IN3005 Computer Graphics - Coursework Report
Name : Alain Siow

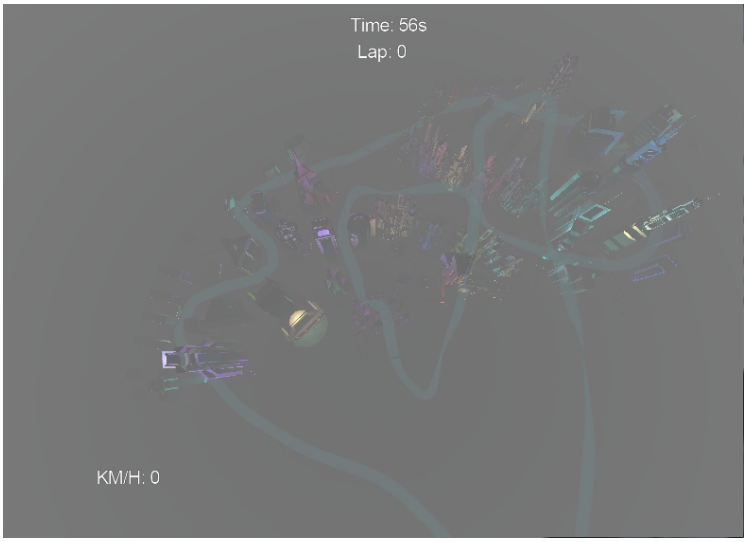
Overview:

Upon acquiring the requirements for this project as well as seeing the background and starship used for the labs gave me the idea of flying a car through a dystopian cyberpunk city, massively inspired by my experience of playing a VR application called AirCar. I also felt that this would be the best way to showcase up vectors for the track as it wouldn't make sense to have normal cars drive sideways or upside down, I wanted it to be grounded so it needed to be sci-fi.

The coursework game has a timer that keeps track of time in seconds as well as pickups that reduces the time. However, the plan from the start was to create a believable and lively environment for the player to fly around and immerse themselves in, periodically stopping to take beautiful screenshots of the game. Therefore the main focus was the appearance of the game, with tall sci-fi buildings lit by colours from different spectrum as well as cars flying around the city.

Prototype:

On the left is a very early screengrab of the city while I was planning out the route of where in the city the track should cut through. On the right is the final version of the route as well as the city.



Controls:

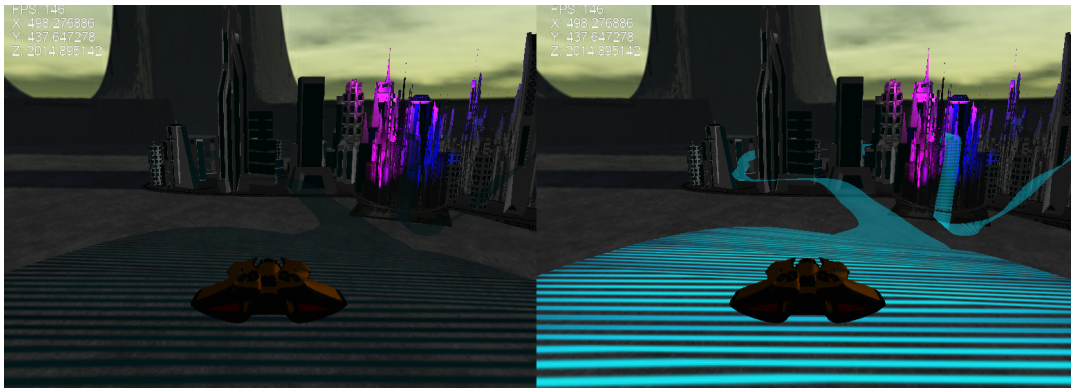
Key	Function
W,A,S,D/ Arrow keys	Car & Freeview camera movement
F	Toggle headlight
C	Change perspective
V	Freeview Camera
F5	Toggle path visibility
F6	Toggle Fog On/Off
F7	Debug HUD (Coordinates & FPS)

Instead of repeating for each section, please bear in mind all requirements were met and implemented.

