School of Computer Science College of Science
Fundamentals of Operating Systems

Final Project

Introduction

The aim of this project is to design a **Tic-tac-toe** game with server-client architecture. Tic-tac-toe, a classic and simple game for two players, is traditionally played on a 3x3 grid. Players take turns marking their distinct symbol, an 'X' or 'O', in the grid's cells, with the goal to place three of their marks in a horizontal, vertical, or diagonal row. In this project you have to implement 3x3, 4x4, 5x5 boards and each player connects the corresponding player with the same selection on the board size. (3x3 and 4x4 boards need 3 connected houses and 5x5 board need 4 connected houses)

Server

The server in this Tic-tac-toe project acts as the central hub for game logic and player coordination. It's responsible for managing the state of the game board, enforcing the rules, and ensuring the correct sequence of player turns in each board size. The server also handles the initiation of games, connecting two clients into a single match, and transmitting each move to the appropriate opponent. Through its network interface, it listens for incoming connections, allowing multiple pairs of players to engage in their own separate games concurrently, providing a scalable and responsive gaming experience.

Client

On the client side, the focus is on user interaction with the game. Each client runs a program that renders the Tic-tac-toe board and captures the player's moves. Once a move is made, it is sent to the server for validation. The client then waits for the server to respond with the updated game state, which could include the opponent's last move or the result of the game. The client software is designed to be lightweight, user-friendly, and able to handle network communications smoothly to provide a seamless gaming experience.

Tips

You need to implement a multi-threaded server-client system to service each client in the same time, if you do not use multi-threading, each player has to wait for another client to connect to the server and the game simply cannot be happened.

Bonus Part

Implementing a graphical user interface (GUI) using built-in libraries is optional and has bonus points based on how much user friendly your implementation is.

Notes

- Your code must be in Python.
- Add comments in every section of your code and explain it in details.
- Do not hesitate to connect your TAs about any possible questions.
- Provide a full report about the whole procedure of implementing this project and possible challenges that you might experience.
- Gather up all your files in a ZIP file and name it like given pattern.
- Pattern: LASTNAME_STUDENTNO_OSFINAL.ZIP