Bilkent University

Department of Computer Engineering

CS 319
Object-Oriented Software Engineering
Term Project

Section 2

Group 2F

Monopoly

Analysis Report

Project Group Members

Adil Meriç	21802660
Burak Öçalan	21703769
Doğa Tansel	21802917
Osman Batur İnce	21802609
Ömer Faruk Akgül	21703163

Supervisor: Eray Tüzün

Contents

1. Introduction
2. Overview
    - 2.1. Game Components
    - 2.2. Player
    - 2.3. Main Menu
    - 2.4. Maps
    - 2.5. Instructions
3. Functional Requirements
    - 3.1. Game
        - 3.1.1. Game Setup
        - 3.1.2. Game İnitialization
        - 3.1.3. Gameplay
    - 3.2. Property
        - 3.2.1. Buildings
        - 3.2.2. Cost
        - 3.2.3. Transactions
    - 3.3. Railroads and Canals
        - 3.3.1. Turkey Railroads and Canals
        - 3.3.2. World Railroads and Canals
        - 3.3.3. Bilkent Restaurants/Cafeterias
    - 3.4. Utilities
        - 3.4.1. Bilkent Grass Areas
        - 3.4.2. Turkey Utilities
        - 3.4.3. World Utilities
    - 3.5. Jail
    - 3.6. Community Chest
    - 3.7. Chance
    - 3.8. Tax
    - 3.9. Free Parking Lot
    - 3.10. Tokens
    - 3.11. Dice
    - 3.12. Modes
        - 3.12.1. Blitz Mode

3.12.2. Reverse Mode
4. Nonfunctional Requirements
4.1. User Interface and Human Factors:
4.2. Documentation
4.3. Hardware Considerations
4.4. Performance Characteristics
4.5. Error Handling and Extreme Conditions
4.6. Quality Issues
4.7. Resources and Management Issues
5. System Models
5.1. Use Case Models
5.1.1. Create New Game
5.1.2. Load Old Game
5.1.3. Mode Editor
5.1.4. Options
5.1.5. How to Play
5.1.6. Credits
5.2. Dynamic Models
5.2.1. State Diagram
5.2.2. Purchasing Item
5.2.3. Paying Rent and Tender Processes
5.2.4. Chance / Community Chest Cards
5.2.5. Reverse Game Mode
5.2.6. Blitz Game Mode
5.2.7. Order Determination
5.3. Object and Class Model
5.4. User Interface
5.4.1. Main Menu
5.4.2. Play Game
5.4.3. Load Game
5.4.4. Player Specialization
5.4.5. In-Game Screen
5.4.6. In-Game Property Screen
5.4.7. In-Game Build Screen

5.4.8. In-Game Pause Screen
5.4.9. Options
5.4.10. Credits
5.4.11. Mode Editor Screen
5.4.12. Mode Editor - Specialization Screen
5.4.13. Different Mode Mock-ups
6. Glossary & References

1. INTRODUCTION

You and three or more adversaries are trying to expand your own business empires. You acquire and sell property, exchange money for districts that you desire. You invest in infrastructure and capitalize your districts, making it harder for your opponents to build when they are busy paying you rent. Slowly and with strategy, you drive your opponents to bankruptcy and take the lead in your area, as you expand your power over multiple districts including your opponents'. Finally, when you drive all your opponents to bankruptcy, you will have monopolized and conquered your area.

Monopoly is a very popular, turn-based strategy game based on capital investment. The purpose of each player is to drive their opponents to bankruptcy by investing heavily on the districts they own, which makes their opponents lose money and therefore buying power by paying rent. The game is simple yet very well-suited for object oriented design. The game also is very easy to customize, which allowed us to add our own features. The game follows the classic rules, but there have been a few modifications for adapting the game for computers. We have also added our own game modes for variety and choice:

- Blitz
- Reverse

We will write the game in Java. As of right now we have not decided on which platform we will use to write the code. However, we will, as much as we can, try to use every bit of knowledge that we learn from our CS - 319 lectures and our instructors.

2. OVERVIEW

2.1. Game Components

To play the game, a few things are needed:

- 3-8 players
- Tokens to represent each player
- A pair of dice
- Property, utility and railroad type cards giving information about those spaces
- Community chest and chance cards
- Houses and hotels
- Money
- Map(s) of 40 slots including
    - 22 districts/properties
    - 4 railroads/canals/stations
    - 2 utility stations
    - 3 chance stops
    - 3 community chests
    - 2 tax administrations
    - a jail entry
    - a jail
    - a free parking space
    - and the GO space

In addition, there are a few extra features that can also be used:

- 3 different game modes
    - Classic
    - Blitz
    - Reverse
- Game saving/loading
- Mode and map editor
- User interface settings
- Game manual and rules
- Credits

2.2. Player

Monopoly is a game to be played with multiple people. Since there are many components on the map that can be easily bought, sold, and traded by players, it is easy for players to bankrupt other players. The game cannot be played with only two players because usually one of the two players gets bankrupt pretty quickly and the game does not last very long. Since we intend for monopoly not to be very short or easy, we decided that at least 3 players must be present for the game to be played. The players will all first decide their turns by dice and then take their turns during the game, buying and selling properties to bankrupt the other players and become the winner.

2.3. Main Menu

The game can be played and altered using the main menu. Games can be initialized, customized, saved and loaded from the play game button. Different game modes can be selected through options. Sound effect levels, music levels and visuals can also be changed through options. We also added a button for a mode editor, allowing the players to edit their maps, properties and prices using the three base game modes.

2.4. Maps

Our game includes three default maps for classic mode and one for the other two modes. The default maps are Turkey Cities, World Cities and Bilkent.

The Turkey Cities Map comprises 22 properties, 4 railroads, 2 utilities, 3 chance slots, 3 community chests, 2 tax administrations, a jail entry, a jail, a free parking lot and a start (GO) slot. There are 40 slots in total. (The same types of areas will be referred to as their original names in the game such as property, community chest, etc.)

The World Cities Map comprises 22 cities (property type), 4 canals (railroad type), 2 international establishments (utility type), 3 chance slots, 3 community chests, 2 tax administrations, a jail entry, a jail, a free parking lot and a start (GO) slot. There are 40 slots in total.

The Bilkent Map comprises 22 buildings (property type), 4 cafeterias and restaurants (railroad type), 2 green areas (utility type), 3 chance slots, 3 community chests, 2 tax administrations, a discipline entry (jail entry type), a discipline center (jail type), a free parking lot and a start (GO) slot. There are 40 slots in total.

2.5. Instructions

Players can look at the instructions and rules before or during the game through the menu. These tell the players what guidelines they have to follow to play the game correctly and reach a conclusion.

3. FUNCTIONAL REQUIREMENTS

3.1. Game

3.1.1. Game Setup

Before the game starts, a user must be able to open the game. Then, the user must be able to:

- read the user manual with the instructions and rules (how to play)
- adjust sound effects and music levels
- display credits
- exit the game
- start the game with the following initialization options.

3.1.2. Game Initialization

The game must have some default settings. If a user wants to start a Monopoly game, they must be able to start the game with the default settings or change these settings before the game starts. To change the game settings, the user must be able to:

- select a mode among the game's three modes (classic, blitz, reverse)
- select a number between 3 and 8 (3 and 8 are included) which represents the number of players in the game
- change players' starting money (note that each player must start with the same amount of money)

- change players' total time (only in Blitz mode) (note that each player must start with the same amount of total time)
- select a map among various default maps (only in Classical mode) (note that there is only one default map for modes other than Classical mode)
- modify the selected map by changing the locations' names and prices (details are specified in Mode Editor)
- save the initialization settings for further use
- and start the game with these initialization settings.

As a last step before the game starts, each player must be able to:

- change their nicknames
- and select a token among the set of unselected tokens.

The player turn order will be selected according to how high each player rolls the dice at the start. The highest rolling will be the first to play. Each player starts with $1,500k, made of two $500k, $100k and $50k; six $20k, five each of $10k, $5k and $1k.

