COMP90015 Distributed System

Semester 2, 2018 Assignment 2 Report

Group: bomb

Student name	Student ID	Student email
Changda Jiang	879725	changdaj@student.unimelb.edu.au
Xingchen Li	935256	xingchenl2@student.unimelb.edu.au
Yiqi Liu	895816	yiqil5@student.unimelb.edu.au
Yubing Yang	883877	yubingy@student.unimelb.edu.au
Tutor: Alisha Aneja (9.00-10.00 am Thu)		

Design and Implementation of distributed scrabble game

1. Introduction

According to the project requirement, the main aim of this task is to design and realize a distributed version of scrabble game which multiple players are able to take turns to play. Based on the detailed analysis, this scrabble game system utilizes client-server interaction model so that multiple clients can connect to a multi-threaded server and perform operations concurrently. Client communicates with server via sockets that uses TCP protocol and thread-per-connection is applied to the system to implement a multi-threaded server. The message exchange format for this system is JSON.

2.Game Architecture

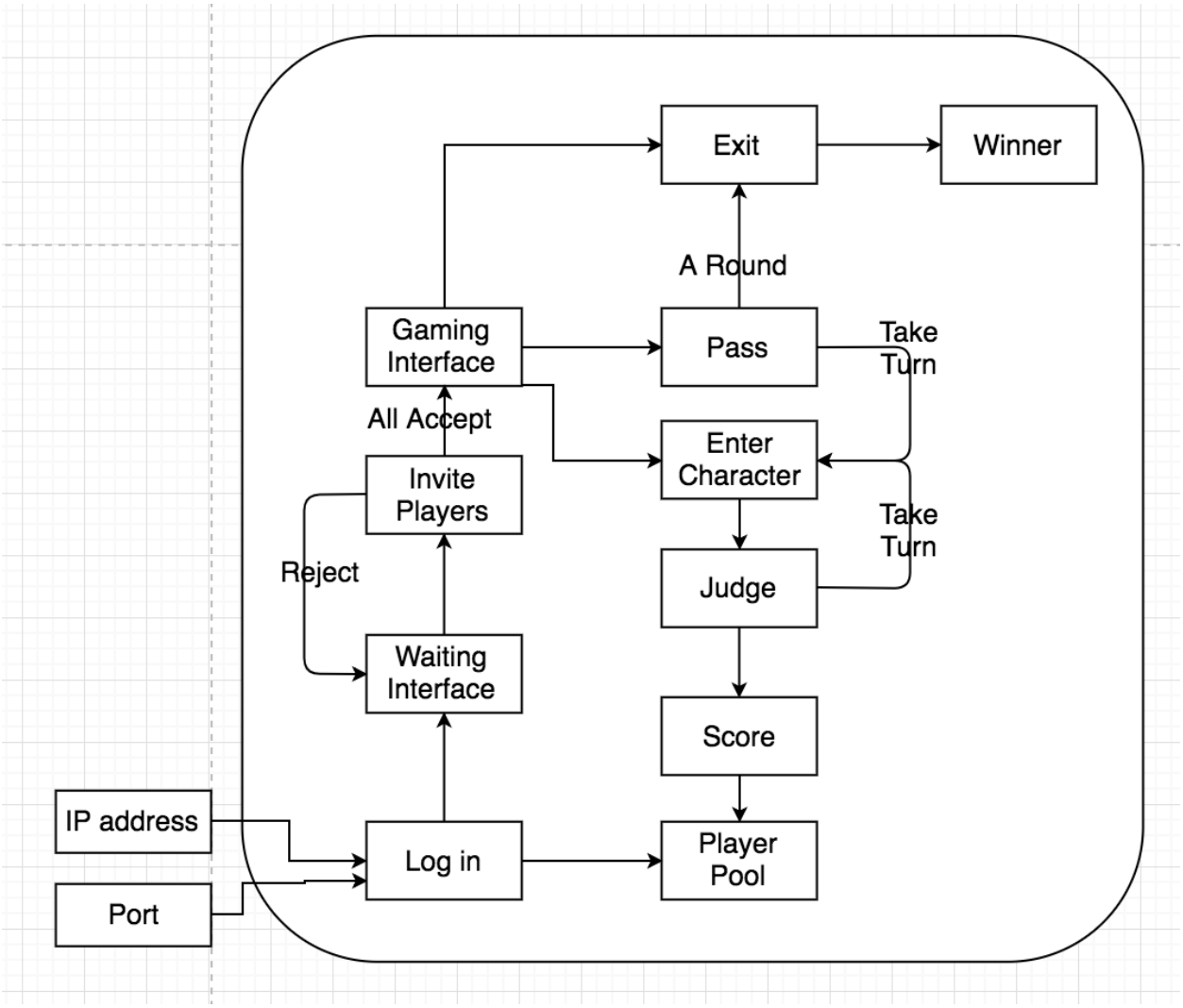


Figure-1

As shown on the above graph, each player can log into the system by inputting server IP address and port, and then they will be placed in a online player list and wait for other players. If someone clicks “start” button, players in waiting interface will receive a notice to accept or reject the game. If all players choose “accept”, the gaming interface will be shown, otherwise the game fails to start. During the game, players enter character in a button which is chosen by themselves, and other players will decide whether the player can get score according to vertical and horizontal words. Player can also clicks “pass” to give up this turn or “exit” to quit the game. When game is over, the winner will be broadcasted to every player.

