CSCI-201L: Team “Oh Chute”
Arfan Rehab, David Garry, Derek Cheng, Jiachang (Ernest) Xu, Nicholas Thompson
Project Name: *Chutes & Ladders, Generations*
Course Producer: Priyanka Shah
Lecture Section: TTh, 11:00am - 12:20pm
Due: Sunday, October 30th, 2016

Revised High-Level Requirements (No Changes Made)

The game will be broken down into two different windows: a login window and a gameplay window. The gameplay window varies in functionality based upon whether or not a user is registered. The functionality of the login window and the two variations of the gameplay window are described below.

Login Window Functionality

The application opens to a login menu, which allows for registration of new accounts, login to a previously established account, and access to local gameplay on a guest account. The required items for registration of an account are a username and password. The requirements for the username are that it be unique (not already established) and 4 or more alphanumeric characters. The requirements for the password are that it be 6+ alphanumeric characters or symbols.

If the user tries to create an account using a username that has previously been established, they will receive an error message notifying them that they must choose a different username. If the user tries to create a username or password that do not meet the above requirements, the user will receive an error message notifying them of their error. Finally, if the user attempts to login to an account using an incorrect username or password, the user will receive an error message indicating that the password entered is incorrect, and that they must try again.

Non-Registered User Gameplay Window Functionality

If the application is accessed through a guest account (no login), only limited gameplay functionality will be accessible to the user. The non-registered users will only be able to play against 3 CPUs, on a single premade standard gameboard with a size of 5x5. The gameplay will progress as follows:

Each player, human or computer, will take turns rolling a 6-sided die. The players turn order will be determined randomly at the start of the game. The number indicated on the die represents the number of spaces that player must move forward. The gameboard consists of tiles on a 5x5 matrix. Regular squares simply act as placeholders to indicate progress along the game. When a user lands on a square that is at the start of a ladder, they move up the ladder, following it all the way to the square at the top of the ladder. When a user lands on a square that is at the top of a chute, the user must move backwards following the chute, all the way down to the square at bottom of the chute. If a user lands on a square that is at the top of a ladder, or the bottom of a chute, they move normally as they would accross any regular square. During gameplay, statistics from the game will be displayed on a live-updating scoreboard. These statistics will include: spaces moved forward by each user (ignoring backwards movement), ladders climbed by each user, and chutes descended by each user, as well as the mean die roll of each player.

The first player to reach the end of the gameboard at tile 25 wins the game. Once a game is over, the user is sent back to the login screen where they can choose to register an account or play another game as a guest.

Registered User Gameplay Window Functionality

Once a registered user successfully logs in, they are brought to a game lobby screen. From the game lobby, they can view accessible games, and select a game to join. The number of users playing in each game and the game room's current theme will be displayed on the game lobby. A user can also view his/her account profile, where they can change their password and view statistics on their playing history. The game lobby will also display a user roster where users can search for other user profiles, view their status (online/offline), and view their playing statistics and awards/achievements.

Once a registered user successfully logs in, they are brought to a game lobby screen. From the game lobby screen, users can access their profile, view accessible games, and view a list of active users. In the user's account profile, the user can change their password and view statistics and awards on their playing history. The list of accessible games will show all joinable games that have not filled the four player maximum. The lobby will create enough accessible games to accommodate all online players.

Once the user has selected a joinable game, the user will enter the game's start screen. After a 30 second waiting period from the moment the first user enters the screen, any user in the game can press the "Start Game" button to initialize the game with 4 players. Games support 1-4 users, and if there are less than 4 users in the game, the remaining player spots will be filled out with CPU players.

Registered users experience the same gameplay features as non-registered users, with a few additions. The board size is increased to 10x10, and the game board is randomized for every game, such that chutes & ladders are randomly positioned for each individual game. The users can also access a library of playable game themes such as "Student Loans and Scholarships" or "Trojans and Bruins". The user selects their preferred theme in their account profile. Each user sees their preferred theme while playing and their preferred theme can only be changed outside of gameplay by changing a user's settings from their account profile.

Registered users will earn awards & achievements for various tasks they accomplish during gameplay. Some of these will be novice-level (i.e. "Played first game!" or "Won first game!"), while others will be earned with extended gameplay (i.e. "Won 20 games!"). When a user wins an award/achievement, the timestamp will be recorded and both the award and time it was earned will be published to the user's profile.

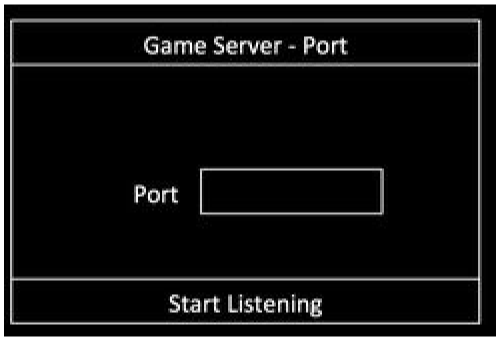
When a game ends in this playing mode, the program does not terminate. Instead, the users are sent back to the game lobby, where they join another game, view their own or another user's profile, or exit the program.

CSCI-201L: Team "Oh Chute"
Arfan Rehab, David Garry, Derek Cheng, Jiachang (Ernest) Xu, Nicholas Thompson
Project Name: *Chutes & Ladders, Generations*
Course Producer: Priyanka Shah
Lecture Section: TTh, 11:00am - 12:20pm
Due: Sunday, October 30th, 2016

Revised Technical Specifications (Changes in Strikethrough)

Game Server

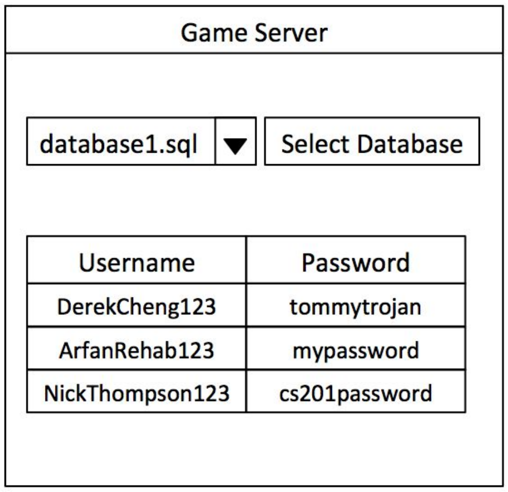
~~*Port GUI (4 hours)*~~



~~Figure 1: Port GUI lets the user enter a port number to listen to~~

~~The Game Server will first display a window for user to enter a port number. After a port number has been entered, the "Start Listening" `JButton` will be enabled. The valid port number should be between 0 and 65535. The "Start Listening" `JButton` should only be enabled if the port number entered in the text field is within the above range. If a certain port number is already in use, then the Port GUI should display an error message: "Port already in use. Please select another port number within the valid range." If a unoccupied valid port number is entered, then we proceed to the Game Server GUI upon the user clicking the "Start Listening" `JButton`.~~

Game Server GUI (8 hours)



Username	Password
DerekCheng123	tommytrojan
ArfanRehab123	mypassword
NickThompson123	cs201password

Figure 2: Tester selects game database .sql file

While in testing, there will be a Game Server GUI in order to allow us to develop the game more easily. Eventually, before the final release, the content described below will not be visible to the user, and the game database will be hard coded without options for a different database.

The Game Server GUI will have a dropdown list on the top for the user to specify which player database we should use for our testing purpose, and a "Select Database" `JButton` next to the drop-down list. The player database that we will use in this project will be written in MySQL. Each player's information includes username, password, playing statistics, and awards & achievements. These will be stored in a Player class, in order to more easily access and modify these data members (further explained in Detailed Design). For testing purposes, the server will have a `JTable` inside of a `JScrollPane` to display each player's username and password, so that present user accounts are immediately apparent. When the "Select Database" `JButton` is clicked, and everything is executed without error, the Game Server GUI will display a message to a `JLabel`, "Ready for clients to log in." If there is an error, the error message will be displayed to the `JLabel`.

Multiple Clients Connections (4 hours)

Since multiple players can play this networked game simultaneously, once a specific player database is selected on the Game Server GUI, you are not allowed to switch to another database, if there are any clients connected to the Game Server. The same rule applies for close operation. The Game Server is not allowed to quit, there are still any clients connected to the Game Server. These two rules ensure that we can resemble the networked game play in the real world.

Login Menu GUI (8 Hours)

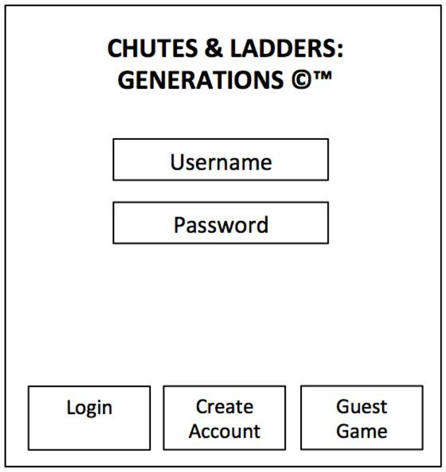


Figure 3: Login screen has text fields for user input and buttons to login, create account, or start a guest game

The login menu is the first screen shown when the application is run. The login GUI is always connected to the server GUI, and acts as a client program. The JDBC drivers will allow SQL to be embedded within our Java code and execute the code on the database programmatically. The menu will consist of two `JTextField`s for the username and password strings that the user will provide, as well as three `JButton`s: a login button, a create account button, and a guest game button. If the guest game button is pressed, neither of the strings in the `JTextField`s will be stored or processed: instead, a new Guest Game GUI will appear. If the register account `JButton` is pressed, the information in the two `JTextField`s is processed in the following way:

