

# Assignment 1: OpenGL Triangle Mesh Viewer

20131218

Park KeeHun

## 1. Introduction

In this assignment, I implement an OpenGL triangular mesh viewer. I take three simple and complex triangular mesh data(bunny.off, fandisk.off, dragon-full.off), which are off-file format, through fstream. Then I render this triangular mesh file to OpenGL viewer. In addition, this viewer can have several functions, such as orthogonal/perspective projection, smooth/wireframe. And it also can have function of rotation, zooming, and panning, which can be used by using a mouse.

## 2. Method

### 2-1. File I/O

```
void read_file(void) { ... }

struct dim_3 {
    GLfloat x, y, z;
};

struct idx_3 {
    GLuint v1, v2, v3;
};
```

By using fstream library, I read off-format file and store it. To store the data efficiently, I use the 2 structures, dim\_3 has 3-coordinates and idx\_3 has 3-vertex index. In read\_file function, after making the per-face normal when taking index data, I make per-vertex normal vector array. And I multiple 10 to data of bunny and dragon-full to larger the initial size of them.

### 2-2. Projection / Rendering

```
void key_input(unsigned char key, int x, int y) { ... }

glutKeyboardFunc(key_input);
```

For implementing keyboard shortcut to switch between different projection & rendering mode, I use this key\_input() function with glutKeyboardFunc(). This function deal with 'key'

parameter like that, 'o' & 'p' is changing the projection between orthogonal and perspective, 's' & 'w' is changing the rendering between smooth and wireframe. And I initialize these with orthogonal and smooth mode in main function.

### 2-3. Multiple light sources

I don't make function for light, just include it in renderScene() function. First light source is located in left side with red color, and second light source is located in right side with sky blue color.

### 2-4. Virtual trackball

```
void mouseButton(int button, int state, int x, int y){ ... }  
void mouseMotion(int x, int y){ ... }  
glutMouseFunc(mouseButton);  
glutMotionFunc(mouseMotion);
```

For implementing mouse control for rotation, zooming, and panning, I use above 4 functions.

mouseButton() is called when user clicks, and in three cases(left, right, middle button & down) it can go to mouseMotion() through 3 boolean variables after initializing current x,y coordinates.

mouseMotion() is called when user drags. In rotation with left button, user can rotate through glRotatef() with angle\_x, angle\_y variables, which are regulated in mouseMotion(). In zooming with right button, user can zoom through glScalef() with zoom variables, which depends on x-axis moving of right button drag. In panning with middle button, user can move the object through glTranslatef() but only in orthogonal projection. Because in perspective projection, we must consider the position of camera, so I use glLookat() and distinguish two cases through 2 boolean variables(is\_ortho, is\_persp).

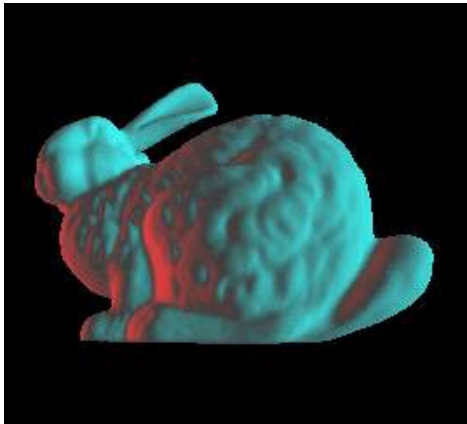
## 3. Result

### 3-1. File I/O

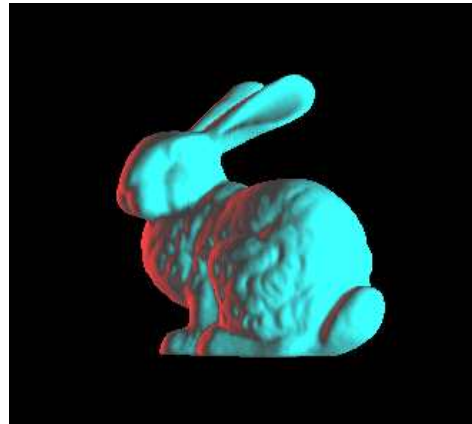
```
b : Open <bunny.off>  
f : Open <fandisk.off>  
d : Open <dragon-full.off>  
::: Enter b/f/d :
```

When building and executing, user can first see above screen. Then user chooses the file he wants to open and enter the character among 'b', 'f', or 'd'. And in this code, it assume that 3 off-files are in same place with Assign\_1.sln, that is in skeleton-code/build.

