Bilkent University

Department of Computer Engineering

CS319 Term Project

Project short-name: Scrolls of Estatia

Group 3F

Final Report

Atakan Sağlam, Sarp Ulaş Kaya, Berk Kerem Berçin, Oğulcan Çetinkaya, Furkan Başkaya

Instructor: Eray Tüzün
Teaching Assistant(s): Barış Ardıç, Emre Sülün and Elgun Jabrayilzade

Final Report
Dec 20, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS319.

Contents

Final Report

Project short-name: Scrolls of Estatia

1. Introduction

Basic game logic of our game is mostly complete. The only feature we have yet to finish the implementation of is trade function. We have implemented all of our classes and their respective methods. In our Model-View-Controller design, the model part is complete. Our implementation of the basic game logic and the classes are mostly matching with our design. We added ColorGroup class and TradeRequest class to handle grouping towns in color groups to make the build function work properly and to be able to construct a trade request on the client side, send it to the server and store it in a token respectively. However, our implementation of the view part differed drastically from our design. We have added multiple classes such as InfoLabel, StyledButton, Subscene, ViewManager, GameViewManager and SquareVisual. These classes are used for encapsulating and modifying our UI components in an orderly way. However, we are still missing some parts of our UI such as our version of the game board, which is currently temporarily replaced with an empty template of the classic Monopoly board as a placeholder. Other than that, the majority of the work left for our project is for connecting the model and view with our controller i.e., connecting our game logic and the UI with the help of our server-player communications.

2. Lessons Learnt

We learned how to work on a project with a group and solve the problems that we face using proper teamwork. We learned the importance of planning ahead. One of the key tools that we used for our planning and organization was GitHub. We realized that things may not always go according to plan and that teamwork is essential to solve unexpected problems in project development. Since these problems can extend the duration of the project's development significantly, we realized that proper work distribution was also important for developing large projects with an object-oriented structure and tens of classes as efficiently as possible, especially with tight deadlines.

We learned the benefits of iterative development as drawing UML diagrams before the implementation helped us a lot while coding. However, we realized that it was also very difficult to construct the diagrams perfectly and we encountered problems that required us to make changes to the designs in these diagrams. We saw how the Strategy Design Pattern made the implementation of our Traveler class easier. Same goes to the Singleton Pattern with the implementation of our Game class. Overall, we have experienced the effects of design patterns first-hand on software development and grasped their importance better.

Using the Model-View-Controller (MVC) Architectural Pattern, we were able to distribute the design of the game logic, user interface, and server-client communications among the group members. This allowed us to work on the project faster as each member was able to work on a separate part of the project simultaneously, and helped us grasp the importance of using architectural patterns in the planning phase of a project more clearly. Using JavaFX, we learned how to design, create and test custom UI components for our project. We designed a dynamic user interface which responds to user inputs and affects the user experience in a positive way. We learned how to use Java Sockets to build an online multiplayer game, and how to build a server that can manage multiple connections from clients. We learned how important it is to carefully manage the ObjectOutputStream and DataOutputStream so that there aren't any synchronization issues between the server and the client.

3. User Manual

Upon launch, you will be greeted with a main menu, which has buttons to host a game, join a game, see the game's tutorial, open the settings menu, show credits and exit the game.

