Project Report

Ahmet UYAR

Multiplayer Game with Server-Client Communication

ReadMe Section:

1. Execution Instructions:

- To run the project, compile and execute the Server and Client Java programs separately.
- When prompted, enter the specified port number for both the Server and Client programs.
- Follow the on-screen instructions to play the game.

Detailed Report:

1. Introduction:

This project offers a practical introduction to computer networks by creating a straightforward multiplayer game that showcases the exchange of data between a Server and a Client using the TCP protocol. It provides a hands-on understanding of essential concepts such as socket programming, real-time interactions, and the significance of reliable TCP-based client-server communication.

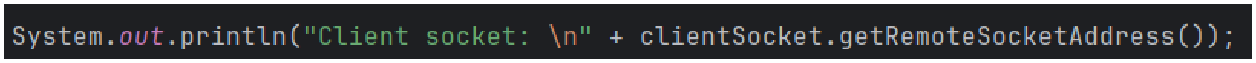
2. Code Overview:

Server.java: Server.java is basically the server, it starts the server to the user specified port and then accepts the client socket. After a successful connection, the server sends information to the user and receives it while setting up the game and running it.

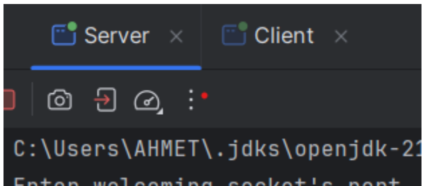
Client.java: Client.java is basically the client side of the game. The client connects to the server via the socket using the port number specified by the user. After connecting to the server, client.java is only responsible for sending and receiving data, nothing else.

3. Server Execution:

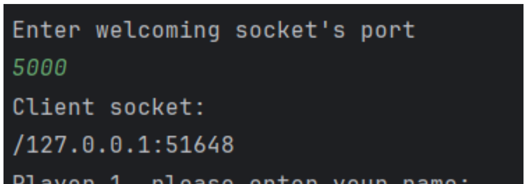
Below you can see the server console in execution. First, the user specifies the port then the client socket is shown using the getRemoteAddress() function. Then the server requests the name of the user. Then the waiting for guess messages are sent and each round winner is announced by the server for both parties. At the end, the overall winner is announced from the server for both sides.



```
System.out.println("Client socket: \n" + clientSocket.getRemoteSocketAddress());
```

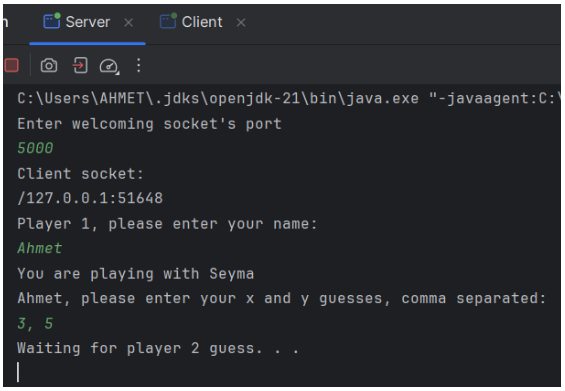


```
C:\Users\AHMET\.jdks\openjdk-21
Enter welcoming socket's port
```



