**Post Mortem**

In this document, I am going to be discussing the process of creating a networked Tron game. I created the game using a client-server structure with the game running on the server and the client renders the information that the server sends to them. This is to prevent the client from cheating.

There were some difficulties during the development process of the game. One of them being how the server sends the trails information to the client without being too large. This would’ve been a problem as the method I initially planned was to record all the players’ previous location and send it to the client. However, after some thought I felt that the client does not need to receive this information to draw the trail. My solution was to process and draw the trail in the client by adding a vertex every frame at where the player is and the server records the previous location of the player and uses it to check collisions. This method may not be the best solution as it means that the server will be recording the position of the player every tick and a new vertex is drawn every frame. This could lead to performance issues later as more and more data is written and needs to be processed.

There could be many improvements to my implementation of Tron. One of them being to can dynamically add more players into the game and maybe add speed modifier so players can choose to speed up or slow down their light cycles strategically.

However, everything else like threading and networking I did not really have much trouble with after doing a bit more research on it before starting the implementation of the game.

In conclusion, I feel that the final game is quite like the original game with many improvements that could be made to it.