Subject: 3D UserSym?
Posted by Brian Daniel on Mon, 03 Aug 2009 14:24:00 GMT
View Forum Message <> Reply to Message

I created at 3D scatter plot and animated a rotation using
XInterAnimate. When I view the scatter plot in the Z vs XY plane
(through manipulating the Az and Ax keywords) my plot points disappear
because they are drawn in just 2 dimensions. Is there a way (maybe
with UserSym) to create 3D plot points? Thanks!

Subject: Re: 3D UserSym?
Posted by David Fanning on Tue, 04 Aug 2009 15:46:08 GMT
View Forum Message <> Reply to Message

Brian Daniel writes:

> PS - I've used the iTools package in the past, but the size of my data
> sets make the interaction really clunky. And again, being new to
> object-oriented stuff turned me off as well. If I start to drown in
> objects again, I may use iTools as a floatation device.

Well, object programs and iTools are pretty much related
in the way Boulder is related to Mumbai. That is to say,
I wouldn't be thinking of iTools as a floatation device.
More like a lead weight, would be my first thought. :-(

Cheers,

David

--
David Fanning, Ph.D.
Coyote's Guide to IDL Programming (www.dfanning.com)
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: 3D UserSym?
Posted by penteado on Wed, 05 Aug 2009 19:48:54 GMT
View Forum Message <> Reply to Message

On Aug 4, 12:46 pm, David Fanning <n...@dfanning.com> wrote:
> Well, object programs and iTools are pretty much related
> in the way Boulder is related to Mumbai. That is to say,
> I wouldn't be thinking of iTools as a floatation device.
> More like a lead weight, would be my first thought. :-(

I disagree with that. I have used iTools as a shortcut to make object graphics: Instead of creating the objects for the view, model, lines, labels, axes and everything, I created an invisible iTool, and then take its view object and put it in my draw window. I suppose it is also possible to borrow widgets setup by the iTool, so that it would be easy to put things like zoom and range controls in my applications, but I have not yet tried to do it.

Subject: Re: 3D UserSym?
Posted by David Fanning on Wed, 05 Aug 2009 20:28:47 GMT
View Forum Message <> Reply to Message

pp writes:

> I disagree with that. I have used iTools as a shortcut to make object
> graphics: Instead of creating the objects for the view, model, lines,
> labels, axes and everything, I created an invisible iTool, and then
> take its view object and put it in my draw window. I suppose it is
> also possible to borrow widgets setup by the iTool, so that it would
> be easy to put things like zoom and range controls in my applications,
> but I have not yet tried to do it.

Well, let us know when you try it. :-)

I've no particular beef with iTools if they do what you want them to do. (Of course, I think aesthetically they are a disaster. For example, why would an iSurface plot come up with a default action to *translate* it, for God's sake?) Other than the fact I just don't find them at all intuitive, which I have always blamed on myself and not on iTools, I just don't find them handy for anything I do.

I know for a fact that if you try to program them to work in your own personal style you better be prepared for a long siege. They are, in a word, freaking complicated. You seem to be figuring them out, and that at least gives me hope.

Putting them into the hands of beginning IDL users, whom I believe is their intended audience, just makes me very, very nervous. They have a tendency to ask these really simple questions that I just have an extremely difficult time answering.

If you want a shortcut for building object graphics programs, I recommend Revolution, which I think you

can find on the IDL Code Contrib web page. That program produces beautiful code which you can actually use and modify easily for your own purposes. And, a significant bonus, you might actually learn about IDL object programming in the process. Something I doubt seriously would happen with a similar time investment in the iTools code.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: 3D UserSym?
Posted by Brian Daniel on Mon, 10 Aug 2009 13:22:31 GMT
View Forum Message <> Reply to Message

I put in some time and effort on learning the basics of objects and object graphics. I relied heavily on Scatter_Plot.pro, an example from a previous post on this thread. I've run across some challenges.

1) I have a memory leak. I've explicitly thrown every object in my Container object and destroy it in my cleanup routine. I have checked to make sure the routine gets called, but I still get 6 objects and 20 pointers still alive after I quit the program. No new objects are created elsewhere in the widget event handlers, but the objects are called in and put back for every one. The other strange thing is that even if I just open and close the GUI (i.e. without creating any user events), I get the same amount of leakage. Any memory advice is greatly appreciated.

2) One axis of the scatter plot is of a much different scale (2 order of magnitude different). The 'orb' object is set with a radius value in data coordinates, so my orbs look like discs... getting me back to the same problem I had to begin with (2D symbols).

3) This is a less important issue, but still worth mentioning. I have a lot of data to plot, and it looks like a big cloud of points. There are 3 data points that are important to see in the context of the whole data set. Using direct graphics, I could just over plot the 3 points and always be able to see them. How can you do that in object graphics? Is there a way to set my full data set to be partially transparent?