Part 2 - Route and camera :

• Route :

Two different tracks were created using Catmull-Rom splines. The first track is the path the player drives on, created with primitives based on the centreline and offsets. Up vector is provided along the track twisting in all directions in 3D space with correct normals. Everything is rendered with appropriate texturing and lighting, do note the track is meant to be hologram hence light is turned on when rendering the track to create a bright blue holographic appearance. Pictured below.



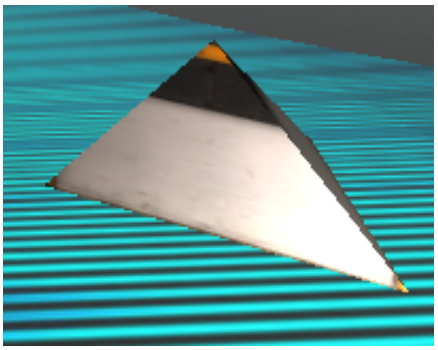
The second track is for the environment cars. 6 points of TNB are calculated at different points along this new track. Each point is populated with several vehicles (around 15) using specified offsets from the TNB point, each point is it's own "flock" of cars (6 flocks total). This way, there is no need to calculate TNB for each environment car.

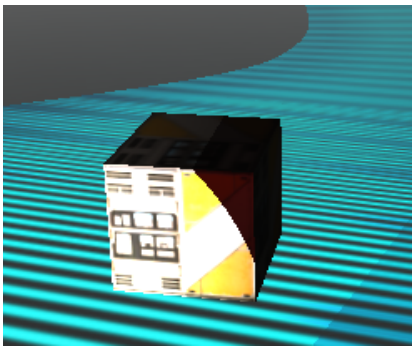
• Camera/viewing :

Freeview is retained as a way to take screenshots as well as debug. There are three different camera modes in the game (excluding freeview), first-person, cock-pit view and third person. The TNB frame is used and different values of offsets were used for each camera mode to provide the perspective in the game. All camera modes shift and rotate correctly as the player strafes along the route of varying angles.

Part 3 - Basic objects, meshes and lighting :

• Basic objects :





Two primitives are created, a cube and a tetrahedron, rendered as GL_TRIANGLE_STRIP. Appropriate texture coordinates and normals are applied to the primitives. Correctly transformed, oriented and scaled. Do note a constant rotation is added to them to help signify them as pickups. There are only two of these primitives in the scene at the start and the reason will be explained in the gameplay section later.

• Meshes:

There are a total of 11 meshes imported into the game. All rendered objects have appropriate location, orientation and scale. Vehicle meshes are repeated to create a busy city. The city itself is made up of three different meshes, the main city, center city, and a downtown city.

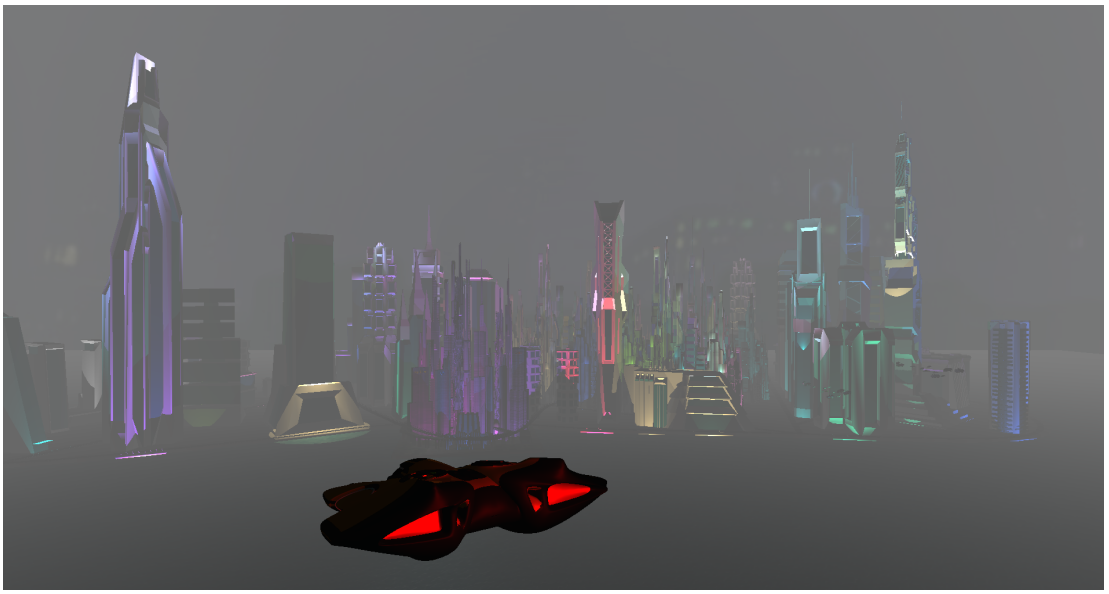
Some complications surrounding wrong winding triangles which stems from the meshes of the main city and center city arised when loaded into the game. How it becomes the meshes as seen in the final deliverable is documented in the Scene Asset section later.

