
Subject: Re: scripted object graphics question

Posted by [George Millward](#) on Fri, 25 Feb 2011 22:59:36 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Feb 25, 12:31 pm, kBob <krd...@gmail.com> wrote:

> On Feb 25, 6:32 am, George Millward <george.millw...@noaa.gov> wrote:

>

>

>

>> Hi there,

>

>> I am trying to convert a bunch of old direct graphics programs to

>> objects.

>> These programs are all called by a cronjob/script so work in the

>> background (no interactivity or windows produced).

>

>> In essence they use this:

>

>> set_plot,'z'

>> write_png

>

>> I now realize that I've never seen an example of object graphics

>> working in a similar way - all object

>> examples seem to be embedded in widgets and the like - based on human

>> interactivity.

>

>> I assume I can do this in object graphics ? Does anyone have a simple

>> example of how this is done ?

>

>> Cheers

>

>> George.

>

> Below is an example using IDLgrBuffer.

>

> Kelly Dean

> Milliken, CO

>

> ;+

> ;

> ; @file_comments

> ; <P>Use object graphics to draw polygon with IDLgrBuffer as the

> ; destination object.

> ;

> ;-----

> PRO BufferOBJ

>

> ; Prepare a polygon (Hypocycloid)

```

>
> scale = 2.0
> cusp = 3.0
> direct = 45.0 * !DTOR
>
> xrad = INDGEN(360) * !DTOR
> yrad = INDGEN(360) * !DTOR
>
> x = scale * ( ( ( cusp - 1.0 ) * COS( xrad ) - direct * $
>             COS( ( cusp - 1.0 ) * xrad ) ) )
> y = scale * ( ( ( cusp - 1.0 ) * SIN( xrad ) + direct * $
>             SIN( ( cusp - 1.0 ) * xrad ) ) )
>
> deltoid = FLTarr( 2, 360 )
> deltoid[0,*] = TEMPORARY(y)
> deltoid[1,*] = TEMPORARY(x)
>
> ; Add polygon to object graphics.
>
> vp_rec = [ -15.0, -15.0, 30.0, 30.0 ]
> navy = [0,0,128]
> dims = [80,80]
>
> oDest = OBJ_NEW( 'IDLgrBuffer' )
> oDest -> SetProperty, DIMENSIONS = dims
> oView = OBJ_NEW( 'IDLgrView', VIEWPLANE_REC = vp_rec )
> oModel = OBJ_NEW( 'IDLgrModel' )
> oEvolute = OBJ_NEW( 'IDLgrPolygon', COLOR = navy )
> oEvolute -> SetProperty, DATA = deltoid
> oModel -> ADD, oEvolute
> oView -> ADD, oModel
> oDest -> DRAW, oView
>
> ; Retrieve image from destination object.
>
> oDest -> GetProperty, IMAGE_DATA = img3
>
> ; Heap variable clean up.
>
> OBJ_DESTROY, oEvolute
> OBJ_DESTROY, oModel
> OBJ_DESTROY, oView
> OBJ_DESTROY, oDest
>
> ; Save image as JPEG
>
> outfile = 'c:\temp\cycloid.jpg'
> WRITE_JPEG, outfile, img3, TRUE=1, QUALITY=100

```

```
>  
> PRINT, STRING( outfile, FORMAT ='( " -- All done : ", A0 )' )  
>  
> END
```

Wow - nailed it !!

I used the `oDest = OBJ_NEW('IDLgrBuffer')` as detailed above by Kelly.....

There was still an X-windows problem coming from TVLCT so I was thinking of the Xvfb solution

from Karl - but I realized that I didn't need TVLCT once you've decided on an r, g and b then it's just:

```
thisPalette=Obj_New('IDLgrPalette')  
thisPalette -> setProperty, red_values = r  
thisPalette -> setProperty, green_values = g  
thisPalette -> setProperty, blue_values = b  
... and that doesn't seem to need any X-window
```

...and it works !!!

Thanks to everyone for their help. Apologies if I'm a little over-exuberant but you know how it is when it comes together!... Like I said above - our basic business is IDL graphics running in the background on LINUX... and I'd been telling everyone that objects was the way to go... so I was getting a little anxious ;o) - and now it's just about beer-o'clock on a Friday - perfect timing !!

Cheers

George.
