Bilkent University

Department of Computer Engineering

CS319 Term Project

3A - Monopoly

Final Report

Ali Taha Dinçer, İrem Ecem Yelkanat, Kutsal Bekçi, Saidcan Alemdaroğlu, Sultan Simge Kürkçü

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

Progress/Iteration 1 - 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 Object Oriented Software Engineering course CS319.

Contents

1 Introduction

We have started the implementation of the Yolopoly game before we submitted the design report. We have used the IntelliJ IDE environment to program the game and the developers of our game have committed the changes via GitHub to be able to work simultaneously and be aware of each other. We as the team have used Jira platform to split the responsibilities of each group member and to be punctual on our works. For GUI implementation, we used javaFX library.

We have implemented all types of squares and cards of the actual game Monopoly. We have created a board that includes squares, cards as well as dices. The visual we have created is quite similar to the actual game. The users can play the game by simply rolling a dice and the pawn will move by the play itself. After the movement of the pawn, the game provides multiple choices for the user if on a property square. There are some obligatory movement required squares on the board. After the player starts to play the game, if it is a single player game, the player can save and start the game from the saved point afterwards.

In addition to the explanation above, we have successfully managed to add extra functionalities to the game Yolopoly such as "Bankman" game mode, multiplayer game option and different themes. We believe that, as the original game and our game stands on the logic of money handling, "Bankman" game mode will breathe a new life to the game.

2 Design Changes

Staggeringly, we have not made many changes to the design of the project. As we have discussed every step during our meetings and most of the members of our group have experienced in the field, we have had a very consistent design plan during the implementation of the game.

During the implementation of the InGameManager class, it has been seen that the game is not working linearly, there are also some important breakpoints in the game such as bankruptcy and auction. Hence, it is decided that to have a state mechanism in the InGameManager class. So that UI can know about these breakpoints and handle corresponding requests given by the players without disturbing any data stored in the class.

Second change was made on the logic we have been used on to gather data from different manager classes. In that case, carrying the data from the managers to the controllers is a constant process. At the beginning of the project, we have

used static fields from the Main class in order to have exactly one object of each manager in the game and having a consistent data flow. However, finally one of us decided to stop this type of usage of and changed each manager to Singleton classes.

Although it is intended to have an online multiplayer gaming system in the game, because of the complex logics that the game contains such as different types of drawable cards and their corresponding movements and actions, most of the time is spent for the bug-fixes in order to have a completely bug-free working game for the players. Hence, the online multiplayer game option feature is abandoned and it is decided to have a multiplayer game option for more than two human players on the same computer.

3 Lessons Learnt

In the implementation stage of the project, we first decided to divide the project into parts and we thought that we could merge these parts and get the final result as a smooth version of the game. On the other hand, when we tried to commit our changes into the project, some problems that stem from the different way of thinking of the group members have occured. Since we didn't want to deal with such complicated problems, we have decided to share the way of implementing the algorithm and other details with each other before we implement it. Another method we have followed in order to be understandable among group members is that we try to write the codes readable and understandable so that confusions will be decreased.

The meetings carried an unremarkable value in our projects because we tried to plan every step of the project. We have used online tools like zoom and discord to be in contact and avoid delaying tasks of ourselves. We conducted meetings every week in the beginning weeks of the project and then we increased the frequency of the meetings as we need to work harder.

4 User's Guide

This section of the report clarifies how a player could advance through the game from the beginning

4.1 Main Menu

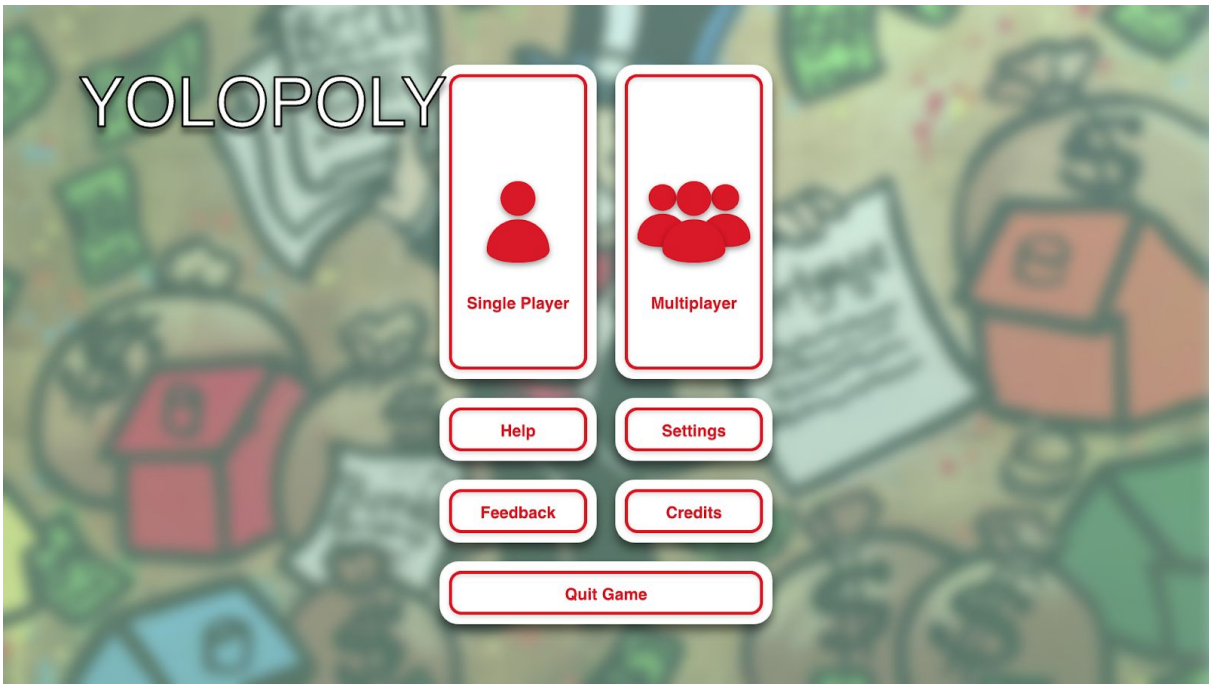

Figure 1: Main Menu Screen

In the main menu there are seven options that the user is able to do:

