Networked 2-Player Game: Socket Programming & Tkinter

This project combines socket programming and Tkinter to create a networked 2-player game with online features. It includes a game lobby and chat system to enhance interaction and gameplay experience.

Key Technologies: Python, Sockets, Tkinter GUI.
Team Members - Gaurav (2303111)

by Gaurav Dewangan

Core Game Mechanics with Tkinter

Game Logic

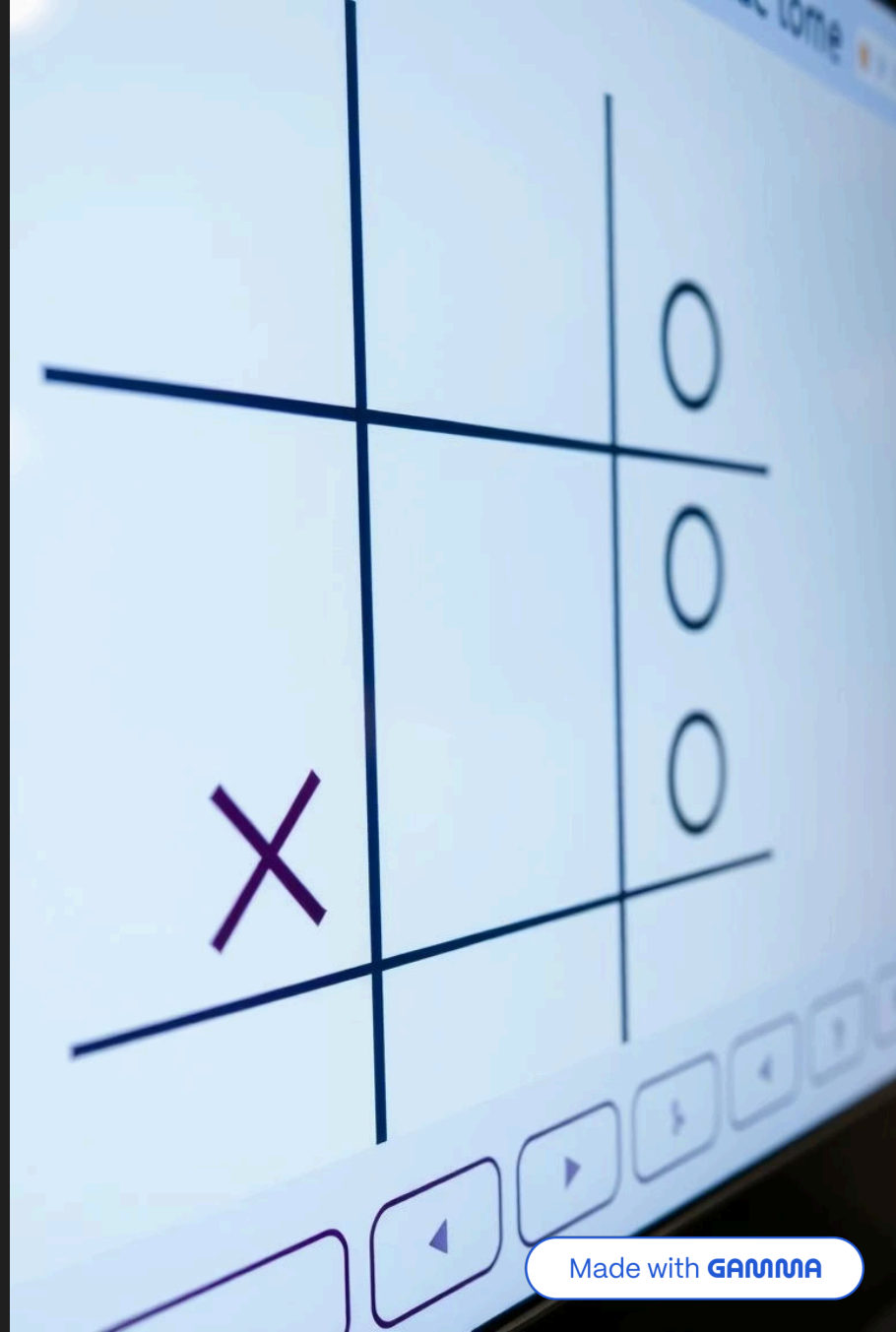
Turn-based rules, win/loss conditions, and player interactions define gameplay.

GUI Elements

Canvas, buttons, and labels crafted in Tkinter create an intuitive interface.

Real Time Updates and Graphics

A classic HP bar helps to visualize the gameplay clearly.



Socket Programming: Client-Server Architecture

Server Role

Maintains connections and synchronizes game state between players.

Client Role

Displays game GUI and handles sending/receiving game data.

Defined CodeList for each message

Messages sent are send with a code which forwards the message to the appropriate function

Data Transfer & Errors

Uses send() and recv() for communication with robust try-except error handling.

