
Computer Graphics Assignment #5

Chinchuthakun Worameth (18B00033)

February 3, 2022

1 Question

1. Define a regular icosahedron whose center is the origin and whose one edge length is 0.8, translate the center to (0,0,-2), and make an animation that it rotates around the vertical axis which passes through the center of the icosahedron. To answer this question, use the same framework of the program and replace the plane to the icosahedron. Put one snapshot on your report and describe how to compile and execute your program.
2. Define normal directions on vertices of the icosahedron, and apply diffuse and specular reflection effects on the surface with a point light source as if the surface of the icosahedron is like that of a sphere. The light is expected to be placed at a point that the effect of the reflections expresses efficiently. Parameters of the light and reflections are also expected to be set from the point of easy to understand the reflection effect. If you do not like a complete dark surface at backside from the light, you can adopt ambient light.

The equation to calculate the intensities Rrgb is defined as

$$R_{rgb} = C_{rgb} \left(\max \left(\frac{N \cdot (-L)}{|N||L|}, 0 \right) + a \right) \quad (1.1)$$

where C_{rgb} is the color of the icosahedron, N is the normal vector at a point, L is the vector of an incident light direction and a is a coefficient of ambient light.

2 Answer

Although there are two separate questions in this assignment, this report address them together by organizing into mathematical procedure followed by implementation details for the sake of readability.

2.1 General Procedure

According to this discussion [1], the Cartesian coordinates of a regular icosahedron of edge length 2 centered at the origin are given by cyclic permutations of the vertices $(\pm\varphi, \pm1, 0)$ where $\varphi = (1 + \sqrt{5})/2$. Therefore, the coordinates for 12 vertices of a regular icosahedron of edge length 0.8 centered at the origin is given by

$$(x, y, z) = 0.4 \times (\pm\varphi, \pm1, 0) \quad (2.1)$$

From these coordinates, we can systematically construct 20 triangles that define the icosahedron by finding sets of three vertices such that all pairwise distance is 2. Mathematically, let $P(i)$ denotes the coordinate of i -th vertex, a set of all triangles V is defined as

$$V = \{v = \{v_1, v_2, v_3\} \mid \|P(v_i) - P(v_j)\|_2 = 2 \text{ and } v_i \in \mathbb{Z}_{12} \forall i, j \in \{1, 2, 3\}\} \quad (2.2)$$

where $\mathbb{Z}_{12} = \mathbb{Z} \bmod 12$.

Next, given a triangle define at vertices $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$, we can calculate its normal vector as

$$\hat{\mathbf{n}} = \frac{\mathbf{p}_{12} \times \mathbf{p}_{23}}{\|\mathbf{p}_{12} \times \mathbf{p}_{23}\|_2} \quad (2.3)$$

where $\mathbf{p}_{ij} = \mathbf{v}_j - \mathbf{v}_i$. We can check whether $\hat{\mathbf{n}}$ is outward or inward by calculating $\mathbf{v}_1 \cdot \hat{\mathbf{n}}$. Specifically, the normal vector is inward if the result is greater than zero. In that case, we simply revert the direction of the normal vector and swap the order of vertices in the triangle when implementing in OpenGL. Finally, based on faces' normal vectors, we can construct normal vector at vertices by averaging normal vectors of their respective adjacent triangles.

2.2 Implementation Details

All vertex data, e.g. vertex coordinates, their normal vectors, and polygons they defined, are calculated in a separate jupyter notebook. However, when actually implement the program using OpenGL, the source codes fail to produce the desired icosahedron for some unknown reasons. Consequently, I cannot proceed to answer the second question. I also included my incomplete implementation as well.

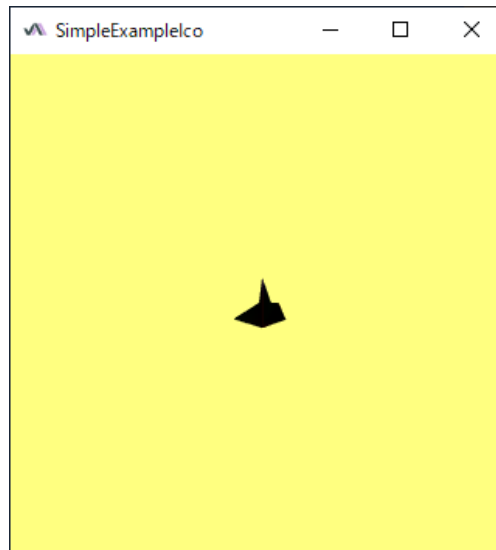


Figure 1: A snapshot of the animation window

References

- [1] "Regular icosahedron," Jan 2022. [Online]. Available: https://en.wikipedia.org/wiki/Regular_icosahedron#Cartesian_coordinates