Tic-Tac-Toe

TTT Protocol

By Khuzaima Mushtaq

Ghal Goziker

Michael Rossiytsev

Introduction:

This document is provides the documentation for the Tic-Tac-Toe Protocol also known as TTTP. In this document we will discuss how the each element of the protocol is defined and implemented, how to build and execute our Tic-Tac-Toe game, and how to play our game from start to finish.

The Design:

The architecture that we chose for our project the Client - Server architecture. Our project assumes that anyone using TTTP can host a server based on our protocol and also that anyone can connect to any TTTP server and play the game if they follow the TTTP protocol. We also do not make any assumptions about the game interface that the client is using. For example, one client could be playing the game on a gui based interface, and another client could be playing the game on a text based interface. The game is played on the server side and the clients can only send their moves to the server when it is their turn, how these moves are parsed on the client side into an interface is totally up to the client. All games are hosted on the server, Clients can create games on the server or join games currently available on the server.

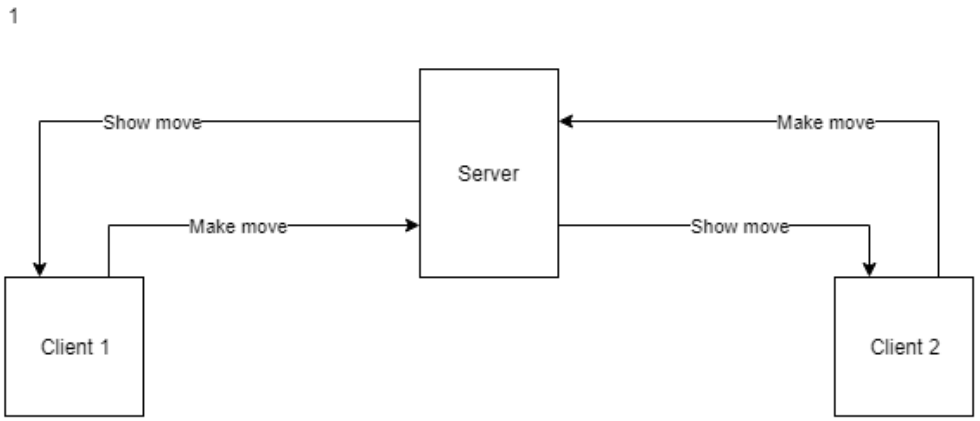

Figure 1

Figure 1 shows the basic architecture of our Protocol.

The Protocol:

This section defines the protocol and all its features. A majority of the messages are broken into three categories that involve HOSTing, JOINing, and GAME-ing. The assignment

required (Register, List, Create, Join, Exit, Unregister) commands are implemented within the scope of each of these categories.

The “HOST + data” message includes data messages that register and unregister a user from a list of possible games. The “JOIN + data” message includes data messages that joins and essentially creates games. The “GAME + data” message includes app specific protocols required for game functionality. All of these categories include an “exit” data type that removes them from the category menu back to the main menu. More details about each protocol are described below.

Tic-Tac-Toe Protocol (TTTP):

Protocol Model: Client/ Server

Protocol Port: 8564

Protocol Format:

****
Command Data
****

Commands:

HOST:

Command: HOST

Data: Name of game to host

Description: This is essentially the ‘Register’ command specified in the assignment requirements. Tells the server to create a joinable game called Data and that client is ready to play. When the user presses ‘Submit’ a “HOST <gamename>” message is sent to create the game.

Examples:

- “HOST <game name>” - begin hosting a game and waiting till somebody joins
- “HOST exit” - exit the host menu back to the main menu

JOIN/List:

Command: JOIN

Data: Game Name to join

Description: This is essentially the ‘List’ command specified in the assignment requirements. The JOIN commands also functions as a LIST command. For example if a client ONLY sends JOIN with no data then it is interpreted as a LIST commands and the server will send back the LIST of games available. Then, If the client sends ‘JOIN BoBSGame’ then the server interprets

it as JOIN GAME and sets the game in motion and removes the game from the List of available games.