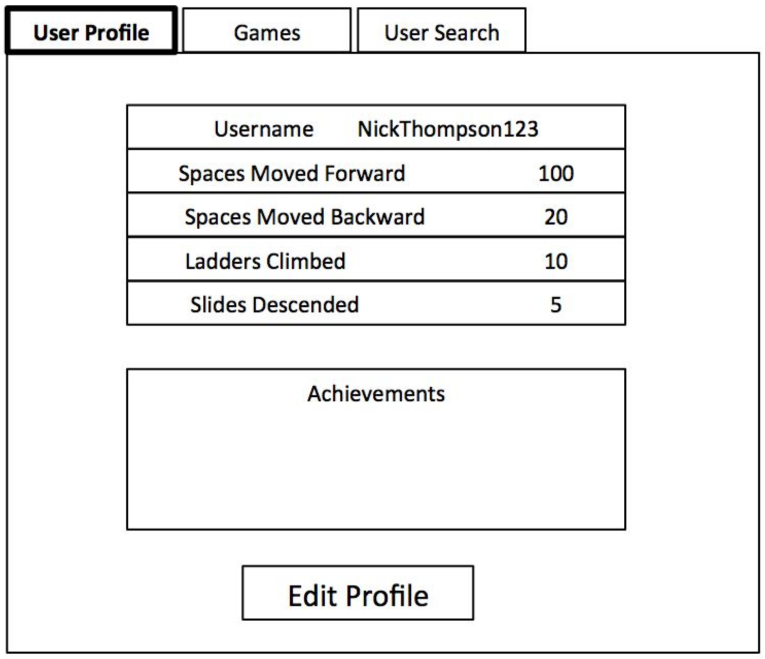
1. 4 or more alphanumeric characters or symbols must be present in the username `JTextField`
2. 6 or more alphanumeric characters or symbols must be present in the password `JTextField`
3. The username must not already exist in the database
4. If any of these conditions are not met, the user will be alerted and an account will not be created

If the login `JButton` is pressed, the program will check to ensure that the username and password are both present and correct. If they are not, the user will be alerted and will not be able to login. If the user username exists and the corresponding password is entered correctly, the user will the game lobby will open and the login menu will disappear. When the local game `JButton` is pressed the application user is taken to a new window with a 5x5 standard game board with 3 CPUs, further described below.

Lobby GUI

The lobby will consist of a `JTabbedPane` of three different JPanels: User Profile, Games Available, and User Search. This menu will be shown after a user logs onto a registered user profile.

User Profile Tab (4 hours)



Username	NickThompson123
Spaces Moved Forward	100
Spaces Moved Backward	20
Ladders Climbed	10
Slides Descended	5

Figure 4: The user profile tab shows player stats and achievements, and has an "Edit Profile" `JComboBox` that allows the user to modify their username or password

The first tab "User Profile" will allow users to keep track of the various stats and achievements they have accrued over time. Users will also be able to view and edit their username and password. At the top of the page the data members from the User class will be displayed as text. These include username, spaces moved forward, spaces moved backwards, slides descended, and ladders climbed. At the bottom of the `JPanel` there will be a "Edit Profile" `JButton` which opens a `JComboBox` with two options, "Edit Username" and "Edit Password." When either of these two menus are clicked a new window will open. This menu will consist of 3 `JTextField`s. The first two will ask for the current username and password. If these two criteria are entered correctly, whatever is entered in the third `JTextField` will replace the username or password respectively when the user clicks the submit button.

Available Games Tab GUI (8 hours)

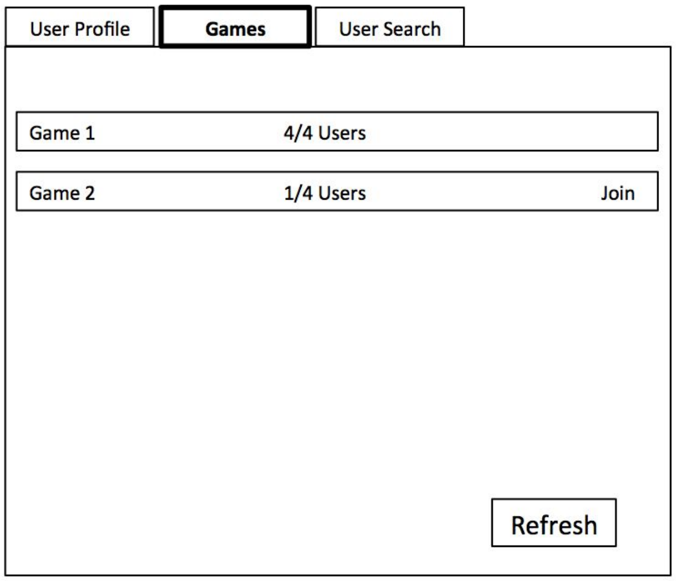
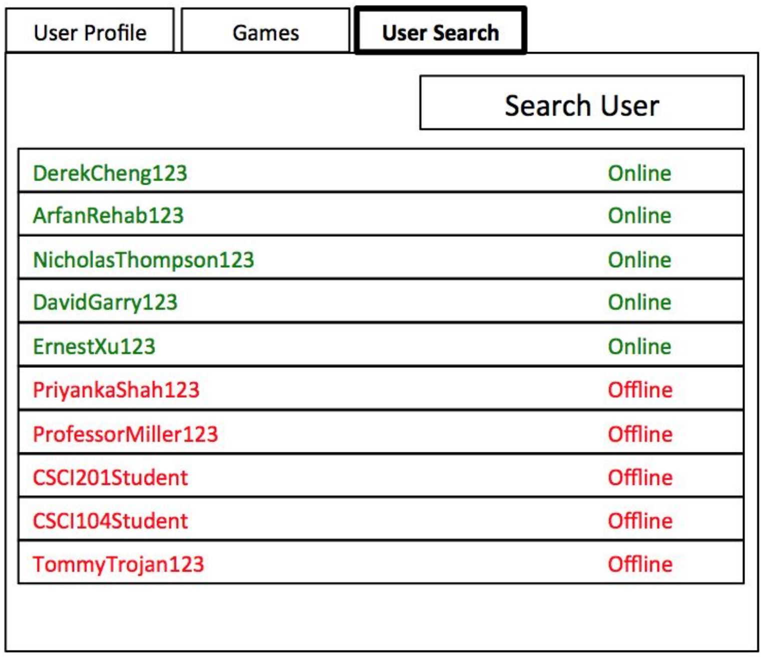


Figure 5: The available games tab has a scroll pane that shows all live games and a refresh button

This tab will contain a `JScrollPane` that allows users to view all games that are currently being played, or that can be joined.. All games will be displayed in a `JScrollPane` of `JButton`s. Each game will be represented by a button with that shows the names of the players currently in game as well as whether or not the game can be joined. At the bottom of the list there will be a refresh `JButton` which updates the game list. Games will be added to the list so that all users currently active can join a game. If the number of available human player slots drops below the number of active players not currently in a game, the game lobby will create a new empty game. This might happen as users login or a game is started with one or more CPUs, removing an available human player slot.

User Search (8 hours)



DerekCheng123	Online
ArfanRehab123	Online
NicholasThompson123	Online
DavidGarry123	Online
ErnestXu123	Online
PriyankaShah123	Offline
ProfessorMiller123	Offline
CSCI201Student	Offline
CSCI104Student	Offline
TommyTrojan123	Offline

Figure 6: User search tab shows all users in a `JScrollPane` and has a search bar to filter the displayed users

This tab consists of a `JScrollPane` of `JButtons` with all the active usernames as well as their active status. If another user is online the username will be green, otherwise the username will be red. At the top of the menu there will be a search bar which allows you to search for other users of the game by their username. If the username is entered correctly and the search button is clicked, the list of usernames will be reduced to just that username. If you click on any of the users profiles, you will be given access to their user profile page, without the edit profile functionality. If a user is viewing another player's profile they will not be able to view their password. Other than that, all user information will be displayed the same as it would on the User Profile Tab.

Gameplay GUI

Start Screen (4 hrs)

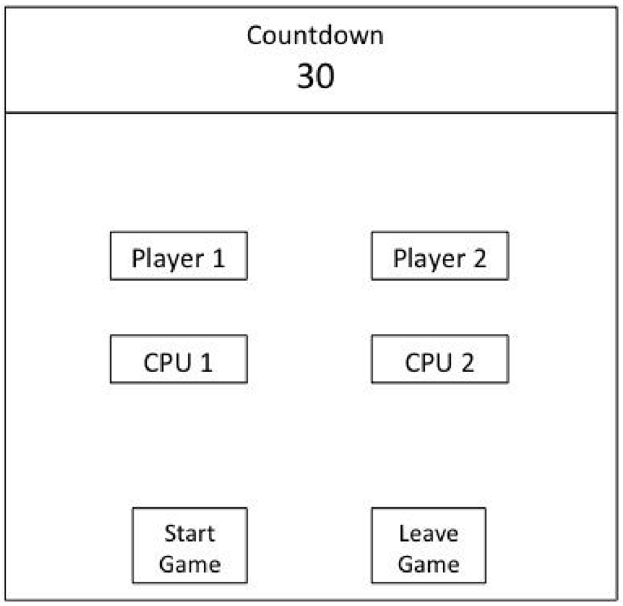


Figure 7: The start screen consists of 4 `JLabel`s for the different players, a `JLabel` for countdown to start of game, and `JButton`s for start and leave game.

The gameplay screen will be shown once a registered user has logged in through the login menu or if the user chooses the guest game button. First, the user will be brought to a start screen `JPanel`, which consists of 4 `JLabel`s denoting the teams participating, a `JLabel` displaying the time until the game can be started, a `JButton` to start the game, and a `JButton` to leave the game. Games support 1-4 users, and if there are less than 4 users in the game, the remaining player spots and `JLabel`s will be filled out with CPU players (as seen in Figure 1). Users can leave the game before the game starts by pressing the “Leave Game” `JButton` and confirming their decision on a `JOptionPane`, after which they will be replaced by a CPU. The “Start” `JButton` is initially disabled, and becomes enabled 30 seconds after the first user enters the Start Screen. A `JLabel` at the top of the screen will count down the seconds until the start button is enabled. Once enabled, any user can press the “Start Game” button to begin gameplay, with the remaining player spots filled with CPU players.

