



Bilkent University

Department of Computer Engineering

Group 2G: Hasbrothers

Project Short-Name: Monopoly Bilkent

Final Report

- Mehmet Yaylacı
- Vural Doğan Akoğlu
- Mert Laleci
- Barış Tiftik
- İlhan Koç

Supervisor: Eray Tüzün

Contents

1. Introduction	3
2. Lessons Learnt	3
3. Design Changes	3
4. User's Guide	4
5. Build Instructions	9
6. What Is Completed?	9
7. What Is Missing?	9
8. Work Allocations	10

1. Introduction

We stand roughly middle in the project. Our analysis and design reports are ready. But, Implementation part is not completed yet. We have classes related to monopoly and we have menus with a game screen. We have some extra features like avatars in the lobby screen. We have added sign-up and sign-in mechanics. They have their own pages and function correctly. However, we could not connect game classes to the GUI of the game.

2. Lessons Learnt

We have learnt that to have a better code the design parts had to be great. When we have tried to merge the UI part of the code with the abstract parts of the game, we had trouble, so we have learnt that when you start to code you have to think about the whole picture. We have also realized that it would be better to start coding earlier as we had trouble merging the various parts of the code. We have also learnt that when one person from the group lacks behind the whole project lacks behind, so it is quite important to give higher priority to the important parts of the code. We think that this project helped us see the real life implementation of a small-scale project that we have already learned the abstract basics in class. This project was useful for us to experience the problems and find the approaches for solutions to these problems. We also learnt and experienced working on a git platform as a team. We tried to inform others with the proper commit comments and pushed our works as frequently as possible.

3. Design Changes

At the beginning we were planning to make an online game, but then because of the time limit we decided to make our game as a multiplayer game(one computer turn based). However, we kept a sign-up and sign-in system to keep player's avatars and names. Since this is a representative sign-up, sign-in system (we saved data in a .txt), we could not use databases as we mentioned in our reports and we could not use sign-up and sign-in as our authentication method.

4. User's Guide

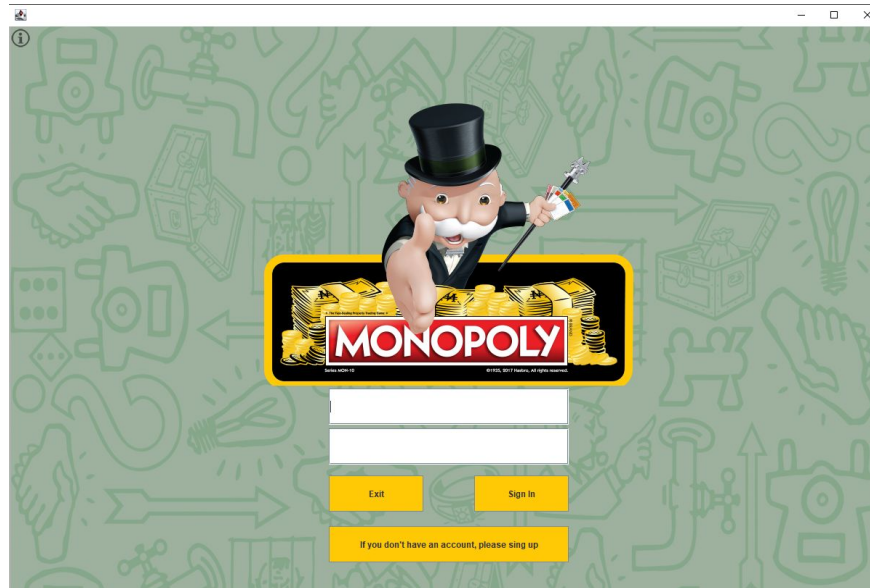


Figure 1: Sign in Screen

In the sign in screen, if the user has a username and id, enter the system by clicking the sign in button. If the user doesn't have an account, click it to sign up.

