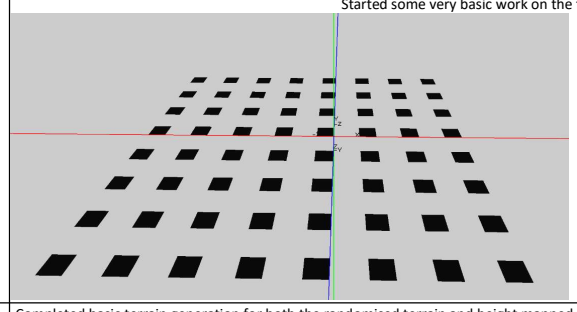
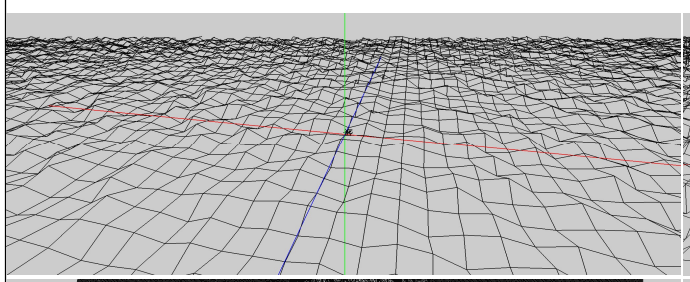
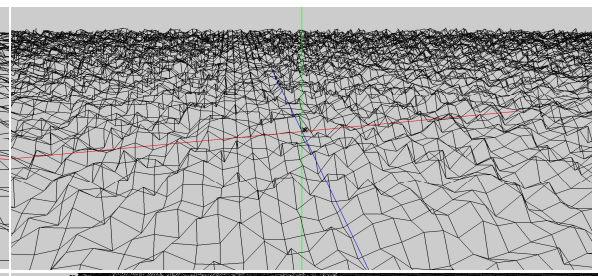
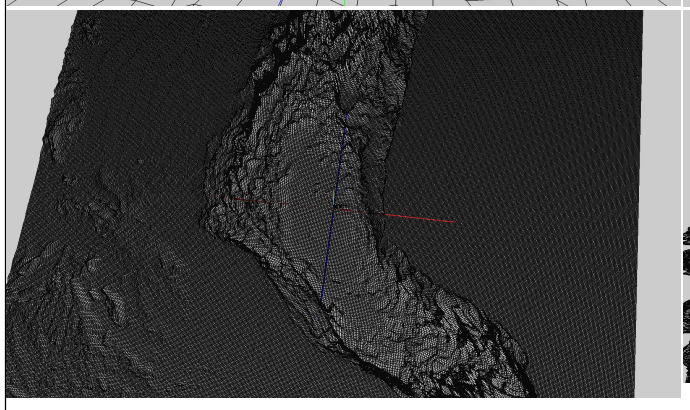
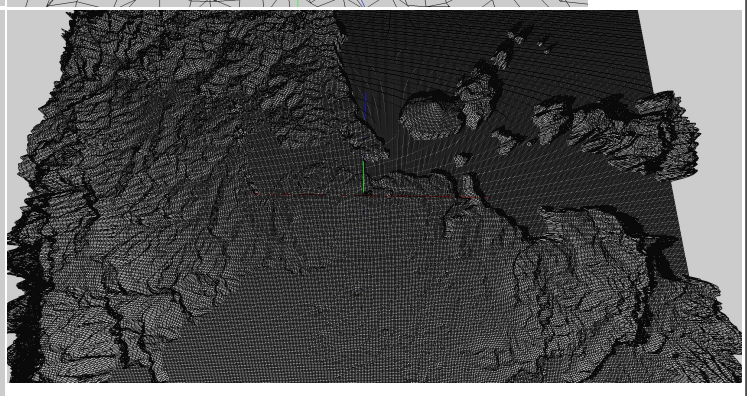
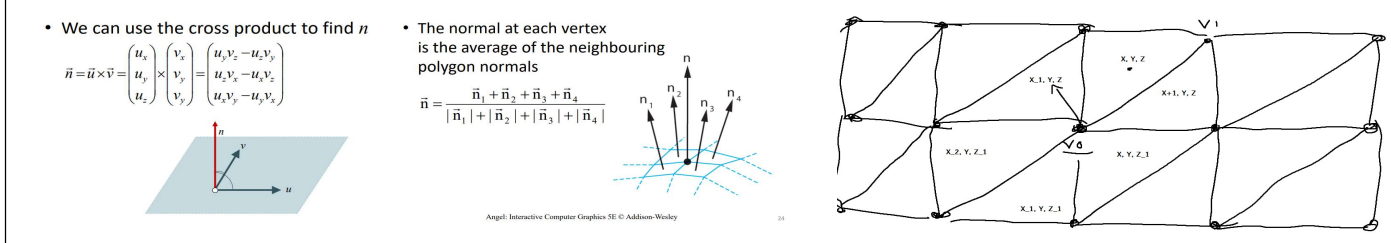
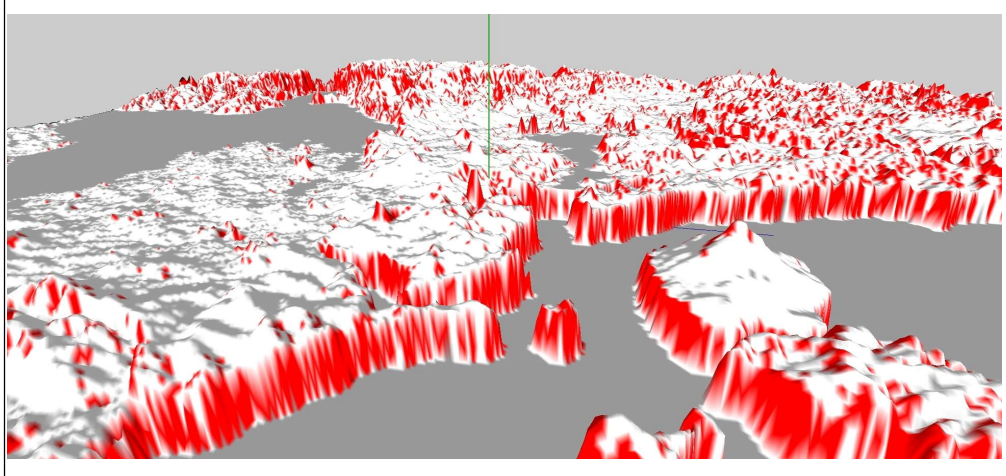