3. Communication Protocol & Message Format

3.1 Communication Protocol

Client communicates with server via sockets that uses TCP and implements a thread-per-connection architecture. The details of how to use and realize sockets and thread-per-connection respectively are explained in the section of design diagram.

3.2 Message Format

JSON object is utilized for data exchange. The details of JSON object are shown below.

key	value	details
"userName"	Player’s input	Player's username to log in.
"logSuccess"	“yes”|”no”	The feedback of login from server .
"x"	Player’s choice	The horizontal location of the input letter.
"y"	Player’s choice	The vertical location of the input letter.
"letter"	Player’s input	The letter that a current player typed into.
"command"	confirm|pass|exit|start|wait|done|finish|waitAccept|inviting|inviteFailure	The commands are used for the server to control the status of the game.
"turn"	Consists of all the players in this round of the game	It indicates if it is the current client's game turn, if yes,release the control of the gaming frame.
"hScore"	Player’s choice	It records the vote of a player.
"vScore"	Player’s choice	It records the vote of a player.
"accept"	“yes”|”no”	It indicates whether a player accepted a game or not.
"rejecter"	Player’s choice	The player refused a game.
"leaver"	Player’s choice	The player quit a game.
"allPass"	Player’s choice	All players passed his turn which means that the game ended.

4. Design diagram

Server:

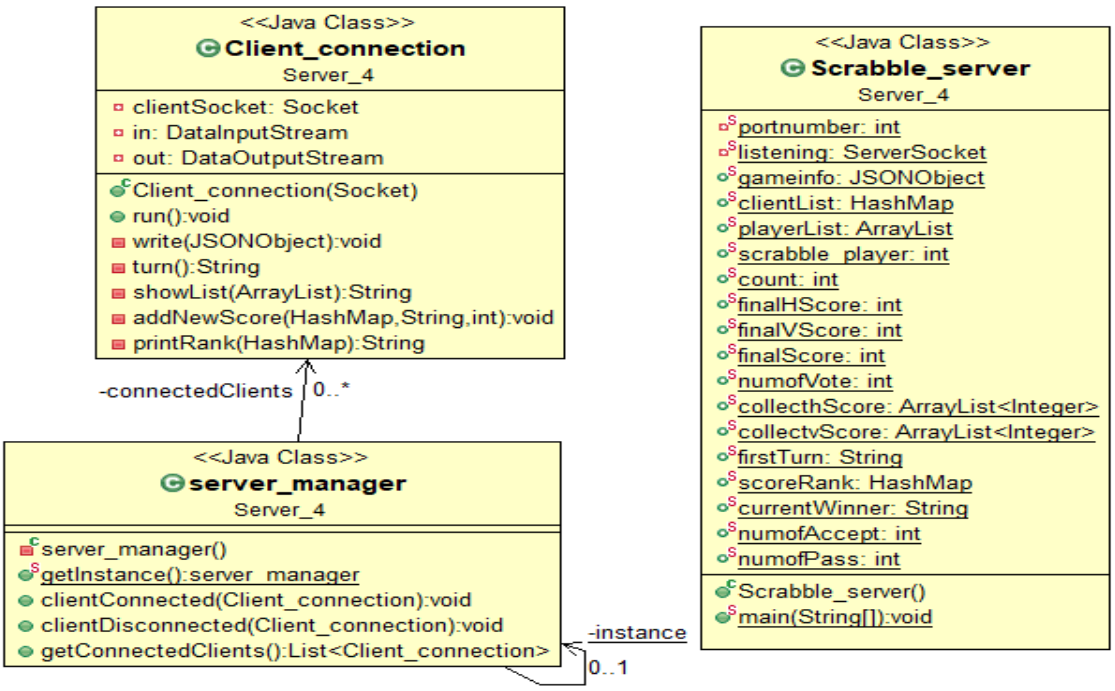


Figure-2

As can be seen from the picture above, the server-side design is comprised of three major classes. The server_manage class is created for processing the connection of incoming clients and maintains every connection in a consistent state and Client_connection keeps listening the incoming message transmitted by each connected client and deals with synchronized issues of writing and taking turns. Also, it is responsible for updating score related problems. Regarding to the Scrabble_server class, all the static instances are defined in it in order to handle the information for each game round and membership pool.