3.1.3. Gameplay

After the game starts, in any given time before the game ends, each user must be able to:

- see the game board, the tokens and the dice
- get informed about each player's current status (nickname, money, remaining time (only in Blitz mode)
- get informed about each location's current status, if any (rent, price, owner, etc.)
- pause the game
- leave the game

There a few general rules and functionalities in the game:

- Players roll the dice during their turn which decides the number of slots they will pass. A player can only use their money (excluding exchanges), acquire properties or start construction on their property during their turn.

- When players pass the GO slot during their turn, they receive $200 from the Bank.

- Houses and hotels may be sold to the Bank for half their original purchase price. This can also be done with other players as a trade or a transaction.

- No player can loan money to another, that is the job of the Bank, which only happens when a property is mortgaged by a player.

3.2. Property

Properties in the game are the lands that can be purchased by the players. Holding properties helps the players achieve their objective which is making their opponents go bankrupt.

There are a few functionalities and outcomes for landing on a property:

- When a player lands on unowned property, he or she may purchase the property if he or she has the money necessary for the transaction. The money goes to the bank and the player receives the title deed for that property. If the player decides not to purchase the land, the land remains unowned (and will not be auctioned off).
    - Owning all properties of the same color grants you the ability to charge double rent or put up houses and hotels on the properties of that color. In this case you are said to have a monopoly.

- When a player lands on unmortgaged and owned property, he or she has to pay the owner rent, depending on whether all properties of the same color are owned by that owner and how many houses or hotels there are on that property. If the player decides not to purchase the land, the land remains unowned (and will not be auctioned off).

- When a player lands on a mortgaged property or on one of their own properties, nothing happens.

- Properties can also be acquired through trade with opponents. The players decide for what price they will trade properties.

- Unlike the original game, it is not possible for an unowned property to be sold to the highest bidder in an auction because implementing a feature for the computer will be useless since the banking will be done by the computer. For that reason bankrupt opponents' houses will be free to buy again.

The properties on our default map for all modes will be cities of Turkey. Below is the chart that shows the properties and their corresponding colors and rents:

Table 1: Turkey Properties and Their Rents

			Rent					
Property	**Cost**	**M'tg**	**Site**	**1 hse**	**2 hses**	**3 hses**	**4 hses**	**Hotel**
Van	60	30	2	10	30	90	160	250
Şanlıurfa	60	30	4	20	60	180	320	450
Erzurum	100	50	6	30	90	270	400	550
Diyarbakır	100	50	6	30	90	270	400	550
Yozgat	120	60	8	40	100	300	450	600
Gaziantep	140	70	10	50	150	450	625	750
Mersin	140	70	10	50	150	450	625	750
Hatay	160	80	12	60	180	500	700	900
Samsun	180	90	14	70	200	550	750	950
Kars	180	90	14	70	200	550	750	950
Kayseri	200	100	16	80	220	600	800	1000
Konya	220	110	18	90	250	700	875	1050
Çanakkale	220	110	18	90	250	700	875	1050
Afyonkarahisar	240	120	20	100	300	750	925	1100
Adana	260	130	22	110	330	800	975	1150
İzmit	260	130	22	110	330	800	975	1150

Bursa	280	140	22	120	360	850	1025	1200
Antalya	300	150	26	130	390	900	1100	1275
Muğla	300	150	26	130	390	900	1100	1275
İzmir	320	160	28	150	450	1000	1200	1400
Ankara	350	175	35	175	500	1100	1300	1500
İstanbul	400	200	50	200	600	1400	1700	2000

The Turkey Cities map is based on the classic game, therefore there are no different kinds of properties.

The World Cities map is only available for the classic mode. Below is the chart that shows the properties and their corresponding colors and rents:

Table 2: World Properties and Their Rents

			Rent					
Property	**Cost**	**M'tg**	**Site**	**1 hse**	**2 hses**	**3 hses**	**4 hses**	**Hotel**
Venice	60	30	2	10	30	90	160	250
Barcelona	60	30	4	20	60	180	320	450
Amsterdam	100	50	6	30	90	270	400	550
Seoul	100	50	6	30	90	270	400	550
Delhi	120	60	8	40	100	300	450	600
San Francisco	140	70	10	50	150	450	625	750
Cape Town	140	70	10	50	150	450	625	750
Singapore	160	80	12	60	180	500	700	900
Dubia	180	90	14	70	200	550	750	950
Hong Kong	180	90	14	70	200	550	750	950

Vienna	200	100	16	80	220	600	800	1000
Sydney	220	110	18	90	250	700	875	1050
Rome	220	110	18	90	250	700	875	1050
Bangkok	240	120	20	100	300	750	925	1100
Tokyo	260	130	22	110	330	800	975	1150
İstanbul	260	130	22	110	330	800	975	1150
Los Angeles	280	140	22	120	360	850	1025	1200
Montreal	300	150	26	130	390	900	1100	1275
Berlin	300	150	26	130	390	900	1100	1275
Paris	320	160	28	150	450	1000	1200	1400
New York City	350	175	35	175	500	1100	1300	1500
London	400	200	50	200	600	1400	1700	2000

The World Cities map is also based on the classic game, therefore there are no different kinds of properties.

The Bilkent map is only available for the classic mode. Below is the chart that shows the properties and their corresponding colors and rents:

Table 3: Bilkent Properties and Their Rents

			Rent					
Property	**Cost**	**M'tg**	**Site**	**1 hse**	**2 hses**	**3 hses**	**4 hses**	**Hotel**
East Campus Library	60	30	2	10	30	90	160	250
U Building	60	30	4	20	60	180	320	450
D Building	100	50	6	30	90	270	400	550
Dorm 81/82	100	50	6	30	90	270	400	550

M Building	120	60	8	40	100	300	450	600
V Building	140	70	10	50	150	450	625	750
EB Building	140	70	10	50	150	450	625	750
SB Building	160	80	12	60	180	500	700	900
T Building	180	90	14	70	200	550	750	950
H Building	180	90	14	70	200	550	750	950
C Building	200	100	16	80	220	600	800	1000
NC Building	220	110	18	90	250	700	875	1050
FF Building	220	110	18	90	250	700	875	1050
FB Building	240	120	20	100	300	750	925	1100
G Building	260	130	22	110	330	800	975	1150
FC Building	260	130	22	110	330	800	975	1150
B Building	280	140	22	120	360	850	1025	1200
SA Building	300	150	26	130	390	900	1100	1275
Main Campus Library	300	150	26	130	390	900	1100	1275
EA Building	320	160	28	150	450	1000	1200	1400
A Building	350	175	35	175	500	1100	1300	1500
EE Building	400	200	50	200	600	1400	1700	2000

The Bilkent map is a slight variation of the classic game; the properties are the campus buildings and not cities.

3.2.1. Buildings

The purpose of building houses/hotels is to be able to collect as much rent as possible. A player who does not have any houses or hotels on his or her properties might not take a lot of

rent from other opponents and thus is likely to go bankrupt. If a player builds as much as possible, they will benefit with the amount of rent they receive from their opponents and they can easily drive them to bankruptcy.

Houses are blocks that can be built on properties owned by a player. They can only be built if the player owns all of the same colored properties. At most four houses can be built on a property. The number of houses on a property will increase its value and therefore the rent paid for stopping on that property. If a property is mortgaged or traded, all houses on it will be sold to the Bank for half their original prices, and no rent will be acquired from the property. Houses are shown with these green icons:

Hotels allow you to increase a property's rent to its highest possible. This will easily allow you to bankrupt your opponents. Hotels cost the same as one house and when it is built on the property the houses on the property are torn down. The higher the cost of a property, the higher is the rent and therefore it is a good strategy to try to acquire the most costly color groups of properties. When a property is mortgaged or traded, the hotel on it must be sold to the bank for half its original price. Hotels are shown with red icons:

Players are able to purchase up to four houses or one hotel for their properties if they own all properties of the same color. They must wait for their turn for these purchases but do not have to be on those properties to build on them.

