KOÇ ÜNİVERSİTESİ
COMPUTER ENGINEERING

COMP416: Computer Networks
Network Programming Project
Client-Server Game Application via TCP Sockets

Due: October 20, 11:59 pm (Late submissions will not be accepted).

Submission of the project deliverables is via Blackboard.

This is an individual project. You are not allowed to share your codes with each other.

Network Programming Project provides relevant hands-on experience with TCP socket programming. It includes programming a game between a server as player 1 and a client as player 2, which can communicate with each other on a local machine through a TCP socket. Through this work, you will familiarize yourself with the Java 'Socket' class and some of the methods essential to socket programming.

Sockets:

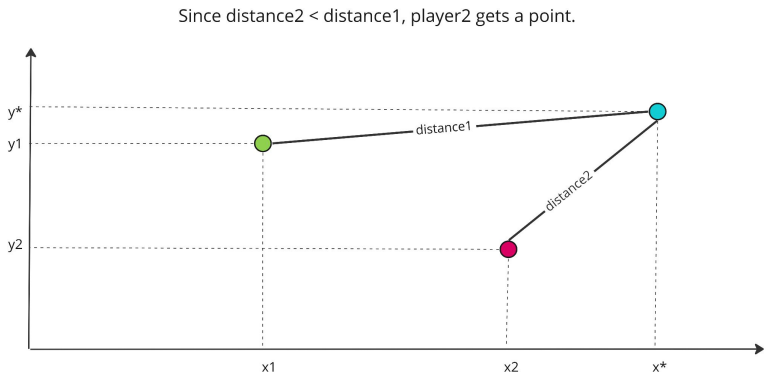
Sockets provide a means of communication between two machines. Each socket comprises the host address and port. For this project, both the client and the server will be executed on the same machine.

(Successful attempts to execute the game client on a machine and the game server on a different machine will result in bonus points awarded).

Game Description

The game application is named Hot and Cold and is played by two players. **Player 1 will be the server and player 2 will be the client.** Player 1, as a server, opens a welcoming socket and waits for the client to connect. After the client connects, the server will generate two random values, x and y. Each random value will be between 0 and 255, inclusive. The server will ask player 1 to enter his name and then send a message to the client to enter his name as well. The user at the server, player one, will enter a guess value for x and y and then send a message to the client (player 2) to guess x and y using sockets. The server then calculates the distances between the randomly generated values and the guesses and adds one point to the player with the closest guess. The game is played for three rounds and at the end, the server announces the winner and closes the connection.

The distance is calculated using the Euclidean distance.



Sample Run of the game:

Server	**Client**
Enter welcoming socket's port	
10000	Enter server socket's port
Waiting for client to connect...	10000
Client socket: /127.0.0.1:9599	Server socket:
Player 1, please enter your name:	localhost/127.0.0.1:10000
Abdulrezzak	Player 2, you will be playing with
Waiting for player 2 name...	Abdulrezzak, please enter your name:
You are playing with Aylanur	Aylanur
Abdulrezzak, please enter your x and	Waiting for player 1 guess...
y guesses, comma separated.	Aylanur, please enter your x and y
124, 124	guesses, comma separated.
Waiting for player 2 guess...	124, 124
Winner for round 1 is Both players	Winner for round 1 is Both players
Abdulrezzak, please enter your x and	Waiting for player 1 guess...
y guesses, comma separated.	Aylanur, please enter your x and y
100, 199	guesses, comma separated.
Waiting for player 2 guess...	220, 240
Winner for round 2 is Abdulrezzak	Winner for round 2 is Abdulrezzak
Abdulrezzak, please enter your x and	Waiting for player 1 guess...
y guesses, comma separated.	Aylanur, please enter your x and y
120, 150	guesses, comma separated.
Waiting for player 2 guess...	175, 170
Winner for round 3 is Abdulrezzak	Winner for round 3 is Abdulrezzak
Game Winner is Abdulrezzak	Game Winner is Abdulrezzak

This video shows how the game is played.

Project Objectives

You are required to develop the previously described game in **Java**, where the connection between the server and the client is handled by sockets.

You can refer to the documentation here for more information on the methods associated with the Socket Class.

The following are the main requirements for this project:

1. Code a game client and server-side capable of sending each other messages through **TCP sockets** as described in the Game Description section. **Make sure that the server is handling the game logic and the client only receives and sends data**.
2. The port number at the server and client **should not be hard-coded**; instead, it should be capable of receiving the port as an argument at the time of execution.
3. Once the connection has been established, both sides should display the complete socket address of the other.
4. Both the client and the server should keep the communication running until the end of the game.

Project deliverables

You should submit your source code and project report (in a single .rar or .zip file) via Blackboard.

Report

The report should at max be 3 pages long and should start with a concise ReadMe section to provide instructions for the execution of your code. Following this, you should provide an overview of the programming and attach the **snapshots** of the execution results depicting the project objectives. The report must include the snapshots for the following:

1. The advertised port by the server being provided as an argument at execution
2. Both sides showing the socket address of the other side once the connection has been established.
3. The game between the server and the client.

Source Code

A .zip or .rar file that contains your implementation in a single Eclipse or Intellij IDEA project or any other IDE.

The report is an important part of your project presentation, which should be submitted as both a .pdf and Word file. Your report should show the step-by-step server-client communication initiation till termination through snippets. The report acts as proof of work for you to assert your contributions to this project. Everyone who reads your report should reproduce the parts we asked to document without hard effort and any other external resource. For codes, you should be taking screenshots instead of copy/pasting the direct code. Strictly avoid replicating the code as a whole in the report or leaving code unexplained.

Good Luck!