CS 319 - Object-Oriented Software Engineering

Fall 2020

Final Report

Monopoly

Group 3D

Cansu Moran - 21803665

Elif Gamze Güliter - 21802870

Melisa Taşpınar - 21803668

Öykü Irmak Hatipoğlu - 21802791

Yiğit Gürses - 21702746

Table of Contents

1 Implementation Process **4**
- 1.1 Work Allocation of the Team 4
  - 1.1.1 Cansu 4
  - 1.1.2 Gamze 5
  - 1.1.3 Melisa 5
  - 1.1.4 Öykü 5
  - 1.1.5 Yiğit 5

2 System Requirements & Build Instructions **6**
- 2.1 System Requirements 6
- 2.2 Build Instructions 6

3 User’s Manual **7**
- 3.1 Main Menu 7
- 3.2 Credits 8
- 3.2 Help 8
- 3.3 Edit a Game 9
  - 3.3.1 Board Selection 9
  - 3.3.2 Edit 10
    - 3.3.2.1 Selecting the Square Type 11
    - 3.3.2.2 Editing a Square as Joker 12
    - 3.3.2.3 Editing as Property 13
    - 3.3.2.4 Selecting Color Group 13
    - 3.3.2.5 Adding New Color Group 14
    - 3.3.2.6 Editing as Chance and Community Chest 15
    - 3.3.2.7 Uploading Icon Pictures and Board Pictures 16
    - 3.3.2.8 Changing currency, rent rate, mortgage rate and board name 16
    - 3.3.2.9 Saving the Board 16
- 3.4 Play the Game 17
  - 3.4.1 Board Selection 17
  - 3.4.2 Player Selection 18
  - 3.4.3 Play 19
    - 3.4.3.1 Roll Dice 20
    - 3.4.3.2 Landing on a Unowned Property 21
    - 3.4.3.3 Auction 22
    - 3.4.3.4 Landing on Community Chest or Chance 23
    - 3.4.3.5 Chance or Community Chest Card 24
    - 3.4.3.6 Landing on Joker Places 25
    - 3.4.3.7 Landing/Clicking on a Owned Property (from the owner’s point of view) 26
    - 3.4.3.8 Landing on a Owned Property (from the another player’s point of view) 27

3.4.3.9 Start Square 28
3.4.3.10 Next Turn 28
3.4.3.11 Resigning 29
3.4.3.12 Losing the game 30
3.4.3.13 Winning the Game 31

4 What is Missing? 31

5 Conclusion 32

1 Implementation Process

We began the implementation process mainly after the first iteration. Since the programming language we had chosen was Java, we decided to use the integrated development environment IntelliJ, which all members already had on their personal computers. We used our main GitHub depository called cs319-Monopoly-3D, to which we connected using our IntelliJ programs, and pushed & merged our code whenever we added, developed or fixed a class.

In retrospect, we developed our implementation in a way that we made all the grand decisions together. For this reason we had a Zoom meeting among the members whenever there was a relatively big decision to be made or an issue to be solved. Plus, we also had meetings when we each completed our tasks and needed to split the tasks of the next step of implementation. For small decisions and problems, we communicated using our WhatsApp group chat. Still, all in all, we had to have frequent Zoom meetings.

1.1 Work Allocation of the Team

The work allocation of our team, describing the work done by each member throughout the semester is as follows:

1.1.1 Cansu

- I made some of the Mockups, some sequence, class & object diagrams with Öykü.
- I decided on design patterns and wrote them on reports. I wrote high-level software architecture on the design report and functional requirements in the analysis report.
- I worked on the entire backend of the gameplay. This included the game logic as well as all the game entities, which were used by the editor. I worked with Öykü, who was responsible from the front end. I connected all the frontend elements to the backend.

1.1.2 Gamze

- Regarding reports, I contributed to the overview, and functional requirements parts in the Analysis Report and also drew some of the sequence diagrams.
- I wrote explanations and scenarios of the class diagram and sequence diagrams.
- In the design report, I wrote most of the class and method explanations of the report and helped my teammates in need.
- Regarding the codes, I was responsible for the backend side of the Editor. I implemented all the backend operations which are needed to edit a default board.
- I worked with Öykü while combining her FXML design files with the backend side.
- I added listeners (to frontend components) and operations to process user input.
- At last, I worked with Cansu, I wrote the Card class for her.

1.1.3 Melisa

- I created the subsystem decomposition & drew its diagrams in both iterations.
- I drew our activity diagram(s) based on the official Monopoly rulebook.
- In the design report, I wrote Architectural Style and Subsystem Services and High-Level Software Architecture: Subsystem Decomposition.
- In the analysis report, I wrote comprehensive explanations for the activity diagrams.
- I coded ScreenManager with Yiğit, coded screens like Help and Credits, helped Yiğit implement Board & Player Managers, Board & Player Selection Screens.
- I helped Yiğit with the use-case and state diagrams.
- I dealt with the editing and formatting of our reports.

1.1.4 Öykü

- I was responsible for the overall UI design of the game.
- In reports, I did the Mockups, sequence diagrams, the class & object diagrams with Cansu and wrote the explanations of the diagrams.
- I wrote the non-functional requirements part and overview part of the analysis report. I wrote the improvement summaries for both 2nd iteration reports.
- I checked all of the reports and gave feedback to my group mates.
- Regarding the code, I made all of the “fxml” files and connected them into their controls, for example buttons and textfields.

1.1.5 Yiğit

- I drew the use-case diagram and created the state diagrams.
- In the Analysis Report I wrote explanations for the use case and state diagrams.
- I implemented FileManager, BoardSelection, PlayerSelection & MainMenu screens.
- I implemented the ScreenManager with Melisa.
- I linked the classes to the file system, to write/read data onto/from the file system.
- I helped Melisa with subsystem decomposition and the activity diagram.

2 System Requirements & Build Instructions

This section has the system requirements of our software & step-by-step build instructions.

2.1 System Requirements

The computer needs to have an up-to-date JVM and sufficient space in the memory, though the game is small. The user should also have a Java IDE, of their preference. The game works both on MacOS and Windows, as can be seen from the screenshots in this report.

2.2 Build Instructions

The steps for building our game using the source code are as follows:

1. The GitHub link with which to find our source code and documentation is https://github.com/gamzeguliter/cs319-Monopoly-3D
2. clone the github project with the command:
   git clone https://github.com/gamzeguliter/cs319-Monopoly-3D
3. Open your java IDE, import the file called “code” in the cloned folder as a new project
4. Add “json-20201115.jar” in the “code” folder to the modules of your IDE
   a. If you are using IntelliJ you can do this by going to **File->Project Structure->Modules->Dependencies**, add a new module as a jar and select the file “json-20201115.jar”
5. Download javafx sdk and add it to your IDE modules as java code
   a. If you are using IntelliJ you can do this by going to **File->Project Structure->Modules->Dependencies**, add a new module as a java code and select the folder “javafx-sdk-*/lib”
   b. Check this link for more information https://www.jetbrains.com/help/idea/javafx.html#vm-options

6. Add the following to the VM options of your Run/Debug configurations: "--module-path <<full path to javafx-sdk-11.0.2/lib>> --add-modules javafx.controls,javafx.fxml"
    a. If you are using IntelliJ you can find VM options through **Run/Debug Configuration->Edit Configurations** which can be find next to the run icon
7. Build Project and then Run Project

3 User's Manual

Below is an almost step-by-step guide as to how to play our game and navigate through the system, with screenshots to help novel users learn our software.

3.1 Main Menu

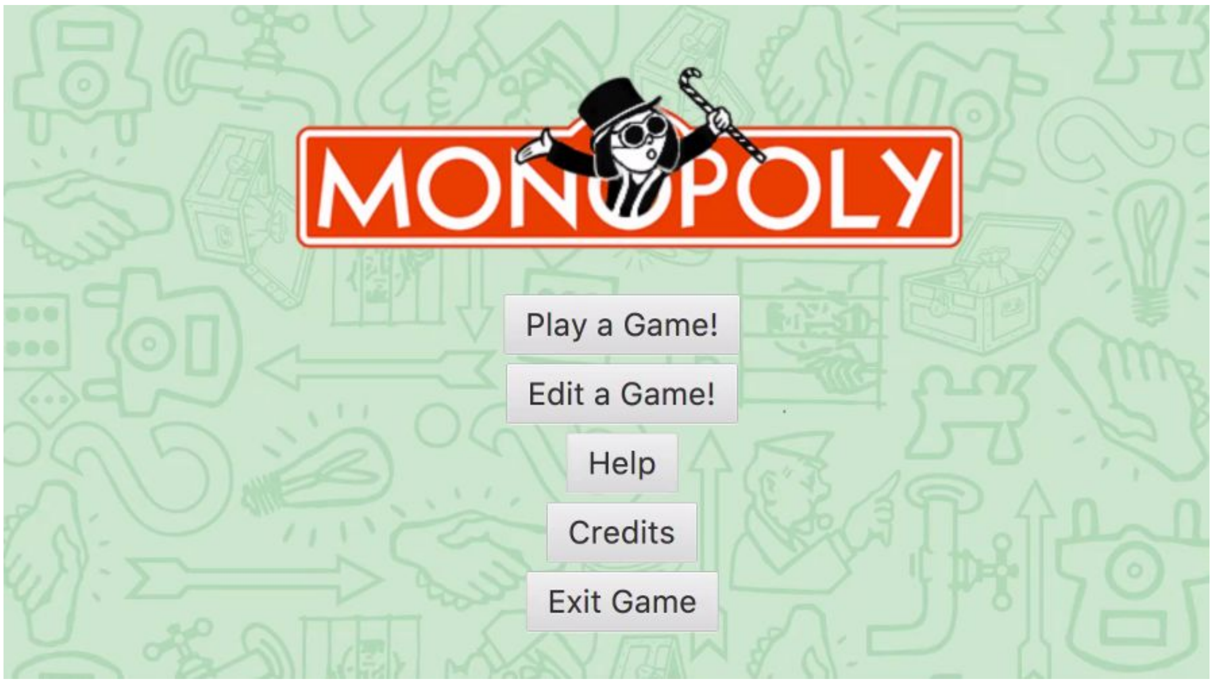


Figure 1. Main menu screen

The game starts on the Main Menu, as can be seen below. Here, you can display things like help and credits by pressing the "Help" or "Credits" button, play a game by pressing "Play a

Game!” button, edit a board by pressing “Edit the Game!” button or you can exit the game by pressing the “Exit Game” button.

3.2 Credits

If the user clicks on “Credits” on the Main Menu, they are taken to the credits screen. On this screen, the names of our group members are written. No screenshots are added for the sake of concision, as this is not an interesting or complicated scenario.