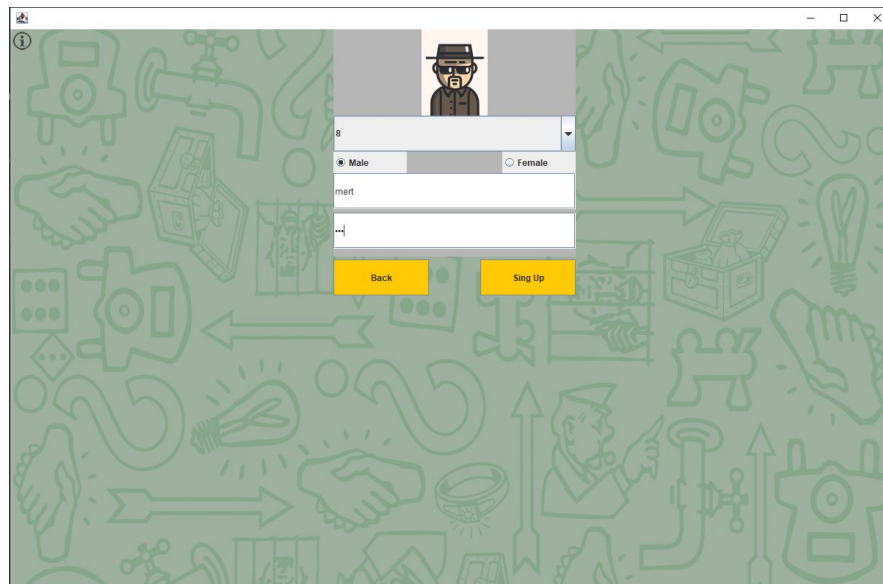


Figure 2: Sign up Screen

In the sign up screen, the user needs to select gender(male/female) and avatar(out of 20), enter an username and id.Then click the button sign up and return the sign in screen and enter identified username and id to sign in the system.

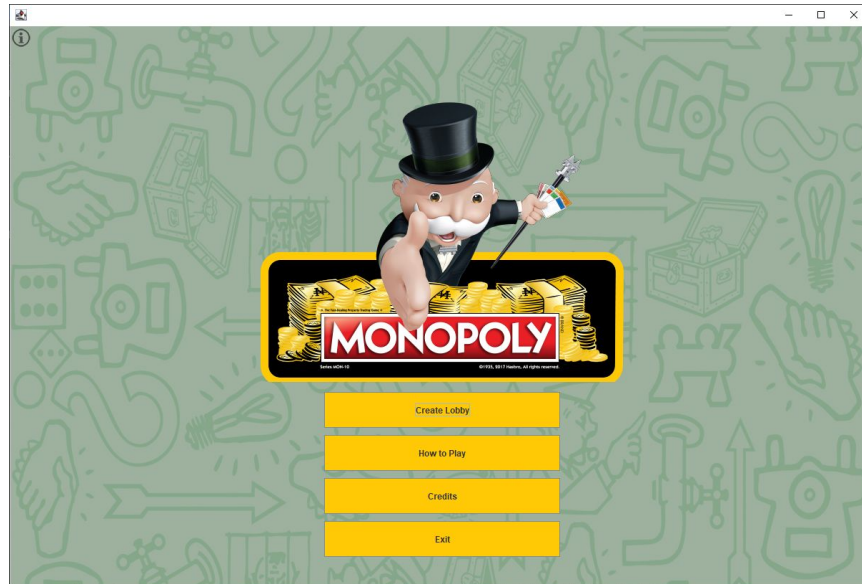


Figure 3: Main Menu Screen

In the main menu screen, the user can see four buttons as following: Create Lobby, How to Play, Credits, Exit.

