

COSC 4370 - Homework 2

Name: Sydney Pospech PSID: 1872314

October 6, 2022

1 Problem

The assignment required the application of our knowledge of OpenGL to recreate three different images. The last portion of the assignment required an original image of our making that utilized nested `glPushMatrix()` calls and creation of a triangle.

2 Method

The only function that I edited was in the `main()` function specifically inside of the separate question functions. While recreating the images I had the pdf pulled up side by side to my code so I could continuously compare with the images provided. To recreate these images I used a combination of the following functions: `gluSolidTeapot()`, `gluSolidCube()`, `glPushMatrix()`, `glPopMatrix()`, `glTranslatef()`, `glScalef()`, `glRotatef()`.

3 Implementation

3.1 `problem1()`

For the recreation of the teapot circle, I used a lot more math than I thought I would. I started by choosing a reasonable radius; for this I chose 1.2 and assigned it to a variable titled *radius*. I then divided 360 by 10 because there were 10 teapots in the circle and there are 360 degrees in a circle. To find the coordinates of the teapots around the circle I used the equations: $x = r\sin(\theta)$ and $y = r\cos(\theta)$. The thing that I found about the stock sine and cosine is that it uses radians and not degrees so I had to convert degrees to radians. I then had to rotate each teapot based on their location in the circle. To do this I used `glRotatef((90-(i*36)), 0, 0, 1)` where *i* is the teapot number; in this case 10. I started at the top of the circle with 90 degrees and from then I went clockwise rotating each teapot.

3.2 problem2()

For this I just edited the x and y values in `glTranslatef()` and then the y value in `glScalef()` like the following:

```
glPushMatrix(); //original line down the origin
glTranslatef(0,0.3,0);
glScalef(0.15,0.75,0);
glutSolidCube(1.25);
glPopMatrix();
```

Going to the left I increased the y value in `glScalef()` by 0.05, decreased the x value in `glTranslatef()` by .18 and increased the y value of `glTranslatef()` by 0.03. I did the opposite for the right hand side. While this is not the more efficient method, I still was able to generate the image as identically as possible.

3.3 problem3()

For this image I created a for loop per row in the pyramid of teapots. I started with the row of 4 teapots because it was on the origin line.

```
for(double i=-0.5; i<0.4;){ // row of 4
    glPushMatrix();
    glTranslatef(i,0,0);
    glutSolidTeapot(teapotSize);
    glPopMatrix();
    i += 0.3;
}
```

From here I would change the value of i so that the teapots would start closer in or farther out to create a nearly perfect pyramid.

3.4 problem4()

For my original picture I created a big 1x1x1 equilateral triangle going through the origin and then created three triangles coming out of every vertex of the big triangle. Then I place teapots on every vertex in sight and then rotated them. The code below displays how I created the original triangle.

```
glBegin(GL_TRIANGLES); //original triangle
glColor3f(1,0,0);
glVertex3f(-1,0,0); //left vertex

glColor3f(0,1,0);
glVertex3f(1,0,0); //right vertex

glColor3f(0,0,1);
glVertex3f(0,1,0); // upper vertex
glEnd();
```

4 Results