3.2 Help

If the user clicks on “Help” on the Main Menu, they are taken to the help screen. On this screen, the user can find information on how to play the game.

3.3 Edit a Game

3.3.1 Board Selection

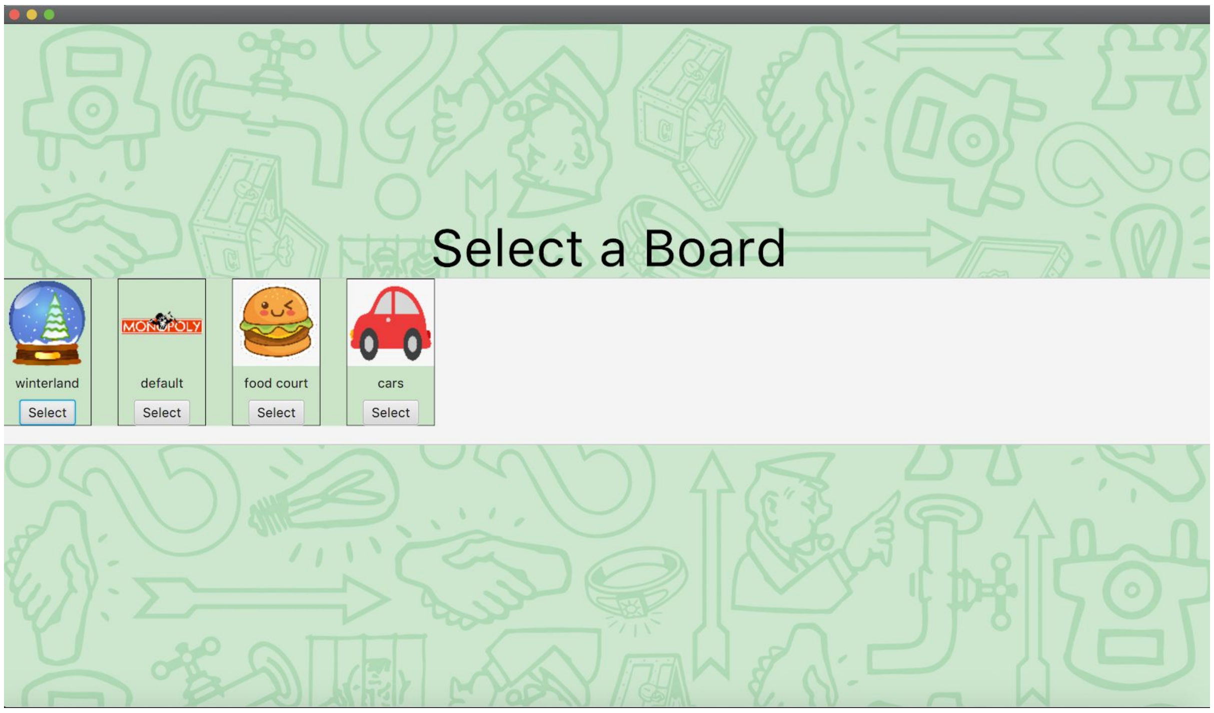
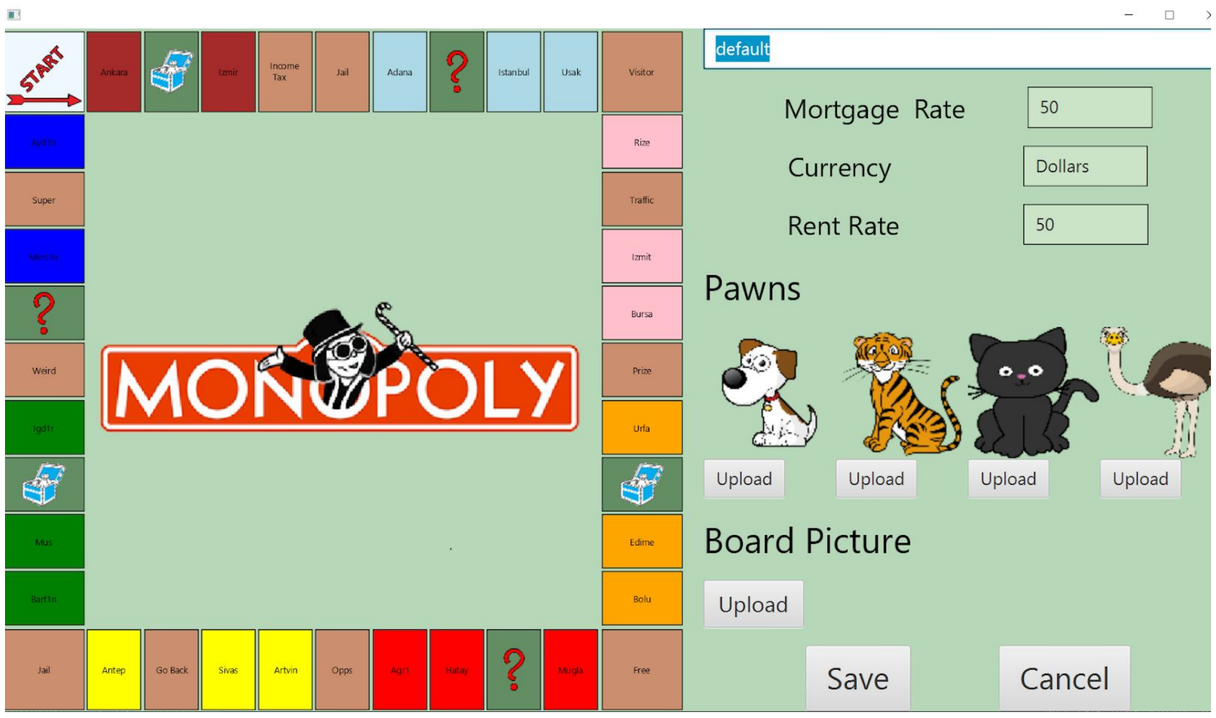


Figure 2. Board selection screen

After pressing on the "Edit a Game!" button, you will be directed to the Board Selection Screen. From this screen, you can choose the board you want to edit.

3.3.2 Edit



default

Mortgage Rate: 50

Currency: Dollars

Rent Rate: 50

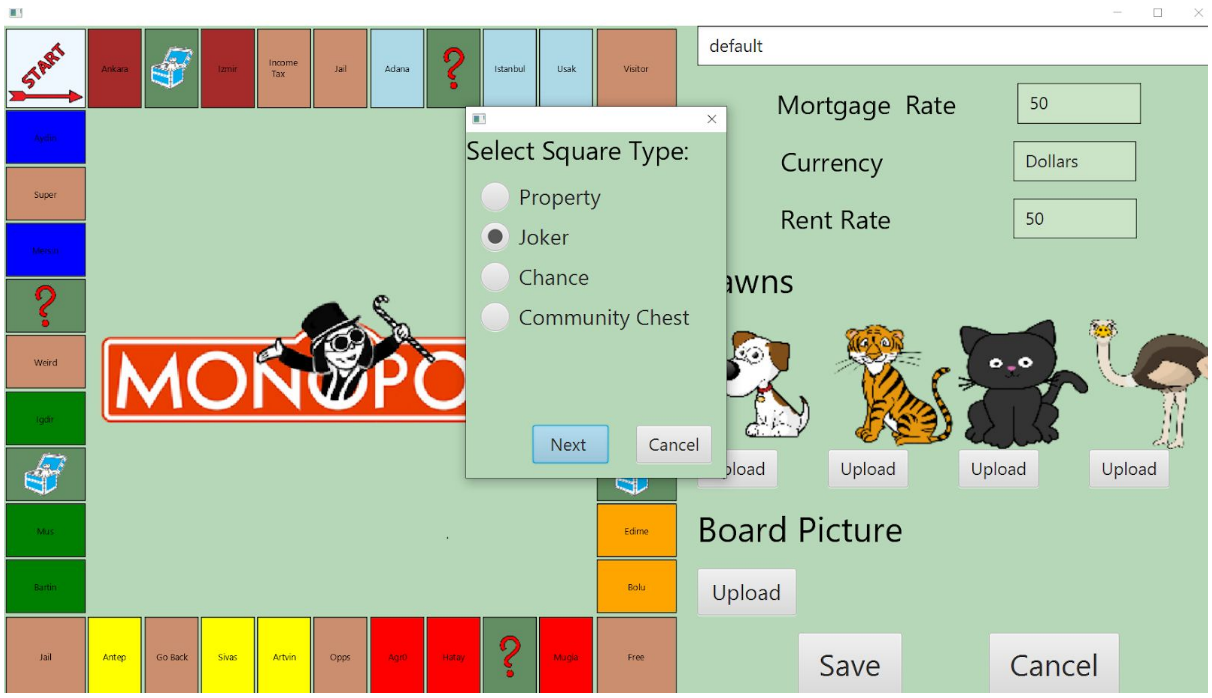
Pawns

Board Picture

Figure 3. Editor screen

If and when the user chooses the “Edit a Board” option on Main Menu, they are taken to the main editing screen. Here, they can change variables like mortgage rate, currency and so on. Plus, they can change pawns, board picture and the properties of individual properties.

3.3.2.1 Selecting the Square Type


Select Square Type:

- Property
- [✓ Joker]
- Chance
- Community Chest

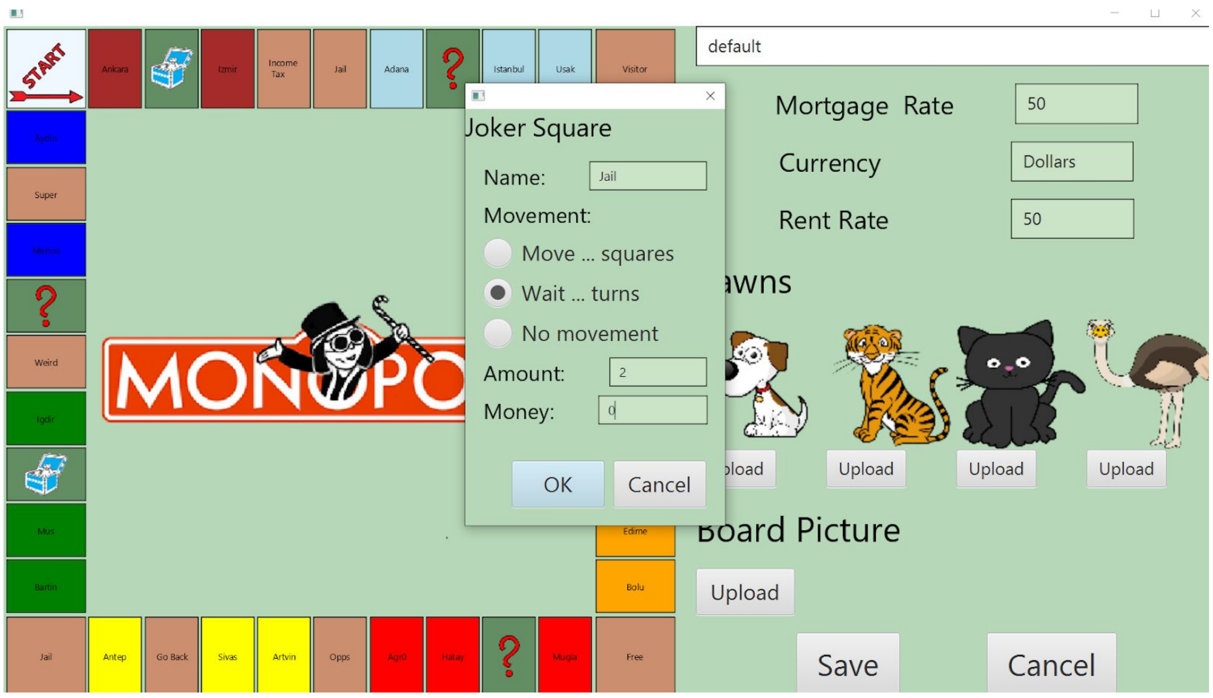
default

Mortgage Rate	50
Currency	Dollars
Rent Rate	50

Figure 4. Select square type dialog

The user can click on a square on the board, and then choose to make that a Joker square, on the dialogue that appears.

