Franklin Fasano

2/16/16

CS 4300

Amit Shesh

CS 4300: What to Expect

Chapter 1

Assignment 2 saw us create a 3D scene with a table with at least 10 objects on it, one being an object we created. A user of the program can use the mouse as a trackball, as in dragging it will rotate the scene. Up to down rotates the scene counter-clockwise around the +X axis, left to right counter-clockwise around the +Y axis, and vice versa for each. Dragging in any other direction will compound the rotations in each direction, and rotate in each axis.

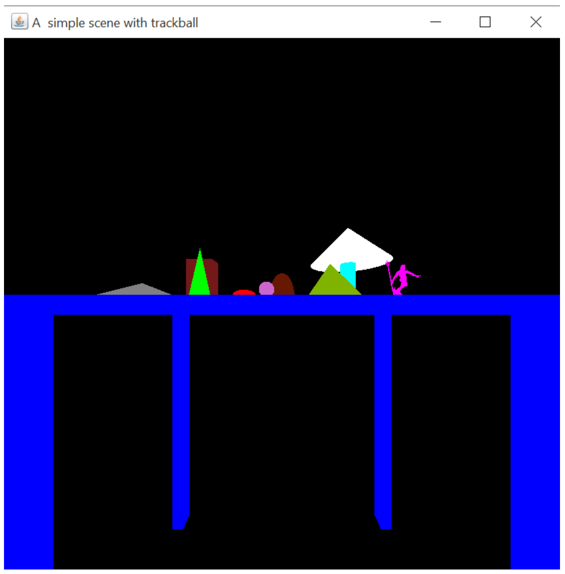

Fig 1 The scene on startup

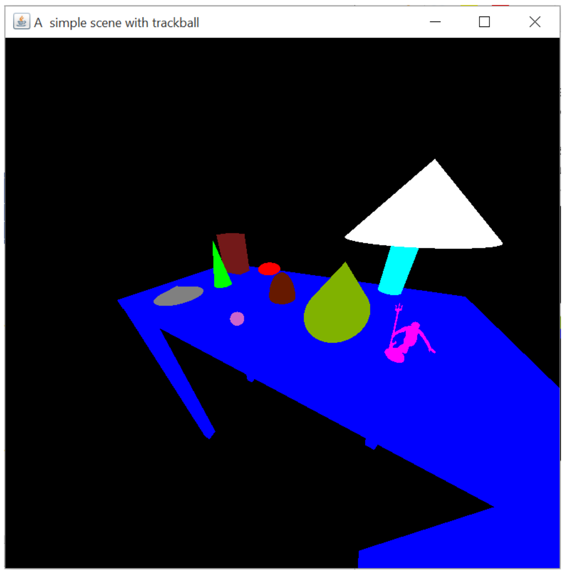

Fig 2 The scene after using the mouse to rotate in the X and Y

The scene is set up and filled with objects within the initObjects() method of the View class. It is then drawn using the draw() method, which also controls where the initial camera is, and what type of rendering it does (in this case, I set it to FILL instead of LINE, which would show wireframe renderings). The initial camera position uses the variable CAMERA_DISTANCE to set how far from the scene the camera is, and this value can be changed in the constructor for View.

I implemented the trackball so that when the user clicks the mouse, it saves the initial X and Y of the mouse position. As the user drags the mouse across the screen, it rotates the camera by an angle calculated by subtracting the mouse's new X and Y from the initial, converting that to radians, and then dividing that by 10 to slow down the trackball speed. It then resets the initial X and Y to the current, so that the trackball does not accelerate the more the mouse moves. These rotation angles are applied to the trackballTransform matrix, which is multiplied onto the rotation of the scene.

The object we created was an object of revolution. This means that it was an object that can be reproduced by drawing a curve, and rotating it 360 degrees to form a 3D object. Any object in the real world that has an outline that could be represented by a curve, which is then rotated all the way around, is an object of revolution. I created mine in the ObjectCreator class by choosing a number of stacks and slices, for the height of the object and the amount of “slices” or chunks of rotation that it is split into, and using a nested for loop. Within these, I used the cosine and sine functions to produce a vertex that followed a pattern, and repeated this for each chunk of rotation. It then progressed down the object, repeating the process to create a curve on the object that is repeated in 360 degrees. These vertices were then added to a mesh, along with triangles for each part. Here is the pseudocode. It creates the top half of an egg.

```
ArrayList verts;

Stacks = 10, slices = 16;

Vector v;

For each stack:

        For each slice:

                Radius = cos(pi * current stack/(total stacks – 1) – pi/2);

                z = sin(pi * current stack/(total stacks – 1) – pi/2)/5;

                x = (cos(pi * current slice/(total slices - 1))*r)/5;

                y = (sin(pi * current slice/(total slices - 1))*r *2) /5;

                v = new vector(x,y,z);

                verts.add(v);
```