Alp Kural - 71959
ME & CE

Comp 416 – Network Programming Project

## ReadMe Section

To run the project code, navigate to the "Hot-Cold Game" folder and locate the "bin" directory. Within this directory, you'll find the compiled bytecode of the project classes, which were generated from the Java source code using the Java compiler (Javac). Using JVM interpreter, you can run the bytecodes present in the stated directory.

Welcoming port of the server should be prepared before any connection request of the client, thus, the player one should enter the port number before the player two to ensure program to run successfully since from the port number given from the player one, server prepares the welcoming port of the server and with the port number from the player two client sends connection request to the server.

## Commented Source Code

The source code of the server and client classes are commented to explain step by step how to achieve the project objectives.

EchoClient Source Code:

```java
import java.io.*;
import java.net.*;

// Second Player
public class EchoClient {
    public static void main(String[] args) throws IOException {

        // Define the hostname for the server
        String host = "localhost";

        // Create a BufferedReader to read input from the console
        BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));

        // Prompt the user to enter the port for the server socket
        System.out.println("Enter server socket's port");

        // Read the port number from the console and parse it to an integer
        int port = Integer.parseInt(stdIn.readLine());

        // Create a socket and connect to the server using the specified host and port
        Socket echoSocket = new Socket(host, port);

        // Display information about the server socket
        System.out.println("Server socket: ");
        System.out.println(echoSocket.getRemoteSocketAddress());

        // Create input and output streams for communication with the server
        PrintWriter out = new PrintWriter(echoSocket.getOutputStream(), true);
        BufferedReader in = new BufferedReader
                (new InputStreamReader(echoSocket.getInputStream()));

        // Receive the first player's name from the server
        String firstPlayerName = in.readLine();

        // Prompt the second player to enter their name and show the first player's name
        System.out.println("Player 2, you will be playing with "
        + firstPlayerName + ", please enter your name:");

        // Read the name of the second player from the console and send it to the server
        String secondPlayerName = stdIn.readLine();
        out.println(secondPlayerName);

        // Play the game for 3 rounds
        for (int ii = 0; ii < 3; ii++) {

            // Wait for the signal from the server to allow user to make a guess
            System.out.println("Waiting for player 1 guess...");
            in.readLine();

            // Prompt the second player to enter their x and y guesses
            System.out.println(secondPlayerName + ", please enter your x and y guesses,"
                    + " comma seperated.");

            // Read the guess from the console and send the it to the server
            out.println(stdIn.readLine());

            // Recieve the round winner feedback from the server and print to the console
            System.out.println(in.readLine());
        }

        // Display the final game result received from the server
        System.out.println(in.readLine());

        // Close the input and output streams and the socket
        in.close();
        out.close();
        stdIn.close();
        echoSocket.close();
    }
}
```