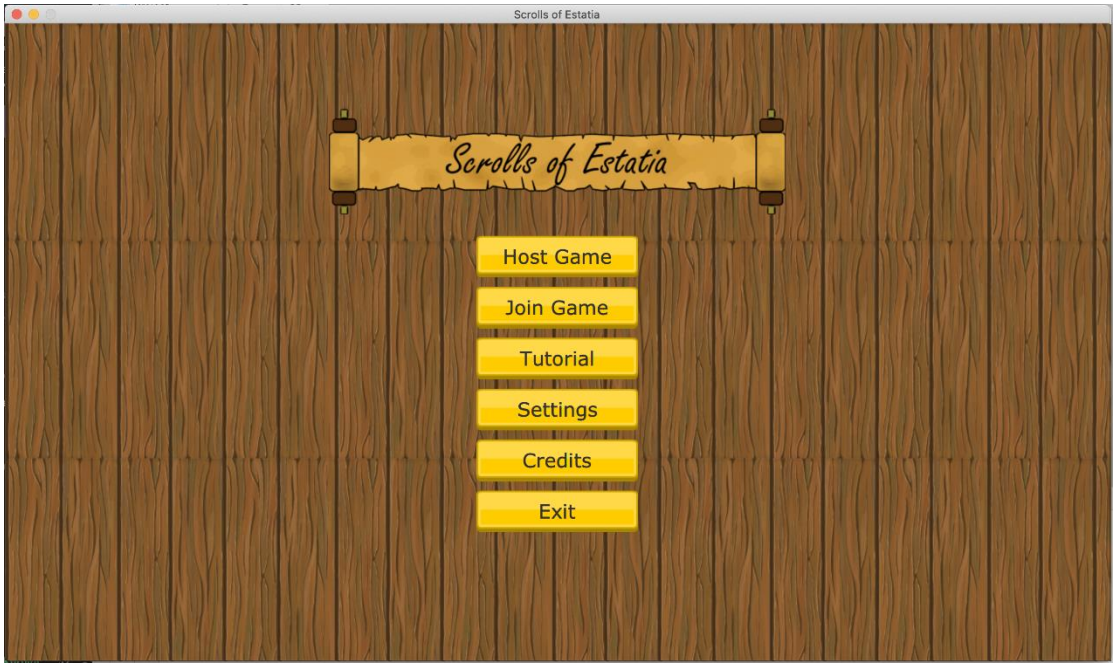

Figure 1: Main Menu Screen.

If you wish to host a game, you must click the "Host Game" button. Then, you have to first select the number of players for the game's lobby. After doing so, you can host the game.

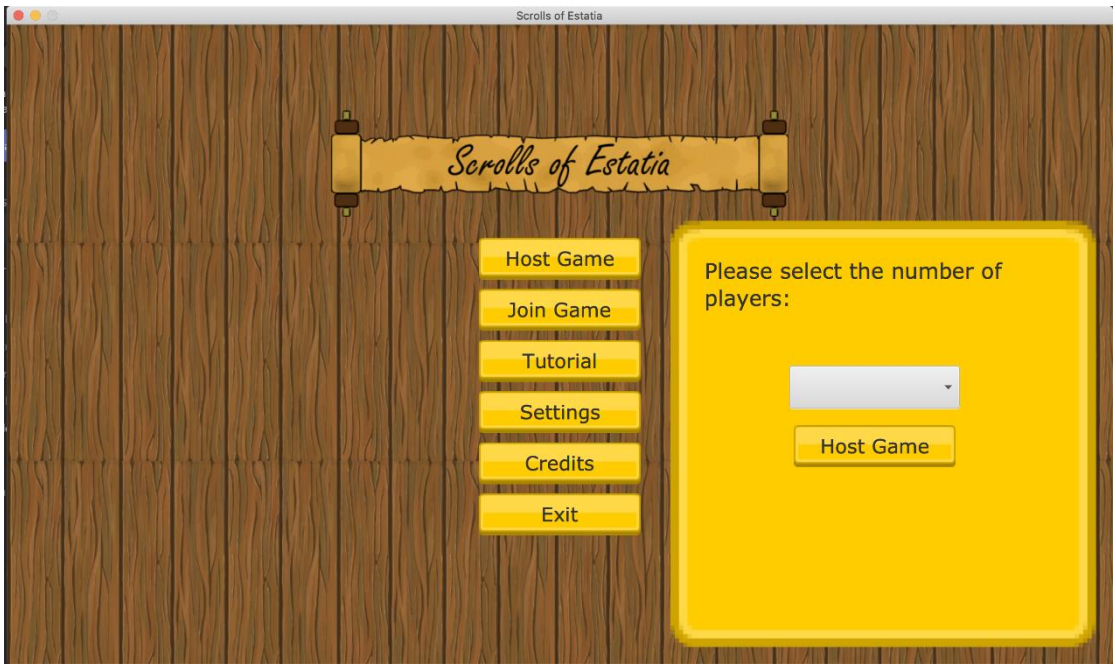

Figure 2: Host Game Sub-menu.

If you wish to join a game, you have to first retrieve the ID of the game lobby from the game's host. After receiving the ID, you can click the "Join Game" button and enter the game ID in the textbox accordingly. You can then join the game.

