



Bilkent University

Department of Computer Engineering

Object-Oriented Software Engineering Project

monopoly-game: Digital version of the well-known board game Monopoly

Final Report

Group 1C: Ziya Mukhtarov, Ege Kaan Gürkan, Alper Sarı, Mokhlaroyim Raupova, Javid Baghirov

Instructor: Eray Tüzün

Teaching Assistant(s): Barış Ardiç, Emre Sülün, Elgun Jabrayilzade

December 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.

Table of Contents

1	Introduction	3
2	Lessons learned.....	3
3	Work Allocation	3
3.1	Javid Baghirov	3
3.2	Alper Sarı.....	4
3.3	Mokhlaroyim Raupova	4
3.4	Ege Kaan Gürkan	4
3.5	Ziya Mukhtarov	4
4	User's Guide (Manual)	5
4.1	Navigating the Main Menu	5
4.1.1	Creating a Lobby	5
4.1.2	Joining an existing Lobby.....	5
4.1.3	Navigating the Options Menu.....	5
4.2	Gameplay	5
4.2.1	Video, Voice and Text Chat	5
4.2.2	Playing a Turn.....	5
4.2.3	Trading	6
4.2.4	Auctions.....	6
5	Build Instructions (System requirements & installation how-to)	6

1 Introduction

The game itself is lacking full functionality, as we had started our implementation by getting a working network system and implementing a useable interface which supports full video and voice chat usage. However, we were not able to follow by implementing the main game logic which allows the players to play the game to its completion such as moving across the board and owning, trading, and buying properties as well as buildings.

The missing functionality is as follows:

- Chance cards and Community Chest Cards
- Being in Jail
- Trading
- Text Chat (Video and Voice chat works)
- Game Ending
- Mortgaging
- Build House / Hotel
- Vote Kick

The remaining game systems such as rolling dice, moving across the board accordingly, buying and auctioning properties, engaging with tiles such as paying the rent or getting money from GO tile, using webcam and microphone to communicate with other players are functional and players are able to access and manipulate them.

2 Lessons learned

Tell us about your experience

While working on the project we quickly learned that work distribution and management was the most important part of any ongoing project. In that vein we had decided to split the work required, especially in front end and back end parts of the project, to two separate groups. While this approach worked for a while, due to complications during development there were instances where a certain functionality or UI screen was needed to make progress and the other team was stuck on a previous task.

This has thought us to be more flexible with our work allocation to avoid situations where the progress on the project would slow down due to a missing module or class.

3 Work Allocation

3.1 Javid Baghirov

- He was responsible for the communication of the entire server-side with the client. He implemented server classes that maintained the flow of data from server functionality to the client side.
- Implemented lobby functionalities that were later revised by team leader Ziya Mukhtarov

- Improved the in-game functionalities written by Alper Sarı and provided the client with the obtained data
- Game: The main Game class was written by him which holds almost all of the data of the game along with the GamePlayers and the Board
- Added new classes to the network-side with the help of Ziya Mukhtarov

3.2 Alper Sarı

- Board and Tiles: The board system which handles the movement of tokens and therefore the players on a set of tiles.
- Properties and Property data handling: The “Property” class he wrote and its assigned child classes as well as the property data class called “TitleDeedData”
- Auctioning and Trading: The system used for trading properties and the auctioning system used to sell an item to the highest bidder.
- Cards: The card system that allows an easy-to-use way of creating and executing separate card functionalities

3.3 Mokhlaroyim Raupova

- She was assigned to the UI team and worked on interfaces mostly. She implemented custom components along with their controllers and FXML files. Helped to improve other components.
- She worked on the design of almost each screen and received help from Ege Kaan Gürkan and Ziya Mukhtarov while implementing the design.
- Worked on screen responsiveness with the help of Ziya Mukhtarov.