Alp Kural - 71959
ME & CE

# EchoServer Source Code:

```java
import java.io.*;
import java.net.*;
import java.util.Random;

// First Player
public class EchoServer {
    public static void main(String[] args) throws IOException {

        // Create a BufferedReader to read input from the console
        BufferedReader stdIn = new BufferedReader(new InputStreamReader(System.in));

        // Prompt the user to enter the port for the welcoming socket
        System.out.println("Enter welcoming socket's port");

        // Read the socker number from the console and parse it to integer
        int port = Integer.parseInt(stdIn.readLine());

        // Create a ServerSocket and bind it to the specified port
        ServerSocket serverSocket = new ServerSocket(port);
        System.out.println("Waiting for client to connect...");

        // Accept a connection from a client
        Socket clientSocket = serverSocket.accept();

        // Print the client socket adress to the console
        System.out.println("Client socket: " + clientSocket.getRemoteSocketAddress());

        // Prompt the first player to enter their name and read from the console
        System.out.println("Player 1, please enter your name:");
        String firstPlayerName = stdIn.readLine();

        // Create input and output streams for communication with the client
        PrintWriter out = new PrintWriter(clientSocket.getOutputStream(), true);
        BufferedReader in = new BufferedReader
                (new InputStreamReader(clientSocket.getInputStream()));

        // Send the first player's name to the client
        out.println(firstPlayerName);

        // Prompt to indicate waiting for the second player's name
        System.out.println("Waiting for player 2 name...");
```

```java
        // Receive the second player's name from the client and display in console
        String secondPlayerName = in.readLine();
        System.out.println("You are playing with " + secondPlayerName);

        // Initializing the scores for each player
        int playerOneScore = 0;
        int playerTwoScore = 0;

        // Play the game for 3 rounds
        for (int ii = 0; ii < 3; ii++) {
            // Generate random target coordinates for the current round
            Random random = new Random();
            int[] target = new int[2];
            target[0] = random.nextInt(256);
            target[1] = random.nextInt(256);

            // Prompt the first player to enter their x and y guesses
            System.out.println(firstPlayerName + ", please enter your x and y guesses,"
                    + " comma seperated.");

            // Parse the player's guesses into integer array
            int[] playerOneGuess = guessParser(stdIn.readLine());

            // Signaling client to allow user to make a guess
            out.println();

            // Receive the second player's guess and parse the guess into integer array
            System.out.println("Waiting for player 2 guess...");
            int[] playerTwoGuess = guessParser(in.readLine());

            // Calculate the Euclidean distance between target and player's guesses
            double distance1 = euclidianDistance(target, playerOneGuess);
            double distance2 = euclidianDistance(target, playerTwoGuess);
```

```java

            // Decide the winner of the round and increment their score by one
            // unless it is a draw
            String winnerName;
            if (distance1 < distance2) {
                winnerName = firstPlayerName;
                playerOneScore++;
            }
            else if (distance1 == distance2) {
                winnerName = "Both players";
            }
            else {
                winnerName = secondPlayerName;
                playerTwoScore++;
            }

            // Display the winner of the current round
            String levelPrompt = "Winner for round " + (ii + 1) + " is " + winnerName;
            System.out.println(levelPrompt);

            // Send the winner of the current round information to the client
            out.println(levelPrompt);
        }

        // Determine and display the winner of the game
        String endPrompt;
        if (playerOneScore > playerTwoScore) {
            endPrompt = "Game Winner is" + firstPlayerName;
        }
        else if (playerOneScore == playerTwoScore) {
            endPrompt = "Game is tied";
        }
        else {
            endPrompt = "Game Winner is" + secondPlayerName;
        }
        System.out.println(endPrompt);

        // Send the winner of the game information to the client
        out.println(endPrompt);
```

```java

        // Close the streams and sockets
        in.close();
        out.close();
        stdIn.close();
        clientSocket.close();
        serverSocket.close();
    }

    // Calculate the Euclidean distance between two points
    public static double euclidianDistance(int[] target, int[] guess) {
        double d0 = target[0] - guess[0];
        double d1 = target[1] - guess[1];

        return Math.sqrt(d0 * d0 + d1 * d1);
    }

    // Parse a comma-separated string of two integers into an int array
    public static int[] guessParser(String line) {
        String[] guess = line.split(", ");
        int[] array = new int[2];
        array[0] = Integer.parseInt(guess[0]);
        array[1] = Integer.parseInt(guess[1]);
        return array;
    }
}
```

Alp Kural - 71959
ME & CE

## Runtime Program Snapshots

```
alpku@AlpLegionY540 MINGW64 ~/Desktop/Hot-Cold Game/bin
$ java EchoServer
Enter welcoming socket's port
10000
Waiting for client to connect...
Client socket: /127.0.0.1:64480
Player 1, please enter your name:
Alp
Waiting for player 2 name...
You are playing with Serkan
Alp, please enter your x and y guesses, comma seperated.
123, 123
Waiting for player 2 guess...
Winner for round 1 is Both players
Alp, please enter your x and y guesses, comma seperated.
12, 12
Waiting for player 2 guess...
Winner for round 2 is Serkan
Alp, please enter your x and y guesses, comma seperated.
212, 212
Waiting for player 2 guess...
Winner for round 3 is Serkan
Game Winner is Serkan
```

Echo Server

```
alpku@AlpLegionY540 MINGW64 ~/Desktop/Hot-Cold Game/bin
$ java EchoClient
Enter server socket's port
10000
Server socket:
localhost/127.0.0.1:10000
Player 2, you will be playing with Alp, please enter your name:
Serkan
Waiting for player 1 guess...
Serkan, please enter your x and y guesses, comma seperated.
123, 123
Winner for round 1 is Both players
Waiting for player 1 guess...
Serkan, please enter your x and y guesses, comma seperated.
123, 123
Winner for round 2 is Serkan
Waiting for player 1 guess...
Serkan, please enter your x and y guesses, comma seperated.
133, 132
Winner for round 3 is Serkan
Game Winner is Serkan
```

Echo Client