The houses and hotels purchased for properties have to be developed evenly, meaning if there are three properties of the same color, one cannot have two houses on it before the rest have one house on them. This means hotels can only be purchased when all properties of the same color have four houses on them.

3.2.2. Cost

House costs change from property to property. Below are the original costs for only one house:

- Browns\Purples and Light Blues - $50
- Pinks and Oranges - $100
- Reds and Yellows - $150
- Greens and Dark blues - $200

Hotels cost the same as 1 house and 4 houses need to be on a property to build a hotel. If a hotel is built, it will replace the 4 houses.

3.2.3. Transactions

When a player owes another money (usually due to rent), he or she can pay by mortgaging properties and paying cash, but if that is not enough houses and/or hotels can be sold to pay the money. Houses and hotels are sold at half their original price, which is a loss of profit.

- Browns\Purples and Light Blues - $25
- Pinks and Oranges - $50
- Reds and Yellows - $75
- Greens and Dark blues - $100

Properties cannot be sold or traded when they have buildings on them; they must be sold to the bank before these transactions occur.

Selling to pay rent will cause the player to lose profit, hence it is a good idea to acquire high-rent properties and build on them before others can. This way after paying a high amount of rent, the same amount or more money can be gained by receiving rent.

3.3. Railroads and Canals

Railroads can also be purchased by players when landed on. A player who lands on an owned railroad station will have to pay the owner rent. The price of rent charged by the owner increases with the number of railroads the player owns. The owner charges as follows:

- $25 for 1 railroad station
- $50 for 2 railroad stations
- $100 for 3 railroad stations
- $200 for 4 railroad stations

Railroad stations can be sold or traded the same as other properties, however they cannot be mortgaged as nothing can be built on them.

3.3.1. Turkey Railroads and Canals

The railroads on our default map for all modes will be from the Turkey Cities map. Below is the chart that shows the railroads and their corresponding colors and rents:

Table 4: Turkey Railroads with Rents

Railroad Station	Cost	M'tg	Rent
Haydarpaşa Station, İstanbul	200	100	25 or 50 or 100 or 200
Ankara Station, Ankara	200	100	25 or 50 or 100 or 200
Merinos Station, Bursa	200	100	25 or 50 or 100 or 200
Alsancak Station, İzmir	200	100	25 or 50 or 100 or 200

The Turkey Cities map is based on the classic game so we used railroads for that map.

3.3.2. World Railroads and Canals

The World Cities map contains canals. Below is the chart that shows the canals and their corresponding rents:

Table 5: World Railroads with Rents

Canals	Cost	M'tg	Rent
Suez Canal	200	100	25 or 50 or 100 or 200
The English Channel	200	100	25 or 50 or 100 or 200

The Bosphorus	200	100	25 or 50 or 100 or 200
Panama Canal	200	100	25 or 50 or 100 or 200

The World Cities map is based on the classic game and the canals are similar in type to railroads.

3.3.3. Bilkent Restaurants/Cafeterias

The Bilkent map contains cafes and restaurants. Below is the chart that shows the cafes and their corresponding rents:

Table 6: Bilkent Railroads with Rents

Cafes and Restaurants	Cost	M'tg	Rent
Speed	200	100	25 or 50 or 100 or 200
Mozart Cafe (B)	200	100	25 or 50 or 100 or 200
Cafe In	200	100	25 or 50 or 100 or 200
Tabldot Cafeteria	200	100	25 or 50 or 100 or 200

The Bilkent map is a slight variation of the classic game; there are cafeterias and restaurants instead of railroads.

3.4. Utilities

Just as other properties and areas, utilities can also be purchased by players and just as railroad stations, they cannot be built on or mortgaged. They can be traded or sold to other players. If an opponent lands on a utility owned by a player, he or she must pay 4 times the dice rolled if the owner owns only one utility station. If the owner owns both utility stations the rent charged will be 10 times the dice rolled.

3.4.1 Turkey Utilities

The utilities on our default map for all modes will be from the Turkey Cities map. Below is the chart that shows the utilities and their corresponding colors and rents:

Utility	Cost	M'tg	Rent
Electric Company	150	75	4×dice or 10×dice
Water Works	150	75	4×dice or 10×dice

Table 7: Utilities on Turkey Map with Rents

The Turkey Cities map is based on the classic game so we used utilities for that map.

3.4.2 World Utilities

The World Cities map contains international establishments. Below is the chart that shows them and their corresponding rents:

International Establishments	Cost	M'tg	Rent
European Union	150	75	4×dice or 10×dice
United Nations	150	75	4×dice or 10×dice

Table 8: Utilities on World Map with Rents

The World Cities map is based on the classic game but there are international establishments instead of utilities.

3.4.3 Bilkent Grass Areas

The Bilkent map contains outdoor areas. Below is the chart that shows them and their corresponding rents:

Outdoor Areas	Cost	M'tg	Rent
Main Campus Dorms Green Area	150	75	4×dice or 10×dice
Mayfest Green Area	150	75	4×dice or 10×dice

Table 9: Utilities on Bilkent Map with Rents

The Bilkent map is a slight variation of the classic game; there are outdoor areas instead of utilities.

3.5. Jail

Either landing on the "Go to Jail" slot, drawing the "Go to Jail" card or rolling three consecutive doubles sends the player to jail where the player's token will be moved to the jail cell on the opposite side of the board. In this situation, the player cannot take any money from the "GO" slot as the token is considered to be moving backwards.

A player in jail can collect rent from another player but cannot make any other transactions such as buying houses or selling houses/hotels or properties.

To get out of jail, either a "Get out of jail free" card can be used, a double can be thrown which allows the player to move the amount rolled but not roll again, or by default the player can wait three turns and in the end of the third turn can exit by paying a $50 fine and moving the amount rolled. The player can also choose to wait one turn to move after the fine is paid.

Landing on the jail cell slot in a normal roll is considered as "visiting" and no action occurs.

3.6. Community Chest

Community chest includes cards which players have to draw when they land on the community chest spaces. These cards include random actions that the player has to take. When a card is drawn, it is used and then placed on the bottom of the deck. The deck is shuffled when the game is reset. The money paid to the Bank due to these cards are placed in the middle of the board, where they are picked up by players who come to the free parking lot.

These are the cards that are included in the Turkey Map:

- Advance to "Go". (Collect $200)
- Bank error in your favor. Collect $200.
- Doctor's fees. {fee} Pay $50.
- From sale of stock you get $50.
- Get out of Jail Free -This card may be kept until needed.
- Go to Jail. Go directly to jail. Do not pass Go, Do not collect $200.
- Grand Opera Night. Collect $50 from every player for opening night seats.
- Holiday. Fund matures. Receive $100.
- Income tax refund. Collect $20.
- It is your birthday. Collect $10 from every player.
- Life insurance matures – Collect $100
- Hospital Fees. Pay $50.
- School fees. Pay $50.
- Receive $25 consultancy fee.
- You are assessed for street repairs: Pay $40 per house and $115 per hotel you own.
- You have won second prize in a beauty contest. Collect $10.
- You inherit $100.

These are the cards that are included in the Mode:

- Advance to "Go". (Collect $200)
- Bank error in your favor. Collect $200.
- Doctor's fees. Pay $50.
- From sale of stock you get $50.
- Get out of Jail Free -This card may be kept until needed.
- Go to Jail. Go directly to jail. Do not pass Go, Do not collect $200.
- Grand Opera Night. Collect $50 from every player for opening night seats.
- Holiday. Fund matures. Receive $100.
- Income tax refund. Collect $20.
- It is your birthday. Collect $10 from every player.
- Life insurance matures – Collect $100

- Hospital Fees. Pay $50.
- School fees. Pay $50.
- Receive $25 consultancy fee.
- You are assessed for street repairs: Pay $40 per house and $115 per hotel you own.
- You have won second prize in a beauty contest. Collect $10.
- You inherit $100.

