Subject: Re: IDLgrPolygon Leak? Posted by David Fanning on Wed, 03 Mar 2010 23:56:37 GMT

View Forum Message <> Reply to Message

kBob writes:

>

> Not a memory leak, but a polygon leak.

>

- > The 2010 IDL User Group meeting got me inspired to work with some
- > object graphics mapping.

>

- > However, I am running into some problems with adding a Shapefile to a
- > object graphics window. When I call the IDLgrPOLYGON to draw and fill
- > the shapefile vertices, there seems to be a leak or overflowing the
- > polygon.

>

- > Any thoughts on why. Sample code below. The IDLgrPOLYGON is set to
- > work with vertices that only have one part. Anything else, the
- > IDLgrPOLYLINE is used.

Kelly, you probably already figured this out, but XObjView doesn't destroy the object that is handed to it. You have to do that yourself. All this memory leaking is just due to the model object you created. For example, you can change your main routine like this.

PRO Test_PolyShp

COMPILE_OPT DEFINT32, STRICTARR

CountryModel, oModelCntry XOBJVIEW, oModelCntry, /BLOCK Obj_Destroy, oModelCntry

;WARNING: Big time memory leak ;Do IDL> HEAP_GC, /Verbose

END

--

David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Sepore ma de ni thui. ("Perhaps thou speakest truth.")

View Forum Message <> Reply to Message

```
On Mar 3, 1:24 pm, kBob < krd...@gmail.com > wrote:
 Not a memory leak, but a polygon leak.
>
  The 2010 IDL User Group meeting got me inspired to work with some
> object graphics mapping.
>
 However, I am running into some problems with adding a Shapefile to a
> object graphics window. When I call the IDLgrPOLYGON to draw and fill
> the shapefile vertices, there seems to be a leak or overflowing the
> polygon.
>
 Any thoughts on why. Sample code below. The IDLgrPOLYGON is set to
> work with vertices that only have one part. Anything else, the
> IDLgrPOLYLINE is used.
>
> Kelly Dean
> Fort Collins, CO
>
 ======== Test PolyShp ========
>
> ;+
 ; <P>Prepare Shapefile Entities for object model.
>
> PRO CountryModel, oModelSHP
>
 COMPILE OPT DEFINT32, STRICTARR
>
> shpFile = FILEPATH( Subdirectory=['resource', 'maps', 'shape'],
> 'cntry02.shp')
> oSHP = OBJ_NEW( 'IDLffShape', shpFile )
> IF ( OBJ_VALID( oSHP ) ) THEN BEGIN
  oModelSHP = OBJ_NEW( 'IDLgrModel' )
>
   oSHP -> GetProperty, N_ENTITIES = num_ent
>
   FOR entn = 0, num_ent-1 DO BEGIN
    ent = oSHP -> GetEntity( entn )
>
    IF (ent.n parts GE 2) THEN BEGIN
>
     cuts = [ (*ent.parts), ent.n vertices ]
>
     FOR partn = 0, ent.N_parts-1 DO BEGIN
>
       oGon = OBJ NEW('IDLgrPolygon', (*ent.vertices)[*,
> cuts[partn]:cuts[partn+1]-1])
       oGon = OBJ_NEW('IDLgrPolyline', (*ent.vertices)[*,
> cuts[partn]:cuts[partn+1]-1])
       oGon -> SetProperty, COLOR = [034, 139, 87]; Forest Green
```

```
oModelSHP -> ADD, oGon
>
     ENDFOR
>
    ENDIF ELSE IF (ent.n_parts EQ 1) THEN BEGIN
>
      oGon = OBJ_NEW('IDLgrPolygon', (*ent.vertices) )
>
      oModelSHP -> ADD, oGon
>
    ENDIF ELSE BEGIN
>
    ENDELSE
    oSHP -> DestroyEntity, ent
 ENDFOR
  OBJ DESTROY, oSHP
> ENDIF ELSE BEGIN
> ENDELSE
>
> END
> ;+
> PRO Test_PolyShp
 COMPILE OPT DEFINT32, STRICTARR
> CountryModel, oModelCntry
> XOBJVIEW, oModelCntry
> ;WARNING: Big time memory leak
> ;Do IDL> HEAP_GC, /Verbose
> END
```

IDLgrPolygons need to be convex in order to render properly. (This is the case for the underlying OpenGL as well)

I'm guessing that is what you mean by "leak". Drawing non-convex polygons with grPolygon can sort of look like a leak.