Client:

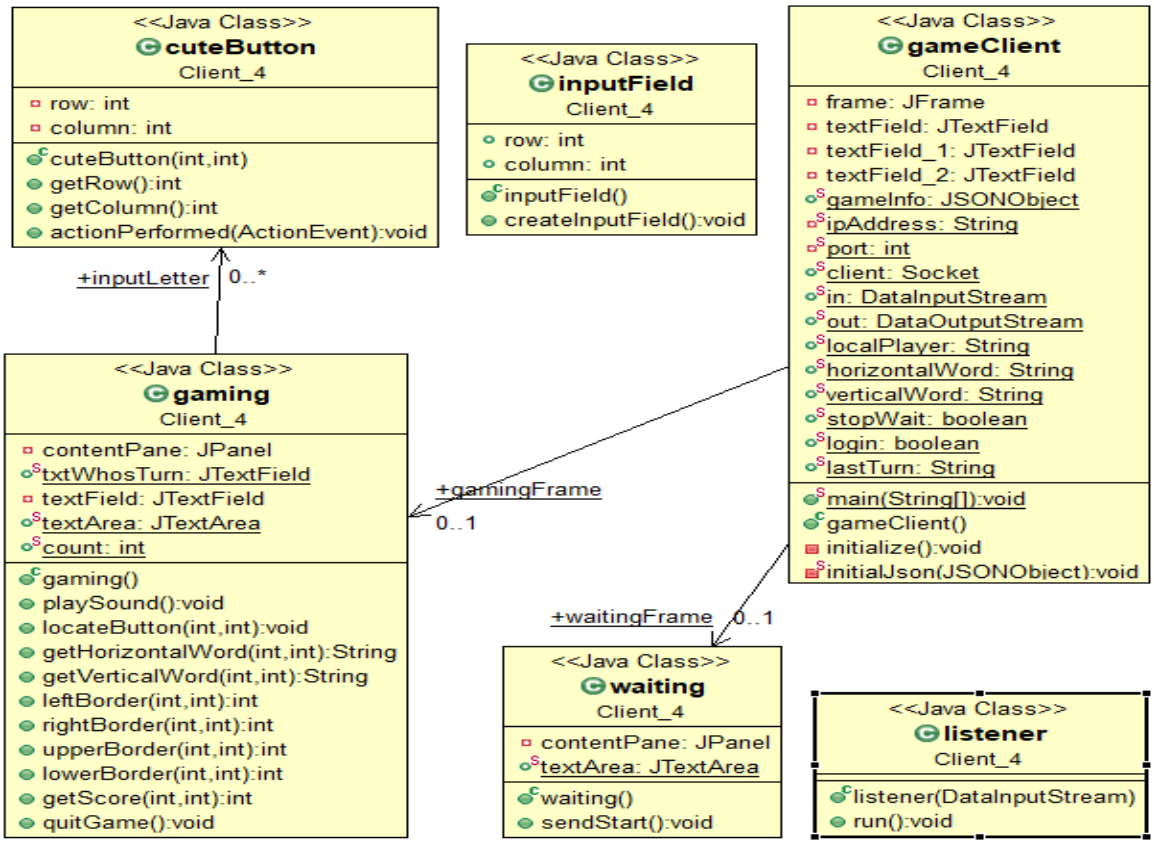


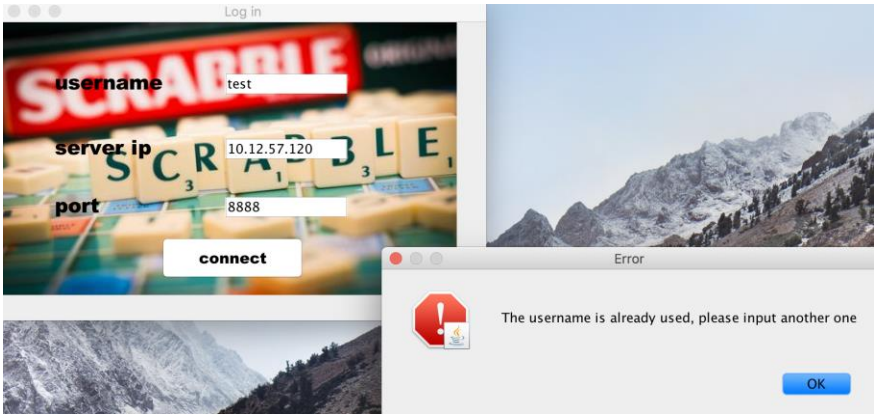
Figure-3

The above diagram presents the relationship between each class in client-side. The cuteButton represents each component (button) in game board and the inputField is formed by iterating inserting cuteButton into it. There are three main user interfaces, i.e. gameClient, gaming and waiting. The gameClient class is associated with log in frame to provide a interface for all users to connect to the server and deals with initializing the JSON for gameinfo. Furthermore, the waiting class is designed for satisfying the requirements of sending invitation to players' friends and the mechanism of only one game in progress is handled here. The gaming class is provided for the function such as displaying the current player, showing the rank for each round, and performing the placing operation etc. Also, rules of this game is processed in this class. The listener class extends thread to keep listening the incoming information sent by server and identifies whether the incoming message is belong to the current player and further handles it by judging the different command in JSON message.