3.7. Chance

The second type of cards are the Chance cards. The chance cards also work like the community chest cards (card is drawn when the player lands on Chance space, action on card is taken and then card is placed on the bottom of the deck, deck is shuffled when the game is reset). Chance cards have much more severe consequences for players compared to Community Chest cards and they often move players more. The money paid to the bank due to these cards also go into the middle, where they are picked up by players who come to the free parking lot.

These are the cards that are included in the game:

- Advance to "Go". (Collect $200)
- Advance to Çanakkale. If you pass Go, collect $200.
- Advance to Gaziantep. If you pass Go, collect $200.
- Advance token to nearest Utility. If unowned, you may buy it from the Bank. If owned, throw dice and pay the owner a total 10 (ten) times the amount thrown.
- Advance token to the nearest Railroad Station and pay the owner twice the rental to which he/she is otherwise entitled. If Railroad is unowned, you may buy it from the Bank. *(There are 2 of these.)*
- Bank pays you a dividend of $50.
- Get out of Jail free. This card may be kept until needed, or traded/sold.
- Go to Jail. Do not pass GO, do not collect $200.
- Make general repairs on all your property: For each house pay $25, For each hotel pay $100.

- Pay speeding fine of $15.
- Take a trip to the Haydarpaşa Railroad Station. If you pass Go, collect $200.
- Advance token to İstanbul. You have been elected Chairman of the Board. Pay each player $50.
- Your building loan matures. Collect $150.
- You have won a crossword competition. Collect $100.

3.8. Tax

When players come to a tax slot, they must pay the Bank the amount of money indicated on the slot. This can either be Luxury Tax which is $100 or Income Tax which is $200. The money paid to the bank goes into the middle, which is picked up by a player when he or she comes to the free parking lot.

In the Turkey Cities and World cities maps, the slots stay as tax slots. However in the Bilkent map, there are disciplinary punishment slots instead. The prices are still the same.

3.9. Free Parking Lot

When a player lands on the free parking space, he or she will receive all of the money in the middle, which comes from players' either paying taxes or paying dues to the bank by drawing chance cards or community chest cards. This creates tension in the game and also gives players the ability to bounce back and not go bankrupt in bad situations.

3.10. Tokens

Tokens are the playing pieces that signifies each player. These are the tokens we have chose for our edition of Monopoly:

3.11. Dice

There are two 6-sided dice used to indicate the amount of spaces moved in the game.

If the dice are rolled as a double (meaning both dies have the same number) the player who rolled the dice has the ability to roll once again. If three consecutive doubles are rolled, the player with the turn goes to jail and moves the piece to the jail cell. While in jail, if the player rolls a double while waiting three turns, he or she can get out and can move the number or spaces rolled. However, there will be no re-roll in this situation.

3.12. Modes

There will be 3 modes in total including the classic mode. For players not to forget which mode they are playing, a mode icon will be located on the screen during the game.

3.12.1. Blitz

In this mode, everything is almost the same except that this game is a race against time. Each player will start with a limited amount of time and as long as it is a player's turn the time that player has will decrease. When a player's time runs out, that person will not be able to make any moves anymore and a pop-up saying they have run out of time will appear. When everyone but one player's time runs out, the game finishes. Whoever has the most money and property (in terms of money) wins.

The time left for a player will be shown at all times. During a player's turn, their clock will be in the middle of the board and their time will be running out. When the turn ends, the clock and the time appears next to the player's name and icon and will be paused.

3.12.2. Reverse:

In this game mode, all of the community chest slots are turned into *reverse* slots. When a player gets to any *reverse* slot, the game flow turns from clockwise to counterclockwise or vice versa. A reverse sign will appear on the board when the game becomes reversed.

As a last note of Functional Requirements, game rules are referenced from [1].

4. NONFUNCTIONAL REQUIREMENTS

4.1. User Interface and Human Factors

- ➔ Any human who is bigger than 7 years old and has played at least one other computer game that has a graphical user interface should be able to play this game.
- ➔ Users who have played the monopoly box game before and have used a computer several times should be able to understand the game in less than 20 minutes, without reading the user manual.
- ➔ Users who haven't played the monopoly box game before should be able to understand the game in less than 3 hours, with reading the user manual and practicing.
- ➔ Users should not be able to crash the game by only using the user interface provided by the game.
- ➔ If the game is working on a user's computer, the user must be able to play the game with only a working mouse and a screen which is capable of fully monitoring the game.

4.2. Documentation

- ➔ A "How To Play" document should be available for the users to help them understand how to play the game
- ➔ Documentation of the source code of the game should be available for the future maintainers of the game to help them understand the structure of the code

4.3. Hardware Considerations

- ➔ Any computer with a recommended hardware which is given below, should be able to run the game.
- ➔ Recommended Hardware:
- ➔ CPU: SSE2 compatible instruction set with clock frequency at least 1.5 GHz
- ➔ RAM: 512 MB
- ➔ Video Card: 256 MB VRAM with Shader Model 3.0 or better
- ➔ Free Disk Space: 150 MB

4.4. Performance Characteristics

- ➔ If the game is played with a computer that has recommended hardware, the game should start in less than 1 second.
- ➔ With the same computer above, the game should respond to any of the user actions in less than 0.25 seconds.

4.5. Error Handling and Extreme Conditions

- ➔ Game should ignore all error prone actions from the user and send a warning message.
- ➔ The crash rate of the game must be lower than 0.1%.
- ➔ In at least %50 percent of crashes, the game should save the current progres. After saving, the game should close itself with an error message.
- ➔ Even if the game can't save the current progress of the game in a crash condition, users should be able to continue from the previous turn after restarting the game.

4.6. Quality Issues

- ➔ The game should keep and display all information correctly.
- ➔ For each user action that is outside the rules of the game, the game should not proceed the action and send a warning message to the user.
- ➔ After a failure, the game should be available to restart in less than 1 second.
- ➔ The game should be not available for at most 5 minutes in a 24 hour period.
- ➔ The game should be portable to Windows, Macintosh and Linux machines.

4.7. Resources and Management Issues

- ➔ The game should be backed up once in a month.
- ➔ The game itself should be responsible for back up.
- ➔ The game should be installed by users.
- ➔ After the submission of the game, the client should maintain the game.

