COSC 4370 - Homework 2

Name: Cyrus Shekari **PSID:** 2042446

February 2023

1 Objective

This program aimed to create multiple 3D scenes using the OpenGL graphics library. The first scene involved the creation of ten teapots arranged in a circular pattern, generated using OpenGL's solid teapot function. The second scene consisted of a staircase-like arrangement of cubes, which gradually decreased in size and descended in a vertical direction. The third scene featured a reverse pyramid of teapots, with six teapots on the first row, five on the second, and so on, with only one teapot on the final row. Lastly, the fourth scene represented an articulated hand with a triangular base, where both the index and middle fingers were extended.

2 Method

This program heavily relies on the OpenGL graphics library to generate four different 3D images. Each image was created using a specific method implemented as a function in the program. The first image is a circular formation of teapots generated by iterating through a while loop that creates a new teapot with the *glutSolidTeapot()* function. The teapot was then positioned in a circular formation using a series of translations and rotations performed by the *glTranslatef()* and *glRotatef()* functions. The second image, a downward staircase, was created by using a while loop to generate a cube with the *glutSolidCube()* function. The cube is scaled in the y-axis direction with the

glScalef() function to create a series of cubes that appear as a staircase. The third image is a reverse pyramid of teapots. This function featured a nested for loop that calculated the horizontal and vertical positions of each teapot in the pyramid. The teapots were then placed in their correct positions using translation functions. The fourth image was an articulated hand approximation with both the index and middle fingers extended. A combination of translation and rotation functions were used to position the fingers and hand. Each finger consisted of two cubes representing the upper and lower joints of the finger.

3 Implementation

The implemented program encompasses 4 functions that work together to generate the desired images using various functions from the OpenGL graphics library. These functions include *problem1()*, *problem2()*, *problem3()*, and *problem4()* that are each responsible for drawing the circular teapots, downward staircase, reverse teapot pyramid, and the articulated hand respectively.

3.1 problem1

The *problem1()* function creates ten teapots arranged in a circular pattern and begins first by initializing three float variables: *size_of_teapot*, *one_cycle*, and *radius*. The *size_of_teapot* variable defines the size of each teapot to be drawn, while the *one_cycle* variable sets the number of degrees in one cycle. The radius variable is used to determine the rotation angle of the teapot around the z-axis. The function then enters a while loop that runs until the camera angle reaches 360 degrees. Inside the loop, the current matrix is pushed onto the stack using the *glPushMatrix()* function, and the object is moved to the center of the screen using *glTranslatef()*. The object is then rotated around the z-axis by the angle defined by the radius variable using *glRotatef()*. Next, the object is translated along the x-axis by two units using *glTranslatef()*. A solid teapot is then drawn on the screen using *glutSolidTeapot()* with the *size_of_teapot* variable as its

argument. After the teapot has been drawn, the current matrix is popped off the stack using *glPopMatrix()*. Finally, the radius variable is increased by 36 to ensure that ten teapots are drawn on the screen.

3.2 problem2

The *problem2()* function creates a downwards staircase and begins by initializing variables that will be used to create a set of stairs. The function sets the maximum number of stairs (*max_stair*) to 20, and the current number of stairs (*curr_num*) to 1. It also initializes the position of the stairs in the x, y, and z directions, and sets the size of each stair to 0.8. A counter variable is also created to track the number of stairs to be drawn. The function then pushes the current matrix onto the stack with *glPushMatrix()* and rotates the scene 180 degrees along the y-axis. A while loop is used to create the set of stairs. Within the loop, each stair is drawn using the *glutSolidCube()* function and scaled along the y-axis by the current number of stairs multiplied by 0.5. The position of the stair is translated in the x, y, and z directions using the *glTranslatef()* function. The y-position and z-position are decreased for the next stair by adding the *increment_height* and *stair_size* values, respectively. Finally, the function pops the current matrix off the stack with *glPopMatrix()* to restore the original transformation matrix.

3.3 problem3

The *problem3()* function consists of a nested for loop that generates a pyramid made of teapots, arranged in rows. The variable *pyramid_level* determines the number of rows in the pyramid, while the variable *teapot_size* sets the size of each teapot. The outer for loop iterates through the rows, while the inner for loop iterates through the teapots in each row. In each iteration of the outer loop, the function calculates the horizontal offset for the row, the starting position for the first teapot in the row, the number of teapots in the row, and the spacing between teapots on the row. The inner loop then generates each teapot in the row by

