CG200 OpenGL Assignment Report

Jason Giancono
16065985

Contents

1 Introduction

For my assignment, I have decided to model a cafe, as Curtin University has a great range of cafes on campus.

2 Models

I have five distinct models in my scene. They are as follows

1. A chair, which is constructed out of rectangular prisms. In order to make the prisms to my liking I created a function which draws prisms from and xyz coordiate and width/height/depth values. It creates the prism by drawing the six faces as quads. The function also maps texture points and normals.

2. A table, which consists of a cylinder with a smaller top diameter than the bottom (like a cone without a point), another cylander ontop of that and two circles which enclose the top cylander. I used the glut functions gluCylinder and gluDisk to achieve this.

3. A Fork. The fork was made in an external program called SketchUp. I then exported it to an SDL file and then turned that SDL file into raw coordinates which I used as a triangular mesh. I used SketchUp because I wanted to try using a triangular mesh object in OpenGL. Although it produced an OK result, it was pretty much impossible to later on apply a texture to the object because of all the points that need to be mapped, so it would really have to be done in the program. It was also pretty tedious turning it into array form to import into my program, if I had had more time I would have written a function to do it for me. It is also pretty hard to implement LOD for the triangular mesh because you can't just change the number of points like the other functions, you would need to export an entirely different model.

4. A Cup. The cup consists of two cylinders and two disks, one which has a hole in the middle which covers the top. I used the same functions for this as the table.

5. A Teapot, to draw the teapot I used the function glutSolidTeapot.

To make my scene look more populated, I put more than one of each object in the scene. I utilised rotations in order to make placing them easier, as I could just rotate the chairs/cups around the table.

3 Light Sources

I have a point light and a directional light which are both white in my scene. I also have a greenish spotlight shining on the teapot/table from above. The entire scene also has a small amount of ambient light.

4 Finishings

I have used five different finishings for the five objects in my scene which affects the way the light bounces off them.

1. Chair: Has the default finish, which no specular, shine or emission
2. Table: Is a bit shiny.
3. Fork: is shiny and also has some specular.
4. Cup: is shiny and has small amounts of emission and specular.
5. Teapot: Has generous amount of specular, emission and specular.

5 Keys

I have used the glut input callback to implement the interaction with keys. I have implemented all the keys listed in the assignment. I have also added a 'q' key which changes the fog type. In order to make zooming simulate zooming a camera in real life, I had a press of the key change the zoom value by zoom + 0.1*zoom. I decided to implement zooming by changing the projection matrix instead of scaling/translating the whole scene, as scaling the whole scene ended up with clipping issues and also changed the perspective a bit, while changing the projection matrix simulates zooming in real life better. I capped the zooming out at a certain value as manipulating the projection matrix too much results in a warped scene.

6 Level of Detail

I decided to have 4 different levels of detail. My levels of detail only effect the round objects, which are the table and the cups. I implemented this by having a constant value which was multiplied by the level of detail. That value was then input as the number of points or 'spines' (as the function calls them) in the gluCylinder and gluDisk functions. 1 is the least detail (only happens when zoomed out far) and 4 is the most detail. The LOD is calculated with some if statements whenever the zoom value changes

7 Key Display

In order to display the keys on the screen, I had to go to the end of the function after all the 3D stuff was done and project the bitmap fonts onto the orthographic matrix. I wrote a little function called bitmap_output to take a string and font and output it. The display also handily shows you the value of the speed, fog and the current level of detail in real time.

8 Animation

My animation is called Spin the Teapot. When you start the animation, the teapot begins to spin and gradually speeds up (set the speed to 1 to see this). It then chooses one of four cups randomly and slows to a stop pointing at the cup. It then moves upward and toward the cup while tilting as if it were pouring and then it is set back down on the table. I achieved all this with rotation and translation of the teapot. I used the rand function to decide on which teapot to stop at and then waited until the teapot was rotated in the right place to start slowing down to start slowing it down. This was done so the teapot didn't just jump about. As this creates uncertainty as you don't know what the clock's value will be when the teapot stops, I had to start a new clock once it had stopped.

8.1 Pausing

In order to pause the animation, I just made it so I can just change ani_play to 0 and everything stops. When it goes back to 1 when you press resume, it continues from where it was.

8.2 Speed

I had an individual speed variable which you and increment up and down. In order to make this speed up the animation, I had the entire teapot section inside a for loop which runs for 'speed' amount of times. This means if speed is 2, it will skip every second frame etc etc. This means when you increase it too much, the animation looks funny (especially the spinning).

9 Texture

I used three bitmap textures, one on 2 chairs which was a wood texture, one which was on the other two which was a blue fabric texture and one on the teapot which was a kind of ceramic looking one. I loaded the texture using libbmpread at https://github.com/chazomaticus/libbmpread which I found to be a better bmp loader than the one on blackboard. The other things were coloured with a function called setColor which I adapted from the openGL example file by Mark J. Kilgard called scube.c. I also used the drawCheck function from this example to draw the chequered floor.

10 Fog

I decided to add fog into my scene (the scene is set before the Curtin smoking ban). You can switch the fog type by pressing q, but LINEAR is the best. The fog also gets worse as you zoom out, which accurately simulates real life.

11 Transparency

In order to provide a good view of the scene while also making the text displayed easy to read I put a transparent box under the text. This was done using the GL_BLEND feature and setting the alpha levels in the rectangle to blend the colour with what was already drawn.

12 Shader Mode

I implemented changing the shade mode from flat shading to smooth shading by having a global variable which changes when you press the P/p key.

13 X/Y Rotation

Learning from the example sample200.c file I implemented rotation about the x and y axis when you press their respective keys. I did this by incrementing a global variable each render and rotating the base matrix by that number.