- The player can start a single player game.
- The player can start a multiplayer game.
- The player may want to learn how to play the game with the help of the help button.
- In the setting option the player is able to change some settings of the game.
- In the feedback option of the main menu, the player can give feedback to the contributor by writing a paragraph.
- In the credits section of the main menu the player can see the contributors of the game.
- The player can quit the game.

4.2 Single Player

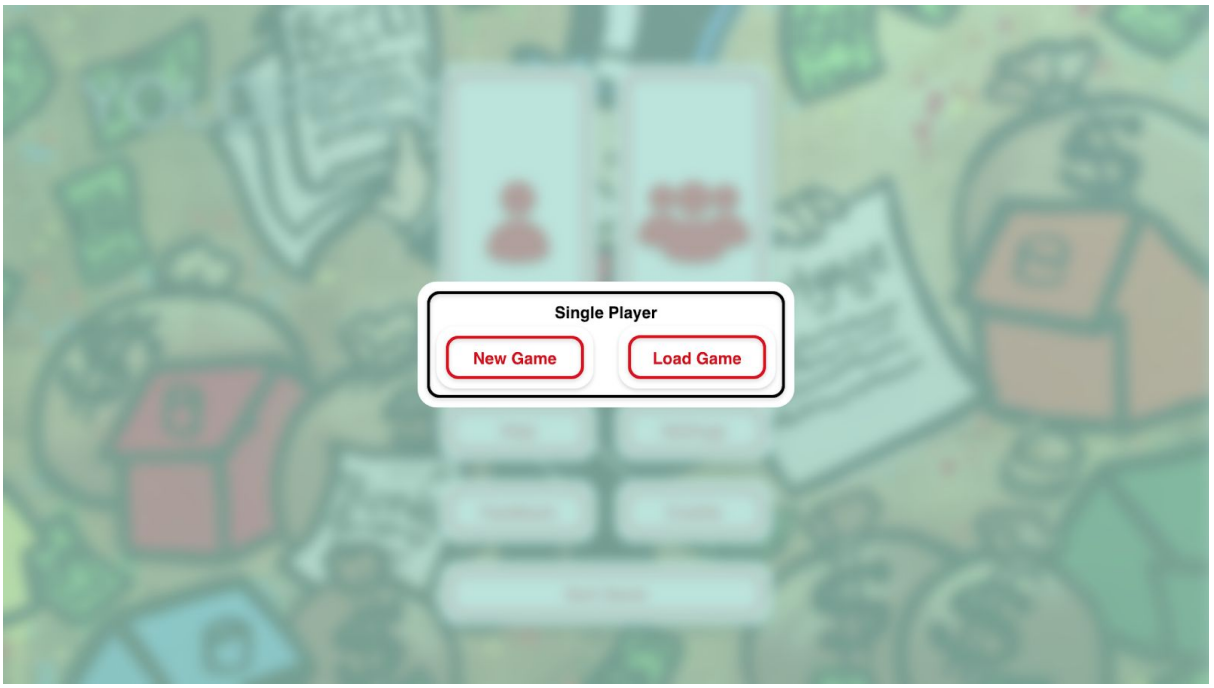

Figure 2: Single Player Options Modal

In the single player menu, the player may choose to start a new game or load a previously started game if there is any.

Figure 3: Saved Games List Modal

If the player clicks on the load game option, the saved games list will appear. In this scenario, choosing one of the previously saved games is expected from the user.