5. SYSTEM MODELS

5.1. Use Case Models

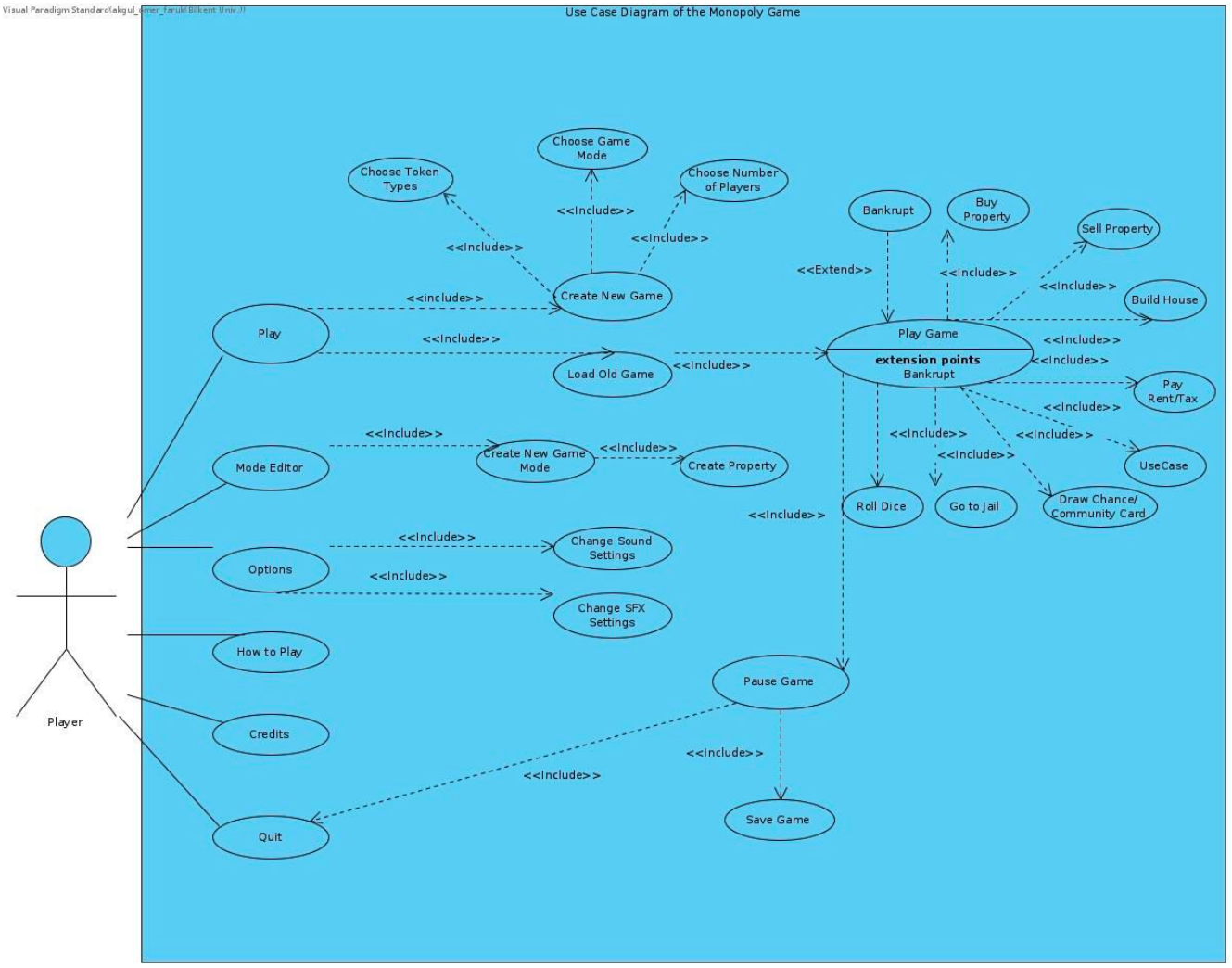


Figure 1

5.1.1. Create New Game

Name: Create New Game

Primary Actor: Player

Stakeholders and Interests: Player wants to play a new game.

Pre-conditions:

➔ The player is in the main menu.

Entry-conditions:

➔ Player clicks on “Play Game” and “New Game”, then fills the demanded areas

Exit conditions:

➔ All players except one bankrupts, i.e. one of the players wins the Monopoly.

➔ Selecting the “Quit Game” option after clicking on the “Pause” icon.

Success Scenario Event Flow:

➔ Player selects “Play Game”.

➔ Player selects “New Game”.

➔ “Player Count” and “Game Mode” are chosen from the given scroll bar.

➔ Player selects one of the given tokens to represent his/her character.

➔ “Start” button is clicked on.

➔ Player “Roll”a dice when it’s time.

➔ Player performs one of the following actions, depending on her position

- ◆ Buying a new property
- ◆ Selling existing property
- ◆ Building a house to the property
- ◆ Paying a tax/rent
- ◆ Going to jail
- ◆ Drawing a chance/ community chest card

➔ Player clicks on “End Turn”

➔ Rolling the dice and ending the turn continues until the player bankrupts or wins the game.

➔ Game finishes.

Alternative Event Flows:

➔ If player wants to finish the game or wants to continue later:

- ◆ Player clicks on the pause option.
- ◆ “Save Game” is selected if the player wants to continue later.
- ◆ Player clicks on the “Quit” button.

➔ Game finishes.

5.1.2. Load Old Game

Use Case: Load Old Game

Primary Actor: Player

Stakeholders and Interests: Player wants to play a saved game.

Pre-conditions:

- ➔ The player is in the main menu.
- ➔ There is at least one saved game.

Entry-conditions:

- ➔ Player selects “Play Game” and “Load Old Game”, respectively.

Exit conditions:

- ➔ All players except one bankrupts, i.e. one of the players wins the Monopoly.
- ➔ Selecting the “Quit Game” option after clicking on the “Pause” icon.

Success Scenario Event Flow:

- ➔ Player selects “Play Game”.
- ➔ Player selects “Load Old Game”.
- ➔ One of the previously saved games is chosen.
- ➔ Game continues from where it left off.
- ➔ Player “Roll”a dice when it’s time.
- ➔ Player performs one of the following actions, depending on her position
    - ◆ Buying a new property
    - ◆ Selling existing property
    - ◆ Building a house to the property
    - ◆ Paying a tax/rent
    - ◆ Going to jail
    - ◆ Drawing a chance/ community chest card
- ➔ Player clicks on “End Turn”

- ➔ Rolling the dice and ending the turn continues until the player bankrupts or wins the game.
- ➔ Game finishes.

Alternative Event Flows:

- ➔ If player wants to finish the game or wants to continue later:
  - ◆ Player clicks on the pause option.
  - ◆ "Save Game" is selected if the player wants to continue later.
  - ◆ Player clicks on the "Quit" button.
- ➔ Game finishes.

5.1.3. Mode Editor

Use Case: Mode Editor

Primary Actor: Player

Stakeholders and Interests: Player wants to play with his/her own map.

Pre-conditions: The player is in the main menu.

Entry-conditions: Player selects "Mode Editor" on the main menu.

Exit conditions: Player pauses editing and clicks on "Quit Editor".

Success Scenario Event Flow:

- ➔ Player selects "Mode Editor".
- ➔ System demands the user to enter "Mode Name" and "Base Mode".
- ➔ Player fills these areas and clicks on "Edit Mode".
- ➔ Players enter required data to create a game mode such as salary and start capital.
- ➔ Player selects locations on the map one by one and edits them. In this step, Create Property use case eases creating a property. Since a property has a name, cost, rent

amounts, mortgage cost and house/ hotel costs, the player has to fill all these blocks and save property.

➔ After filling all the locations on the map, the player clicks on “Pause” and selects the “Save Mode” option.

➔ Player returns back to the main menu.

Alternative Event Flows:

➔ If player gives up creating a new mode:

- ◆ Player clicks on the “Pause” option while creating a new mode.
- ◆ System offers the player two options: “Save Game” and “Quit Editor”.
- ◆ Player selects “Quit Editor”.

➔ Mode editor is terminated without creating a new mode.

5.1.4. Options

Use Case: Options

Primary Actor: Player

Stakeholders and Interests: Player wants to change the game settings.

Pre-conditions: The player is in the main menu.

Entry-conditions: Player selects “Options” on the main menu.

Exit conditions: Player clicks on “back” while he/she is in Options.

Success Scenario Event Flow:

➔ Player selects “Options”.

➔ Player adjusts sliders to change sound and SFX level.

➔ Player clicks on the “Save” option.

➔ Main menu is displayed.

Alternative Event Flows:

- ➔ Player does not want to change the existing settings:
  - ◆ Player goes back to the main menu without saving the changes.

5.1.5. How to Play

Use Case: How to Play

Primary Actor: Player

Stakeholders and Interests: Player wants to learn the general rules of the game.

Pre-conditions: The player is in the main menu.

Entry-conditions: Player selects “How to Play” on the main menu.

Exit conditions: Player clicks on “back” while he/she is in How to Play.

Success Scenario Event Flow:

- ➔ Player selects “How to Play”.
- ➔ Player reads the guide and slides the window using the scroll bar.
- ➔ Player clicks on the “back” option.
- ➔ Main menu is displayed.

5.1.6. Credits

Use Case: Credits

Primary Actor: Player

Stakeholders and Interests: Player wants to read about credits of the Monopoly game.

Pre-conditions: The player is in the main menu.

Entry-conditions: Player selects “Credits” on the main menu.

Exit conditions: Player clicks on “back” while he/she is in Credits.

Success Scenario Event Flow:

- ➔ Player selects “Credits”.
- ➔ Player reads the content and slides the window using the scroll bar.
- ➔ Player clicks on the “back” option.
- ➔ Main menu is displayed.

