Cs319 Term Project
Analysis Report

Section - 03

Group - 3E

Coronapoly

Group Members

1. Osman Buğra Aydın - 21704100
2. Ertuğrul Aktaş - 21802801
3. Muhammed Doğancan Yılmazoğlu - 21801804
4. Oğuzhan Angın - 21501910
5. Mehmet Alperen Yalçın - 21502273

Table Of Contents

1. Introduction

Monopoly is one of the oldest and most popular board game which was released on March 19, 1935. The main purpose of the game is to force opponents into bankruptcy by buying and developing various properties and becoming monopoly. The structure of Monopoly is very suitable to adapt various modifications and there are many different Monopoly variants similar to our game. Coronapoly is the newest modification of the Monopoly designed by Group 3E which features the effects of Covid-19 on the economy and society.

Despite and because of the countless preventive measures taken by the governments, everything about our lives has changed and those changes are what inspired us to make Coronapoly. Even though there are many major aspects of the Corona period that we can adopt and implement in Coronapoly, we will not be altering core mechanics and rules of the original Monopoly game. In Coronapoly we will be adding, removing, and changing rules, mechanics and terms such as social distancing, corona risk level, corona test centers, going into quarantine instead of jail, new area names etc. We believe that with those additions and changes flow of the game will be altered greatly and player experience will be improved by a lot. However, understanding of

the new rules will not be hard and ending conditions of the game will still be the same.

2. Overview

2.1 Gameplay

Coronapoly is a game inspired by a famous board game known as Monopoly with some new features and changes. In the game players aim to establish monopoly in the economy while trying to face the real-world economic problems which are caused by Covid-19. End goal is to establish dominance over the economy and bankrupt other opponents. To achieve this goal, players roll two dice to move around the map to accomplish actions like buy, trade and develop properties. Properties allow players to collect rent from their opponents while players can also earn money from Chance and Community chests, loot boxes and responsible citizen rewards. Loot boxes spawn randomly on the map and stay there until they are taken by a player. If they are taken by a player, loot boxes reward the player and respawn in the next round.

In Coronapoly, players have to face the consequences of the Covid-19 situation. Players may face a penalty if they ever land on slots which other

players stand due to violation of social distancing rule. Each slot has different corona risk levels which affect the individual risk level of the players. Free Test Centers check the player's current infection status. If the players are sick, then they stay in the quarantine zone until they are cured. If they are not sick they are rewarded for being a responsible citizen.

2.2 Players

There is one type of player who will travel along the map. The player of the game has some certain properties that he will use in the game such as buying lands, being infected, having and using money, being able to conduct some commercial actions. Firstly, a player can buy the land that he stands on. Buying means he will own the place and will not give any money when he comes to his own place. Besides, he can construct houses and hotels on his lands if he owns all the lands which are in the same category. Secondly, players may be infected by deadly virus Covid-19. The virus will affect the gameplay of the infected player. Thirdly, players have some amount of money. They can spend and earn it in the game according to their needs and desires. However, a player always must be able to pay his bills or rents. If a player cannot satisfy these, he will be ejected from the game automatically. Finally, there are various different types of commercial actions. Also, a player can mortgage his house to have extra money. Every player must roll the dice before he travels along the map. Players can

trade their lands with each other. The user can assign bot players in the game. Bot players are the ones that are controlled by the computer and decide to sell, buy or any action that a player can do according to some algorithms. The number of bots will be determined by the user as well as their level of playing hardness.

2.3 Initial Economy

This game's economy focuses on being a monopoly through other players in the map. The economy is influenced by capitalism. Players can have some valuable assets in terms of money and lands. Their aim is to earn more money by using their assets. They can use their money to buy the lands that they step on. There are several different lands to stand on. The neighborhoods have rent income when other players came on it. Also, the owner of the neighborhood can build three houses in his neighborhood if he owns all neighborhoods with the same color. Building houses increases the rent prices which results in making owners richer. After building three houses to all neighborhoods with the same color, he can construct a hospital on his place which will have an expensive rent price to the visitors. However, there are some chance and state cards which can make house and hotel owners pay some extra fee. Apart from neighbourhoods. There are transportation stations which can be a good source for income. There are

four stations on the game which are bus, train, ship and plane. The purchase cost and income level increases respectively. If a player owns a station, every visitor should pay a certain amount of money which will multiply by the number of stations that you have. However, there is different characteristic feature on the stations. A player who stands on an unsold station must buy the station. Otherwise, there will be a public auction which will result on sale of the station. The starting money of auction starts from a certain amount. The station belongs to the player who offers the highest price for the station. Another type of land is state departments which regulate water and electricity. Any player who stands on the place can buy it. The visitors of the place will pay a certain amount of money which is calculated by the dice that they have rolled. Therefore, the cost of visiting these state departments might be cheap and expensive at the same time. On the other hand, mortgage can be a choice for getting extra money. By mortgaging a land, the player will have some money from the state. It is a good feature for the economy of the game because it prevents players from going bankrupt and enables players to have extra cash to buy more lands or bills, etc. However, it has a bad effect for the player's economy since mortgage means not being able to get rent income from your own lands. There are two types of cards which are state and chance. These cards have some special events or incidents that affect players in both good or bad ways. For instance, players may inherit some money from their parents or they may pay some traffic tickets. These cards are formed to have an enjoyable experience for players and to enhance

economic activities of the game. At the beginning of the game, players have some beginning money which will enable them to buy some properties and to start the competition but they don't have any lands just starting money. After some certain number of rounds, players can start to trade with each other. Trade can happen with valuable assets which are lands and money.

2.4 Map

Map of the Coronapoly consists of slots which add up to form the actual map. Slots consist of properties, Quarantine Zones and Free Test Centers. Map is a square of slots. Each slot can hold a property which player can buy or instructions for the player to complete. Chance and Community cards are placed in the diagonal small empty squares of the map.

2.5 Pandemy

The outbreaking virus pandemic has a huge impact on the game. Every player has the chance of being sick due to Covid-19. Players might be infected with the virus in three different ways such as lack of social distance, virus frequency level of a city and chance cards in the table. First possibility is getting a virus

from other players who are in the same city with them. If there are more than one player in a city and one player is infected by Covid-19, healthy players might be infected. This chance will be announced at the end of the round. Second possibility is being affected by the city that the player stands on. Every city has a different possibility of being infected by Covid-19. Whenever a player steps on a city, he should roll the dice again and try to exceed a predetermined number with the summation of the dices. The number will be calculated with the corona level of the city. If the corona risk is higher, the summation of the player's dice should be higher. Final possibility is getting a virus from the chance or state cards. There are some cards that will directly infect the player. On the other hand, a player who is infected can travel along the map. In addition, he can spread the virus to other players and increases the corona level of his current city. There is just one solution to prevent the spread of the virus. There is a place on the map that makes corona tests. There is a little possibility that the corona test may result wrong but most probably it will diagnose an infected player. If a player is diagnosed with corona, he will go into the quarantine which means he won't be able to play the game for some rounds. Besides, there will be a hospital fee if a player is infected. Therefore, the aim of a player who wants to dominate the game should be the prevention from the Covid-19.

2.6 Rules

· Players’ rolls die before starting the game. Who lands the higher value goes first in clockwise direction.

· Player who lands or passes the Go slot gets 200 TL from the bank

· Players who land on Income Tax or Luxury Tax slots 100 TL pay to the bank.

· Development can be only done if the player holds all the properties of the same color.

· A second house cannot be built until all properties of the same color have a house on them.

· Players draw the top card from the decks if they land on Chance or Community chest slots.

· Player follows the instruction stated on the drawn card.

· Players’ infection level changes according to slots they pass through.

· Players do not know their infection level until they go through Free Test Center slots.

· If players infection level is high enough, they go to the quarantine zone and stay there until they are cured.

· If a player lands on a slot and if it is not owned then they can buy that property by paying its price. They can refuse to buy the property.

· If a player lands on a slot and if it is owned then they must pay the rent to the owner of the property.

· Players can develop their properties when it is their turn by buying houses from the bank.

· Players collect double rent if they own the entire group.

· If a player cannot pay what he owes then he is bankrupt and out of the game.

· If a player owes debt to the bank, they should give all assets to the bank. If a player owns debt to another player, they should give all assets to that player.

· Players cannot choose to go bankrupt. They should do whatever to pay the debts until they are bankrupt.

3. Requirements

3.1 Functional Requirements

3.1.1 Main Menu

This is the screen that will come up first when the user launches the game. In this menu, the user will have a variety of options to choose from which are explained in the following sections. Other than the listed options, the user will be able to quit the game by pressing the `EXIT` button placed on the bottom of the screen. There will be background music for the first time that the main menu initialized.

3.1.2 Play Game

The `Play Game` option on the main menu will take the user to another screen where the user will be able to choose from 2 options: Singleplayer or Multiplayer. The only difference between the two options is that singleplayer game offers a game with A.I. (Artificial Intelligence) players while multiplayer game offers a game with other real players.

3.1.2.1 Singleplayer

The singleplayer game can be accessed from the main menu. After clicking the singleplayer option, the user will select a pawn to play with and then the game will start with a specified number of A.I. players. The user will only control his moves and watch when it is not his/her turn.

The singleplayer game will end when the user is bankrupt or all of the other players are bankrupt. The user will be able to roll dice, choose his actions during his turn such as buying a property or refusing to buy it. The user can also manage his properties in terms of the additional buildings which increase the rent of that property.

3.1.3.2 Multiplayer

The multiplayer game can be accessed from the main menu as well. After the user clicks the multiplayer option, there will be options such as adding “real” players and choosing pawns for them. What is meant by “real” is that the multiplayer game lets more than one user to play the game at the same time. The users will have to share the same screen when playing and play when it is their turn. If any A.I. is added to the game, then the A.I.’s turn will be done automatically without wasting any time for the actual players.

The multiplayer game will end when all of the actual players are bankrupt or one of them bankrupts all of the remaining players. The options for the user is the same for a multiplayer game but it is multiplied since each actual user will have to go through the same option cycle such as buying, managing property etc.

3.1.3 How to Play

The How to Play option on the main menu will take the user to a screen where all of the Coronapoly rules are listed. In this part, the user can learn all of the details of Coronapoly as well as the common rules with the original Monopoly game. The user can learn about:

- Game Rules
- Map Contents and their functions
- How to win
- Frequently Asked Questions

3.1.4 Settings

The settings option in the main menu is for setting the controls for the user. In this menu, the user can adjust the volume of the game, mute sound notifications or change the window size.

3.1.5 Credits

The credits option will take the user to another screen where the developer information is found. The page will contain the contributions of each of the developers to the game as well as their contact information to receive feedback from the users.

3.2 Non Functional Requirements

3.2.1 Usability

Monopoly is one of the most common board games so almost everyone which is interested in board games knows how to play. However, our game called Coronapoly has some new features such as neighborhoods have corona

risk and players can be COVID-19 positive by visiting these neighborhoods. Therefore, we are explaining these new features in the how to play section of the game so users can understand the basics of the game and play easily. The similarity of our main menu with many other games helps users to familiarize the menu faster. Our game window is organized very well so that players can distinguish the board, their properties, buttons etc. and make moves easily. Also, our game board is very similar to the original Monopoly game board and it makes users comfortable when they play.