4.3 Multiplayer Settings & Single Player Settings

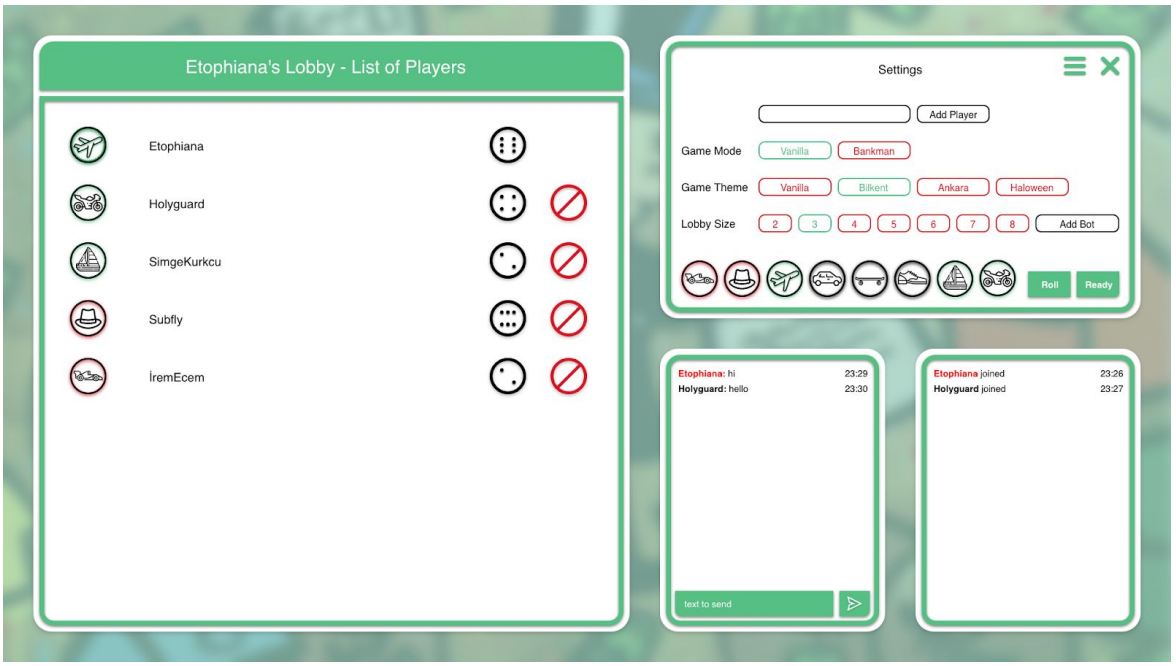

Figure 4: Game Lobby Settings Screen

The multiplayer game screen is the same as the single-player game screen in the name of available settings. Creator of the game can change game mode, game theme and available players in addition with adding human players and bots. Players can write on chat and see the game login that screen

4.4 Help

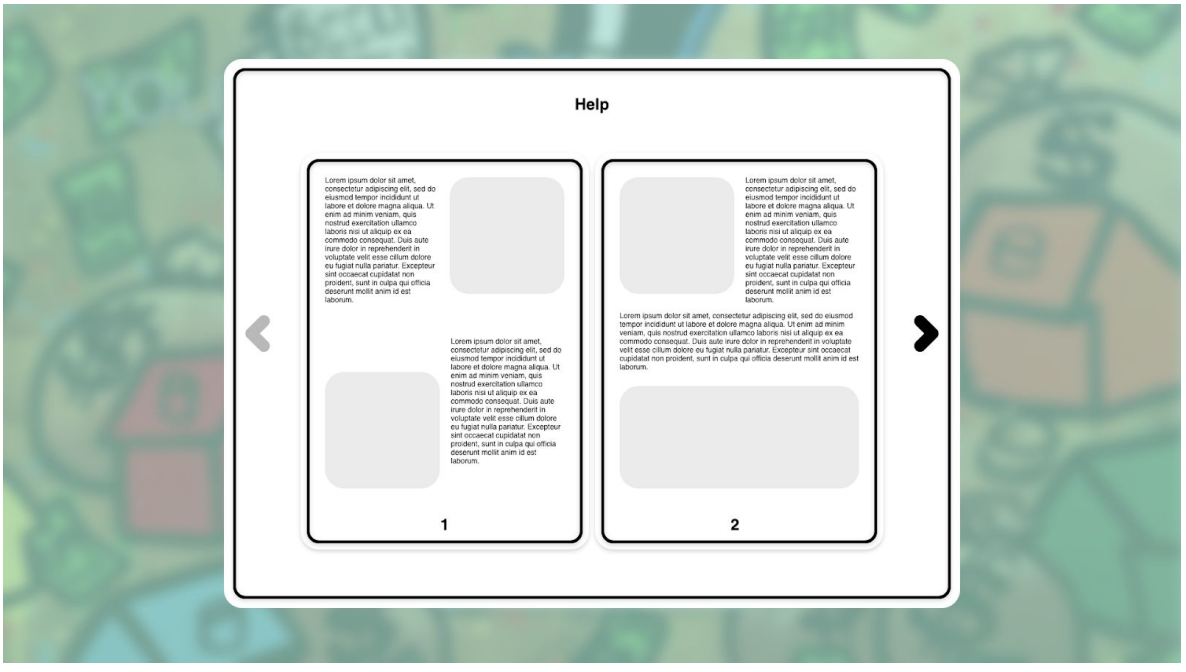

Figure 5: Help Modal

In the help option of the main menu, the player can have the instructions about how to play the game. The player can continue to the main menu by clicking on the exit cross symbol on the upper right corner.