Game Screen (6 hrs)

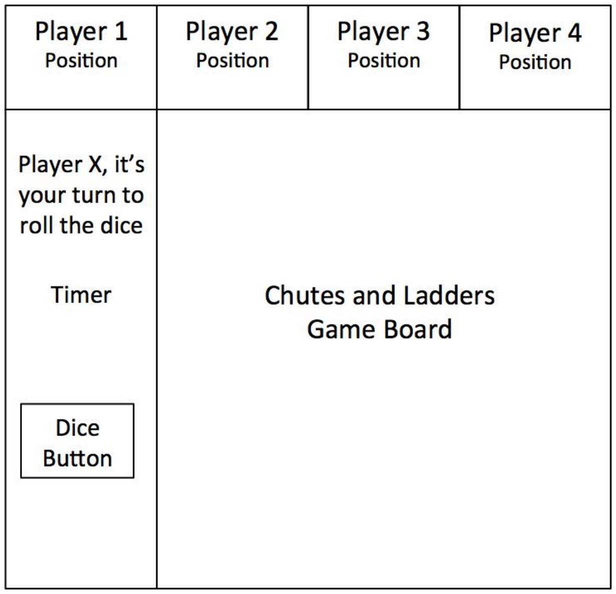


Figure 8: The gameplay screen consists of a 4 `JLabel`s for the scoreboard at the top, a GridLayout for the game board, a `JLabel` to denote whose turn is it, a `JLabel` for the timer, and a `JButton` to roll the dice

The game board will consist of a grid layout that will be a 10 by 10 matrix for registered users; for guests, it will be a 5 by 5 matrix. The board will be customized based on the selected theme, which will be made through graphics: colors, images, etc. The chutes and ladders will consist of images that are randomly adjusted on the grid, and the players will be shown on the board through movable 3D points (mouse listeners). On the side of the screen, the die will be an image that remains on the side of the screen and spins around randomly every time a player rolls it. There will also be a timer `JLabel` that implements the Runnable interface and uses a thread to keep tabs on the current player’s time.. After 30 seconds, the timer resets and the game progresses by switching the current player’s turn in clockwise order. The player’s turn will also be represented by a `JLabel` above the timer that updates its text with the current team’s name.

Scoreboard (4 hrs)

The scoreboard will consist of `JLabel`s that shows all the players’ current scores. It will be updated accordingly as each player moves up a ladder or lands on a chute (snake) and moves down. The scores are adjusted relative to where each player is on the board (how close each player is to winning). In addition, each `JLabel` will have an action listener to bring up a `JDialog` on the player’s profile to show their statistics and awards.

CSCI-201L: Team “Oh Chute”
Arfan Rehab, David Garry, Derek Cheng, Jiachang (Ernest) Xu, Nicholas Thompson
Project Name: *Chutes & Ladders, Generations*
Course Producer: Priyanka Shah
Lecture Section: TTh, 11:00am - 12:20pm
Due: Sunday, October 30th, 2016

Detailed Design Document

Hardware Requirements

Windows:
- Windows 8 (Desktop)
- Windows 7
- Windows Vista SP2
- RAM: 128MB
- Disk Space: Recommended 512MB
- Processor: Minimum Pentium 2 266MHz processor

Mac OS X:
- Intel-based Mac running Mac OS X 10.8.3+, 10.9+

Linux:
- Oracle Linux 5.5+
- Oracle Linux 6.x (32-bit), 6.x (64-bit)
- Oracle Linux 7.x (64-bit)
- Red Hat Enterprise Linux 5.5+ (32-bit), 6.x (64-bit)
- Ubuntu Linux 12.04 LTS, 13.x

Software Requirements

Java 8
Eclipse IDE for Java EE Developers (Luna/Mars)

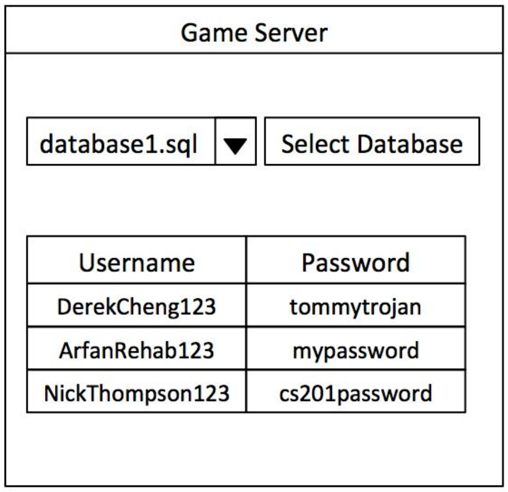
Server Package: Class Diagrams & GUI Mockups

GameServer
- ServerSocket : ss
- ServerListener : serverListener
(+) GameServer()
(+) main(String) : void
(+) sendDatabase(HashMap<String, User>) : void
- listenForConnections() : void

GameServer - Retrieves the hard-coded ServerSocket, and Creates a GameServerGUI. The user database can be sent through the creation of a ServerListener.

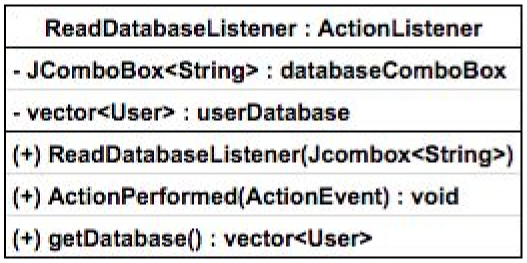
GameServerGUI : JFrame
(+) long : serialVersionUID
- JTextArea : loginInfoTextField
- JScrollPane : loginInfoScrollPane
- Jbutton : selectDatabaseButton
- JComboBox<String> : selectDatabaseComboBox
(+) GameServerGUI()
- initializeComponents() : void
- createGUI() : void
- addActionAdapters() : void
(+) addLoginInfo(pair<String, String>) : void

GameServerGUI - A window for the Server to allow a user to select a user database (for the testing purpose of course), and to send it to all connected Clients. The username and password information will be displayed for testing purpose.



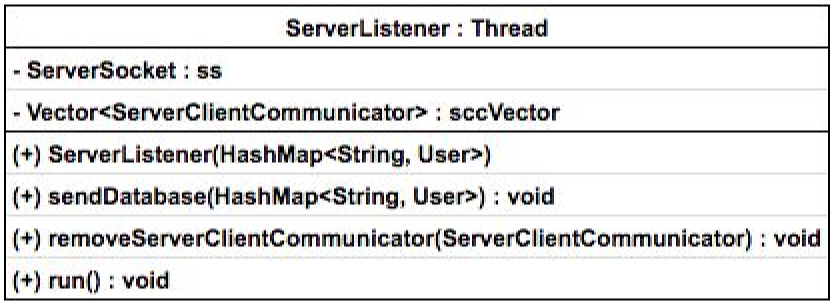
Username	Password
DerekCheng123	tommytrojan
ArfanRehab123	mypassword
NickThompson123	cs201password

Figure 1: Tester selects game database .sql file



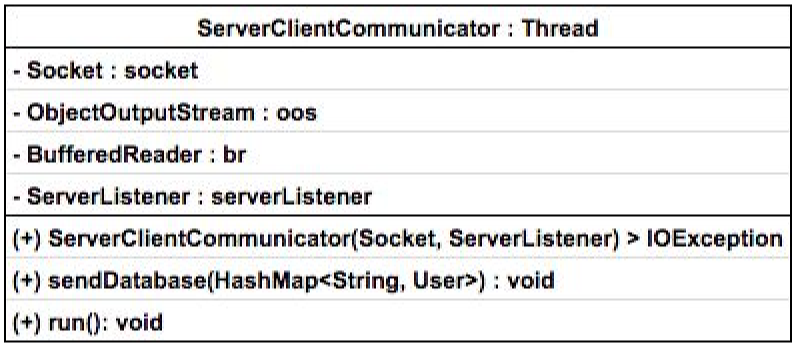
ReadDatabaseListener : ActionListener
- JComboBox<String> : databaseComboBox
- vector<User> : userDatabase
(+) ReadDatabaseListener(Jcombox<String>)
(+) ActionPerformed(ActionEvent) : void
(+) getDatabase() : vector<User>

ReadDatabaseListener - Reads the information from the database that the user chooses on the GameServerGUI. All the username and password information will be displayed on the GameServerGUI.



ServerListener : Thread
- ServerSocket : ss
- Vector<ServerClientCommunicator> : sccVector
(+) ServerListener(HashMap<String, User>)
(+) sendDatabase(HashMap<String, User>) : void
(+) removeServerClientCommunicator(ServerClientCommunicator) : void
(+) run() : void