3.2.2 Reliability

As a rule based game, in monopoly there are a lot of pitfalls in terms of complicated behavior of actors. There are so many scenarios to face and moves to perform by the player such as manipulating the game by selling properties for very low price to another player and our game has restrictions to prevent such actions. With the auto save mechanism, we are keeping the data loss in minimal levels. The system saves the data of the game every five turns so players can reset the game in case of an event of a failure or exception to continue the game with the latest saved state.

3.2.3 Performance

The game should run on more than 30 frames per second to make sure that users have a smooth game experience. The responsiveness of the system

will not be more than average human response time in order to serve a latency free experience. For instance, when a player presses the roll dice button, the dice will be rolled immediately. We will establish such experience with high performing code by writing proper algorithms.

3.2.4 Supportability

The game might change with the new consequences of corona in our lives such as whether the vaccine is successful or not. Therefore, to be able to support those changes we will use the strength of a good packing. With low size and easy installation, the game might be updated by manually downloading the new installer or auto update feature might be implemented in order to make the updating process a lot easier when an internet connection is available.

3.2.5 Constraints

In terms of implementation we are restricted to not use pre-made game engines such as Unity and Unreal Engine. And we are required to use an object

oriented programming language. For this project we will be using Java programming language with JavaFX gui library.

3.2.6 Packaging

Clean product packaging and distribution is one of the most crucial parts of the process. The package that will be distributed will be free of redundant files, compressed to minimum size and easily installable. Our final product will be distributed as an installer which will consist of just an installer file, including a basic user interface and expected to be handy for the users that lack sufficient computer literacy. An installer will also be beneficial in making a proper installation and placing files in the right directories.

3.2.7 Legal

As a monopoly variant our product could easily violate copyright terms of the original Monopoly game. So we will be making every user interface element from scratch such as the game board and changing certain rules to make sure that the game is distinctive enough to not violate the mentioned terms.

3.2.7 Security

As coronapoly is an offline game that will be played on one computer, there is not much need for security. However, there might be a situation to protect all the properties and money from other human users. When a player is away from the keyboard in a multiplayer game which is being played in front of just one computer, other human users can trade his or her wealth to themselves. To prevent that there will be password protection in every commercial action.

4. System Models

4.1 Use Case Diagram

4.1.1 Use Case Diagram For Gameplay

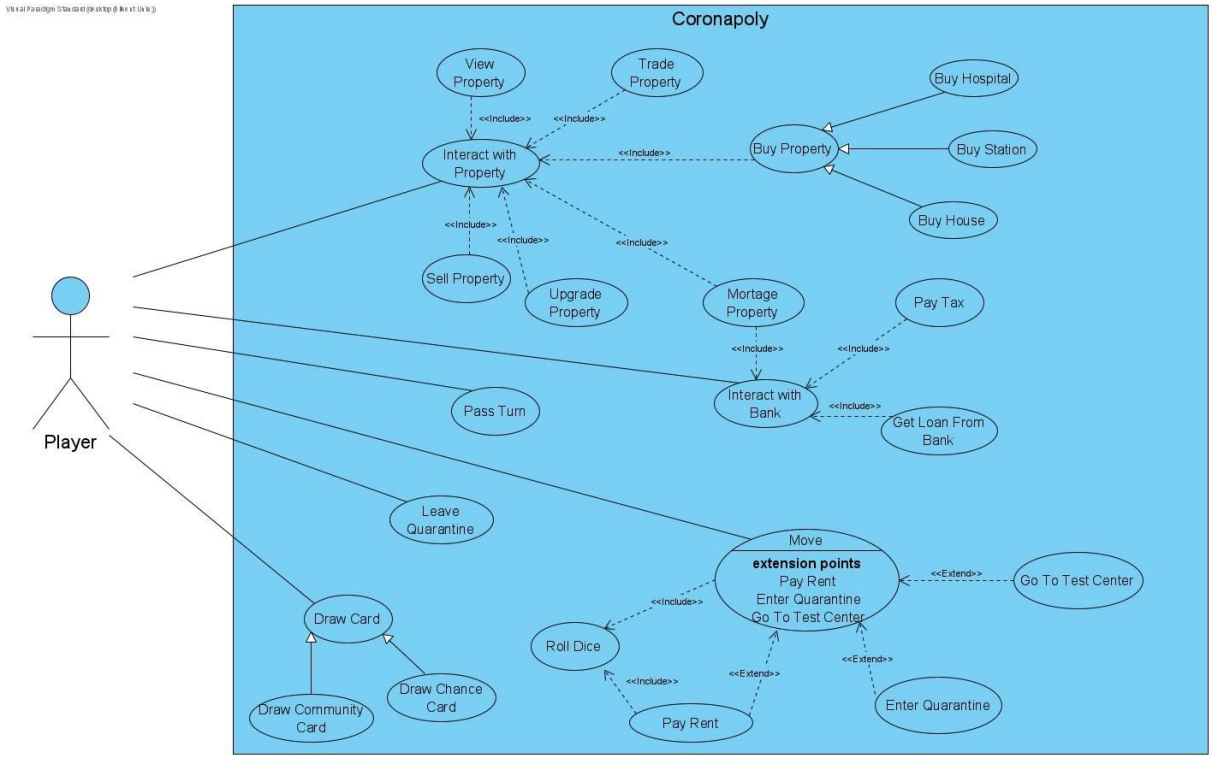

Diagram 4.1.1: Use Case Diagram of gameplay of the Coronapoly game

Use Case 1

Use Case: Interact with Property

Actor: Player

Stakeholders and Interests:

· Player can buy a new property.

a. Player can buy a new house.
b. Player can buy a new hospital.
c. Player can buy a new station.

· Player can sell the property.

· Player can upgrade the property.

· Player can view the properties of the property.

· Player can trade the property with another player.

· Player can mortgage the property.

Entry Condition:

· It is the player's turn and the player stands on a property .

Exit Condition:

· Player quits game

· Player bankrupts

· Player wins the game.

· Player passes its turn.

Main Event Flow:

· Player roll dice.

· System moves the player according to the dice roll.

· Player lands on a property.

· Player interacts with that landed property.

· System registers the applied interaction to the game.

· Player passes its to the next player.

Alternative Event Flow:

a. Player does not land on a property:

· Player does not land on a property.

· Player play its turn

· Player passes its turn.

b. Player lands on a property:

· Player skips its turn without interacting with the property.

· System passes turn to the next player.

c. Player quits game:

· Player presses the pause game button.

· System shows pause menu.

· Player presses the quit game button.

· System returns to the main menu.

Use Case 2

Use Case: Interact with Bank

Actor: Player

Stakeholders and Interests:

· Player can pay tax to the bank

· Player can mortgage one of its properties.

Entry Condition:

· It is the player's turn and the player moved after rolling the dice .

Exit Condition:

· Player quits game

· Player bankrupts

· Player wins the game.

· Player passes its turn.

Main Event Flow:

· Player roll dice.

· System moves the player according to the dice roll.

· Player lands somewhere on the board.

· Player accomplishes the tasks of the tile if there are any.

· Player interacts with the bank.

· System registers the applied interaction to the game.

· Player passes its to the next player.

Alternative Event Flow:

a. Player lands on a tax title and does not required own the money:

· Player land on a income tax title.

· Players current amount of currency is checked by the system.

· Player fails to pay the tax with money.

· Player sell or mortgage their properties to get some money.

· Player interacts with the bank to pay the tax.

b. Player skips the turn:

· Player skips its turn without interacting with the bank.

· System passes turn to the next player.

c. Player quits game:

· Player presses the pause game button.

· System shows pause menu.

· Player presses the quit game button.

· System returns to the main menu.

Use Case 3

Use Case: Move

Actor: Player

Stakeholders and Interests:

- Player can change their tile on the board.
- Player can go to the quarantine title directly.
- Player can go to the free test center title directly.

Entry Condition:

- Player needs to roll the dice before moving .
- Player needs to pay their previous tax before moving.

Exit Condition:

- Player moves to a different tile.
- Player quits game
- Player bankrupts
- Player wins the game.
- Player passes its turn.

Main Event Flow:

- Player roll dice.
- System moves the player according to the dice roll.

· Player lands somewhere on the board.

· Player accomplishes the tasks of the tile if there are any.

· Player interacts with the bank or property.

· System registers the applied interactions to the game.

· Player passes its to the next player.

Alternative Event Flow:

a. Player needs to go to the quarantine:

· Player land on a free test center.

· System checks player if the player is sick.

· Player is sick.

· System moves player to the quarantine tile.

· System passes turn to the next player.

b. Player needs to go to the free test center:

· Player land on a tile that player needs to draw a card.

· Player draw card that requires player to move to the free test center.

· System moves player to the free test center tile.

c. Player quits game:

- Player presses the pause game button.
- System shows pause menu.
- Player presses the quit game button.
- System returns to the main menu.

Use Case 4

Use Case: Draw Card

Actor: Player

Stakeholders and Interests:

- Player can draw chance card.
- Player can draw community chest card.

Entry Condition:

- Player needs to land on a chance tile.
- Player needs to land on a community chest tile.

Exit Condition:

- Player completes the instructions of the drawn card.

- Player quits game
- Player bankrupts
- Player wins the game.
- Player passes its turn.

Main Event Flow:

- Player roll dice.
- System moves the player according to the dice roll.
- Player lands on a chance or community chest tile.
- Player draws card from the corresponding stack of the landed tile.
- Player completes the instructions of the drawn card.
- System registers the applied interactions to the game.
- Player passes their to the next player.

Alternative Event Flow:

a. Player is not able to complete a instruction of the card:

- Players current amount of currency is checked by the system.

· Player fails to pay the required amount of the card.

· Player sells or mortgages their properties to get some money.

· Player pays the required amount of the card.

b. Player quits game:

· Player presses the pause game button.

· System shows pause menu.

· Player presses the quit game button.

· System returns to the main menu.

Use Case 5

Use Case: Leave Quarantine

Actor: Player

Stakeholders and Interests:

· Player can leave the quarantine tile.

Entry Condition:

· Player needs to be cured.

· Player needs to have a leave quarantine card.

Exit Condition:

· Player moves from the quarantine tile to a different tile.

· Player quits game

· Player bankrupts

· Player wins the game.

Main Event Flow:

a. Player has the leave quarantine card:

· Player is in the quarantine tile.

· Player plays the one of the owned leave quarantine card.

· Player plays their turn.

b. Player does not has the leave quarantine card:

· Player is in the quarantine tile.

· System skips the turn of the player until the player is cured.

· When player is cured player plays their turn

Alternative Event Flow:

a. Player quits game:

· Player presses the pause game button.

· System shows pause menu.

· Player presses the quit game button.

· System returns to the main menu.