5.2. Dynamic Models

5.2.1 State Diagram

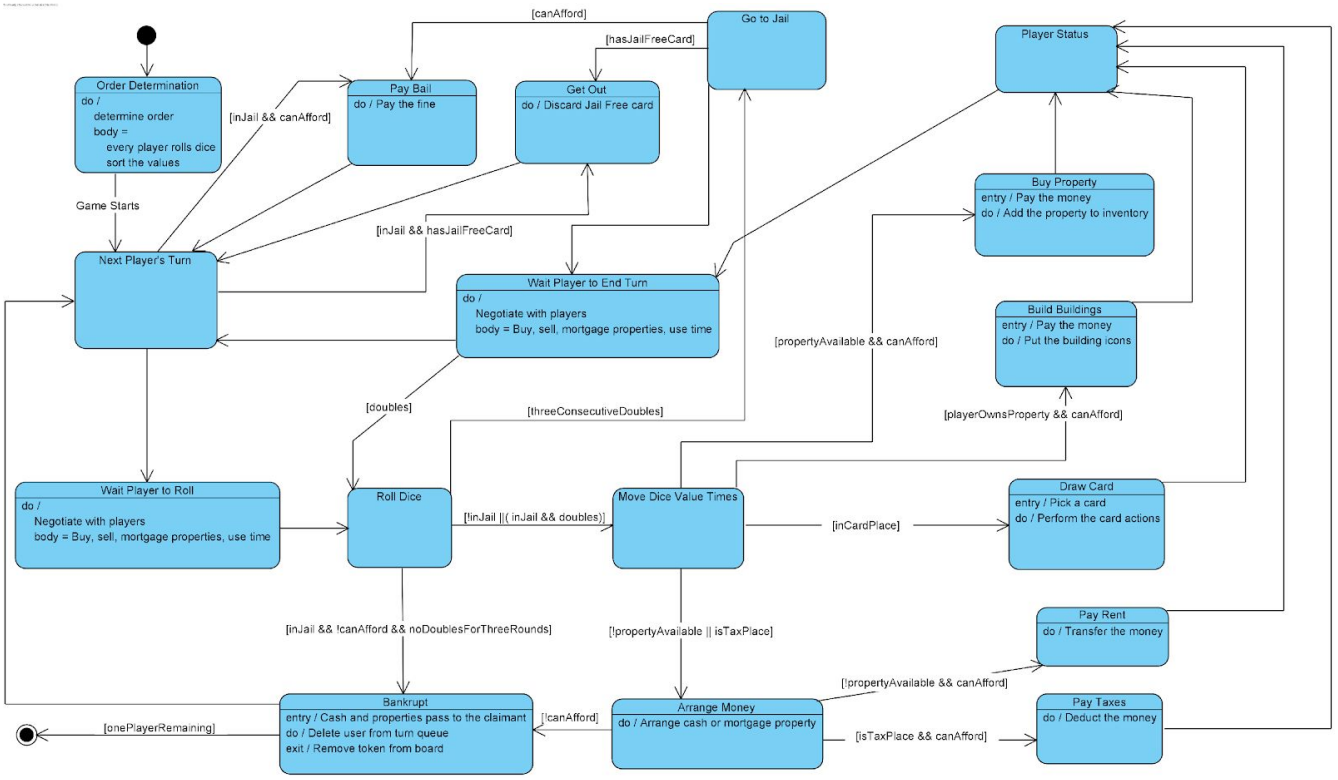


Figure 2: State Diagram of The Classic Mode of Our Monopoly Game

As it is stated right under the figure, the state machine diagram above describes the general flow of a classic mode Monopoly game. At the start, in the order determination state, the order of the players are determined by rolling dice. Then, according to the order, the turns start. The player can wait before the roll to make any kind of bargains or arrangements. Then, the player rolls the dice and if the player is not in jail or the player is in jail and throws a double, the token is moved. If the player throws the third consecutive doubles, the player goes into the jail. However, if the player is in jail, cannot afford the bail and could not throw doubles for three consecutive turns, the player **goes bankrupt.**

Then, according to the place that the token lands, the flow divides into four main parts. These are **Arrange Money, Draw Card, Build Buildings** and, **Buy Property**. The **Draw Card** state is in action when the property is a card place. Then, the user takes actions accordingly. The **Buy Property** state can be chosen when the property is available and the

user can afford the property. The **Build Buildings** state can be chosen when the player owns the property and can afford any kind of building actions. The **Arrange Money** state is just before **Pay Taxes** or **Pay Rent** state. If the player is not able to arrange money through cash or properties, the player goes **bankrupt** and then if there is only one player remaining the game ends.

If the player can afford, the player pays the dues rent if the property is not available or the player pays the taxes if the player is in a tax place. In addition to all actions after the **Move Dice Value Times** state, if the conditions are satisfied, even if the player can afford the available property that the player is on, the player might not choose to buy it. Therefore, the player may go to the **Wait Player To End Turn** state directly.

After **Pay Rent, Pay Taxes, Draw Card, Buy Property, or Build Buildings** if the player does not go **bankrupt,** the player moves to the **Player Status** state. This is not an actual state, I have used this state not to use many pointing arrows. From this state, the player goes to the **Wait Player To End Turn** state. If the player **goes to jail**, the player has two options before the turn is over. The player can choose the **Get Out** state if the user does have the **Jail Free** card. Besides, the player can choose the **Pay Bail** stage if the users can afford to pay the bail. After these states, the next state is **Next Player's Turn.** And this is basically the flow of the game until there is only one player remaining, the game ends.

5.2.2. Purchasing Item

The diagram below displays the high level interactions between classes, when a player comes to the property which is not purchased yet.

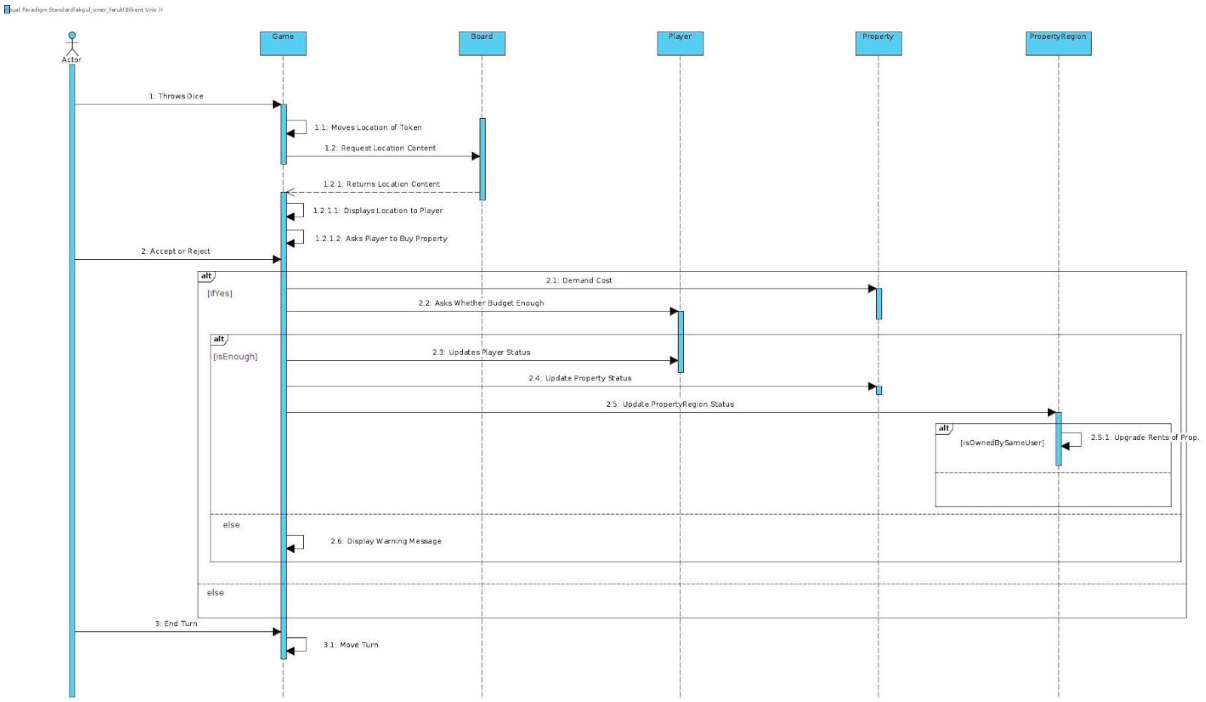


Figure 3

Game class acts as an engine of the Monopoly so most of the interactions are triggered by Game class.

