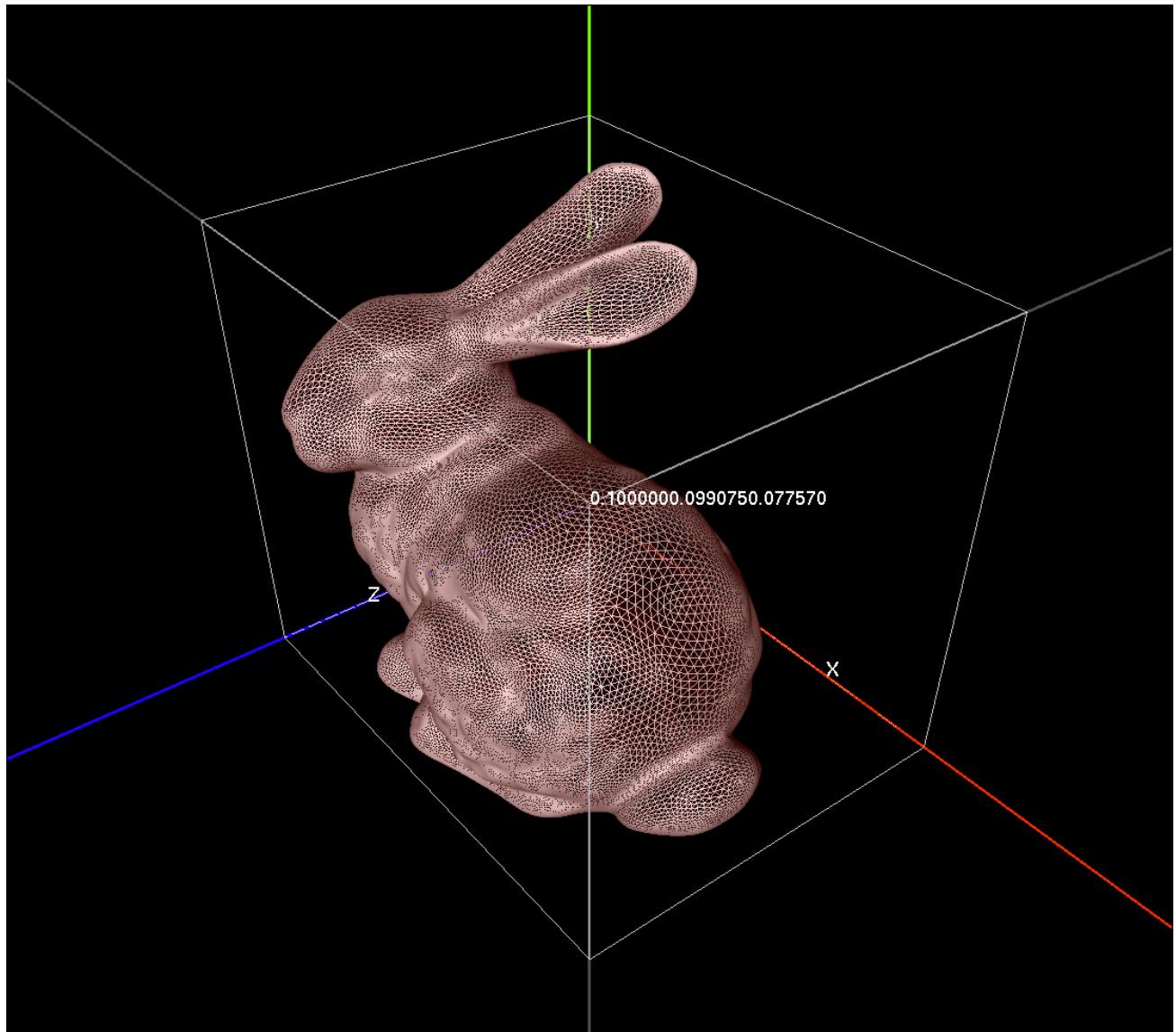


CZ4004 Assignment

Goyal, Shubham
U1122584C

2 November 2014



User Manual

This OpenGL mesh viewer is designed to run on any system capable of running OpenGL and GLUT. Complete interaction with the program requires a keyboard and a mouse with all 3 buttons (left, right and middle.)

Keyboard Inputs

Input Key	Result
W	Render object as WIREFRAME
E	Render object as POINT cloud
S	Render solid object. Press to toggle between FLAT and SMOOTH shading
V	Toggle between PERSPECTIVE and ORTHOGONAL projection
1	Load Bunny model
2	Load Gargoyle model
3	Load Knot model
4	Load Eight model
5	Load Cap model

Mouse Inputs

Note: Mouse operations require holding one of the 3 buttons and moving the mouse

Hold Button	Horizontal Movement	Vertical Movement
Left	Rotate object along Y axis	Rotate object alone X axis
Middle	N/A	Zoom in / out
Right	Translate camera along X axis	Translate camera along Z axis

- This program uses a single light source.
- The floor plane is rendered based on the size of the model.
- Initial camera position is always relative to the size of the model.
- Max coordinates of the bounding box are displayed for the user's information.

