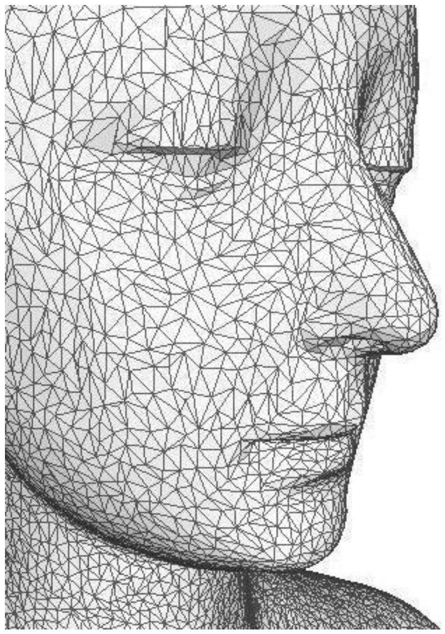
CS 116A – Introduction to Computer Graphics – Project 1 – Mesh Viewer – Part 1/2

September 4, 2024

Introduction



In this assignment you will learn how to programmatically create your own indexed Mesh class, reading data from a Wavefront .obj file and displaying the mesh in an OpenFrameworks interactive 3D Window.

Part I – Create your own Mesh Class and Interactive Viewer (10 points)

- Create your own C++ class called *Mesh.* For implementation of this class used the indexed triangle vertex method we discussed in class. In particular there are two key storage elements in this class: (1) the list of vertices in the mesh and (2) a list of indices to the triangle vertices. Rather than used a fixed length array for these elements, use the **C++ vector** template class which is a container class which allows you to create a dynamic arrow of any length. (ref: http://www.cplusplus.com/reference/vector/vector/). Your mesh should support any number of vertices and faces (triangles) for your Mesh, thus the requirement for using a dynamic array.
- Create a method in your class called *draw()* to draw the mesh in wireframe. This method should iterate through all the triangles you have created and draw each triangle. It is acceptable to draw the same edge multiple times for this assignment. Use an EasyCam as described in the class exercises so that you will be able to move the camera around and view your mesh interactively.
- Create a simple piece of test geometry using your new Mesh class. You can create the test geometry by hard-coding the some vertex data/triangle indices in your source file. (you will generalize this approach using .obj files in Part II below)

Part II - Create the Wavefront .obj file reader (10 points)

1. Study the Wavefront .obj file format specification on the wiki: https://en.wikipedia.org/wiki/Wavefront_.obj_file
2. For this assignment, we will only be concerned about the "v" (vertices) and "f" (faces) in the file. You can ignore normal and texture coordinates for now.
3. Create a method in your app class to read the .obj file, parse each line in the file while loading the vertex and face (triangle) data into your class. Use the ofFile class for all file handling, (open, close and reading).
4. Search for an .obj model in the internet that you like and test your viewer using this file or alternatively, create your own model in Maya (or any 3D modeler) and export to .obj.
5. After your file loads, print some diagnostic information to the console which includes:

    a. The total number of vertices
    b. The total number of faces.

c. The size of your Mesh structure (in kB) (include only size of vertex and triangle data

Part III – Calculate and Display Face Normals

Design a simple panel interface using ofxGUI that includes a toggle button to select whether or not **face** normals are displayed. Use **Barycentric Coordinates** to find a reasonable approximation for a point in the center of each triangle and draw the normal from the center of the triangle along the normal direction. Provide a GUI slider to adjust the normal length when drawn.

Important - What to Submit

1. Submit only ofApp.h and ofApp.cpp files for your project (and any other sources files required to build your project). Do not submit the entire OF project. Zip it into a .zip file using the following naming convention:

   Project1-Meshes-<your name>-<date>.zip.

   As an example:

   Project1-MeshViewer-KevinSmith-09042019.zip

   No other compressed format will be accepted (zip is available natively on both windows and Mac).

2. Submit a video demonstrating part 1,2 and 3 (in the same video). Use a screen capture program like Camtasia (or equivalent). Mobile phone videos or "screeners" will not be accepted.

Due Date

Due date will be posted in Canvas system. You will need to manage your time working on this project to complete it by the deadline.

Grading Criteria

See the rubric.