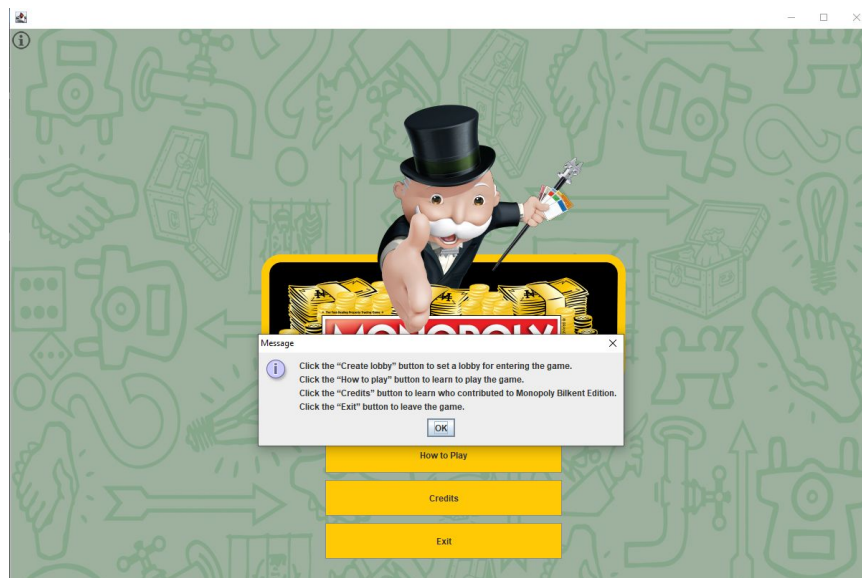


Figure 4: Main Menu info Pop-up

At the top left of each screen, there is a user guide(i icon) that explains the functions of the buttons on the screens, so, you can read the necessary information from the pop-up screen that opens by clicking this icon.

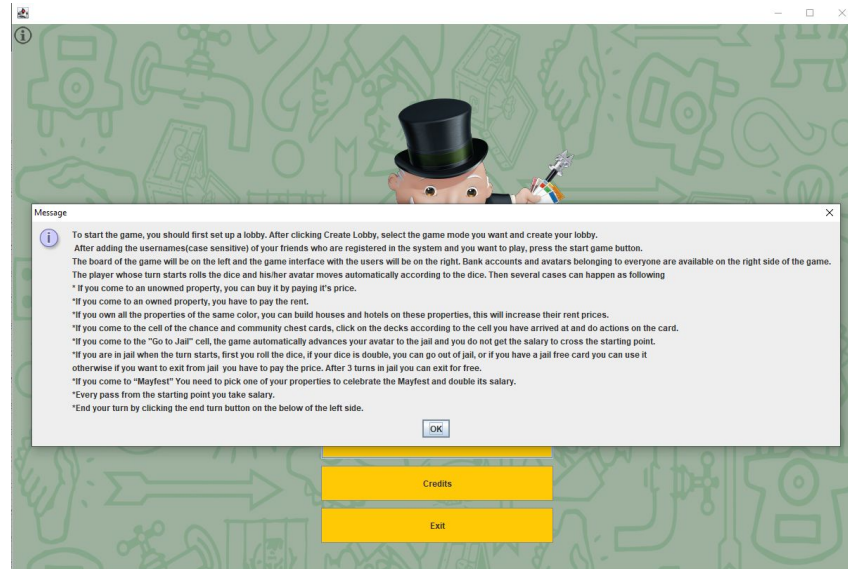


Figure 5: How to play pop-up

The user can click the “How to play” button to learn for playing the Monopoly Bilkent Edition.