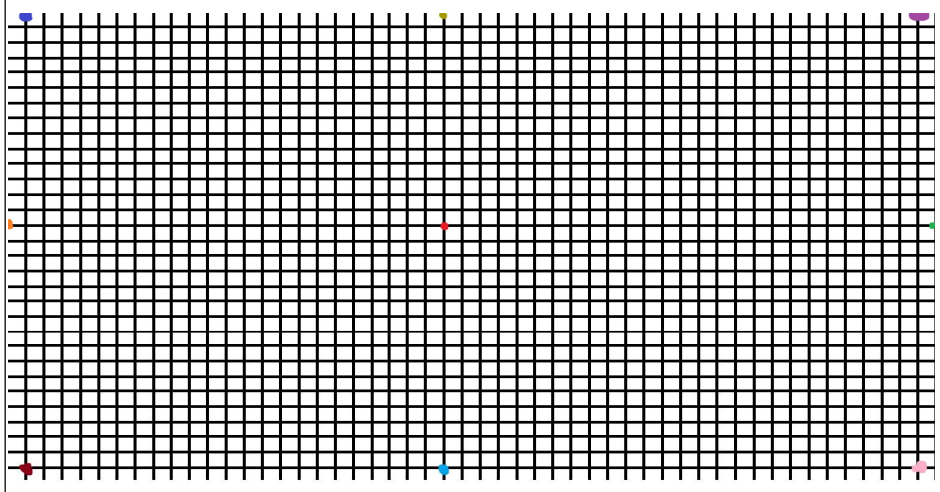
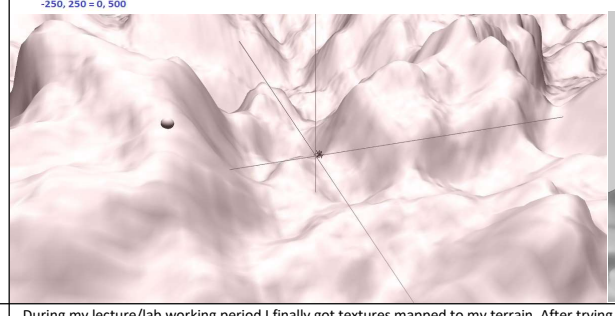
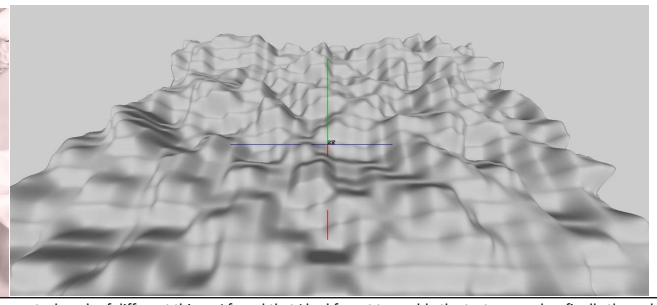
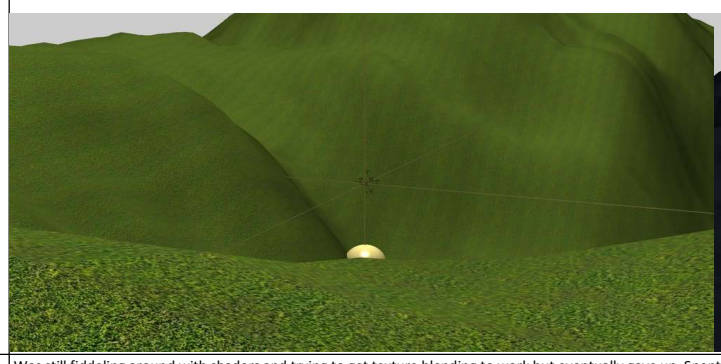
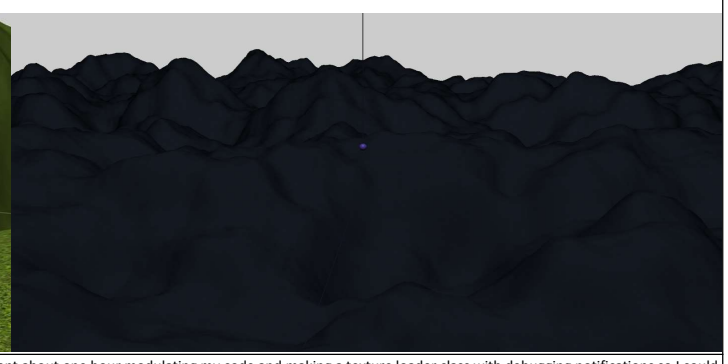
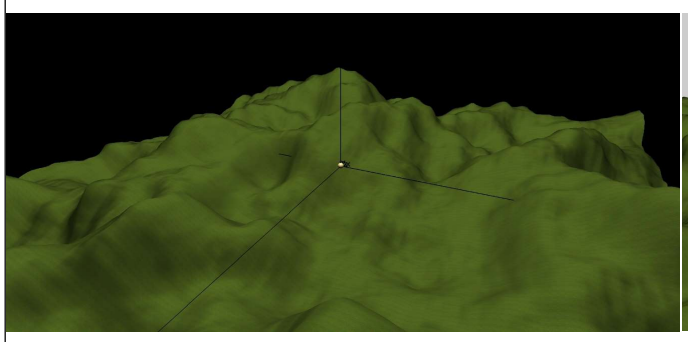
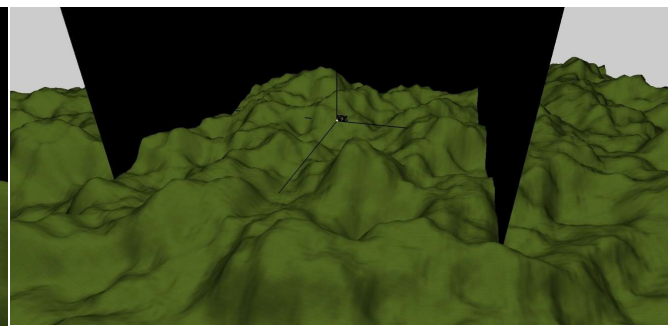
Personal Work Log