4.1.2 Use Case Diagram For Menus

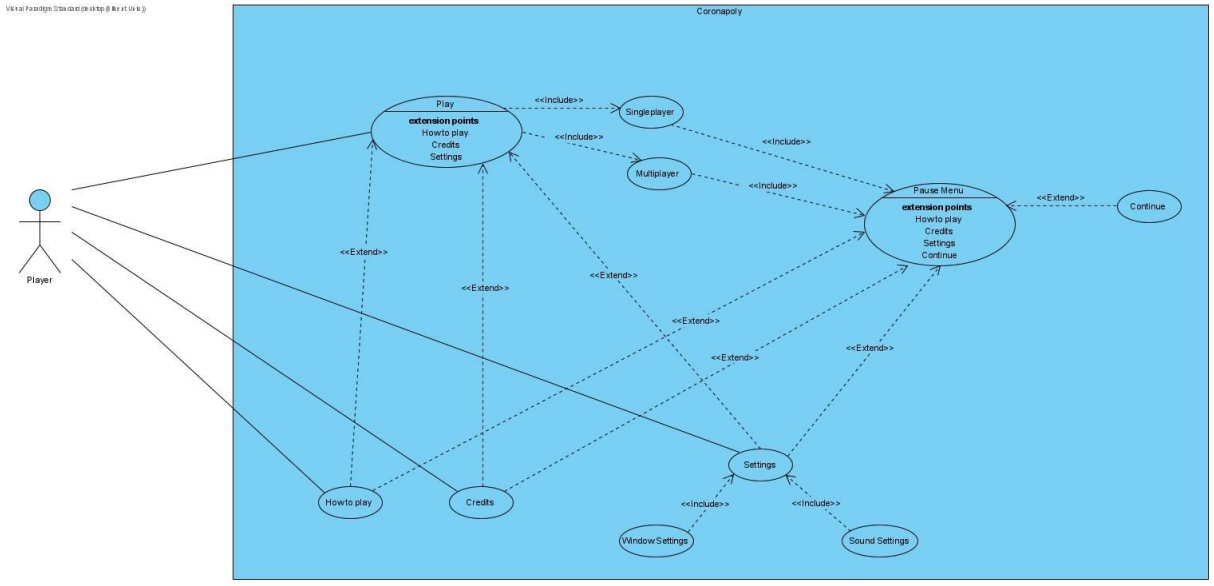

Diagram 4.1.2: Use Case Diagram of Menus of the Coronapoly game

Use Case 1

Use Case: Play

Actor: Player

Stakeholders and Interests:

· Player can start a new Coronapoly game.

· System initializes and starts the game.

Entry Condition:

· Player is on the start menu and clicks the play button.

Exit Condition:

· Player quits game

· Player bankrupts

· Player wins the game.

Main Event Flow:

· Player chooses game mode.

· System initializes and starts the game.

· Player and other participants roll dice.

· System chooses which player goes first.

· Players play their turns.

· System updates properties and currencies of the player after their turn ends.

· System destroys bankrupt players.

· System ends the game if one player is left.

Alternative Event Flow:

· Player presses the pause game button.

· System shows pause menu.

· Player presses the quit game button.

· System returns to the main menu.

Use Case 2

Use Case: Pause Menu

Actor: Player

Stakeholders and Interests:

· System stops the game and shows a pause menu.

Entry Condition:

· Player presses the Pause menu button from the game interface.

· Player presses the “esc” key from the keyboard.

Exit Condition:

· Player presses the “esc” key from the keyboard.

· Player presses the exit button frım game menu.

Main Event Flow:

· Player presses the “esc” key or pause menu button.

· System shows pause menu.

Alternative Event Flow:

· Player presses the “Continue” button or “esc” key.

· System stops displaying the pause menu.

Use Case 3

Use Case: Settings

Actor: Player

Stakeholders and Interests:

· Player can choose the window size of the game.

· Player can choose the sound level of the game.

Entry Condition:

- Player clicks the “Settings” button from the main menu.
- Player clicks the “Settings” button from the pause menu.

Exit Condition:

- Player clicks the “Return Menu” button from the main menu.
- Player clicks the “Return Menu” button from the pause menu.
- Player clicks “esc” from the keyboard.

Main Event Flow:

- Player clicks the “Settings” button from the main or pause menu.
- System shows settings menu.
- Player slides a slider to change the sound level of the game.
- System changes the sound level of the game.
- Player chooses the screen size of the game from the dropbox menu.
- System resizes the game.

Alternative Event Flow:

- Player clicks the “Return Menu” button or “esc” key.

· System returns to the previous menu.

Use Case 4

Use Case: How To Play

Actor: Player

Stakeholders and Interests:

· System shows How To Play menu.

Entry Condition:

· Player clicks the “How To Play” button from the main menu.

· Player clicks the “How To Play” button from the pause menu.

Exit Condition:

· Player clicks the “Return Menu” button from the main menu.

· Player clicks the “Return Menu” button from the pause menu.

· Player clicks “esc” from the keyboard.

Main Event Flow:

· Player clicks the “How To Play” button from the main or pause menu.

· System shows How To Play menu.

Alternative Event Flow:

· Player clicks the “Return Menu” button or “esc” key.

· System returns to the previous menu.

Use Case 5

Use Case: Credits

Actor: Player

Stakeholders and Interests:

· System shows credits menu.

Entry Condition:

· Player clicks the “Credits” button from the main menu.

Exit Condition:

- Player clicks the “Return Menu” button from the main menu.
- Player clicks “esc” from the keyboard.

Main Event Flow:

- Player clicks the “Credits” button from the main menu.
- System shows How To Play menu.

Alternative Event Flow:

- Player clicks the “Return Menu” button or “esc” key.
- System returns to the main menu.

4.2 Activity Diagram

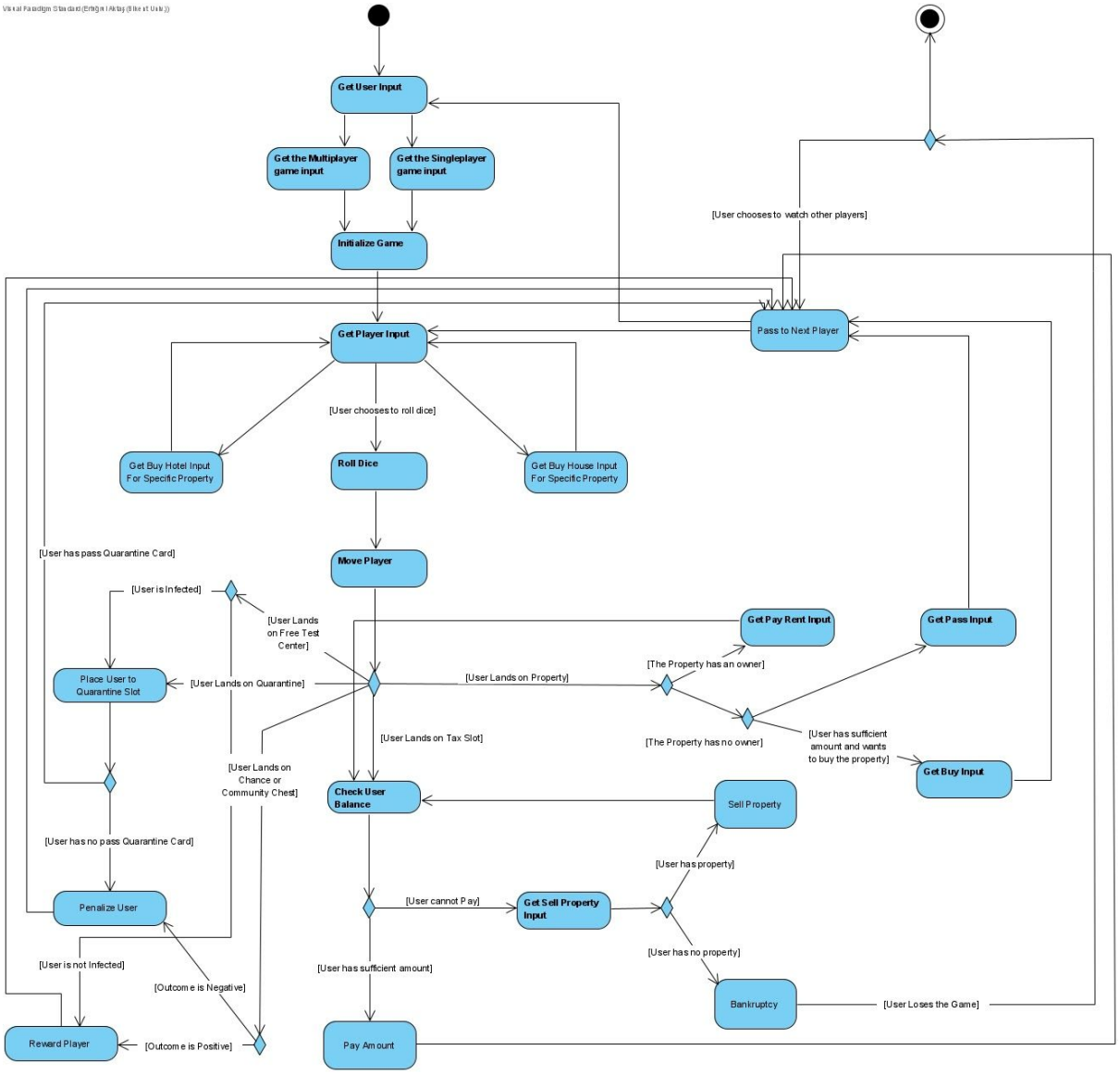

Diagram 4.2.1: Activity Diagram of gameplay of the Coronapoly game

Above lies the activity diagram for the game Coronapoly. This diagram shows the working principle of the game in terms of interactions with the system.

The game starts with the user's choice of singleplayer or multiplayer. However, this choice does not affect the working principle of the system since the game is turn based and players play on the same computer.

The first scenario of the activity diagram occurs when the player's turn starts. The player can buy or sell the houses and hotel on his/her property before rolling a dice if the player has such properties but this option is not mandatory. The mandatory thing here is the act of rolling dice since after managing properties, the game loops back to the same input activity and only proceeds when the player rolls his/her dice.

The rest of the scenarios occur after the dice roll is completed. According to the roll number, the player must land on a spot. The spot which the player will land could be one of five different options that each have unique responses:

Landing on property: If a player lands on a property, there are two options. The property could be owned by some other player that forces the current player to pay rent. Paying rent redirects the diagram into checking the user's balance. If the player has enough money to pay the rent, his turn ends and the specified amount is taken from the player's balance. However, if the player does not have the cash to pay rent, then the system checks for available property to sell. If the player has property to sell, then the game continues but if not, then the player loses the game as it would mean bankruptcy for the player. In multiplayer, the user can still stay in the game and watch other players or choose to quit the game. If the property which the player landed on is not owned by any other player, then the player could buy the property if he/she has the required cash or they could just skip their turn.

Landing on Tax Slot: If a player lands on a tax slot, the player's balance is checked once more to ensure that he/she is capable of paying the tax. Otherwise, the player goes through the procedure which was explained above once more.

Landing on Go to Quarantine: If a player lands on go to the quarantine slot, he/she must be moved to the quarantine slot. After that, it is checked whether the player has a quarantine pass card which could be obtained from community chests and chance cards. If the player has one, then he/she continues to play but if not, then the player is penalized with skipping extra turns.

Landing on Free Test Center: If a player lands on a free test center, then according to the player's infection level, the player either gets penalized for having an infection and goes to quarantine or gets rewarded for keeping a good infection level.

Landing on Community Chest or Chance: If a player lands on a community chest or chance, the player will get a random reward or penalty independent of anything. The outcome is generated randomly unlike the free test center where the infection level is calculated based on the gameplay.

