

CSCI-6050

Project Proposal

Richard Hoehn

November 9, 2024

Overall Concept

The project is a networked Tic-Tac-Toe game using a client-server architecture implemented in the C programming language. The server acts as the game coordinator, handling multiple clients concurrently using threads. Once clients connect to the server they are paired for matches, and can play a full game of Tic-Tac-Toe with each other via the server. The server ensures proper synchronization and game state management to allow for a smooth multiplayer experience.

Overall, the server will send and receive information to display to the client, meaning no major logic will reside on the client's side.

Server Description & Operation

The server operates as a centralized main processor of the game, accepting incoming connections from clients.

When clients connect, they are assigned a thread to handle communication and game logic. The server pairs clients into matches, manages the game state, checks for a win, loss or draw, and sends information updates back to each clients.

The server utilizes sockets for network communication, and thread-based concurrency is handled using threads. Synchronization mechanisms, such as

who gets to play when, are built to ensure that shared games are accessed safely by different threads.

Services

The following is a high-level list of the services that the server will handle:

- **Matchmaking Service:** The server pairs connected clients into game sessions. This will be a one-by-one association, meaning as clients connect, they are paired in order of connections to each other. The first needs to wait until the second client connects.
- **Game State Management:** Tracks the state of each game, processes moves, and checks for game outcomes (win, loss, or draw).
- **Client Communication:** Manages sending and receiving game board layout to clients, and consequently waits for the client to respond.

Client Description & Operation

The client program connects to the server over a network and interacts with it to play Tic-Tac-Toe. Upon connecting, the client waits for a match / game to be initiated. If it (client) is the first connection, or essentially an "odd" connection, it waits for the next client to connect. Once paired with another client, the game begins, and the client can send moves "X" or "O" plays to the server. The server processes each move and sends back the updated game state and board display until a win, loss, or draw is declared.

Clients utilize sockets to establish network connections and communicate with the server. The user interacts with the client program through a simple text-based/command-line interface for inputting their moves and viewing game progress.

Languages & Libraries Used

The programming language used for this project will be the **C Programming** language, as we have used in class this semester. The following **Libraries** will be utilized to get the project completed:

- `csapp.h`:
For socket programming and network communication. This was provided by the content class Lab 6 files we used.
- `stdio.h` and `stdlib.h`:
For input/output operations and memory management.
- `pthread`:
For handling thread-based concurrency on the server.
- `ncurses`:
For creating a text-based user interface, providing features like window management, user input handling, and screen manipulation & clearing.

Conclusion

This project aims to develop a simple and interactive Tic-Tac-Toe game using server-client architecture with concurrent thread handling. The use of C programming, socket communication, and libraries such as `pthread` and `ncurses` provide a solid experience in building networked applications. This project furthers the learning of networking and threading concepts but also demonstrates real-world application development through interactive game playing programming.

Signature:  _____
Richard Hoehn
PhD Student MTSU