All the actions in the diagram above starts when the player rolls a dice. Game class moves the token of the player forward based on the sum of dice. Then, a query is sent to the Board class to learn the content of the place. The content of the location including name, cost and rent information is displayed. If the player decides to purchase, Game withdraws the price of money from the player's account and updates the assets of the player. Also, the status of property is updated such that the owner information can be learned during the upcoming steps. However, if the player’s account is less than the price, a warning message is displayed on the screen. Finally, the actor ends his/her turn, which triggers Game to move the turn.

5.2.3. Paying Rent and Tender Processes

The diagram below is designed to show how objects interact with each other, when the player comes to the place owned by another player. In addition, the bankrupt process for the player whose money is going to be withdrawn is explained.

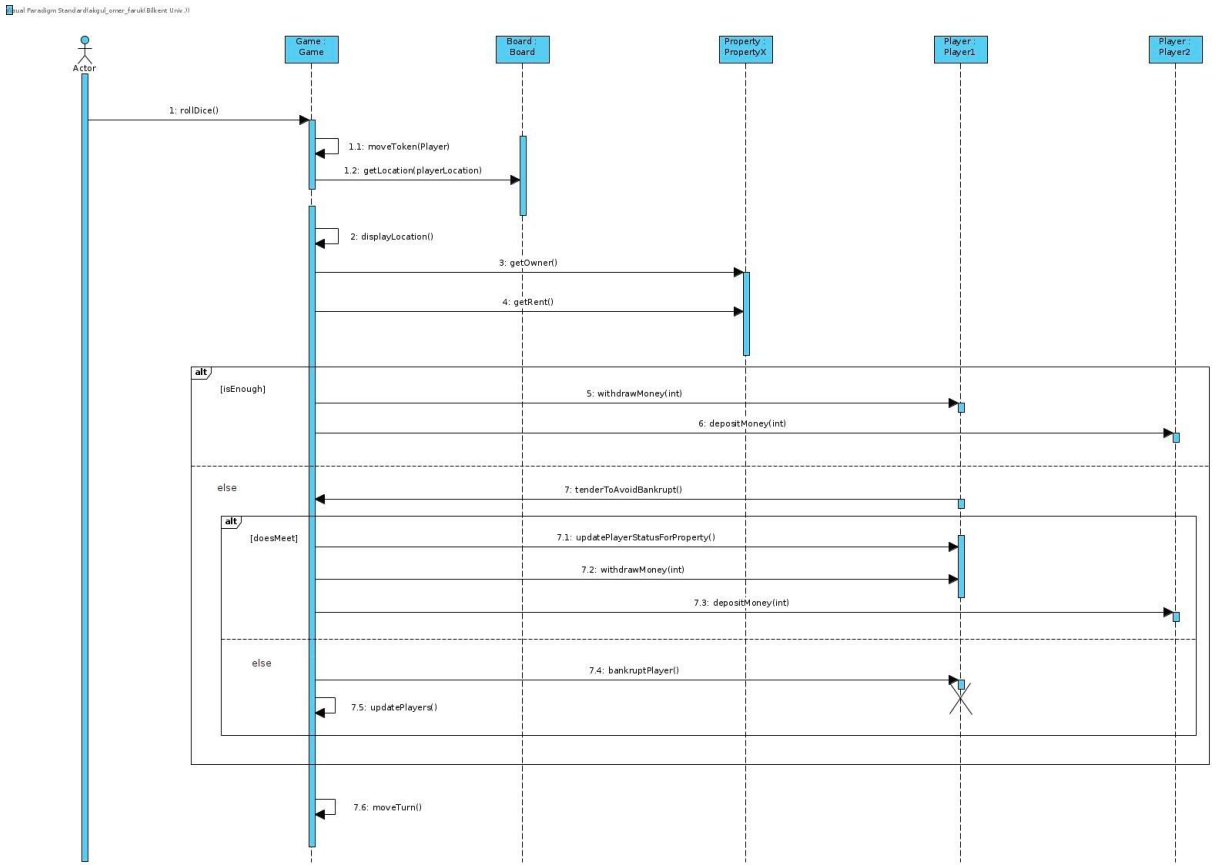


Figure 4

As in the previous scenario, the interaction starts when the actor rolls the dice. However, this time, the location is owned by another user. Thus, the current player has to pay a rent to the owner. After learning the owner and rent amount, Game withdraws the rent from the current player and deposits to the account of the owner. Nevertheless, the current balance of the player might be less than the rent. In such a scenario, the tender process should be initialized by the player to avoid bankruptcy. If the player earns enough money to pay the rent, the game will move on, but if the player still cannot pay, then the player is bankrupt.

5.2.4. Chance / Community Chest Cards

The following diagram aims to visualize the interaction of objects, when the player comes to the chance block. It could be integrated with the diagram above but, instead, drawing a separate diagram is preferred to increase the understandability of diagrams. The diagram is approximately the same for community chest block except the payment address.

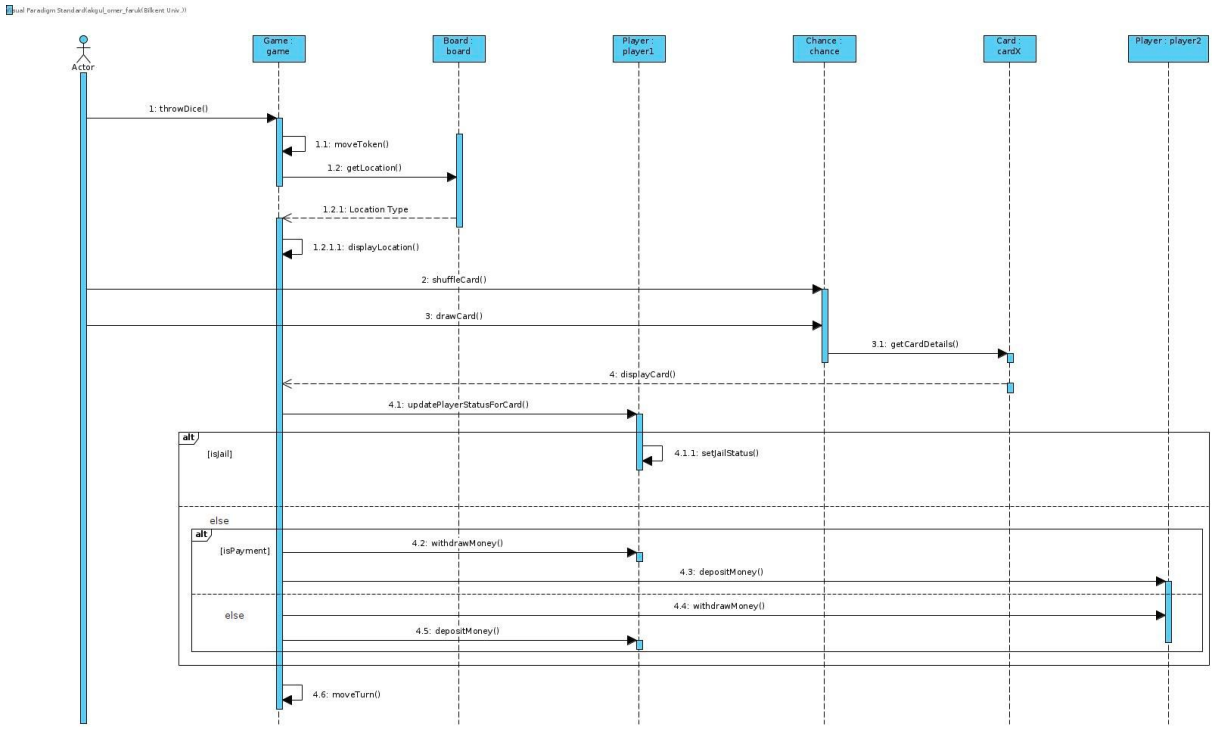


Figure 5

When the player comes to the Chance location, the following process is performed. The process is valid for the Community Chest, as well. The player shuffles the cards and draws one of them. Chosen card object is displayed on the screen. Then, the game object updated the status of the player according to the card drawn. For instance, if the card says “Got to Jail for 3 Rounds”, the jail status of the player is updated accordingly. If the card says “Take money from all the other players”, the game object withdraws money from other users and deposits it to the current user.