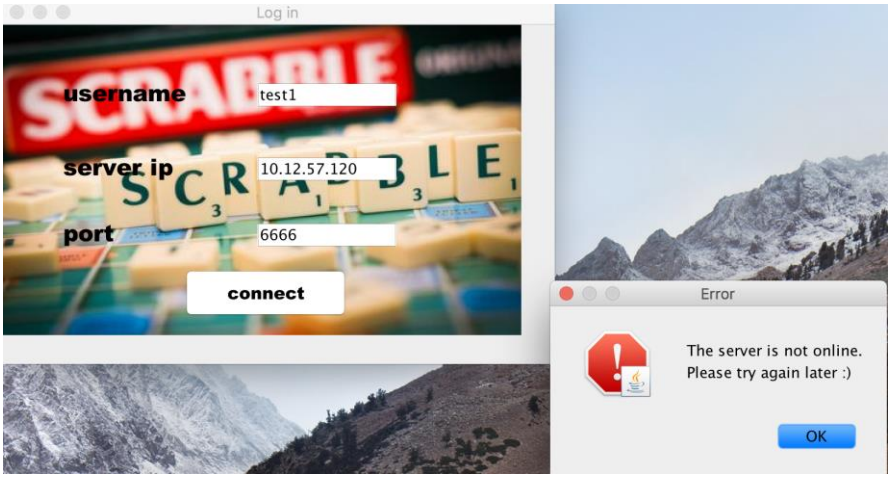
5. Implementation

5.1 Log In Interface

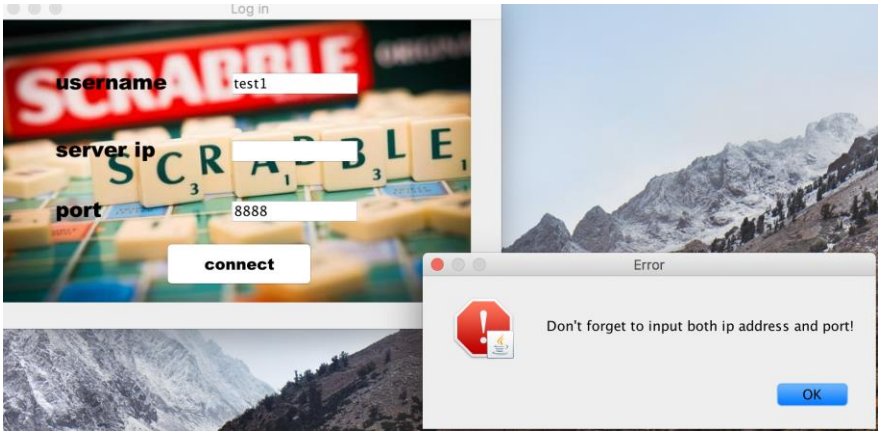
5.1.1 Unique username handling



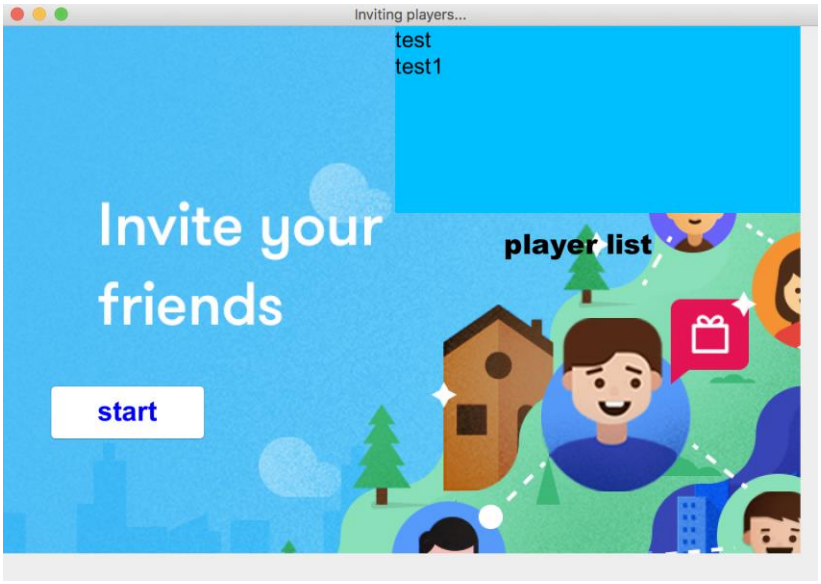
5.1.2 Invalid IP address and Port number handling



