



SOFE4790 Distributed Systems (Fall 2024)

Assignment 1

Honor code: By submitting this assignment, I Mohammed Adnan Hashmi (100753115) affirm this is my own work, and I have not asked any of my fellow students or others for their source code or solutions to complete this assignment, and I have not offered my source code or solutions for this assignment to any of my fellow students.

Name: Mohammed Adnan Hashmi

Banner ID #: 100753115

1. Application idea

The application idea behind the assignment was to create a Jeopardy-style game where a local server would be created to host an array of questions ranging from three different categories that the players on the client side can answer. Any players can enter the client, which is recorded on the server side. This application/game aims to answer the most questions correctly and have a higher score than the other players connected.

2. Describe the two core functionalities

Two core functionalities of this application are its ability to use sockets for real-time communication between the server and client. It handles multiple client connections through the use of multithreading where a separate client handler thread handles each client that connects to the server. In addition to this, each player that joins must enter their username as a login functionality. These two core functionalities are the basis of the entire application where the user can select a category for the questions they would like to answer.

3. Describe the two novel features

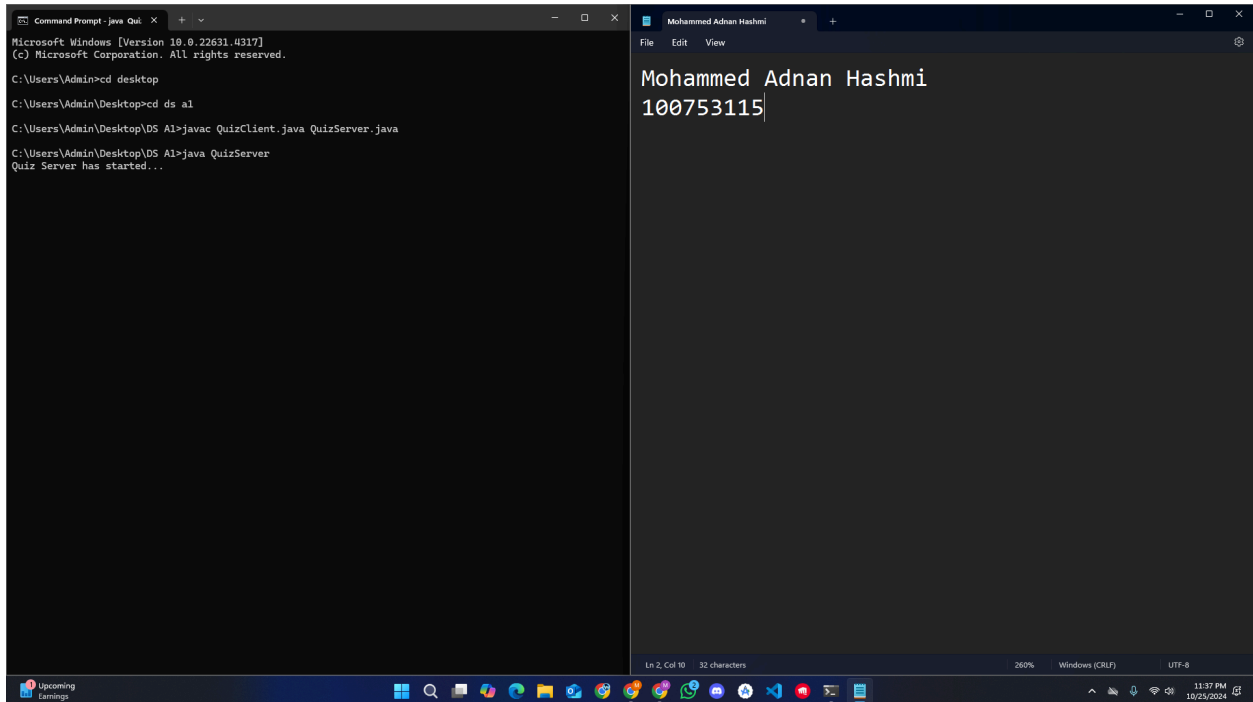
Two novelty features that were implemented within the application is real time score updates and time-based bonuses for fast response. The purpose of these features is to allow any clients that are connected to the server to have an additional layer of user experience and challenge to the game. Whenever a player enters in an answer, they are prompted with either a confirmation or rejection of their answer alongside their current score as well as anyone else connected to the server. This allows the player to keep a record of how many points they have and how many points their opponents have. In addition to the real-time score tracking, a time functionality was implemented which awarded bonus points to any player that was able to answer the question within 5 seconds of it being asked. These two functionalities were implemented by using a map to store the client's username and their current scores where the update scores method is called to update the map. Once the map was synchronized, a broadcast method was used to send the updated score to all the clients connected to the server in real-time. For the time bonus functionality, a bonus time constant was defined at the beginning of the code and set to 5 seconds alongside the points value of 5 points. Anytime a client enters an answer, the server records the time using a function that would retrieve the current time in milliseconds. It would then compare the two times and either award or skip over the bonus given if the player met the requirement.

4. Challenges and solutions

Some challenges that were faced within the assignment were initializing the port from the client side to the server side as any time I would run the client side, it would refuse to the server side. This was a quick fix simply testing and changing the port number was required to fix this issue. In addition to the client not connecting, another challenge encountered was the loops created not working correctly where they would duplicate messages twice constantly and would either crash prevent the user from inputting any more values. This was resolved by changing the loop condition to loop for conditions using specific variables rather than leaving it open-ended such as `while(true)`.

5. Testing

The code is working but I am getting an error of the client refusing to connect to the server where it didnt do that before. I have tested the entire code multiple times but it is not connecting now. Nothing in terms of the code has changed at all. The issue I mentioned earlier has happened before and all I did was change the port number which resolved the issue but doing it again is no longer working. A screenshot has been provided to showcase the issue



Compiling the server and client file within cmd and running the server. The server should output a message saying it has started. This is done successfully.

