COSC 4370 - Homework 2

Name: Juan Yanez PSID: 2029097

October 2022

1 Problem and Objective

This assignment wants the creation of four different 3d images with openGL. The first three are simple models, ten teapots forming a circle, a staircase with twenty steps, and a pyramid starting from six teapots and ending at one teapot at the last row. The fourth image is anything creative, however, it needs to use openGL's transformation mechanisms in a nontrivial way, including a nested glPushMatrix instance and creating at least one triangle by manually placing its coordinates.

2 Method

The first three images are simple to implement using openGL's resources and some loops. All models start out with a glPushMatrix() to have a space for allocation of the shapes needed to replicate the images. The fourth model uses multiple nested glPushMatrix to be able to control groups of objects together.

3 Implementation

All the implementation is done in main within predetermined functions using openGL while work will be put in a backup txt file. The first two images use loops to simplify the creation of the models while the third image uses while loops for each row of teapots in the image, making it easier to find problems with my numbers.

3.1 Teapot Circle

The teapot circle has two variables that are needed to create the shape and move to the next position. Size determines the size of each teapot and r determines the position of each teapot in terms of rotation, starting at 0 degrees. The while loop continues on until r was less than or equal to 360. Inside the loop, the teapot is first rotated 90 degrees to align it with the y axis just like the reference image. Next the teapot is rotated(axes) r degrees according to what position the teapot is in, this r value increases by 36 each time to create 10 teapots each pointing outward with respect to what angle they were in. After the rotation, the position is shifted 1.5 units on the x

axis to create a similar radius to the reference image; this shift stays constant throughout the loop. When put together, each teapot is placed on y = 0, x = 1.5 after having the axes rotated according to each teapot.

3.2 Stairs

The stairs are implemented with a for loop, with the loop being limited by how many steps there are on the stairs. First cubeNum is defined at 0, this tracks which cube the loop is in to be able to scale it properly, x = -1.35 which helps determine the start position of the model, where stairs increase in size from right to left. The y position helps each cube stay level with the previous one on the bottom of the model. Size determines each cube size and height determines the height of the next step. Inside the for loop, limited to 20 iterations for each step, first the cube is shifted to its correct position, starting from the right side smallest step, and the bottom is extended to align with the rest. The cube is then scaled to its proper position by keeping x and z constant but increasing y with 1.5 + (cubeNum *.2) which increments the step size through each iteration. The cube is then created and increments go as follows, x+=.15 to space the cubes, making each step equal length, y+=height*.2 to increment y so it keeps aligning with the bottom of the model, and cubeNum++ to track which cube the algorithm is on.

3.3 Teapot Pyramid

The pyramid is implemented with multiple while loops to allow for fixing individual rows when creating the model. First some universal variables are declared to avoid any variations between the teapots, num = 0 which tracks each pot, and spacing which changes per loop to create a new start point which go from -1.15, -.95, -.75, -.55, -.35, -.15 which determines the position of each teapot for each row, going from right to left. There are 6 while loops, each loop is similar, first the position is shifted each time to the right by, incremented by adding .45 units to spacing in each iteration, while keeping y constant at its respective height. Each loop, the shift is at y= 1.1, .55, 0, -.55, -1.1, -1.65 to create the proper shape of an upside down pyramid. A teapot is then placed, with size .15.