4.3 Object and Class Diagram

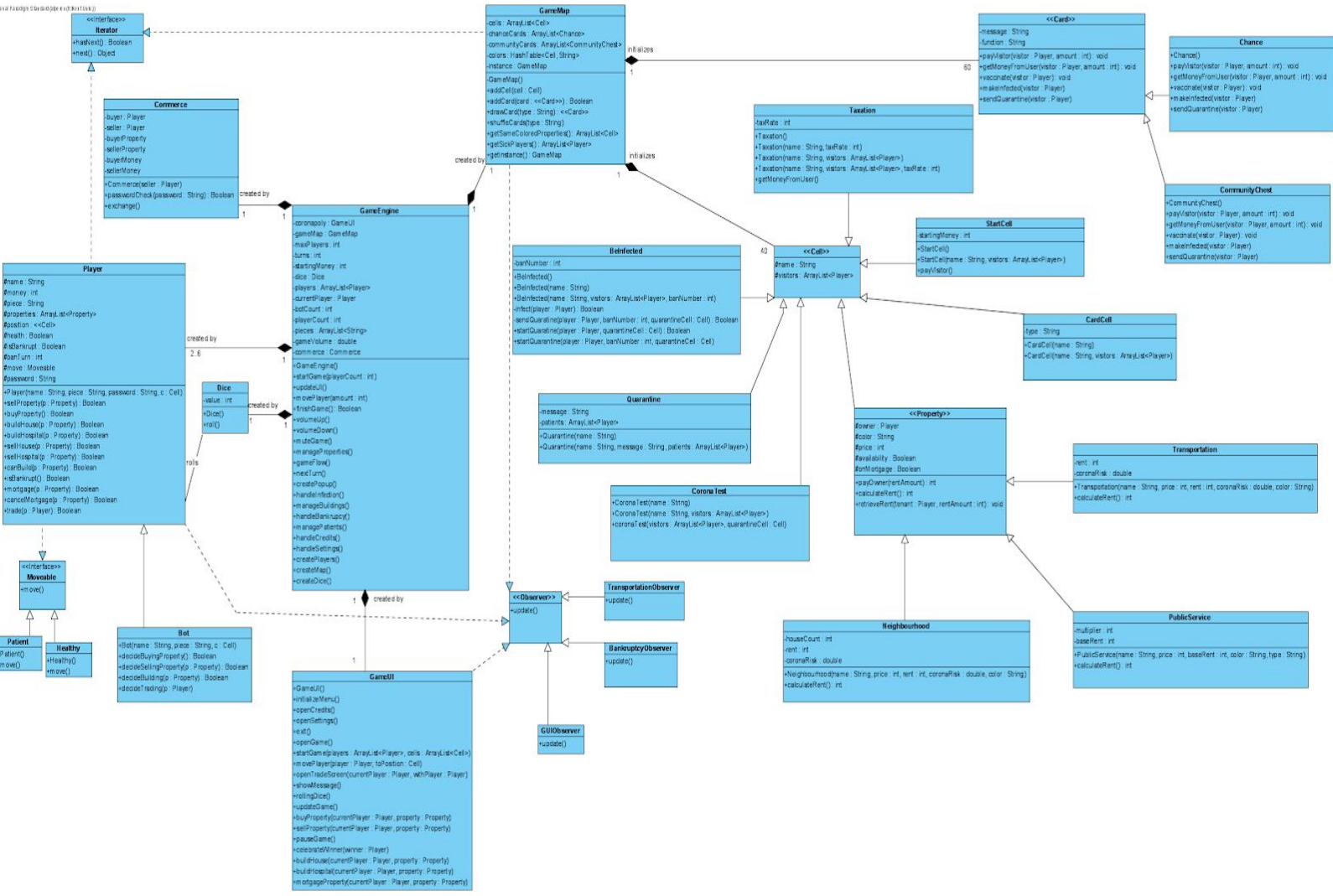

Diagram 4.3.1: Object and Class Diagram the Coronapoly game

4.4 Sequence Diagrams

4.4.1 Main Menu

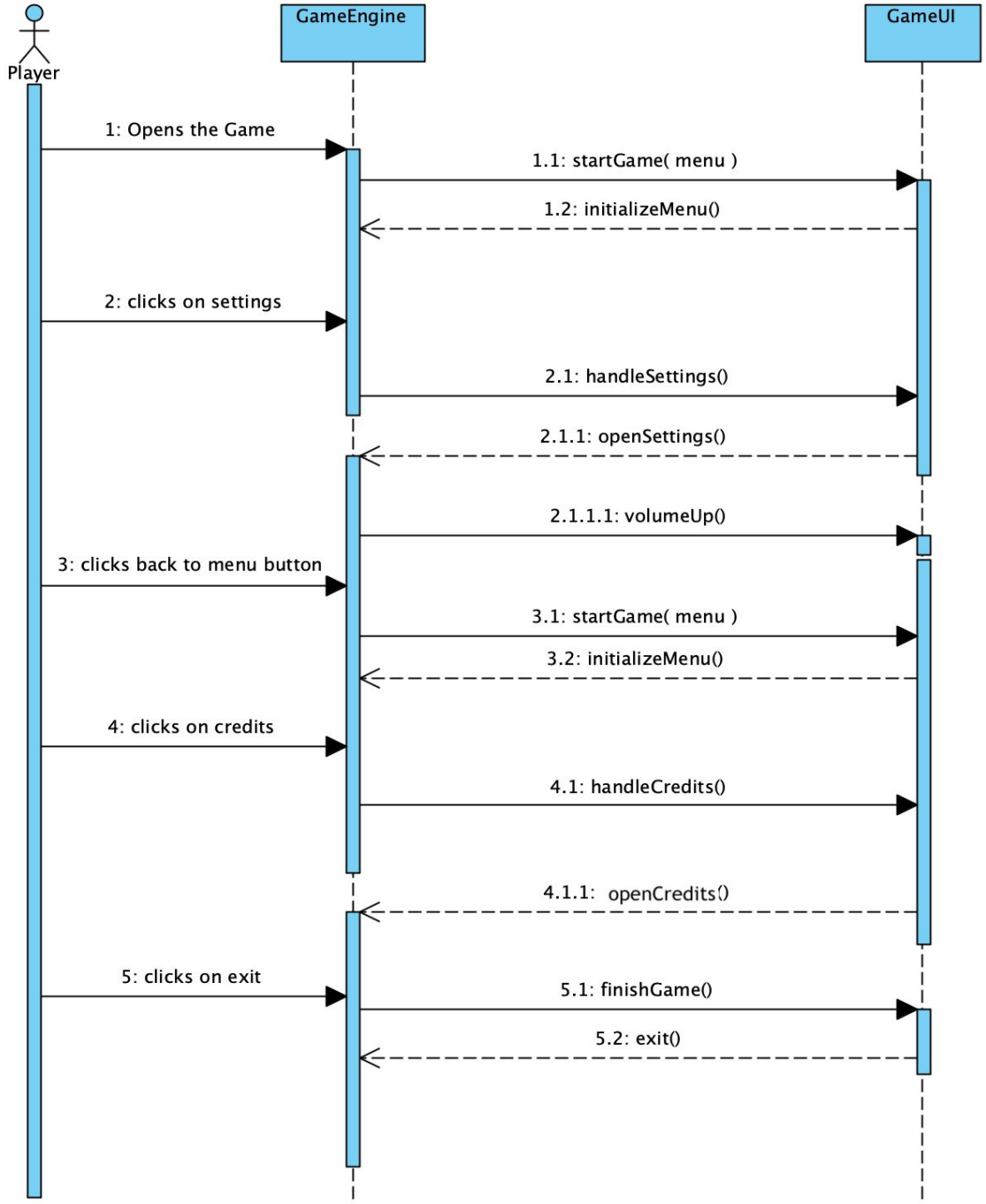

Diagram 4.4.1: Main Menu Sequence Diagram the Coronapoly game

This sequence diagram covers the actions in the main menu. Player opens the game and interactions between the player and our program start with the class named GameEngine. GameEngine sends a signal to GameUI class which handles all user interfaces in the game. GameMenu starts the game with the help of startGame() method and indicates in parameters that the user interface that should be opened is the main menu. GameUI responds to the request with a particular method called initializeMenu to obtain visible parts of the menu. When the player clicks on the settings section, GameEngine triggers GameUI with handleSettings method which results in openSetting() method from GameUI. As a result, the settings screen will be opened. There is a go back button to return the main menu in every window of the menu except the main menu. Back button sends a signal to GameEngine to arrange an user interface with the help of startGame(menu) method. Then, GameUI returns the main menu interface with the initializeMenu. To open credits, GameEngine sends openCredits() method to GameUI which results in credits' interface with the help of openCredits method. If the player wants to close the program, GameEngine closes the program with a method called finishGame and GameUI closes the windows.

4.4.2 Play Game

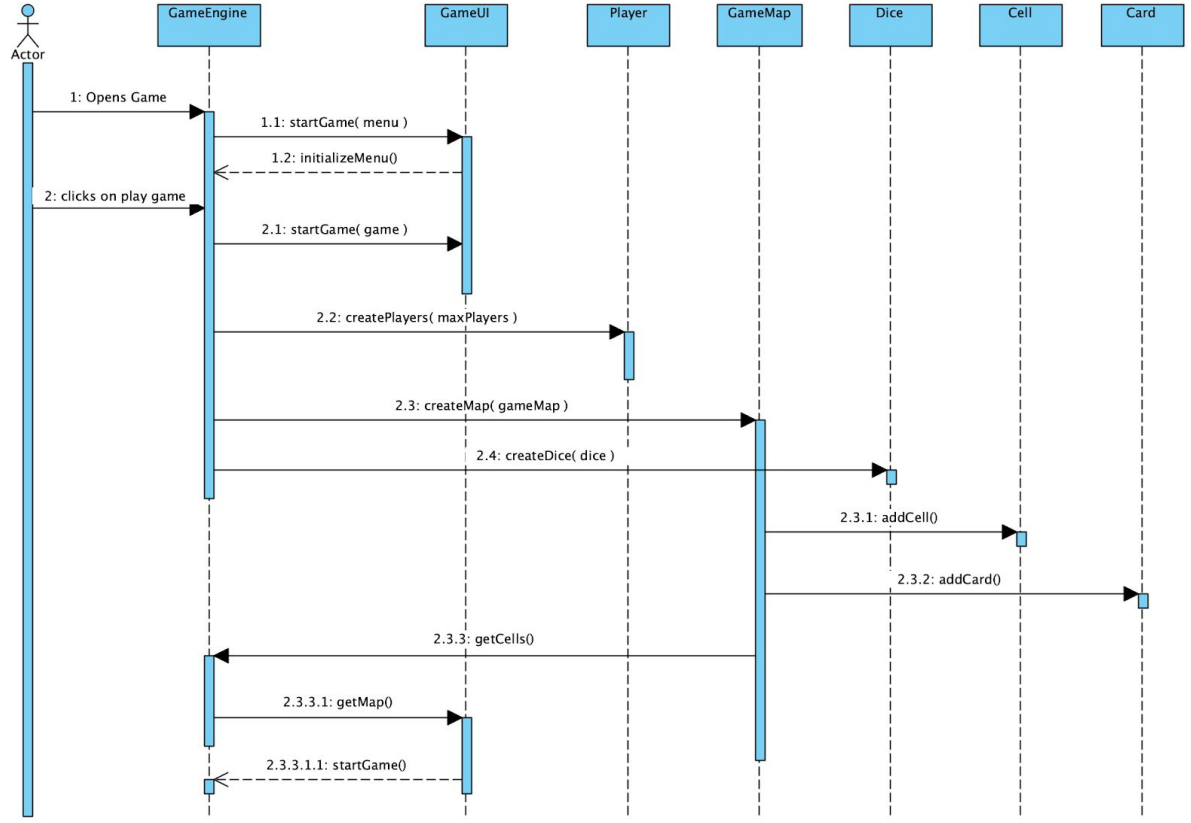

Diagram 4.4.2: Play Game Sequence Diagram the Coronapoly game

This diagram covers the interactions when a player starts to play a game. Player opens the application and the main menu is created by GameUI. In the main menu, the player clicks play game and GameEngine starts to create the game by using other classes. First of all, players in the game are created with the determined amount of players by the user. Then, GameEngine creates the GameMap that players will play on. Dice is created by GameEngine to be used in the game. GameMap consists of many cells which players will travel on. Also, GameMap creates a lot of cards which include interesting features to the

game. After GameMap obtains all the data for the game, it returns the data to GameEngine. GameEngine controls the data and sends them to GameUI that will visualize the game by returning an interface according to the data with startGame method. Finally, users can play the game.