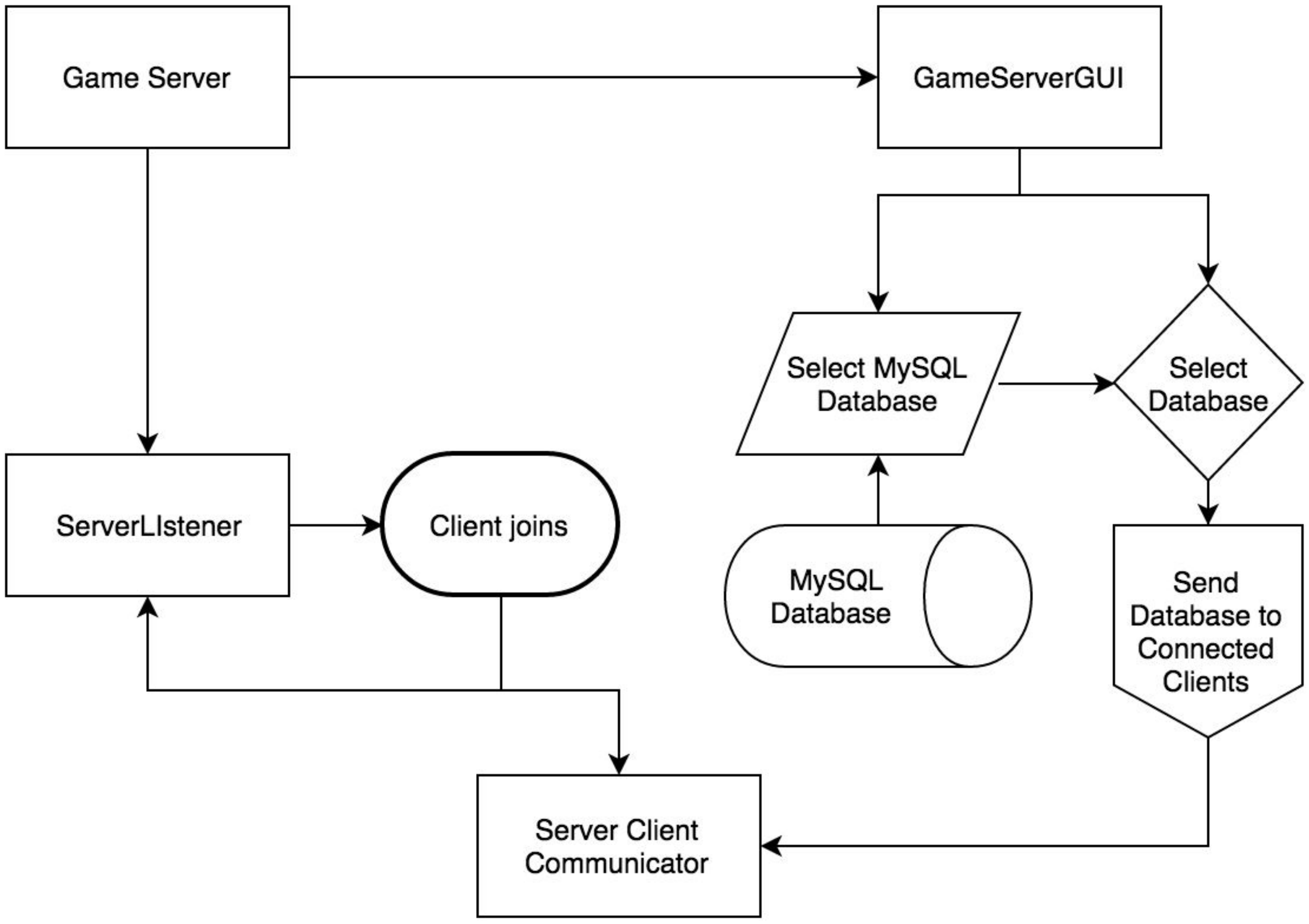
ServerListener - Listens for any connection of the client programs. Once the connection is successfully established, the ServerListener will creates a new socket and passes it to a ServerClientCommunicator.



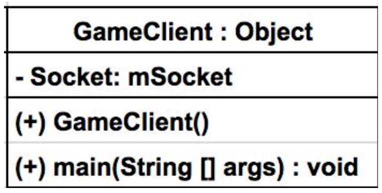
ServerClientCommunicator : Thread
- Socket : socket
- ObjectOutputStream : oos
- BufferedReader : br
- ServerListener : serverListener
(+) ServerClientCommunicator(Socket, ServerListener) > IOException
(+) sendDatabase(HashMap<String, User>) : void
(+) run(): void

ServerClientCommunicator - Maintains a single connection with a specific Client, and sends the user database to the Client.

Server Package: Database Schema (ER Diagram)

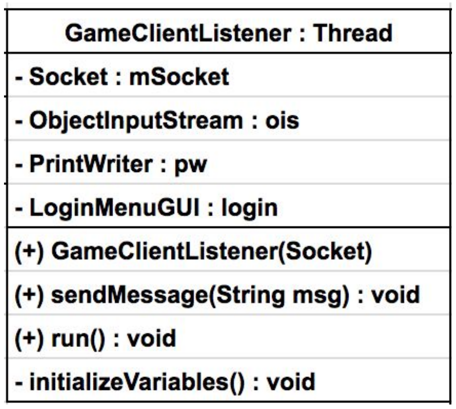


Client Package: Class Diagrams and GUI Mockups



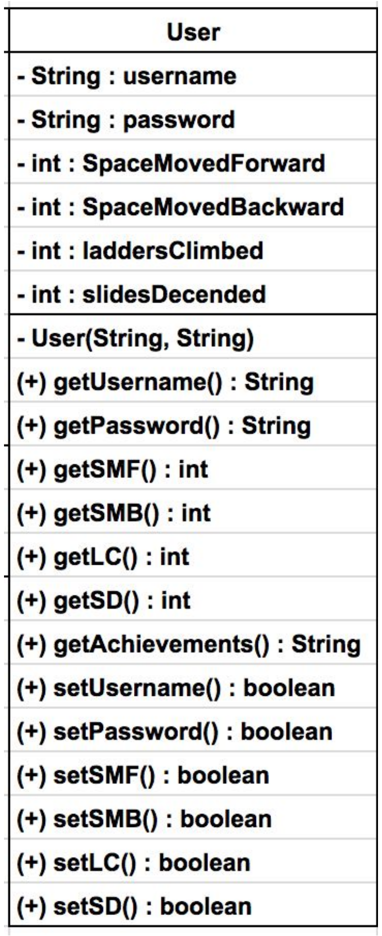
GameClient : Object
- Socket: mSocket
(+) GameClient()
(+) main(String [] args) : void

GameClient - Responsible for creating a GameLobbyGUI with the hardcoded socket.



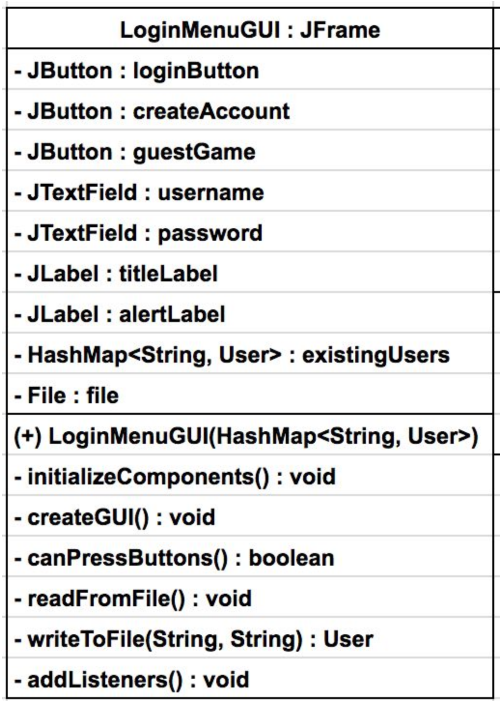
GameClientListener : Thread
- Socket : mSocket
- ObjectInputStream : ois
- PrintWriter : pw
- LoginMenuGUI : login
(+) GameClientListener(Socket)
(+) sendMessage(String msg) : void
(+) run() : void
- initializeVariables() : void

GameClientListener - Listens to the GameServer that the client is connected to waiting for updates.



User
- String : username
- String : password
- int : SpaceMovedForward
- int : SpaceMovedBackward
- int : laddersClimbed
- int : slidesDecended
- User(String, String)
(+) getUsername() : String
(+) getPassword() : String
(+) getSMF() : int
(+) getSMB() : int
(+) getLC() : int
(+) getSD() : int
(+) getAchievements() : String
(+) setUsername() : boolean
(+) setPassword() : boolean
(+) setSMF() : boolean
(+) setSMB() : boolean
(+) setLC() : boolean
(+) setSD() : boolean

User - Class to store user data for users of the application. Contains identity and in game statistics.



LoginMenuGUI : JFrame
- JButton : loginButton
- JButton : createAccount
- JButton : guestGame
- JTextField : username
- JTextField : password
- JLabel : titleLabel
- JLabel : alertLabel
- HashMap<String, User> : existingUsers
- File : file
(+) LoginMenuGUI(HashMap<String, User>)
- initializeComponents() : void
- createGUI() : void
- canPressButtons() : boolean
- readFromFile() : void
- writeToFile(String, String) : User
- addListeners() : void

LoginMenuGUI - Accepts the database from the SeverClientCommunicator, and allows the user to attempt to login to an existing account, create a new one, or join the game as a guest.

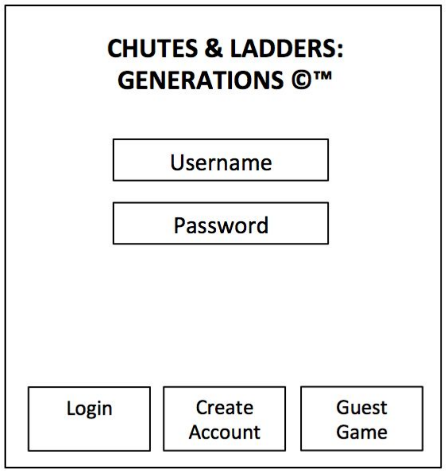
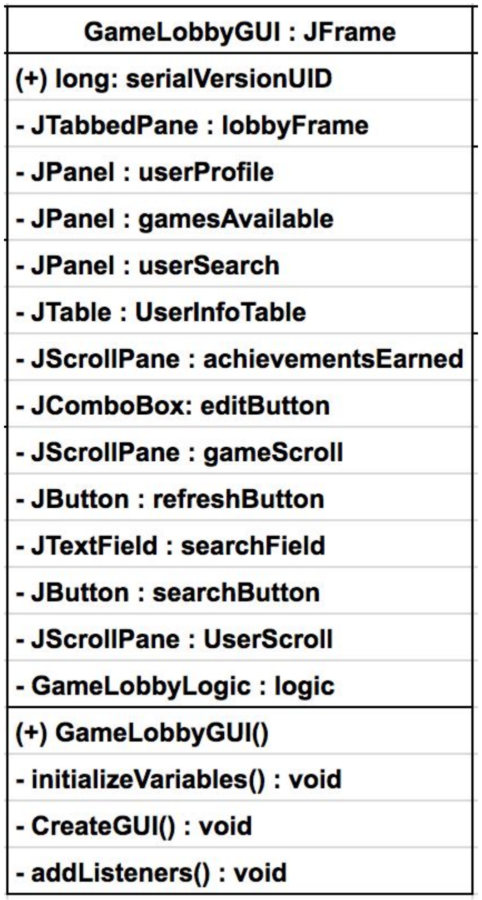
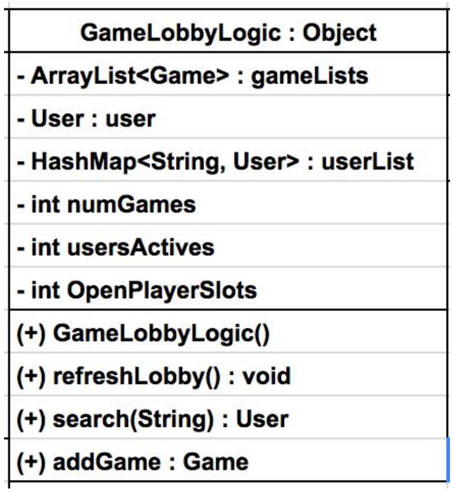


