# Log Book

Student Name: Jekyoung An

Student ID: 20125059

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Time Spent | Date | To do | Work Done | Detail | Bugs |
| 10 mins | 6/05 |  | Created a template for project | Followed the template from the lecturer.  Downloaded the “animationController-lights.c” file from the lectures and followed the template into my own project.  Uploaded the template to GitHub for future changes to be saved. |  |
| 1 hour | 13/05 | * Rotation camera movement * Draw a drone | * 3 axis camera movement | Used the snowman controls from the lectures to make a basic camera movement. Can only move in 3 axis without rotation.      The problem with this template that was provided from the lecture slide is that other fundamental functionality for a basic camera movement is not provided like rotation and look up and down.  Rotation will be the next task to work on. |  |
| 1 hour | 14/05 | * Rotation camera * Move drone forward in its facing direction * Attach camera to drone | * Draw a drone | Drawn a basic shape of the drone. I used hierarchical modelling to draw the drone.    I encountered a problem where a lot of the elements are used multiples of times. I made a helper function for this like the drawArm function with takes in a parameter for which side it should draw on the drone, so it is easier to draw.  Another problem was when rotating the drone, it seems like the propeller stops spinning. I think this is a visual problem because the propeller spins at the same rate as the drone. Once the camera is attached to the drone, I think this bug will be fixed but not sure at this stage.  Also, the drone can rotate but I am struggling to calculate the forward movement. I am looking through the lectures to see how to calculate for this. | * Propeller stops spinning when rotating the drone |
| 2 hours | 15/05 | * Fix drone shape * Ground implementation | * Rotation camera * Move drone forward in its facing direction * Attach camera to drone | I have solved the problem to calculating the movement for its facing direction using trig. Calculating sin and cos for the direction will calculate the forward movement of the drone.  Using yaw, the drone can track the angle that it is facing.  The camera is attached by setting it’s position in init and lookat function. I wanted the camera to be placed a little bit higher than the drone so you can see the world clearly.    Also the problem with the propeller is solved. My assumption was correct and the propellar was just rotating the same speed as the drone so it looked like the propeller froze but since the camera is following the drone the visual bug is fixed.  I thinking I need to work on the ground so it spans out like a world should. |  |
| 30 mins | 16/05 | * Ground Implementation | * Changed drone shape | I made the drone arms thinner, so it looks better. |  |
| 30 mins | 16/05 | * Texture mapping | * Ground Implemented | A for loop is used for the flat ground. I used a 250x250 dimension ground with each block of ground to be 10x10. A normal facing up is calculated onto each corner of the 10x10 ground so light can be reflected of the ground.  I had problems with the loop. I didn’t know what the values should be and I wanted the origin (0, 0) to the the middle. Now the middle of the scene is the (GROUND\_WIDTH / 2 , GROUND\_LENGTH / 2). I could have started from the negative axis to make (0,0) the middle but I didn’t want to do negative calculation. |  |
| 2 hours | 04/06 | * Make lighthouse * File separation | * Texture map * Ground collision * Reshaped the drone | I thought the drone should have shorter arms, so I changed the appearance once again.  I’ve added textures using the PPM file format. I converted a jpg file into PPM with Irfanview. Using the template from the lectures I loaded the texture onto the ground so the ground now has the asphalt texture I got from the internet.  Also I added so the drone cannot go below the ground.    This code will restrict the movement of going below 0 in the y axis.  <http://www.texturise.club/2013/12/seamless-tarmac-asphalt-texture-maps.html>  There is around 1500 line of code, and it is becoming a problem to navigate around the file. I think for the next task I should start separating the file |  |
| 30 mins | 05/06 | * Sky box | * File separation | I have separated the logics into separate files so that it’s easier to navigate and debug parts. This will save a lot of time since I don’t have to scroll around a 1.5k lines of code to find one thing and added more code and files will become easier in the future.  The problem I encountered was I didn’t copy and paste properly sometimes which led to bugs and it was hard to trace back since I cannot control z back. For example. I moved the logic for camera, but I forgot to import the math library. So, the camera init did not work but the scene still loaded without errors. Once I imported the math.h the camera initialized properly.  I disabled the ground texture while working on the scene because texturing the ground takes awhile to load. |  |
| 30 mins | 05/06 | * Make Lighthouse | * Sky Box | I have added the sky by texturing the sphere with the sky texture.    The sky moves together with the drone in the x and z axis. But for the y direction the sky box does not move. This is so it give the effect that the drone actually looks like it moves up into the sky.  I increased the cameraZFar plane for the camera to 300 and used this calculation for the radius of the sphere. I increased the cameraZFar because the sky looked too closely which distracts the emersion.  Initially, I was wondering if I should use a cylinder or a sphere because a sphere will warp the texture worse than the cylinder. I realized the earth is a sphere, so I decided to use a sphere instead. |  |
| 1 hour | 05/06 | * OBJ loading | * Made lighthouse | I made the lighthouse with cylinders and spheres. The normals were automatically calculated by the quadratics so the lighting should be sorted out. I made a stand with the light source in the middle. I hacked the light by making a cylinder then temporarily turning on the alpha channel so I can decrease the transparency. Fidgeting with the alpha channels made the lighting weird if the drone enters the light source but I don’t think there is other way to implement this without external libraries.  The light source spins around and randomly change direction at random intervals.    This function is call each time the direction changes so the lighthouse knows when to change direction next time.  I think I should start loading in buildings with OBJ loading and storing them into a display list. I would also like to have some trees. |  |
| 1 hour | 07/06 | * Making the scene * Add fog | * OBJ loading | Loaded the OBJ and tested if the objects loaded onto the scene.    The building and the tree load properly onto the scene. There was a problem when getting the OBJ on the internet. Some OBJ did not have the models centred into the middle. I had to open the OBJ file into blender and manually centre object. Also, each models had different scaling compared to the other. I had to either scale them in blender or when saving into the display list, so all objects fit into the scene. |  |
| 10 mins | 07/06 |  | * Add fog | Added the fog from the template code from the lecture slide. I adjusted the value to be much lower since my scene has a long render distance so minimal fog was required for the sky to be seen. |  |
| 1 hour | 07/06 |  | * Material used | I implemented materials instead of colors so that the lights can be reflected.  That is the example code for the ground material. In future objects materials will be used instead of colors so the lighting can work properly.  There was a problem when I forgot to disable color mode so I can use materials it took a while but I got my friend to check my code and he spotted the problem.    I uncommented this code so now the materials work properly. |  |
| 3 hours |  |  | * Drawing the full scene | I used a separate file called city.c for the city scene to be drawn.    I made a forest at the back of the building. The trees are in a display list, and I used an array of int at init. This is so a random set of trees can be spawned and drawn.  The building is drawn manually within the city by translating each building. |  |
| 1 hour |  |  | * Drawing the scene | I made parts of the ground load the water texture, so it looks like an ocean. I placed the lighthouse in the middle of the ocean and applied the brick texture to the ocean.  I added the road texture just like the ground. Making a square then applying the texture onto it. |  |

