Subject: Re: Tessellate Question--fewest convex polys
Posted by karl_schultz on Fri, 08 Feb 2002 23:11:23 GMT
View Forum Message <> Reply to Message

David Fanning <david@dfanning.com> wrote in message
news:<MPG.16cdc43310dbf739897fb@news.frii.com>...
> Noname (parrhasius@altavista.com) writes:
>
>> If MESH_DECIMATE really
>> does what I need, I apologize for this thread and don't bother to answer,
>> unless there's some trick to using it... I thought I checked it out.
>
> I've never had the ...er, opportunity to use this
> yet. I'm afraid someone else will have to address
> this question.
>
> Cheers,
>
> David

MESH_DECIMATE is used to simplify polygonal meshes by collapsing mesh vertex pairs, selecting the pairs carefully so that the simplified mesh represents the original mesh as closely as possible.

It is typically used for things like 3D terrain maps and polyognal models in order to produce a mesh that has fewer triangles, yet still retains the original 3D shape as much as possible. Try the little demo program called 'decimate' in the IDL 5.4 and 5.5 releases.

Using MESH_DECIMATE on a 2D (flat) mesh is a little out of the ordinary, since the criteria used to select the vertex pair to collapse would select any vertex pair in the mesh. That is, it does not matter which pair it selects; you will always have a flat mesh regardless of which and how many pairs are collapsed. This would make the pair collapse look a little random, or related to the order of the triangles in the mesh. But it would still reduce the number of triangles in the mesh, depending on the percentage setting.

It is a little hard to say if this is the right tool for the data described in the original post. MESH_DECIMATE starts by turning the mesh into triangles, so it actually increases the number of polygons in the mesh as it gets started. This is required, because the vertex collapse criteria evaluation depends on the mesh only containing triangles. Then, it will collapse vertices and the number of tris will decrease and the size of the triangles will increase. You'll end up with a bunch of largeish triangles that define your mesh.

If you (original poster) really needs to have large n-gons instead of

large triangles, then this won't work. I'd suggest googling around on the net for this sort of thing. In a way, this is sort of a reverse constrained tessellation - combining triangles into convex polygons. I don't think that there is anything in IDL that will do this directly. The main purpose of the tessellator object is to allow you to convert possibly concave polygons into a set of triangles that can be drawn using IDLgrPolygon.

Karl
RSI