4.5 Settings

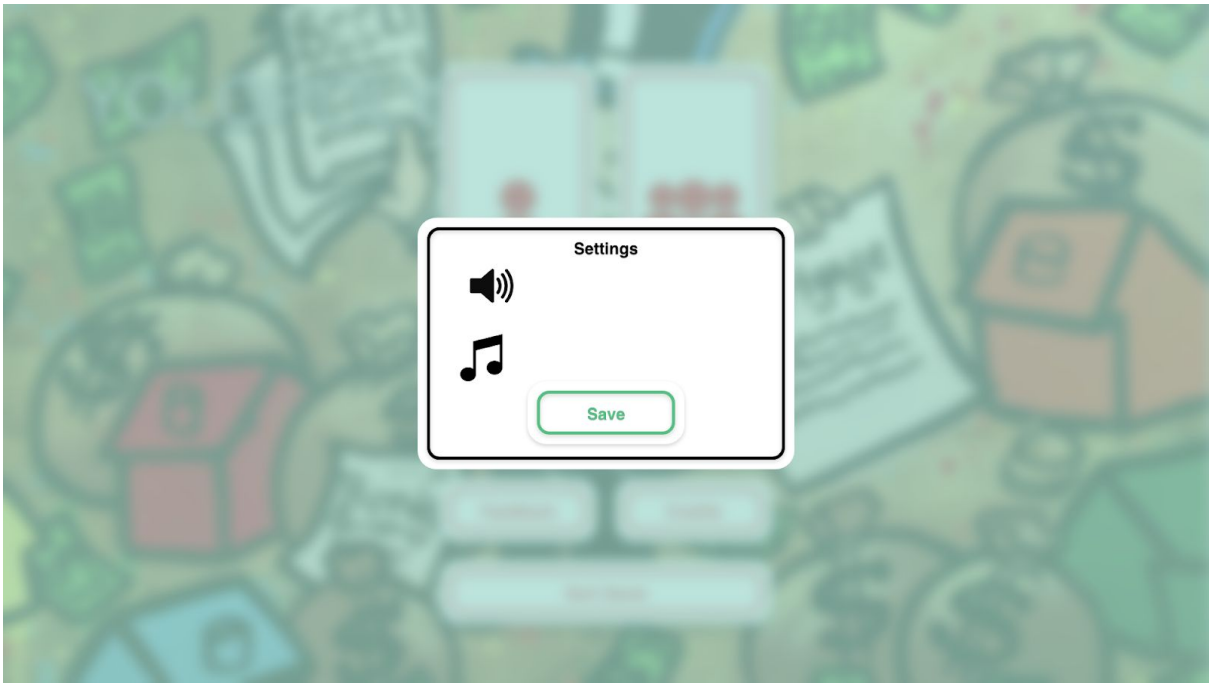

Figure 6: Settings Modal

In the settings option, the player is able to change the settings of the game such as game music and game effects.

4.6 Feedback

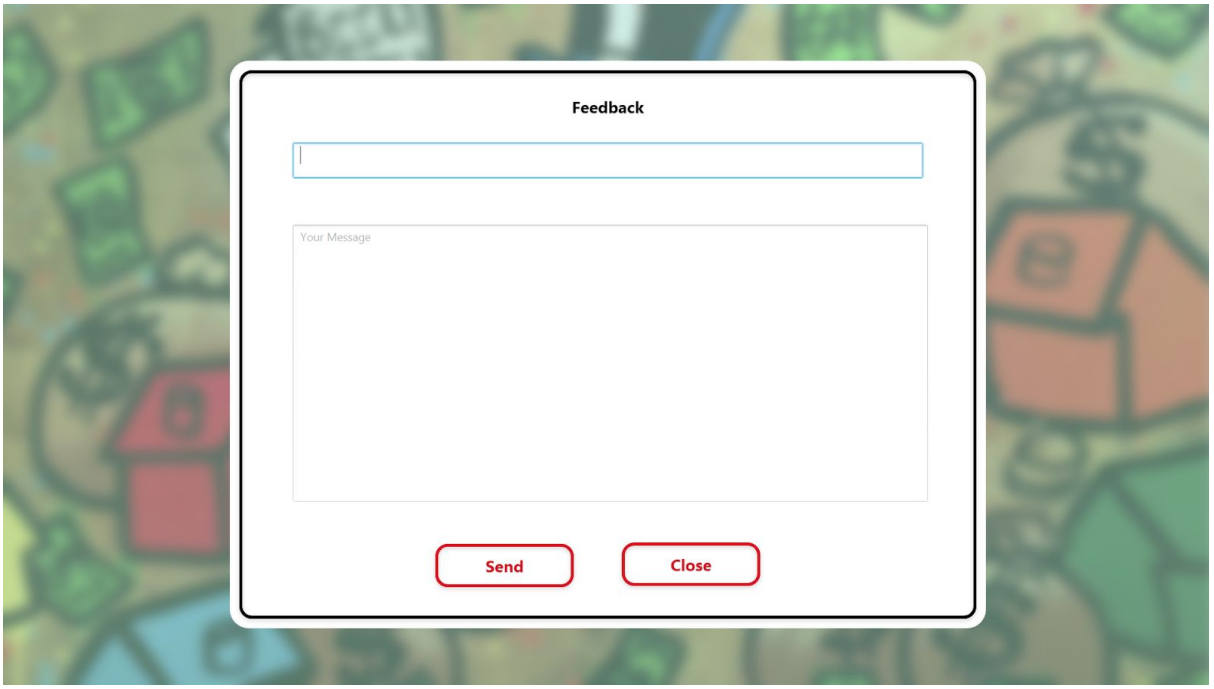

Figure 7: Feedback Modal

The feedback option allows the users to give feedback for the game by simply writing a paragraph in the related empty box.

