

Assignment 1

CMSC 740, Spring 2007

Due: 11:00am Tuesday, February 20, 2007

This assignment involves displaying and painting a 3D model. The project web-page is at www.cs.umd.edu/class/spring2007/cmcs740/assq1/. Compile and run the startup code from the above page after linking with OpenGL and GLUT. The startup code that we are giving you does the following:

- It opens up a GLUT window of size 800 x 800.
- It displays a wireframe cube.
- It provides a user interface library called `iui` for rotating and zooming the cube by the mouse operation. Please refer to `iui.pdf` file for usage.

(a) Drawing a teapot (15 points): Draw a teapot model instead of the cube with OpenGL's smooth shading (`GL_SMOOTH`). The startup code has some components to read the teapot model (`teapot.ply`) and render it. All you have to do is to compute the vertex normals. Look for `needNormal()` function in `Smesh` class and write functions for computing the vertex normals. The startup code reads `teapot.ply` and allocates and fills up `verts` and `tris` members in `Smesh` class. The startup code is a guideline; feel free to change it as you wish. You will first have to link the vertices to their adjacent triangles for computing vertex normals. Here is the list of functions that you might want to fill up in `Smesh` class.

```
// 1. allocate triangles neighboring each vertex
computeVertNeighbors();
// 2. compute the areas of triangles
computeTriangleAreas();
// 3. compute triangle-normals
computeTriNormals();
// 4. compute vertex-normals: Each normal of a vertex is the weighted sum of the
// normals of the neighboring triangles. Weights are simply the areas.
computeVertNormals();
```

(b) Selecting and Deselecting triangles (15 points): Use the right button of the mouse to select the triangles below the current mouse locations. If a user clicks the right button with a `CONTROL` or `SHIFT` key pressed, you should deselect the triangles below the current mouse locations. To make it easy to select or deselect small triangles, fix the brush size to be 10 pixels. In other words, you should select all the triangles within 5 pixels radius from the current position of the mouse. Selected vertices will then be drawn by `drawSelectedTriangles()` function in `Smesh` class. All you have to do is to set each of `pSelectedTris` array member in `Smesh` class accordingly. Here are two functions that you might want to write in `Smesh` class.

```
// 1. draw each triangle in the color which indicates its id
drawTrianglesWithIDs()
// 2. read back all the triangles centered at (iX, iY) with the radius 'BRUSH_SIZE'
// from the back buffer
readPixelsFromBackBufferAndAssignToSelectedTriangles(int iX, int iY, bool bSelected)
```