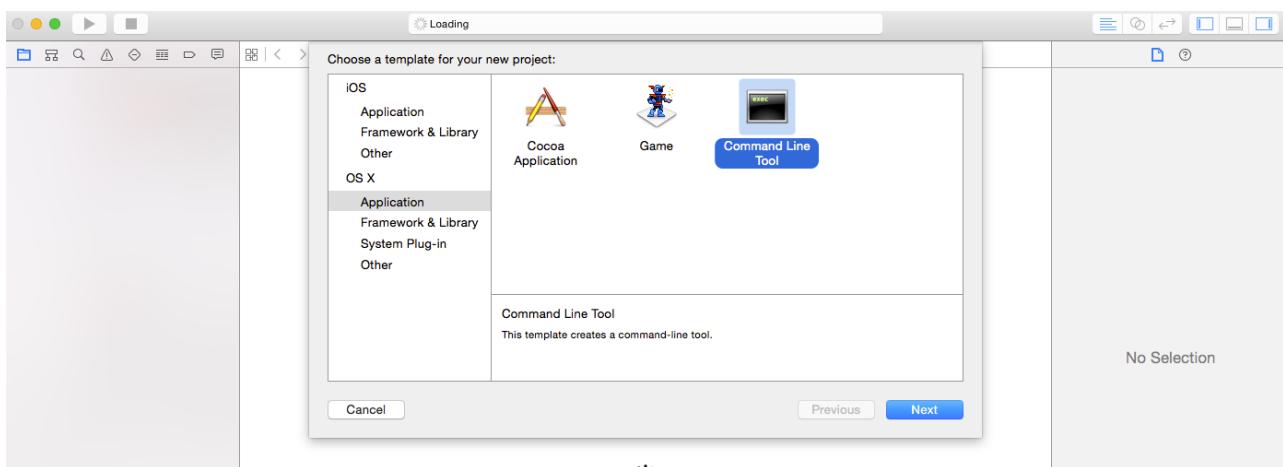
Compiling Instructions

Compilation platform: Mac OSX 10.10 - Yosemite

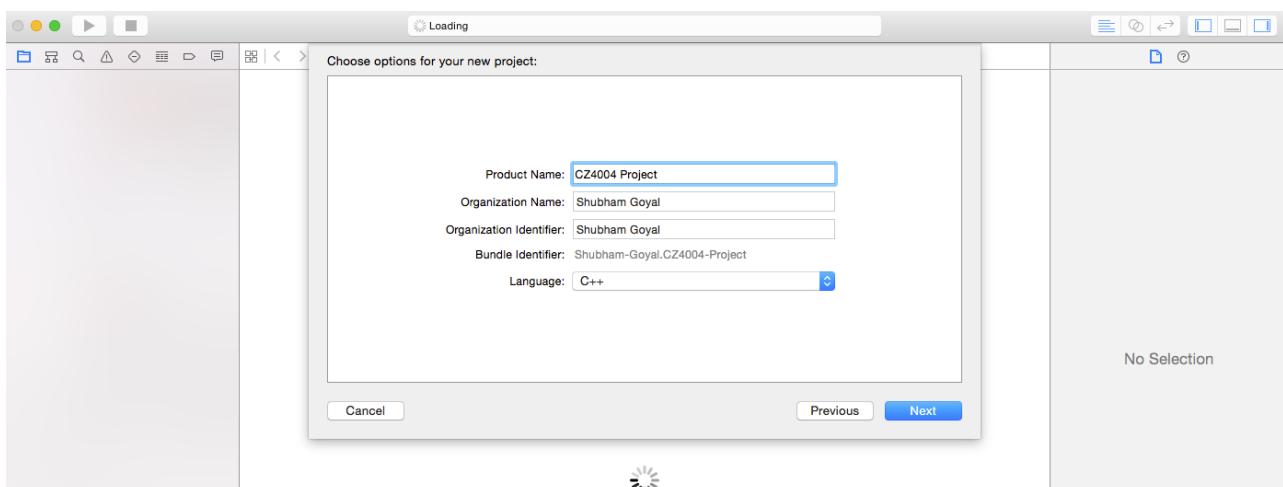
Development Tool: Xcode 6.1

Libraries used: OpenGL , GLUT

1. Open Xcode, which is supplied as a free application in the App Store for all OS X devices
2. Create a new project. Choose the command line tool under OS X application

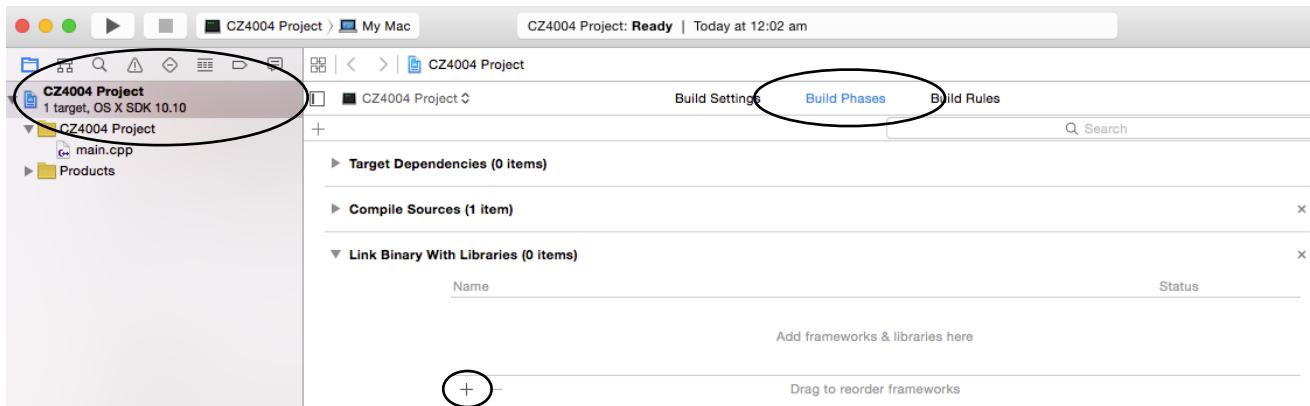


3. Enter a name for the project and select language as C++.

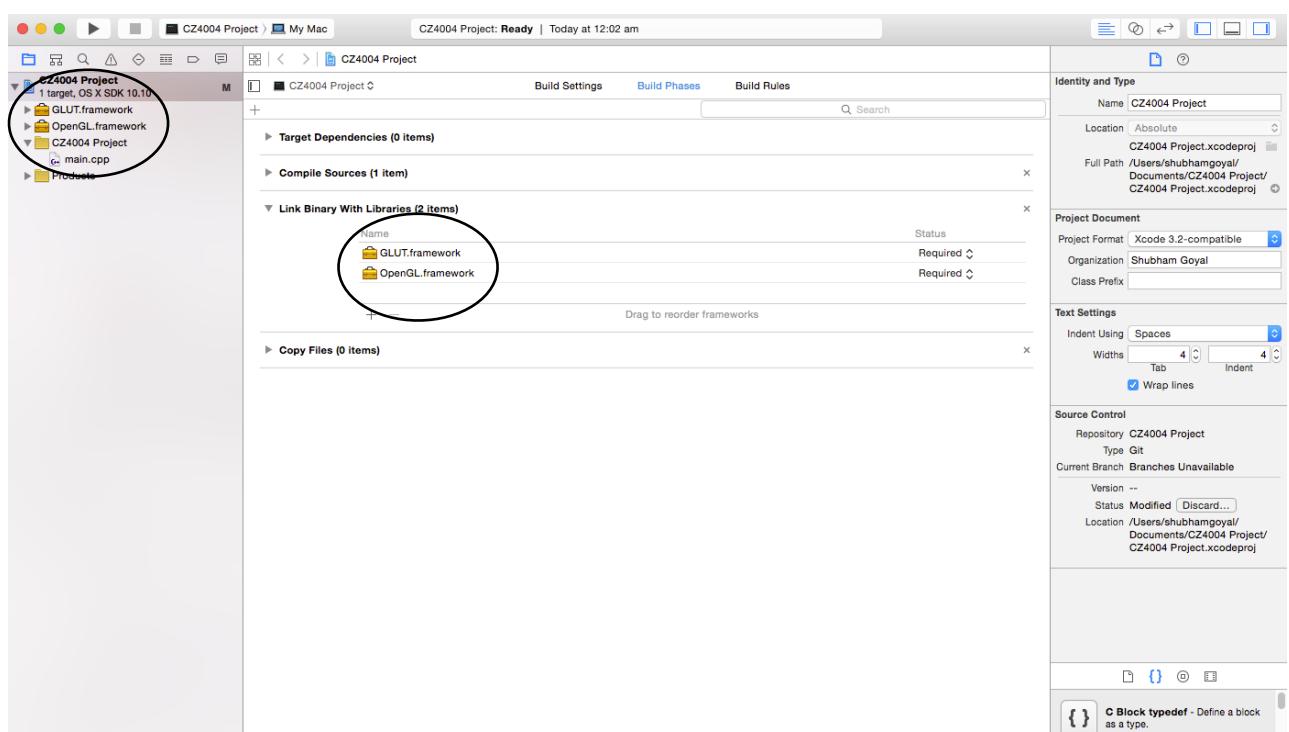
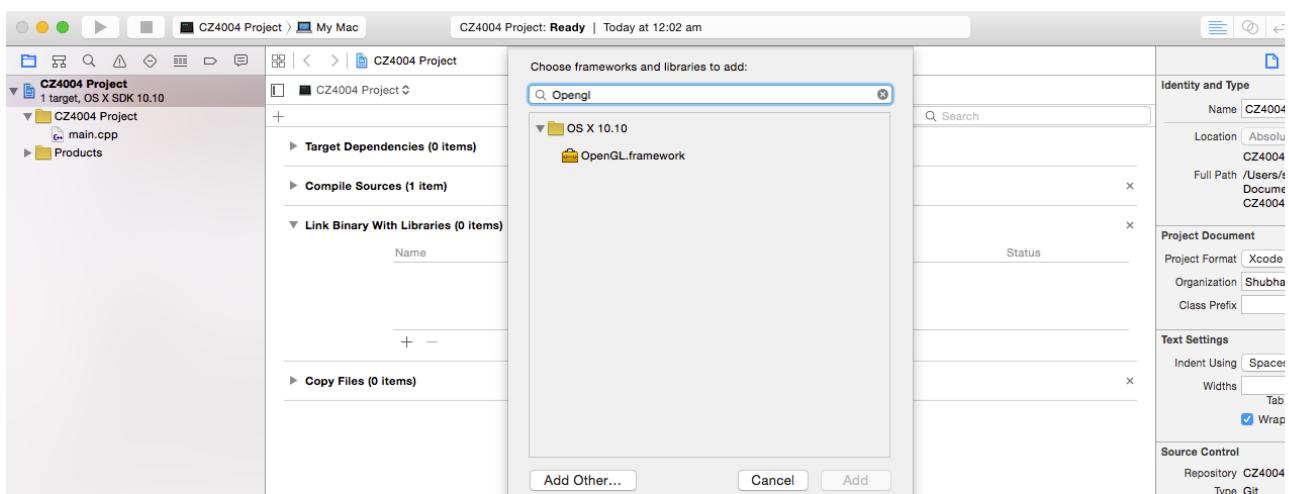