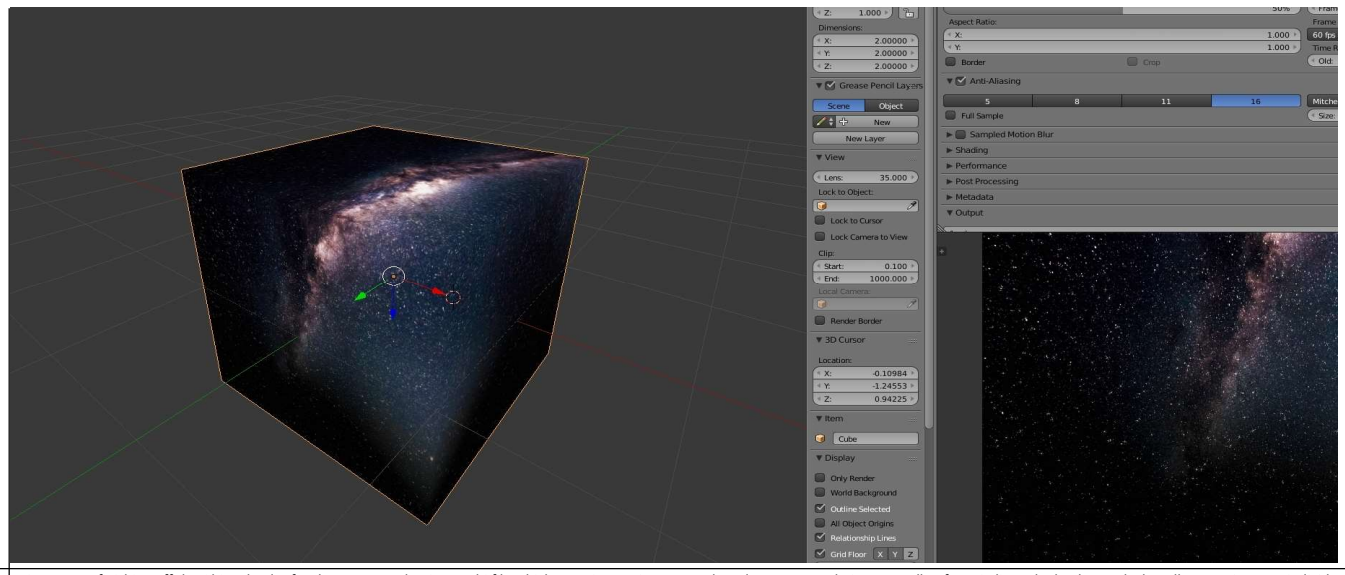
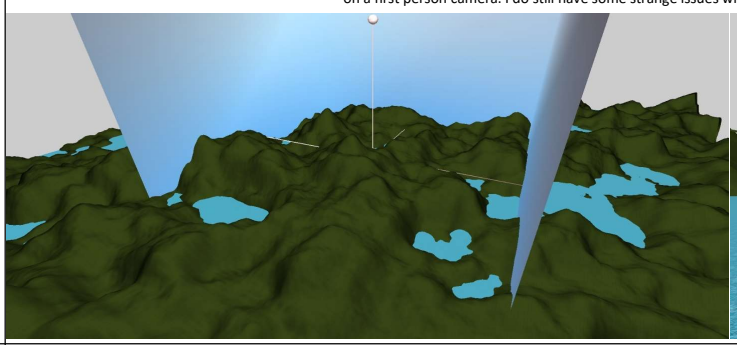
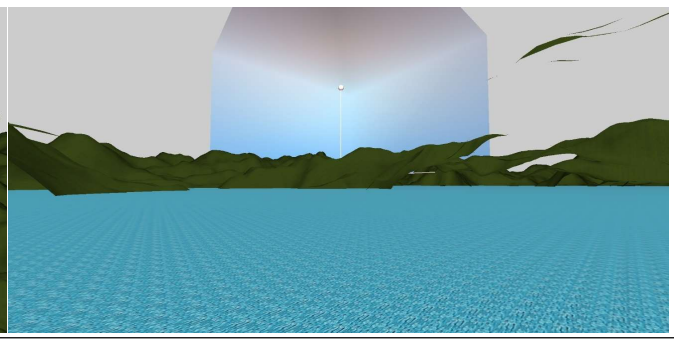
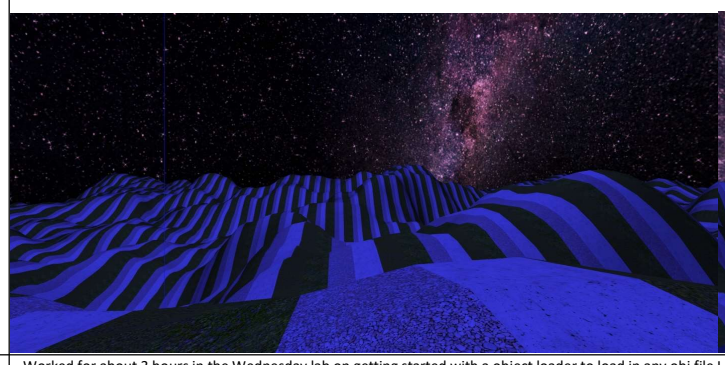
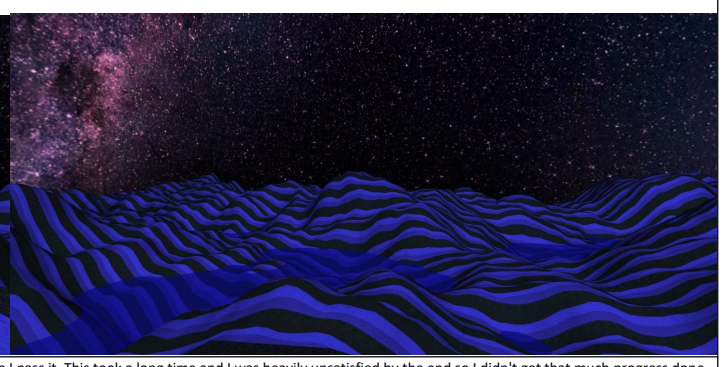
Nikkolas Diehl - 16945724

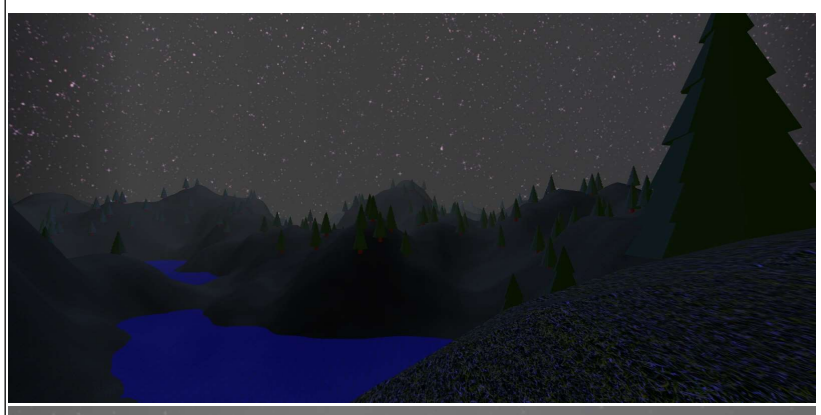