• Lighting:

The game currently consists of a directional light (the sun) though this is hardly seen as the game is set in the dark though not complete darkness with a value of 0.1f. A pointlight coloured red is also implemented and serves as the backlight of the player's car.

The main bulk of the lights are spotlights, with 62 of them manually and carefully placed throughout the city with different colours, angle and position. One of the spotlight serves as the headlight for the player's vehicle with the ability to be switched on/off. This moving light uses the camera's view matrix as a part of its direction parameter so when the vehicle strafes left and right, the light will still point ahead of the track instead of to the left or right of the track. This also rotates the light in the right direction as the player goes sideways/upside down along the track.

The spotlights are the most important part of the city as it needed to be colourful with vibrant colours to create a futuristic cyberpunk city. That is why I went through great lengths to fix the main City mesh with plain white buildings, so the lights can paint the buildings so to say so the coloured lights really stand out in the scene.



All lights are attenuated in the fragment shader to my specific liking so they fade as the distance becomes greater. This is done by calculating the length of the distance between the light position and the vertex position (in eye coordinate). I have also taken out material calculation from spotlight and pointlight which seems to increase the framerate slightly with minimal appearance difference. Other attempts such as changing divide to multiply yielded no result and was reverted.

Part 4 - Head's up display (HUD), gameplay, and advanced rendering :

• HUD:

A very simple HUD is implemented which shows the time elapsed in seconds, number of laps and the player's speed.

• Gameplay:

The player is able to control their vehicle position by moving forward or strafing around the track. Vehicle drag is added to simulate a constant speed lost as the car moves forward if the forward key is not held down. This way the player can slow down to avoid obstacles or stop for a screenshot.

There is a timer which encourages the player to get from start to finish as fast as possible with pickups that reduces the time on the timer. The pickups use basic collision detection of distance between the pickup and the player.

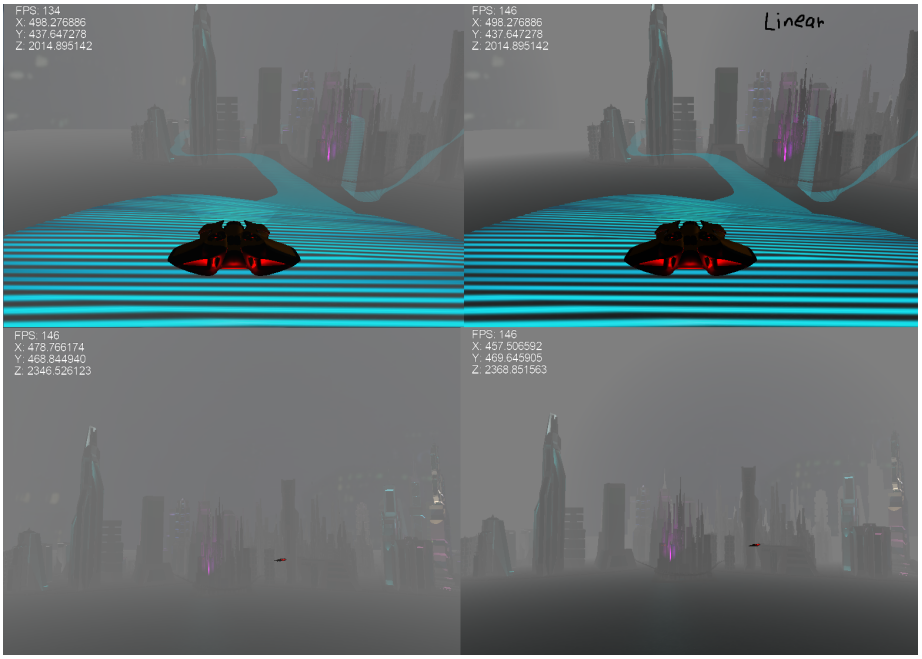
The choice of having only two pickups found at the start of the track is intended. Since we were told gameplay makes up a very small part of the coursework, the initial plans was to create a game similar to AirCar (mentioned above) and The Crew where exploration and taking in the sights is the primary part of the game. Therefore, the two pickups at the start are used to satisfy this part of the requirement and show that I can implement it and can easily be placed around the map. As I feel that littering the track with pickups take away the immersion of the city.