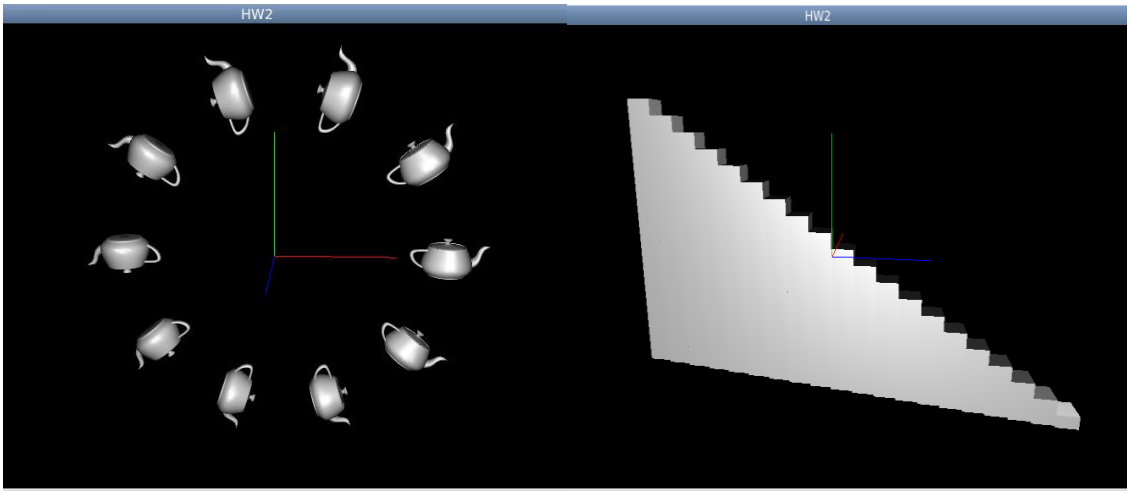
calculating its horizontal and vertical position and calling the *glTranslatef()* and *glutSolidTeapot()* functions to draw it. The *glPushMatrix()* and *glPopMatrix()* functions are used to save and restore the transformation matrix before and after each teapot is drawn, so that each teapot is drawn independently of the others.

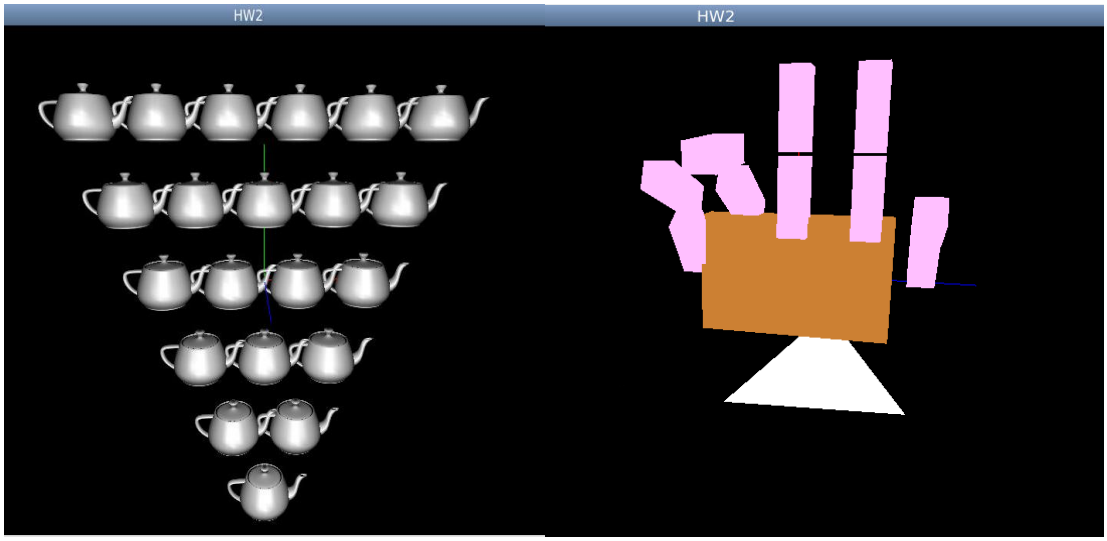
3.4 problem4

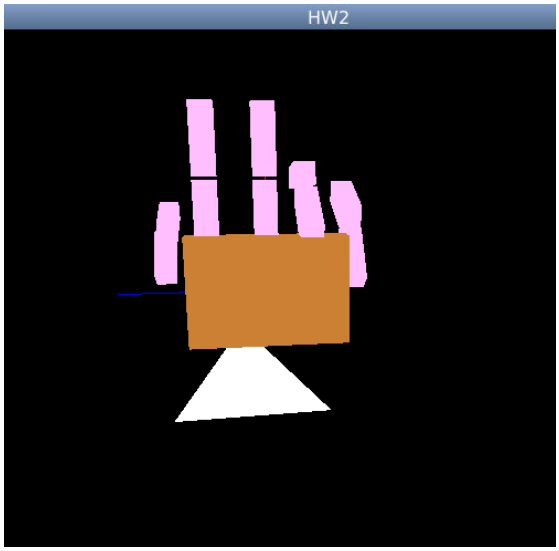
The *problem4()* function creates a 3D model of a hand. The function first clears the depth buffer and disables lighting to see the figure. It then sets the RGB color values to make the displayed hand pink. The function uses static float values to store the angles of rotation for each finger and the palm of the hand. The function then proceeds to create each finger of the hand using translations, rotations, and scaling with respect to the palm of the hand. Each finger is composed of two joints, and the joints are connected by scaled cubes using *glScalef()*. The cubes are drawn using *glutSolidCube()* and are rotated depending on which finger they belong on. Since the image aimed to produce a hand with both the middle and index finger pointing up while the rest of the fingers were bent towards the palm, each cube object that represents a joint on the finger must be rotated appropriately. The function also creates the palm of the hand as the base of the model using a *glPushMatrix()* call to preserve the current model view matrix. Nested applications of *glPushMatrix()* were used to render an equilateral triangle that represented a stand for the hand to rest on. Finally, the function uses *glPopMatrix()* to restore the original model view matrix.

4 Results

The results of the program are four 3D images that each represent a circular arrangement of ten teapots, a downward staircase of cubes, a reverse pyramid of teapots, and a pink hand on a triangular stand that has both its index and middle finger pointed up to emulate a peace sign.


HW2
HW2


HW2
HW2


HW2