Subject: Q:how to use MESH_OBJ
Posted by Rick Towler on Tue, 28 Mar 2000 08:00:00 GMT
View Forum Message <> Reply to Message

Hi all,

First a big thanks to everyone who responded to my post a few weeks back on z-buffer ordering. Paraphrasing Emeril, I have been forced to kick my IDL programming up a notch and your comments have helped.

Today I am wrestling with the MESH_OBJ routine. I am trying to generate a surface that represents a 3d fish. The closest example I can come up with is the teapot in the IDL demos except that the code to actually generate the teapot mesh isn't included (or I haven't found it. The program that displays the teapot loads a file that contains the mesh and polygon data.)

I have an array of x,y,z points that make up the fish body. I have had success manipulating the data a little and using the "extrusion" surface type to generate a collection of thin oval cylinders that when viewed from the side look o.k. The problem with this approach is that these individual extrusions aren't connected to their neighbors so when the polygon normals start to point away from the eye you can start to see in between them. And, when viewed head or tail on, the object virtually disappears (the extrusions have no thickness). The problem here seems to be the polygon list. If I omit the polygon list from the call to IDLgrPolygon the resultant object will be interconnected and I will not have the problems describe above. But, I do introduce another problem in that it takes far to long to generate the IDLgrPolygon object without the polygon list information.

Since the problem seems to be with the polygon list (the mesh points are fine, they just aren't connected), I tried using a polygon list generated from a separate call to MESH_OBJ where I chose the triangulated surface. When I pass the mesh data from the MESH_OBJ "extrusion" call and the polygon data from the MESH_OBJ "triangulated" to my IDLgrPolygon call I get a ton of errors (invalid connectivity list detected(invalid vertex reference)) and finally an object that is interconnected. The only problem here are the errors, and what seem like overly complex meshes (hard to explain).

I thought that the "triangulated" surface type would be the ticket but taking this approach yields unexpected results. I must be missing something. Even if I define 6 vertices for a simple 3d diamond and pass that to MESH_OBJ,1 what I get when I pass that data to IDLgrPologon is an object that sort of looks like you took the diamond and cut along the edges and laid it out. This isn't exactly what it looks like, but it certainly isn't a closed object.

What very basic thing am I missing here? Is this the wrong tool? How did the RSI people generate the surface mesh and ploy list for the teapot? Is there a way to fix the polygon list generated by the MESH_OBJ "extrusion" routine so all the little guys are connected?

And lastly, am I just screwed and do i have to generate the polygon list manually?

Thanks!

-Rick Towler