Reiterating the above, it is more of an exploration type game and was planned as is since the beginning. However, if exploration counts as gameplay, there would be two types of gameplay implemented.

• Advanced rendering:

Do note even though the shader program used is called SpotlightProgram, it is very much the main shader program consisting of different lights as well as fog. It was initially used to experiment with Spotlights and to differentiate from the main ShaderProgram but all shaders gradually moved into the SpotlightProgram where I make the changes and additions.

There are two advanced rendering techniques implemented for the game. The first one is fog. The fog I used initially was linear as guided under the OpenGL Cookbook. However, after session 8 I quickly implemented the exponential fog and found it to be more fitting for a dystopian cyberpunk city and retained the exponential version. Comparison below.



Exponential fog left, linear fog right.

The second technique is animation using discard in fragment shader. By toggling the visibility of the path with F5, the path disintegrates with the discard animation instead of disappearing. This fits perfectly well with the holographic theme of the track. Two uniform booleans were passed from the client, one that checks whether its rendering the track and the other checks whether it should discard the path over some time.

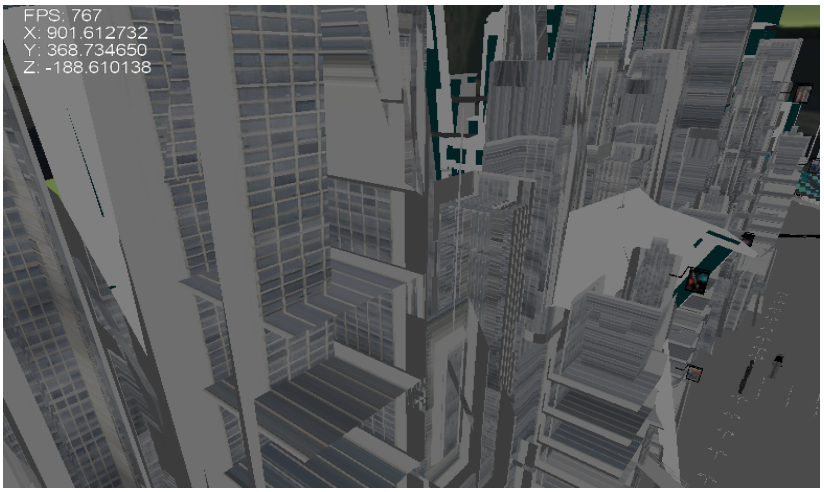
```
if(!showTrack) {
    if (vTexColour.r < fract(discardTime/5)) {
        discard;
    }
}
```

• Scene Assets:

The main city and center city were two very problematic meshes to use but did not deter me from fixing them, this section details how I fixed these meshes using a combination of Blender and OpenGL.