3.3.2.2 Editing a Square as Joker



Joker Square

Name: Jail

Movement:

Move ... squares

[✓ Wait ... turns]

No movement

Amount: 2

Money: 0

default

Mortgage Rate 50

Currency Dollars

Rent Rate 50

Figure 5. Select square type dialog

For that newly created Joker square, the user will be asked to assign a rule to that square. This square can have the power of moving the incomer players, or making them wait for a specified number of turns. Plus, it can change the balance of the players who land there. The power of this square is decided on this dialogue.

3.3.2.3 Editing as Property

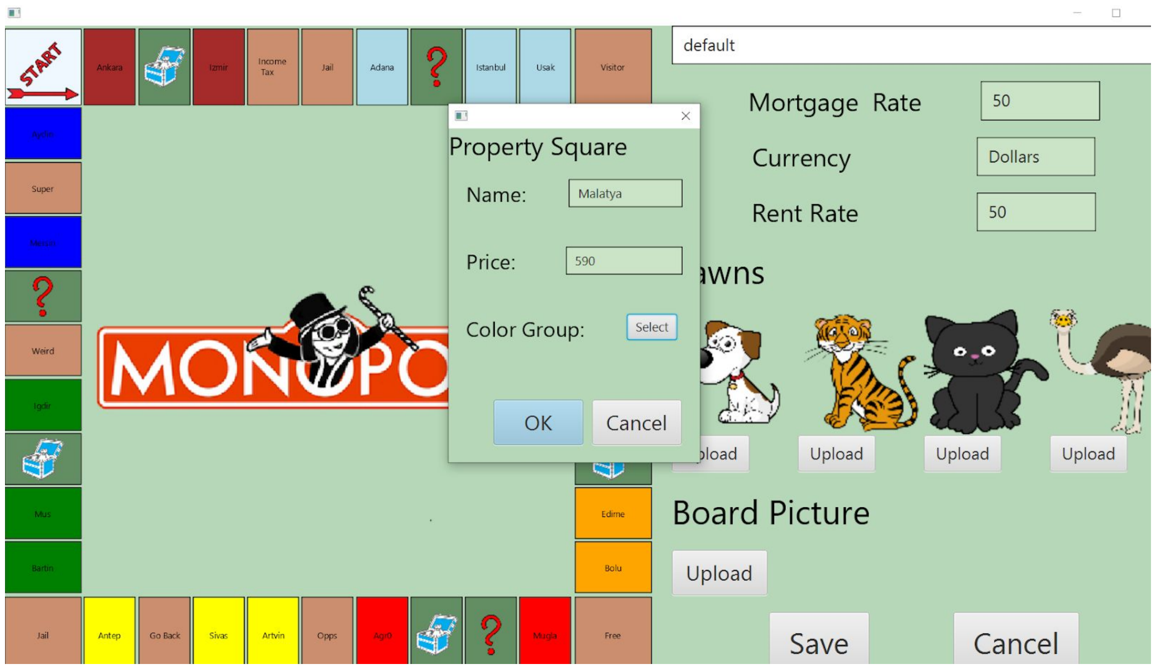


Figure 6. Edit property dialog

If the user clicks on a property, a dialogue appears where they can change this property's color group, name and price.

3.3.2.4 Selecting Color Group

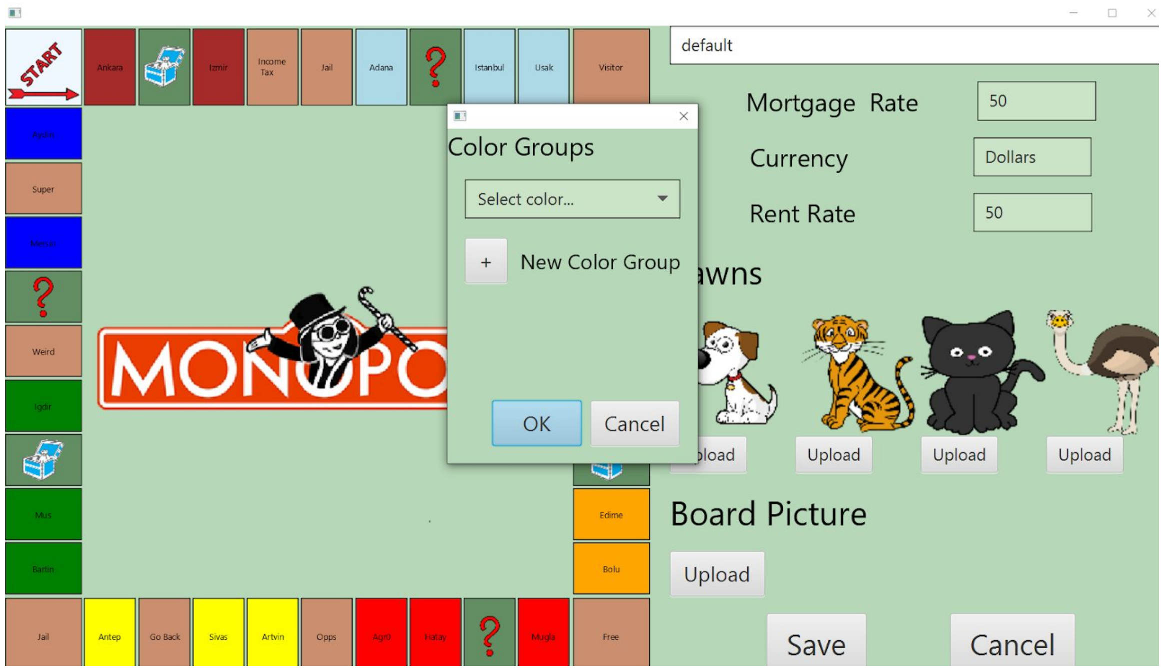


Figure 7. Select color group dialog

In order to change the color group, the user should click on the select button next to color group in the previous dialogue. Here, in this dialogue, they can select a pre-existent color group or create a new one.

3.3.2.5 Adding New Color Group

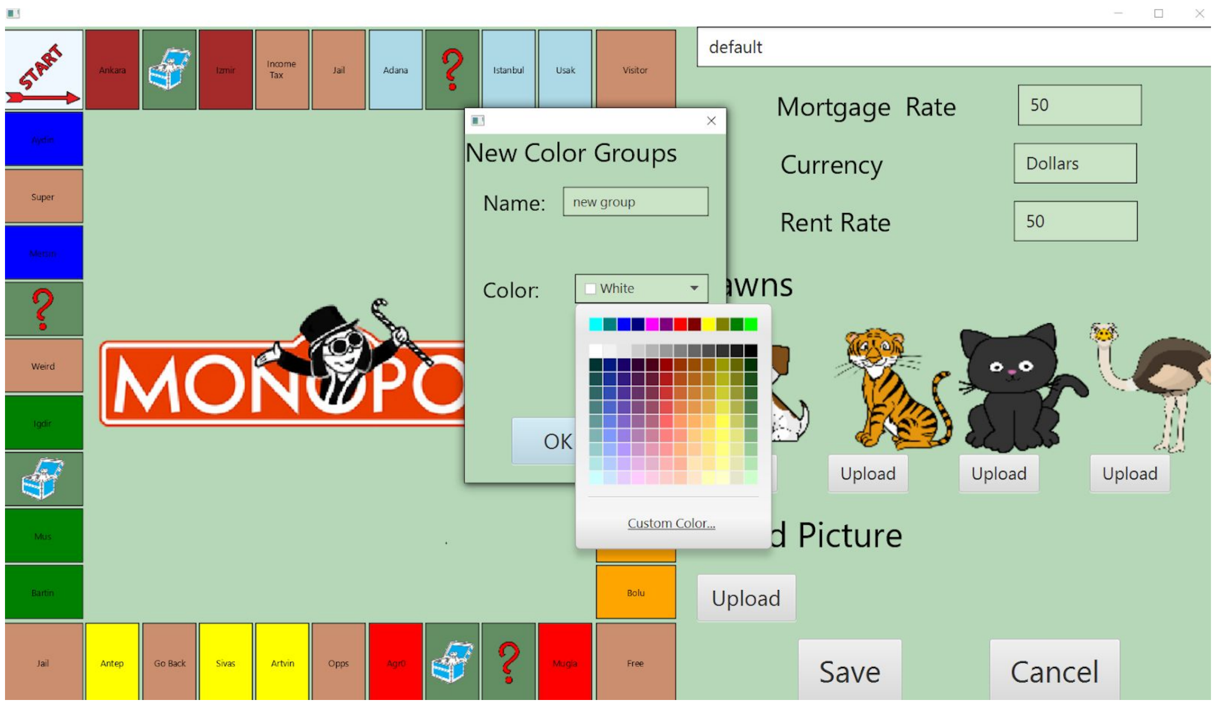


Figure 8. Adding new color dialog

If they wish to create a new color group, after pressing the plus button next to “New Color Group”, a color picker appears where the user can create a new color group.

