

Oluwole Aibinu  
S3479719

12345678910



**USER'S**

**MANUAL**

**Guessing Game**  
***Network Programming***

**Prepared by: Oluwole Aibinu**  
**Student id: S3479719**

### **Requirements**

- Java SE 1.8
- netprog1 and 2 servers

### **Program Structure**

#### Single Player

The server generates a number between 0 and 9. The client's then guess the number (up to a maximum of 4 guesses).

- If the client guesses the number correctly, then the server shows "Congratulation".
- If the client fails to guess the number, for each incorrect guess, the client gets a clue:
- After four incorrect guesses, the server then shows the correct guess and prompt user to either continue playing or quit the game

Type of Socket Used: TCP for client to server connection.

#### Multiplayer Player

The server maintains a lobby queue. Clients register with the server (using their first name) to be added to the lobby queue before they can play. Clients can register at any time.

The game is played in rounds. At the start of each round, the server takes the first three clients from the lobby (or all the clients if there are less than three clients in the lobby) and starts the round.

- First the server announces the name of 3 participants.
- Each player can guess at any time (with their number of guesses tracked by the server).

Once all plays have either:

- Correctly guessed the generated number,
- Reached their maximum guess of 4,

The server announces to all the clients the number of guesses for each client.

- The players that choose to play again are added back into the end of the lobby queue, and the process repeats with a new round.

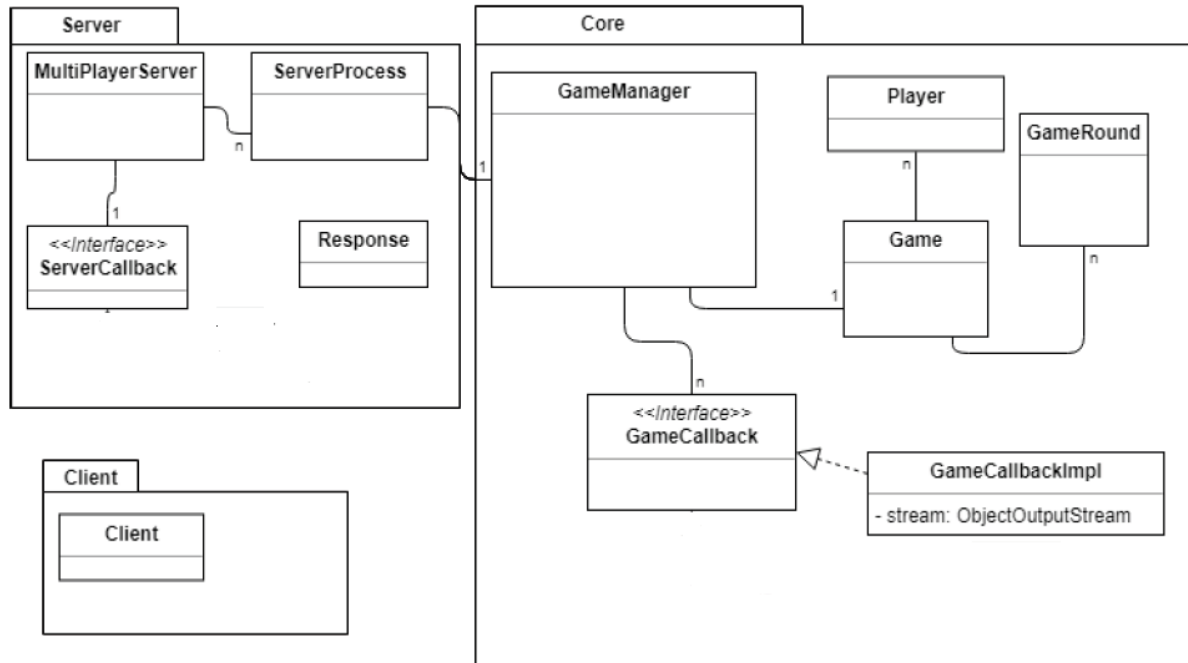
Type of Socket Used (TCP and UDP): TCP is connection from client to server while UDP protocol was used as the different users' needs to be broadcast information when some other player wins the game, or any new players join the game.