4. Choose a location to save your project

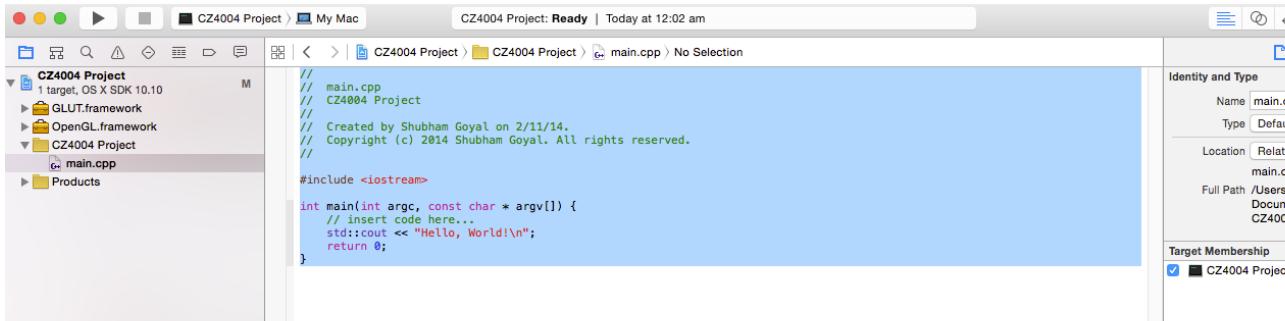
5. Under the main project settings, choose “Build Phases” and click the “+” button.



6. Search for “OpenGL” and “GLUT”. Add them to the project.



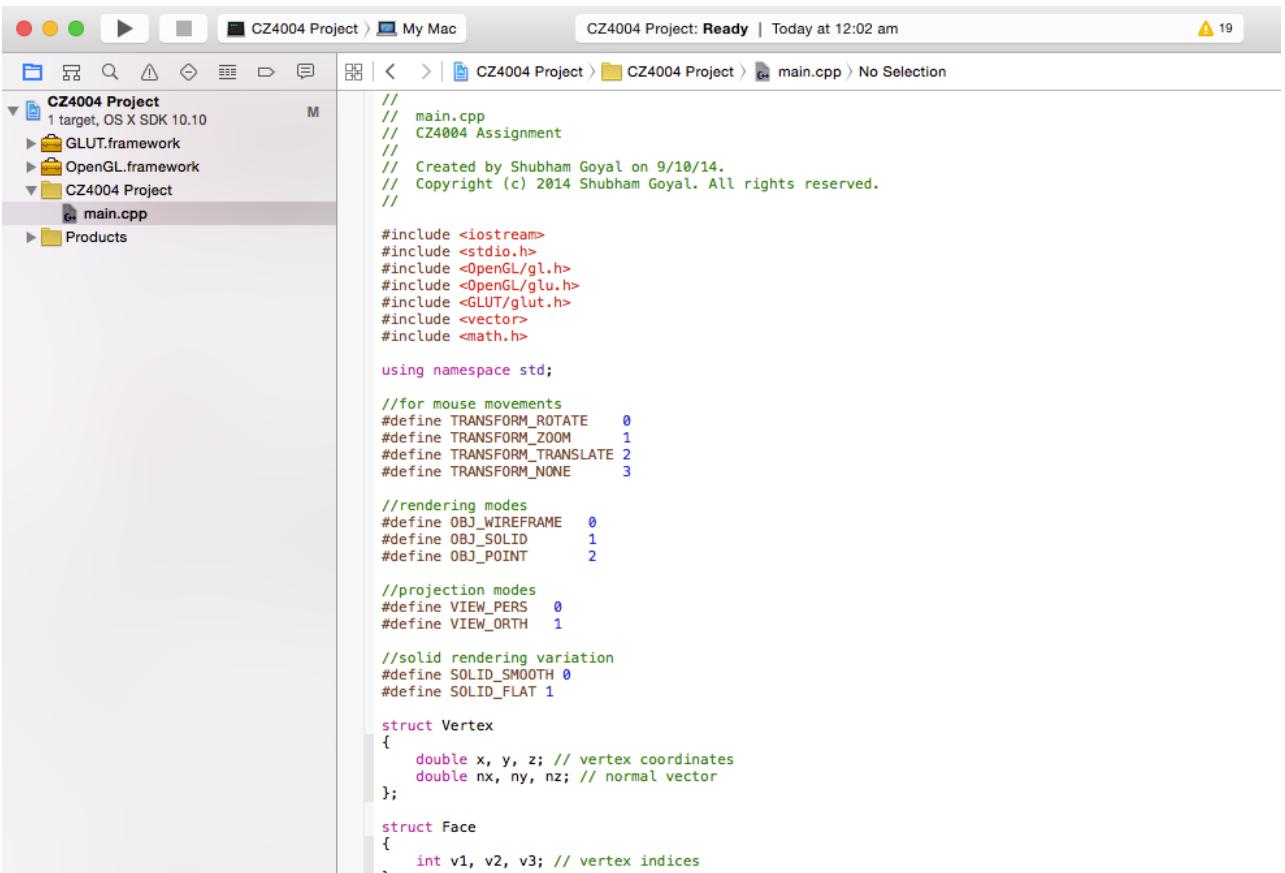
7. Replace the default main.cpp file with the supplied source code file.



```
// main.cpp
// CZ4004 Project
//
// Created by Shubham Goyal on 2/11/14.
// Copyright (c) 2014 Shubham Goyal. All rights reserved.
//

#include <iostream>

int main(int argc, const char * argv[]) {
    // Insert code here...
    std::cout << "Hello, World!\n";
    return 0;
}
```



```
//
// main.cpp
// CZ4004 Assignment
//
// Created by Shubham Goyal on 9/10/14.
// Copyright (c) 2014 Shubham Goyal. All rights reserved.
//

#include <iostream>
#include <stdio.h>
#include <OpenGL/gl.h>
#include <OpenGL/glu.h>
#include <GLUT/glut.h>
#include <vector>
#include <math.h>

using namespace std;

//for mouse movements
#define TRANSFORM_ROTATE 0
#define TRANSFORM_ZOOM 1
#define TRANSFORM_TRANSLATE 2
#define TRANSFORM_NONE 3

//rendering modes
#define OBJ_WIREFRAME 0
#define OBJ_SOLID 1
#define OBJ_POINT 2

//projection modes
#define VIEW_PERS 0
#define VIEW_ORTH 1

//solid rendering variation
#define SOLID_SMOOTH 0
#define SOLID_FLAT 1

struct Vertex
{
    double x, y, z; // vertex coordinates
    double nx, ny, nz; // normal vector
};

struct Face
{
    int v1, v2, v3; // vertex indices
};
```