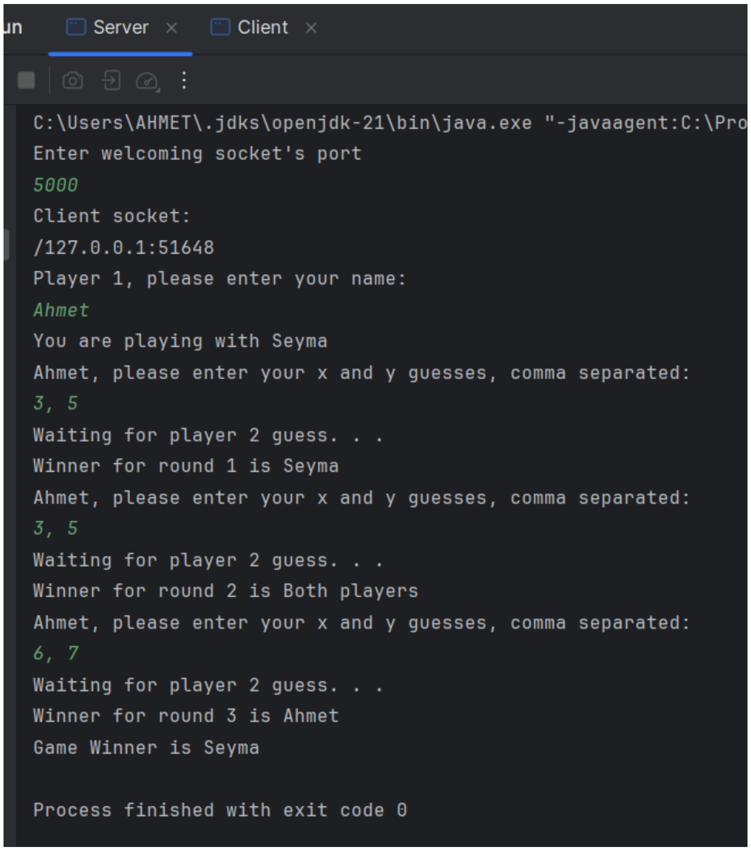
```
Enter welcoming socket's port
5000
Client socket:
/127.0.0.1:51648
Player 1, please enter your name:
```

Here you can see that the server is awaiting for client guesses for x, y values.



```
C:\Users\AHMET\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\
Enter welcoming socket's port
5000
Client socket:
/127.0.0.1:51648
Player 1, please enter your name:
Ahmet
You are playing with Seyma
Ahmet, please enter your x and y guesses, comma separated:
3, 5
Waiting for player 2 guess. . .
```

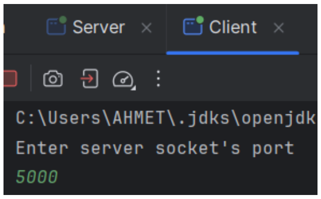
Below you can see the entire server-side execution.



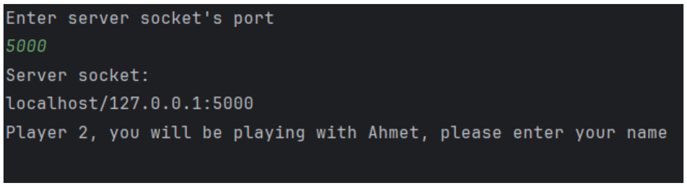
```
C:\Users\AHMET\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Pro
Enter welcoming socket's port
5000
Client socket:
/127.0.0.1:51648
Player 1, please enter your name:
Ahmet
You are playing with Seyma
Ahmet, please enter your x and y guesses, comma separated:
3, 5
Waiting for player 2 guess. . .
Winner for round 1 is Seyma
Ahmet, please enter your x and y guesses, comma separated:
3, 5
Waiting for player 2 guess. . .
Winner for round 2 is Both players
Ahmet, please enter your x and y guesses, comma separated:
6, 7
Waiting for player 2 guess. . .
Winner for round 3 is Ahmet
Game Winner is Seyma

Process finished with exit code 0
```

4. Client Execution:

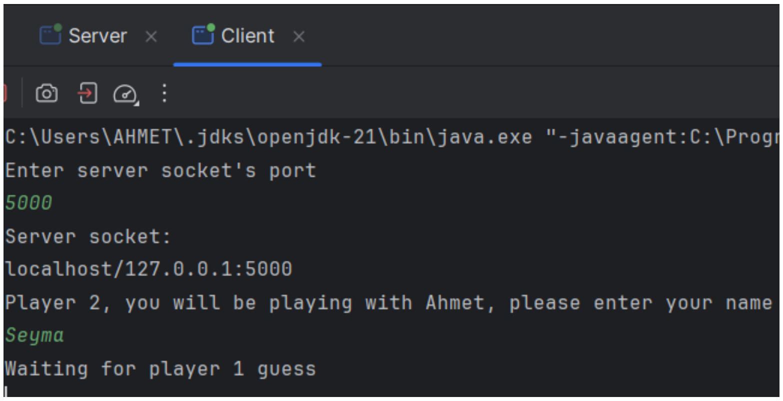
Client first enters the server port, then she awaits for the name request and then enters her name. After the client successfully sends her name to the server, the server sends a message that player 1 is guessing. Each round both parties are informed of the round winner. At the end, the overall winner is announced from the server for both sides.