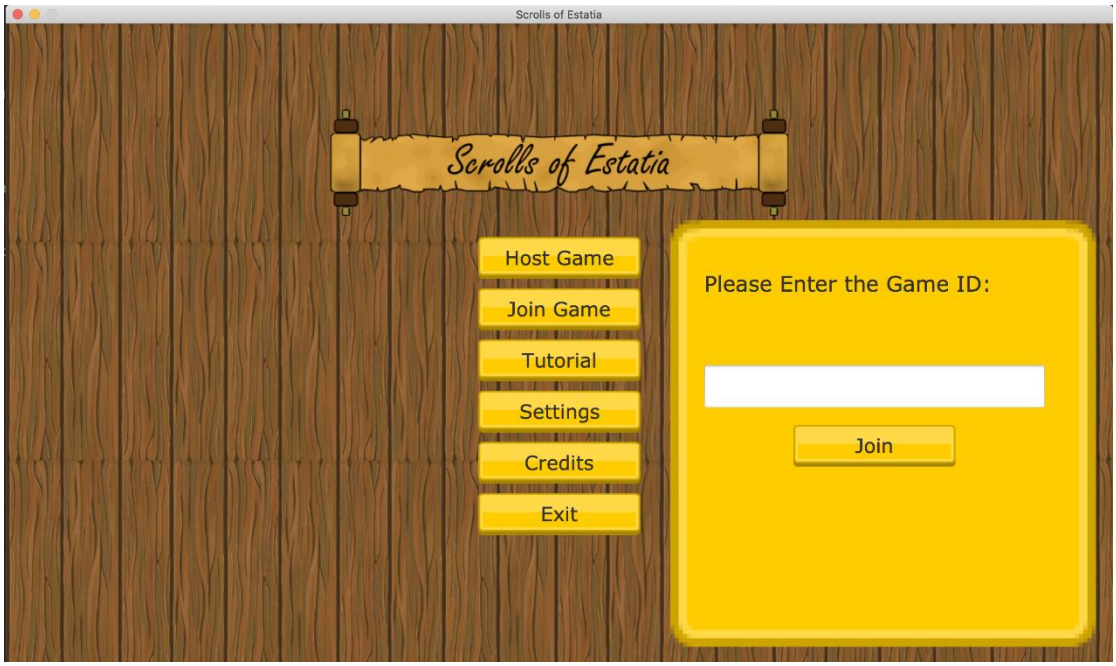

Figure 3: Join Game Sub-menu.

In the game lobby, you can select your class using the drop-down menu on the bottom left. There are 9 different classes for you to chose from with one of them having two different strategies. The classes that each player in the lobby picked appears on the list above the drop-down menu, where each player is represented with their player number and their class. Your class is shown with a "[YOU]" text next to it. Among all the players, only the host has access to the "Start Game" button. The game ID is also shown so that you or the other players can invite other players to the lobby. You and all the other players also have the option to leave the lobby using the "Leave Lobby" button.