4.4.3 Buy a Property

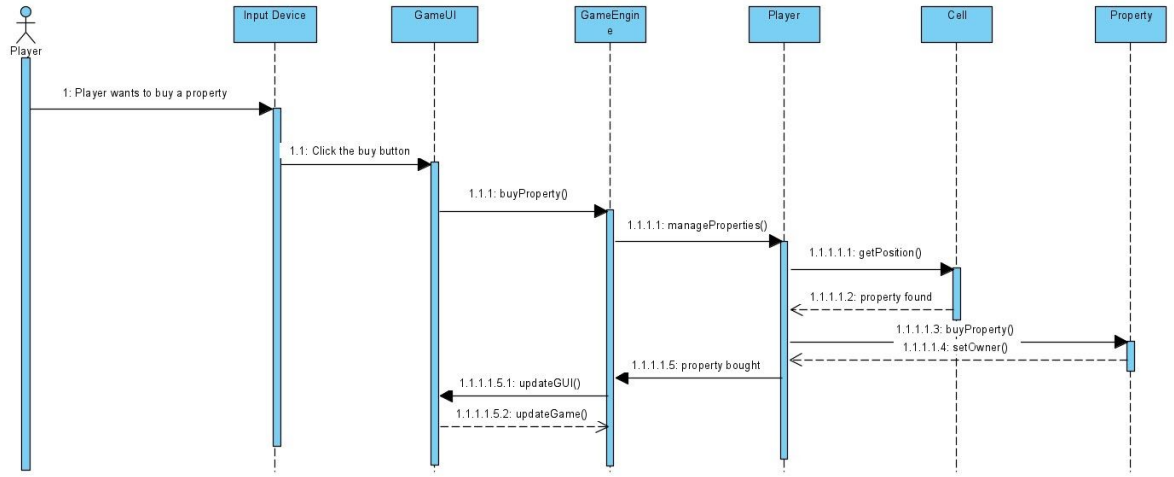

Diagram 4.4.3: Buy Property Sequence Diagram the Coronapoly game

Player wants to buy a property and clicks the buy button with a mouse. The command of mouse calls the buyProperty method from GameUI and it activates manageProperties method from GameEngine class. Player object gets the position of the cell with getPosition and it returns a cell. After returning the player object, call the buyProperty method to buy the property and property class sets the owner. This operation makes the player have the property. Finally,

GameEngine calls updateGUI method to make the operations visible to users and GameUI class calls updateGame to transfer the difference to the master class of the game.

4.4.4 Build a House

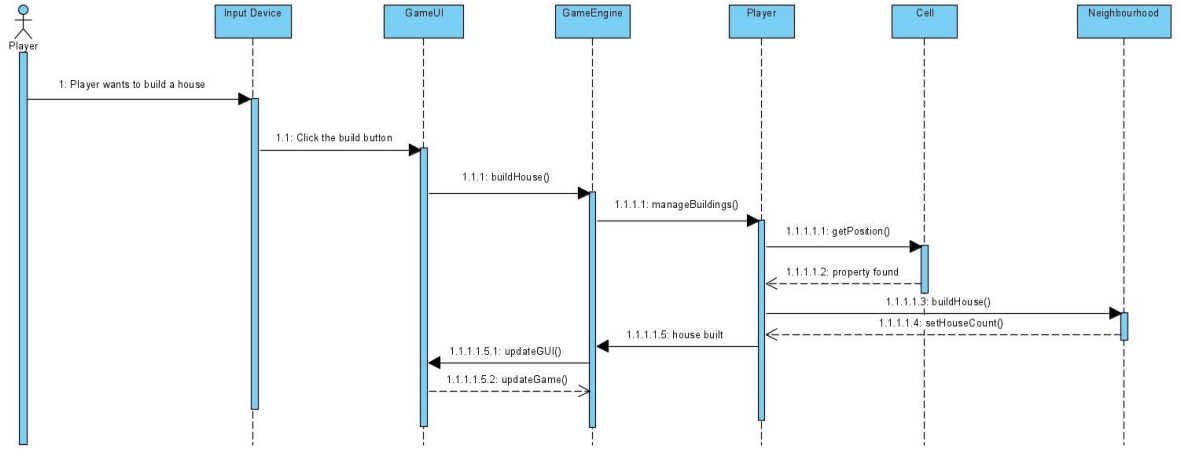

Diagram 4.4.4: Build House Sequence Diagram the Coronapoly game

Player wants to build a house and clicks the build button with a mouse. The mouse command of mouse calls the buildHouse method from GameUI and it activates manageBuildings method from GameEngine class. Player object gets the position of the cell with getPosition and it returns a cell. After returning the player object call buildHouse method and property class set the house count of this property. If the house count of a property is four, it is called hospital instead of a house. This operation makes the player have the house or hospital in a property that the user owned. Finally, GameEngine calls updateGUI method to

make the operations visible to users and GameUI class calls updateGame to transfer the difference to the master class of the game.

4.4.5 Mortgage a Property

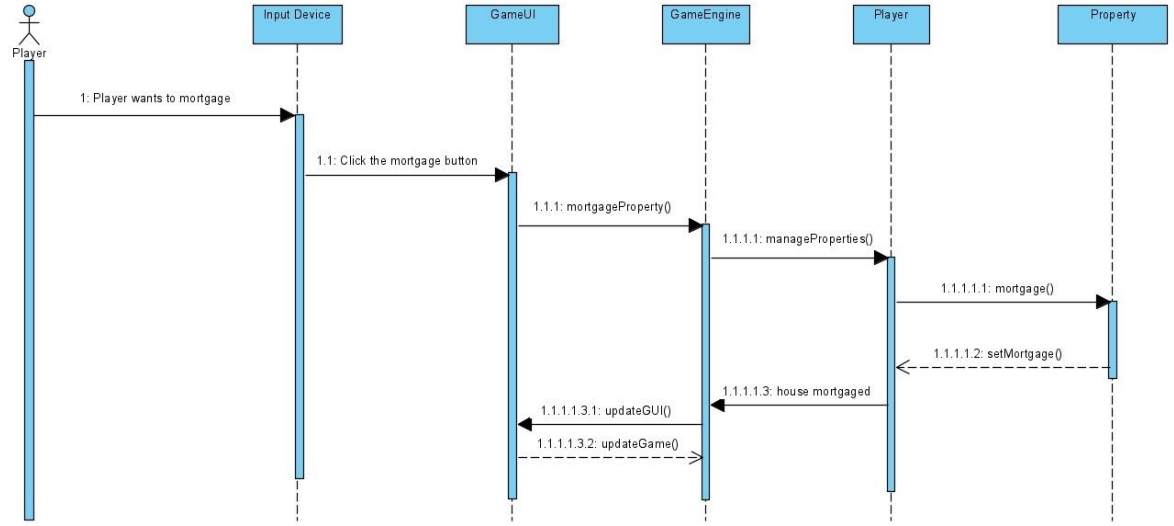

Diagram 4.4.5: Mortgage Property Sequence Diagram the Coronapoly game

Player wants to mortgage a property and clicks the mortgage button with a mouse. The command of mouse calls the mortgageProperty method from GameUI and it activates manageProperties method from GameEngine class. Player object calls the mortgage method to mortgage a property that he/she owns and the Property class calls setMortgage method to make this property mortgaged. Finally, GameEngine calls updateGUI method to make the operations visible to users and GameUI class calls updateGame to transfer the difference to the master class of the game.

4.5 State Diagrams

4.5.1 Entering Quarantine

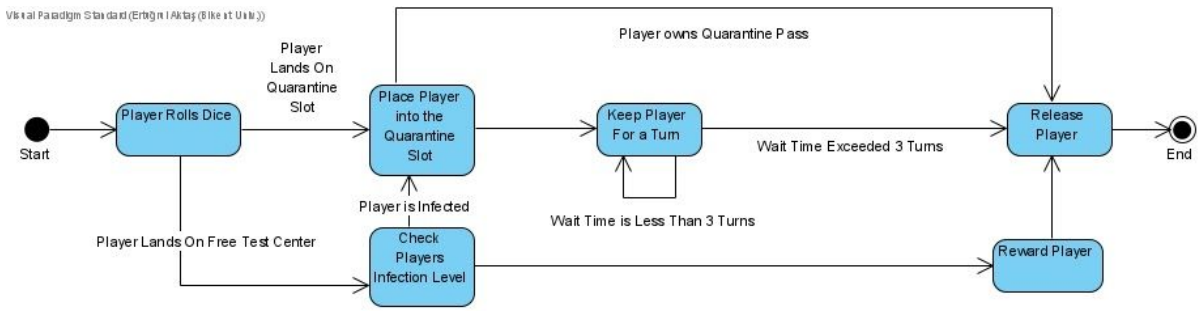

Diagram 4.5.1: State Diagram for Entering Quarantine

The figure demonstrates the quarantine mechanics of the Coronapoly game which is a tweaked version of the original monopoly jail mechanic. When the player rolls dice, the user must land on either the quarantine slot or the free test center slot in order to stimulate the quarantine mechanic of the game. If the player lands on the quarantine slot, it is checked whether the player owns a free quarantine pass, if so then the player is released immediately after he/she is placed on the quarantine slot. If not, the player is released after he/she is kept at the quarantine slot for three turns. The other scenario where the player lands on a free test center is a bit different than the quarantine slot in terms of the outcomes. In this case, the player could get rewarded if the infection level of the player comes out negative and later released from the free test center slot. If not, the player is put into the quarantine slot and the scenario above begins.