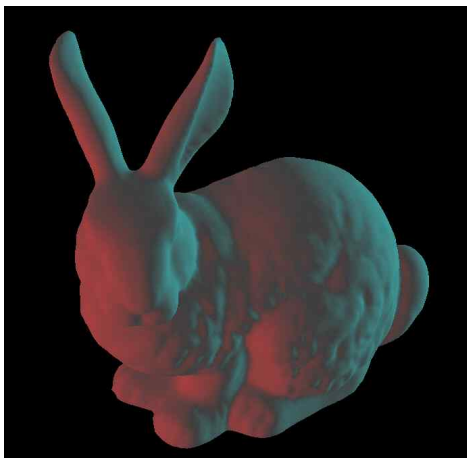
### 3-2. Projection / Rendering



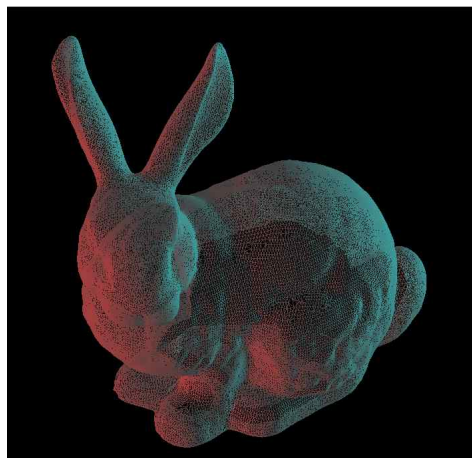
Perspective Projection



Orthogonal Projection

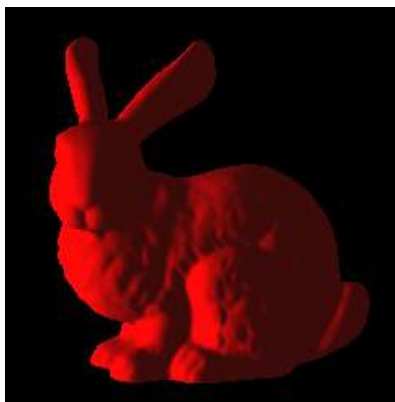


Smooth Rendering

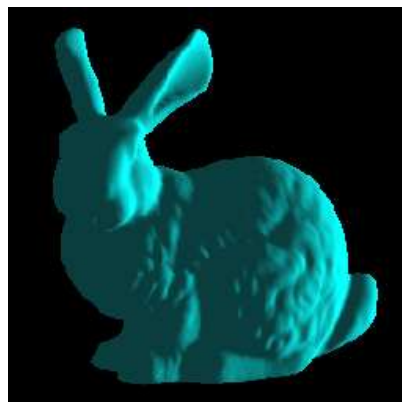


Wireframe Rendering

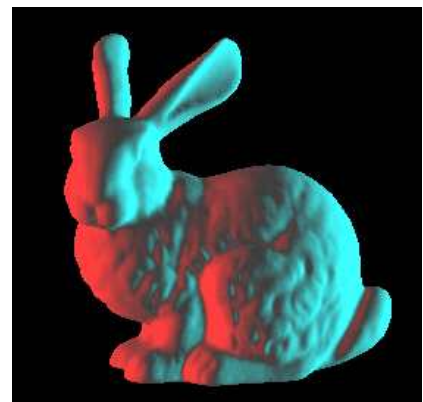
### 3-3. Multiple light sources



Left-red light

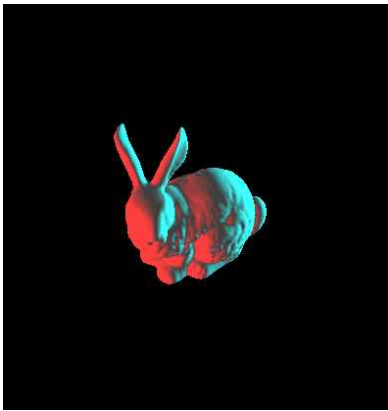


Right-sky blue light

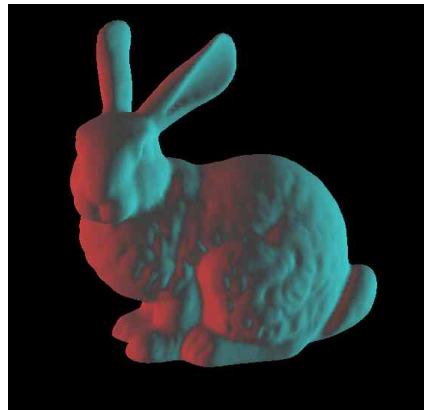


Both light source

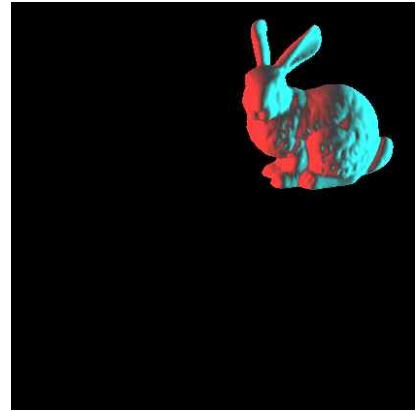
### 3-4. Virtual trackball



Rotation

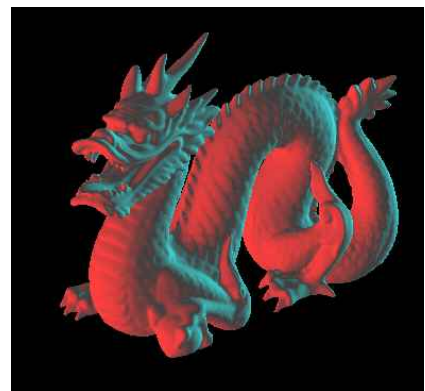