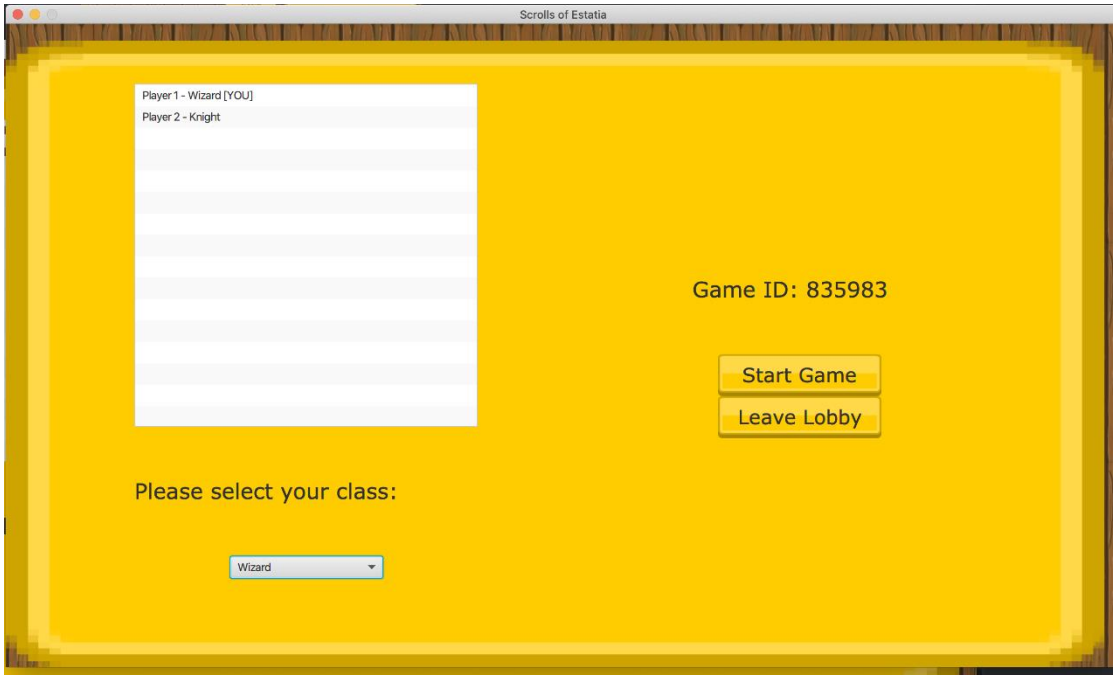

Figure 4: Game Lobby.

Based on the class you pick, you will get different perks for your token:

- Picking the traveler class will let you move one extra square once every two turns or three extra squares every five turns based on the movement strategy you pick.
- Picking Noble will have you start out with extra money and pay less tax throughout the game.
- Picking Knight will make it so that you pay less rent throughout the game.
- Picking Treasure Hunter will grant you with a random Fortune Card, which is a card whose effects activate instantly, in each turn.
- Picking Wizard will grant you with a Mana Bar, which will grant you a random Scroll, which is a card that you can keep in your hand and use in one of your turns, each time it gets full.
- Picking Fortune Teller will let you pick between two cards each time you get to draw a Fortune Card or a Scroll by landing on a Scroll or Fortune square.
- Picking Thief will let you steal money from the other players each time you land on the same square as them.
- Picking Builder will let you build on their properties for cheaper.
- Picking Cardinal will let you build Chuches and Cathedrals instead of the Inns and Mansions that other classes build. These buildings will increase the rent of the property they are built on by a larger factor than Inns and Mansions.

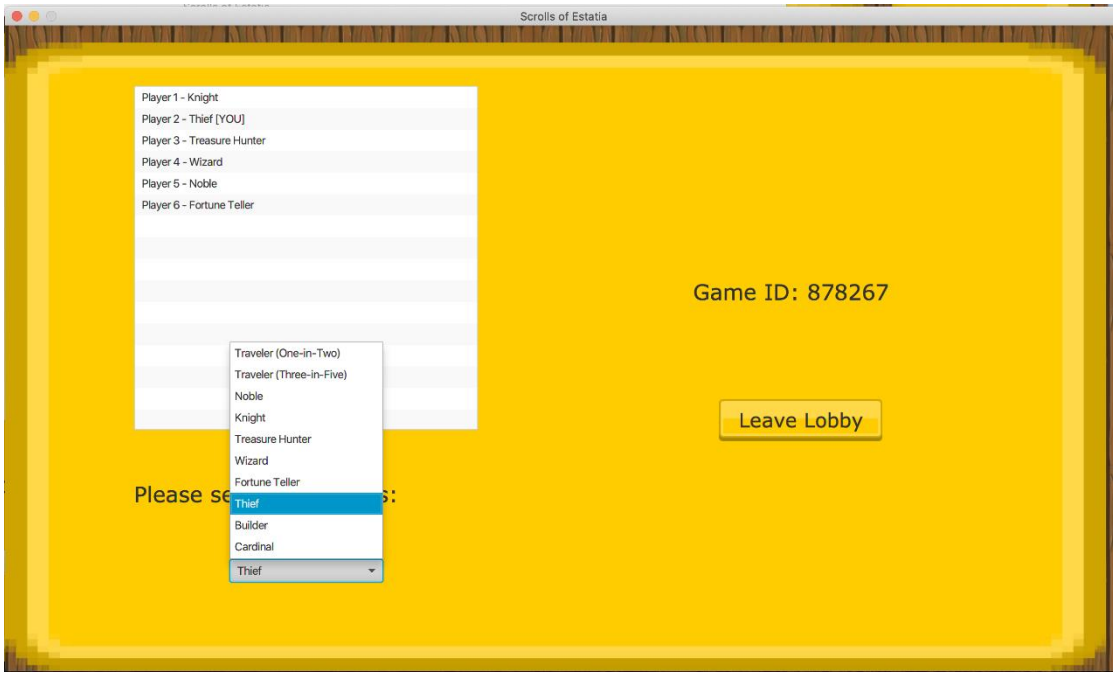

Figure 5: Class Selection Drop-down Menu in the Game Lobby.