4.5.1 Buying Property

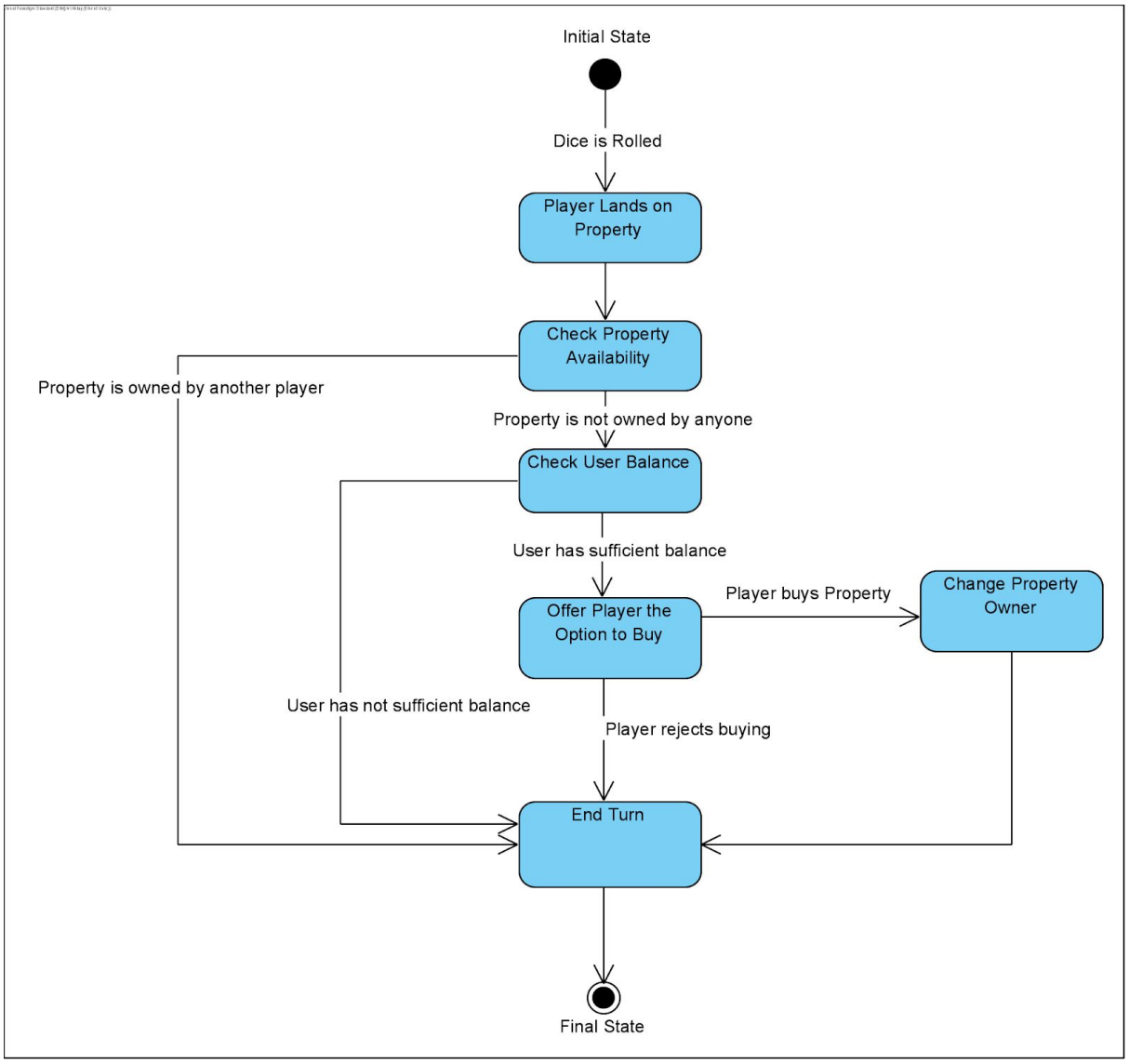

Diagram 4.5.2: State Diagram for Buying Property

The buy property statechart demonstrates the state changes of the game when a buy operation is handled. After the dice is rolled and the player lands on a property, the availability of the property is checked. If the slot is owned by another player, the turn ends for the buy property statechart. The same scenario occurs when the player has insufficient balance even though the property is available. If both of the conditions hold and the user chooses to buy the property, the owner of the property is changed and the game continues. If the player rejects to buy the property then the turn ends once more.

4.6 Screen Mockups

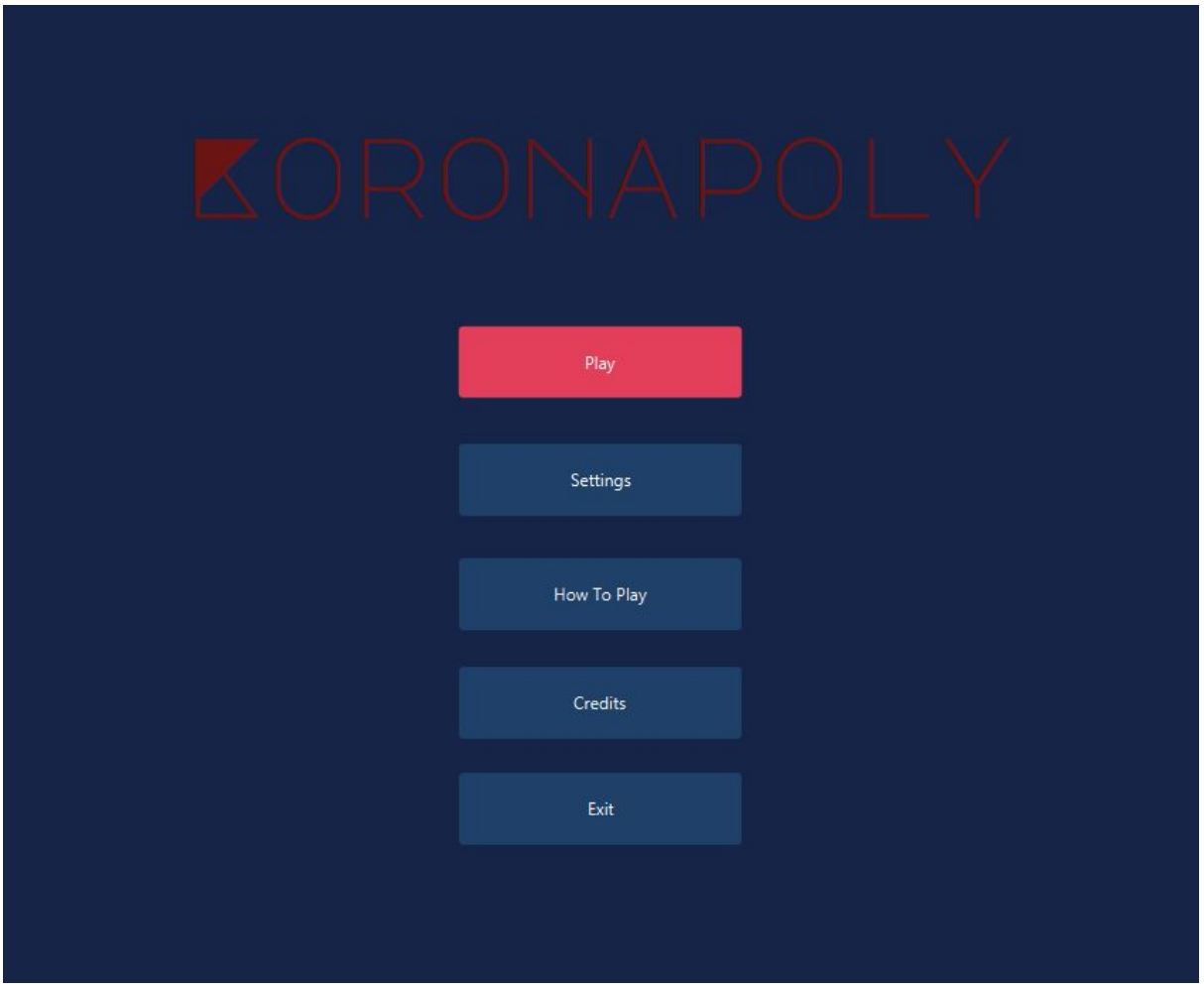

Diagram 4.6.1: Main Menu Mockup

Main opens up as the player starts the game. This menu consists of 5 buttons in total. These buttons are Play, Settings, How To Play, Credits and Quit. If the player chooses Quit button, then the system exits the game.

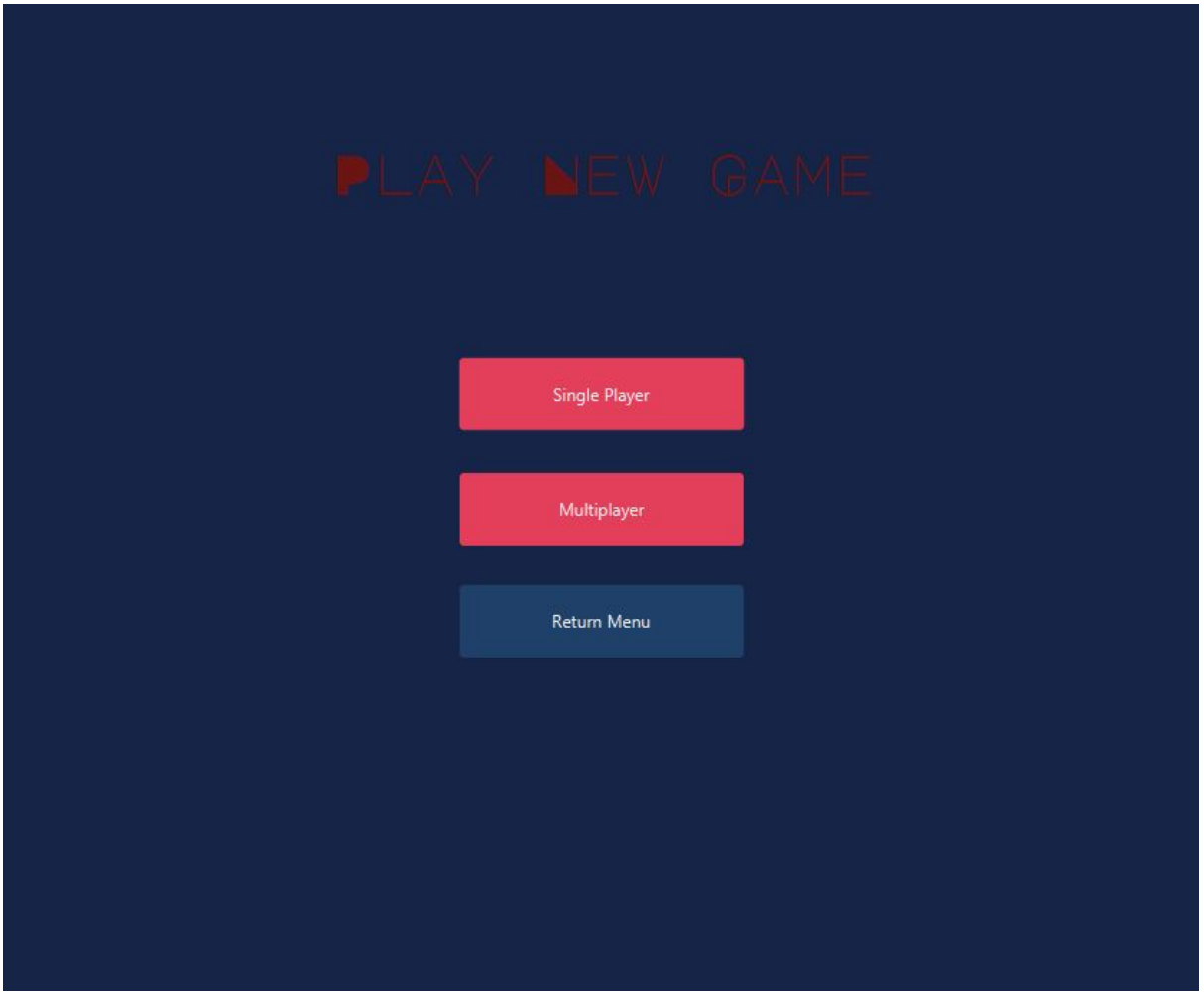

Diagram 4.6.2: Play Game Mockup

After the player clicks the Play buttons, the Play New Game menu shows up. This menu is where players can set up the game as either singleplayer or multiplayer mode. If the player chooses to click the Return Menu button, then the system returns to the main menu.