### **Overview of files**

- Core - contains code to manage the game state
  - Game.java - the state of the game is reflected in this class
  - GameRound.java - a game can have multiple rounds.
  - Player.java - class that represents the player
  - GameManager.java - model class that bridges the game to the GameCallback
  - GameCallback.java - contains series of game event methods (eg. onGuessAdded() )
  - GameCallbackImpl.java - used to send replies to the client based on events
- Server - contains code to manage server processes
  - MultiPlayerServer/SinglePlayerServer.java - contains the main() method
  - ServerProcess.java - main controller class. Each client will have 1 instance
  - ServerCallback.java - has network related event (eg. onClientConnected() )
  - Response.java - Wraps server's messages/commands and sent to clients as object
- Client
  - Client.java - contains the main() method.

### Overview class diagram

Class Diagram is similar for single and Multiplayer implementation.



### How to Launch

Available Commands

Go up to the path where the folder TaskASinglePlayer and TaskAMultiPlayer are present and run the below make commands to execute:

#### Single-player version

- make singleplayer-server (to start the server)

- make singleplayer-client (to start the client)

By default, client connects to port 61995 and to server ip address of 10.102.128.22

Alternatively, client can specify port number and server address he wants to connect to by running the following command

- make singleplayer-client <server ipaddress> <port number>

#### Multi-player version

- make multiplayer-server (to start the server)

- make multiplayer-client (to start the client)

By default, client connects to port 61995 and to server ip address of 10.102.128.22

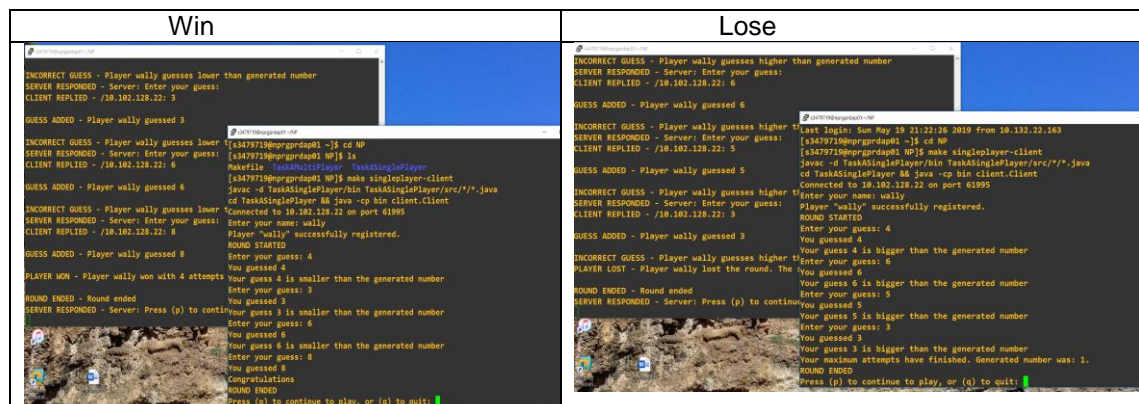
Alternatively, other clients can specify port number and server address he wants to connect to by running the following command

