COSC 4370 - Homework 1

Name: Morrison Fowlks PSID: 1855790

October 2021

1 Problem

The purpose behind this assignment is to implement certain functions that create the desired illustrations through the use of OpenGl and it's transformation mechanisms, as well as shapes like glutSolidTeapot or glutSolidCube. This was achieved through the use of creativity and determination, where one attained an overall familiarity and confidence with the use of OpenGl.

2 Method

When generating the desired illustrations in the given main.cpp file, one should first become familiar with the many functions and shapes used in OpenGL. In this assignment, one will be working with the shapes glutSolidTeapot or glutSolidCube, depending on each function. In each sequential problem function, one will be using functions from the OpenGL library like glPushMatrix, glPopMatrix, glRotatef, and glTranslatef the majority of the time for this given assignment. Put simply, glPushMatrix pushes the matrix onto the stack whereas glPopMatrix pops or removes the matrix from the stack. The other functions in this assignment are somewhat rudimentary, with glRotatef simply rotating a given object and glTranslatef translating a given object. All listed functions are based on an x, y, and z plane, making many of these functions have parameters which will affect said plane. One will use the past functions to implement the four total problem functions, in turn creating unique and creative results.

3 Implementation

When creating the warranted problem functions one will first need to grasp the functions listed previously, as well as the two shapes one will be dealing with. Each problem function has its own creative yet challenging task, although if given time and effort, one will be able to expertly solve said challenge, and gain a better understanding of OpenGl and computer graphics as a whole.

3.1 problem1

The purpose behind this illustration is to construct a circle of the object glutSolidTeapot, where there are 10 objects tied to that of the x and y plane. When creating this function, one will first create an initialization of said objects, and the angle of each said object (being 360, that of a full circle). One would then create a "for" loop looping through the entire 360 degrees, first pushing the matrix using the function glPushMatrix. In this loop one would then rotate and translate each teapot object, which will be of size 0.2. Once this is complete one will end the loop by implementing the function glPopMatrix. Once the following steps have been taken, one will achieve the wanted illustration of problem one, creating a rather interesting 3D circle.

3.2 problem2

The purpose behind this illustration is to construct a set of stairs, using the object glutSolidCube, with a total of 15 steps, all tied to the x and y plane. When creating this function, one would first initialize said steps, push the matrix, and translate the first step to the position -1.60 on the x axis. One would then create a nested "for" loop, one for that of the number of steps, and another for the creation of each singular step. In this nested "for" loop creating each step, one would first push the matrix onto the stack, translate and create each singular block, and then pop the matrix using glPopMatrix. Once the following steps have been taken, one will have a beautiful illustration of stairs, with a total of 15 steps.