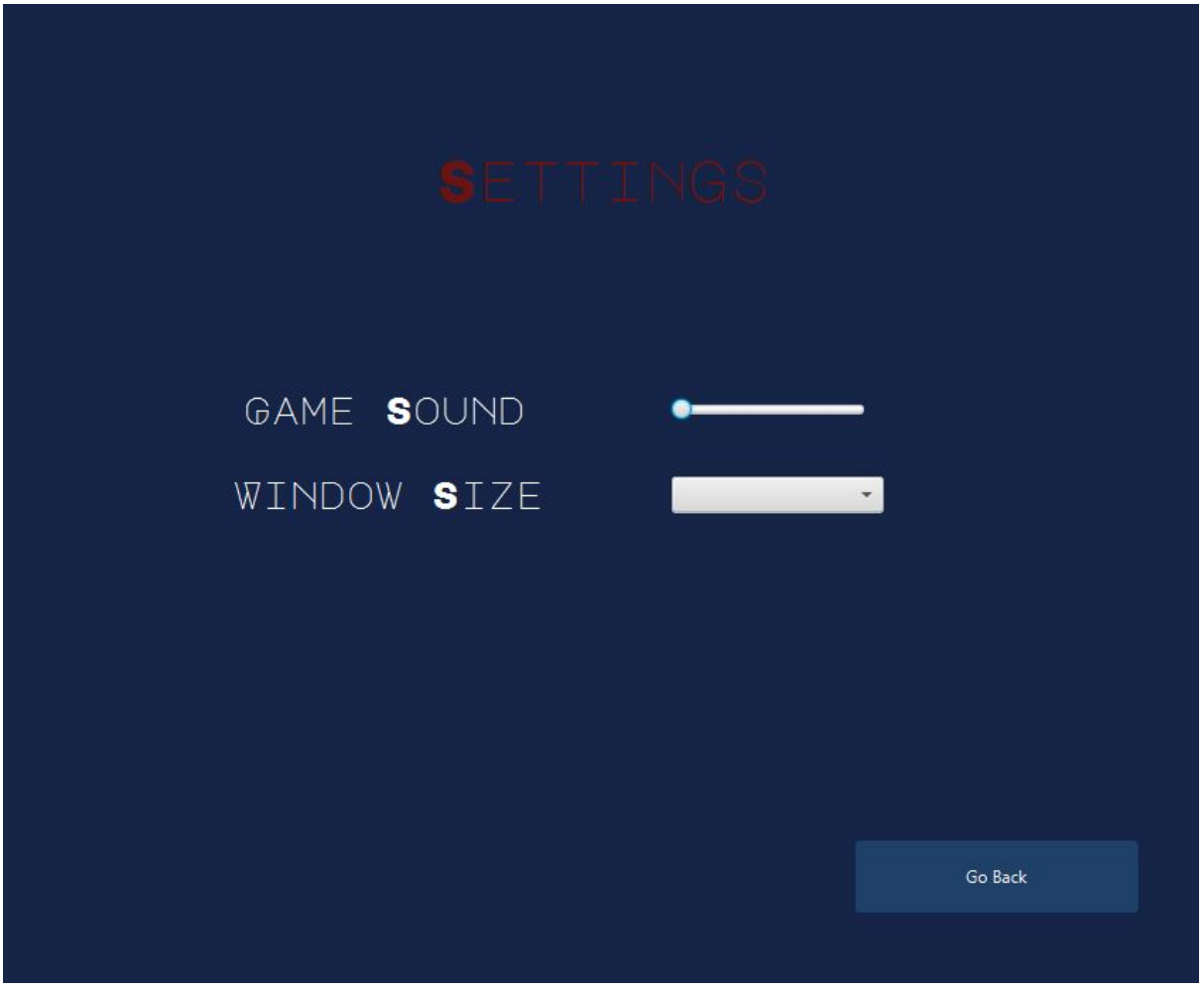

Diagram 4.6.3: Options Mockup

Settings menu enables the player to change music volume and screen size of the game. If the player chooses to click the Return Menu button, then the system returns to the main menu.

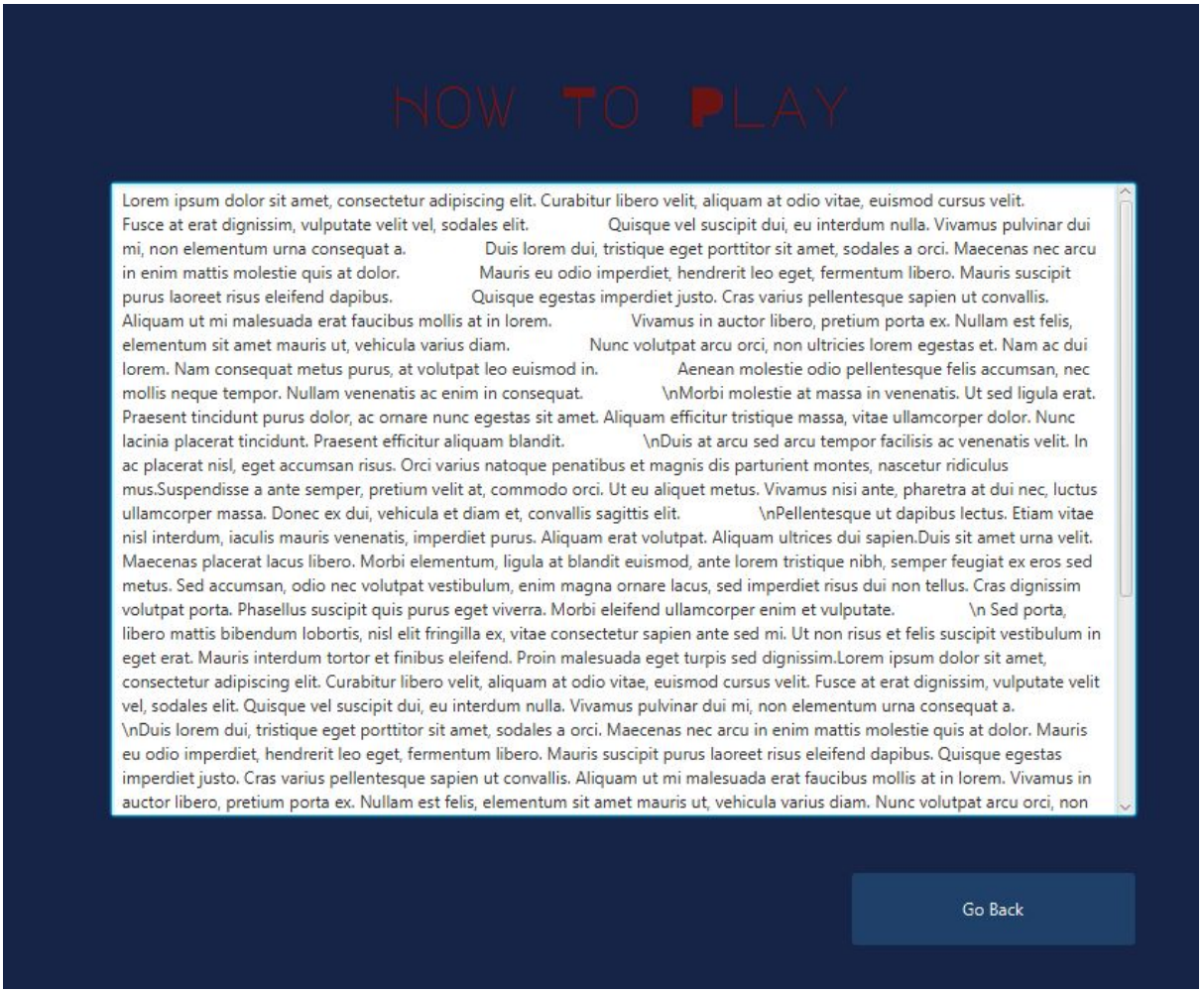

Diagram 4.6.4: How to Play Mockup

The How To Play menu explains the rules and interface of the game to the player. Players can return to the previous menu by clicking the Return Menu button.

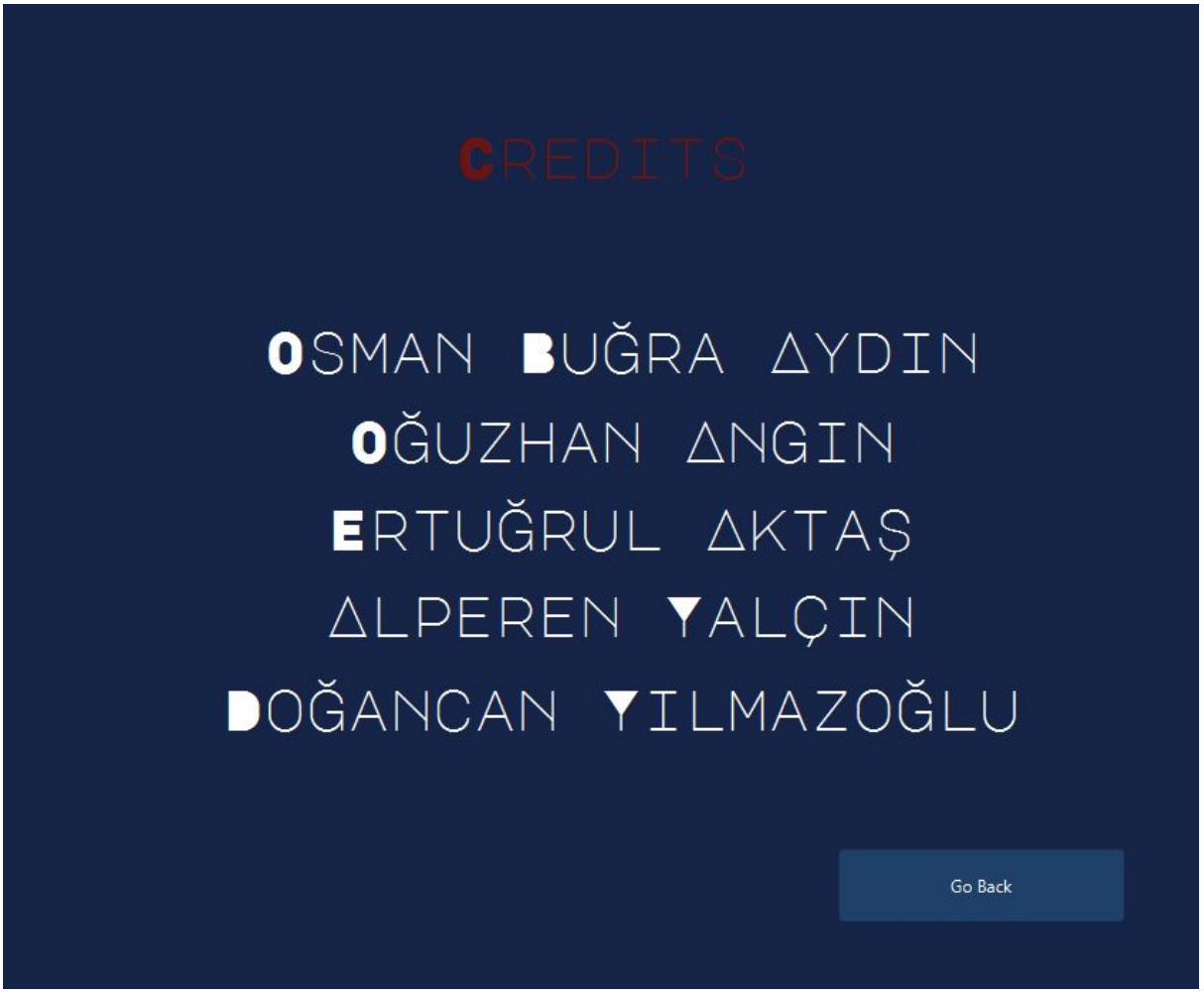

Diagram 4.6.5: Credits Mockup

Credits menu shows the developers of the game to the player.