5.1.3 Invalid Inputting



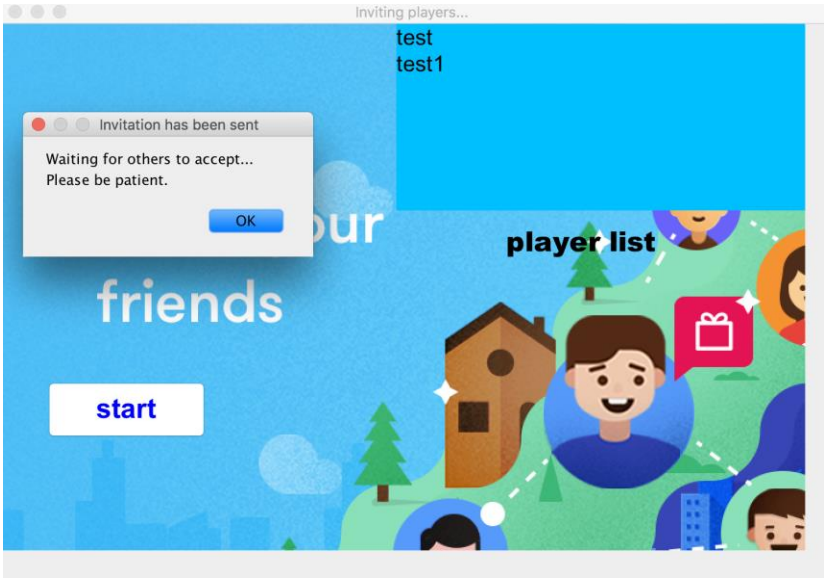
5.2 Waiting Interface



5.2.1 Invitation Mechanism

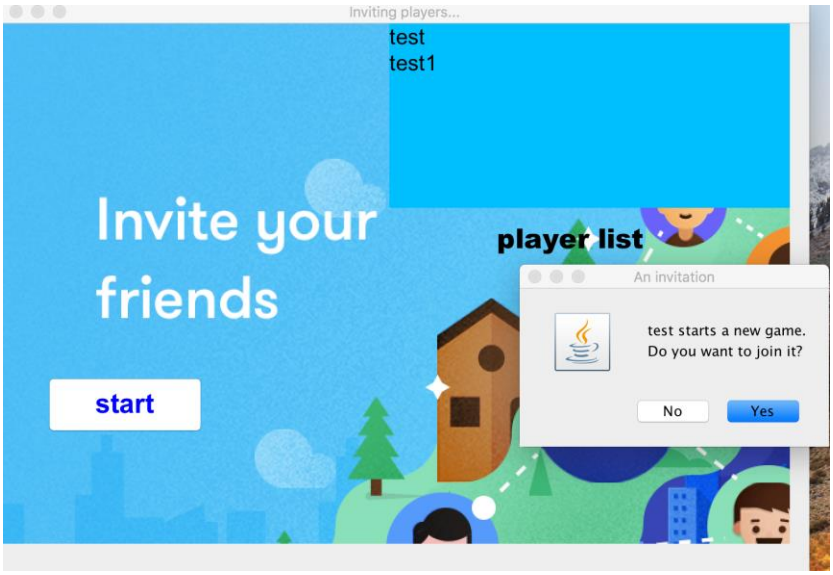
5.2.1.1

If a user-- “test” clicks start button, the interface will show



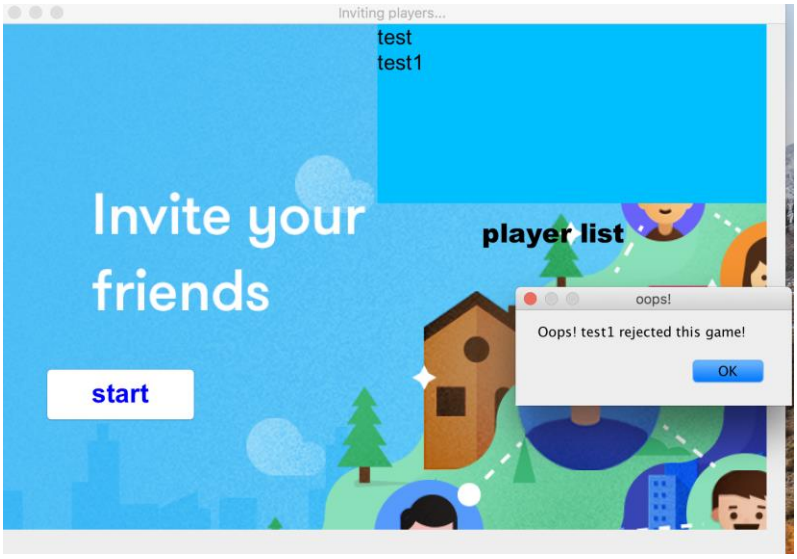
5.2.1.2

Other users interface



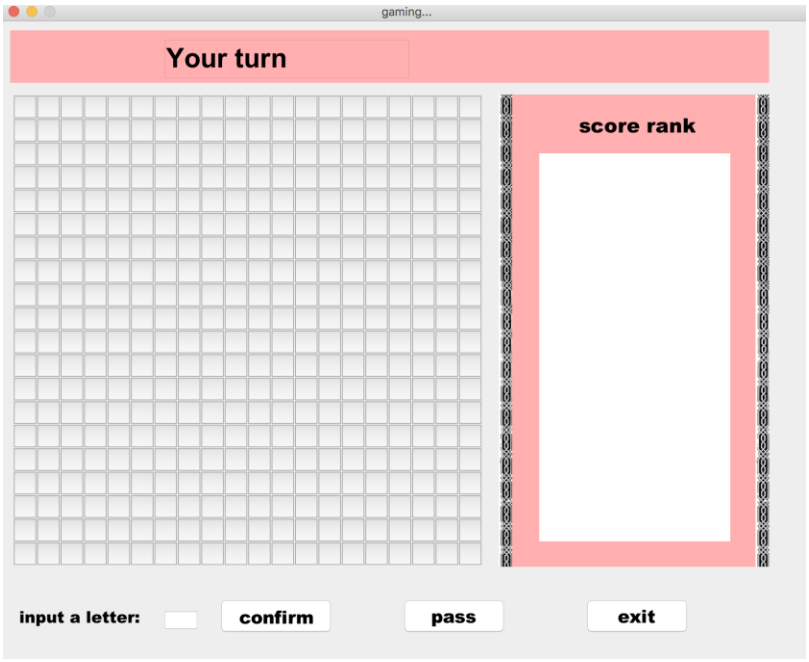
5.2.1.3

If someone rejects the invitation, user interface will show the notice

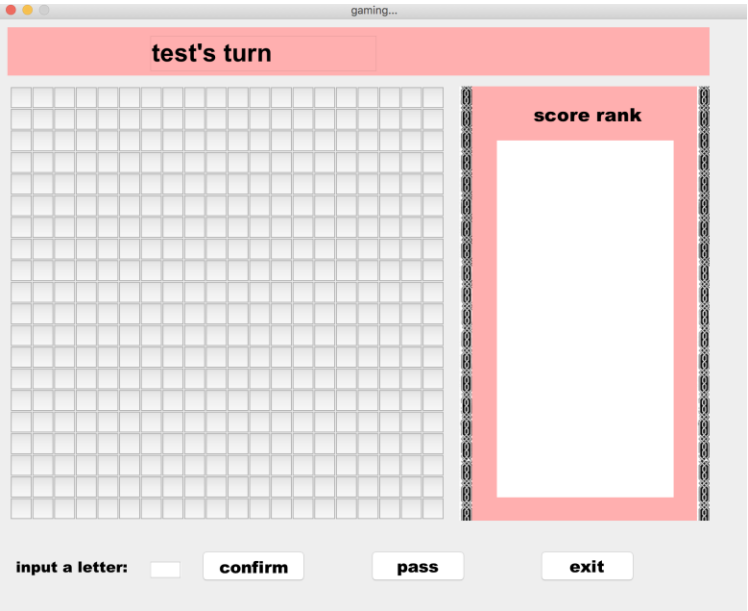


