**Computer Graphics Class Assignment2**

2017029470 김종원

1. **Which requirement you implemented**
2. **Manipulate the camera in the same was as in ClassAssignment1**

Toggle perspective projection / orthogonal projection by pressing ‘v’ key.

Orbiting, panning, and zooming are implemented. (Also reference grid plane.)

1. **Single mesh rendering mode**

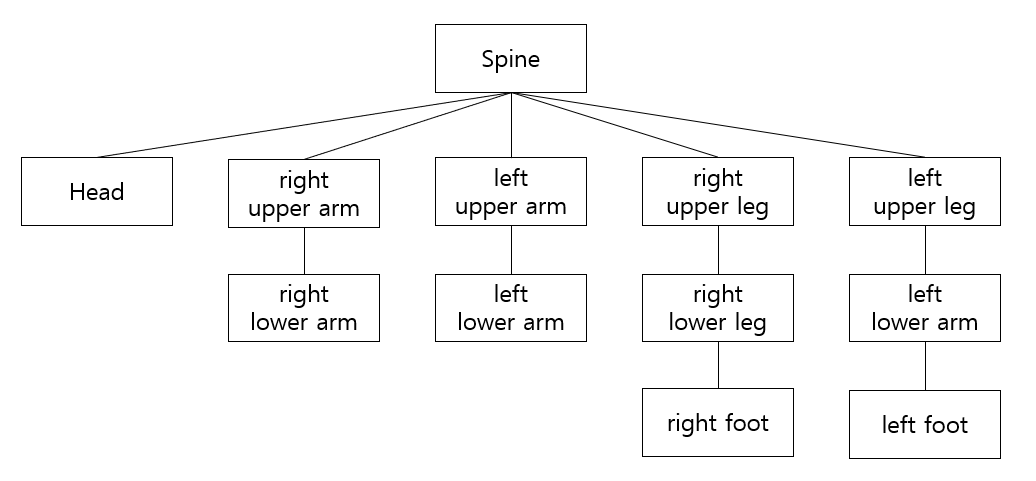
In ‘createArrays’ function, I read lines of obj file ‘f’ and split each line. Then read the file line by line, I created vertexArray and IndexArray.

If the first letter of each line is ‘v’ and the second letter is blank, the next three numbers have vertex position information. If the second letter is ‘n’ and not blank, the next three numbers have normal vector information.

When the first alphabet is ‘f’, I check the length of the line to print the specified 5 pieces of information specified in Assignment 2 pdf. And then separate the line by ‘/’ to save indices of object and normal vector of each vertex.

1. **Animating hierarchical model rendering mode**

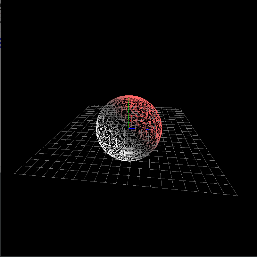
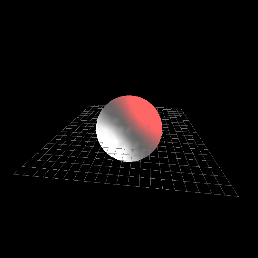
I chose the human model as the rendering model. The hierarchy of the model is as follows.



When a user presses a key ‘h’ on my viewer, my program should run in “animating hierarchical model rendering mode”. (The opposite holds as well.) in hierarchical model rendering mode, the human model starts to rotate around the y-axis even if no other input is provided. In fact, it is a model that describes a person’s walking and when the left foot goes forward, the right hand goes forward, and when the right foot goes forward, the left hand goes out. The rotated angle and the translated values were calculated directly through the position in the 3D coordinate system of the actual model.

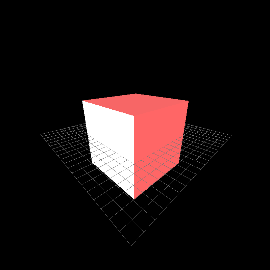
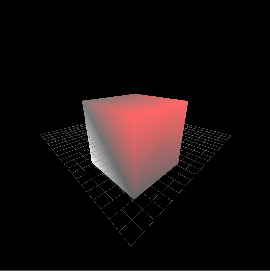
1. **Toggle wireframe / solid mode by pressing ‘z’ key**

Using ‘solid’ variable, I implemented wireframe and solid mode. If the value of the ‘solid’ is 0, I used glPolygonMode(~, GL\_LINE), otherwise I used glPolygonMode(~, GL\_FILL).

1. **Toggle [forced smooth shading] / [shading using normal data in obj file]**

I made ‘smooth’ variable to determine the smooth shading mode. In render function, if the value of ‘smooth’ is 0 I used the ‘drawObject\_glDrawArray( )’ function, otherwise I used the ‘drawObject\_glDrawElement( )’ function.

1. **Load & render a mesh that does not have the same number of vertices of all polygons**

I used ‘triangulation’ algorithm to divide a polygon into triangles. Using a given one specific and two other points, we can divide the n dimension polygon into n-2 triangles.

1. **A hyperlink to the video uploaded to internet video streaming services**

<https://www.youtube.com/watch?v=bT05H3PWf3c>

1. **Lighting configuration**

I used two light sources.

GL\_LIGHT0 = Position (3, 4, 5) / Type : point light / Color : Red

GL\_LIGHT1 = Position (-3, -4, -5) / Type : directional light / Color : White