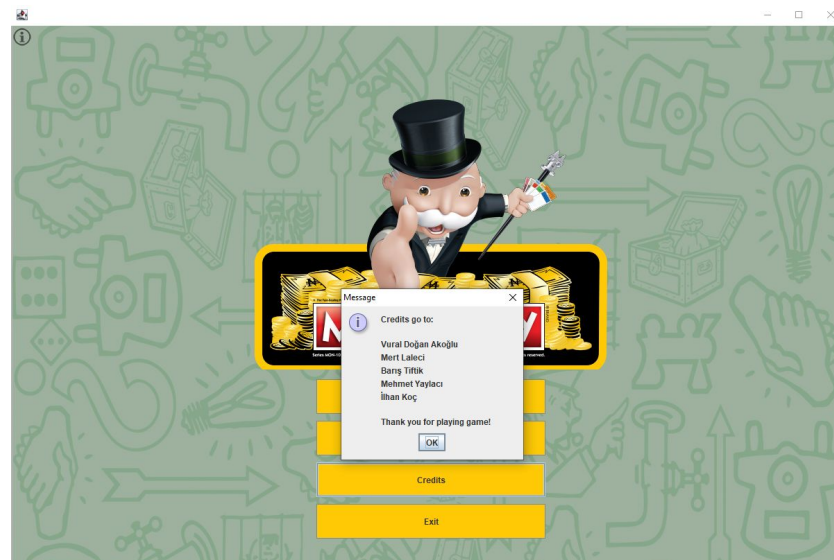


Figure 6: Credits pop-up

The user can click the button “Credits” to find out the people’ name who have contributed to the Monopoly Bilkent Edition.

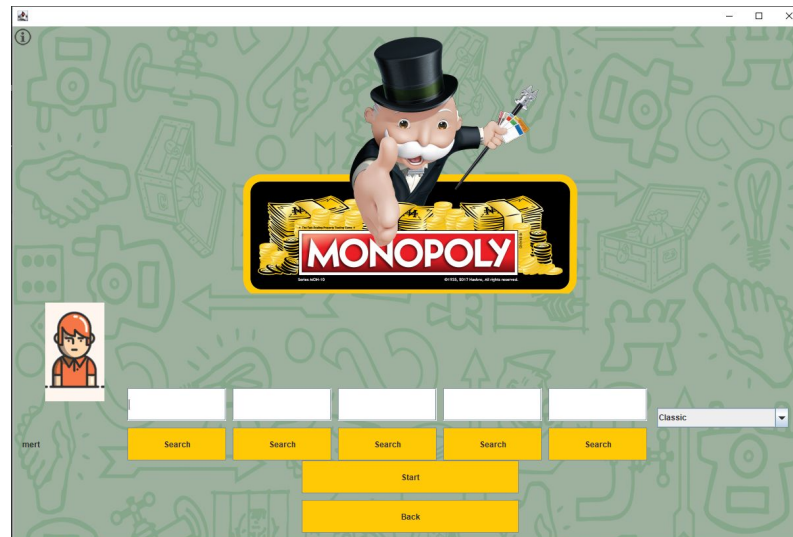


Figure 7: Lobby Screen

After the user click the create lobby button to play the game, see this screen. In the Lobby screen, the user can add opponent players by writing their username up to 5 if they are already in the system, see the message “successful”, otherwise see the message “not exist user” , and select game mode to start the game.

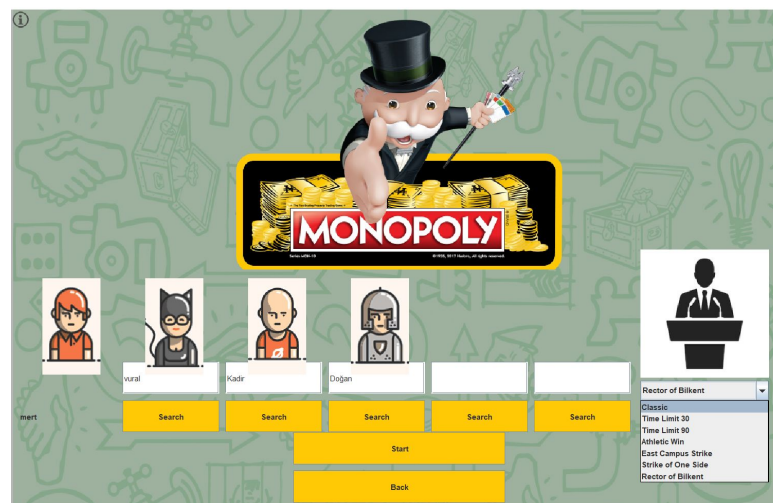


Figure 8: Lobby Screen

The user can click the button “Start” to start the game after adding players and selecting mode.