5.3 Gaming Interface

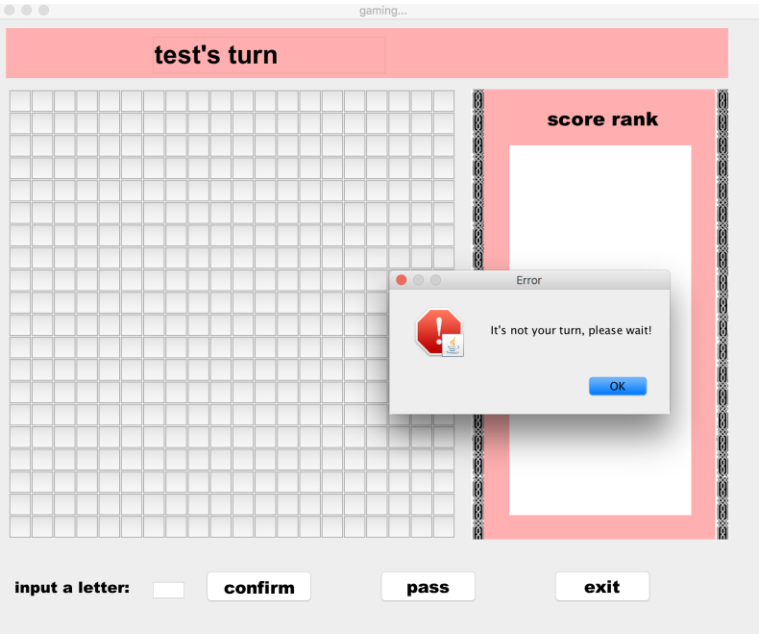
5.3.1
Current player Interface



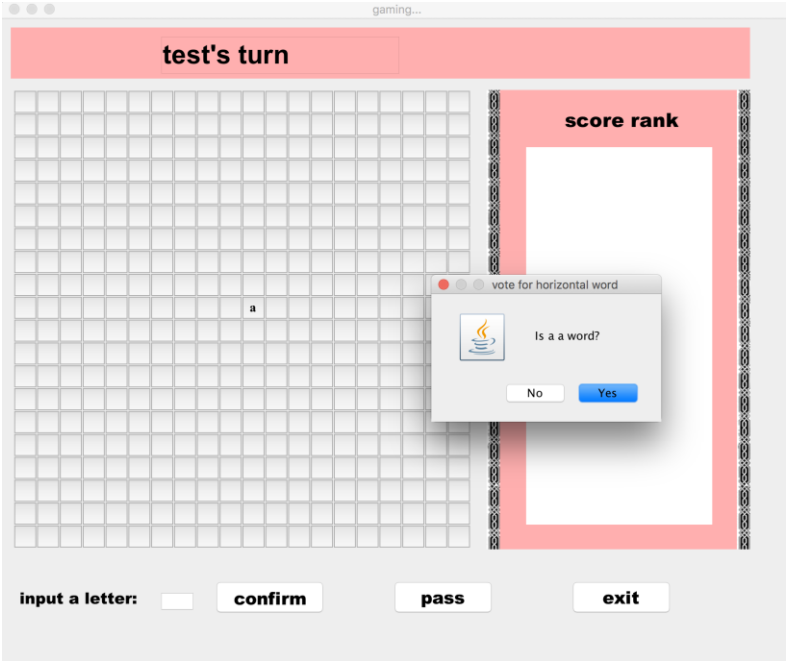
5.3.2
Other players Interface

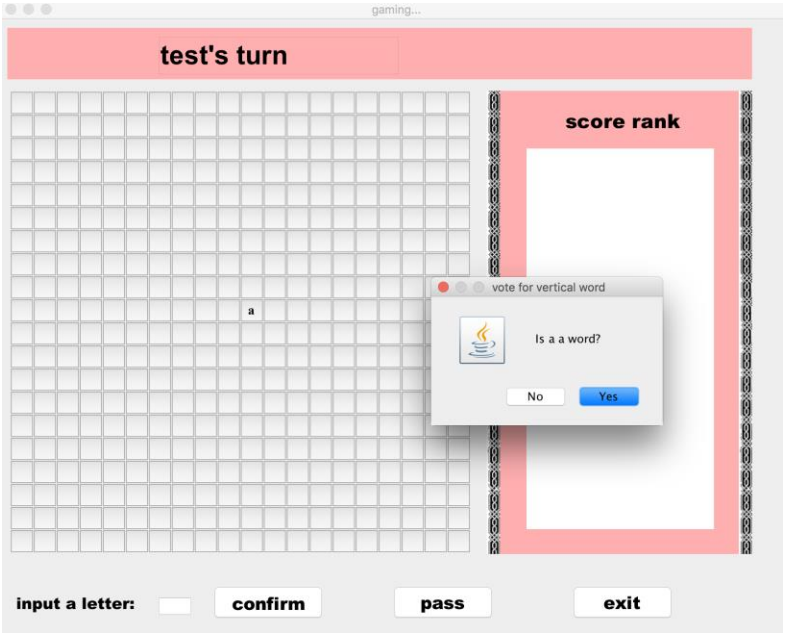


5.3.3
Player wants to perform operation when it is not his/her's turn

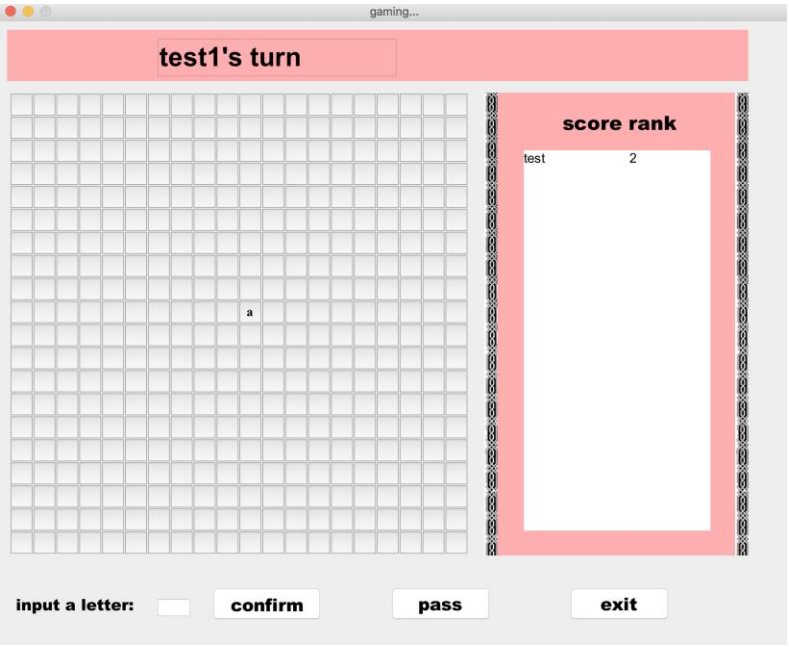


5.3.4
Scoring state(There will be 2 times of voting for vertical and horizontal words)

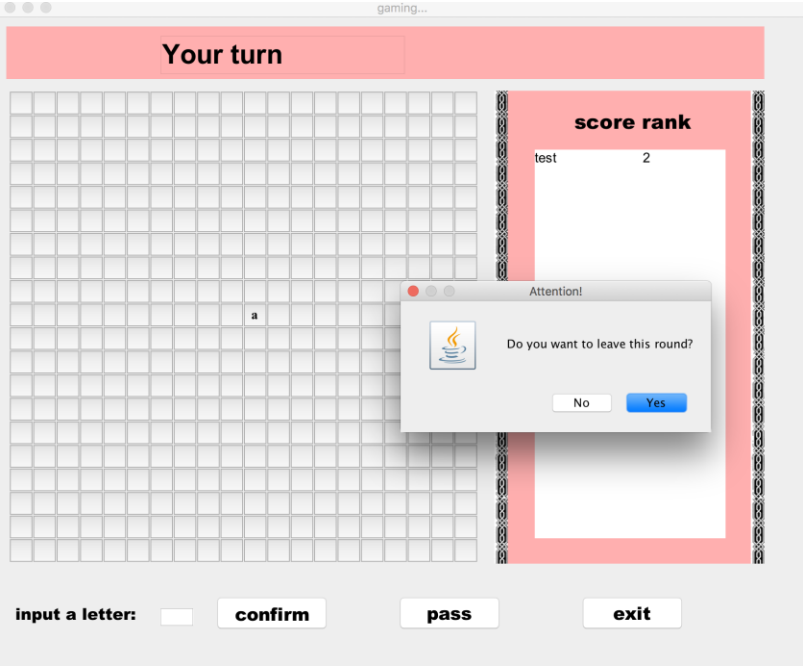




