Subject: Re: Displaying three images simultaneously (using Object Graphics)
Posted by Dick Jackson on Tue, 08 Nov 2005 22:00:45 GMT
View Forum Message <> Reply to Message

"Dick Jackson" <dick@d-jackson.com> wrote in message
news:PZ7cf.441815$1i.267001@pd7tw2no...
> "Karl Schultz" <k____schultz@rsinc.com> wrote in message
> news:pan.2005.11.08.19.05.56.672000@rsinc.com...
>
>> I should point out that starting with IDL 6.2, IDL renders images using
>> texture-mapped polygons, doing some of the steps above for you. Further,
>> there is a new TRANSFORM_MODE property that will treat the image as a
>> polygon during transforms, instead of just transforming the opposite
>> corners and making a new 2D box from the new corner locations.
>
> Well, how about that. Thanks for the tip, Karl. My new example would then
> be:
>
> im1=Obj_New('IDLgrImage', BytScl(BIndGen(3, 10, 10)), Location=[-10,10], $
> Transform_Mode=1)
> im2=Obj_New('IDLgrImage', BytScl(RandomU(seed, 3, 10, 10)), $
> Transform_Mode=1)
> im3=Obj_New('IDLgrImage', 255B-BytScl(BIndGen(3, 10, 10)),
> Location=[10,10], $
> Transform_Mode=1)
> myModel = Obj_New('IDLgrModel')
> myModel->Add, [im1, im2, im3]
> XObjView, myModel
>
> Another advantage of this is that we get away from the IDLgrPolygon
> texture map's required use of image sizes that are a power of two (the
> images were resampled to 16x16 in my previous example)

Someone else at RSI has kindly cautioned me that the IDLgrImage won't give exactly the same behaviour as a texture-mapped IDLgrPolygon. The Image objects don't really "live" in the same 3-D space as Polygon objects. Rather, they are just drawn into the view in sequence along with the other objects, depending on the order they were added.

This example (images all at Z=0, polygons at Z=-10, 0 and 10) shows the rather bewildering appearance that mixing these objects in one view can give:

```
im1=Obj_New('IDLgrImage', BytScl(BIndGen(3, 10, 10)))
im2=Obj_New('IDLgrImage', BytScl(RandomU(seed, 3, 10, 10)))
im3=Obj_New('IDLgrImage', 255B-BytScl(BIndGen(3, 10, 10)))
poly1=Obj_New('IDLgrPolygon',
[[-10,10,-10],[0,10,-10],[0,20,-10],[-10,20,-10]], $
```

```
        Color=[255,255,255], Texture_Map=im1, $
        Texture_Coord=[[0,0],[1,0],[1,1],[0,1]])
poly2=Obj_New('IDLgrPolygon', [[0,0],[10,0],[10,10],[0,10]], $
        Color=[255,255,255], Texture_Map=im2, $
        Texture_Coord=[[0,0],[1,0],[1,1],[0,1]])
poly3=Obj_New('IDLgrPolygon', [[10,10,10],[20,10,10],[20,20,10],[10,20,10]],
$
        Color=[255,255,255], Texture_Map=im3, $
        Texture_Coord=[[0,0],[1,0],[1,1],[0,1]])
im4=Obj_New('IDLgrImage', BytScl(BIndGen(3, 10, 10)), Location=[-10,10], $
    Transform_Mode=1)
im5=Obj_New('IDLgrImage', BytScl(RandomU(seed, 3, 10, 10)), $
    Transform_Mode=1)
im6=Obj_New('IDLgrImage', 255B-BytScl(BIndGen(3, 10, 10)), Location=[10,10],
$
    Transform_Mode=1)
myModel = Obj_New('IDLgrModel')
myModel->Add, [poly1, poly2, poly3, im4, im5, im6] ; Images show up in front
;myModel->Add, [im4, im5, im6, poly1, poly2, poly3] ; Polygons show up in
front
XObjView, myModel
```

So, if you want images to look as if they are really in the same 3-D space as other geometric objects, use texture-mapped IDLgrPolygons (or IDLgrSurfaces, I suppose).

Cheers,
--
-Dick

Dick Jackson / dick@d-jackson.com
D-Jackson Software Consulting / http://www.d-jackson.com
Calgary, Alberta, Canada / +1-403-242-7398 / Fax: 241-7392