8. Make sure to update the string that points to the folder path where the 3D models are stored on your computer.

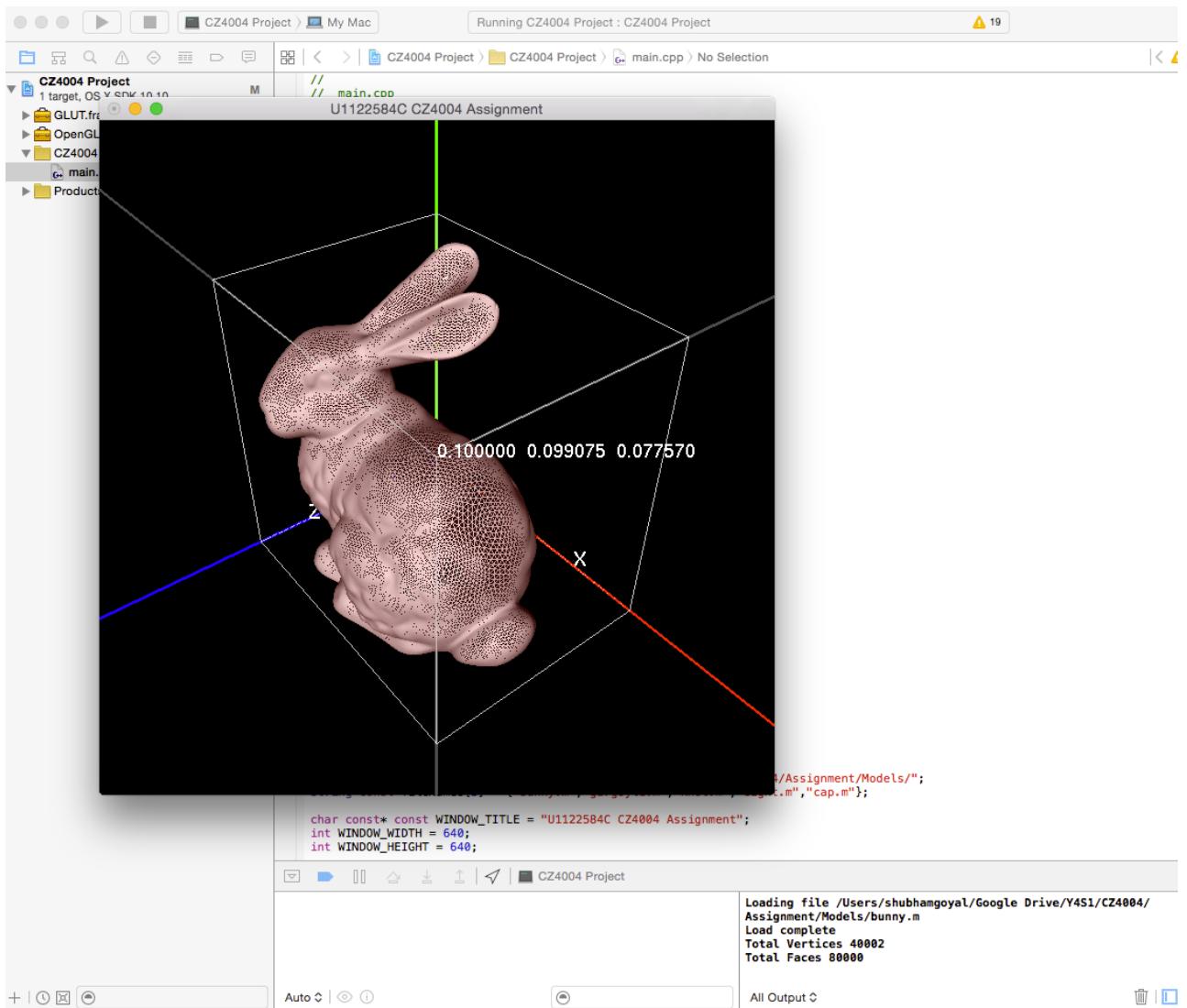
```
vector<Face> faces; // array of the triangles

string const folder = "/Users/shubhamgoyal/Google Drive/Y4S1/CZ4004/Assignment/Models/";
string const filenames[] = {"bunny.m", "gargoyle.m", "knot.m", "eighth.m", "cap.m"};
```