Lobby Implementation

User Authentication

Login system to identify players and manage sessions securely.

Lobby Features

Displays online players and a variety of tabs

Chat System

Online private, lobby chat and global chat systems,

Friend System

Add other users as your friends and see if they are online

Chat Functionality

Real-time Messaging

Players communicate instantly using socket-based message exchange.

Seperated Chats in different Tabs

Different tabs seperate Global, Private .

Display Widget

Tkinter Text widget shows chat history clearly and interactively.

Multithreaded I/O

Threads enable chat and game actions to run concurrently without lag.

Threading and Concurrency

1

Handle Multiple Clients

Separate threads allow simultaneous client connections effectively.

2

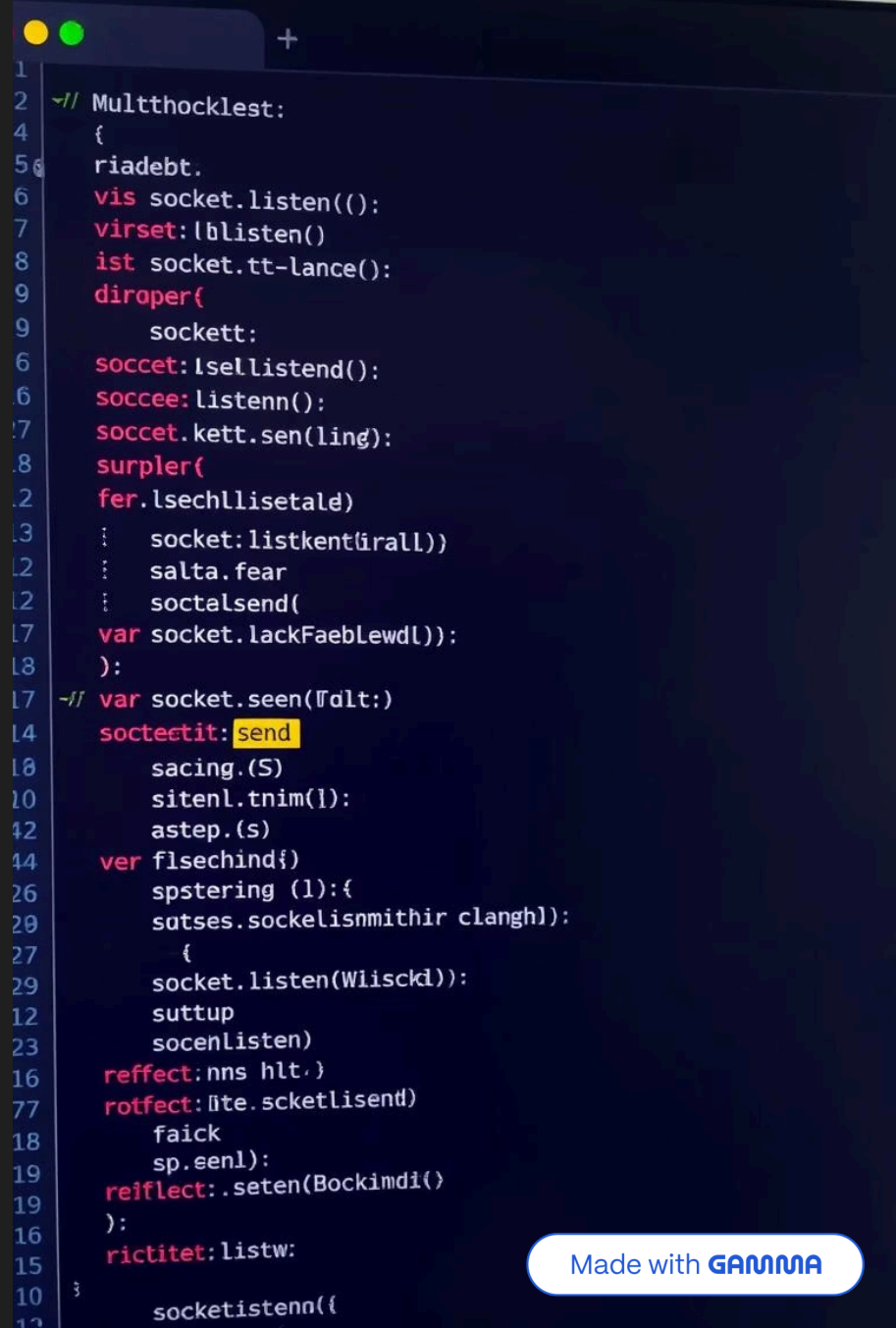
Dedicated Threads

Use distinct threads for listening, sending, and game logic processing.

3

Server with Log

The server has log which shows received and sent messages and helps in debugging.



Demonstration & Future Enhancements

Graphical UI Interface

Improve the User interface.

Friend System and Private Matches

Improve the friend system and allow private matching

More moves and Effects

Improve the variety of moves

Character Creation

Allow users to store and create multiple characters.