# Final Reflective Statement

I didn’t manage my time for this assignment properly. I was too focused on other assignments and work. I started mostly working on this assignment 2 days before the due date which is not enough time to make a good scene for this assignment. This made me rush this project and hand in the assignment late. If I had managed my time properly and started working, I think I would have made a better scene that I would be proud of.

Other than that, this course taught me a lot about computer graphics and OpenGL from with minimal library. I previously learnt modern OpenGL course from Udemy to get me prepared for this paper. But that course used external libraries. This course made me learn how it’s like to code from scratch and see how these modern libraries were developed.

This paper also taught me C from scratch. I rarely coded in C or C++ and preferred not to. This course taught me basic of memory management and procedural coding. I often code in JavaScript or Python which deals with a lot of these problems automatically.

I had a lot of struggles implementing ideas into code. There was a lot of trial and error when writing code then running it. I would often guess the values then wait for the scene to load which wasted a lot of time. I should have planned out earlier and have more detailed planning, so I don’t have to guess.

Good skills to take out of this course is I learned a lot about OpenGL. Although I didn’t make my most proud scene from this, I would still like to learn Computer Graphics and will still be working on OpenGL during the holidays. I would like to learn SDL2 and push my OpenGL skill further and make a product that I will be proud of. I am ashamed from this paper but I would still like to make something better with OpenGL so I can show it to my employer later in the future.