Thanks for everybody's help so far. You've helped me come a long way
in a short time.

> If you want a shortcut for building object graphics
> programs, I recommend Revolution, which I think you
> can find on the IDL Code Contrib web page. That program
> produces beautiful code which you can actually use and
> modify easily for your own purposes. And, a significant
> bonus, you might actually learn about IDL object programming
> in the process. Something I doubt seriously would happen
> with a similar time investment in the iTools code.
>
> Cheers,
>
> David
>
> --
> David Fanning, Ph.D.
> Fanning Software Consulting, Inc.
> Coyote's Guide to IDL Programming:http://www.dfanning.com/
> Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: 3D UserSym?
Posted by David Fanning on Mon, 10 Aug 2009 13:39:05 GMT
View Forum Message <> Reply to Message

Brian Daniel writes:

> I put in some time and effort on learning the basics of objects and
> object graphics. I relied heavily on Scatter_Plot.pro, an example
> from a previous post on this thread. I've run across some challenges.
>
> 1) I have a memory leak

Yikes! I guess so. :-(

It appears half the cleanup routine has gone missing in
that Scatter_Surface routine I gave to you last time.
You can download a corrected version here:

http://www.dfanning.com/tip_examples/scatter_surface.pro

In particular, no one was cleaning up any of the objects
in the info structure in the previous version. :-(

Obj_Destroy, info.thisPrinter, info.thisWindow, info.thisPolyline
Obj_Destroy, info.thisTrackball, info.thisModel
Obj_Destroy, info.xaxis, info.yaxis, info.zaxis

Cheers,

David

P.S. I'm pretty sure Coyote was working on this late last
week. I'll have a word with him.

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: 3D UserSym?
Posted by penteado on Mon, 10 Aug 2009 19:00:36 GMT
View Forum Message <> Reply to Message

On Aug 10, 10:22 am, Brian Daniel <Daniels...@yahoo.com> wrote:
> 2) One axis of the scatter plot is of a much different scale (2 order
> of magnitude different). The 'orb' object is set with a radius value
> in data coordinates, so my orbs look like discs... getting me back to
> the same problem I had to begin with (2D symbols).

The orb object inherits IDLgrModel. So you can use its scale method to
turn the orbs from spheres to ellipsoids, as in

osph->scale,sx,sy,sz

With the 3 arguments being the scale factor in directions x,y,z, which
are initially 1.0.

Similarly, if you want to change their orientation, you can use the
rotate method.

Subject: Re: 3D UserSym?
Posted by penteado on Mon, 10 Aug 2009 19:46:32 GMT
View Forum Message <> Reply to Message

On Aug 10, 10:22 am, Brian Daniel <Daniels...@yahoo.com> wrote:
> 3) This is a less important issue, but still worth mentioning. I have
> a lot of data to plot, and it looks like a big cloud of points. There
> are 3 data points that are important to see in the context of the
> whole data set. Using direct graphics, I could just over plot the 3
> points and always be able to see them. How can you do that in object
> graphics? Is there a way to set my full data set to be partially
> transparent?

You can do it either way (making them always appear, or making the others transparent). You need to create a new orb object to be used for those special points, and overplot those with the new object.

Returning to the example I gave above, to make the special points show over the others:

```
x=randomu(seed,100) & y=randomu(seed,100) & z=randomu(seed,100)
sel=[1,3,5,7] ;indexes of the points to make special
osph=obj_new('orb') ;sphere object for the regular points
s_osph=obj_new('orb') ;sphere object for the special points
osph->setproperty,color=[255,255,0],radius=2.0
s_osph->setproperty,color=[255,0,0],radius=3.0
iplot,x,y,z,/scatter,sym_object=osph;plot the regular points
 iplot,x[sel],y[sel],z[sel],/scatter,sym_object=s_osph,/over; plot the
special points
s_osph->setProperty,depth_test_function=8;make the special points
always appear over the others
```

The property dept_test_function is also inherited from IDLgrModel, see the documentation for more details.

You can make the regular spheres transparent, but this makes drawing much slower, (probably too slow if you have a cloud of points) since instead of just figuring out which object goes on top, it is necessary to calculate the result of seeing one through the other. Anyway, it is done changing the alpha_channel of the orbs, which they inherit from IDLgrPolygon, and defaults to 1.0 (opaque):

```
iplot,x[sel],y[sel],z[sel],/scatter,sym_object=s_osph;plot the special
points
iplot,x,y,z,/scatter,sym_object=osph,/over;plot the regular points
osph->setproperty,alpha_channel=0.5;make the regular points
semitransparent
```