view settings.

-Rick

Exhibit A:

```
orb=obj_new('orb', pos=[20,0,0], radius=0.5, style=1, density=0.5)

oview=obj_new('idlgrview', viewplane_rect=[19,-1,2,2], eye=3, $
   zclip=[1,-1])

oview -> add, orb

owin = obj_new('idlgrwindow')

owin -> draw, oview

for n=0,500 do begin
   orb->rotate,[0,0,1], 0.01
   owin -> draw, oview
endfor

orb -> reset
orb -> setproperty, pos=[0,0,0], color=[100,200,100]
oview -> setproperty, viewplane_rect=[-1,-1,2,2]

for n=0,500 do begin
   orb->rotate,[0,0,1], 0.01
   owin -> draw, oview
endfor

obj_destroy, [oview, owin]
```

end