Subject: Re: Depth visibility with Object Graphics !!!
Posted by Karl Schultz on Wed, 26 May 2004 14:30:31 GMT
View Forum Message <> Reply to Message

"Antonio Santiago" <d6522117@est.fib.upc.es> wrote in message
news:40B239EE.10308@est.fib.upc.es...
> Hi,
>
> i am working with Object Grpahics and i hace a problem :)
>
> I have put on a model two objects. Firts, i must put an IDLgrPolyline
> betwen two points (a line) that i can modify to select one trajectory.
> Second i put an IDLgrImage.
> My problem is that image is drawing over the polyline. I play with
> DEPTH_TEST_DISABLE and DEPT_TEST_FUNCTION, but always the image is
> drawing over the polyline.
>
> Any idea?

The IDLgrImage does not interact with the depth buffer as it is drawn. So, if you draw the image after the polylines, the image will overwrite the polylines where they overlap, independent of the depth of the polylines. If you draw the image first, you will always see the lines. In general, it is usually better to arrange scenes containing images so that the images get drawn first.

The reason behind this is that IDLgrImage objects are drawn with an OpenGL pixel primitive (glDrawPixels), which is a special 2D image drawing primitive. The intent of IDLgrImage is to provide a "billboarding" style image drawing primitive, so the usage of this primitive makes sense.

The discussion in this thread about using a texture-mapped polygon gives you the flexibility of drawing the image in space whereever you want with the full interaction with the depth buffer if that is what you would expect. So, you have both methods available to you in IDL.

The IDL (6.1) docs say:

Image objects do not take into account the Z locations of other objects that may be included in the view object. This means that objects that are drawn to the destination object (window or printer) after the image is drawn will appear to be in front of the image, even if they are located at a negative Z value (behind the image object). Objects are drawn to a destination device in the order that they are added (via the Add method) to the model, view, or scene that contains them. To rotate or position image objects in three-dimensional space, use the IDLgrPolygon object with texture mapping enabled.