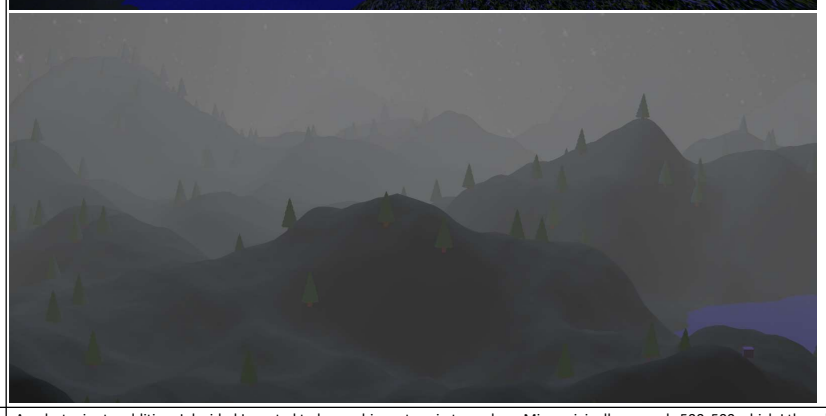
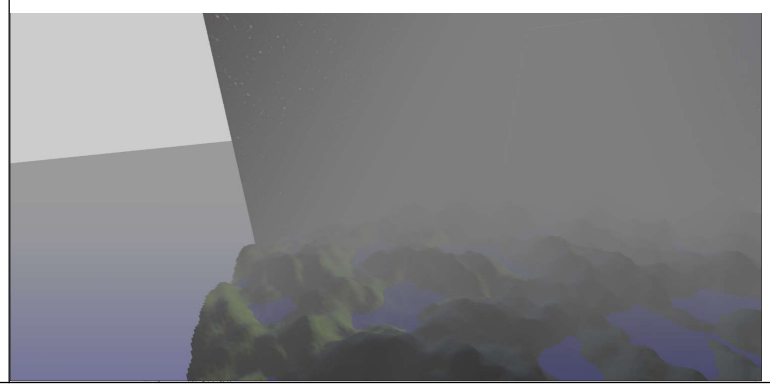
Total Hours Spent on the Computer Graphics Assignement 1: 68.05

Number of Logs: 16

Date	Hours	Difficulty with task (Easy / Moderate / Hard)	What did I learn	Personal Reflection Of Events During This Time
1/03/2019	1		Texture mapping and started some ideas about bitmap bump maping	Started the project and got basic files working. Started working on the terrain but it wasn't working
3/05/2019	0.34		—	Started some very basic work on the terrain. Got it to display a little but but didn't have much time to work on it today. 
