BOUN SWE

SWE 544 – 2015 Fall

Project 2

Sinan Can SÜRÜCÜ

12/1/2016

Contents

Development Process

I try to add my notes in every commit but I will explain the development process with the steps that I followed.

1) Basic client and server functions are created.
2) Protocol rules and needed functions are created that are using these rules.
3) Ticket and User classes are created for using these as objects on the server side.
4) Tickets with 5x3 numbers are created randomly for every user.
5) Picking number randomly is implemented, server pick up a number randomly between 1-100 and announce it to all connected users.
6) Broadcasting message is implemented for the server side to announce messages to all users.
7) Server and client has two threads each, for incoming and outgoing messages.
8) Server side has been redesigned, with every connection server creates a thread.
9) Server assigns a ticket to users, and clients now available to show these tickets to users.
10) UI fixes has been made, Users can select the numbers and selected numbers will be seen differently.
11) Cinko validation and invalid 3 cinko request will be resulted as banning to user.
12) Game winning condition and announcement are added.
13) Login with username now mandatory, this username will be shown to the other users in the game.
14) Current Cinko number will be shown under username to every user.
15) Final tests and bugfixes are finished.

Commit Notes

1) Commit:dbab848 04/01/2016
- Repository has been created and first commit has been tried with a readme file

2) Commit:9238c10 04/01/2016
- Requirements and Design document has been added to the repository

3) Commit:a05b709 08/01/2016
- Server - Client basic functions and protocol interfaces are defined

4) Commit:fdce351 08/01/2016
- UI design completed and related files are added to the repo

5) Commit:0f04003 11/01/2016
- Server side Ticket and User classes are defined, when a client connects Server creates a ticket for the user with random number between 1-99.

6) Commit:c60aa36 11/01/2016
- Server is picking up number from the numbers list and returns it. Client side command is available now, for testing protocol rules

7) Commit:68f350a 12/01/2016

- Broadcast message in server has been defined. Server-multiple client messaging added

8) Commit:6805e9c 12/01/2016

- Server side now has two threads for incoming-outgoing messages.

9) Commit:66a2ffa 12/01/2016

- Client side now has two threads for incoming-outgoing messages

10) Commit:13b1bd1 12/01/2016

- Client side now has two threads for incoming-outgoing messages. (FIX: Wrong file commit) .

11) Commit:6d79da0 12/01/2016

- Picked number on the server now broadcast to all clients, client can check for Cinko (dummy service). Client-Server messaging is succesfully implemented. Broadcast and message sendin to a selected client is completed. Server side has been redesigned, it now creates a new thread for every client connection.

12) Commit:6d79da0 12/01/2016

- Server side generates the tickets and assigns it to the clients, and clients shows the tickets to the user

13) Commit:3c7bc03 12/01/2016

- UI fixes has been made. User can se the picked number with different background color, so the numbers that are picked before can be seen on the left side of the UI. User also can sign/select number on his/her ticket

14) Commit:413bd9e 12/01/2016

- Cinko validation is added, user is banned after 3 invalid cinko requests

15) Commit:15d4be8 12/01/2016

- Game winning condition and announcement has been added. Players can see each other's cinko status dynamically

16) Commit: 12/01/2016

- Username will be used for login function, and the winner will be announced by username. Also users can see each others username and cinko statusses.