3.3.2.6 Editing as Chance and Community Chest

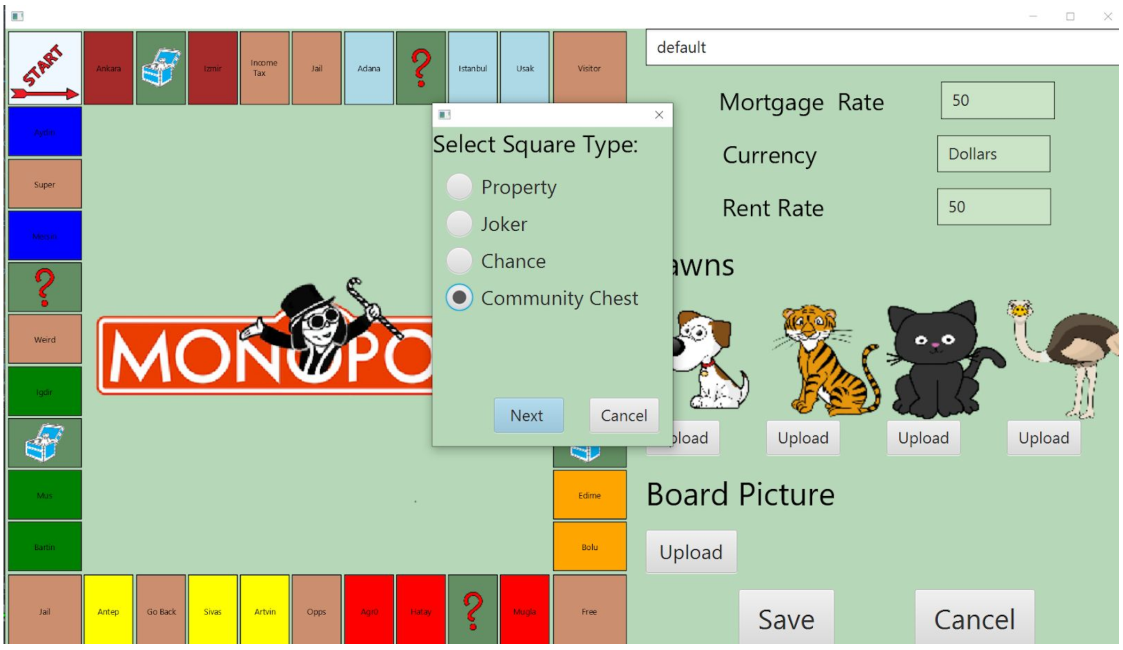
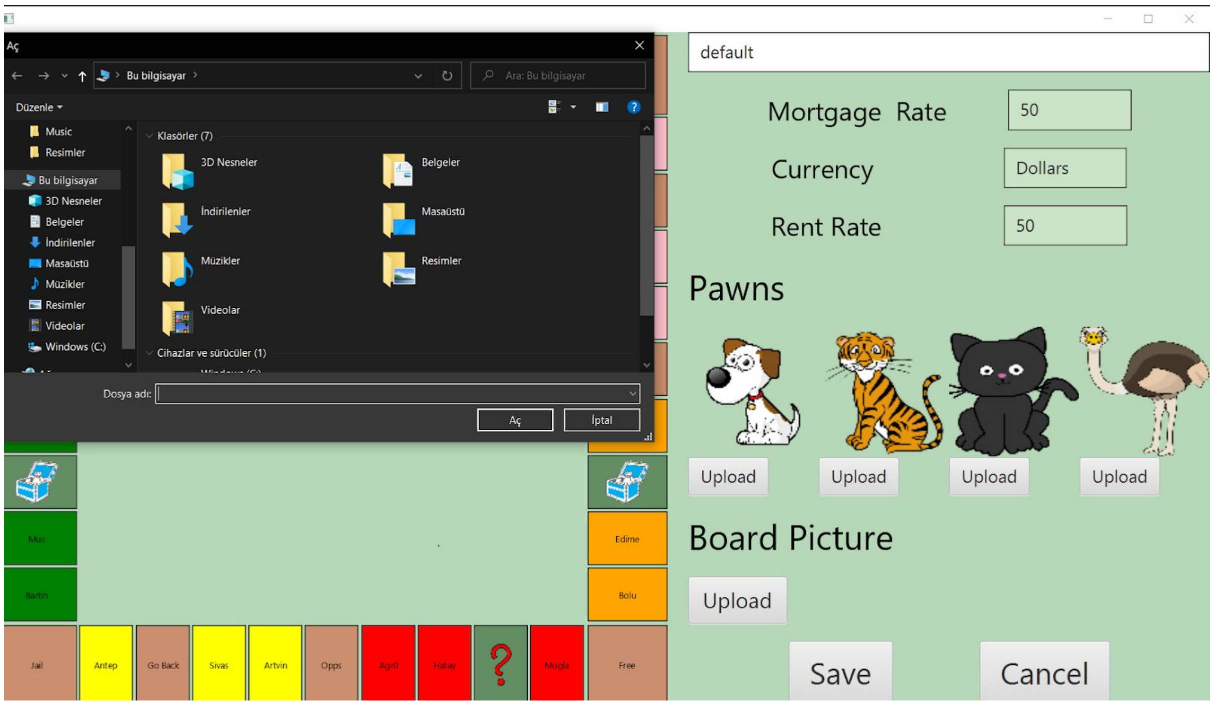


Figure 9. Select square dialog, community chest selected

In order to set a square as Chance or Community Chest, you should press on the Chest or Community Chest ratio buttons on the Select Square Type screen.

3.3.2.7 Uploading Icon Pictures and Board Pictures


default

Mortgage Rate 50

Currency Dollars

Rent Rate 50

Figure 10. Uploading pictures from local computer

At the right side of the editing screen, there are four icons that can be used in the game as player icons. By pressing the upload buttons under each icon, you can upload a picture from your local computer in order to use it as an icon.

You can also change the board picture by pressing the “ Upload” button under the Board Picture text.

3.3.2.8 Changing currency, rent rate, mortgage rate and board name

You can change the name , currency, mortgage rate and the rent rate of the game from the editing screen. In order to do this, you should change the default texts in the text boxes.

3.3.2.9 Saving the Board

After the editing is done, you should press the “Save” button in order to save your new Monopoly board.

3.4 Play the Game

3.4.1 Board Selection

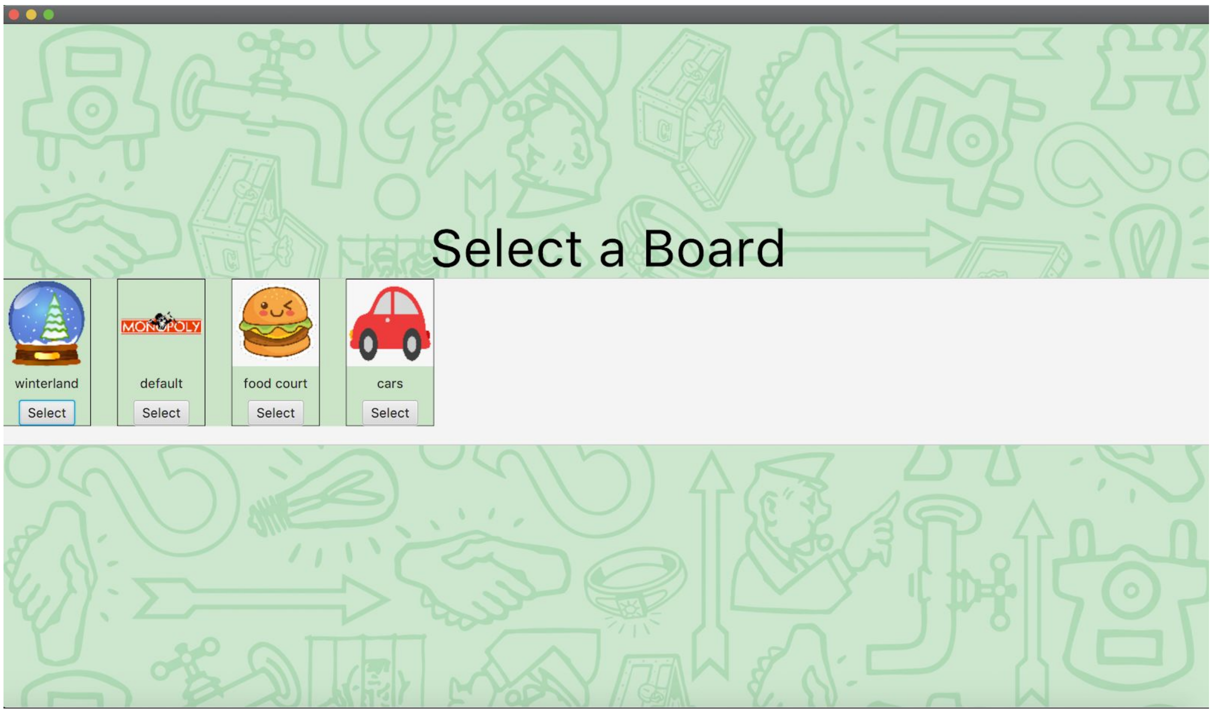
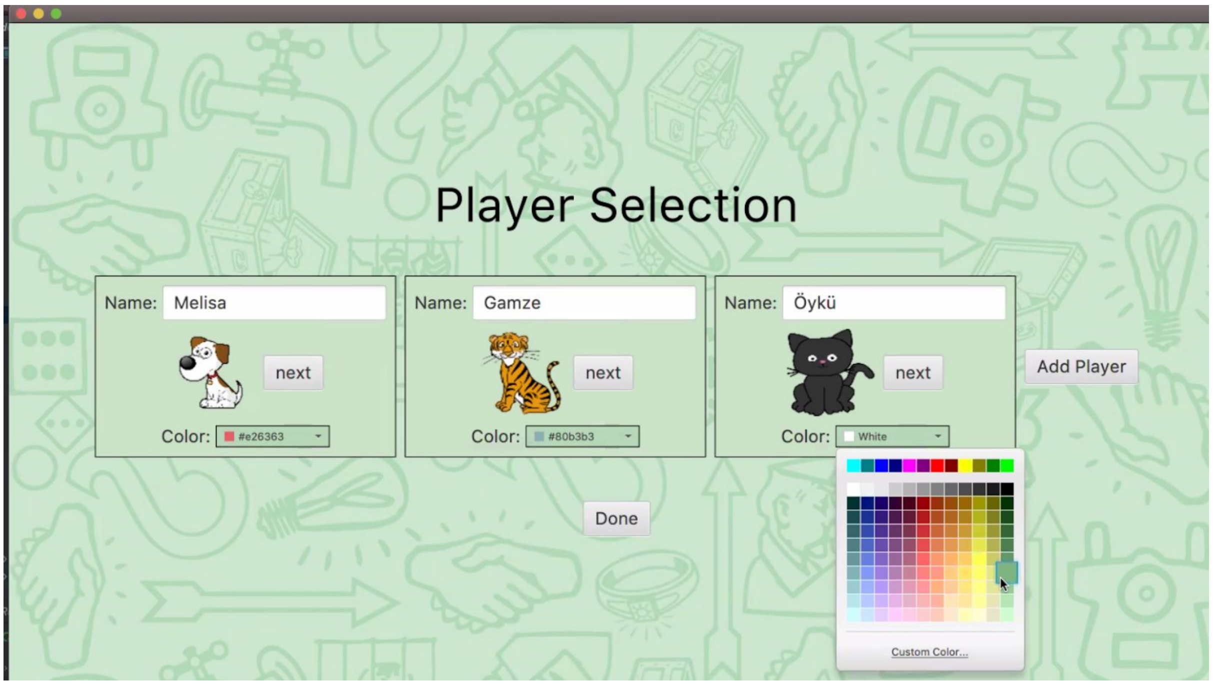


Figure 11. Board selection screen

After pressing on the "Play a Game" button, you will be directed to the Board Selection Screen. From this screen, you can choose the board you want to play. You can either choose an edited board or the default board.

