Geometrie Algorithms Fall, 2016

TP 3

Problem. In this TP you have to write functions with the goal of writing the algorithm to compute simpifications of a mesh.

1 Functions

1. `void myMesh::contractEdge(myHalfedge *e, myPoint3D *p)`.

   Given a `myHalfedge *e`, the function collapses `e` into a vertex. So the two vertices of `e` are merged into a single vertex placed at point `p`. There might be duplicate edges going to this new vertex, and you should remove any such duplicates.

2. `void myMesh::simplifyMesh(int numvert)`.

   Calling this function performs mesh simplification on the current mesh by reducing its number of vertices by `numvert`. So if the mesh had 100 vertices before, afterwards it should have (100−`numvert`) vertices. You can either use the **Quadric Simplification** Algorithm we covered in class, or implement the following algorithm: do "edge contraction" on the edge with the smallest length. The new vertex from this edge contraction should be the midpoint of this edge. Now repeat the above process on the next smallest edge and so on. Do this for `numvert` steps.

2 Other code to write

Write other code in the `main.cpp` file to get the following popup menu options working:

1. `Mesh Operations -> Simplification`