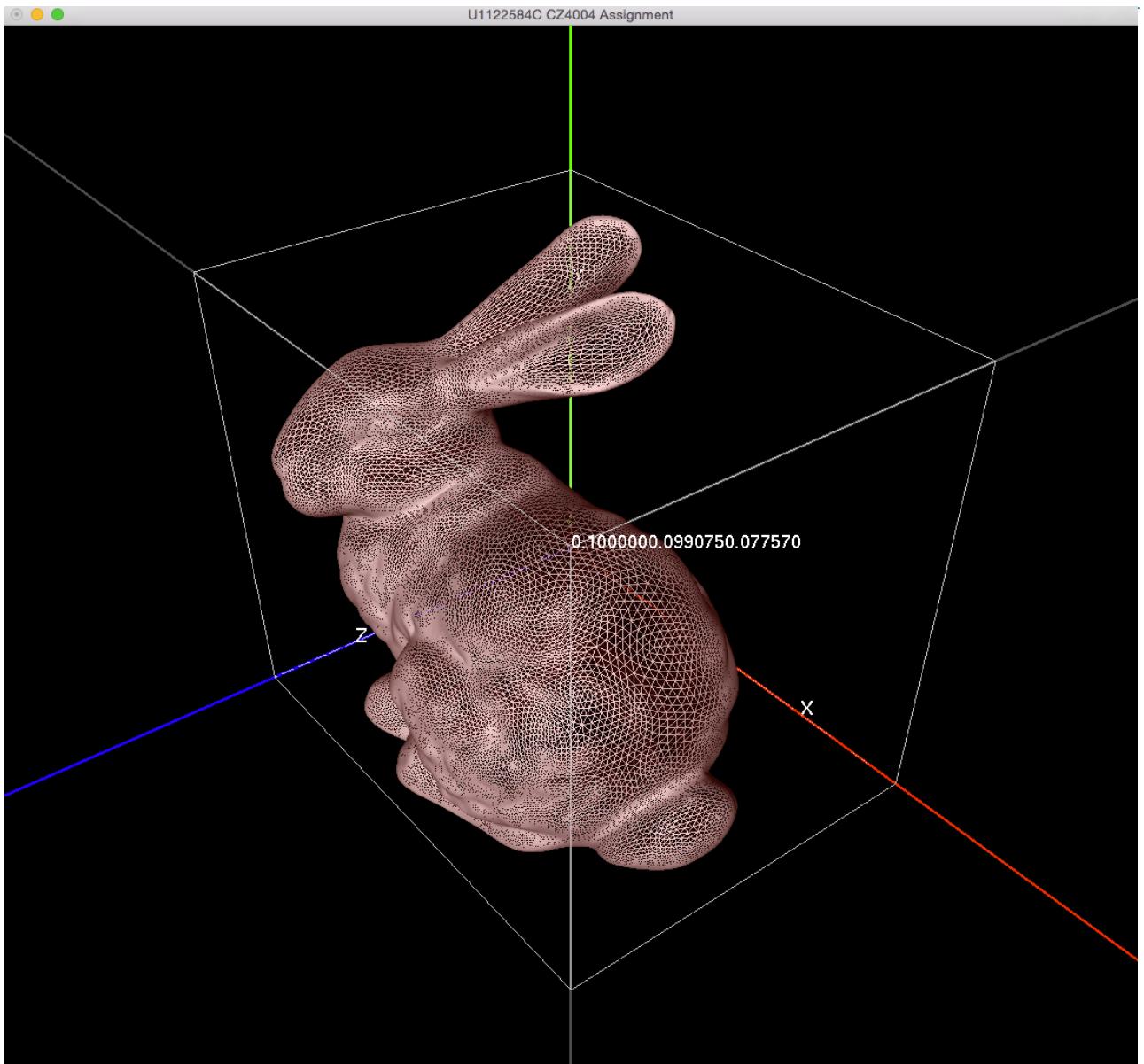
```
char const* const WINDOW_TITLE = "U1122584C CZ4004 Assignment";
int WINDOW_WIDTH = 640;
```