5/05/2019	5.21		—	Completed basic terrain generation for both the randomised terrain and height mapped terrain and tested both. The random generation I found quite hard to get to work since I was working from a 1D array of height maps which I had a lot of trouble with. Eventually though I did get it to work and display a proper terrain with random height. When I was working on the height map it took me significantly less time to program since it was already written out on the lecture notes and because I had already worked for ages on the code so I understood it quite easily. I also wasted time with the random generation cause I was "translating" the terrain using smart math when I could have just generated from 0,0 and then translated it across to be in the center. Bellow are some examples of my random generation followed by my height mapped terrain    
8/05/2019	7.5		—	Worked on fixing up my normalisation function to find the normals of all triangles in the terrain. The normals are now properly calculated by finding the average of all sruface normals around a vertex. I also redid my colouring function so that it colours based off the angle of the hills. Anything more than 45 degrees means red, otherwise white. This is a proof of concept to make sides of hills a different texture.  

12/05/2019	9		—	Worked for about 6 hours on getting a perlin noise algorithmic terrain working. The algorithm goes through 3 de-noising functions that smooth out the noise. The first function called getNoise sets a seed with relation to x, z and a random value between 0-1,000,000,000 that is previously set. Once the seed is created, it then returns a random float between -1f and 1f. This randomised value would look very extreme for the terrain to use so the random value goes through the getSmoothNoise function. This takes the corner heights, side heights and centre height of a given vertex at z, x and returns the average of all those values. These randomised heights would still be a bit too extreme and have no interpolation in between them so the height value then goes through a final function called getInterpolatedNoise. This simply interpolates height values between each value it is given to make the terrain very smooth. Finally the noise is then mixed with another randomised interpolated noise value with a lower amplitude to create a very natural looking smoothed noise value. All in all the terrain looks very smooth and runs well. I then spent a while making a test character/player that could move with keyboard controls and that changed height with the floor it was standing on. This sortof worked but I had some bugs where the player couldn't strafe properly and also the character couldn't smoothly move down hills as shown here: https://i.gyazo.com/5c0361bec7625ce14b3dd082a5d5b078.mp4. Show bellow is some images of my perlin noise terrain. I also turned off colouring and started fiddeling around with lighting and placed two lights. One for the sun and one for the moon   
13/05/2019	3		I learnt some basic stuff about texture mapping	During my lecture/lab working period I finally got textures mapped to my terrain. After trying out a bunch of different things, I found that I had forgot to enable the textures and so finally they where mapping correctly to my terrain. I got grass working in higher definition than I thought I would be able to since I thought it would lag and I added pebbles, snow and groundcover images for later when I planned to try bind different textures based on the height of the terrain. I also finally got the movement of the character working and it turns out the issue was that I was using the heading angle value in degrees when the formula is meant to only use radians.
13/05/2019	8		I learnt a lot about shaders and shader programs	After looking around for ages and ages to try find a way to easily map different terrain based on my height map at a given z, x value, I found that most people suggested to make a shader. After searching around I found a tutorial made in OpenGL 3.0 (can be found here: https://youtu.be/AyNZG_mqGVE and here: https://youtu.be/4w7lNF8dnYw). The code was not that hard to convert back to OpenGL 2.0 and after I did that I searched around on blackboard for easier ways to actually run the shaders. Once I found some spare code that the teacher had given us I actually found the algorithm for reading in shaders and running them from the tutorial ran and worked much much better and was easier to read so I ended up sticking with that. Once I then got some basic lighting working and finally got textures back up and running I was extremely pleased. I had a fully functioning shader that currently was able to take in a texture and display it on a specific object without changing any of my original code but simply by adding and playing around with shaders. Once I had the shader running, I finally added my sun light, moon light and added materials to everything and correct material to each of the lights to create a very nice looking day time and a very spooky night time as shown here: I also added a golden material to the 'character' sphere for now so that I could more easily see it for programming later. The next thing I want to do is add the skybox and get a day/night time working and then after that, on Wednesday I want to ask the teacher about using my shader to run different textures on my terrain like I originally intended  
14/05/2019	3		—	Was still fiddeling around with shaders and trying to get texture blending to work but eventually gave up. Spent about one hour modulating my code and making a texture loader class with debugging notifications so I could more easily see how long my program was taking to buffer. Added sky textures and started work on the skybox using verticy arrays but didn't know how to properly bind a texture so gave up for now.  
