Online Technology

Project 1

TDD

**Aim:**

The aim of the project is to use Winsock in C++ to create a server and then to create clients to join the server and interact with one another. The aim is to make the game Tag with 3 players that run on different consoles/machines. Updating the player on one client will update a different player on the other client. The players will detect position updates, color updates and end game updates as well as normal chat message updates.

**Server:**

The server will start up Winsock information, set the address to the local address of 127.0.0.1, and a port of 1111. It will then then bind the address to a listening socket and listen for connections from clients. It will set up a separate thread to manage the sending out of packets from clients. Upon receiving a connection from a client, it will create a new thread for that connection to handle its incoming and outgoing packets.

When processing packet the server will use flags to determine which type of packet is coming, then get ready for the data that matches that type. The server then takes the data and bundles it into a packet, this packet is then sent out to every other client than the one who it came from so the other clients can update their game accordingly.

The server does some checks for safety and ensure the whole package is sent and not a part of it.

**Client:**

The client is setup very similar to the server except that it only receives packets from the server and will only send packets to the server. When it receives packets from the server it can then use that information in the game to update a position or a part of the game. The client is setup with the same IP and port as the server so they match and then it attempts to connect. It’s thread for handling packets is very similar in structure, just doesn’t send to other clients and acts on information as opposed to just sending it along.

**Game:**

When starting the game, it attempts to connect to the local host IP and server. If it failed then it determines there is no server and becomes the server and host for the game. It then sets up a client to play on it as well. If a server exists, it becomes a client who only sends information to and from the server. The client class then sends out a separate thread that listens for new connections constantly. When a player sets the color or moves or ends the game, the client then tells the server/hosts. The host then tells the other clients the new position and the clients update their game with this packet knowledges, the game then detects collision on the server and if it happens then it ends the game and displays time.