9. Running the project should correctly open the window and the models.

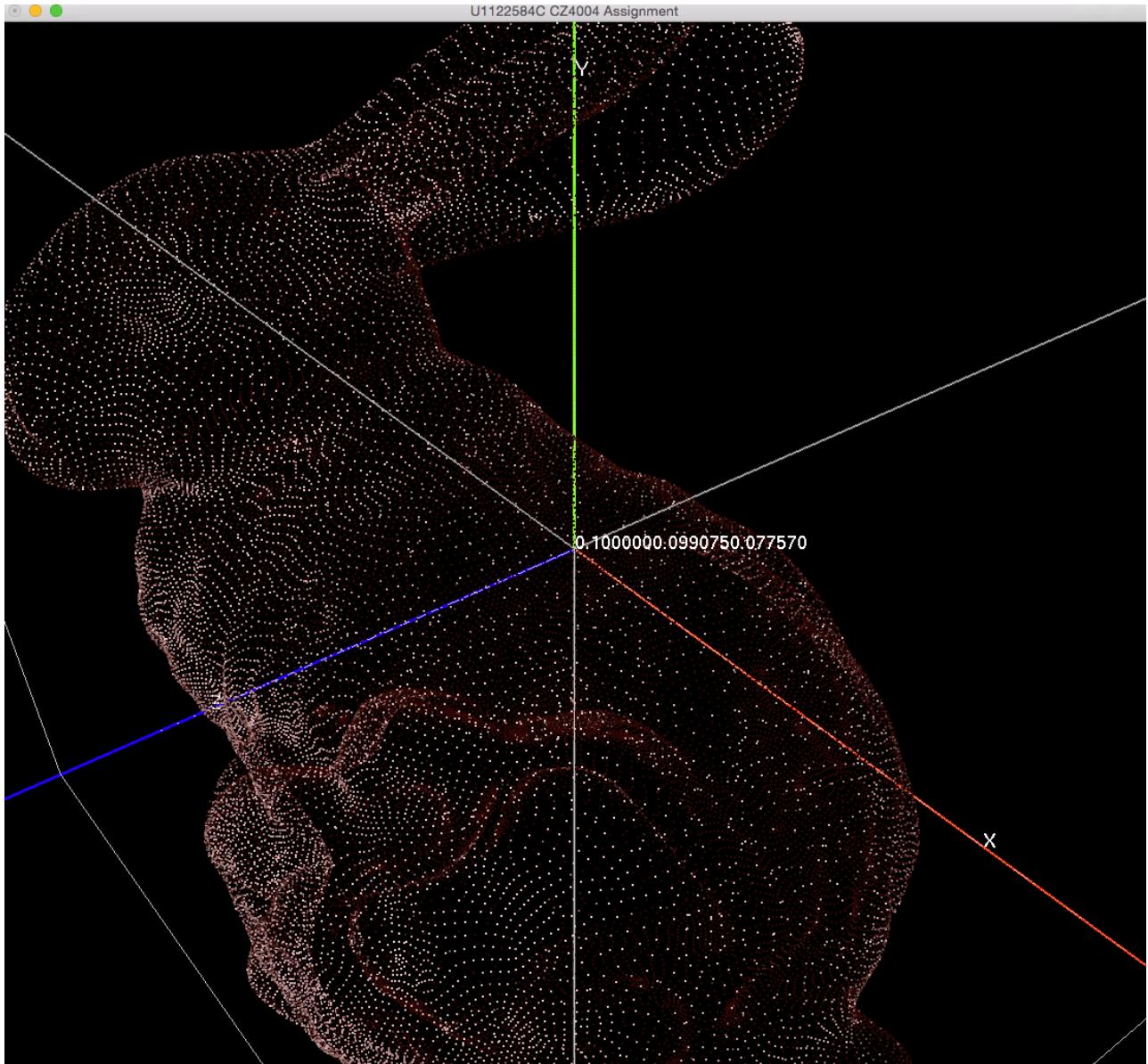
Sample Screenshots



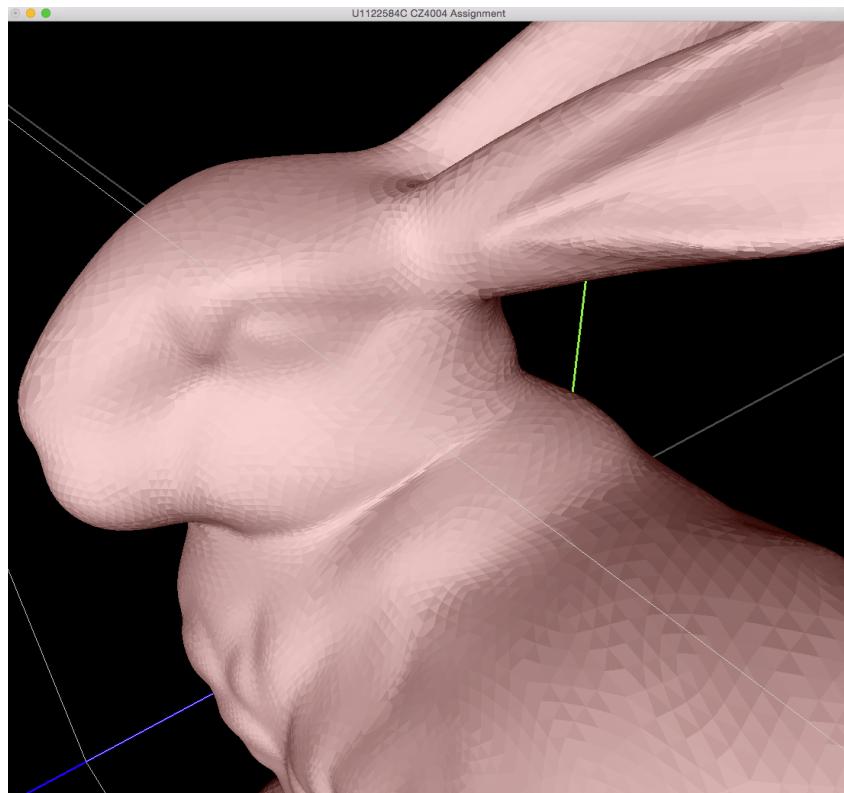
OpenGL output in the window and the console output at the bottom