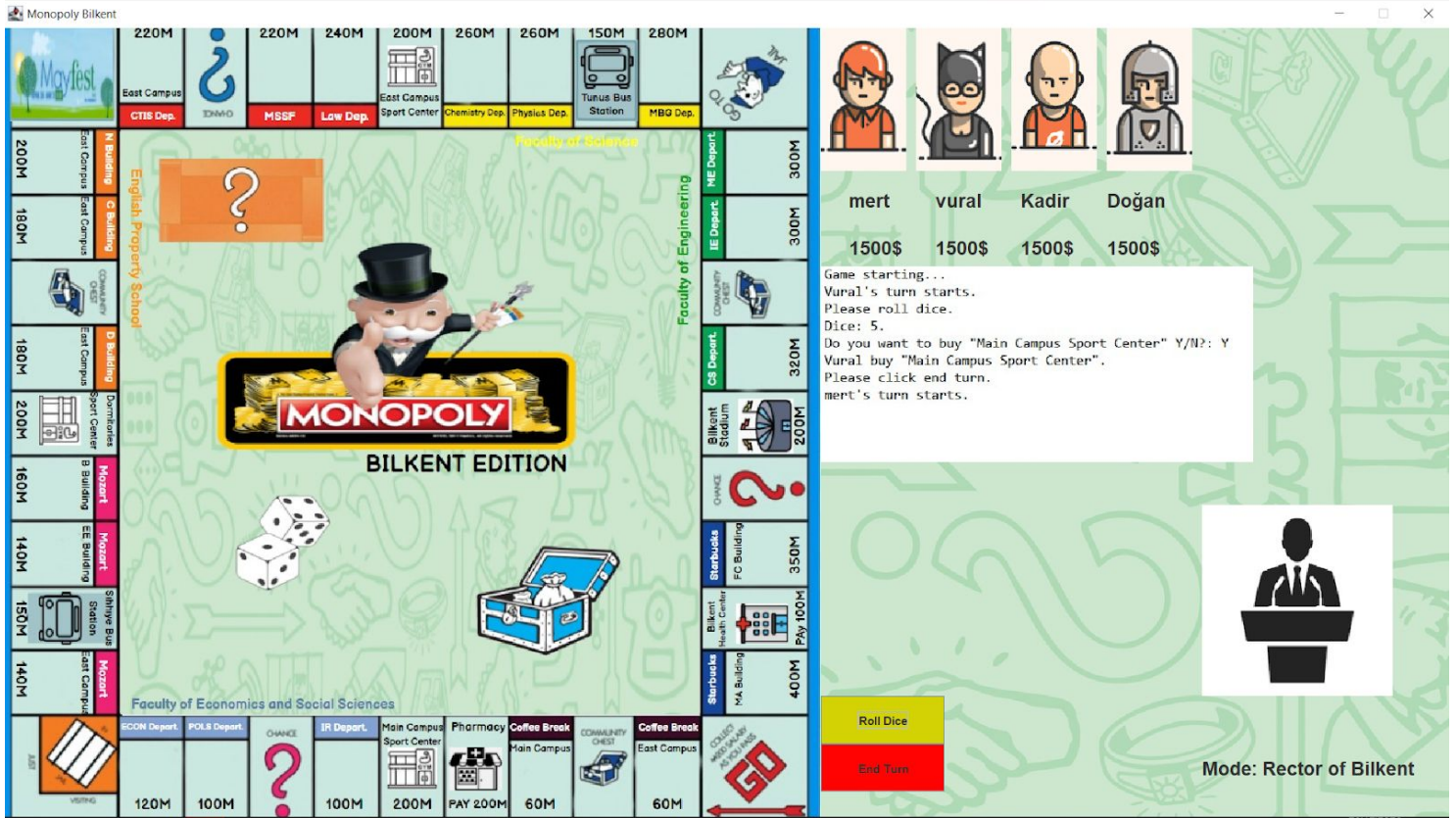


Figure 8: In game Screen

The game screen has two parts left and right, the board of the game will be on the left and players with their avatars, usernames and their bank accounts in the same order of the lobby on the right side of the screen. Also, the right side of the game screen has an interface which describes the game flows and Mode icon which shows the game mode to remember how to quickly finish this game to players, “roll dice” and “end turn” buttons for players to use during their turns.

5. Build Instructions

- Enter the link below to see our reports and implementation of the game
<https://github.com/mehmetaylacci/2G-Monopoly-Hasbrothers>
- Clone the project to any Java IDE or via command line and run (we have tested in IntelliJ).
- After running the game, you can either check the info boxes at the upper-left of the screen or click the “How to play” button on the main page.

6. What Is Completed?

We have created sign-in and sign-up pages. These pages work properly and recognize the users that have signed-up. We have added avatars that users can choose when they are signing-up. We have added the lobby implementation. Users can open an offline lobby and add players that are in the list of registered users. Our code also implements modes that change the dynamics of the game. As we have not completely implemented the game, choosing modes will change the texts in the GUI and will also change the variables on the backend of the code.

7. What Is Missing?

Since our game was not implemented completely, half of our in-game functional requirements including sound effects are missing as we could not connect the gui parts of the game with the abstract parts of the game. Some of the missing parts are game logic with the main loop and the connection between entity - boundary objects. Also, the GUI parts of the code lack updating the board like moving the players or building new houses. GUI implementations of cards are lacking. Our code also misses cell initialization in the Board class. Although we use sign-up and sign-in systems as our authentication method. Our system is not safe for important personal data since we save users' information in an unprotected .txt file.