Figure 2: Login screen GUI has text fields for user input and buttons to login, create account, or start a guest game



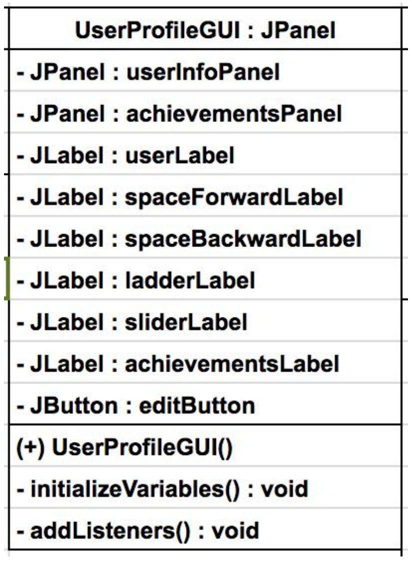
GameLobbyGUI : JFrame
(+) long: serialVersionUID
- JTabbedPane : lobbyFrame
- JPanel : userProfile
- JPanel : gamesAvailable
- JPanel : userSearch
- JTable : UserInfoTable
- JScrollPane : achievementsEarned
- JComboBox: editButton
- JScrollPane : gameScroll
- JButton : refreshButton
- JTextField : searchField
- JButton : searchButton
- JScrollPane : UserScroll
- GameLobbyLogic : logic
(+) GameLobbyGUI()
- initializeVariables() : void
- CreateGUI() : void
- addListeners() : void

GameLobbyGUI - Allows users to view profile and all available games, as well as search the list of all users. Designs what the users sees when views the game lobby. It has three tabs, for viewing the user profile, games available, and user search.



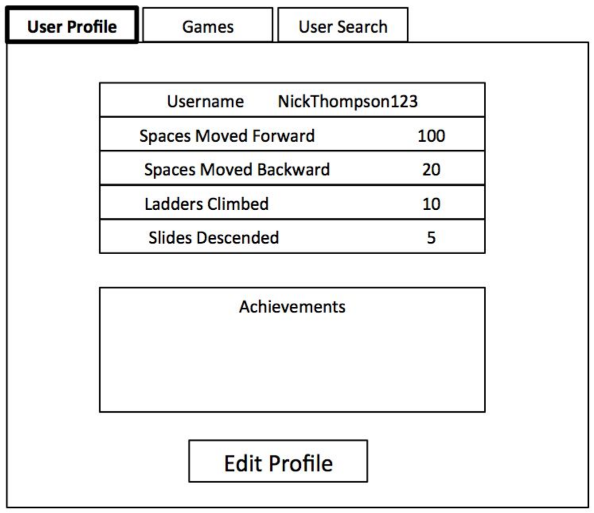
GameLobbyLogic : Object
- ArrayList<Game> : gameLists
- User : user
- HashMap<String, User> : userList
- int numGames
- int usersActives
- int OpenPlayerSlots
(+) GameLobbyLogic()
(+) refreshLobby() : void
(+) search(String) : User
(+) addGame : Game

GameLobbyLogic - Contains list of users and available games. Also populates the game lobby with games so that each active user has a game to join.



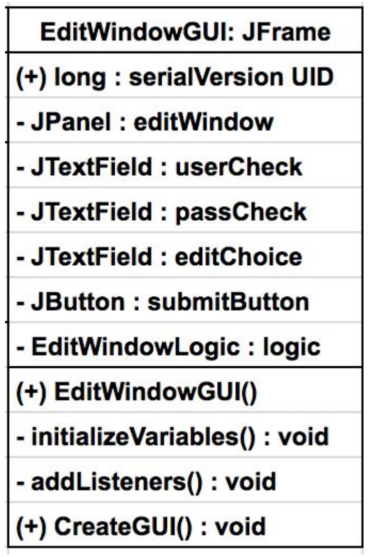
UserProfileGUI : JPanel
- JPanel : userInfoPanel
- JPanel : achievementsPanel
- JLabel : userLabel
- JLabel : spaceForwardLabel
- JLabel : spaceBackwardLabel
- JLabel : ladderLabel
- JLabel : sliderLabel
- JLabel : achievementsLabel
- JButton : editButton
(+) UserProfileGUI()
- initializeVariables() : void
- addListeners() : void

UserProfileGUI - A tab on the LobbyGUI user stats and gives them access to an edit account info button.



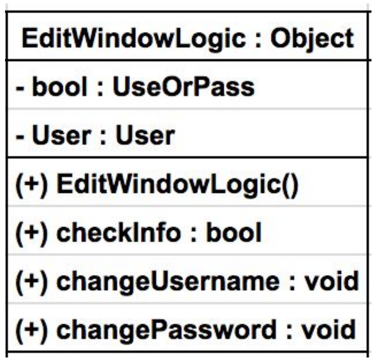
Username	NickThompson123
Spaces Moved Forward	100
Spaces Moved Backward	20
Ladders Climbed	10
Slides Descended	5

Figure 4: The user profile tab shows player stats and achievements and has an edit profile button.



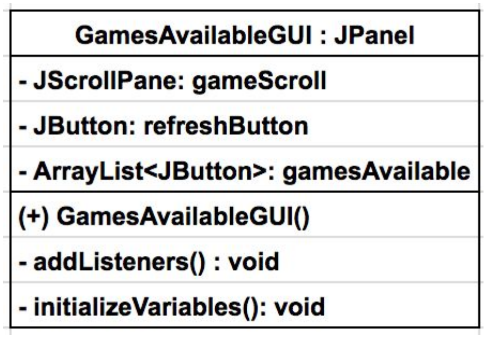
EditWindowGUI: JFrame
(+) long : serialVersion UID
- JPanel : editWindow
- JTextField : userCheck
- JTextField : passCheck
- JTextField : editChoice
- JButton : submitButton
- EditWindowLogic : logic
(+) EditWindowGUI()
- initializeVariables() : void
- addListeners() : void
(+) CreateGUI() : void

EditWindowGUI - Allows users to edit their username and password linked to their account.



EditWindowLogic : Object
- bool : UseOrPass
- User : User
(+) EditWindowLogic()
(+) checkInfo : bool
(+) changeUsername : void
(+) changePassword : void

EditWindowLogic - Allows users to edit the username and password linked to their account.



GamesAvailableGUI : JPanel
- JScrollPane: gameScroll
- JButton: refreshButton
- ArrayList<JButton>: gamesAvailable
(+) GamesAvailableGUI()
- addListeners() : void
- initializeVariables(): void

GamesAvailableGUI - A tab on the LobbyGUI that displays the games available.

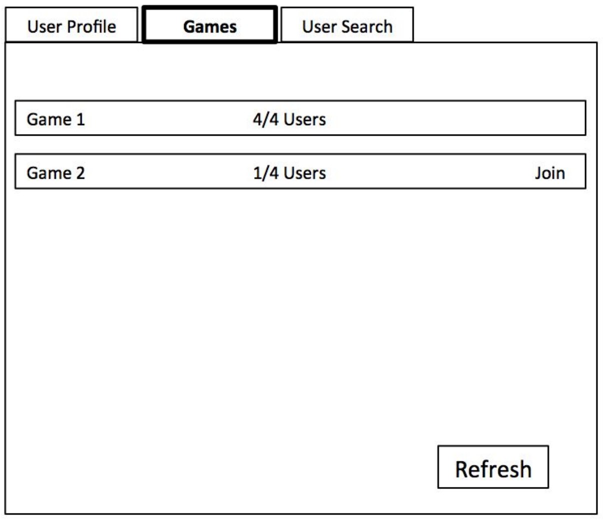
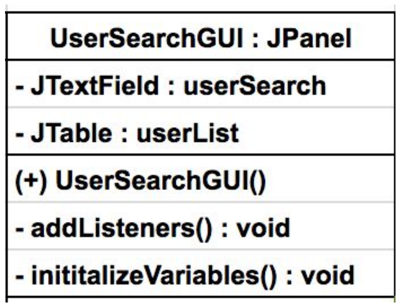
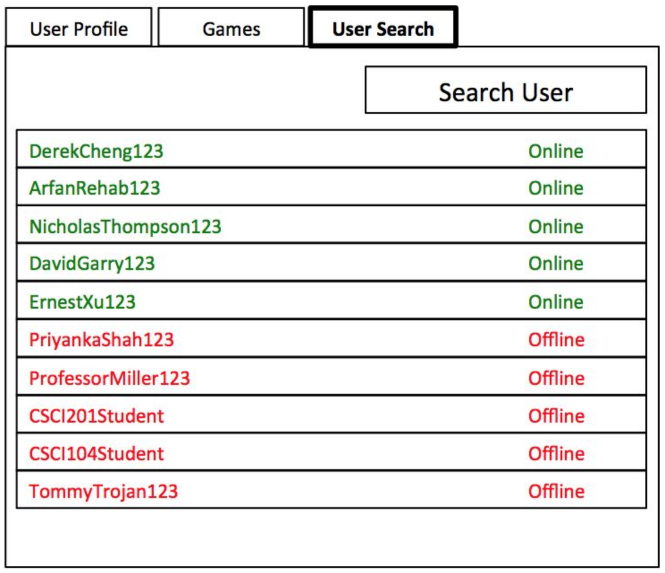


Figure 5: The available games tab has a scroll pane that shows all live games and a refresh button



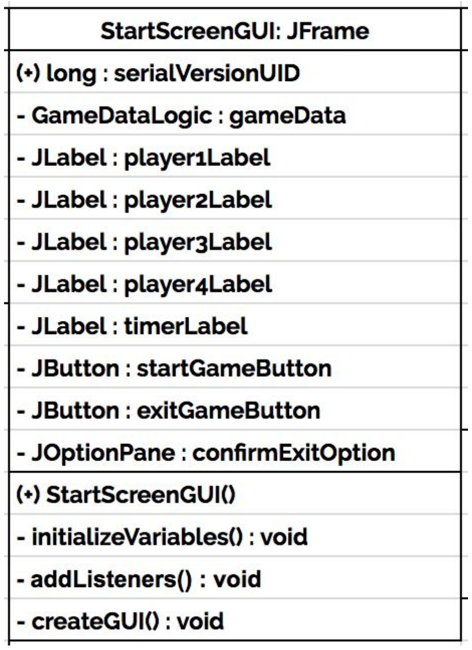
UserSearchGUI : JPanel
- JTextField : userSearch
- JTable : userList
(+) UserSearchGUI()
- addListeners() : void
- inititalizeVariables() : void

UserSearchGUI - A tab on the LobbyGUi that allows users to search through the entire database of users and see the other players that are online.



