public class TMeshTriangleSeparator

Provides the ability to section out meshes based on user defined subset functions. Can calculate chunks for both convex and concave meshes if proper mesh data is used. By using a callback method all calculations can be offloaded onto a worker thread, further multithreading is used in concave mesh calculations. Double edges (vertices) can also be detected for more appropriate results.

public TMeshTriangleSeparator(ITriangleSeparator tc)

Initializes arrays to be used for calculations later on.

tc

ITriangleSeparator interface implementation to be used in further calculations

public void setTriangleCutter(ITriangleSeparator tc)

Specify the ITriangleSeparator interface implementation to be used in further calculations.

tc

ITriangleSeparator interface implementation to be used in further calculations

public bool divideMesh(Mesh m, out List<Chunk>[] chunks, out Mesh mesh)

Divides the mesh into chunks based on the ITriangleSeparator implementation being used. Both chunks and edges are divided into the 0th and 1th subset.

m

Input mesh.

chunks

Lists to be filled with chunks. [0] list contains chunks that are in the base subset, [1] list chunks that are not.

mesh

Mesh containing data indexed by the chunks and edges.

Return Value

True if division was successful, false if it wasn't.

public void divideMesh(Mesh m, SeparatorFinishedHandeler ev)

Divides the mesh into chunks based on the ITriangleSeparator implementation being used. Both chunks and edges are divided into the 0th and 1th subset.

m

Input mesh.

ev

Callback to be used upon completion.