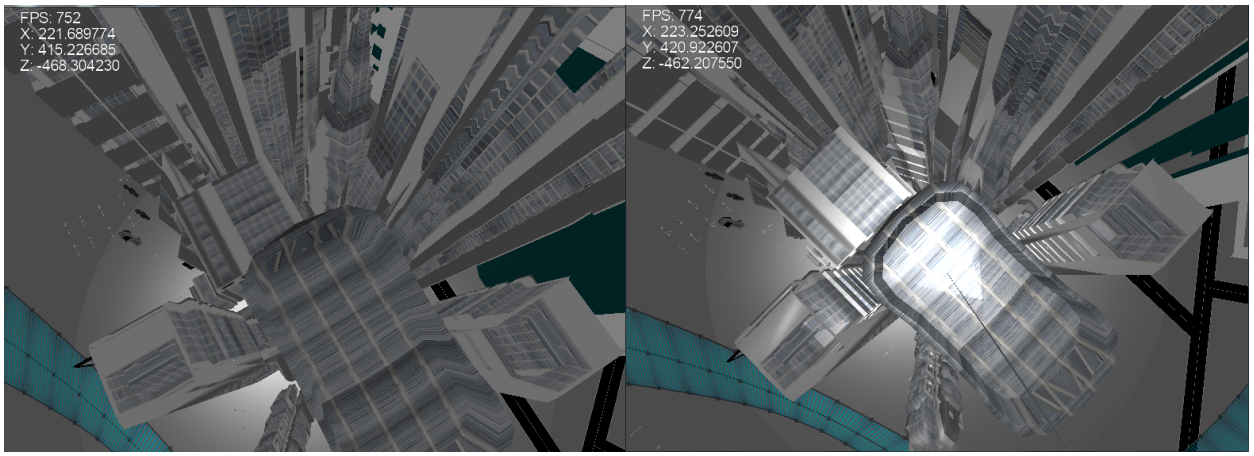
The main city mesh gives an error when imported to the game or the older Blender version recommended from Advanced Games Tech which uses Assimp configured by Dr. Slabaugh as well. The latest Blender version was then downloaded and the mesh opened correctly, though exporting it as obj and importing it to the game will yield a different error. I figured the Assimp we are using could be outdated and loaded the mesh to the old version of Blender mentioned above and then exported it as obj again. The mesh then finally loads into the game but without the coloured materials. I attempted many different methods to no avail and finally decided to manually add material colours for each building in Blender instead as the mesh is perfect for what I have planned which is mentioned in the lighting section.

The third problem arised as both the main city and center city mesh for some reason consist of triangles with reversed winding which creates this see through effect as you look at the building. Picture below.



This issue was briefly fixed by turning off CULL_FACE when rendering the two meshes but it very quickly became evident this is not the ideal solution when lighting was implemented. Because the normals for these triangles face the wrong way, they do not react to light. After some time googling (StackExchange, 2015), there is a function in Blender which recalculates the normal for a given mesh. Using this, I was able to fix most of the triangle winding issues and have them react to light.

Below is a before and after picture.



This was not a complete fix as some triangles still have the wrong winding though it is a minimal amount compared to before but the main issue of lighting was fixed. In order to fix these small see through bits, CULL_FACE was once again turned off. Therefore with the combination of Blender and CULL_FACE, the meshes function well enough you would not notice the defects. All other meshes were a simple straightforward import excluding rotation and scaling in Blender.

Meshes:

CenterCity: Royalty Free
https://free3d.com/3d-model/sci-fi-downtown-city-53758.html (14/03/2021)

City: Royalty Free
https://free3d.com/3d-model/sci-fi-city-83682.html (13/03/2021)

Downtown: Royalty Free
https://free3d.com/3d-model/sci-fi-downtown-city-23035.html (14/03/2021)

FlyingCar: Royalty Free
https://free3d.com/3d-model/hn48-flying-car-10381.html (17/03/2021)

Freighter : Royalty Free
https://free3d.com/3d-model/si-fi-freighter-13915.html (17/03/2021)

PatrolCar: Royalty Free
https://free3d.com/3d-model/city-patrol-vehicle-84293.html (17/03/2021)

PoliceCar: Royalty Free
https://free3d.com/3d-model/flying-car-nr3-25519.html (17/03/2021)

Starship: Royalty Free
https://free3d.com/3d-model/wraith-raider-starship-22193.html (17/03/2021)

Transport : Royalty Free
https://free3d.com/3d-model/futuristic-transport-shuttle-rigged--18765.html (17/03/2021)