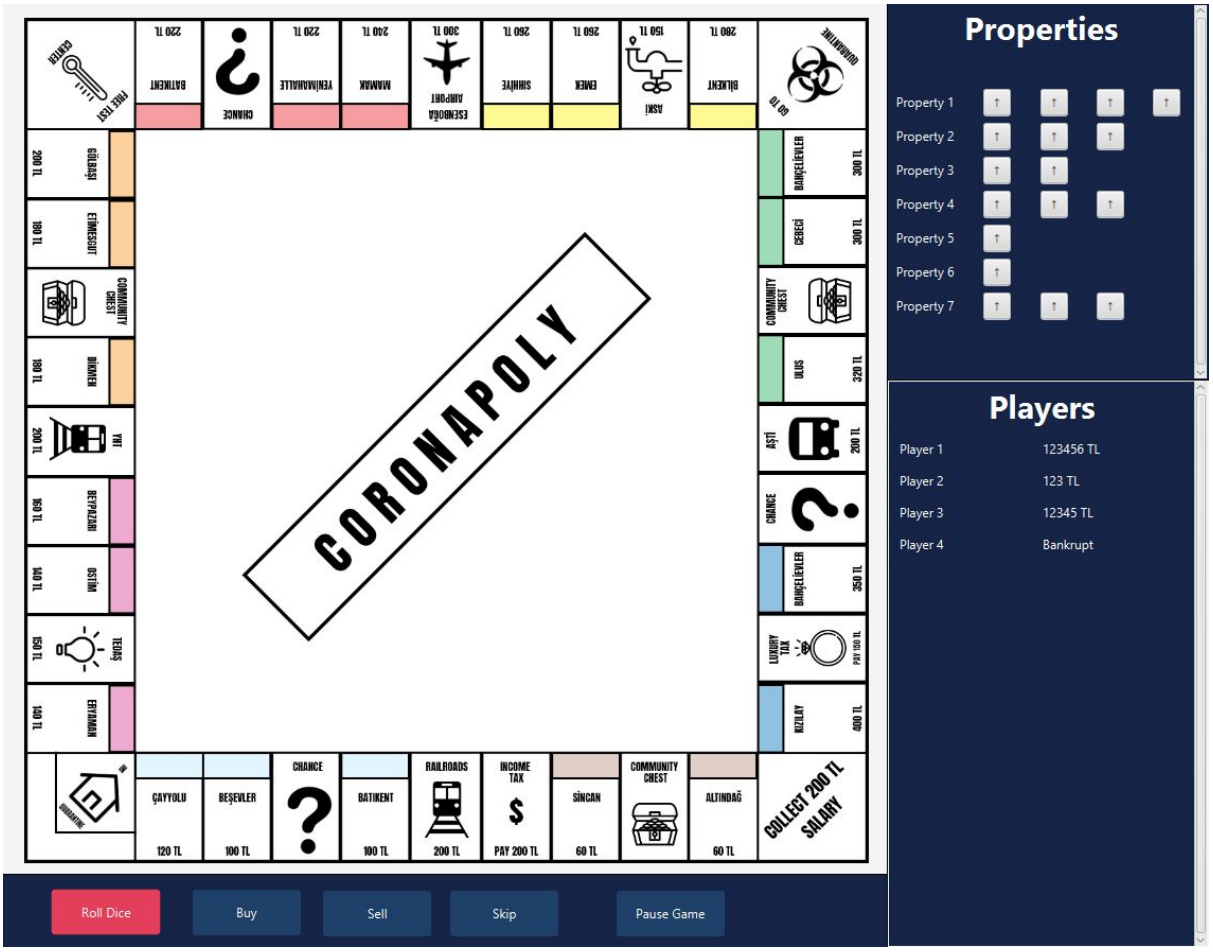

Diagram 4.6.6: Gameplay Mockup

In the Game screen of the game, players can view game boards, action buttons, other players and their properties. By using action buttons in the bottom of the screen, players can accomplish requested actions to complete their turn or pause the game. Players can buy or upgrade their properties from the Properties section of the game screen. Other players' wealth and bankrupt situations are shown to players in the Players section of the game menu.

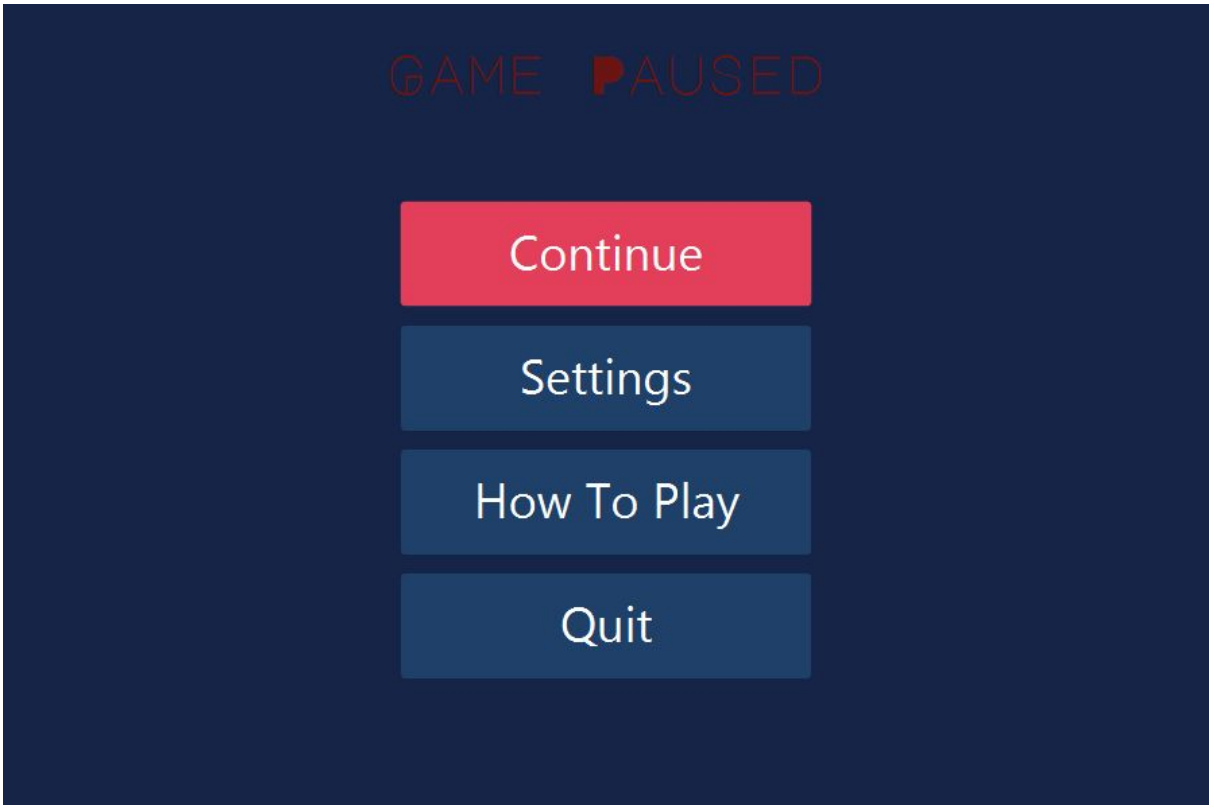

Diagram 4.6.7: Pause Game Mockup

Pause menu shows up when the player presses the “esc” key or “Pause Game” button during the game. Players can access settings and how to play menus by clicking corresponding buttons. Players continue the game by clicking the “Continue” button or exit to the main menu by clicking the “Quit” button.

5 Improvement Summary

In the gameplay part, the social distance rule has been modified. It was indicating that there might be an infection if a player stands on the cell whose adjacent cell has player or players on it. This rule has been replaced with the

possibility of the infection can only increase if there are more than one players on the same cell.

In the player part, there was not any information about the bot players and what they are. Little piece of information has been added at the end of the paragraph.

In the initial economy part, the house count on a neighbourhood has been modified. It was indicating that there can be four houses built in a neighbourhood. This rule has been replaced with building three houses and then being able to build a hospital which will have an expensive rent price.

In the map part, the map was first designed to be a square of slots with a cross of slots in the center of that square. Since this idea is not planned to be implemented, this part of map design is removed from the map part of this report.

In the main menu part, the default frame which enables to exit, minimize and maximize the screen with three buttons provided by the operating system is omitted. Instead of the default frame, the specific exit button in the main menu is created. Also, the information about background music is added in this section.

In the nonfunctional requirements, there was not the section named Security. We have added this section to provide the best experience to the players.

Use case diagram part had some major issues. There were no use case diagrams from the gameplay and the given diagram was missing some relations. Previous use case diagram is revised and renewed to introduce possible relations that were missing in the first iteration. Also a new use case diagram created which focused gameplay elements of the game. Use cases like “buy property” or “pay tax” are added make sure this use case diagram to show it was for Coronapoly game. Both of the diagrams now have captions which were missing in the first iteration.

In the Activity Diagram part, the diagram lacked an exit which is fixed in this iteration. Additionally, the report lacked the state diagrams of the in-game mechanics, hence the mechanics of “going into quarantine” and “buying property” was illustrated using statechart diagrams.

6 Conclusion

In conclusion, Coronapoly is based on the board game Monopoly which is about creating a monopoly in the game’s economy. On the other hand, the project Coronapoly focuses on the game’s goal by adding the Covid-19 virus to the game. The purpose of the game is creating a monopoly in an pandemy environment with few game twists that make the game more fun.

The game will be developed in Java and will be available in both singleplayer and multiplayer game mods. The gameplay of Coronapoly will be simple enough to appeal to users from all ages. The difficulty of the game stays

the same despite the addition of Covid-19 to the the. This is because the amount of rewards for the players are increased as well as the punishments for the players.