There are 40 squares on the game board. The squares which have color tags, along with Smiths, Stables and Docks, can be bought. Each of these areas force all the players other than the owner to pay a certain amount of rent when they land on them. Smiths collect rent based on the payer's dice roll outcome and the rent of Stables and Docks increase as you buy more of them. The remaining squares with color groups are Towns, and you can increase their rent by erecting buildings on them. However, you can only erect these buildings on a Town once you land on it and have all the Towns in that color group. The color of the Town's name will change accordingly to the number of buildings built on it and the border of each Town will be highlighted in their owners' colors. The buying price of the Town will be shown if it is unowned. If it is owned, the current rent will be shown instead, and if it is mortgaged, the unmortgage price will be shown. As mentioned above, landing on Fortune squares will make you draw a random card from the Fortune deck. These cards' effects will be instant. Landing on Scroll squares will make you draw from the Scroll deck. Unlike Fortune Cards, Scrolls will be kept in your hand, and you will be able to use them in any of your turns. Landing on King's Tax and Queen's Tax will make you lose a certain amount of money when you land on them. Landing on Go To Dungeon will send you to the Dungeon square and you won't be able to play for three turns. Landing on the Dungeon square normally will not have any effects. GO, which is the beginning square, is where all the players start from. After each loop, you will gain a certain amount of

money as you pass this square. Landing on Feast will give you certain perks based on your class:

- Traveler: You move extra squares for the next four turns.
- Noble: You earn a certain amount of money.
- Knight: You will evade the next rent you would have to normally pay.
- Treasure Hunter: You draw a Scroll and a Fortune Card.
- Wizard: The amount of Mana you regenerate will be permanently multiplied by a factor of 1.15, e.g. 1 --> 1.15 --> 1,3225 -->
- 1,520875.
- Fortune Teller: You steal a Scroll from a random player if possible.
- Thief: You steal from every player.
- Builder: Makes you earn a certain amount of money for each Inn you built. You earn triple this amount for each Mansion you built.
- Cardinal: Randomly teleports you to one of your towns that has a Church or a Cathedral if possible.

During your turn, you first have to roll the dice using the "Roll Dice" button to start your turn. Before doing so, all the options that you are presented with are disabled. You can decide what you wish to do with the rest of your turn after rolling the dice.

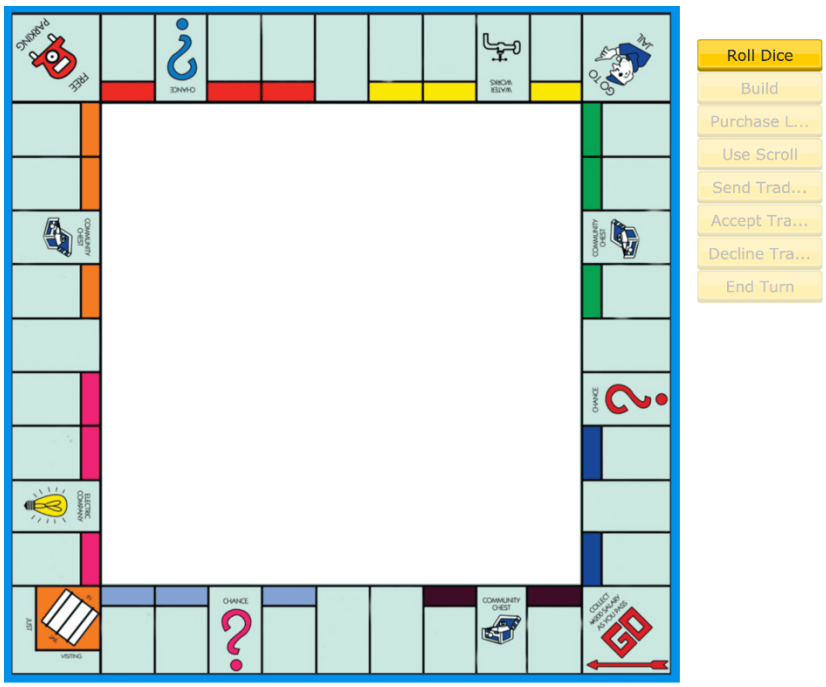

Figure 6: Game Board and the In-game Actions the Player can Perform Before Rolling The Dice.