3.4.2 Player Selection


Player Selection

Name: Melisa

Color: #e26363

Name: Gamze

Color: #80b3b3

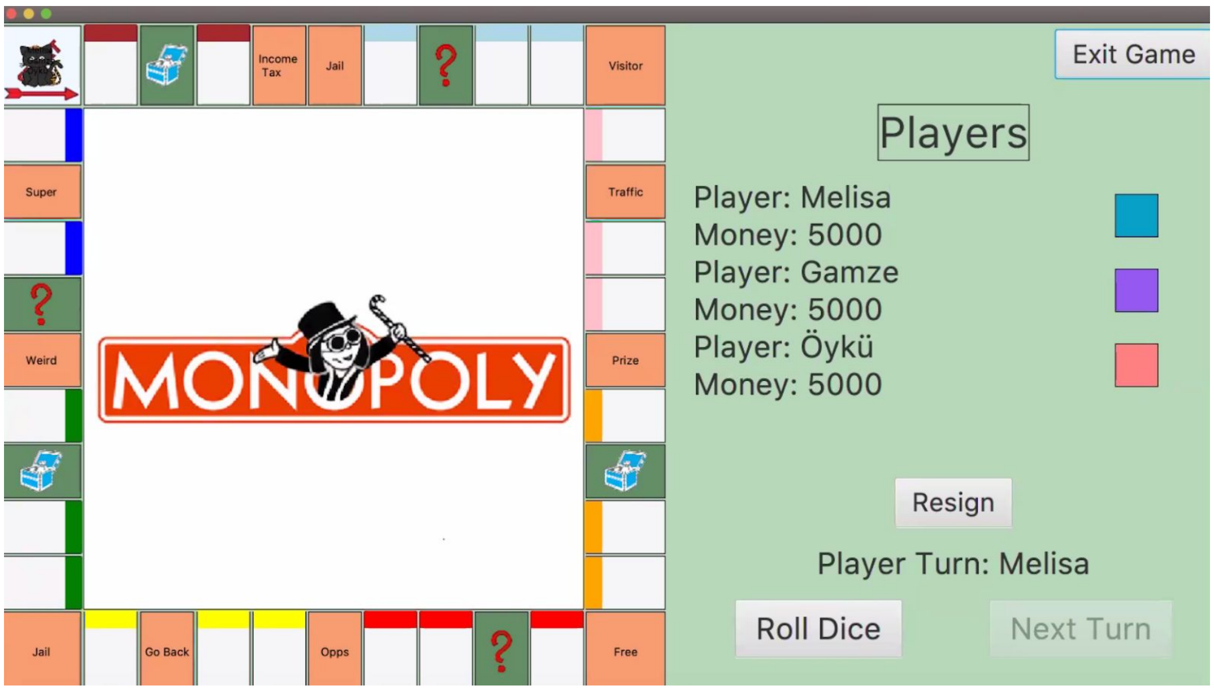
Name: Öykü

Color: White

Figure 12. Player selection screen

After choosing the board you want to play, you will be directed to the Player Selection Screen. From this screen, you can choose the name, icon and the related color of the player. After that, you should press the “Add Player” button in order to add the player to the game. You can add at 2-4 players and you cannot give the same color to the multiple players. When you press “Done”, you will be directed to the game screen.

3.4.3 Play



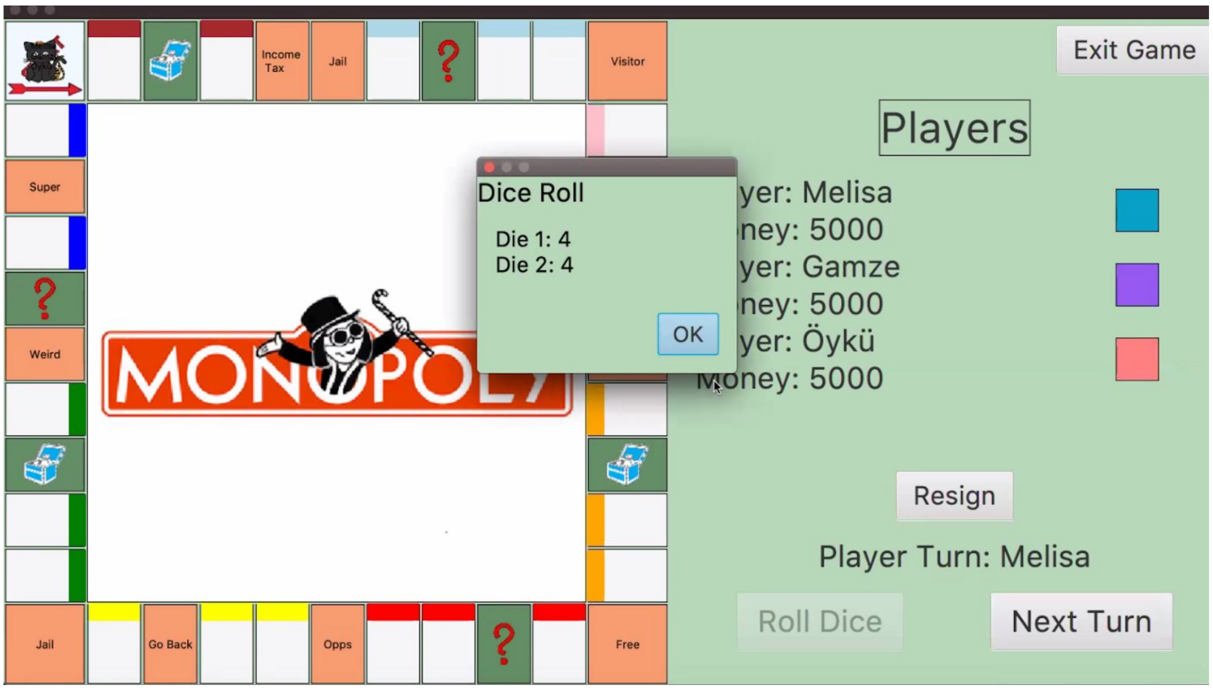
Player: Melisa
Money: 5000
Player: Gamze
Money: 5000
Player: Öykü
Money: 5000

Player Turn: Melisa

Figure 13. Play screen

After choosing the players and the board, the game will start. The first player will be the player who was added first.

3.4.3.1 Roll Dice



Dice Roll

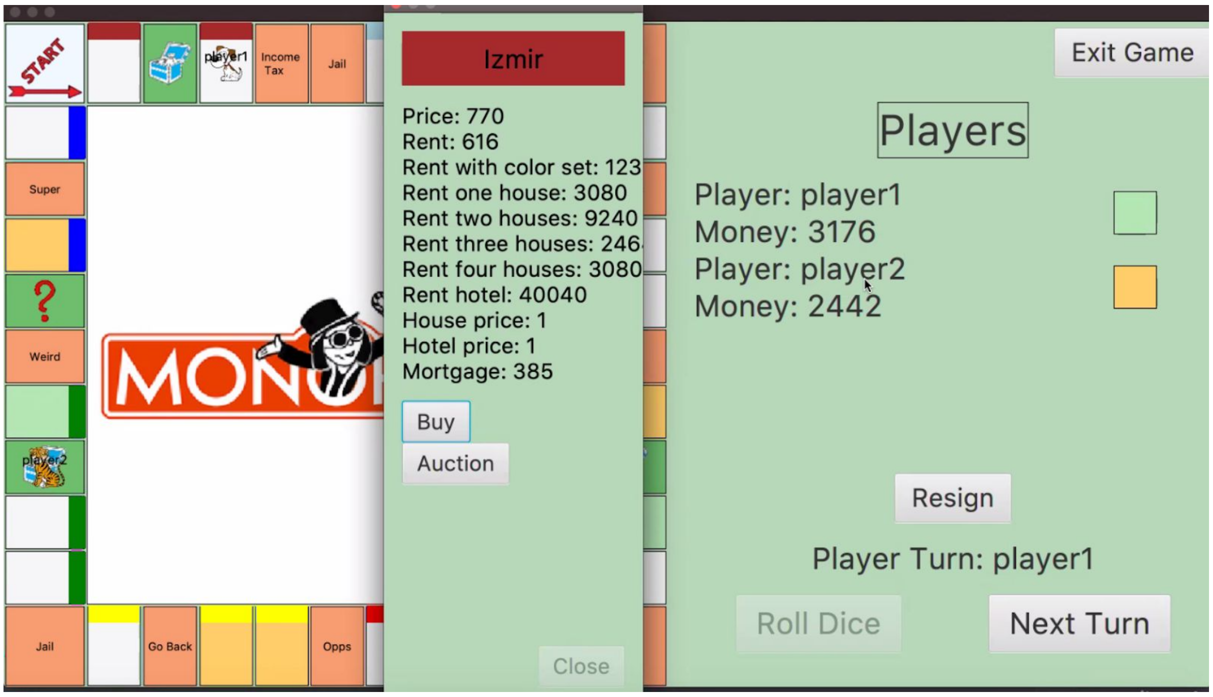
Die 1: 4
Die 2: 4

OK

Figure 14. Roll dice dialog

In order to roll dice, you should press the “Roll Dice” button. After you roll dice, a screen will come up and you should press the “OK” button in order to continue. After you press “OK” , your icon will be moved automatically to the related square.

3.4.3.2 Landing on a Unowned Property



Izmir

Price: 770
Rent: 616
Rent with color set: 123
Rent one house: 3080
Rent two houses: 9240
Rent three houses: 246
Rent four houses: 3080
Rent hotel: 40040
House price: 1
Hotel price: 1
Mortgage: 385

Players

Player: player1
Money: 3176
Player: player2
Money: 2442

Player Turn: player1

Figure 15. Unowned property dialog

When a player lands on an unowned property, this property dialog will open. It can be seen that the color of the dialog is the default Monopoly color, owned by noone. One can buy this property by pressing the “Buy” button or if the player does not want to buy this property, he/she can press the “Auction” button to place the property on auction.

