A2: Planet in space

2017313135 소프트웨어학과 권동민

Data Structure

I don't use any special data structure compared to default cgcirc program. But i used index buffer to render(not vertex buffer only) and remove toggle key.

Algorithm

Set vertex position

I used the mathematical expression as follows.

$$P(x,y,z) = (rsin\theta cos\varphi, rsin\theta sin\varphi, rcos\theta)\ (0 \le \varphi \le 2\pi, 0 \le \theta \le \pi)$$

This is calculated in function "create_circle_vertices". And to set vertex's position, we have to set the number of vertices and angle($\theta\ and\ \varphi$) of each vertex. I made la(latitude) and lo(longitude) variables used in the function and set 36, 72 each in that project. So the angle became $\theta = \frac{\pi i}{36}, \varphi = \frac{2\pi j}{72} (0 \le i \le 36, 0 \le j \le 72)$. In that range, actually the number of θ is 37 and the number of φ 73, it is just for calculating easier.

Set index buffer

For set index of vertices, we have to consider back-face culling of opengl. We set the option(culling face) true, so if the indexes of triangle vertices are set counterclockwise, the area created by triangle vertices is displayed on screen, otherwise it isn't displayed. I didn't use the origin of sphere.(because different from circle, we don't have to use origin to make areas that are displayed on screen.) so in each vertex(suppose that the index of the vertex is k), one piece of area(two triangles) of the sphere consists of indexes(k, k+(lo+1), k+(lo+2), k, k+(lo+2), k+1). These are the result of considering counterclockwise direction.

Changing coordinate system

To set the position x, y, z on screen, we have to change each axis to fit with opengl coordinate system. For this, I used projection matrix as follows.

$$projection\ matrix = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

rotation

I set rotate_flag and if "R" key is pressed, the value of flag will change. And if flag is true, the angle that is used in rotation matrix is increased constantly. (the speed depends on computer).

toggle color

When set the vertices' position, also set the color and its value is same as follows.

$$T(x, y) = (\frac{\varphi}{2\pi}, 1 - \frac{\theta}{\pi})$$

And I made uniform value (color_type) for deliver it to fragment shader. If the type is 1, set the frag color to (tc.xy, 0, 1), else if the type 2, set to (tc.xxx, 1), else set to (tc.yyy, 1).