Examples:

- “JOIN ” - lists games
- “JOIN <selected game>” - creates game with the selected host game
- “JOIN exit” - exit the join menu back to the main menu

GAME:

Command: GAME

Data: ‘ready’, ‘wait’, ‘move’

Description: This command is used to play the actual game. Based on the Data this command has various functions. the Ready and Wait Data are used by the server to inform clients if it is their turn to make a move. For example the server might tell player 1 to wait while player to makes a move and vice versa.

The ‘move’ data is the actual place on the tic-tac-toe board that the client wants to make a move on. Tic-Tac-Toe Board sample

0,0 0,1 0,2

1,0 1,1 1, 2

2,0 2,1 2, 2

The tic-tac-toe board is basically a double array, on each of which a move can be made. So when it is a clients turn the client would send the the command as ‘GAME 12’ indicating that the client wants to make a move on board block [1][2].

Examples:

- “GAME ” - Start the game
- “GAME <coordinates>” - Makes a mark on the game board at specified coordinates which are detailed above.
- “GAME exit” - exit from the game back to the main menu

The Game:

This section describes the implementation of our protocol and the game that we made based on the protocol. We made our project in Java because java sockets are easier to use than C/C++ sockets and all team members are comfortable using java. Our GUI is made using Java

swing library which is a GUI widget toolkit for java. Figure 2 shows the class diagram for our game. Figure 3 attempts to illustrate the class diagram more seamlessly.

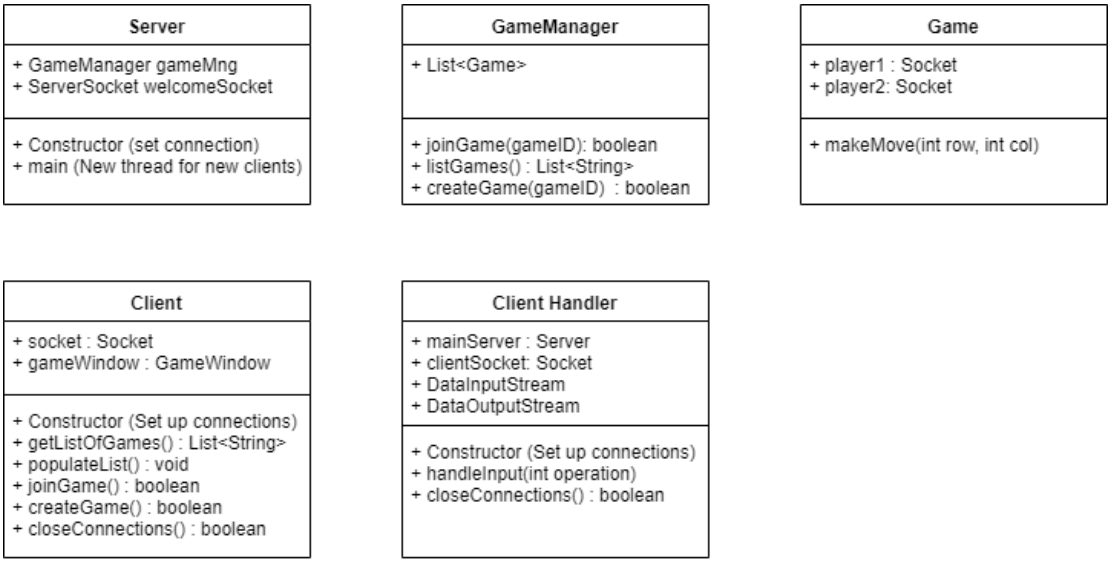

Figure 2

Server:

The server is the host that all clients connect to. The server manages the games using the GameManger and allows clients to join games through that.The server also uses the ClientHandler to handle each client connection and manage multiple clients in multiple threads.

GameManager:

The GameManager manages games and keeps track of all the games running on the server.