3.3 problem3

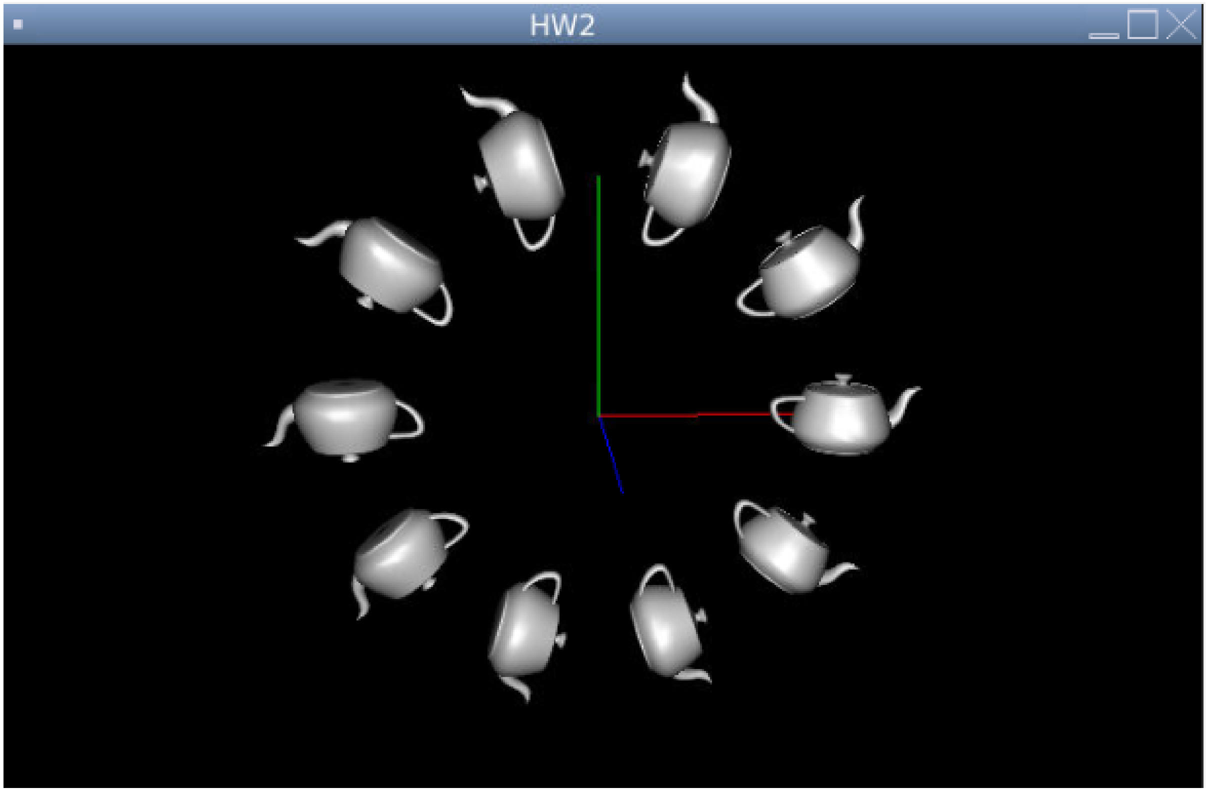
The main purpose behind this illustration is to construct a pyramid of 21 teapots, of the object glutSolidTeapot, all tied to the x and y plane. When creating this function, one should first initialize the height, distance between each teapot, x, and y of the given pyramid. One would then create yet another nested for loop, this time having one"for" loop for each row, and one "for" loop for each teapot. In the nested "for" loop generating each teapot, one would first push the matrix, translate the matrix, generate each teapot of size 0.2, and finally pop the matrix. One would then make x= x-dist/2 and y=y-0.5 in the main "for" loop which generates each row. This all in turn would result in one magnificent and truly unique pyramid, perfectly as intended.

