Subject: Re: "Object Graphics and Vectors" Reloaded
Posted by David Fanning on Wed, 04 Aug 2004 18:06:38 GMT
View Forum Message <> Reply to Message

Ralf Schaa writes:

> okay, I thought 'well, before taking David's hint I give it a try
> myself, this can't be soo hard ...'
>
> This is what I have:
> a nice 3d object plot in data-coordinates with all rangings from -2.5e4
> to 2.5e4, so far so good.
> Now I add a vector with Rick Towler's vector object, which needs as
> input a 'location' (the startpoint of the vector) and a 'magnitude' (the
> endpoint), these are defined in normalized coordinates.
> Okay, all i have to do is give the location and the magnitude in values
> which are near my ranging-values, actually I calculated them with the
> formula taking from the idl help:
>
> NormX = -range[0]/(range[1]-range[0]) + 1/(range[1]-range[0]) * DataX
>
> I need the other way round:
>
> DataX = NormX * ((range[1]-range[0])) + range[0]
>
> e.g a vector in normalized coordinates from [0,0,0] to [1,1,1]
> would be in datacoordinates
> from [range[0],range[0],range[0]]
> to [3*range[1],3*range[1],3*range[1]]
>
> but nothing exciting happened ...
>
> so where is my bug ?

I don't know. You are confusing me. :-)

Here is what I would do. First, figure out
how big a "unit" vector is in your coordinate
system. Perhaps it is 0.05 of the length of the
universe:

```
unitLen = 2.5e4 * 2 * 0.05
```

Then find out the current "range" of the vector
object:

```
vector -> GetProperty, XRange=xr, YRange=yr, ZRange=zr
```

Suppose the vector you wish to draw is 3.2 times the
size of a unit vector.

```
  xs = Normalize(xr, Position=[0, 3.2*unitLen])
  ys = Normalize(yr, Position=[0, 3.2*unitLen])
  zs = Normalize(zr, Position=[0, 3.2*unitLen])
```

Now, scale the vector:

```
  vector -> SetProperty, XCoord_Conv=xs, $
YCoord_Conv=ys, ZCoord_Conv=zs
```

Without something to *see*, I don't have a great deal
of confidence in this solution. But I think it has to
be *something* like that. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