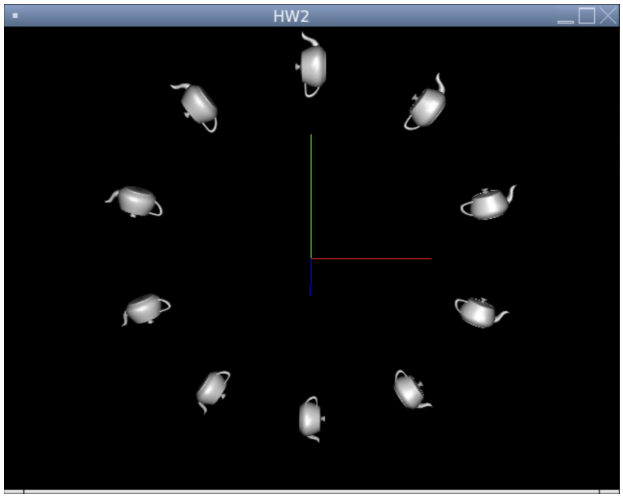
3.4 Creative Problem

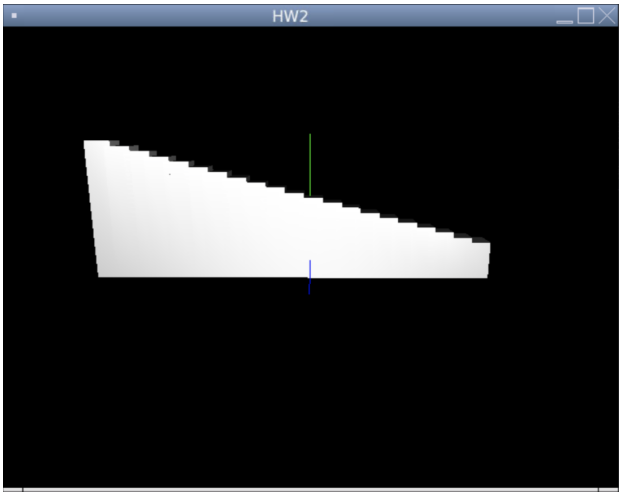
The fourth problem involves making a spider with two eyes and six legs. First the base for the model was created as a triangle by selecting each coordinate manually. The triangle was created by calling glBegin(GL_TRIANGLES) and following up with three calls to position each corner, after the triangle's coordinates are selected, glEnd is called. The creation of the spider model involved nesting glPushMatrix to keep the legs and eyes in their proper place. First 3 variables are declared, r = .1, h = .4 which is the height of the leg base connecting to the body, and ss = 30 which allows for a round cylinder. The main body is positioned first to guide the legs and eyes. To hold everything glPushMatrix is used, inside the main matrix, the body is placed using glutSolidSphere after being shifted up .5 units on the y axis, with .5 as its radius and 30 for slices and stacks. The eyes are the next thing that is placed, they consist of 2 spheres. First another

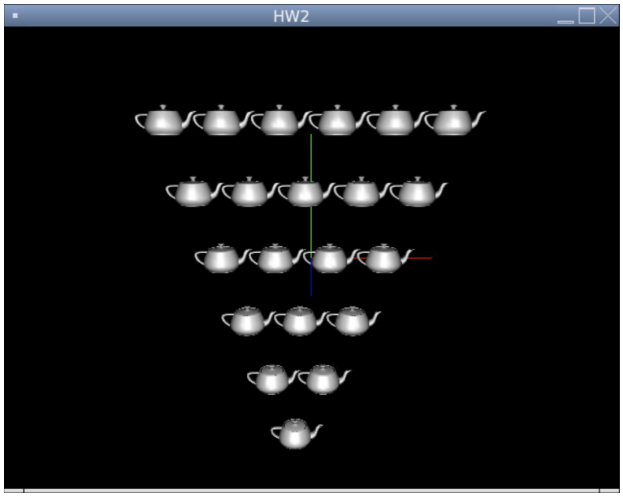
matrix is pushed and nested inside the main glPushMatrix for the right eye to keep it aligned with the main body, the sphere is shifted .2 units to the right and .4 units forward to make them visible. For the left eye, another nested matrix is called inside the right eye's, this way only a simple glTranslated needs to be called shifting the eye -.4 units. The legs are the last objects implemented, they consist of 4 objects, 2 spheres with a radius of .4, however they are scaled with glScaled to avoid having to change up the radius. Inside the main body's pushMatrix, each leg has another pushMatrix nested and the first leg follows one main process. First everything is scaled with glScaled by .9 units, then shift the cylinder to the correct position by having x=.2 and y=.5. The axes are then rotated by 90 degrees on the z axis and the cylinder is positioned. At the end of each cylinder there are spheres to signify joints which are positioned by shifting the sphere on the rotated z axis by .4 units, putting it at the end of the first cylinder. All axis values are then scaled up by 1.1 to create a sphere with a slightly larger radius than the cylinder which is created. The same exact steps are followed to create the second half of the leg, which is nested inside the main leg's nested glPushMatrix. For each leg after the main one, the same steps are also followed but have an extra call to rotate the whole leg model on the y axis for 6 legs being at the left and right sides of the main body sphere.

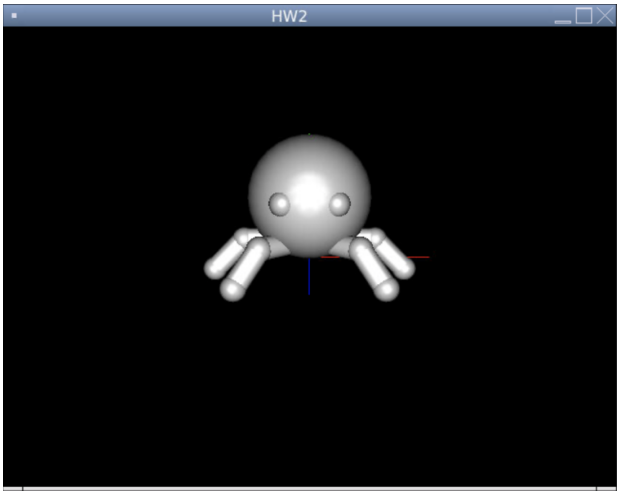
4 Results

The results require a script to be ran in order to create a 3d environment where the models can be viewed.


HW2


HW2


HW2


HW2