DerekCheng123	Online
ArfanRehab123	Online
NicholasThompson123	Online
DavidGarry123	Online
ErnestXu123	Online
PriyankaShah123	Offline
ProfessorMiller123	Offline
CSCI201Student	Offline
CSCI104Student	Offline
TommyTrojan123	Offline

Figure 6: User search tab shows all users in a `JScrollPane` and has a search bar to filter the displayed users



StartScreenGUI: JFrame
(+) long : serialVersionUID
- GameDataLogic : gameData
- JLabel : player1Label
- JLabel : player2Label
- JLabel : player3Label
- JLabel : player4Label
- JLabel : timerLabel
- JButton : startGameButton
- JButton : exitGameButton
- JOptionPane : confirmExitOption
(+) StartScreenGUI()
- initializeVariables() : void
- addListeners() : void
- createGUI() : void

StartScreenGUI - The loading page users see before the game has started.

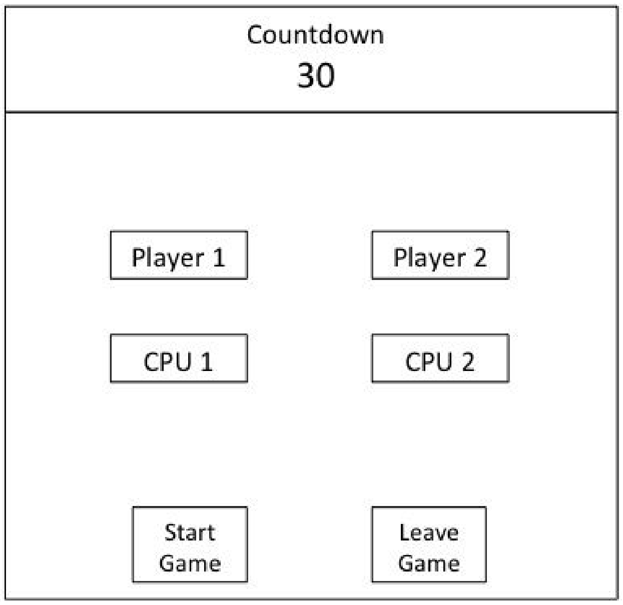
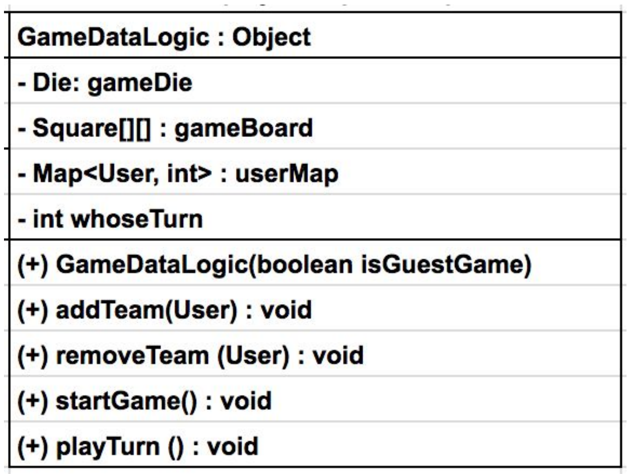


Figure 7: The start screen consists of 4 `JLabel`s for the different players, a `JLabel` for countdown to start of game, and `JButton`s for start and leave game.



GameDataLogic : Object
- Die: gameDie
- Square[][] : gameBoard
- Map<User, int> : userMap
- int whoseTurn
(+) GameDataLogic(boolean isGuestGame)
(+) addTeam(User) : void
(+) removeTeam (User) : void
(+) startGame() : void
(+) playTurn () : void

GameDataLogic - The game Data is responsible for rolling the dice for every player's turn, and updates each player's positions accordingly. It includes a gameBoard and a map to keep track of the user's position based on whether he lands on a chute or ladder. When a user enters the game, the user is added to the userMap with the addTeam() method. Consequently, when a user exits the game, they are removed with the removeTeam() method. When the game is started from the StartScreenGUI, startGame() is called to randomly generate a gameBoard array (10x10 for registered game, 5x5 for guest game) of Squares, and fill the userMap with CPU users if necessary. The playTurn() method is used to have the current player roll the dice, update their position in the userMap, and update whoseTurn so the next player is queued up.

GameScreenGUI: JFrame
(+) long : serialVersionUID
- GameDataLogic : gameData
- JPanel : gameBoard
- JLabel : player1Label
- JLabel : player2Label
- JLabel : player3Label
- JLabel : player4Label
- JLabel : timerLabel
- JLabel : whoseTurnLabel
(+) GameScreenGUI(GameData gameData)
- initializeVariables() : void
- addListeners() : void
- createGUI() : void

GameScreenGUI - The game screen that displays and handles game play.

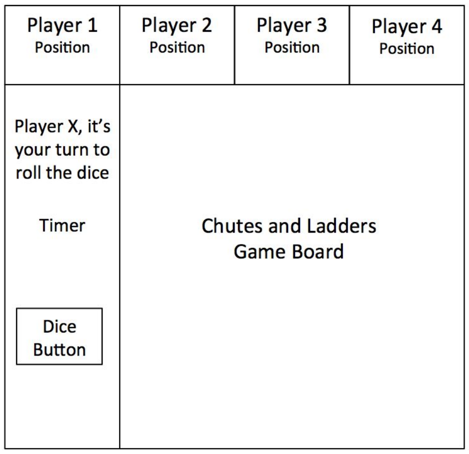
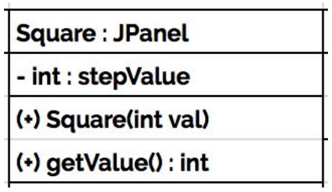
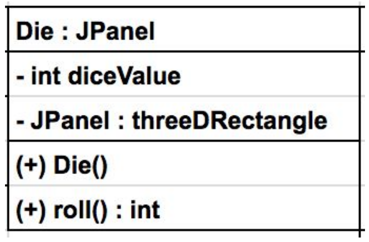


Figure 8: The gameplay screen GUI with the game board squares and the dice.



Square : JPanel
- int : stepValue
(+) Square(int val)
(+) getValue() : int

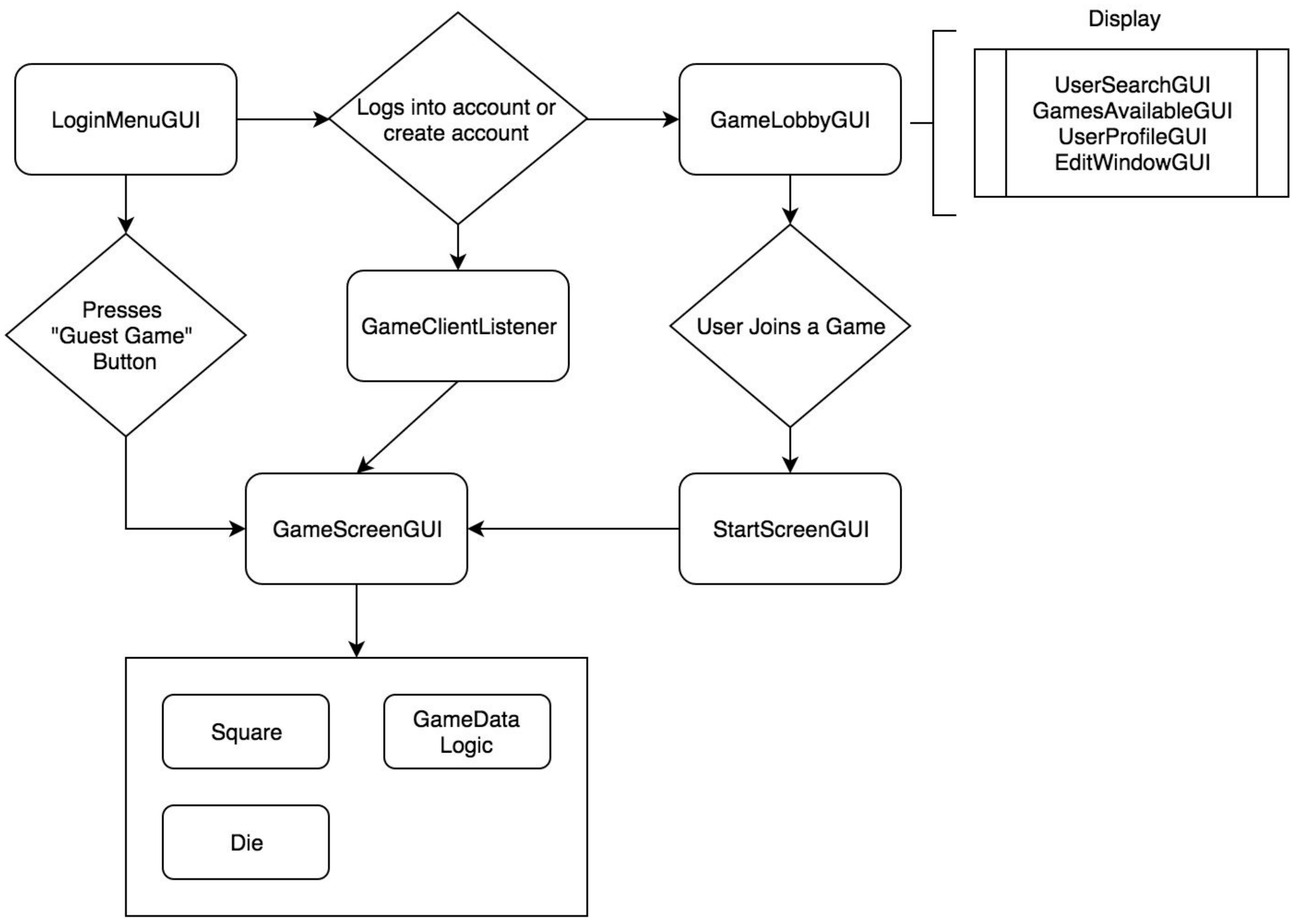
Square - Square class constructor takes an int to determine the square's special effect (0 for normal square, positive value for ladder, negative value for chute).