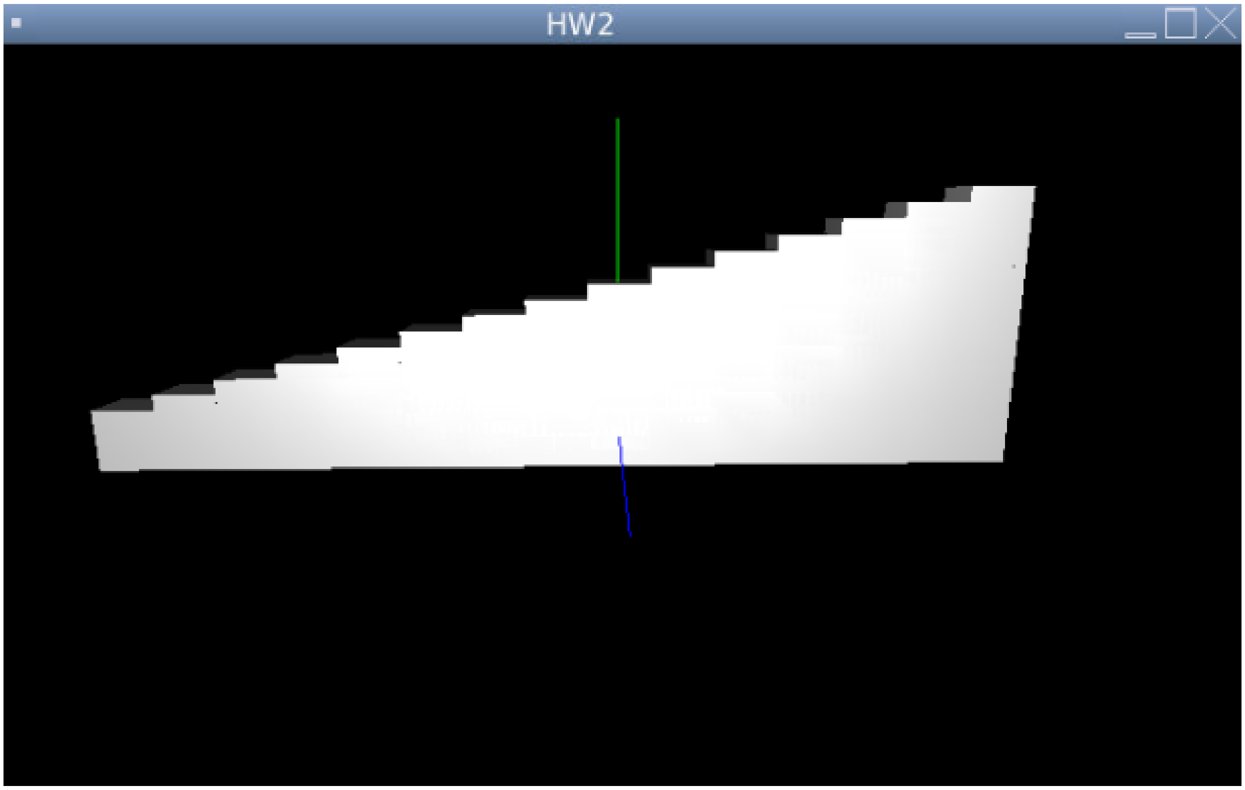
3.4 problem4

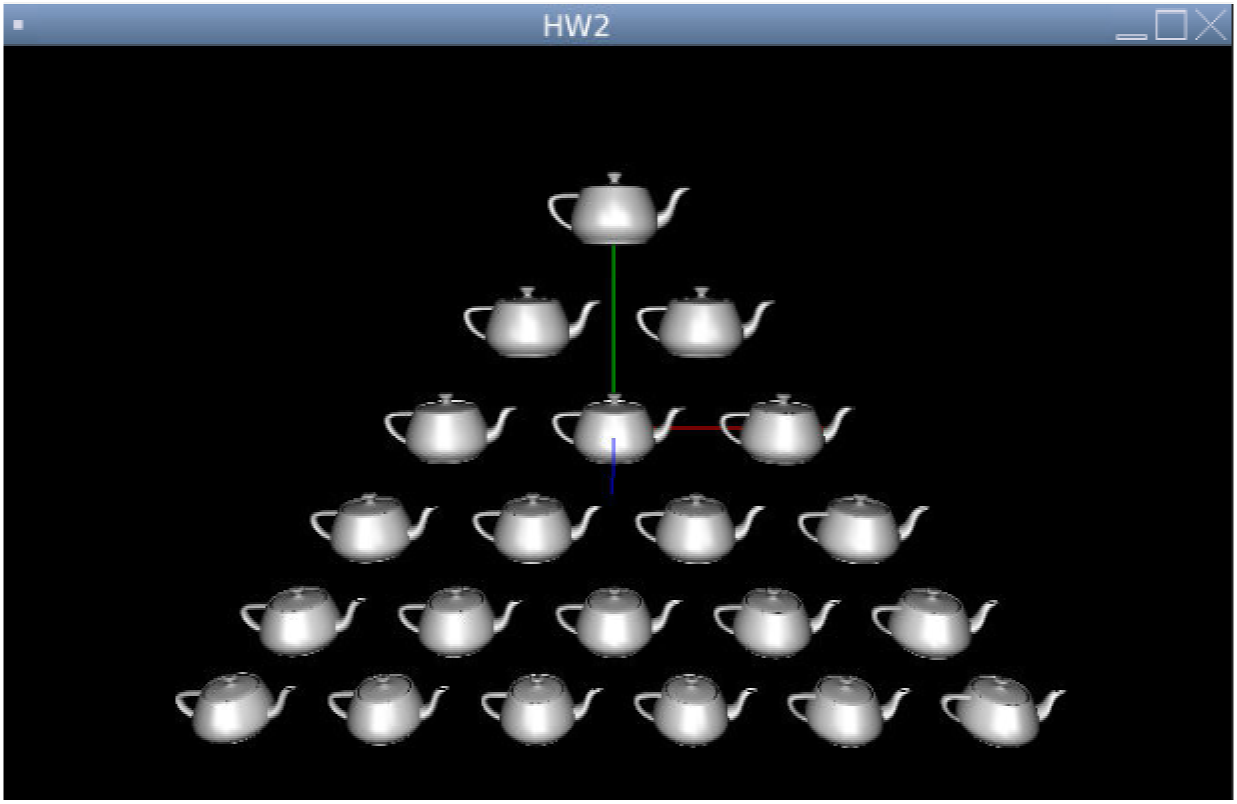
For this problem, I decided I wanted to create a sort of entrance way, like that of the game Minecraft, with a mix of spanish missions seen throughout much of our home state. With that being said, I first created 2 triangles by feeding in its coordinates directly, and made these the door of the entrance way. I did so by implementing glBegin, and putting the vertex of each triangle in the desired coordinates, making a simple door for my entrance way. Once this was done, I ended each triangle with the function glEnd. After I created the triangles, I then sent out to work on the frame of the entrance way. I did so by creating blocks, and first pushing the matrix onto the stack. Once the matrix was pushed, I translated the first block to -0.7, and then created a cube that was the size of 0.4. Once this was done I translated each sequential block, and maintained the size of each, until finally popping the matrix. Once this was complete, I had my minecraft inspired spanish mission entrance complete, even with a little door in the middle.