Man: Royalty Free
https://free3d.com/3d-model/human-man-low-poly-55381.html (17/03/2021)

Textures:

Sea: Royalty Free
https://www.sketchuptextureclub.com/textures/nature-elements/water/sea-water/sea-water-texture-seamless-13246 (21/03/2021)

Crate and Tetrahedron: Attribution-NonCommercial
https://polycount.com/discussion/74895/pk02-sci-fi-texture-set-released (21/03/2021)

Skybox: Creative Commons
https://sketchfab.com/3d-models/skybox-a-town-in-mist-aff4aef944024c12a426af84a86e0217 (15/03/2021)

The skybox was not seamless and therefore required modifications. Gimp and Blender was used to correct this under the guidance of this tutorial.
https://www.youtube.com/watch?v=P29p8-0K1Yg

References:

(StackExchange, 2015)
https://blender.stackexchange.com/questions/12165/how-to-make-all-faces-flip-to-the-right-consistent-direction (15/03/2021)

Reflectance:

The strength of the game is definitely the environment as a whole. As mentioned before, the plan had always been to build an immersive dystopian cyberpunk city the player can fly through and explore. The final result came out better than I had expected and was well received among my friends. In that regard, I've done what I set out to do. Aside from that, I have always taken pride in the level of polishness and attention to details I add to all my projects. All the spotlights, track pathing, environment vehicles, which spotlight colour should this building have and where, are all carefully calculated and manually placed.

The weakness of the game at its current state is that there is nothing much to do outside of exploration. Same could be said about games like AirCar and The Crew, like them it was a conscious decision to leave the game as is as an exploration game and nothing more. With that in mind, there isn't really a weakness per se. However, due to having to turn on culling for the couple of city meshes and the numerous lights, I have struggled to maintain 100fps flying in the middle of the city. Even though optimisation I would guess is not really a part of the course, I have attempted to optimise it with some success as documented in the lighting section. I personally don't see performance as a weakness for this course but I thought I will include it here anyway.

One might argue the project's success is due to the city meshes. It is true on one hand but on the other, it definitely was not an easy journey and looking back, I would have told myself to just use a different mesh. From all the problems described in the meshes section, tremendous willpower and dedication was needed to make them work. I still remember during a lab session, someone with the same idea as me said their city mesh did not show texture in obj format, which is quite common with free assets. He then gave up and used a different one instead, and rightly so as it really isn't worth the time fixing it and it probably would not have counted towards overall marks. This thought very briefly crossed my mind but I decided against it and manually painted each building instead. However problematic the mesh was, from countless error import to wrong triangle winding, I took it as a learning experience and with a bit of luck was able to sort out most of the issue and in the process further honed my Blender skills. I'd like to add I am the type of person that gets no satisfaction if the usage of the work of others is the sole reason for my project's success. Having to deal with the meshes the way I did in a way offset that balance.

Overall, I am very happy with what I have accomplished in two weeks, which was the planned amount of time I wanted to spend on this project. Each day was over 12 hours of work but that is due to my work ethic of working long hours on it instead of spreading it over periods of time. I am especially happy with learning how to convert any image to a seamless skybox using Gimp and Blender. As a result, my photoshop and Blender skills have improved by a large margin. Last but not least and perhaps the most important, being able to understand and modify GLSL code. I still remember 3-4 weeks into the course, I happened to install a shader for Minecraft and liked every aspect of it except for the water reflection which I felt was exaggerated. Reading some post online suggested I might be able to modify this in the files

and to my surprise, they were written in GLSL. I made some very simple modifications and was able to achieve the water reflection I wanted.

As for expansion, due to the nature of the requirements, racing games like this tend to require enemy players or AI to keep them interesting. Multiplayer could be added and mechanics from Mario Kart could be used but these are very much outside the scope of the module. If this game ends up being one of the many maps for a racing game, I am sure it will be a good looking one. This has been a very intense past couple weeks for me with an immediate move back to my individual project with similar work hours. Nevertheless this is a piece of work I am very happy and proud to have as an addition to my portfolio page.

Recorded a video for my portfolio page : https://www.youtube.com/watch?v=4ZMfNF2tZz0