Die : JPanel
- int diceValue
- JPanel : threeDRectangle
(+) Die()
(+) roll() : int

Die - When the dice is called with the roll(), it randomly updates diceValue to a value from 1-6 and returns diceValue.

Client Package: Schema (ER Diagram)



CSCI-201L: Team “Oh Chute”
Arfan Rehab, David Garry, Derek Cheng, Jiachang (Ernest) Xu, Nicholas Thompson
Project Name: *Chutes & Ladders, Generations*
Course Producer: Priyanka Shah
Lecture Section: TTh, 11:00am - 12:20pm
Due: Sunday, November 6th, 2016

Test Cases

Server & Networking:

Test #	**1**
Test Description	Only one server should be able to bind to a port at a time
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Run another game server
	4. Press "Start Listening", Port specified should be default 6789
Expected Result	The first server should start, while the second server gives an error message of "Port already in use".
Actual Result	TBD
Screenshot	TBD

Test #	**2**
Test Description	A server loads in a correctly formatted .sql database file
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
Expected Result	Server should output all the username and password on the GUI
Actual Result	TBD
Screenshot	TBD

Test #	**3**
Test Description	A server loads in a badly formatted .sql database file
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select baduser.sql from the menu
	4. Pressss "Select Databse"
Expected Result	Server should print the errors to the console, and ask the user to re-select a database
Actual Result	TBD
Screenshot	TBD

Test #	**4**
Test Description	A server should send its database to client upon connection
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
Expected Result	The server acknowledges the client, and sends the database
Actual Result	TBD
Screenshot	TBD

Test #	**5**
Test Description	A server should send its database to a client after connection
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Run a game client
	4. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	5. Select user.sql from the server menu
	6. Press "Select Database"
Expected Result	The server sends the database to the client after connection
Actual Result	TBD
Screenshot	TBD

Test #	**6**
Test Description	A client should not be able to connect to a bad host
Steps to Run Test	1. Run a game client
	2. Specify Hostname to be 'badhost'
	3. Post speficied shoud be default 6789
	4. Press "Connect"
	5. Wait a few seconds
Expected Result	The client's request times out, and an error message "Unable to connect to host" is displayed

Actual Result	TBD
Screenshot	TBD

Test #	**7**
Test Description	A client should not be able to connect to a bad port
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port speficied should be default 6789
	3. Run a game client
	4. Speficity Hostname to be 'localhost'
	5. Port specified should be custom 6788
	6. Press "Connect"
	7. Wait a few seconds
Expected Result	The client's request times out, and an error message "Unable to connect to host" is displayed
Actual Result	TBD
Screenshot	TBD

<u>**Login:**</u>

Test #	**8**
Test Description	Non-existing username when attempting to log in
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'invalid' (non-existing username), Password specified should be 'csci201' (arbitrary password)
	8. Press "Login"
Expected Result	An error message "This username doesn't exist" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**9**
Test Description	Wrong password when attempting to log in
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'valid' (existing username), Password specifiecd should be 'incorrect' (incorrect password)
	8. Press "Login"
Expected Result	An error message "Incorrect password" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**10**
Test Description	Correct username but incorrect password when attempting to log in

Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'valid' (existing username), Password specifiecd should be 'incorrect' (incorrect password)
	8. Press "Login"
Expected Result	An error message "Incorrect password" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**11**
Test Description	Username does not meet criteria when attempting to create account
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'abc' or '123' (alphabet or number-only username), Password specified should be 'csci201' (arbitrary password)
	8. Press "Create Account"
Expected Result	An error message of "Username does not meet criteria" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**12**
Test Description	Password does not meet criteria when attempting to create account
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"

	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'valid201' (valid username), Password specified should be 'abc' or '123' (alphabet or number-only password)
	8. Press "Create Account"
Expected Result	An error message of "This password does not meet criteria" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**13**
Test Description	Username already exists when attempting to create account
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Username specified should be 'valid 201' (existing username), Password specified should be 'csci201' (arbitrary password)
	8. Press "Create Account"
Expected Result	An error message "This username already exists" should be displayed
Actual Result	TBD
Screenshot	TBD

Test #	**14**
Test Description	Local game begins correctly without username nor password
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect", Port specified should be default 6789 and Hostname specified should be 'localhost'
	7. Leave Username and Password empty

	8. Press "Login"
Expected Result	The client should start a non-networked game without error
Actual Result	TBD
Screenshot	TBD

User Profile

Test #	**15**
Test Description	Correct User Profile Information
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
Expected Result	The game lobby shows the correct username, ladders and sliders climbed/descended and achievements in the user profile tab.
Actual Result	TBD
Screenshot	TBD

Test #	**16**
Test Description	Edit Username works when password/username entered correctly, and username criteria is met
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Username"
Expected Result	The username displayed in the profile tab shows the newest one.
Actual Result	TBD
Screenshot	TBD

Test #	**17**

Test Description	Edit Username does not work when password/username entered incorrectly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Username"
Expected Result	An error message should be displayed "Username/password combination does not exist"
Actual Result	TBD
Screenshot	TBD

Test #	**18**
Test Description	Edit Password works when password/username entered correctly, and password criteria is met
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Password"
Expected Result	When logging in the next time, the old password associated with the current username should display an error message and the new one should work correctly.
Actual Result	TBD
Screenshot	TBD

Test #	**19**

Test Description	Edit Password does not work when password/username entered incorrectly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Password"
Expected Result	An error message should be displayed "Username/password combination does not exist"
Actual Result	TBD
Screenshot	TBD

Test #	**20**
Test Description	Edit Username does not work when password/username entered correctly, and username criteria is not met
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Username"
Expected Result	Display an error message "Username already exists in the database/requires 4 or more alphanumeric characters/symbols"
Actual Result	TBD
Screenshot	TBD

Test #	**21**

Test Description	Edit Password does not work when password/username entered correctly and password criteria is not met
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press "Edit Profile"
	10. Choose "Edit Password"
Expected Result	Display an error message "Password requires 6 or more alphanumeric characters/symbols"
Actual Result	TBD
Screenshot	TBD

Test #	**22**
Test Description	User achievements are updated properly after won in gameplay
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Choose an active game
Expected Result	The achievements textbox displays: games won, trophies earned, etc.
Actual Result	TBD
Screenshot	TBD

Test #	**23**
Test Description	User statistics are updated properly after each game

Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Choose an active game
Expected Result	The following is updated after a game has been played/won: spaced moved forward, spaced moved backward, ladders climbed, slides descended
Actual Result	TBD
Screenshot	TBD

Test #	**24**
Test Description	Program exits properly from lobby
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", port should be 6789
	3. Select user.sql from menu
	4. Press Select Database
	5. Run a game client
	6. Press "Connect"
	7. Username and Password specified should be valid
	8. Press "Login"
	9. Press the X button on the lobby
Expected Result	The program closes the game lobby window and all other running applications within the server
Actual Result	TBD
Screenshot	TBD

Game Lobby, Games Available, UserSearch:

Test #	**25**
Test Description	Clicking on an available game brings you into that game's start screen
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "Games Available" tab
	9. Click on first available game button
Expected Result	View of that game's start screen
Actual Result	TBD
Screenshot	TBD

Test #	**26**
Test Description	Clicking on a full game does not bring you into that game
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Repeat (1-7) with second user(u2)
	9. (u2)Click on "Games Available" tab
	10. (u2)Click on first available game button
	11. (u2) After timer expires, start game with CPUs
	12. (u1)Click on "Games Available tab
	13. (u1)Click on full game button
Expected Result	Error Message: "That game is full, cannot join"
Actual Result	TBD
Screenshot	TBD

Test #	**27**

Test Description	Refresh Button Shows Updated List
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Repeat (1-7) with second user(u2)
	9. (u1)Click on "Games Available tab
	10. (u2)Click on "Games Available" tab
	11. (u2)Click on first available game button
	12. (u1)Click "Refresh" Button
Expected Result	Games available shows a user has joined one of the games in the game lobby
Actual Result	TBD
Screenshot	TBD

Test #	**28**
Test Description	If the number of users available is greater than available player slots because a player joined, a new game is created
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Repeat (1-7) with three more users(u2-u4)
	9. (u1-u4)Click on "Games Available" tab
	10. Repeat (1-7) with fifth user (u5)
	11. (u1) Press "Refresh" Button
Expected Result	A games available screen with two open games that are empty
Actual Result	TBD
Screenshot	TBD

Test #	**29**
Test Description	If the number of users available is greater than available player slots because a game is started with CPUs, a new game is created
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Repeat (1-7) with second user(u2)
	9. (u1)Click on "Games Available tab
	10. (u2)Click on "Games Available" tab
	11. (u2)Click on first available game button
	12. (u2)After time expires, start game with CPUs
	12. (u1)Click "Refresh" Button
Expected Result	Games available shows full game as well as a new game empty game to join
Actual Result	TBD
Screenshot	TBD