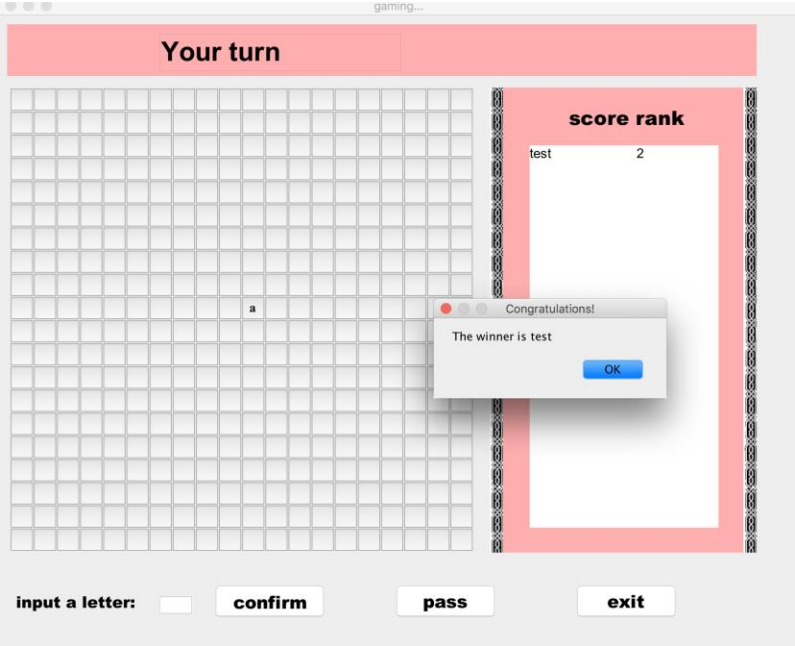
5.3.5
Score Rank Update and Taking Turns



5.3.6
Exit Feature



5.3.7
Show Winner



6.Critical Analysis

•UI and Button Click Music Design

A lovely UI and button click music design can provide a relaxed gaming atmosphere for user.

•Memory Usage

The system will generate a very high memory usage, since server transmits stream data all the time. When we are testing, our computers usually are hot and noisy. Therefore, optimizing structure and communication method will be our main task when we have more time.

•Exception Handling

In our system, we consider multiple exceptions during log in, waiting and gaming states such as, wrong IP address, repetitive player name. However, since limited time, we cannot handle all of exception. For example, server cannot receive the vote from player due to unstable network. If time is sufficient, this problem would be fixed.