Project Report

(King-Thief-Minister-Police Role Playing Game)

It is a **CLI based** simulation of **King, thief, minister** and **police** game. It is a type of **role playing** game that is a popular pastime in India. It involves four players each player takes up the role of either the king, thief, minister and police and the **minister has to guess and identity the thief**.

This folder contains programs to run a 4-player find-the-thief game. It contains both the server and client programs. Clients will be able to play this game with each other over the network using the server. Each player is represented by a client and a separate thread at server will communicating with each player.

Application Layer Protocol -

- This game uses TCP to pass information between server and client.
- Player.java will handle the game play of each player in different threads.

GameServer Class

GameServer class is the central server and is the controller that controls the four player threads. This controller is passed to Player Class. In the program, we have set the port number to 5000 and number of rounds to 5.

GameServer class defines points for each of the four defined game characters. It also randomly assigns the game characters to each player and sets the '**whoIsKing**' variable to identify the king and every player get to know which player is king so that he will be out of the choices available to the minister.

Methods of GameServer Class -

1. `public void execute()`

The execute() function accepts the four clients(player 0,1,2,3) connection and then start one thread for each player. All the threads remains suspended until we have all the four players onboard to play the game.

2. **`public randomizeCharacter()`**

   This function takes an array of game characters and shuffles the array in each round so that each player is assigned a random game character in each round.

3. **`public void resetAll()`**

   This function basically resets the game. It randomizes the characters array, gets the new king after randomizing and resets the scores of each player to 0, resets the winner the minister's guess.

Player Class

```
public Player(Socket s, GameServer t, int playerNumber)
```

Player class handles the game play of each player in different threads. 's' is the socket that is created between server and the four players. 't' is the game controller, is an object of GameServer Class. 'playerNumber' is either 0,1,2 or 3 for each player.

The game will run for five rounds. We have used switch case to define the task of each player. It is the task of player '0' or first player to compute the score after each round and get the maximum score to identify the winner of the game.

Important class data members -

1. **`private GameServer control`**

   Control is the object of the GameServer class as it needs to access the status of each player and their scores.

GameClient Class

This class represents the client or the player. It sends the connection request to the GameServer. On completion of connection the player will be assigned a game character by the server. If the assigned game character is minister it have to guess the thief's id that can be 0,1,2 or 3 and excluding the king's id and its own id. At the end the final scores are displayed to each client and the winner is declared to all.

build commands

```
$ git clone https://github.com/shivamHCU/Find-the-thief-Multithreaded.git
$ cd game
$ javac *.java
```

Then, open two additional command line windows in the same directory (a total of five) and run the following in the first:

```
$ java GameServer
```

In each of the other windows, run the following:

```
$ java GameClient <IP Address>
```

The server window should output the following it will launch the game when any 4 players are connected to the server.

```
Server awaiting connections...
 1 out of 4 is connected
    Waiting for another players
 2 out of 4 is connected
    Waiting for another players
 3 out of 4 is connected
    Waiting for another players
 4 out of 4 is connected
All Player is connected! Now we can start the game! 
Game Started...
```

At which point the game can begin. At the end the detailed score and the result will be shown to you as given below.

```
SERVER : You are Winner !!.
+-------+-------+-------+-------+
|  P0   |  P1   |  P2   |  P3   |
+-------+-------+-------+-------+
| 800   | 500   | 0     | 1000  |
| 1000  | 500   | 800   | 0     |
| 0     | 800   | 1000  | 500   |
| 800   | 0     | 1000  | 500   |
| 1000  | 800   | 500   | 0     |
+-------+-------+-------+-------+
| 3600  | 2600  | 3300  | 2000  |
+-------+-------+-------+-------+
```

Submitted By –

1. Shivam Gangwar (17MCMC49)
2. Dipali Gupta (17MCMC63)