4.7 Credits

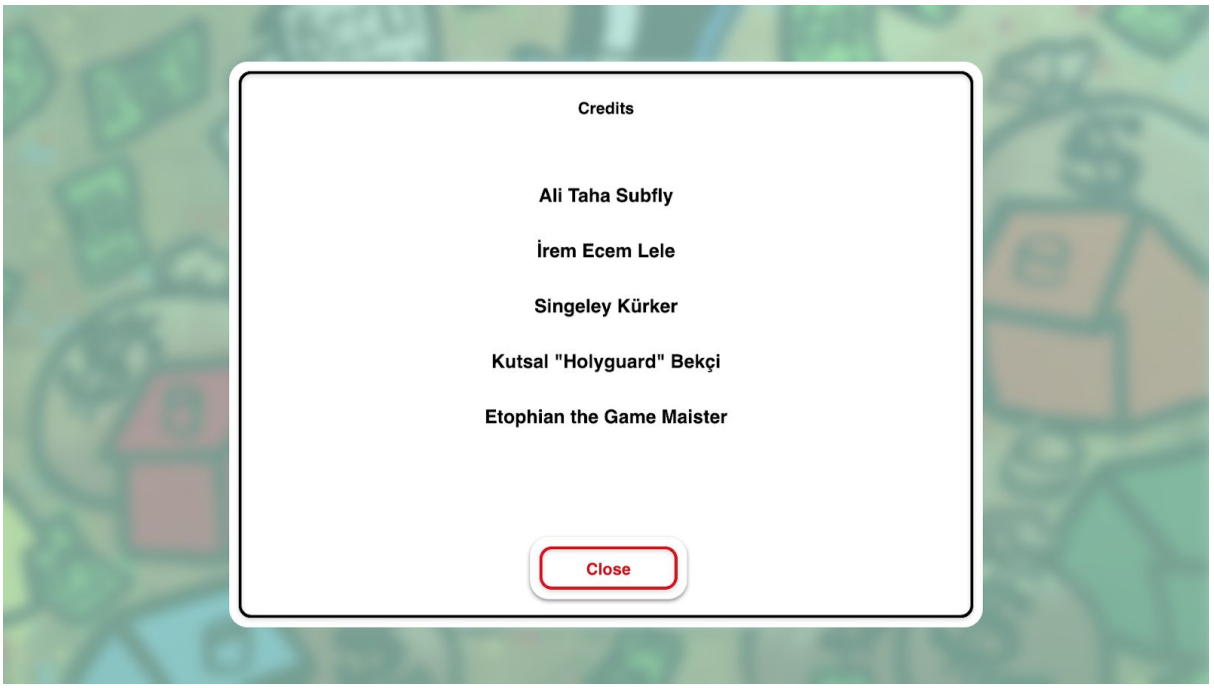

Figure 8: Credits Modal

The credits section of the main menu shows the contributors of the game.

4.8 Quit Game

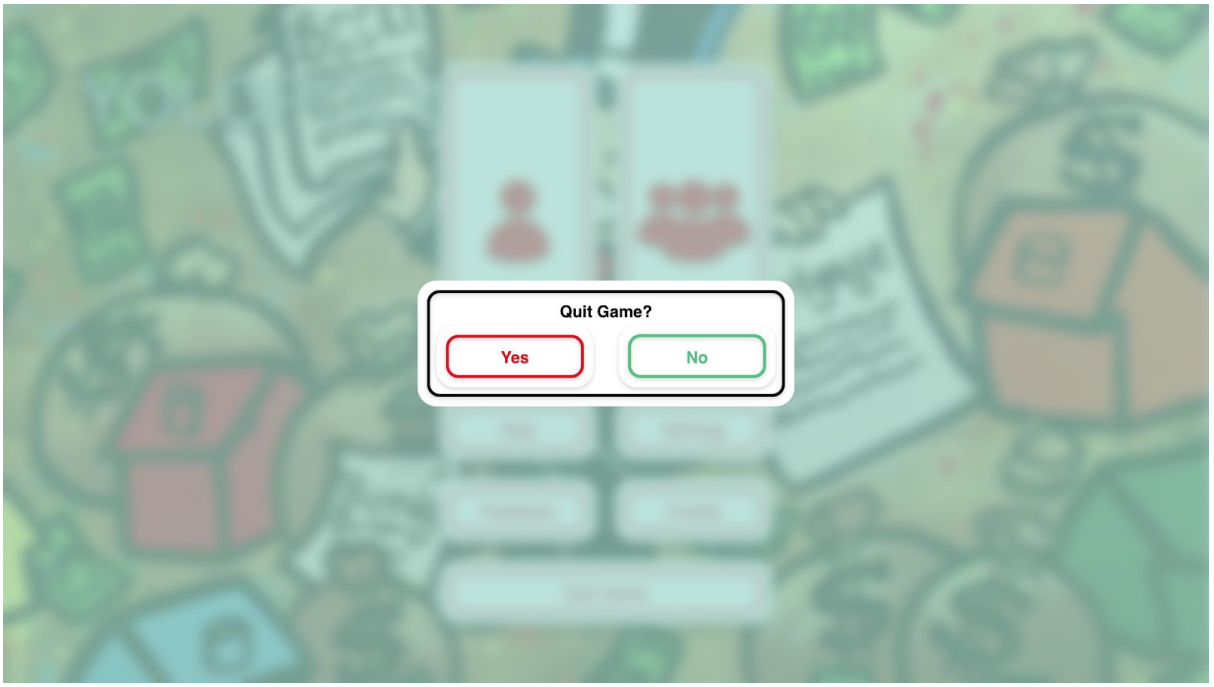

Figure 9: Quit Game Modal

Exit game button closes the game.