3.4 Ege Kaan Gürkan

- From controllers to FXML files to custom components, he has touched upon almost all UI elements, starting and finishing some from ground-up, and sometimes getting help from Ziya Mukhtarov and Mokhlaroyim Raupova.
- Wrote all of the CSS classes and attributes for the UI elements.
- Worked with Ziya Mukhtarov to improve the responsiveness of the UI.
- Set up the EC2 server used in the project to host the game server.

3.5 Ziya Mukhtarov

- He was our “Project Leader”. He reviewed all the codes written for this project and gave coding advices frequently. The parts that were written by him is shown below
- All low-level networking that powers our Monopoly application
- Webcam and microphone access and sending it over the network
- Fixed server-side lobby functionality that was written by Javid before the iteration 1 deadline
- Improved responsiveness of nearly all UI screens when they are resized
- Connected server to client for all screens and provided the client UI with the incoming data

4 User's Guide (Manual)

4.1 Navigating the Main Menu

4.1.1 Creating a Lobby

While in the main menu, the user can click on the button labeled "Play Game" which will take them to the lobby screen, from here they will have the option of creating a lobby. While creating the lobby they can choose the maximum player count that lobby can have as well as decide to make said lobby private, which will allow the user to set a password for said lobby in which only the player that know said password can join.

4.1.2 Joining an existing Lobby

After pressing the "Play Game" button, the user is able to choose one of the lobbies in the lobby list to join, if said lobby is created as public. They can alternatively choose a private lobby to join, in which case they will be prompted to enter the required password to join it.

4.1.3 Navigating the Options Menu

Pressing the "Options" button while in the main menu screen will take the user to the options menu where they can change game settings to customize their playing experience.

The user can also access the options menu within the gameplay screen after a game has started

4.2 Gameplay

4.2.1 Video, Voice and Text Chat

The user can set whether or not they wish to use their microphone or video while playing the game by clicking on the relevant buttons next to their name which will enable and disable voice and video chat functionality.

They can also choose to talk over text chat, in which case they will need to click on the text chat icon and type their message before pressing send.

4.2.2 Playing a Turn

The player, at the start of their turn rolls 2 dice, which causes them to move a certain distance forwards on the board. They then take an action depending on the tile that they have arrived in.

In the case that the player has arrived in a property tile, they will either have the option of purchasing said tile if it is unowned or pay the appropriate rent amount if it is owned by someone else. They can also build buildings on the tile if it is owned by them.

The players can also made to draw a card when they land on the appropriate tile, and will be sent to jail if they step on the "Go to Jail" tile. To get out of jail the player can use a "Get out of Jail Free" card or keep rolling their dice until they roll a double value.

Players can also participate in Trades ad Auctions as explained below, these take place regardless of the tile any of the players are on.

4.2.3 Trading

After a player starts trade with another player, they will be given the choice of deciding what items they wish to put in the trading list. After both sides have put in and removed items according to their preference from the trading pool, they can choose to accept the trade. In the case where both sides accept the trade, the trade is finalized and the tradeable items switch owners.

4.2.4 Auctions

Once an item is put to auction, all players take turns deciding if they wish to increase the auction bid amount of the auctioned item. After all of the bid increases are done the players decide if they agree to the auction. If all players decide not to bid and agree to the auction, the item is sold to the final bidder at the bid price and the auction is closed.

5 Build Instructions (System requirements & installation how-to)

We are using Apache Maven (<https://maven.apache.org>) for our project management. Hence, all of our dependencies are automatically installed when our project is built with Maven. Many modern IDE's support Maven out-of-the-box. To build our project in such an IDE, the project folder should be imported as a Maven project to the project's workspace.

To build the project without using an IDE, Maven command-line tool should be installed to the system. For instructions, see <https://maven.apache.org/install.html>. After installing Maven, our project can be built by executing the following command in our project's root folder:

```
mvn install
```

Executing this command will download all the dependencies and generate the jar files required for running our project. Note that JRE 14 is required to run the jar files. The jar files will be generated in the <subsystem>/target folder. For instance, client jar file can be found at client/target folder. Then using the java -jar command, it can be executed (Note that the version number in the jar's name can be different):

```
java -jar client/target/client-0.0.1.jar
```