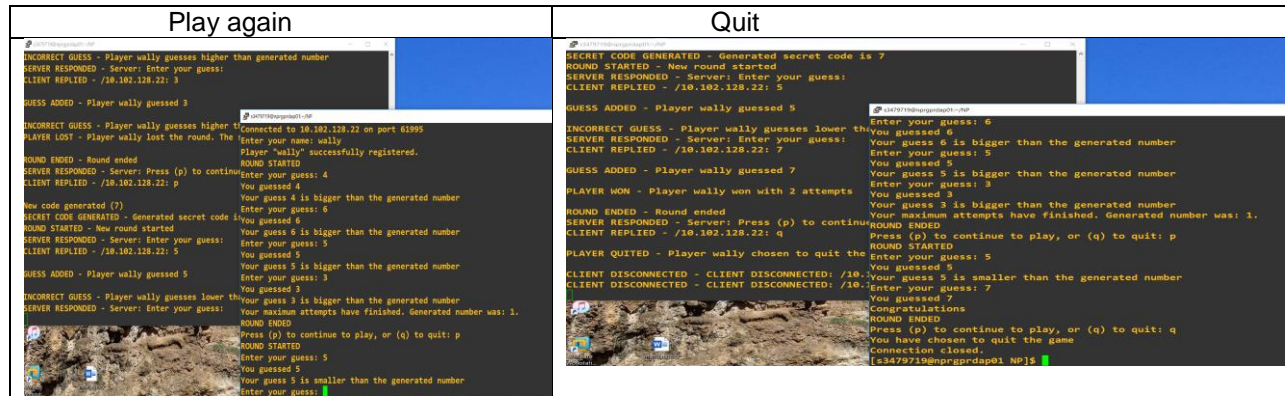
- make multiplayer-client <server ipaddress> <port number>

### Screenshot for every step

### Single player

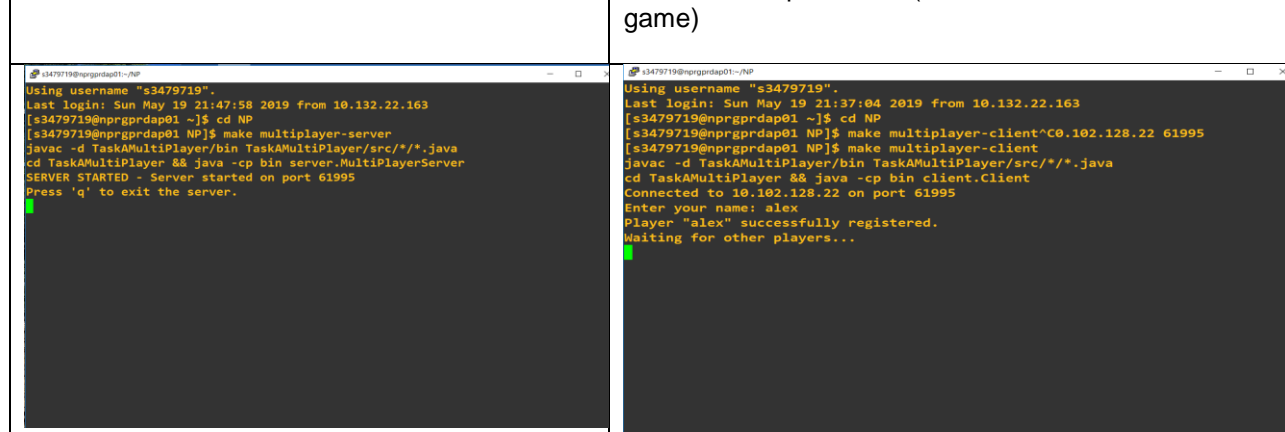
Open to terminal windows one for server and one for client