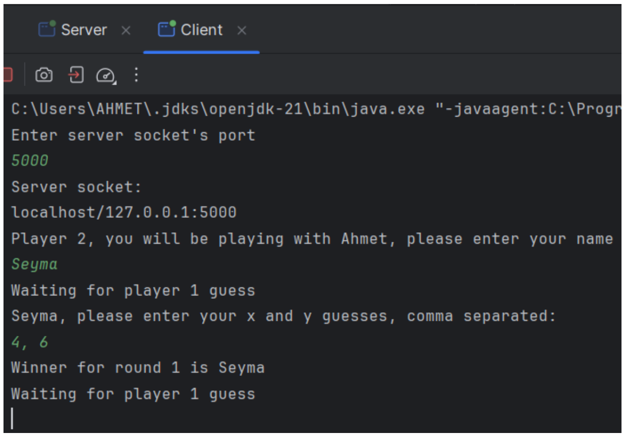
```
C:\Users\AHMET\.jdks\openjdk
Enter server socket's port
5000
```



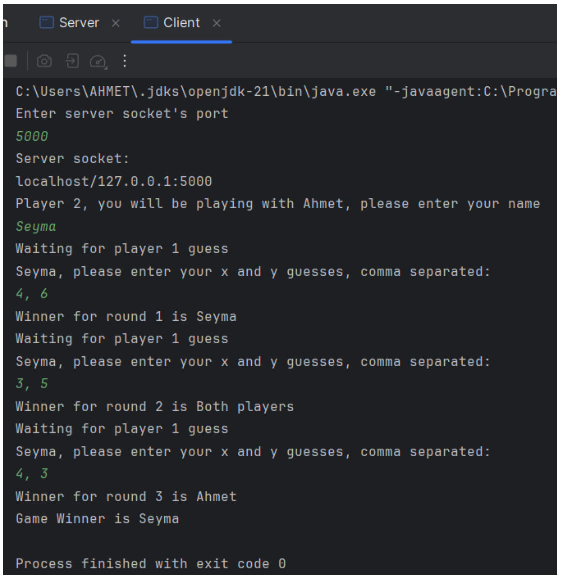
```
Enter server socket's port
5000
Server socket:
localhost/127.0.0.1:5000
Player 2, you will be playing with Ahmet, please enter your name
```



```
C:\Users\AHMET\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Progr
Enter server socket's port
5000
Server socket:
localhost/127.0.0.1:5000
Player 2, you will be playing with Ahmet, please enter your name
Seyma
Waiting for player 1 guess
```



```
C:\Users\AHMET\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Progr
Enter server socket's port
5000
Server socket:
localhost/127.0.0.1:5000
Player 2, you will be playing with Ahmet, please enter your name
Seyma
Waiting for player 1 guess
Seyma, please enter your x and y guesses, comma separated:
4, 6
Winner for round 1 is Seyma
Waiting for player 1 guess
```



```
C:\Users\AHMET\.jdks\openjdk-21\bin\java.exe "-javaagent:C:\Progra
Enter server socket's port
5000
Server socket:
localhost/127.0.0.1:5000
Player 2, you will be playing with Ahmet, please enter your name
Seyma
Waiting for player 1 guess
Seyma, please enter your x and y guesses, comma separated:
4, 6
Winner for round 1 is Seyma
Waiting for player 1 guess
Seyma, please enter your x and y guesses, comma separated:
3, 5
Winner for round 2 is Both players
Waiting for player 1 guess
Seyma, please enter your x and y guesses, comma separated:
4, 3
Winner for round 3 is Ahmet
Game Winner is Seyma

Process finished with exit code 0
```

5. Results:

Overall the project objectives were successfully achieved. Please see the last snapshots of both server and client