Subject: Bug in IDLgrPolygon ?
Posted by Thomas Gutzler on Thu, 12 Dec 2002 10:18:48 GMT
View Forum Message <> Reply to Message

Hi,

I just detected some weird behaviour in IDLgrPolygon. I generated an Object with this data:

```
 arr = Transpose([[-1, 1,-1, 1,-1, 1,-1, 1], $
                  [-1,-1,-1,-1, 1, 1, 1, 1], $
                  [-1,-1, 1, 1,-1,-1, 1, 1]])
  ;vertex:   1  2  3  4  5  6  7  8
 arr_colors = Transpose([[255,255,  0,  0,255,127,  0,127], $
                         [190,  0, 65,255,  0,  0,255,255], $
                         [  0,190,255, 65,  0,255,255,  0]])
 arr_polys = [[4, 0, 1, 3, 2], $ ; vertices 1243
              [4, 0, 2, 6, 4], $ ; vertices 1375
              [4, 2, 3, 7, 6], $ ; vertices 3487
              [4, 1, 3, 7, 5], $ ; vertices 2486
              [4, 0, 1, 5, 4], $ ; vertices 1265
              [4, 4, 5, 7, 6]]   ; vertices 5687
 poly = OBJ_NEW('IDLgrPolygon', arr, polygons=arr_polys,
vert_color=arr_colors, SHADING=1)
```

which is a colored cube.

The onlinehelp to IDLgrPolygon sais:

 POLYGONS (Get, Set)
 [...] To ignore an entry in the POLYGONS array, set the vertex count, n, to 0.

Fine - I set Element 25 (the first of the last line) from 4 to 0 an looked forward to see an "open cube" beacuse the last polygon shouldn't be drawn.
What I saw was an open cube ... and an error:

 % OBJ_NEW: Error, invalid connectivity list detected (invalid final polygon).

If I just leave the last (or any other) line away there's no error and the result is, what I expeced, an open cube. But this can't be the solution, because I want to interactively permit polygons being drawn.

My question is: Is this a bug ?
or: what did I wrong ?

Thanks,

Thomas

Subject: Re: Bug in IDLgrPolygon ?
Posted by Thomas Gutzler on Mon, 16 Dec 2002 05:28:24 GMT
View Forum Message <> Reply to Message

Thomas Gutzler wrote:
>
> Karl Schultz wrote:
>>
>> If it were me, I'd think about just copying the disabled polygon out to
>> another data structure, along with its starting index. Then, I zap the poly
>> in the original list to zero. I copy the polygon back to the same place to
>> reenable it.
>
> I'm going to do this.

This leads me directly to my next question.
Is it possible to select a single polygon of a IDLgrPolygon-Model with
the mouse ?
I found the 'SegmentLayout.pro' wich does something like this. But if I
understood the program right it generates an own model for every segment
and I don't want to generate 100000s of models (one for every polygon of
the surface).
XObjView has a nice function (Select) that shows the type of a Model if
klicked with the mouse. Perhaps this could be enhanced to return the
koordinates (or better, their indexes) of a polygon of a polygon-model
but unfortunately I couldn't find the code in xobjview.pro :(

At last I want to view the model, select a polygon by mouse which won't
be shown till it's selected again.

ideas ?

Thanks in advance,
Tom

Subject: Re: Bug in IDLgrPolygon ?
Posted by Karl Schultz on Mon, 16 Dec 2002 16:08:11 GMT
View Forum Message <> Reply to Message

"Thomas Gutzler" <tgutzler@ee.uwa.edu.au> wrote in message
news:3DFD6478.1B64C1D6@ee.uwa.edu.au...
> This leads me directly to my next question.
> Is it possible to select a single polygon of a IDLgrPolygon-Model with

> the mouse ?

The Select method can return object references to atomic graphic objects like IDLgrPolygon. IDLgrPolygon objects can contain lists of independent and/or "meshed" polygons. If any one of the polygons contained in an IDLgrPolygon object is selected, Select returns the object reference for that IDLgrPolygon object.

I think that you want to be able to determine which polygon within an IDLgrPolygon object is selected?

After the call to Select, use the PickData method to get the location of the selection point in model coordinates. You can then compare this point against all of your polygons in that object to see which polygon contains that point. Right now, there's no way for IDL to tell you directly.

> I found the 'SegmentLayout.pro' wich does something like this. But if I
> understood the program right it generates an own model for every segment
> and I don't want to generate 100000s of models (one for every polygon of
> the surface).

You don't need a model for each polygon. I think that there are two reasonable choices:

1) Make an IDLgrPolygon object for each polygon (or segment, if I understand your terminology). This makes selection easy, since Select will give you an object that contains only one polygon (segment).

2) Put multiple polygons in IDLgrPolygon objects and use Select, followed by PickData to determine the selected polygon (segment).

The first method is the easiest, but might take up a lot of space if there are lot of segments. IDLgrPolygons aren't that expensive, so there would have to be a LOT of them to make this method unreasonable.

If your "segments" form a mesh and you want to display them with smooth shading so that shared vertices use averaged normals, you're forced to put all the segments in a single IDLgrPolygon object. You could still make another segment-per-object representation for selection use only if you don't mind the duplication (see the SHARE_DATA property for some help in this area).

> XObjView has a nice function (Select) that shows the type of a Model if
> klicked with the mouse. Perhaps this could be enhanced to return the
> koordinates (or better, their indexes) of a polygon of a polygon-model
> but unfortunately I couldn't find the code in xobjview.pro :(
>
> At last I want to view the model, select a polygon by mouse which won't

> be shown till it's selected again.
>
> ideas ?
>
> Thanks in advance,
> Tom