Usage of The System

1) Running Server.py

- Firstly, Server.py should start running and on the console, listening message must be seen as in the image below.

```
Listening for incoming connections...
```
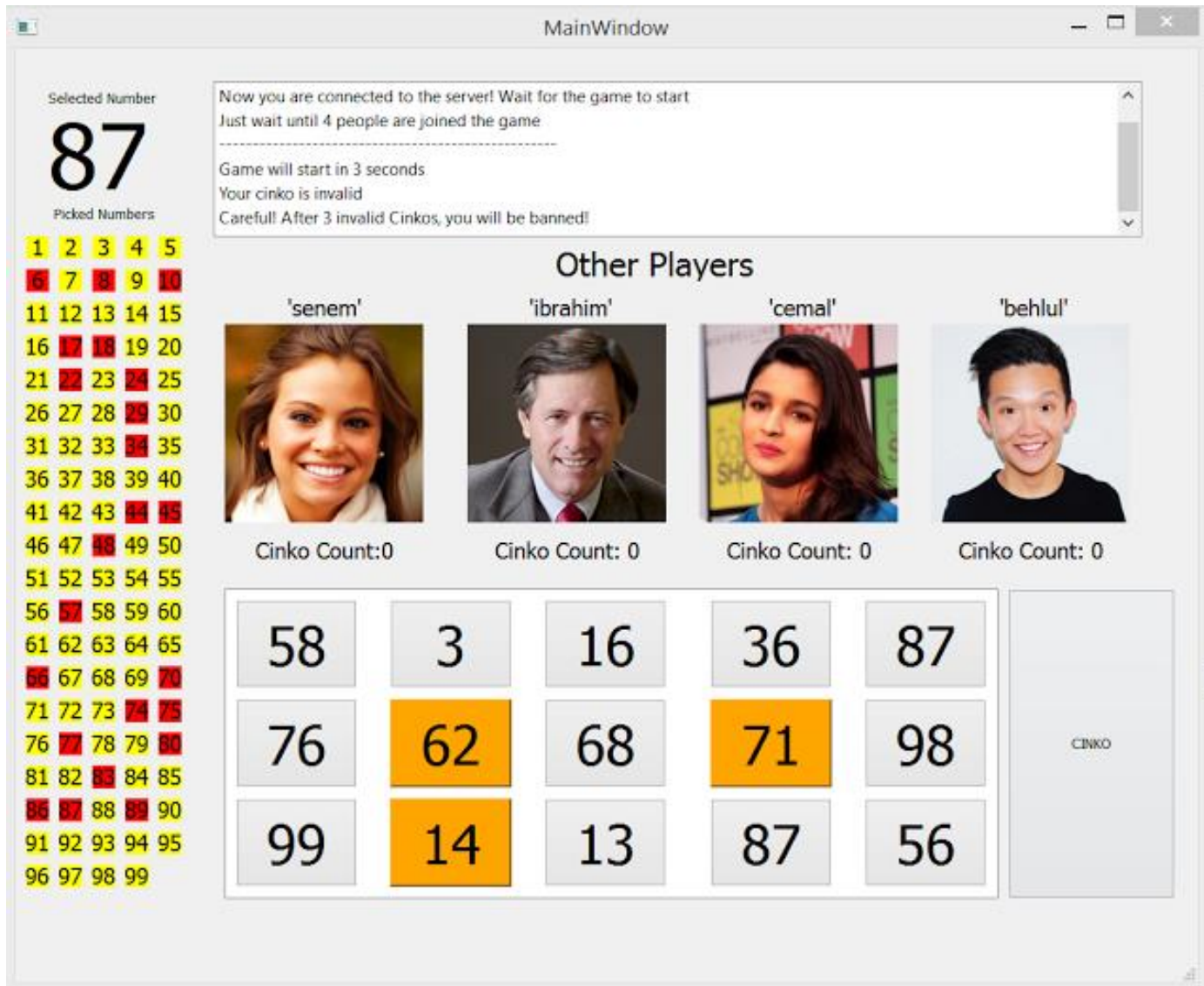
2) Running Client.py

- After running Server.py, it is now available to run Client applications by starting Client.py and the screen should be seen as in the picture below.

```
Please enter username:
```

- Username must be entered.

3) Client UI

- UI of the client is shown in the picture below.

- On the left pane there is the selected number at the top, and the past of selected numbers in the game at the below of it.
- On the center-up there is a textbox that the incoming messages are shown.
- On the center, there are the user which are in the game, and username and Cinko count datas are shown with dummy pictures.
- User can select the number on the ticket just by clicking on it.
- The Cinko button will be used for validation of the Cinko on the server. After 3 invalid request, user will be banned for the session.

Differences From Design&Requirements Document

There are some missing points, (because of deficiency technical skills, lack of time etc.) but also some changes upon to these reasons and other things that listed below.

1) Sessions

- The plan was supporting multiple game sessions on the same server, but now it is only possible to run one session on the game.
- Regarding to these lack of functionalities, there are some missing parts on protocol rules which are shown on the table below.

Request	Parameter	Response	Parameter	Definition
LSQ		LSA	sessionList	List game sessions
JSQ	sessionName	JSA		Request to join a game session approved
		JSD		Request to join a game session declined
CSQ		CSA	sessionName	Create new game session approved

2) Select Number

- From the client, it was planned to get the selected number as the user selects/clicks on the number on the ticket. As the server know the ticket detail and selected number , there is no need to send this data to the server.

Request	Parameter	Response	Parameter	Definition
SEN	pickNumber	NSA	selectedNumber	User selects number

3) Requirements

- There are some changes on some requirements also, which are listed below and the changes are in the notes next to it.

    1. User gets the current game sessions list–Not implemented
    2. User can create a new game session or can join an existing game sesison. –Not implemented
    3. Game starts after 60 seconds of the first user enters the session (or creates the session) or the max number of players(5) is reached. – Game starts after online people number reaches 5.
    4. User selects the number, the number should be sent to the server as it is signed.
    5. If a user quits from the game, the other users will be informed. –Not implemented