COSC 4370 - Assignment II

Name: Nimet Ozkan | ID: 1961233

February 2022

1 Objective

Assignment requires the implementation of four different algorithms to render various 3D OpenGL shapes. The assignment content provided three different 3D models and asked us to design them our own way. The first object is a cycle but created using 10 3D Utah teapots, the second; is a ladder and is built using 15 cubes in OpenGL; the third is a pyramid of teapots using 21 Utah teapots. Finally, the fourth is a deliberately left blank problem because it requires us to design something using our creativity.

2 Method

There are four empty functions that need to be filled in this assignment: **problem1, problem2, problem3 and problem4**. No need to change any other part of existing code. The code runs on its own, showing what problem is currently running, with the variable "*curProblem*" we can enumerate depending on the current problem we are working on, and we would like to see the result. For Example: When we say *curProblem = 1*, we mean running a "Utah *teapot cycle*" or *curProblem=3* to create a "Utah *teapot pyramid*". All these functions (problems) are implemented by making use of the OpenGL graphics library and its available functions. Once the desired image is created in the window, we can zoom in/out with the right mouse button and rotate it with the left mouse button. Use the right arrow to look at the next object, and the left arrow on the keyboard for the previous object. All the requirements to run and use the code were shown in the README.md file.

3 Implementation

The 3D functions used in the assignment are provided from the ready-made functions in OpenGL. Adding **"<GL/glut.h>"** makes functions in **glu.h** and **gl.h** usable and useful. For the given problems, all function implementations used different algorithms but used common OpenGL library transformation and matrix functions. For example: *glRotatef* is a function that produces a specified number of counterclockwise rotations around the vector (x,y,z), and this is used in each problem implementation, while *glScalef* specifies the scales along each axis according to the specified factors, and so on.

3.1 Problem #1: Utah Teapot Cycle

In the assignment instruction, it was determined that 10 teapots rotated around the center of the camera as a full circle between 0 and 360 degrees. Therefore, the initial angle is shown as 0, and the full cycle is 360 degrees. For the algorithm, teapots of size 0.3 were created with *glutSolidTeapot(teapot_size)* while this cycle was not yet finished. To get 10 teapots by testing in this cycle, the radius to move forward after each teapot creation is determined as 36. for each teapot we translated the drawing with z = 0, saying *glTranslatef(0,0,0)* then rotated it along the z-axis using glRotatef(radius,0, 0, 1), then moved it along the x-axis *with glTranslatef(1,0, 0).*

3.2 Problem #2: The Cube Ladder/Stairs

As observed in the assignment instructions, a 15-level ladder is required in the assignment instructions, in a loop where 15 contiguous cubes of 1 scale each are created using *glutSolidCube(1).* Each newly generated cube higher than the previous with an increase of 0.26 from the y-axis is . Each new cube is designed larger than the previous one by scaling *glScalef(1, curr_num*.5, 1)* larger on the y-axis (height).

3.3 Problem #3: The Pyramid of Utah Teapots

A 6-high pyramid was built. In other words, six layers/levels or rows. Starting from the first level (1), a camera was placed in the middle of the pyramid and teapots were created in each layer with a size of 0.25 and a distance of 0.6 between them. These operations were achieved using *glTranslatef(ratio, (pyramid_level/2-level) * .6f, 0)* and *glutSolidTeapot(teapot_size).* To create a balance between the levels, a ratio is determined depending on whether the current level is odd or even.

3.4 Problem #4: Creativity; Articulated High Five!

Using additional resources, an orange 5-finger articulated hand was obtained to capture the resemblance to human skin color. In this particular problem, the hand is planned piecemeal as the top or bottom of a particular piece, similar to human bone joints. This orange 3D articulated hand features the upper and lower joints of the hand, thumb, index finger, little finger, and middle finger. They are arranged so that one hand stays around the y-axis, and the index and middle finger are scaled vertically, as if making a peace gesture. For each joint, the *glTranslatef , glRotatef, glTranslatef, glPushMatrix, glColor3f, glScalef, glutSolidCube, glPopMatrix, and glPopMatrix* functions are used and specified, respectively.

4 Conclusion and Results

The outputs of the program were .png files. A loop for the first problem, a ladder for the second, a pyramid for the third, and a hand for the fourth problem are easily visible when viewed.