**Implemented functionality table**

|  |  |
| --- | --- |
| **Function** |  |
| Client establishes a connection with the server | Yes |
| Client is assigned a unique ID when joining the game | Yes |
| Client displays up-to-date information about the game state | Yes |
| Client allows passing the ball to another player | Yes |
| Server manages multiple client connections | Yes |
| Server accepts connections during the game | Yes |
| Server correctly handles clients leaving the game | Yes |

**Protocol**

**Client Threads**

In the client process run 2 threads, both by the main class but one of them overridden in the ClientSideListener.

The ClientProgram thread is created when the program is run and runs until the program is closed. Its purpose is to create a client object, the Client Listener thread, to take input from the user and pass commands based on this input to the server through the socket. It also informs the user of the available commands.

The ClientSideListener thread is created after the program is run and passed the try block in the main method and runs until the program is closed. The purposed of this thread is to listen to information passed by the server, to process this data and display to the user the processed information into an easier to read format.

**Server Threads**

In the server process runs the main thread of the server and one thread for each connected user.

The main thread is created when the main method of the server class is called and is terminated when the server is closed. Its purpose is to wait for incoming users trying to connect and to create a new thread for each connected user.

The other threads are created when a user tries to connect to the server and run until the user decides to leave the game. The purpose of one such thread is to create a player object for the user, update the list of connected players, listen for commands from the client side, process those commands, write updates about the game state to each connected player and write to the client-side information based on the processed commands.

**Project Review**

I enjoyed working on this project and I believe I did a good job with it.

The easier parts were the creation of commands and the Player and Game class.

The most challenging part was updating each client automatically with information about the game state when a player joined/left the game and when the ball was passed. Another challenging part was splitting the client program into two threads, one for listening for updates and information from the server and the other for taking input directly from the user.

I am proud of the entire project but particularly proud of the server update system.

I don’t believe there are any problems with the program.

The project management was alright, I started with the server program development and used putty to test it until I got everything right and until I created the client program.

If I was to do the project again I wouldn’t do it different. I genuinely had fun working the way I did and it wasn’t as time consuming as I expected it to be.