3.4.3.3 Auction

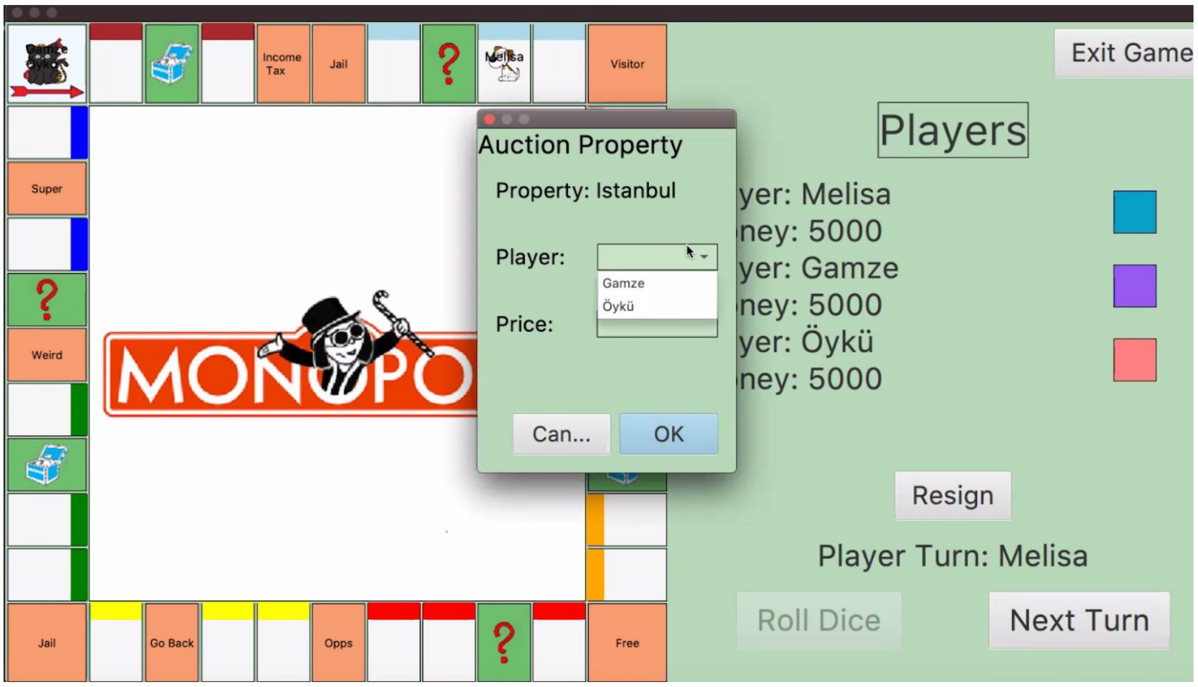
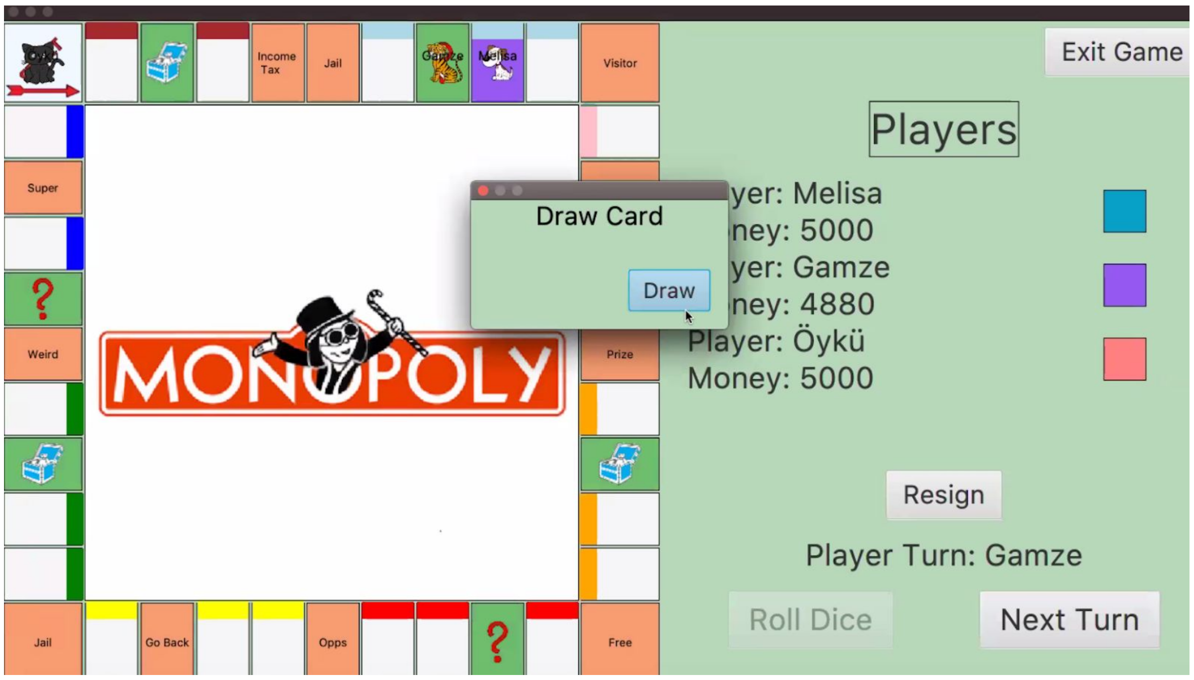


Figure 16. Auction dialog

When a player presses “Auction” on an unowned property, the Auction Property dialog will open. In this dialog, the player who won the auction and the price of the property will be given by players. If no one wants to buy the property, the players can press the Cancel button.

3.4.3.4 Landing on Community Chest or Chance



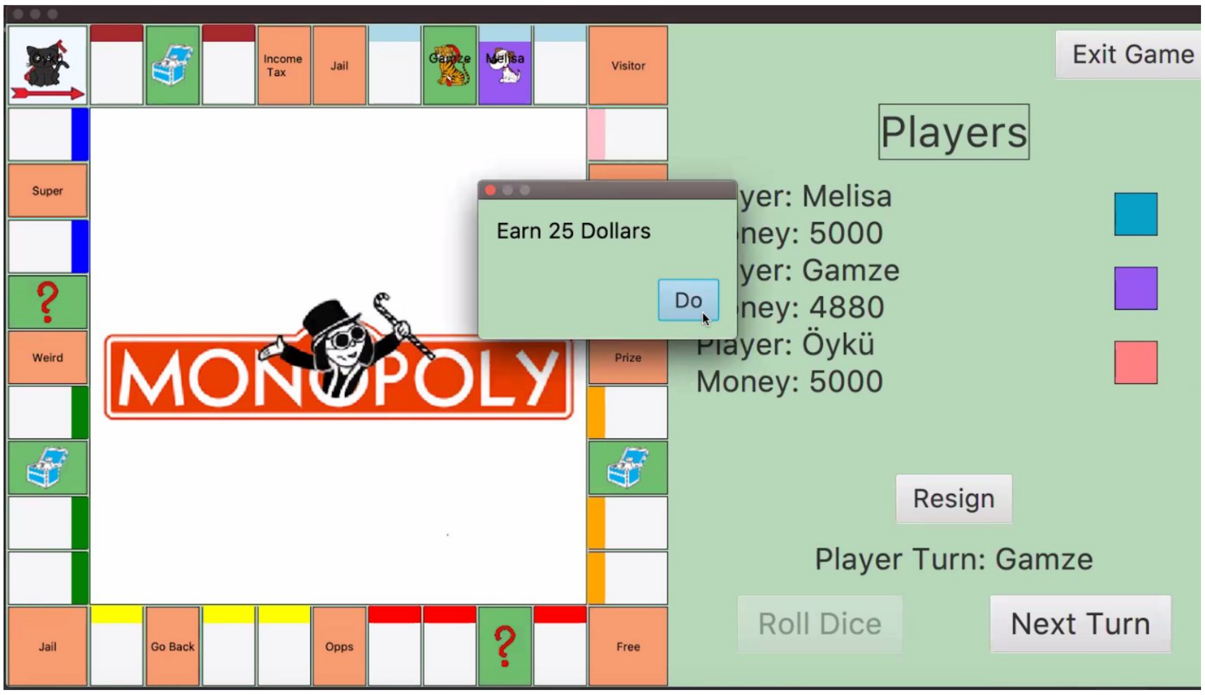
yer: Melisa
ney: 5000
yer: Gamze
ney: 4880
Player: Öykü
Money: 5000

Player Turn: Gamze

Figure 17. Draw chance or community chest card

When one lands on a community chest or chance square, a dialog that says Draw Card will pop up, the player can draw the card by pressing the Draw button.

3.4.3.5 Chance or Community Chest Card



Earn 25 Dollars

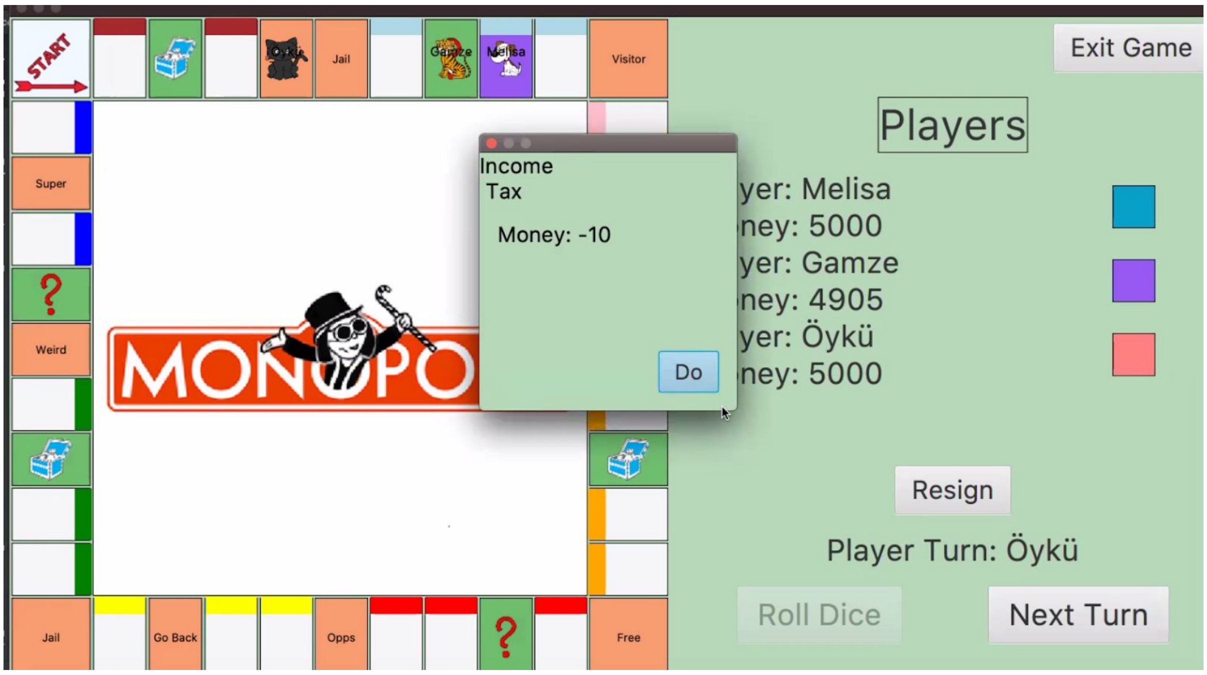
yer: Melisa
ney: 5000
yer: Gamze
ney: 4880
Player: Öykü
Money: 5000

Player Turn: Gamze

Figure 18. Chance or community chest card dialog

This is an example of a community chest card. By pressing the “Do” button on the dialog, one can execute the sentence on the card dialog.

