

COSC 4370 Homework 2

Ivan Madrid

Assignment Problem:

Use OpenGL commands to recreate the first three images shown in the assignment, and then create an original image that uses OpenGL in a nontrivial way, with at least one instance of nested `glPushMatrix` and rendering at least one triangle by feeding its coordinates directly.

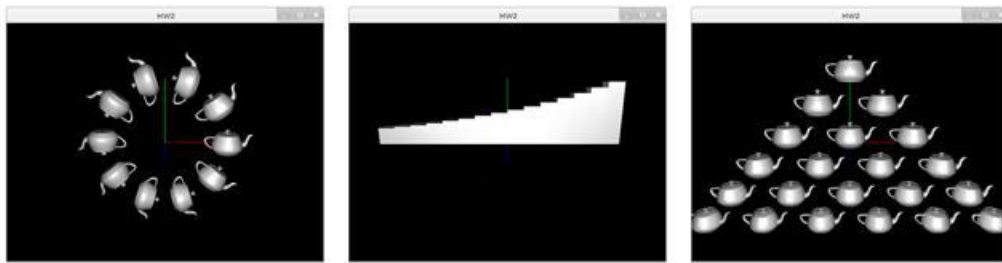


Figure 1: Images 1, 2, and 3

Algorithms and Implementations:

We divide the problem into three sets for the first image, using the teapots in the first quadrant as the basis. In clockwise order, teapot 1 is the basis for teapot 5, 6, and 10. Teapot 2 is the basis for teapot 4, 7, and 9. Teapot 3 is the basis for teapot 8. The algorithm works by first defining the basis teapot, then translating and rotating it to create the next teapots. The steps are done for all sets.

For the second image, the algorithm creates fifteen columns of cubes with increasing height. We first define the height increase between columns as 0.05, and the height of the pillar is calculated as the height increase times the column number. After defining these variables, a cube is created at the bottom and then using the height increase, more cubes are created to fill in

the blank space between the top and bottom cubes of the column. Then the increasing height variable is increased by 0.001 to make the increase in height not be constant.

For the third image, the algorithm creates six rows of teapots with the number of teapots as the row number. To start, we define the height of the first row as 1.45, the height decrease between rows as 0.55, and the distance between the first teapot and the y-axis as 0.5. Then, we start by translating to the first teapot position, with x being the distance between teapots times the row number, starting at 0, y being the height, and z being 0. Then the teapot is created and translated two times the distance between the first teapot and the y-axis to the left until we have created all the teapots needed for the row. The height decrease variable then decreases the height, and the steps are repeated for all six rows.

For the fourth image, I created a smiling face wearing a party hat using a sphere as the head, two smaller spheres for the eyes, a triangle for the mouth, and a cone for the hat. I started by first rotating the first sphere by 45 degrees in the y-direction and then created the sphere with a radius of 0.6, 10 slices, and 10 stacks. Then two push matrices are pushed, and then we create the triangle with the vertices being at (-0.2, -0.05, 0.6), (0.0, -0.2, 0.6), and (0.2, -0.05, 0.6). The matrix is popped, and the spheres for the eyes are created with a radius of 0.2, 10 slices, and 10 stacks. The first sphere is translated to (-0.2, 0.15, 0.35) and the second to (0.4, 0.0, 0.0). Then the cone is created at the top of the sphere and rotated by -180 degrees in the y-direction and z-direction and 45 degrees in the x-direction.

Results:

The results from the program running from the command line are:

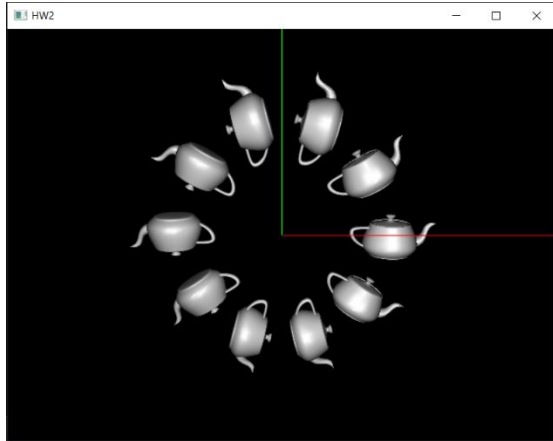


Figure 2 Image 1

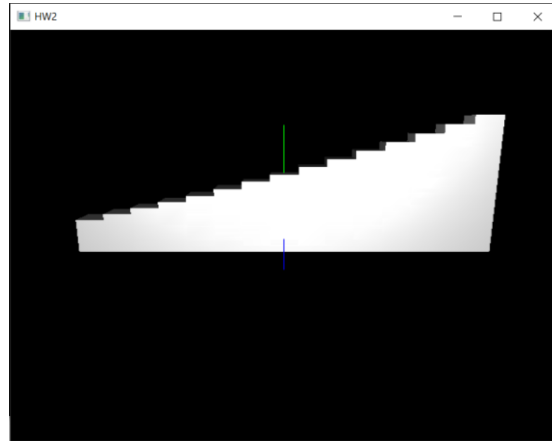


Figure 3 Image 2

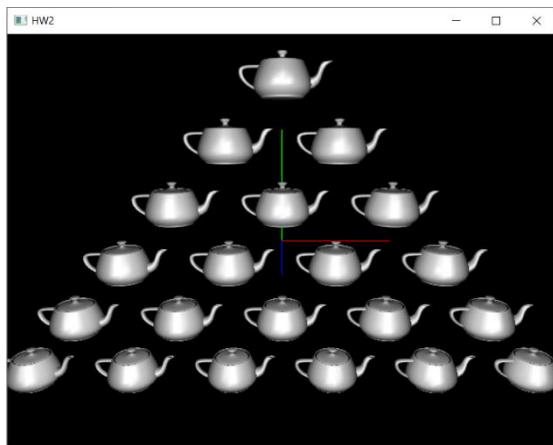


Figure 4 Image 3

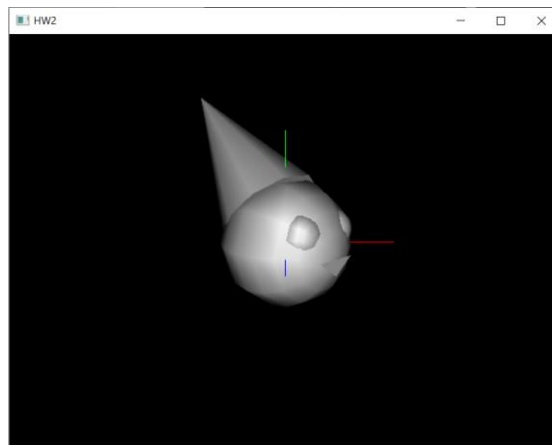


Figure 5 Image 4

All the images display what the must and image 4 is created within the criteria.