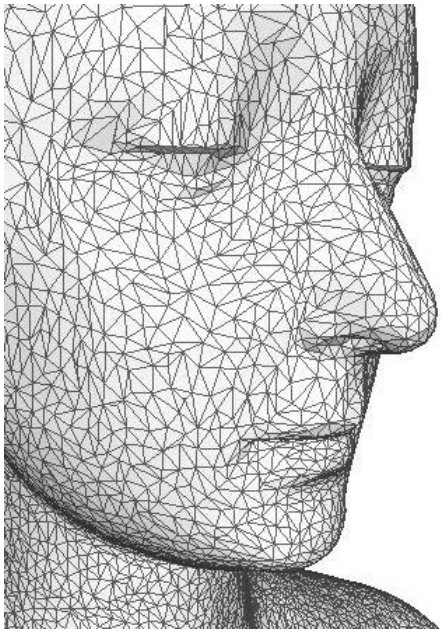
CS 116A – Introduction to Computer Graphics – Project 1 – Mesh Viewer – Part 1/2

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 (15 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 load the file into your Mesh class you created in Part 1 (and create your vertex list and triangle index list in the class as you load it in). You will need to use C++ file IO. I suggest looking at the iostream functions. There is a summary here: https://www.cs.fsu.edu/~myers/c++/notes/fileio.html. The easiest way to allow the user to select the file to load (without using a file browser) is to use drag-and-drop. I will give an example in class. If you are more comfortable using traditional stdio calls like fopen() and fscanf() etc, you can use those if you choose.
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)

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

Due Date

This is ten-day assignment due at the end of next week. Due date will be posted in Canvas system. You will need to work on this project a little bit each day in order to complete it on time.

Grading Criteria

To receive the highest grade on the assignment, the code most be robust and well documented (commented) with all functionality specified included. Source code files must be signed by author.