			Learnt the basics from small video tutorials on blender. I made 3 skyboxes with blender for night, day and clouds	I spent about 2 hours working on the sky boxes as I decided to do them inside blender which allowed for much more creativity and faster building. I learnt heaps about modeling in blender and texturing with UV maps in blender. I made 3 skyboxes that I will be using. Clouds, night and day which will have transitions. I also made sure to invert the normals so they all point inwards.

15/05/2019	2			
20/05/2019	4		—	Spent ages finishing off the Object loader for the game. It takes in an .obj file which contains vertices, normals and texture translations as well as faces in lists. The loader reads that all, parses it out as individual, properly formatted float arrays and then passes them into a RawModel object that saves them and then draws them on a draw call. Once that was all done I quickly got a water level working with a quick texture and started working on a first person camera. I do still have some strange issues with texturing on my skybox which I will try fix tomorrow  
20/05/2019	7		—	Finished off and worked on a heap of different things: - The skybox textures where fixed. I ended up going in with photoshop and changing the texture types to a standard unwrapped cube mapped texture. Before the texture was in a weird standard that required me to break appart the UV map in blender and move it around. - Skybox is now fully functional and other objects are easily able to load into the world with the new and improved RawModel class and ObjLoader - Textures redone to fit the new skyboxes and the cloud skybox texture was removed of it's blackness to add a 3D ness to the scene. It didn't end up looking so good cause the clouds had very obvious seems so I removed them temporarly and thought about adding physically rendered clouds later - Tried again with multi-texturing after someone said my idea of all the textures within one could work but failed. Left it on the stripy texture for now for debugging - Finished off the first person camera. The camera now moves with the character and looks ahead of the character correctly. The character and camera both rotate correctly, move correctly based on a heading and strafe correctly. I then converted the current code I had (which was using Q and E to rotate) to use the mouse movement with a mouse motion listener in the player class. The mouse motion listener checks if the mouse is in the 'move' zone near the edge of either side and top and bottom of the frame. If it is within a specific 'move' zone it will rotate or pitch accordingly The camera also correctly follows these rules - Finished off the night time and day time simulation by correctly matching up the materials for the sun and moon. The day time has already been shown, but finally I have a very nice night time (attached bellow) - I also fixed up water a bit so it was rendering in the correct order and was a bit more blue rather than incredably gray in the original version I had. - Tomorrow will revert the textures to grass only until I figure out multi texturing and then start adding fog as that is a crucial requirement of the assignement. I plan to add a blue fog if youre underwater, otherwise there will be a far away fog. I also plan to add a proper height checking algorithm so the movement isn't so janky and I can then add a proper velocity and jumping mechanic. Also one really nice thing to note is after fixing up the file management and fixing the textures so they're a proper cube map, the game runs alot smoother and starts up significantly faster.  
22/05/2019	3		—	Worked for about 3 hours in the Wednesday lab on getting started with a object loader to load in any obj file I pass it. This took a long time and I was heavily unsatisfied by the end so I didn't get that much progress done unfortunately. I kept having issues where the scanner wouldn't scan in the correct things and it would end up with null pointer exceptions over and over. I decided to make it more modular later on so that each and every array size was dynamically set.
27/05/2019	2		Learnt more about fog and found it was actually insanely easy to set up	I talked to the teacher a bit in the Monday lecture about fog and found it was actually very easy to set up. To keep it optimised I set all the static variables and data like start and end length of the fog in the initiation function and only changed which fog was being displayed based on height of the player. I ended up very quickly and very easily getting fog working. Originally though the fog was insanely thick as shown here. I decided to fix this later as the class time had ended and we had to leave but originally the fog density was at 0.1 which I had been told is wayyy too thick. Most people seemed to be using around 0.005 for their fog.