Game:

The game keeps track of the actual TicTacToe game between two clients and all the moves that have been made, if there is a winner etc.

ClientHandler:

The ClientHandler is used by the Server to manage all the clients. The clientHandler has the input/output streams to a given client and handles the input coming from a given client.

Client:

The Client is the user that interacts with the server and is able to create games, join games and play games. It uses a GameWindow, which is the GUI for our game. The gameWindow and

the GUI is updated based on the information the server gives to the Client about the moves that have been made on the game board.

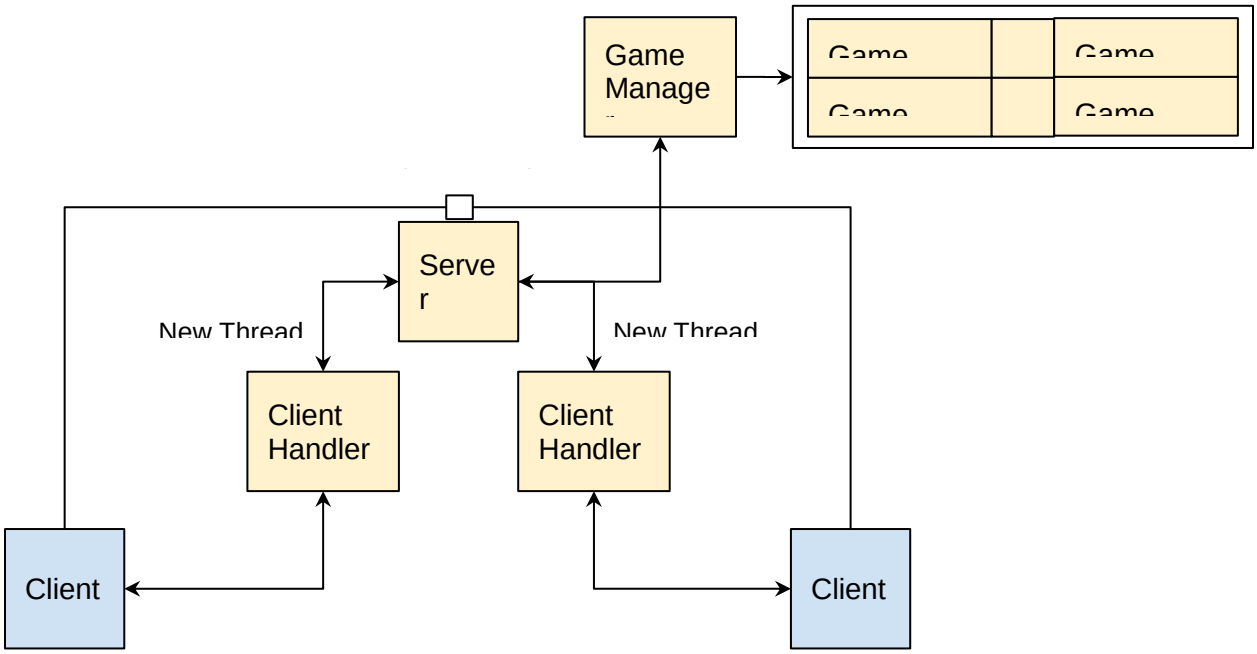

Figure 3

Compiling, Running, and Playing:

To compile and run our project simply download the all the project files and open them in any java IDE (eclipse, Netbeans, etc). To be able to actually build the project, you'd have to create a new project in the IDE first and then import all of the project files. Use the IDE to run and compile the server or the client. You can run the server or the client on any machine, just make sure that the IP and port (on line 476 in Client.java) that the client is trying to connect to is correct and there is a server running on that IP. Once the server is running and the clients are connected, host a game in the host page using any client. Once a game is hosted, use any client to join the game in the join page. Once the game has been joined the server will show if it is your turn or not. Make moves on the Client GUI by pressing buttons which correspond to the appropriate position on the Tic-Tac-Toe Board. The clients go back and forth making moves until the game is done, at this point you can host or join more games or exit.