8. Work Allocations

Barış:

- **Analysis Report: Class Diagram** and its explanations
- **Design Report: Class Diagram and UI Diagram and TradeOffs**
- **Implementation:**
Board.java, Card.java, CardLocation.java, Cell.java, ChanceCard.java, ChanceLocation.java, CommunityChestCard.java, CommunityChestLocation.java, Dice.java, Estate.java, GameEngine.java, JailLocation.java, Location.java, Property.java, Rentable.java, SpecialLocation.java, SportProperty.java, StartLocation.java, TaxLocation.java.

Mehmet:

- **Analysis Report: Class Diagram, Sequence Diagram** and their explanations
- **Design Report: Class Diagram and UI Diagram**
- **Implementation:**
Mode.java, Property.java, CreateLoby.java, Login.java, Board.java, Player.java and helped: **MainPage.java**

Vural:

- **Analysis Report :** Use-case , state diagrams, activity diagrams, Mock-ups of the game, functional, non-functional requirements
- **Design Report :** Subsystem decomposition, hardware software mapping, access control and security, server-side, improvement summary
- **Implementation:**
SignUp.java, CreateLoby.java, MainPage.java, Login.java, BoardGUI.java classes and GUI implementation of them

Mert:

- **Analysis Report** : Use-case, state diagrams, activity diagrams, Mock-ups of the game, current system, functional, non-functional requirements, improvement summary
- **Design Report** : Design goals, Boundary Conditions, server-side (final object design),
- **Implementation**: SignUp.java, CreateLobby.java, Login.java, MainPage.java, BoardGUI.java classes and GUI implementation of them, design and creation of the game board.png

Ilhan:

- **Implementation**: ChanceLocation.java, CommunityChestLocation.java, player.java, initializations of 32 cards in Board.java