29/05/2019	6		Learnt more about object loading and material loading but found it extremely annoying to work with	I worked for about 2 hours in the class, and then went over to a lab and worked for a further 4 hours before going home on getting my Rawmodel class and object loader class working. The biggest problem I had was that each obj file I had was formatted slightly different like it had been made in multiple different versions of blender. Because of this I ended up having to dump all of my original tree models I planned to use and instead just use the pine tree which worked the best. I ended up spending way too long focusing on getting any model working and decided in the end that for the assignement requirements, I didn't need to work so hard on such a trivial thing so instead I cleaned up my code and made it nice and modular and got it to work consistently 100% of the time with one of my tree models and then started working on the tree class. After I had gotten that written and simply drawing one tree object from the main class I went home. At this point I had temporarly disabled the fog to better see the trees as there was also weird colouration issues because the tree was using a material not a texture. My material loader took less time to make but I wasn't super happy with it

2/06/2019	5		__	Finally finished off my project. I quickly wrote up a forest class which chose a random size based off of the terrain size and then chose a random densisty based off of the random size. Then from that I got a tree count which made new trees in an ArrayList and drew them. For such a finalistic class, this took very little time. I then re-enabled fog, played around with the default settings so it looked slightly better and also made sure the player spawned a bit closer to 0, 0, 0 as he sometimes spawned right near the edge of the world. I then fixed some lag issues with the forest when sometimes millions of trees spawned in a tiny area (this may or may not happen again... Not really sure). I also then made a randomizer for night and day. I ended up not having a day night cycle and instead just had a random boolean for if it's night or day. Just to note: since I didn't have enough time and since I needed to focus on exams and other assignments, there is no collision detection with the trees. There is *technically* collision with the floor but this is a false collision as the player is literally just being set to the height of the floor beneath him. My favourite photo of my landscape is the second one attached. Perfect fog and texturing and the skybox and the trees all together make this look spooky and awesome. One thing to note and I'm not entirely sure why this is, but it might be to do with how I am using them, but adding the trees to a display list make the game ALOT (SIGNFICANTLY) more laggy. I think I might be drawing them or using them wrong but turing the display list off greatly improves FPS on ALL machines I've tested the game on (home computer with 1080ti, laptop with 870M, laptop with integrated intel graphics card, AUT lab computer with a 960 I believe). In conclusion: I only used a display list for the debugging lines and sphere's but otherwise it was way too laggy  
2/06/2019	2		__	As a last minute addition, I decided I wanted to have a bigger terrain to explore. Mine originally was only 500x500 which I thought was quite small. So in order to save on fps and also make a bigger world, I quickly came up with a trick that I thought was quite nice. In the terrain class, I would only draw the vertexes of my GL_TRIANGLE_STRIP if and only if the distance from the player to a given x, z position of terrain is under the skybox size limit (with some slight overhead). What this means is that the terrain is only ever drawn close to the player and never further away than the skybox limit. This massively saves on fps and allows me to get a realtively quick loading, very smooth 1200x1200 sized world. I tried going to 2000x2000 and whilst it is playable, it takes too long to load. 1500x1500 was also quite playable but the higher I went by 100 steps, it loaded slower and slower. At 3000x3000 it took about half a minute to load and at 10000x10000 it didn't seem to load after a full minute to I killed it. I also added this same feature to the trees but it didn't work the same way and for some reason there is a goldy-lox area around the entire rendering area where the trees go from unloaded to loaded to unloaded and back to loaded when the player gets close. 

Conclusion:

WHAT DID WELL: I was very happy with my water and fog systems as well as my flashlight. After coding them they worked brilliantly and almost immediately started working after finishing them. It made me quite happy. I was also extremely proud of my movement system. Even though it was a bit janky, I was very happy that I could control the looking for both pitch and turning with my mouse and movement with my keyboard.
What I wasn't that happy about was my object loader and Rawmodel classes. They seemed a bit buggy and whilst they worked perfectly for what I currently have in my scene, I felt that if I tried to add some other object, it would break the loader. There was a lot of issues I had with scanning in weird objects or objects drawn in older versions of blender.
I'm also quite happy with my debugging notifications. I think in some way debugging notifications really adds to it and makes it seem like a real game starting up with a loading bar or something. I also like how many controls I have for many different things

WHAT WOULD I DO DIFFERENTLY NEXT TIME: If I had more time and energy I would deffiantely try add shaders. I attempted to add them because I thought it would be so cool to have shimmering water and multi-texturing with shader blending but I never had time and was extremely confused by them so gave up. I would also try get multi-texturing working. I tried for so long and some might say I wasted way too much time on it, but I really wanted to get them working for a more realistic terrain.

WHAT ARE THE SHORTCOMINGS: I'm missing a lot of detail to the world. Right now there's kinda the bare minimum for the assignment. According to the teacher I did really well and my world looks really good, but I still felt like I could have added rocks or texturing other stuff or moving water textures but I did not have anymore time. I also don't really have animation in my scene. The character and skybox move and so does the spot light but nothing else is animated.