CITS3003 Project

Cameron Armstrong

21194619

# Introduction

Our task was to add the functionality requested in the project pdf into the skeleton program provided to us and to add additional functionality of our own. I chose to work individually in this project. I used MinGW on Microsoft Windows 8 to compile and I used Notepad++ to edit the source files.

# How well the program works

I managed to complete all of the functions except for part E which was the task in which we had to modify the program so that it scales the scene appropriately when the window width is adjusted. The rest of the functions, including my own custom one, appear to work as expected.

# Building the program

This program took many hours to complete, with many of those hours being trying to fix errors in my code that caused the function to either not perform, or to perform incorrectly.

Part A was to add full control over the camera. This was an easy task which just involved multiplying the view matrix by the rotation variables.

Part B was to have the program able to rotate objects. This was done by multiplying the model matrix with its rotation matrices calculated from the angles.

Part C was to add a menu item to adjust the ambient, diffuse and specular light. This was done by adding the required menu item. The code for the actual function had already been implemented.

Part D was to make objects not clip so far away from the camera. I simply did this by reducing the nearDist variable to 0.001 which seemed to help a little, although I would have liked to be able to further improve this function.

Part E was to modify the reshape function to make the scene scale appropriately when the window is horizontally resized. I could not figure this one out. I experimented with the projection matrix by dividing my width and/or height in the various arguments which produced some results which indicated I was on the right track, but I could not figure out the exact way to do it.

Part F was to modify the vertex shader to make objects get brighter when the object is closer to the light source. I did this by first calculating the distance to light by using the length function on the light vector and then dividing the light by the distance.

Part G was to move these calculations into the fragment shader. This task took me a long time. The main reason it did is because the floor would act odd and not reflect the light properly at all. I managed to fix this issue in the end by flipping the floor 180 degrees so that the normal is in the correct direction.

Part H was to change specular light so that it shifts towards the colour white. This was a simple solution and just involved me multiplying the specular value by the colour white (vec3(1.0,1.0,1.0)) instead of the rgb value.

Part I was to add a second light into the scene which was directional. This was done by calculating the vector from the light to the origin and not using any specular light so that it looked like the sample solution.

# My custom functionality

Part J was to add custom functionality to the program. I added a function to the menu called LSD which makes the scene go all trippy. This was quite an involved task. It first involved me adding a new Custom functionality menu and LSD in the submenu. I needed a timeParam variable because I wished for there to be changes in the scene over time. I attempted to do this the same way as in the labs, however I either got errors or nothing happened. I am not sure what went wrong there. I just added my own timer instead which is just a floating point variable that increments by 1.0 after each call of the display function. This is passed onto the shaders. The fragment shader takes this variable and adjusts the red, green and blue variables by multiplying them by the sine of 0.01 x the timer. The red, green and blue variables are given different rates of change so that you get many different colours as time passes. The scale of the object was done in a similar way in the main C++ code, but I added an offset determined by the meshId variable so that different objects would change size at a different time from the others. I added the glClearColor function into the display function and this is also affected by the sine of the time so the background shifts from black to white. Finally I added code so that you are able to disable LSD mode by selecting it again and everything returns back to sanity.

# A reflection on my experience with this project

This task was difficult and stressful when I could not figure things out, however it was fun and rewarding once I began to complete the tasks. I chose to work alone and I do not regret this. I enjoyed being able to perform on my own without the pressure of needing to perform well to please your partner. I found the help forum to be useful and the lectures to be helpful also. I also used the recommended textbook a little. When I got really stuck I used external websites and forums to find out how things work in OpenGL and I did this without copying and pasting any external code into my project. I think we were given the perfect amount of time to complete this project.