4.9 In Game Screen

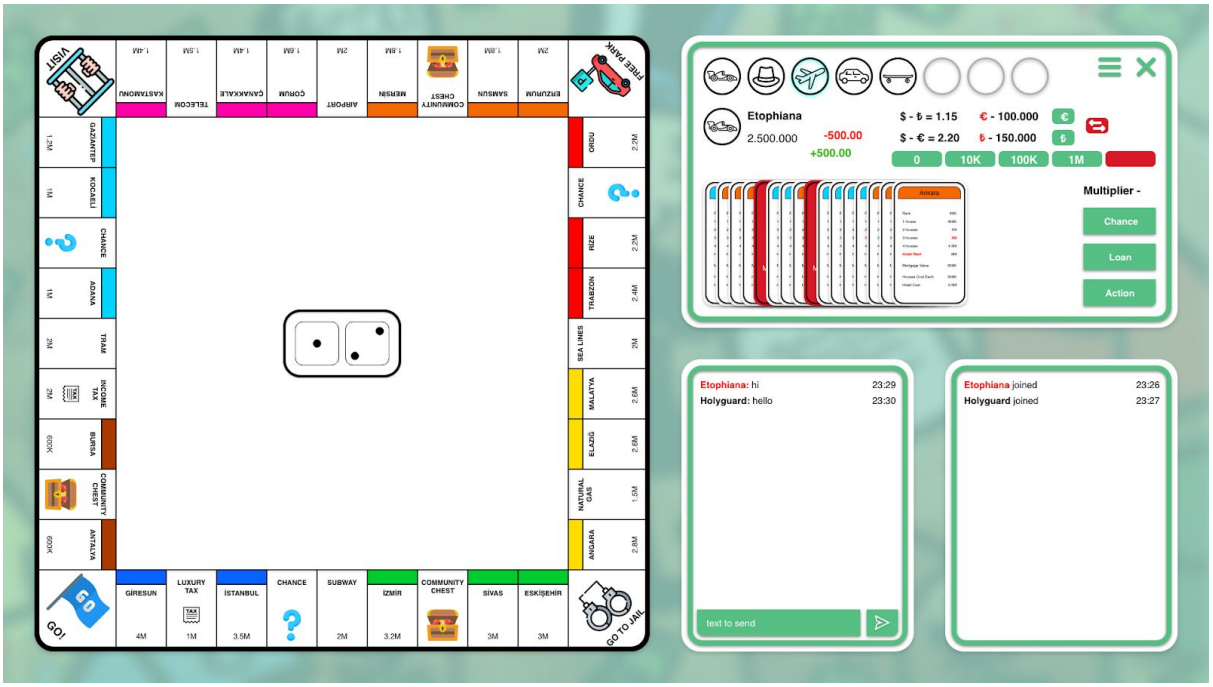

Figure 10: Game Screen

The players are navigated to this screen when the settings are selected. Players can see the board of the game in the left part of the window. In the right part, the upper part of the window, there can be seen the deck. From the deck, players can see their money, their properties as well as other players money and properties. Only ın the "Bankman" mode of the game, the players can see the currencies, exchange between currencies, get loans or take the chance and get a payment multiplier from that deck. Players can follow the game by looking into the log and can chat with the bots in the chat.

4.10 Auction Screen

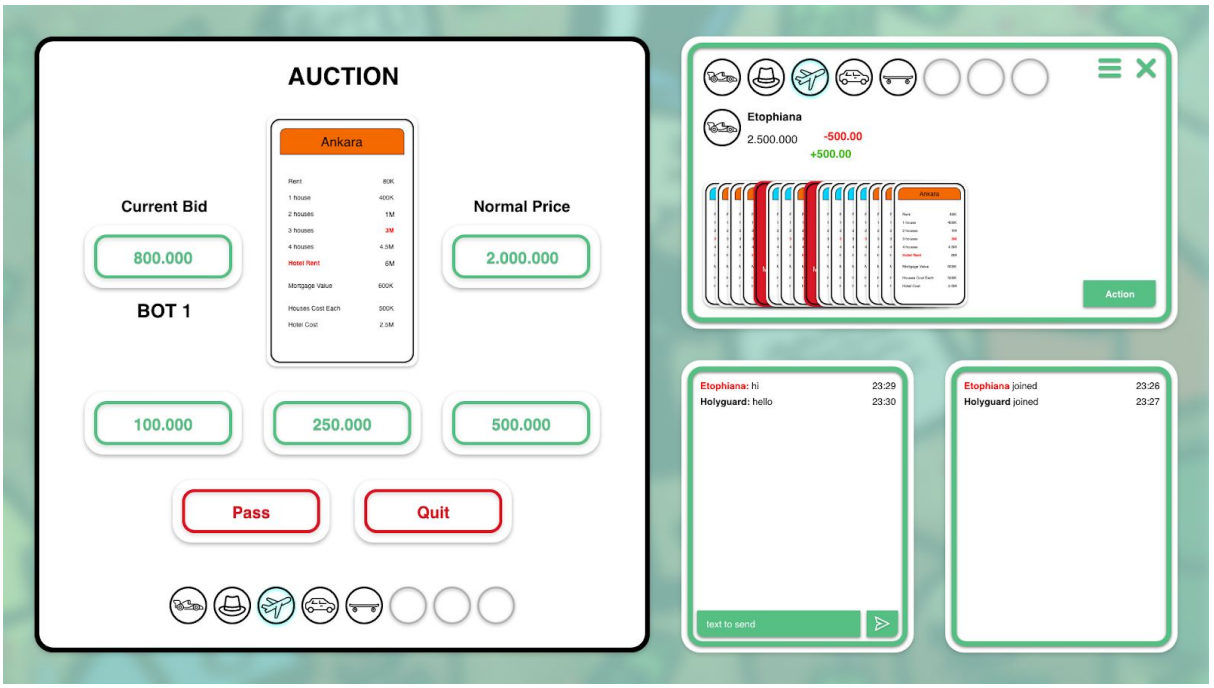

Figure 11: Auction Screen

The players will see this screen when they get into an auction.The players can get into auction by giving bids. Players are able to quit from auction or pass the turn. Players can see their deck and money and follow the auction from the log.