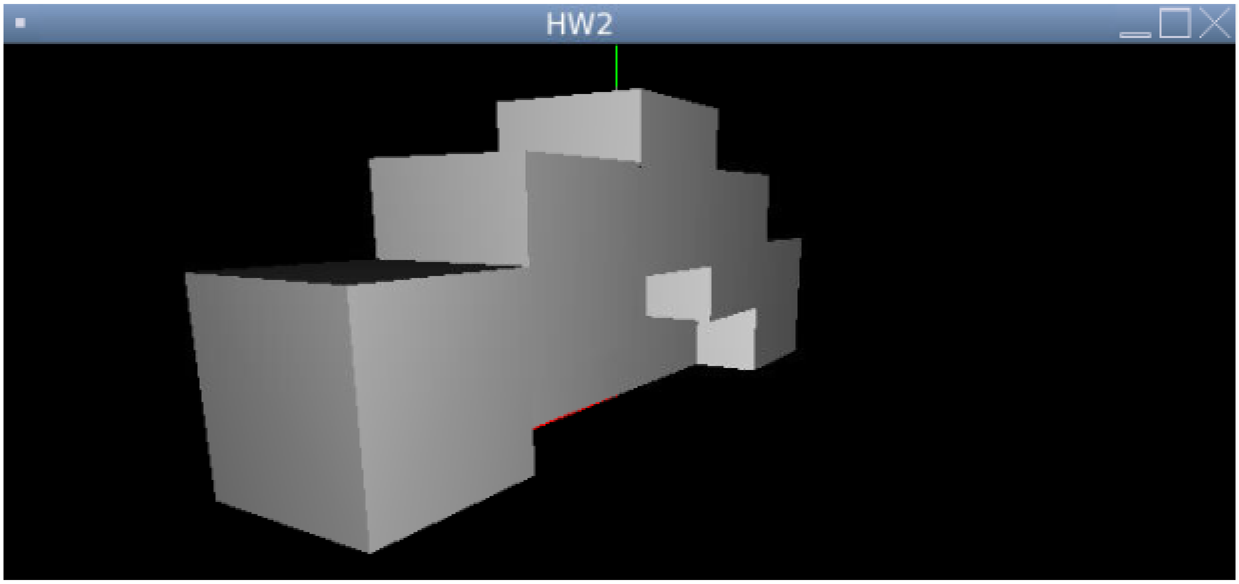
4 Results

Given the main.cpp file and OpenGL library used in this project, one was able to create numerous scenes and illustrations that were once thought to be impossible. After completing this given assignment, one should now feel a sense of accomplishment and pride being able to use their creativity and coding skills in a positive and efficient way.

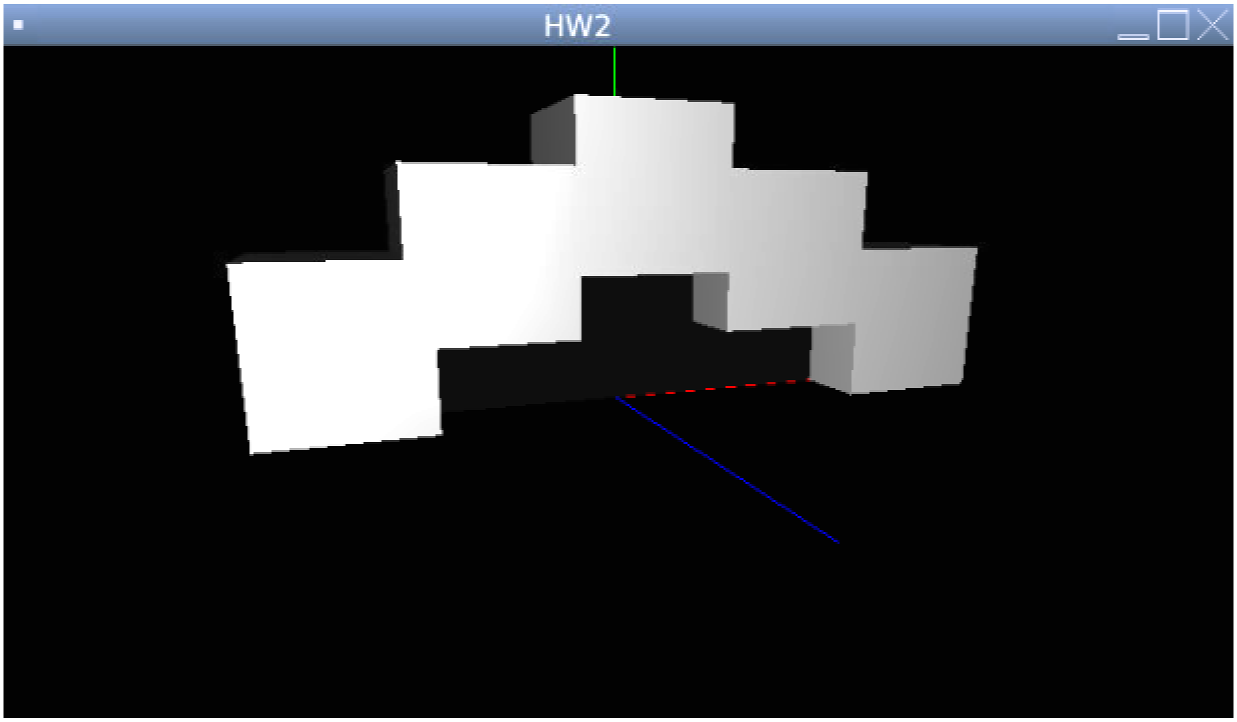

HW2


HW2


HW2



HW2



HW2