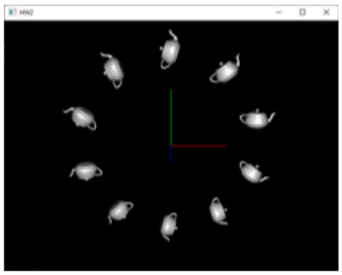
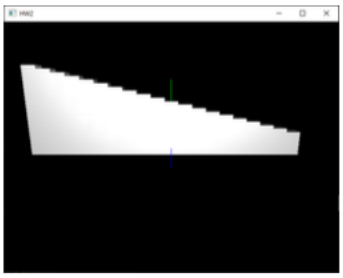
Sham Hintolay
1618608

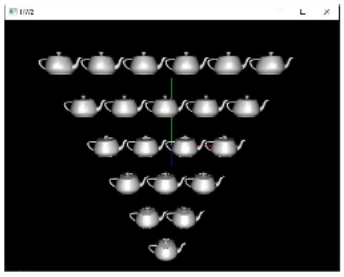
Homework 1 Report

1. Problem

In this Homework, we are tasked with recreating the following images.







Sham Hintolay
1618608

2. Method

In order to succeed in the assignment, we only made changes in the “problem functions” within the main.cpp file.

For problem 1, we need to layout teapots in a circular fashion, while also rotating the teapots. This was completed by using gltranslatef() to move along a circular path, while using glRotatef() to rotate the teapot, and generated the teapots using glutSolidTeapot().

For problem 2, we need to create a stair case. This was completed by using glutSolidCube() to generate the cubes that replicate the staircases, and used glTranslate(), along with glPushMatrix() and glPopMatrix(), to move to the side and stack cubes.

For problem 3, we needed to create a upside-down pyramid of teapots, starting with 6 and decreasing down to 1 on the bottom. This was completed by starting from the top to bottom, and using glutSolidTeapot() to generate the teapots, and using glTranslatef() to move to the side, and down, to continue placing teapots. glPushMatrix() and glPopMatrix() were also used to help moving to the side.

Sham Hintolay
1618608

3. Implementation Details

For problem 1, we used a while loop that with the condition (i <= 10) to create 10 teapots. We also initialize a theta value at 36 degrees, and increment theta by 36 with every iteration of the loop (since 360 degrees/ 10 teapots = 36 degrees per teapot). Then, using theta, we calculate the placement of each teapot using the following formulas :

```
xf = (cx + ((float)(px - cx) * cos(theta)) - ((float)(py - cy) * sin(theta)));
  yf = (cy + ((float)(px - cx) * sin(theta)) +((float)(py - cy) * cos(theta)));
```

These formulas help us find the x and y of our placement along the circle. Cx and Cy represent the center of the circle we want to rotate around, while px and py represent how far we want to start out from the center. The changing theta value is what moves us along this imaginary circle and is what makes sure we get different values everytime to translate the teapot to. When placing teapots, we also want to rotate the actual teapot, but we do not want the rotation to be saved into the current position and orientation, so we use glPushMatrix() and glPopMatrix() to make sure the rotation of the teapot isint carried forward in the loop.

Sham Hintolay
1618608

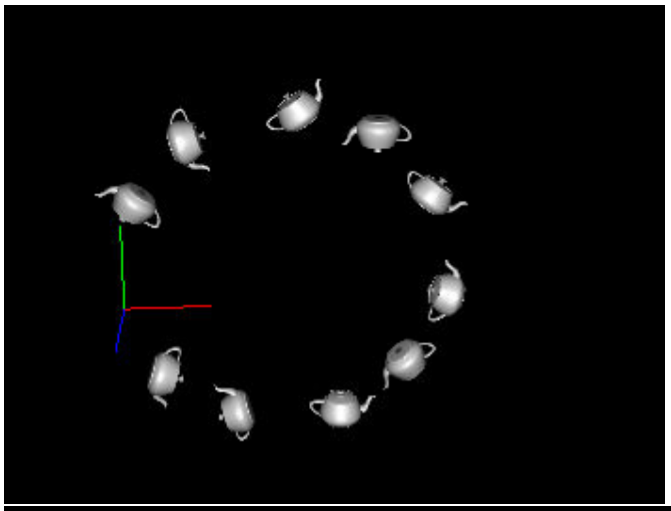
For problem 2, we need to use a nested loop. The outer loop will be used to designate the placements from the bottom of the staircase, and the nested loop is used to build on top of these designated spots to create elevation from the previous stair to the next. 15 stairs are being created, so the outer loop is controlled by the conditional while(i < 15), with i starting at 0. We also use i to represent how many stairs to build up on the current spot we are at. So for example, after placing a cube down, since i=0 we don't build on top, but the next loop when i=1, the nested loop is activated, and we build i-times on top of the base stair. The inner while loop is surrounded by glPushMatrix() and glPopMatrix(), because we do not want the translations that come from stacking the stairs to be present after we finish stacking at a specific spot. glTranslatef() is used in the inner loop to move in the positive y direction to stack stairs, and glTranslatef() is used in the outer loop to move in the positive x direction to begin the process for the next stair.

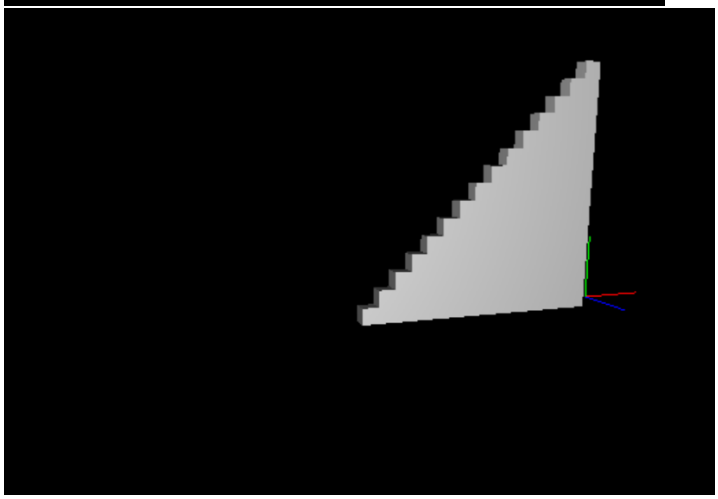
Problem 3 has similar principles to problem 2. In order to create this upside down pyramid, we want to fill out rows with the inner loop, and then move down a row to start a new row of teapots using the outer loop. Since we are starting with 6 teapots and dwindle down to 1, we initialize int i=6, and use while(i>0) for the outer loop. We decrement i every iteration of the outer loop. The inner loop is used to place the teapots in a row, and we move positon-to-position using glTranslatef() to move in the positive x direction. When finished filling the row, we use glTranslatef() again in the outer loop to move down and

start a new row by moving in the negative y direction. In order to keep the translations done in the inner loop separated from other rows, we surround the inner loops with glPushMatrix() and glPopMatrix().

4. Results

These are the output images generated :





Sham Hintolay
1618608