Server Side

1. Create socket
2. Bind socket to port
3. Listen for connections
4. Accept Connection
5. Initialize board
6. Start Game Loop (while true)
   1. If Server’s turn (always the first turn)
      1. Input choice (Always on first iteration)
      2. Check if position is occupied
      3. Check for win or draw
      4. If false, send updated board to client
   2. Else If Client’s turn
      1. Input choice
      2. Check if position is occupied
      3. Check for win or draw
      4. If false, send updated board to client

Client Side

1. Create Socket
2. Convert address from text to binary
3. Connect to Server
4. Start Game Loop (while true)
   1. Wait for server to send the board
   2. Check if received buffer contains a game result
   3. Display the updated board
   4. Input choice
   5. Check if position is occupied
   6. Check for win or draw
   7. If false, send updated board to client

Code for PlantUML

@startuml

!theme aws-orange

actor Client

participant Server

== Connection Setup ==

Server -> Server : Create socket

Server -> Server : Bind socket to port

Server -> Server : Listen for connections

Client -> Client : Create socket

Client -> Server : Connect to server

alt Connection successful

Server -> Client : Accept connection

Client -> Client : Display "Connected to server!"

else Connection failed

Client -> Client : Display "Connection Failed"

end

== Initialize Board ==

Server -> Server : Initialize board

== Game Loop ==

loop While true

Client-> Server : Wait for server

Server -> Server : Input choice

alt If input is invalid

Server -> Server : Invalid choice. Try again.

end

alt If position is occupied

Server -> Server : Cell already taken. Choose another.

end

alt If input is valid

alt If Win

Server -> Client : Notify Client that Server wins

Server -> Client : Close connection

end

alt If Draw

Server -> Client : Notify Client that Server wins

Server -> Client : Close connection

end

else

Server -> Client : Send updated board

end

Client -> Client: Input choice

alt If input is invalid

Client -> Client: Invalid choice. Try again.

end

alt If position is occupied

Client -> Client: Cell already taken. Choose another.

end

alt If input is valid

alt If Win

Client -> Server: Notify Server that Client wins

Client -> Server: Close connection

end

alt If Draw

Client -> Server: Notify Server that Client wins

Client -> Server: Close connection

end

else

Client -> Server: Send updated board

end

end loop

@enduml