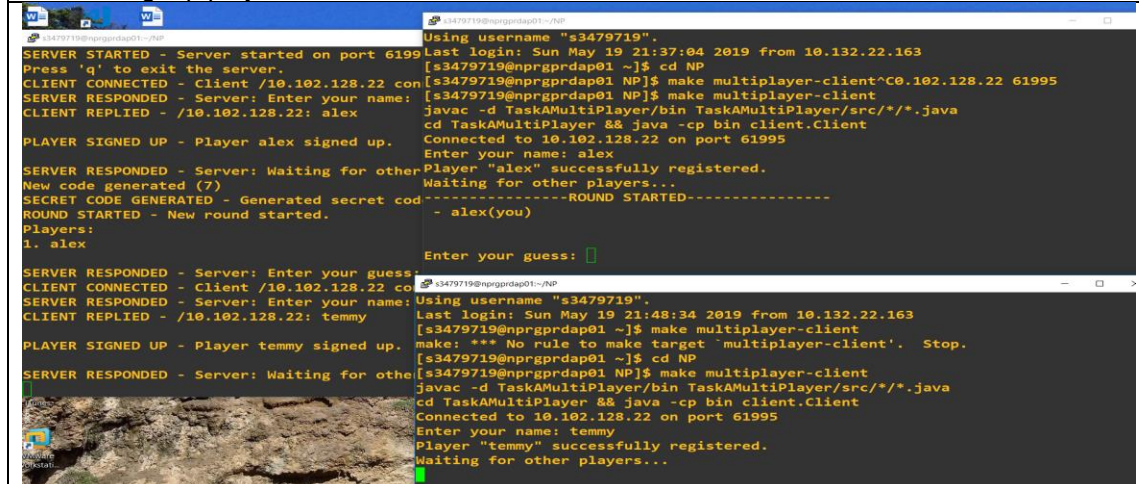


## Multiplayer

### Connect server



### Signup players





```
Play game

SERVER RESPONDED - Server: Enter your guess:
CLIENT REPLIED - /10.102.128.22: 1

GUESS ADDED - Player henry guessed 1

INCORRECT GUESS - Player henry guesses lower than generated
number
SERVER RESPONDED - Server: Enter your guess:
CLIENT REPLIED - /10.102.128.22: 4

GUESS ADDED - Player henry guessed 4

INCORRECT GUESS - Player henry guesses lower than generated
number

[s3479719@nprgrdp01 ~]$ cd NP
[s3479719@nprgrdp01 NP]$ make multiplayer-client
javac -d TaskAMultiPlayer/bin TaskAMultiPlayer/src/*.java
cd TaskAMultiPlayer && java -cp bin client.Client
Connected to 10.102.128.22 on port 61995
Enter your name: temmy
Player "temmy" successfully registered.
Waiting for other players...
-----ROUND STARTED-----
- alex
- alex(you)

Enter your guess: 4
You guessed 4
-----ROUND STARTED-----
- alex
- temmy

Enter your guess: 3
You guessed 3
Your guess 3 is smaller than the generated number
Enter your guess: 1
You guessed 1
Your guess 1 is smaller than the generated number
Enter your guess: 4
You guessed 4
Your guess 4 is smaller than the generated number
Enter your guess: 9
You guessed 9
Your guess 9 is bigger than the generated number
You lose. Answer is 7
You lose. Answer is 7
```

### Show the game winner at end of round

```
SERVER RESPONDED - Server: Press (p) to continue to play, or
r (q) to quit:
SERVER RESPONDED - Server: Press (p) to continue to play, or
r (q) to quit:
ROUND ENDED - Round ended.
WINNERS:
1. temmy (1 guesses)
2. alex (3 guesses)
LOSERS:
- henry
FORFEITED:
* There are no players forfeited for this round *

SERVER RESPONDED - Server: Press (p) to continue to play, or
r (q) to quit:
Enter your guess: 7
You guessed 7
YOU WIN. Number of guesses made: 1
Wait for other players to finish...
-----ROUND ENDED-----
WINNERS:
1. temmy (1 guesses)
2. alex (3 guesses)
LOSERS:
- henry
FORFEITED:
* There are no players forfeited for this round *

Press (p) to continue to play, or (q) to quit: ERROR: Pl
ease enter 'p' or 'q'.

Enter your guess: 4
You guessed 4
Your guess 4 is smaller than the generated number
Enter your guess: 6
You guessed 6
Your guess 6 is smaller than the generated number
Enter your guess: 7
You guessed 7
YOU WIN. Number of guesses made: 3
Wait for other players to finish...
-----ROUND ENDED-----
WINNERS:
1. temmy (1 guesses)
2. alex (3 guesses)
LOSERS:
- henry
FORFEITED:
* There are no players forfeited for this round *
```

## Logs

All game and communication logs are recorded for singleplayer and multiplayer respectively.  
To view, use the following command  
-vi communications.log (shows logs of all server to client communication)  
-vi game.log (shows logs of all the game activities)