Homework 1: Read and Draw a 3D Surface Mesh

Write an OpenGL program to read and display a 3D mesh (in .obj format).

1. Write a function, "`bool ReadOBJFile(const char filename[])`", to read in an obj file and store the elements into a vertex list and face list.

2. Implement a "`ComputeBoundingBox()`" function, to compute the bounding box of the object, its diagonal axis length, and its center, then put the camera at:

```
x_cam = x_BCenter;
y_cam = y_BCenter;
z_cam = z_BCenter + 1.5 *BDiagonalAxisLength;
```

   and look towards the bounding box center (`x_BCenter, y_BCenter, z_BCenter`).

   Some more explanation about this computation is given in the next page.

3. Implement the "`Render_Mesh()`" function to finish the OpenGL rendering. (See next page).

4. Follow "Ex8.cpp" to perform keyboard + mouse-controlled rotation (R, mouse left button), panning (T, mouse left button), and zooming (Z, mouse left button) functions.

Note 1: Please write all you codes in one "hw1.cpp" file. Test and make sure it compiles and renders correctly, then upload your "hw1.cpp" file only. I will compile and run it on my computer to see your result.

Note 2: You only need to consider the simplest OBJ format in this homework. It contains a list of vertices (each row starts with a keyword "v", e.g., `v x y z`), and a list of faces (each row starts with a keyword "f", e.g., `f vInd1 vInd2 vInd3`). Note that the vertex index `vInd` starts at `1`. So if you have store vertex positions in an array, the first vertex is at position `0`, and each vertex's location is `vInd-1`.

The homework is due: 11:59pm, Feb. 10th.

If you are late for x days, each one more day you get 5% off. Namely, your homework score will be calculated as: (the score based on your codes) * (0.95^x)

Computing Bounding Box (BB) of a 3D object

A bounding box (BB) of an object is defined as a minimal coordinate-axis-aligned box that completely contains this object.

BB can be simply defined by the minimal and maximal x, y, and z coordinate values of the objects' vertices: $BB = [x_{min}, x_{max}] \times [y_{min}, y_{max}] \times [z_{min}, z_{max}]$

The center of a BB can be simply defined as the

$$x_{BCenter} = \frac{x_{min}+x_{max}}{2};\ y_{BCenter} = \frac{y_{min}+y_{max}}{2};\ z_{BCenter} = (z_{min} + z_{max})/2$$

The diagonal axis length of a BB roughly measures the size of the object:

$$l = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2 + (z_{max} - z_{min})^2}$$

Rendering a Triangle Mesh

Each triangle mesh contains a list of faces. Each face f has three points (p_i, p_j, p_k) where each point p_i has three coordinates (x_i, y_i, z_i) .

Therefore, you can implement `Render_Mesh() based on the` following pseudo-code:

```
glBegin(GL_TRIANGLES);
//Let F denote the total number of faces
for (int ind=0; ind< F; ++ind) {
        let f be the ind-th face;
        get f's three points p_i, p_j, p_k using their vertex indices;
        get p_i's coordinates (x_i, y_i, z_i); same for p_j and p_k
        glVertex3f(x_i, y_i, z_i);
        glVertex3f(x_j, y_j, z_j);
        glVertex3f(x_k, y_k, z_k);
}
glEnd();
```