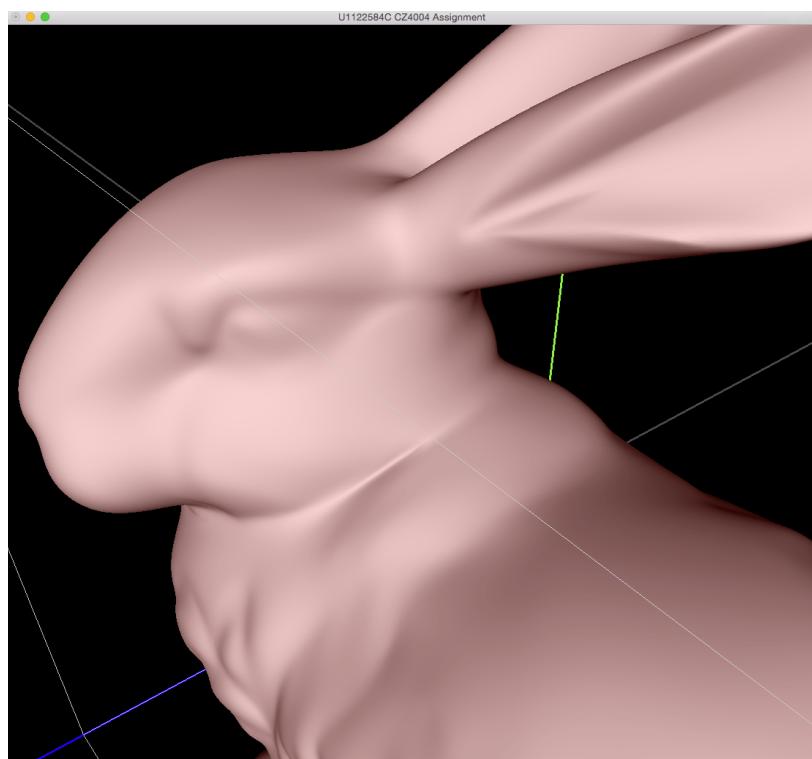
Wireframe rendering



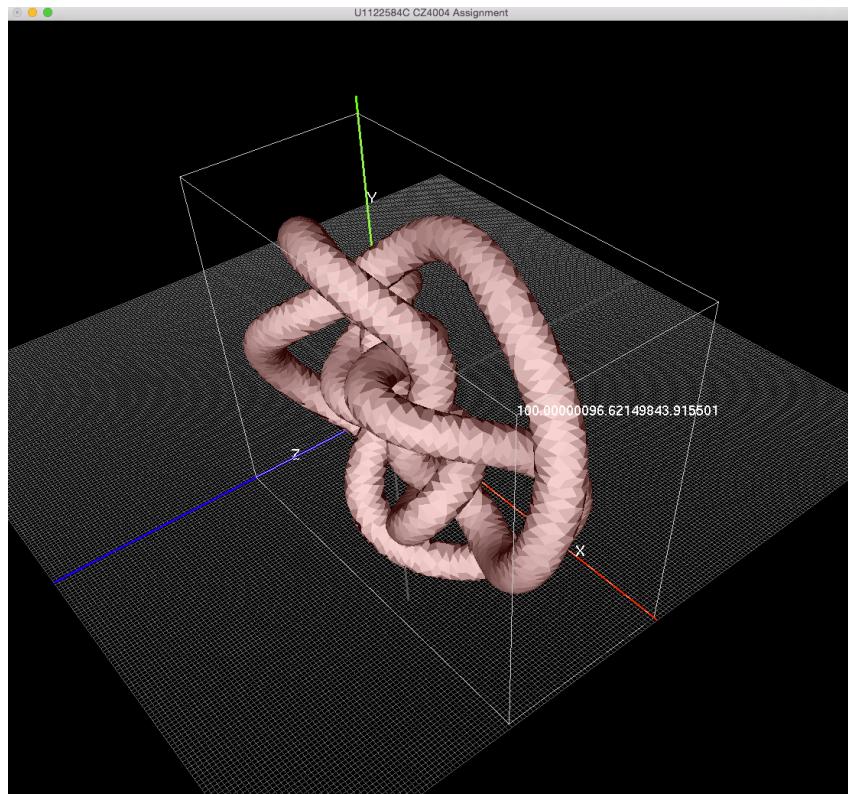
Point cloud rendering



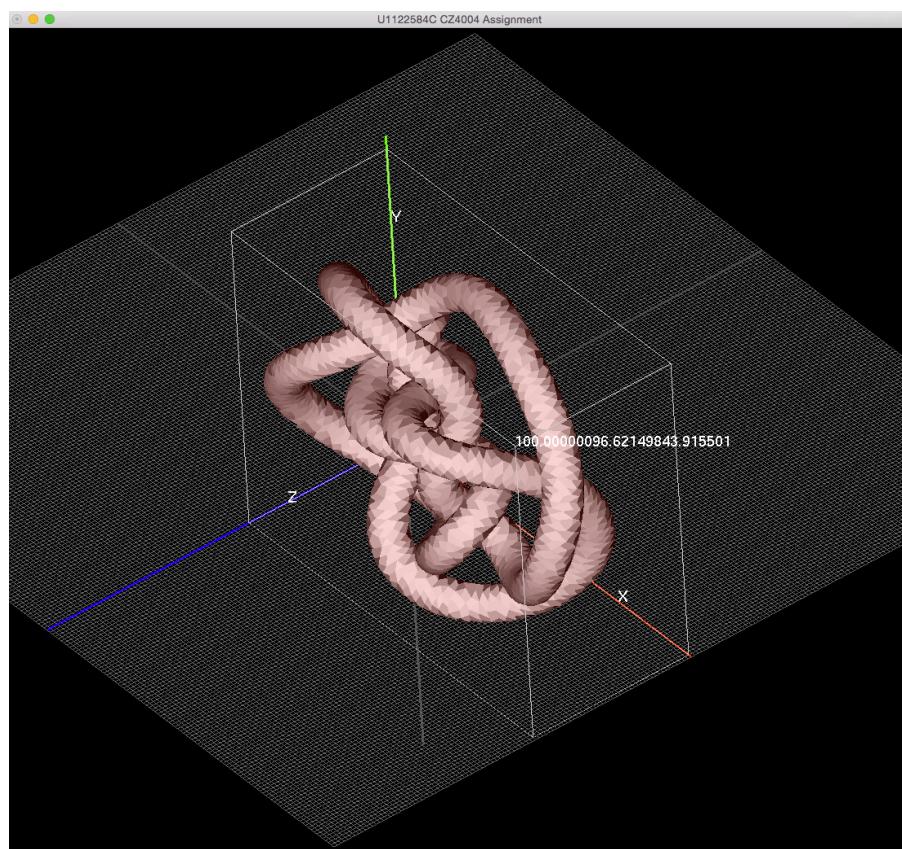
Flat shading



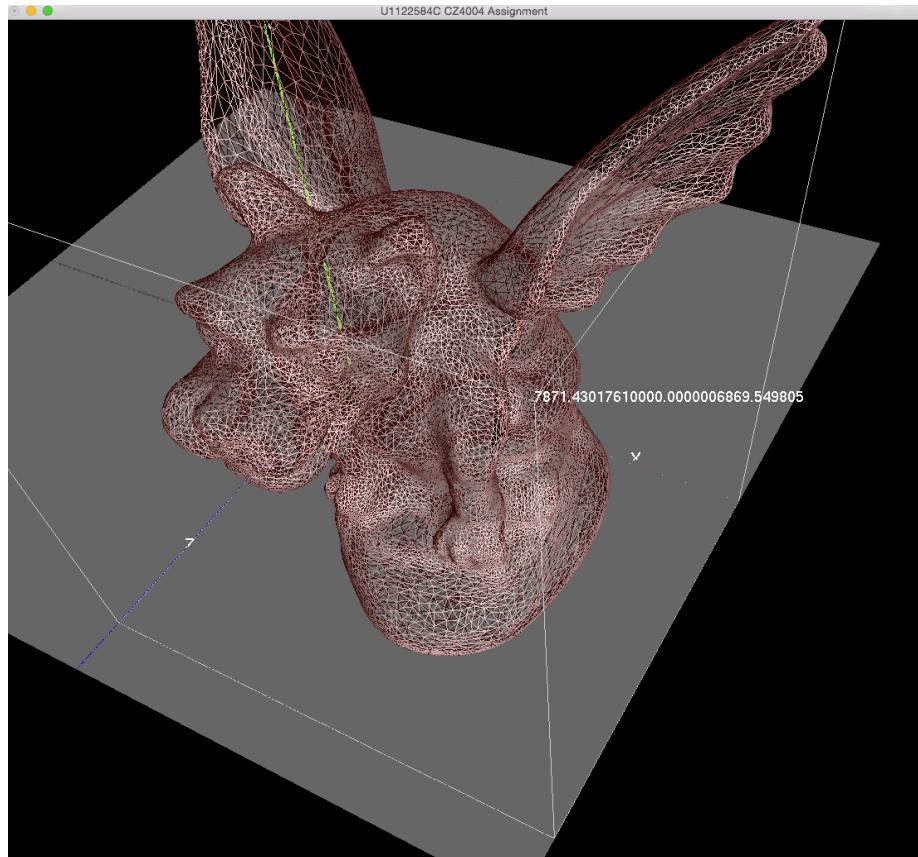
Smooth shading



