

Project Title: Designing a Gaming Server in C++

Team Members:
Nitin Kumar & Prasan Chandak

Project Requirements:

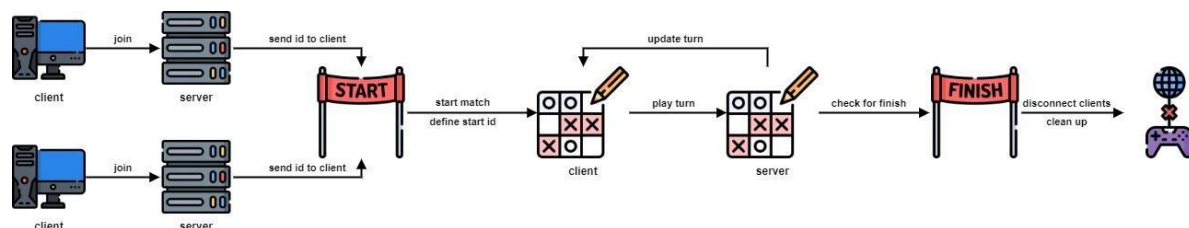
1. The function of a game server is to wait for connection from the clients (players) initially until the number of players required to play the game connect to it.
2. Once the players are connected, the server communicates a player's moves to all other (or the concerned) players.
3. As per the game, the moves are sequential(one after the other thus one at a time) the other player needs to wait for the other player to move.
4. The function of the server may also be to match players in a multiplayer game according to some criteria.
5. The server can also manage multiple game

Design Principle for a simple Tic Tac Toe game:

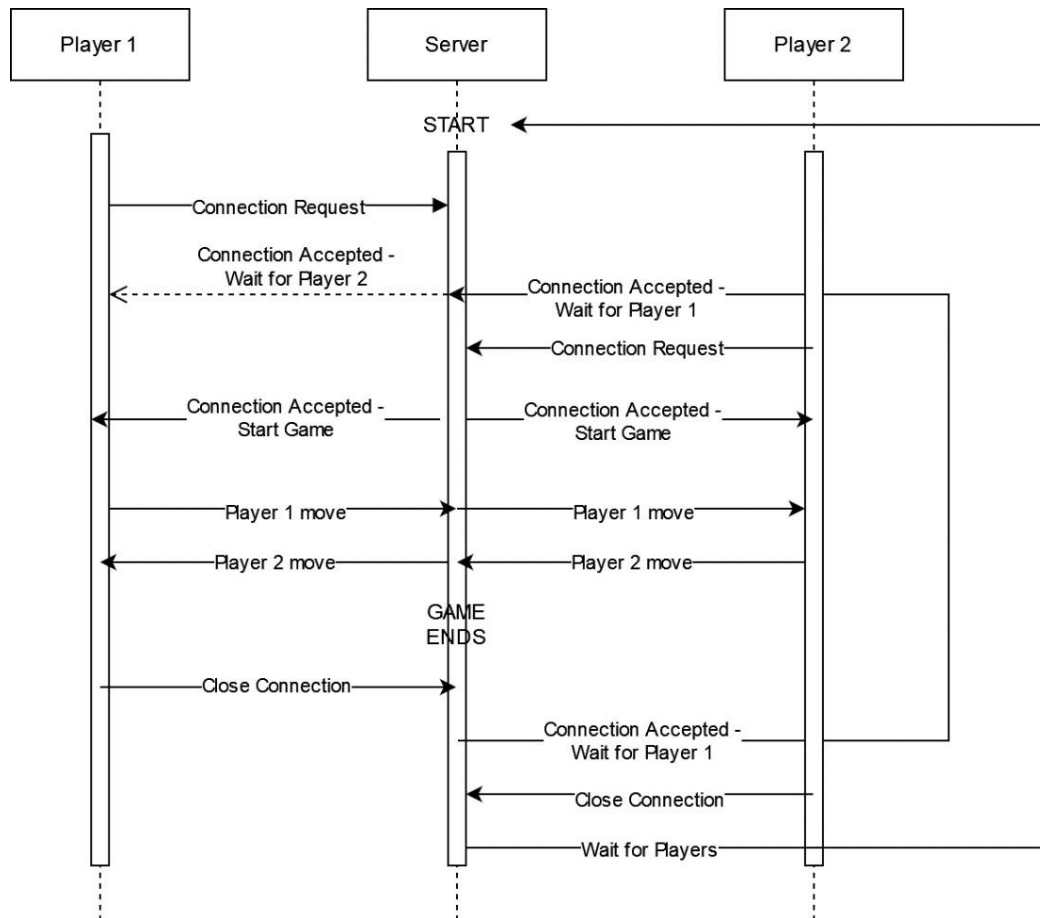
The game can use TCP/IP protocol to establish connection between client and server. This game should be able to connect up to 2 clients. Any clients after 2 connections would not be connected to the server or be kept in a lobby for match up with future players. When a player marks a value, the action listener fetches the value and sends it to the server which is then returned back to the other player. The received value is written as the opponents move. A toggle function can be used for the buttons to prevent players making more than 1 move at a time.

Project Design:

Principle:



Sequence Diagram:



Code Explanation:

1. Functions used on Server Side

- `create_socket()` - creates and binds a socket and return socket descriptor
- `get_clients()` - listen and accept the client connection request
- `pthread_create()` - create thread for execute_game function
- `execute_game()`
 - `check_move()` - for checking either move is valid or not
 - `get_player_move()` - get both player move alternatively and update board
 - `update_board()` - update the board on server side
 - `send_update()` - send update to both client
 - `check_board()` - check for winner after every move
 - `recv_int()` - to receive int msg from both client
- `write_clients_msg()` - write const char message to both clients
- `write_clients_int()` - write int message(move) to both clients

2. Function used on client side

- `connect_to_server()` - create and connect to the socket and return the socket descriptor to communicate with server
- `Recv_int()` - to receive int msg from server side
- `Recv_msg()` - to receive char msg from server side
- `Take_turn()` - get client move and send the move to server
- `Get_update()` - get the move as int from server side and update the board
- `draw_board()` - to draw the tic_tac_toe board