Subject: Re: Drawing 3D Objects properly
Posted by Rick Towler on Thu, 05 Sep 2002 18:33:35 GMT
View Forum Message <> Reply to Message

Yes, xobjview is a *very* useful tool. Too bad it doesn't get any ink.

Your problem must be a Mac thing? It looks good to me with either projection. Attached is an image showing your programs output. Nix that image, my news server won't let me post it (sheesh, it is only 8k). But what I did see looked correct. I can send the image off list if you wish.

{x86 Win32 Windows Microsoft Windows 5.5 Aug 28 2001 32 64}

-Rick

"M. Katz" <MKatz843@onebox.com> wrote in message news:4a097d6a.0209051008.6857593@posting.google.com...
> Thanks for the tip to use xobjview! In xobjview, I saw that the 3D
> objects behaved as they are supposed to. This led me to tweak all of
> the settings on my IDLgrView, until I found the culprit.
>
> The PROJECTION keyword is supposed to change from orthogonal to a
> near/far view. In IDL 5.4 (at least), I see that it does more than
> that. If you run my simple side-by-side test script below, you'll see
> the issue.
>
> With PROJECTION=1, one polygon successfully hides the one behind it.
> With PROJECTION=2, the red square (first drawn) *always* blocks the
> green square.
>
> Run the test script below by copying it into a new file, name and save
> the file, and then .run the file. Move the mouse around in the IDL 0
> window (the third window on screen, possibly on the right of the other
> two). You can control the rotation direction and velocity. Click in
> that window to quit.
>
> I'm very interested to know if this is platform speciic or not. (I'm
> running IDL in Classic mode on MacOS 10.1.5)
>
> M. Katz
> (Script Follows)
>
> ;---
>
> ; object-viewer test script demonstrating projection diffrerences

```
>
>    ;--- create two identical windows
>  mywindow1 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
>     LOCATION=[50,50])
>  mywindow2 = obj_new('IDLgrWindow', quality=2, dimensions=[200,200], $
>     LOCATION=[50,300])
>    ;--- create two views with different PROJECTION values
>  view1 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $
>     color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=1)
>  view2 = obj_new('IDLgrView',EYE=1000, zclip=[999,-999], $
>     color=[0,0,0], VIEWPLANE_RECT = [-1,-1,2,2]*2, PROJECTION=2)
>
>    ;--- Create two identical models: A red and a green square with
>  different z values
>  model1 = obj_new('IDLgrModel')
>  model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]-0.5, COLOR=[200,50,50])
>  model1 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]+0.5, COLOR=[50,200,50])
>
>  model2 = obj_new('IDLgrModel')
>  model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]-0.5, COLOR=[200,50,50])
>  model2 -> Add, obj_new('IDLgrPolygon', [-1,1,1,-1], [1,1,-1,-1], $
>    [0,0,0,0]+0.5, COLOR=[50,200,50])
>
>  view1 -> Add, model1
>  view2 -> Add, model2
>
>    ;--- Allow the user to manipulate the graphics as they are displayed
>  window, xsize=200, ysize=200
>
>  repeat begin
>    cursor, xc, yc, /normal, /nowait
>
>    model1 -> Rotate, [0,1,0], (xc-0.5)*5
>    model1 -> Rotate, [1,0,0], (yc-0.5)*5
>    model2 -> Rotate, [0,1,0], (xc-0.5)*5
>    model2 -> Rotate, [1,0,0], (yc-0.5)*5
>
>    mywindow1 -> Draw, view1
>    mywindow2 -> Draw, view2
>  endrep until (!mouse.button GT 0)
>
>  end
```