Michael Wadih
1663993

Homework 2 Report

OpenGL

Assignment Problem(s):

This assignment asks to utilize OpenGL functions with C++ and GLUT library to translate, scale, rotate, and generally manipulate objects within the GLUT library to replicate given scenes.

The two main objects from GLUT are the famous Utah Teapot and a cube. Problem 4 is a free-for-all with the requirement of a triangle drawn from 3 vertices.

Constraints:

In this assignment I only used what the homework pdf states: OpenGL and GLUT functions such as glutSolidTeapot, glTranslatef, glRotatef, etc.

Problem 1:

I used a nested push/pop matrix for this problem.

First, I created the initial teapot at (1, 0, 0) using glTranslatef. Then, I returned the point to the origin before a for loop which created the rest of the scene.

This for loop iterated nine times, each time rotating along the z-axis and translating one unit out before placing a teapot. The scene is finished by the end of the loop with 10 teapots in a circle, each one 36 degrees from the other along the z-axis and one unit away from the origin.

Problem 2:

This problem required non-linear scaling (unspecified) and constant translation to keep the bottom of each rectangular prism flush with the ones next to it.

I first generated a primary cube and translated it to the left some amount to match the image. After this, I used a for loop to iterate through the rest of the cubes, each one a constant distance from the other (touching perfectly).

While looping, I used a scalar value for the y-direction which increases at a rate apart from the for loop. Since scaling in any direction scales in both positive and negative direction, I translated each object in the y-direction by that scalar.

The scene is complete by the end of the for loop.

Problem 3:

I used another nested push/pop matrix for this problem.

Since the triangle is equilateral, it could be easily formed by iterating through each “level” and increasing each subsequent row’s number of teapots by one. I used a nested for loop to solve this.

The outside for loop iterates through the whole triangle by layer or row. Each iteration translates down and to the left, then lets the inside for loop generate a number of teapots equal to what number row the outside loop is on. Within each row, the teapots’ spacing is constant.

Michael Wadih
1663993

This is where the nested push/pop matrix comes into play. Since the nested matrix is placed after the outer loop's translation down and to the left, after the inner loop finishes, the program starts to use the outer loop's translation again until the next glPushMatrix().

The scene is complete after the outer loop finishes.

Problem 4:

I decided to use two new GLUT objects (new to the assignment) for this problem.

First, I created a solid cylinder that points in the z-direction (this is how it defaulted). Then, since I needed to use a nested matrix and a triangle, I thought I could do something similar to problem 1 by rotating around a fixed point.

Within this nested matrix I wanted to create a triangle to duplicate and rotate the duplicates around the z-axis. First, I translated the point to right past the positive flat end of the cylinder. Then, within a for loop that iterates 6 times, a triangle is created with the same coordinates each time that is rotated 60 degrees each time. This creates an image reminiscent of a propeller or turbine for an airplane.

Finally, after the nested matrix, I created a cone that sits in the middle of all blades on the turbine. The scene is completed after this.

Results:

Overall, my results are very similar to what was given. Since no formulas, coordinates, or object sizes were given, I consider my results to be a success. The only discrepancy I can easily notice is for problem 2. My "graph" curves at a different rate, but it is very difficult to match it perfectly since there was no scalar given.