4.11 Draw Card Screen

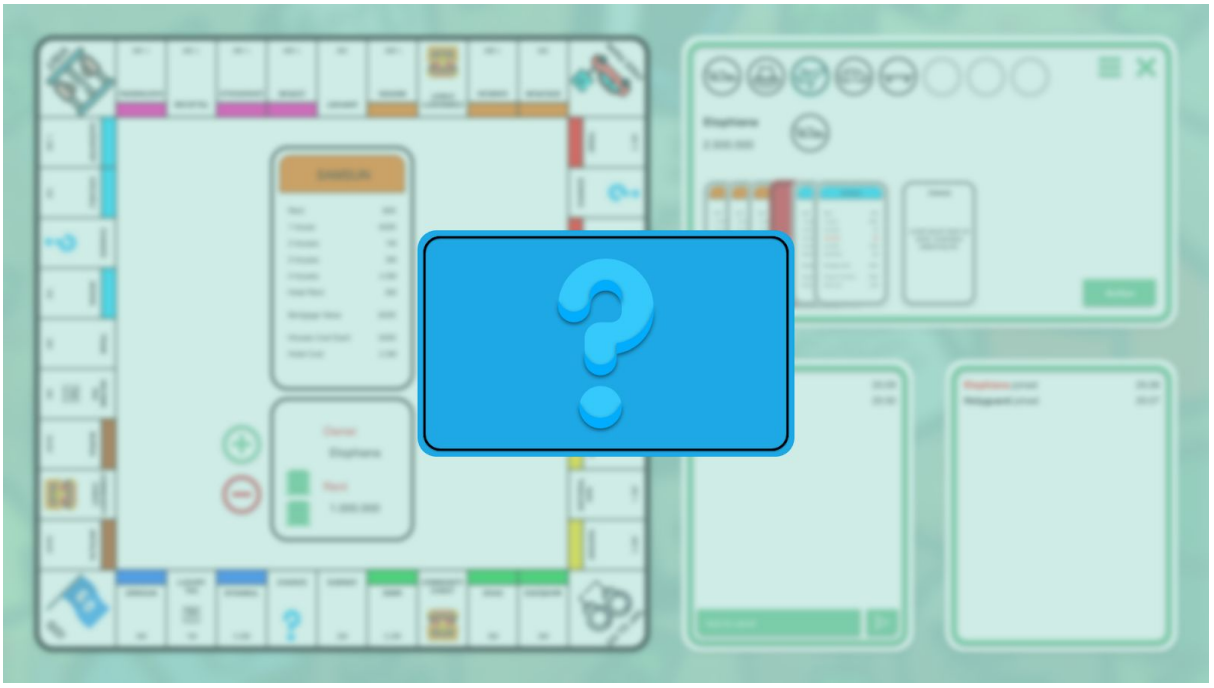

Figure 12: Draw Card Modal

The players will see this screen when they draw a chance or a community chest card. The screen can be closed by pressing the card.

4.12 Property Screen

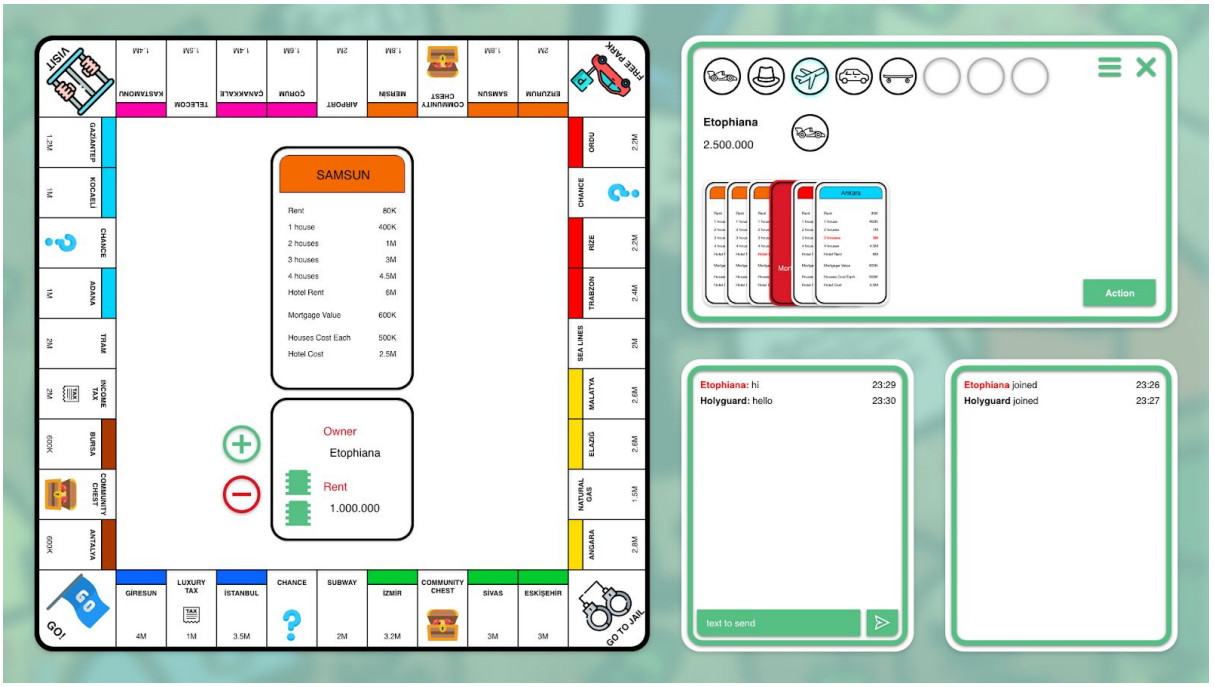

Figure 13: Card Modal

Finally the players can see their properties by clicking the squares on the board or hovering the mouse in the properties ordered in the deck.

5 Build Instructions

In order to install the game, you need the Standard JAVA Runtime Environment installed on your computer.

For better UI, the required resolution of the screen is 1920x1080. If your computer does not have or support this resolution:

- If you are using MacOS, you can download software from this site. After installation, change the resolution of your screen by selecting the resolution shown below:

Figure 14: Settings for MacOS

- If you are using Windows, follow the instructions on the site and change your resolution to 1920x1080.