3.4.3.6 Landing on Joker Places



Income
Tax

Money: -10

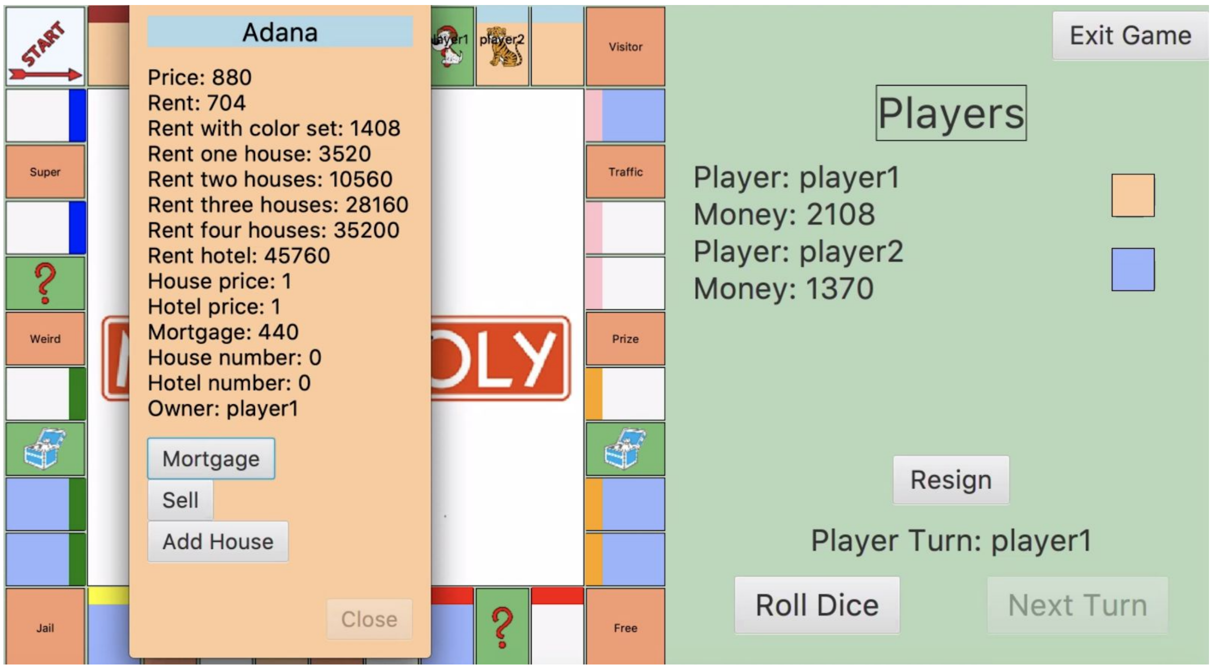
yer: Melisa
ney: 5000
yer: Gamze
ney: 4905
yer: Öykü
ney: 5000

Player Turn: Öykü

Figure 19. Joker square dialog

When a player lands on a joker square, this dialog will pop up. In this dialog, the amount of money earned or lost, the amount of squares to move backward or forward, or the amount of turns the player will wait is specified. By pressing the “Do” button, the game will implement what the card says.

3.4.3.7 Landing/Clicking on a Owned Property (from the owner's point of view)


Adana

Price: 880
Rent: 704
Rent with color set: 1408
Rent one house: 3520
Rent two houses: 10560
Rent three houses: 28160
Rent four houses: 35200
Rent hotel: 45760
House price: 1
Hotel price: 1
Mortgage: 440
House number: 0
Hotel number: 0
Owner: player1

Players

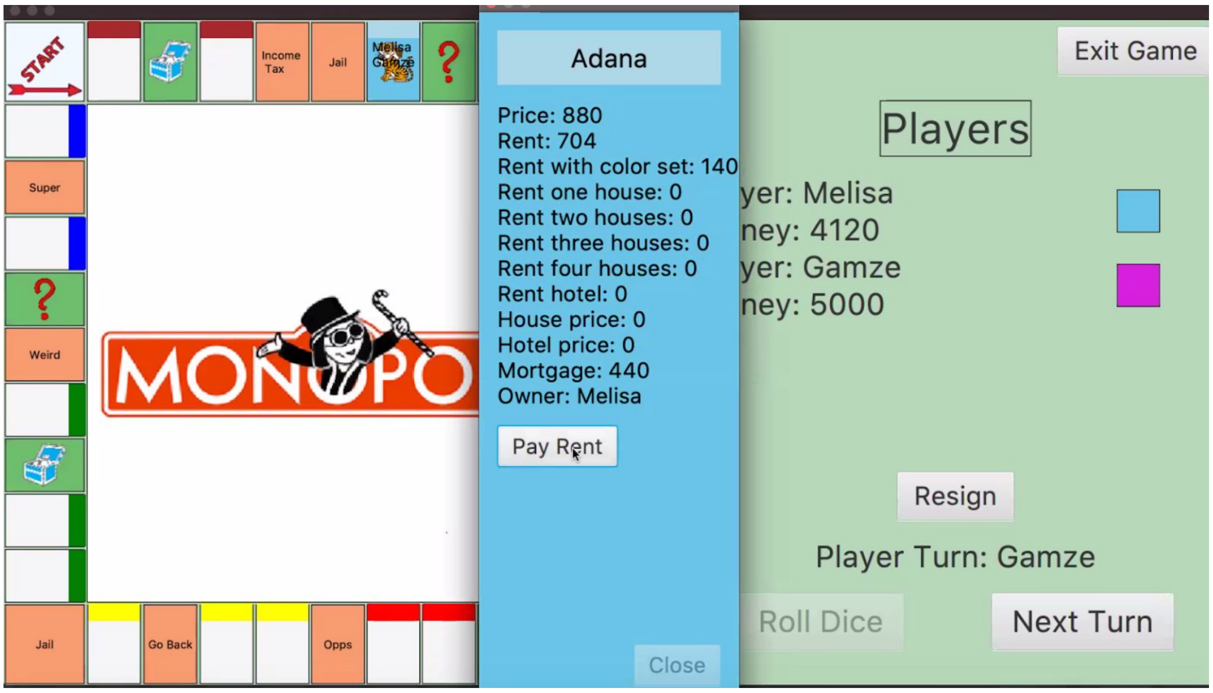
Player: player1
Money: 2108
Player: player2
Money: 1370

Player Turn: player1

Figure 20. Owned property dialog, seen by the owner of the property

When a player lands on or clicks on a property that he/she owns, this is the dialog that will pop up. It can be noticed that the dialog is painted in the player color of the owner. The owner player can press the "Mortgage" button to mortgage the property, "Sell" button to sell the property or "Add House" button to add a house to the property.

3.4.3.8 Landing on a Owned Property (from the another player's point of view)



Adana

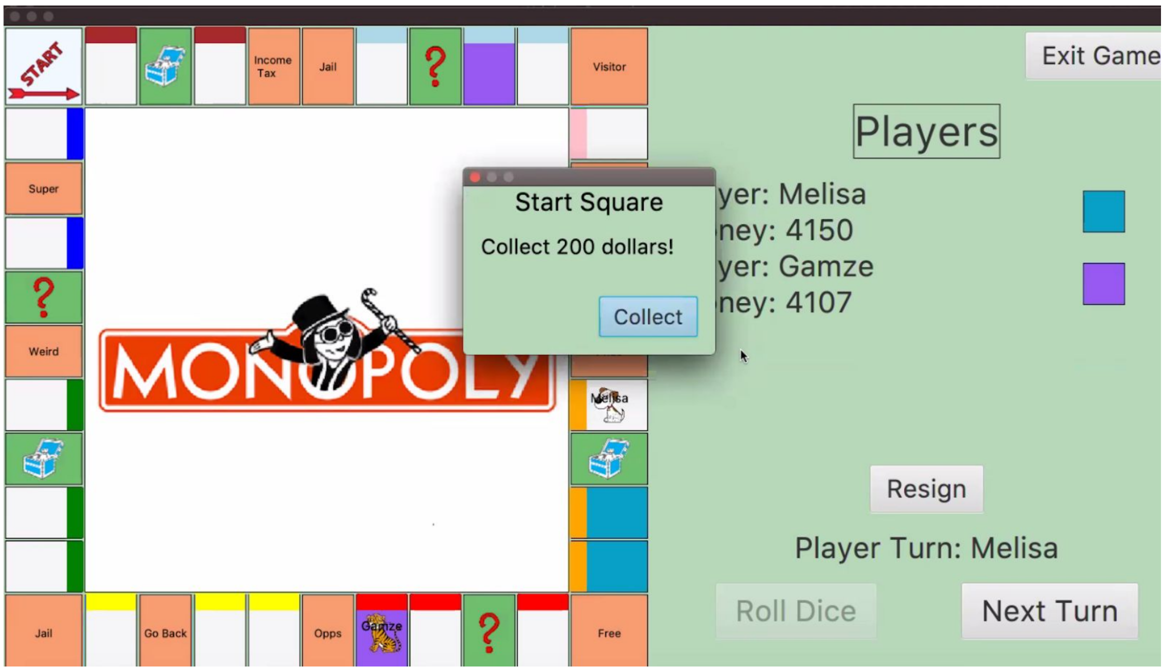
Price: 880
Rent: 704
Rent with color set: 140
Rent one house: 0
Rent two houses: 0
Rent three houses: 0
Rent four houses: 0
Rent hotel: 0
House price: 0
Hotel price: 0
Mortgage: 440
Owner: Melisa

yer: Melisa
ney: 4120
yer: Gamze
ney: 5000

Figure 21. Owned property dialog, seen by a player other than the owner

When a player lands on a property that is owned by another player, this dialog will pop up. It can be seen that the background color of this dialog is the player color of the owner. By pressing the “Pay Rent” button, the player can pay the rent to the owner of this property.

