Tank application

This application utilizes the HTML 5 canvas to allow it to draw all the components and display the interactions. HTML canvas is sufficient for this game but more complex games where more interactions and fast response times would likely induce latency. The aim with this applications was to show the platforms ability to cope a lot and request and display and process them in close to real-time.

When tackling the problem we started by creating the complete game in the Desktop applications with the controls and logic all situated there. The next step was to move the controllers out which was successful with a little tweaking since had to add way to detecting which player was which. After that the game logic was then moved out from the Desktop and to a server that would deal with all collision calculation and movement tasks and then at every interval an update of all the object to the canvas.

The main complexities with this was the amount of requests being sent by each user and the updates from the server. In order for there to be no noticed jittery movements the refresh rate needed to be less than 30 milliseconds on the canvas since any less than this would be a noticeable drop in frame rate. This means the server will send an update of all locations or all the items on the screen at this interval.

Some other small complexities were the simple maths for making the tank move in a straight line along the path it was facing at any angle. The bullets then had to be fired along this path and change direction when they bounce. There position had to be monitored for collisions with other tanks not including yourself, and also when it exceeded the bounds of the screen.

Some improvements that can be made are mainly to reduce the amount of data being sent. Only send updated object then the whole screen and sprites would not have to get sent over. This wasn’t necessary for the current implementation but a future planned implementation with obstacles may required this.