UML -Class diagram Explanation

The game was only a small game so not many classes were needed as the game is very simple. Most of the classes are needed for the networking. Attached are 2 PDF files with UML diagrams. The first one contains a diagram showing how all the classes connect together. All the classes rely on the main.cpp file, which is the start file, which is looped through constantly, coming off this is the game class and the client class both of which use the main, to be initialised or call functions. Relying on the game class are the two line classes, which are just for each different line as they are heavily linked and used in the game class and are there mainly for creating and fetching the data stored for the lines. From the Client class this is then linked to the message types as this tells the class what data it is sending to the server, which is then also linked as the client and server will exchange data constantly.

All of the main classes that don’t handle networking are quite simple, using lots of getters and setters to retrieve values and in the game function a few update functions which handle the updating of each scene. Whereas the networking code handles things such as sockets, receiving data, sending data and looking for connections .

POST MORTEM

This assignment I feel went okay as there were bits that I found really hard, but creating the general basic tron game I found really easy as it was enjoyable to code although it is by no means perfect.

The movement of the “car” I found really easy to implement after having a look through the SFML database to see what was part of it. However, I did want it so that you could only have one key pressed at once, however I couldn’t find a function that would allow me to do that as there was only “KeyPressed”, so you can press two keys and move diagonally, however sometimes you do crash into your own line while doing this.

Implementing the trail behind the car wasn’t hard as I used a Line class and created instances of the line in a vector which is then rendered as the car moves around, originally this was all connected together, however once I added the collisions in it became separate boxes that would need to be offset slightly as otherwise the car would collide with the line instantly, causing the game to instantly be over. I have added this into a feature to add an element of skill as the line if you are careful can allow the car to fit through a gap with precision, which I felt added a fun element to the game, while being like the original tron it also allowed for a more.

The basic SFML library was good to work with as I felt like for this project the basic shape of a rectangle was enough to produce the tron game. I did however add a custom font into the game, however I didn’t feel the need to add any custom textures etc.. although it may have made it look better.

The hardest part I found was implementing the network aspect as when I attempted to add it the program gave me linking errors, however I then fixed this as I discovered that the sfml-network-d.lib and the sfml-system-d.lib files were not linked into the project, which once I added theses in fixed the errors I was experiencing; however this did take me several days to figure out as I believed that the project was set up fully with the linking.

Sending and receiving sockets was my next problem, as the program would just get stuck in an infinite loop. I managed to get the network working to the extent that it would assign the player a number – 0 or 1 depending on the client ID so that it didn’t assign the same number multiple times, as there was only going to be 2 clients connect at any time there isn’t a need for more than 2 numbers. The network also connects a client to a server and all the code is in place for the client and server to send and receive data from each other to update positions of objects, game states etc.. however I could not get it to be able to pick up the data in the correct place as the program would either loop infinitely or lose the data from the packet before I could retrieve it.

I also struggled with threading as none of the tutorials would work correctly (and no I wasn’t using the SFML threading) as the thread would error out or not work at all and not be iterated through, which along with the networking became a real problem, but I think I could have looked a lot more at trying to get threading to work but my priority was the networking as this was a networking task.