After rolling the dice, the rest of the options are either enabled or disabled based on whether you are allowed to perform them or not, such as whether you are able to build on any of your properties or not. Using these options, you can do the following:

- You can build on the property you are on using the "Build" button if you own all the properties in that color group, have sufficient funds and the building limit on that property is not yet reached.
- You can purchase the land you are on using the "Puchase Land" button if the property is not owned by another player and you have sufficient funds.
- You can use one of the Scrolls in your hand using the "Use Scroll" button.
- You can send a trade request to one of the players by using the "Send Trade Request" button.
- You can accept a pending trade request by using the "Accept Trade Request" button.
- You can decline a pending trade request by using the "Decline Trade Request" button.
- You can end your turn by using the "End Turn" button.

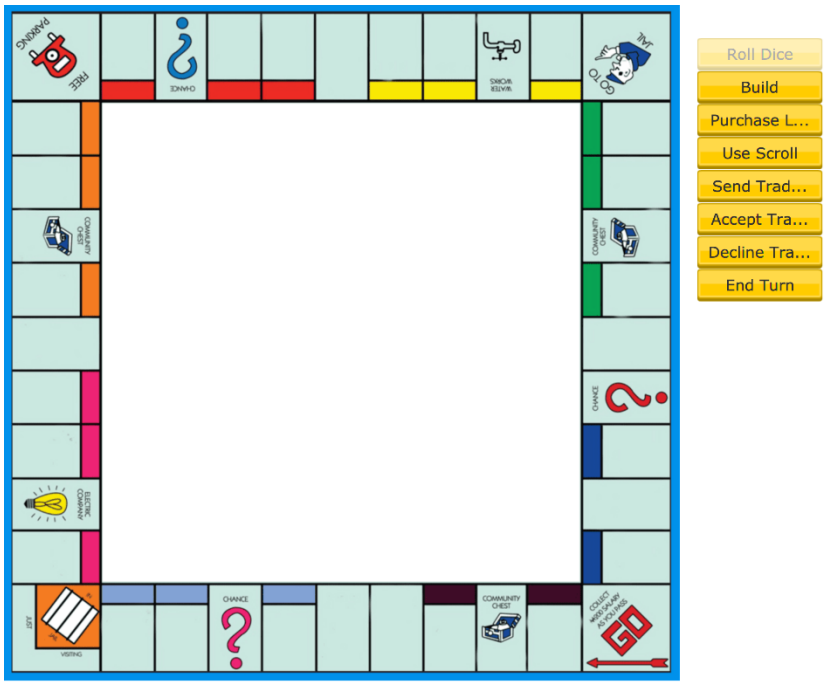

Figure 7: Game Board and the In-game Actions the Player can Perform During Their Turn After Rolling The Dice.

4. Build Instructions

System Requirements:

- Our game is written with Java, so you need to have Java installed to run our game.
- We don't have jar files for our game, so you need to run the game from the IntelliJ IDEA IDE.

Build instructions:

Our game is designed as an online multiplayer game, however as it currently stands, we don't have a dedicated remote-server and the client takes "localhost" as the IP address of the server, therefore currently the server needs to run on the same machine as the clients. In order to play the game, clone the latest branch of the game from our GitHub repository, and open the repository folder as a project with IntelliJ IDEA and build the project. Then, run one instance of the Server class, and run as many instances of the Player class as needed (If you are unable to run multiple instances of the Player class, on IntelliJ IDEA, click "Run" from the top and click "Edit Configurations", from the menu on the left select "Player" and then click "Modify options" from the right, and from the dropdown menu, make sure "Allow multiple instances" is enabled). Host a game with one of the Player instances and join the game with the other Player instances with the game ID. Do not stop any of the instances while they are running. Only when you wish to end the current session and start a new game, stop all running instances and run the Server class and Player classes again.