Test #	**30**
Test Description	If a game ends, it dissapears from the lobby and the users availables updates accordingly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "Games Available" tab
	9. Click on first available game button
	10. After timer expires, start game with CPUs
	11. Play full game, Pressing "Roll" until someone wins
	12. Return to Lobby
Expected Result	Game lobby screen without the previous game in the scrollpane

Actual Result	TBD
Screenshot	TBD

Test #	**31**
Test Description	User list shows users correctly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "User Search" tab
Expected Result	Scrollpane with names: TestUser, DerekChang123, ArfanRehab123, NicholasThompson123, DavidGarry123, ErnestXu123, PriyankaShah123, ProfessorMiller123, CSCI201Student, CSCI104Student, TommyTrojan123
Actual Result	TBD
Screenshot	TBD

Test #	**32**
Test Description	Search Feature Finds Username Correctly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "User Search" tab
	9. Enter "ProfessorMiller123" into search bar
	10. Press "Search"
Expected Result	A single "ProfessorMiller123" account button remaining on user search tab
Actual Result	TBD
Screenshot	TBD

Test #	**33**
Test Description	Search feature does not find unexistent username
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "User Search" tab
	9. Enter "NotAUser123" into search bar
	10. Press "Search"
Expected Result	Any empty scroll pane on the user search window
Actual Result	TBD
Screenshot	TBD

Test #	**34**
Test Description	Clicking on a User Profile Searched brings to guest view of that profile
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "User Search" tab
	9. Enter "ProfessorMiller123" into search bar
	10. Press "Search"
	11. Click on "ProfessorMiller123" profile button
Expected Result	Veiw of "ProfessorMiller123" guestprofile which includes username, stats, and achievements, but not password or edit capabilities
Actual Result	TBD
Screenshot	TBD

Test #	**35**
Test Description	User list shows active status of users correctly
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login Using test user with correct username and password
	8. Click on "User Search" tab
Expected Result	Scrollpane with names: In Green: TestUser In Red: DerekChang123, ArfanRehab123, NicholasThompson123, DavidGarry123, ErnestXu123, PriyankaShah123, ProfessorMiller123, CSCI201Student, CSCI104Student, TommyTrojan123
Actual Result	TBD
Screenshot	TBD

Local Gameplay & Die:

Test #	**36**

Test Description	The board generated is 5x5 for local games
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
Expected Result	The game should start, with a grid that should have 25 squares total, with both a length and width of 5
Actual Result	TBD
Screenshot	TBD

Test #	**37**
Test Description	The game has 3 CPUs for local games
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
Expected Result	The game should start with 3 CPUs, and the human player should get to go first
Actual Result	TBD
Screenshot	TBD

Test #	**38**
Test Description	Each player moves the number of spaces designated on a die
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll the die
	4. Count the spaces that the character moves
Expected Result	The player should move the correct number of spaces as indicated by the die. Run test on both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**39**
Test Description	If a player lands on a ladder bottom square, they move to the top of the ladder

Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until the bottom of a ladder is reached
Expected Result	When the player stops on the bottom square, the player should be automatically moved to the square corresponding to the top of the ladder. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**40**
Test Description	If a player lands a slide top, they move to the bottom of the slide
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until the top of a ladder is reached
Expected Result	When the player stops on the top of a slide square, the player should be automatically moved to the square corresponding to the bottom of the slide. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**41**
Test Description	If the player lands on a normal space, nothing happens
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until a normal space is landed on.
Expected Result	When the player stops on a normal square (including the top of a ladder or bottom of a slide), the player should not move anywhere, and the next player should get to go. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**42**
Test Description	The first player to reach square 25 Wins

Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until a player reaches square 25
Expected Result	When the player is on a square whose value added to the roll die is greater than or equal to 25, the player should stop on square 25 and the game should be declared ended.
Actual Result	TBD
Screenshot	TBD

Test #	**43**
Test Description	The players turns progress in normal order
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until the game has ended, noting the order of players.
Expected Result	The order of the players should never change, and the first person to go should be the human player.
Actual Result	TBD
Screenshot	TBD

Test #	**44**
Test Description	When a player wins, the game ends
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button
	3. Roll die until a player reaches square 25.
Expected Result	The game should end when a player reaches 25, showing the winner's name and then exiting the application (for local game only).
Actual Result	TBD
Screenshot	TBD

Test #	**45**
Test Description	Die rolls when die button is pressed
Steps to Run Test	1. Run a Game Client
	2. Select "Guest Game" button

	3. Roll die
Expected Result	The die gif animation should begin, and then a number should appear on the screen showing the value that was rolled.
Actual Result	TBD
Screenshot	TBD

Test #	**46**
Test Description	Die is truly random (unit test)
Steps to Run Test	1. Run the int rollDie() method 1,000,000 times inside a loop, storing results in an array of length 7 by incrementing the occurrence of each position
	4. Print out the contents of the array after the loop
Expected Result	The array should contain an equal (+/- 0.5%) number of 1,2,3,4,5, and 6 with no occurrence of 0.
Actual Result	TBD
Screenshot	TBD

<u>Networked Gameplay:</u>

Test #	**47**
Test Description	Start screen is filled with correct # of CPUs
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
Expected Result	1 user and 3 CPU players in the start screen
Actual Result	TBD
Screenshot	TBD

Test #	**48**

Test Description	If more users enter, the CPU will be replaced by user
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Repeat steps 1-9 for a second user
Expected Result	First user will see a CPU player replaced with another user, so there will be 2 humans and 2 CPUs after step 10
Actual Result	TBD
Screenshot	TBD

Test #	**49**
Test Description	If a player lands a slide top, they move to the bottom of the slide
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it is user's turn, click on the die image to roll die
	12. Repeat step 12 until user lands on top of a slide
Expected Result	When the player stops on the top of a slide square, the player should be automatically moved to the square corresponding to the bottom of the slide. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**50**
Test Description	If a player lands a ladder bottom, they move to the top of the ladder
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it is user's turn, click on the die image to roll die
	12. Repeat step 12 until user lands on bottom of a ladder
Expected Result	When the player stops on the bottom square, the player should be automatically moved to the square corresponding to the top of the ladder. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**51**
Test Description	The board generated is 10 X 10
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
Expected Result	The game should start, with a grid that should have 100 squares total, with both a length and width of 10

Actual Result	TBD
Screenshot	TBD

Test #	**52**
Test Description	When the timer runs out, the player's turn label updates
Steps to Run Test`	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it becomes the user's turn, do nothing and let 10 second timer run out
Expected Result	The game should cycle to the next CPU player and update the player's turn label to show it is no longer the user's turn
Actual Result	TBD
Screenshot	TBD

Test #	**53**
Test Description	Each player moves the number of spaces designated on a die
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen

	11. When it is user's turn, click on the die image to roll die
Expected Result	Player should initially move the number of spaces equal to the number shown by the die (spaces moved before slides and ladders). Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**54**
Test Description	If the player lands on a normal space, nothing happens
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it is user's turn, click on the die image to roll die
	12. Repeat step 12 until user lands on a normal space
Expected Result	When the player stops on a normal square (including the top of a ladder or bottom of a slide), the player should not move anywhere, and the next player should get to go. Should pass for both human and machine players.
Actual Result	TBD
Screenshot	TBD

Test #	**55**
Test Description	The first player to reach square 100 wins
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"

	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it is user's turn, click on the die image to roll die
	12. Play game and repeat step 12 until a player (user or CPU) has reached square 100
Expected Result	When the player is on a square who's value added to the roll die is greater than or equal to 100, the player should stop on square 100 and the game should be declared ended.
Actual Result	TBD
Screenshot	TBD

Test #	**56**
Test Description	The players turns progress in normal order
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. Roll die until the game has ended, noting the order of players
Expected Result	The order of the players should never change, and the first person to go should be the human player.
Actual Result	TBD
Screenshot	TBD

Test #	**57**

Test Description	When a player wins the game ends
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. Roll die until a player reaches square 100.
Expected Result	The game should end when a player reaches 100, showing the winner's name and exiting back to the game lobby
Actual Result	TBD
Screenshot	TBD

Test #	**58**
Test Description	After the game each users stats and achievements are updated
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "User Profile" tab and take note of number of games played, number of games won, and achievements
	9. Click on "Games Available" tab
	10. Select an available game
	11. Start the new game with CPUs from the start screen
	12. Roll die until the game has ended
	13. Game will exit back to the lobby screen
	14. Click on "User Profile" tab

Expected Result	User profile should display the updated and correct games played, games won, and achievements to reflect the events of the game (+1 games played, +0/+1 games won)
Actual Result	TBD
Screenshot	TBD

Test #	**59**
Test Description	Die rolls when die button is pressed
Steps to Run Test	1. Run a game server
	2. Press "Start Listening", Port specified should be default 6789
	3. Select user.sql from the menu
	4. Press "Select Database"
	5. Run a game server
	6. Press "Connect"
	7. Login using test user with correct username and password
	8. Click on "Games Available" tab
	9. Select an available game
	10. Start the new game with CPUs from the start screen
	11. When it is user's turn, click on the die image to roll die
Expected Result	The die gif animation should begin, and then a number should appear on the screen showing the value that was rolled.
Actual Result	TBD
Screenshot	TBD

CSCI-201L: Team "Oh Chute"
Arfan Rehab, David Garry, Derek Cheng, Jiachang (Ernest) Xu, Nicholas Thompson
Project Name: *Chutes & Ladders, Generations*
Course Producer: Priyanka Shah
Lecture Section: TTh, 11:00am - 12:20pm
Due: Sunday, November 13th, 2016

Deployment Document

To deploy this application within Eclipse, import the ChutesAndLaddersGenerations.zip file into Eclipse. This should generate a project called ChutesAndLaddersGenerations with src and resources directories. The src directory contains the gameserver, gameclient, packages. The gameserver directory contains the java files pertaining to the server side. The gameclient directory contains the java files pertaining to the client side.

To execute the Game Server, run gameserver.GameServer.

To execute the Game Client, run gameclient.GameClient.