E0 271: Graphics and Visualization Assignment #1

Weightage: 20%

Due: August 30, 2021

Goals

- Learn OpenGL graphics pipeline.
- Learn basic GLSL/Shaders.
- Learn transformations

Data and the tasks

Geometry of the objects/shapes is stored as a mesh in formats such as OFF/obj. Use the sample program which has already been shared on teams and implement the following. Protein mesh data for this assignment can be found here. The code to read these files is provided. **[N]** denotes the weightage of each task out of **100**.

1.[15] Draw the Barnsley fern fractal. The background color should be white, and the fractal should be drawn in green. Reference: https://en.wikipedia.org/wiki/Barnsley_fern.

(a) Draw a single instance.

(b) Draw two instances side by side, one of them being a horizontally flipped version of the other.

2.[20] Load and display a protein mesh, available as an OFF/obj file. Determine and apply appropriate model transformations and orthographic projection to the mesh such that it appears in the middle of the screen. On pressing key '1' apply : a) translation in -x direction b) clockwise (z-axis) rotation to camera, update the output appropriately. Compare the output if order of transformations is swapped. Write your observations.

3.[15] Rotate the mesh by 30 degrees clockwise around x-axis passing through (0,0,0). Now, color the vertices in the mesh based on their current depth (distance from the camera) using shaders.

(a) Use appropriate colormap to do this (refer notes below).

(b) Substantiate your choice in the text file which is used to describe running times (refer to submission section).

4.[25] Slice the mesh in two halves using an axis parallel plane, apply appropriate transformations to the halves so that they are detached from each other, and display both halves.

5.[25] Simulate and display a moving rubber ball trapped in a cube. The ball should start inside the cube with some preset velocity. On colliding with any of the walls of the cube, it should bounce back and stay inside the cube. For drawing the ball, use the `sphere2.off` mesh available inside the zip file here.

- For simplicity's sake, the cube should have walls perpendicular to the x, y and z axes.
- When the ball hits a wall, it should go through a squeeze animation followed by an unsqueeze animation.
- Display the edges of the cube, and apply a slow continuous rotation to the whole scene.

To facilitate better demonstration, rotate the mesh continuously along a fixed axis at an appropriate rate, for tasks **2,3,4**.

Submission

1. Format (zip following)
    (a) Directory with name <Student name>
        i. 5 directories with name <#task no.>
            A. code files
            B. make file
            C. readme file explaining how to execute/use your code
            D. observation file explaining description of methods used, analysis, discussion of running times, any key learning/observation.
        ii. <data> folder
2. Any submission beyond **11:59 pm, Aug 30th** will be considered as a **late submission**.

Notes

1. Avoid hard coding values in intermediate instructions of code. Define constants/values in global section. So that changing and testing the code is easier.
2. For the third task, to enhance the visualization, we can consider the depth to be a scalar function/field and use different colormaps. A colormap maps function values to colors. For better understanding, refer the material provided at https://sciviscolor.org/
3. For slicing task, there is no need to compute exact intersection of the mesh with the plane, triangles in the original mesh can be used as the intersection boundary. That is, there is no need to introduce new vertices and do remeshing. The triangles which intersect with the boundary can be shown on one side of the slicing plane.

Additional tasks (will not be graded)

1. Study lighting in OpenGL and implement Phong shading.
2. Design a synthetic scalar field on the protein mesh and use colormaps to visualize it.