It does not look like your code takes any steps to ensure that the polygons are convex. It may be the case that your database consists of only convex shapes, but I don't know that.

To solve this, look at IDLgrTesselator. It will take an arbitrary input polygon and emit a covering set of triangles that can then be stored in IDLgrPolygon.

Subject: Re: IDLgrPolygon Leak?

Posted by KRDean on Thu, 04 Mar 2010 04:00:28 GMT

```
On Mar 3, 5:10 pm, Karl <karl.w.schu...@gmail.com> wrote:
> On Mar 3, 1:24 pm, kBob < krd...@gmail.com > wrote:
>
>
>
>
>
   Not a memory leak, but a polygon leak.
   The 2010 IDL User Group meeting got me inspired to work with some
>> object graphics mapping.
>
>> However, I am running into some problems with adding a Shapefile to a
>> object graphics window. When I call the IDLgrPOLYGON to draw and fill
>> the shapefile vertices, there seems to be a leak or overflowing the
>> polygon.
>
>> Any thoughts on why. Sample code below. The IDLgrPOLYGON is set to
>> work with vertices that only have one part. Anything else, the
>> IDLgrPOLYLINE is used.
>> Kelly Dean
>> Fort Collins, CO
>
>> ;+
>>
>> : <P>Prepare Shapefile Entities for object model.
>> :
>> PRO CountryModel, oModelSHP
>> COMPILE OPT DEFINT32, STRICTARR
>
>> shpFile = FILEPATH( Subdirectory=['resource', 'maps', 'shape'],
>> 'cntry02.shp' )
>> oSHP = OBJ_NEW( 'IDLffShape', shpFile )
>> IF ( OBJ_VALID( oSHP ) ) THEN BEGIN
   oModelSHP = OBJ NEW( 'IDLgrModel' )
    oSHP -> GetProperty, N ENTITIES = num ent
>>
   FOR entn = 0, num_ent-1 DO BEGIN
>>
     ent = oSHP -> GetEntity( entn )
>>
     IF (ent.n_parts GE 2) THEN BEGIN
>>
      cuts = [ (*ent.parts), ent.n_vertices ]
>>
      FOR partn = 0, ent.N_parts-1 DO BEGIN
>>
        oGon = OBJ_NEW('IDLgrPolygon', (*ent.vertices)[*,
>> ;
```

```
>> cuts[partn]:cuts[partn+1]-1])
        oGon = OBJ_NEW('IDLgrPolyline', (*ent.vertices)[*,
>>
>> cuts[partn]:cuts[partn+1]-1])
        oGon -> SetProperty, COLOR = [034, 139, 87]; Forest Green
        oModelSHP -> ADD, oGon
>>
       ENDFOR
>>
      ENDIF ELSE IF (ent.n_parts EQ 1) THEN BEGIN
>>
        oGon = OBJ_NEW('IDLgrPolygon', (*ent.vertices) )
>>
        oModelSHP -> ADD, oGon
>>
      ENDIF ELSE BEGIN
>>
     ENDELSE
>>
   oSHP -> DestroyEntity, ent
>> ENDFOR
>> OBJ_DESTROY, oSHP
>> ENDIF ELSE BEGIN
>> ENDELSE
>> END
>> ;+
>> ;
>>
>> PRO Test_PolyShp
>> COMPILE_OPT DEFINT32, STRICTARR
>
>> CountryModel, oModelCntry
>> XOBJVIEW, oModelCntry
>
>> ;WARNING: Big time memory leak
>> ;Do IDL> HEAP GC, /Verbose
>
>> END
> IDLgrPolygons need to be convex in order to render properly. (This is
  the case for the underlying OpenGL as well)
> I'm guessing that is what you mean by "leak". Drawing non-convex
  polygons with grPolygon can sort of look like a leak.
> It does not look like your code takes any steps to ensure that the
> polygons are convex. It may be the case that your database consists
 of only convex shapes, but I don't know that.
>
> To solve this, look at IDLgrTesselator. It will take an arbitrary
> input polygon and emit a covering set of triangles that can then be
> stored in IDLgrPolygon.- Hide quoted text -
>
```

> - Show quoted text -

Thanks Karl,

This is what I was looking for. Apparently, there are "holes" in the vertices and it appears IDLgrTesselator will help close them up.

Kelly Dean Fort Collins, CO