You can install and play the game in the following ways:

- You can clone or download the source code from GitHub. After, open Yolopoly directory inside the main directory named 3A-Molopoly in your IDE supporting JAVA (Intellij recommended). Edit the configurations of the project, select Main.java file (com.yolopoly.Main) as the main class.

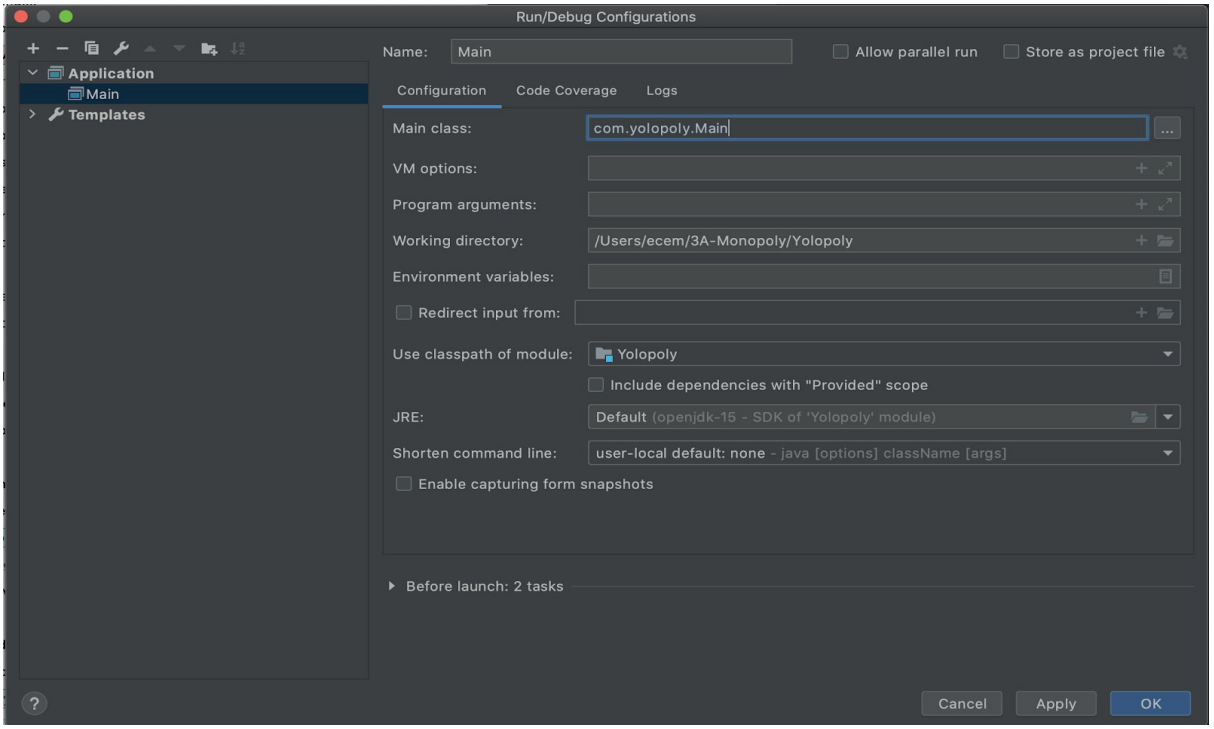

Figure 15: Example Configurations

- If the IDE does not recognize that it is a maven project, right click on the pom.xml file in the Yolopoly directory and select the option "Add as Maven Project", make sure to reload the project just in case.

6 Work Allocation

Ali Taha Dinçer

- Wrote the game overview in the Analysis Report with İrem Ecem Yelkanat.
- Designed and drawn class diagram in the Analysis Report with İrem Ecem Yelkanat.
- Designed and drawn subsystem decomposition diagram in Design Report with İrem Ecem Yelkanat.
- Designed and drawn class diagram in the Design Report with İrem Ecem Yelkanat.
- Contributed in game logic functions.
- Designed project file structure with İrem Ecem Yelkanat.
- Implemented save and load game with İrem Ecem Yelkanat and Kutsal Bekçi.
- Implemented the mechanics of "Bot" players.

İrem Ecem Yelkanat

- Wrote the functional and nonfunctional requirements in the Analysis Report.
- Designed and drawn use case diagrams in the Analysis Report with Sultan Simge Kürkçü.
- Designed and drawn class diagram in the Analysis Report with Ali Taha Dinçer.
- Designed and drawn subsystem decomposition diagram in Design Report with Ali Taha Dinçer.
- Designed and drawn class diagram in the Analysis Report with Ali Taha Dinçer.
- Contributed in game logic functions.
- Implemented music and feedback functionality.
- Implemented save and load game with Ali Taha Dinçer and Kutsal Bekçi
- Designed project file structure with Ali Taha Dinçer.
- Implemented extra functionality "Bankman Game Mode".

Kutsal Bekçi

- Designed and drawn sequence diagrams in the Analysis Report.
- Designed and drawn activity diagrams in the Analysis Report.
- Designed and drawn state diagrams in the Analysis Report.
- Contributed in game logic functions.
- Implemented building logic.
- Implemented save and load game with Ali Taha Dinçer and İrem Ecem Yelkanat.
- Tested the software and proven himself as the real bug-hunter.

Saidcan Alemdaroğlu

- Design and implementation of the project's user interface.
- Design of the images for the project.
- Connecting model classes with GUI by controllers.
- Contributed in game logic functions.
- Contributed in designing of the project structure.

Sultan Simge Kürkçü

- Designed and drawn use case diagrams in the Analysis Report with İrem Ecem Yelkanat.
- Organised the documentation of the sketch of the project in Design Report.
- Put json files in order and edit their indexes.