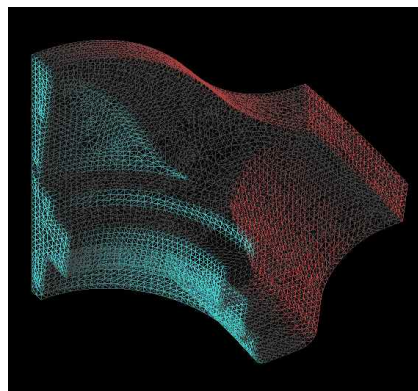
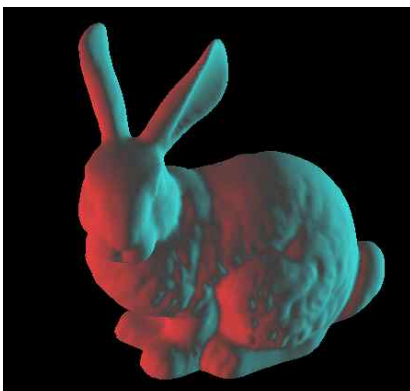


Zooming



Panning(Translation)

## 4. Conclusion



Through this assignment implementation, I can learn and understand about basic of OpenGL. From reading and storing the data of off-files, to rendering them into 3D image and controlling them, I can know about overall process of OpenGL usage. Especially, it is very nice that I become able to deal diverse glut functions and understand about projection concept. Moreover, from keyboard and mouse, taking the input to make virtual trackball functions, such as rotation, zooming, and panning, is also great chance to me.