3.4.3.9 Start Square



Start Square

Collect 200 dollars!

yer: Melisa
ney: 4150
yer: Gamze
ney: 4107

Player Turn: Melisa

Figure 22. Start square dialog

Every time a player passes through the Start Square, they are given 200 units of money, in this case it is dollars.

3.4.3.10 Next Turn

After your turn is done, you should press the next turn button in order to game to continue.

3.4.3.11 Resigning

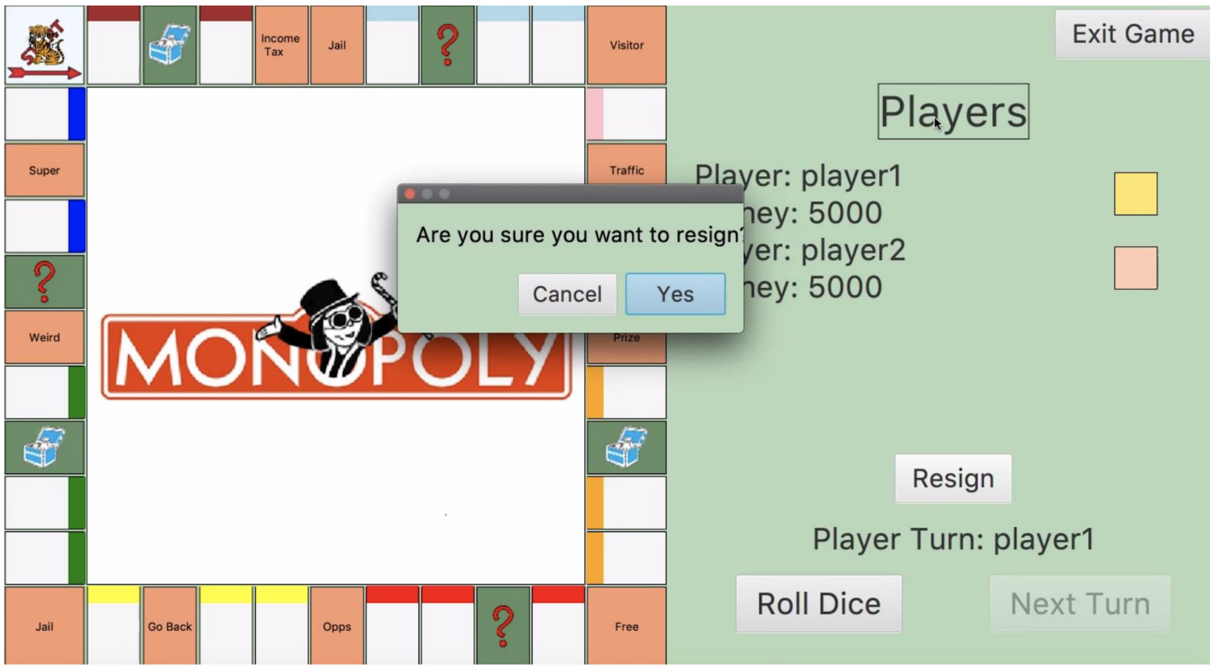
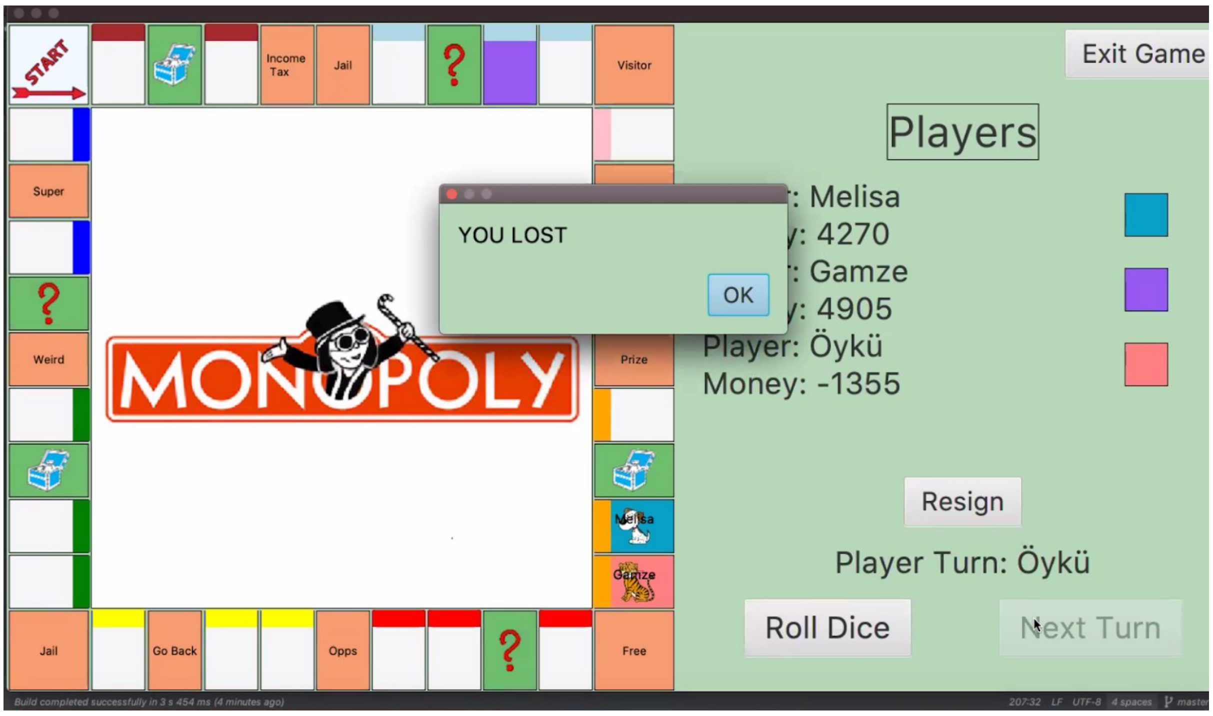


Figure 23. Resign confirmation dialog

When a player wants to resign, he/she can press the “Resign” button of the game screen and this dialog will pop up which is a confirmation dialog that asks the player if he/she is sure. By pressing the “Yes” button, the player will resign from the game.

3.4.3.12 Losing the game



YOU LOST

: Melisa
y: 4270
: Gamze
y: 4905
Player: Öykü
Money: -1355

Player Turn: Öykü

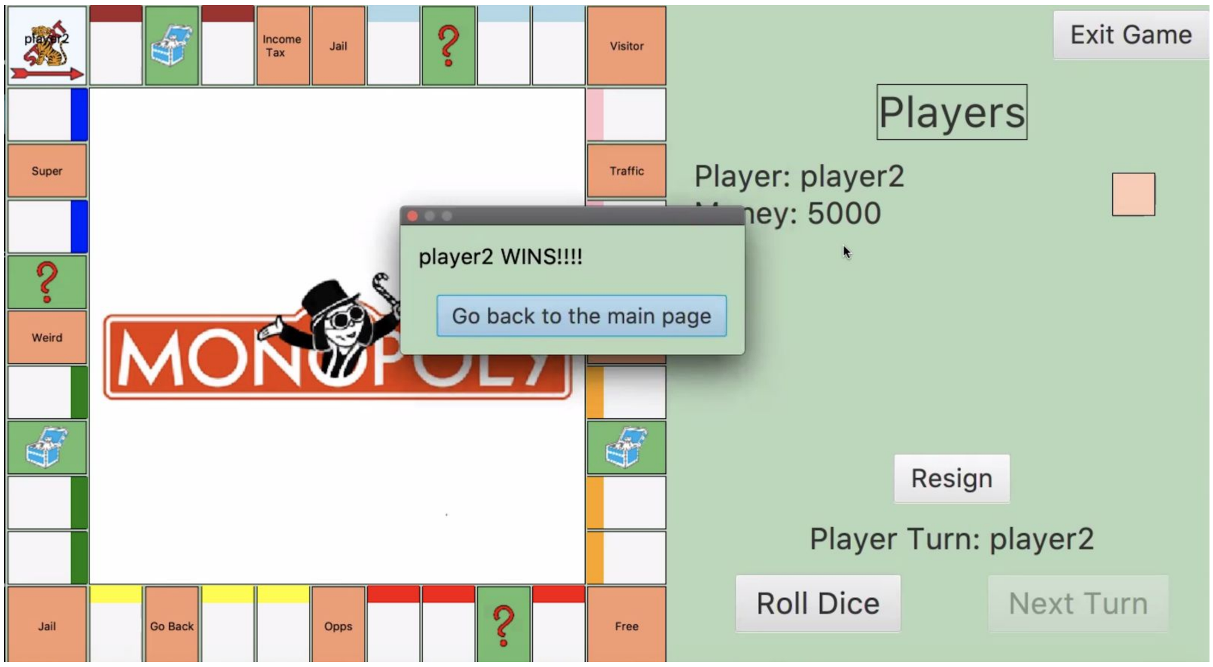
Build completed successfully in 3 s 454 ms (4 minutes ago)

207:32 LF UTF-8 4 spaces master

Figure 24. Losing the game dialog

When a player’s balance is negative at the end of their turn, they are registered as bankrupt, and they lose the game. A dialogue saying “You Lost” to that player appears, and after pressing OK they are officially out of the game.

3.4.3.13 Winning the Game



Player: player2
Money: 5000

player2 WINS!!!!

Player Turn: player2

Figure 25. Winning the game dialog

When every player else is resigned or bankrupted and there is one player remaining, the player is selected as the winner of the game and this dialog which prompts the name of the winner will be shown. By pressing the "Go back to the main page" button, the game will be directed into the main menu.

4 What is Missing?

Having completed the functional requirements we promised, we do not have something missing that would affect the game play. However, perhaps from the visual aspect, we have not added the option to attach a new image (chosen by the user) onto community chests. I.e., we have set a default image for these chests. Plus, we also didn't create a settings screen as we omitted the screen resizing option and the game audio. We also did not implement the in-game menu.

5 Conclusion

While doing the project, we made sure we put the principles of object oriented design that we learned in the class into practice as much as possible. We set functional requirements and design goals, and we mostly met these goals by the end of our implementation. During this project, we learned how to build a system from scratch. Or to be more precise, we learned how to analyze and design a system that is going to be built by a group of people, and then build that system as a group of 5 coders. After all, following the processes of analysis and design, implementation is but building that predetermined system with coding. For all the aforementioned processes, we had to develop our communication skills. This included communicating via the system diagrams we created, setting up group meetings to discuss our project, and work as a team of coders using Git.