Omar Watad

PSID: 2036321

HW2 COSC4370

Problem 1: turn 10 teapots into the shape of a circle.

Problem 2: make a staircase out of cubes.  
problem 3: make a sort of "pyramid" out of 21 teapots.  
problem 4: do anything you like with transformations.  
save results, change main cpp, and attach screenshots.

Solutions:

Problem 1:

The teapots must all be turned at a particular angle in addition to rendering each individual teapot, a for loop as well as a mathematical functions is required to determine the right x and y values for each.

Looking at the document's accompanying image, we deduce that each of the teapots needs to be rotated, therefore we begin our code by setting the variable int Zawyi to 0 for later use. But, for the time being, we focus on the procedure of actually positioning the teapots on the screen in their correct placements. This may be accomplished by using a straightforward for loop, which iterates ten times (as there are ten teapots in the image). Now we control each of the teapots individually.   
We must initially call on glPushMatrix because we want the changes the teapots go through to reset each time the loop iterates. From here, using the function glTranslatef, we can provide the x and y positions at which each teapot should be present (x, y, z). Since z is not an axis we would be using in this scenario, we can set it to 0. We must use cos(((pi\* 2) \* i) / 10) and sin(((pi \* 2) \* i) / 10) to calculate and find the x and y coordinates since the teapots must adopt the shape of a circle. The number 10 represents the total number of teapots we will be including in the image. This will appropriately build the teapot circle but disregard the rotation that needs to be performed on each of them. As a result, we must then invoke the glRotatef(A, x, y, z) function, where A denotes the rotational angle of the object. In this case, the rotation will be done on the z axis and will not affect the x or y axes, therefore x = 0, y = 0, and z = 1. Remember how we established a Zawyi (the angle at which rotation happens) variable earlier initialized it to 0? Now that each of our teapots is correctly rotated, we can call glRotatef(angle, 0, 0, 1). With the function glutSolidTeapot, we can now call the function that will truly represent our teapot after all the transformations have been correctly implemented (0.2). In order to stay as true to the original image as possible, 0.2 here denotes the size of the teapot. As a result, we must execute glPopMatrix() to remove the matrix that we had first pushed onto the stack when our series of transformations is complete. Before completing the function, there is still one more step, which is to change the variable for the angle that each teapot is tilted at (Zawyi). Since there are 10 teapots and a circle with a 360 degree circumference, we may divide 360 by 10 to get 36, indicating that each teapot needs to be rotated an additional 36 degrees from its predecessor. As a result, we update Zawyi each time with: zawyi = zawyi + (360/10).

Problem2:

According to the document's illustration, cubes should be used in place of teapots for this particular issue. We must choose the x position to start at because we will be moving from the leftmost step to the rightmost step. We set GLfloat x = 2 as our first line of code because the image suggests that this value is close to 2. The y position appears to be around 2, but it is crucial to double this value to obtain GLfloat y = 4 because we will subsequently call the scaling method. After we have the initial x and y positions, we can create a for loop that will iterate 15 times, which corresponds to the number of steps in the original image. As we are working with cubes, changing the cube's size will also change the values of its x, y, and z axes. Nevertheless, since our image will be made up of rectangles with various side lengths, we must use the OpenGL scale function. As the x and z values won't be scaled, we ignore them. But since y is being scaled, we may set y = y, keeping in mind that the y on the right-hand side is the variable we initialized at the very start of the issue. Hence, for our following line of code, we may call glScalef(1, y, 1). Now that we have translated our x to the x value we initially initialized and our z to 0, since it shouldn't be affected by the transformation, we can call the glTranslate(x, y, z) function. Due to the scaling method used earlier, the value 0.1 for our y must be entered. The scaling function makes an effort to scale from both ends rather than just one end of an axis. As a result, instead of having a flat bottom, our staircase will have a second staircase that rises in a counterclockwise direction. By applying a value of 0.1 to our translation, the influence of the scaling function is virtually eliminated, and the staircase always appears as it should. Now that we are aware of this, we may call glTranslate (x, 0.1, 0). We may now draw the cube itself because the required transformation functions have been invoked. We select a size of 0.25 for each cube and call it glutSolidCube in a manner similar to the teapot (0.25). Then, we must call glPopMatrix(). It's equally crucial to keep in mind to modify the x and y values we've been using throughout the for loop, as each step requires a drop in the x location and an increase in the y value, which indicates how much scaling should be performed each iteration. We can refer to x = x - 0.2 and y = y - 0.2.

Problem 3:

Declaring GLfloat x1 = 0 and GLfloat y1 = 1 at the beginning of the code for this issue will help. We will need two x and y values for each of the two for loops we create. The x1 and y1 variables clearly describe the starting values the inner loop will use and whose values will be modified within the outer loop. We set the x1 and y1 coordinates to be the same as the topmost teapot as it is the one we will begin by adding to the scene first. From here, we can construct our outer loop to manage where each of the diagonal rows is placed. The loop starts with 6 because there can only be a maximum of six teapots in a diagonal, hence six diagonals must be made. We will set the inner x and y values to the outer x and y values, the outer for loop must have two lines. This is due to the fact that the initial values for the outside x and y values will be used to determine how the first teapot produced in the diagonal will look, but since we only want these values to be used as initial values without change, we must create separate variables to do so. GLfloat x = x1 and GLfloat y = y1 must be set as a result. Following this initialization, we can design the inner for loop, which will iterate through the range of j = 0 to i. The significance of the second requirement is that it reduces the number of teapots that need to be drawn for each diagonal that is formed From this point, glTranslate(x, y, 0), which takes the x and y values we initialized earlier but leaves the z unchanged.