CSE 101 Project Description Fall 2021

Quarto Multinode

Introduction

In this project, our goal is to get a better understanding of python and working with sockets. For this purpose, we are going to write a two-player game that two users can play over a local network

This project consists of two nodes (clients, players) and a server. The server is responsible for managing the game and passing data between nodes as the players take their turn playing the game.

The two nodes connect to the server using sockets. (which we will discuss later).

There are two main points in this project.

1. writing a fully functional turn-based two-player game,
2. separating the game into two nodes, and connecting them to the server through sockets.

The Game

For the game part of this project, we are going to implement a game called Quarto.

The game is played on a 4×4 board. There are 16 unique pieces to play with, each of which is either:

- tall or short
- black or white
- square or circular
- hollow-top or solid-top

Players take turns choosing a piece which the other player must then place on the board. A player wins by placing a piece on the board which forms a horizontal, vertical, or diagonal row of four pieces, all of which have a common attribute (all short, all circular, etc.).

For a better understanding of the game, we encourage you to play it with your friends or with a computer online.

To implement the game, besides the rules of the game, you will need to implement a decent menu and UI. Although a TUI (Text User Interface) is enough for this project, obviously a better UI has bonus scores. The minimum required interface should allow players to choose a name and change it, connect to a server, play the game. In each round, they are shown the board of the game and the pieces and are asked to do their part (either choose a piece for their opponent or place a piece on the board). Then the winner and loser should be declared and an option to play again must be available to both nodes and a scoreboard should be implemented to store each player's name and how many matches he has lost and how many matches he has won . Of course, any addition and extra behavior could lead to a bonus.

TUI example:

```
What do you wish to do?
  1.Play
  2.Quit
User input: 2
See you next time :)
```

Sockets

We use sockets to connect each node (player) to the server. The server controls the flow of the game and then asks each node to play their turn. The data is transferred through socket connections. All together, sockets provide us with a way to transfer data between two ends.

To advantage this ability, we use the python library Socket-io. This library ables us to open socket connections and transfer data between the players (nodes, clients) and the server. For more information search online.

Just To Make Sure…

Just to make sure that we are on the same page on what the required features are to get the full marks:

- The game which has a proper server and client implementation.
- A scoreboard that shows how many matches each player has lost and how many matches has won.
- A working TUI which allows players to start a game, enter server address, choose a name, show the scoreboard, quit and of course play the game again and again.
- Proper structure and a clean code.

Final Thoughts

There are some points to mention here.

We will talk about this project immensely in a meeting later on. But it is best for you to get yourself familiarized with both the game Quarto and basic socket understanding. (the knowledge of sockets needed in this project is minimal).

There are lots of ways to improve and be creative here. So try to add things as you wish.

Your project should be clean. This both means that it should have a clear structure and a clear flow, and also to write your code in a clean manner. (we mentioned clean code in the previous weeks, and that your code should be modular, you should use descriptive variable and function names, your functions should be short and do only one thing, and …)

Good luck and have fun ;)