Subject: Re: Transparent texture mapped polygons
Posted by lyubo on Mon, 25 Mar 2002 18:15:55 GMT
View Forum Message <> Reply to Message

Hi Rick,

First, thanks again for your response.

I have studied David's example about transparent images and also the example in "What's new in IDL 5.5", but I still couldn't get it to work with texture mapped polygons. It works fine if I try to blend two images, but it doesn't work with polygons.

I've been trying to overlay two images and draw them in 3D. To do that I create two texture mapped planes (with the images as texture) but I have problems
making one of the planes transparent. I create an "alpha" image, as in David's example,
for the foreground plane hoping that it will be blended with the background image (the other
plane) but it doesn't work because when I add the background plane to the model and then
the foreground plane (the one with the alpha channel), the foreground plane will
not show up on the screen. It is there, but is covered by the background plane.

I was trying different things and I noticed that if I switch the order of adding the planes to the model (first the "alpha" plane and then the background plane),
both planes will be displayed, but the foreground plane isn't transparent, rather it is
drawn over the background plane.

Below is the code that I used to test this, a part of it comes from David's example.

If you see what's wrong I would really appreciate your help.

Thanks,

Lyubo

```
;******************************************************** ****************
********
oImage=OBJ_NEW('IDLgrImage',(*volume.image)[*,*,depth/2])
oPlane = OBJ_NEW('IDLgrPolygon', verts,
 COLOR=[255,255,255],TEXTURE_COORD=[[0,0],[1,0],[1,1],[0,1]], $
```

```
                    TEXTURE_MAP=oImage, XCOORD_CONV=xs,
YCOORD_CONV=ys, ZCOORD_CONV=zs)

foregroundImage=(*volume.image)[*,*,20]
s = Size(foregroundImage, /Dimensions)
alpha_image = BytArr(4, s[0], s[1])

(*pTlbState).oPalette[2]->LoadCT, 3
(*pTlbState).oPalette[2]->GetProperty, red_values=r, green_values=g,
blue_values=b
alpha_image[0,*,*]=r[foregroundImage]
alpha_image[1,*,*]=g[foregroundImage]
alpha_image[2,*,*]=b[foregroundImage]
    ; Pixels with value 0 with be totally transparent.
    ; Other pixels will start out half transparent.
blendMask = BytArr(s[0], s[1])
blendMask[Where(foregroundImage GT 10)] = 1B
alpha_image[3, *, *] = blendMask * 128B

alphaImage = OBJ_NEW('IDLgrImage',alpha_image, INTERLEAVE=0,
BLEND_FUNCTION=[3,4], PALETTE=(*pTlbState).oPalette[2])
oBlendPlane = OBJ_NEW('IDLgrPolygon', verts, COLOR=[255,255,255],
TEXTURE_COORD=[[0,0],[1,0],[1,1],[0,1]], $
  TEXTURE_MAP=alphaImage, TEXTURE_INTERP=1, XCOORD_CONV=xs,
YCOORD_CONV=ys,
ZCOORD_CONV=zs)

(*pDisplayState).oModel[1,2]->Add, oPlane
(*pDisplayState).oModel[1,2]->Add, oBlendPlane
```