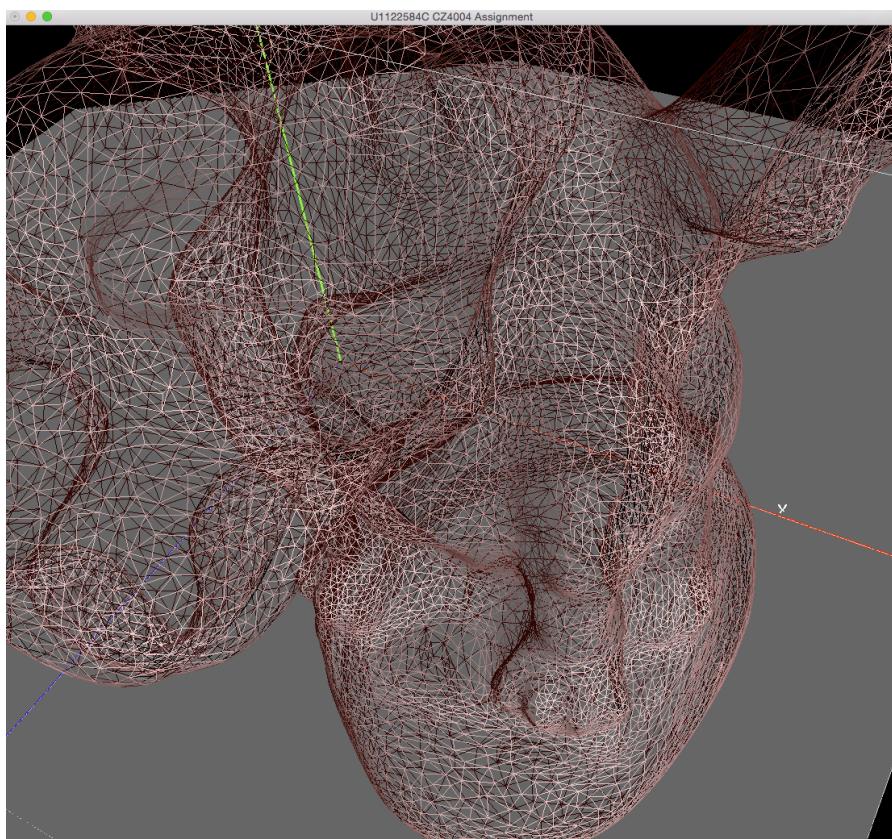
Perspective projection

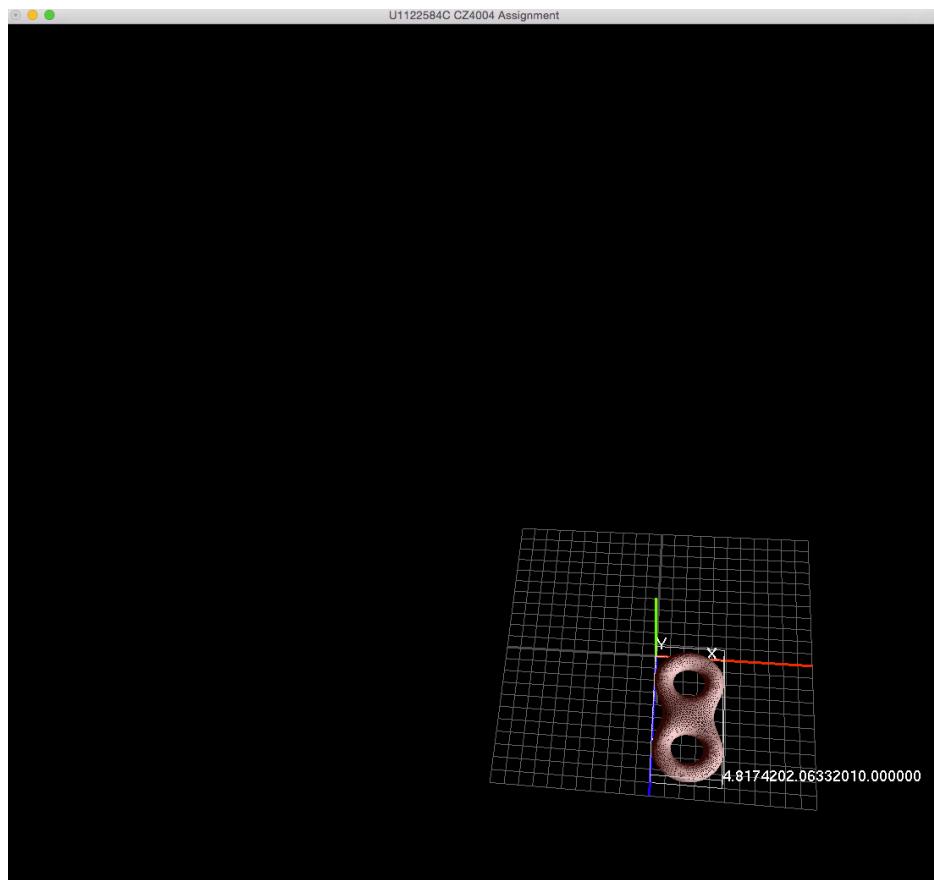


Orthogonal projection

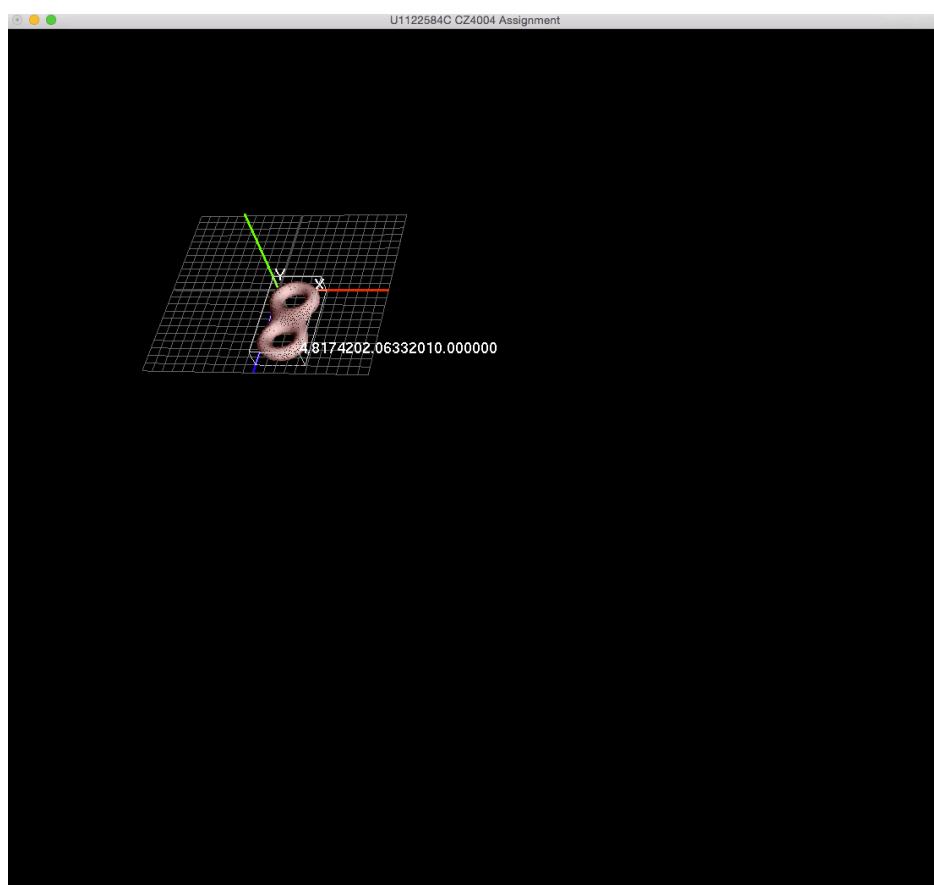


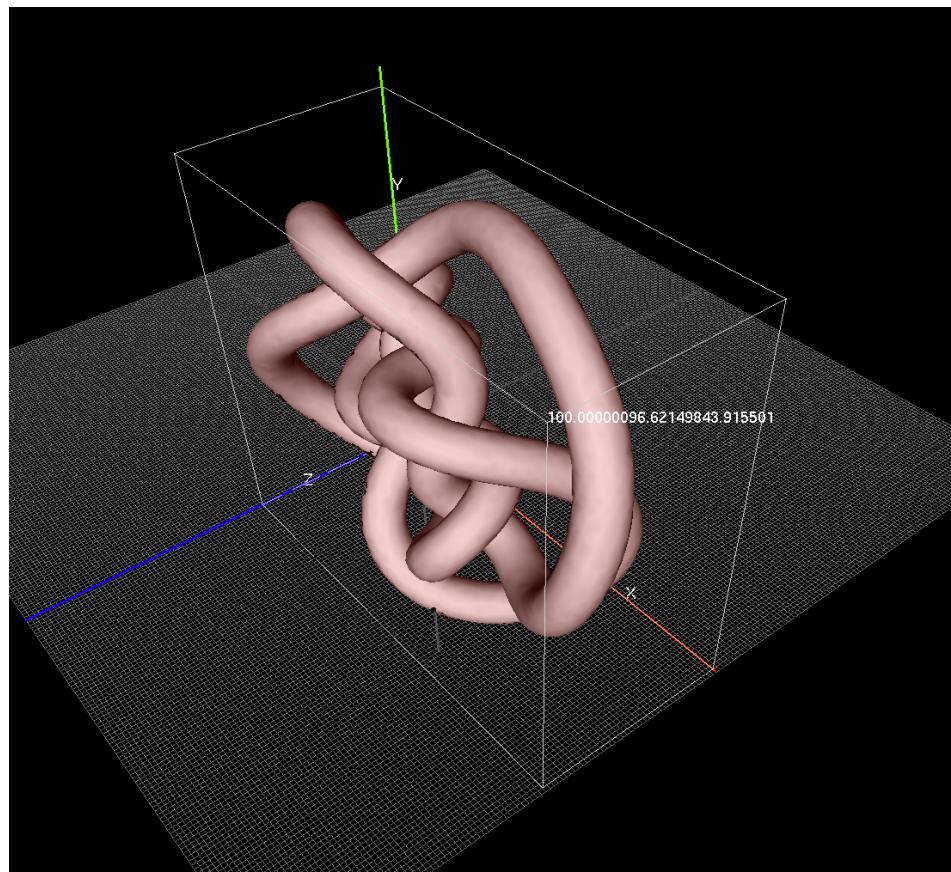
Zoom using middle mouse





Camera translation





Object rotation

