Subject: Re: IDLgrPolygon question
Posted by Dick Jackson on Fri, 19 Dec 2014 06:56:11 GMT
View Forum Message <> Reply to Message

nata wrote, On 2014-12-18, 5:25am:
> Hi guys,
>
> I am wondering if it is possible to create the following image with ONLY 1 instance of an IDLgrPolygon and without repeating vertices...
> http://pdroms.de/wp-content/uploads/2012/04/ColorSquares.png
>
> Create the polygons and their connectivity it's not difficult. What I don't see is how to associate different colors to each square.
> I don't know how the VERT_COLORS property would work since almost all vertices are shared between different polygons.
>
> Thank you for your help in advance,
> Bernat

Hi Bernat,

First point to make: IDLgrPolygon (and IDLgrSurface) can give you the colored quadrilaterals (quads) with Style=2, but
if you need the black lines, you need a second "grid" object, rendered slightly in front of the colored object. I'll
show you how.

If you're OK with IDLgrSurface, it's very easy: each quad takes color of its lower-left vertex.
If you need IDLgrPolygon, each quad takes color of lower-left vertex. (this makes some colored triangle meshes
impossible, but for this it will work out fine)

Compile and run this program to see both ways of doing this. I hope my comments here give a good enough explanation:

=====

```
PRO SurfaceAndPolygonVertColorsTest

;;    Set up the colors for the blocks

blockDims = [16, 14]                          ; [nCols, nRows]
blockColors = BytScl(RandomU(seed, [3, blockDims])) ; [3, nCols, nRows] RGB

;;    Set up the colors for the vertices (with an extra row and column of
;;    colors that will be ignored)
```

```
vertDims = blockDims + [1, 1]
vertColors = BytArr([3, vertDims])  ; Make array with extra "row" and "column"
vertColors[0, 0, 0] = blockColors   ; Lay blockColors into vertColors
vertColors = Reform(vertColors, [3, Product(vertDims)], /OVERWRITE)

;;    Make string for title to confirm lower-left and upper-right block colors

colorStr = String(Format='(" LL=[",I0,",",I0,",",I0,"] '+ $
                          'UR=[",I0,",",I0,",",I0,"]")', $
                 blockColors[*,0,0], blockColors[*,-1,-1])

;;    With IDLgrSurface, each quad takes color of lower-left vertex

oSurface = IDLgrSurface(FltArr(vertDims), Style=2, VERT_COLORS=vertColors, $
                        DEPTH_OFFSET=1)
oSurfaceGrid = IDLgrSurface(FltArr(vertDims), Style=1, $
                          DEPTH_OFFSET=0)
XObjView, [oSurface, oSurfaceGrid], Title='IDLgrSurface'+colorStr, $
   STATIONARY=Obj_New('IDLgrLight')     ; A "null light" to display pure colors

;;    With IDLgrPolygon, each quad takes color of first vertex (Mesh_Obj lists
;;    lower-left vertex first)

Mesh_Obj, 1, verts, conn, FltArr(vertDims)
oPolygon = IDLgrPolygon(verts, Polygons=conn, Style=2, VERT_COLORS=vertColors,$
                        DEPTH_OFFSET=1)
oPolygonGrid = IDLgrPolygon(verts, Polygons=conn, Style=1,$
                          DEPTH_OFFSET=0)
XObjView, [oPolygon, oPolygonGrid], Title='IDLgrPolygon'+colorStr, $
   STATIONARY=Obj_New('IDLgrLight'), $  ; A "null light" to display pure colors
   XOFFSET=420

END

=====
```

Cheers,
-Dick

Dick Jackson Software Consulting Inc.
Victoria, BC, Canada
www.d-jackson.com