5.2.5. Reverse Game Mode

When one of the players comes to the reverse block of the game, game direction changes as shown in the diagram below.

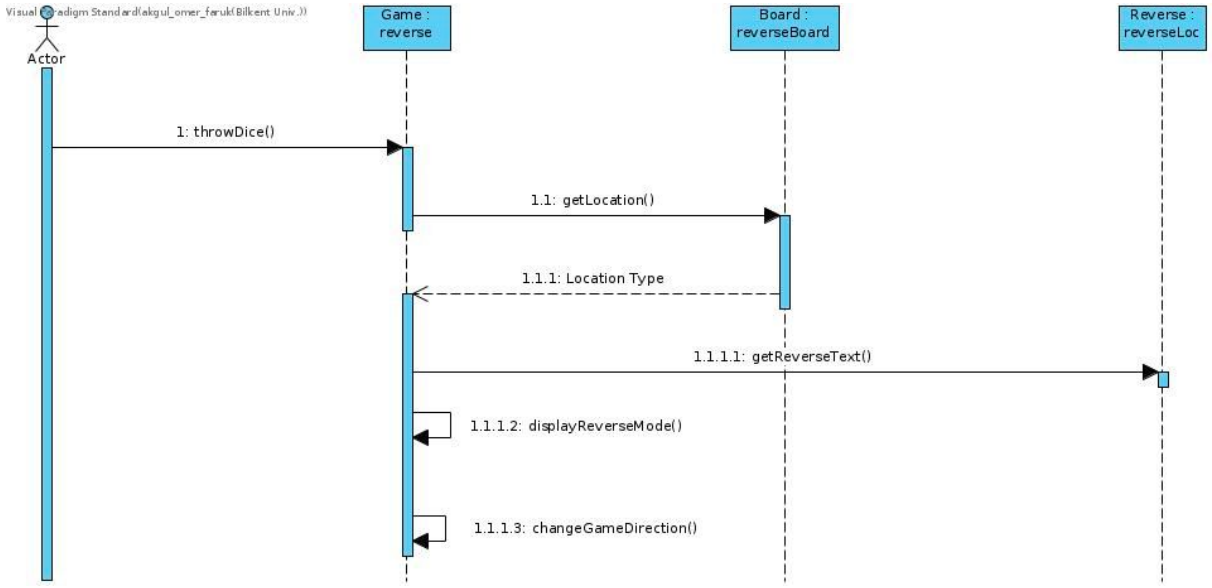
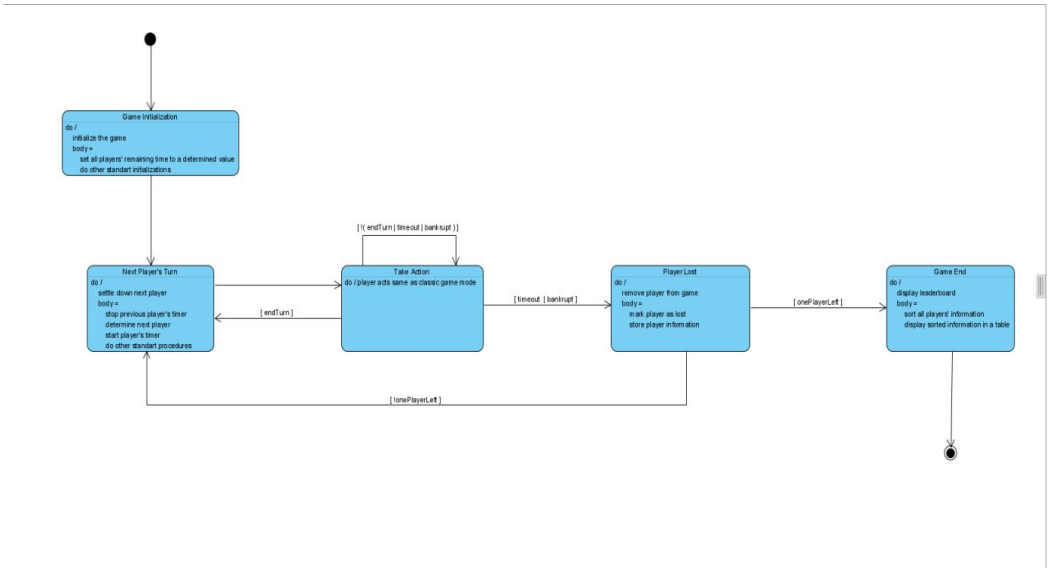


Figure 6

When the player comes to the reverse block and the direction is changed, all the players start to move back until one of the players comes to the reverse block again.

5.2.6. Blitz Game Mode

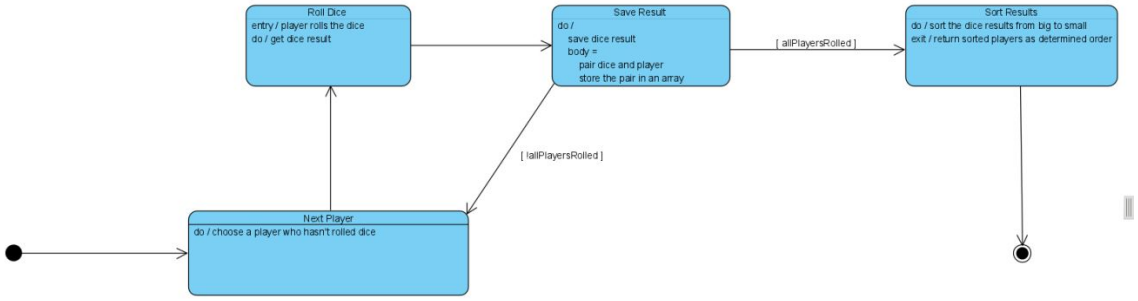
The following state diagram represents Blitz mode. Conceptually, Blitz mode is inspired from the Blitz Chess. In this diagram, the main purpose is to display the differences of Blitz mode from Classic mode. The activities that a player can do in a turn is the same as a Classic game. Therefore, a turn of a player is abstracted in the diagram.



Initialization of the Blitz mode is nearly the same as Classic mode. Additionally, each player is given some amount of total time. When the turn comes to a player, this player's remaining time starts decreasing. During his/her turn, the player can take action just like the Classic game (i.e. rolls dice, moves the token and acts accordingly). After the current player decides to end the turn, the current player's timer stops and the next player's turn and timer starts. Players lose the game if they go bankrupt or their time ends. When a player loses, his/her current wealth (determined from money and owned locations) is saved. The game continues until one player remains unlost. When there is only one player left in the game, there is no point in playing. So, the game ends and a leaderboard prepared from players' wealth is displayed. Note that a wealthy player whose time is up at an early stage of the game can still win the game.

5.2.7. Order Determination

Before starting a game, a turn order which decides who comes next after the current player ends the turn must be determined.



To determine the turn order, each player rolls dice for once and dice results are saved. After rolling step finishes, the results are sorted. Those sorted results represent the determined order. During the game, the player who has the highest dice result starts first and the player with the lowest dice result comes last.

5.3. Object and Class Model

Class Diagram

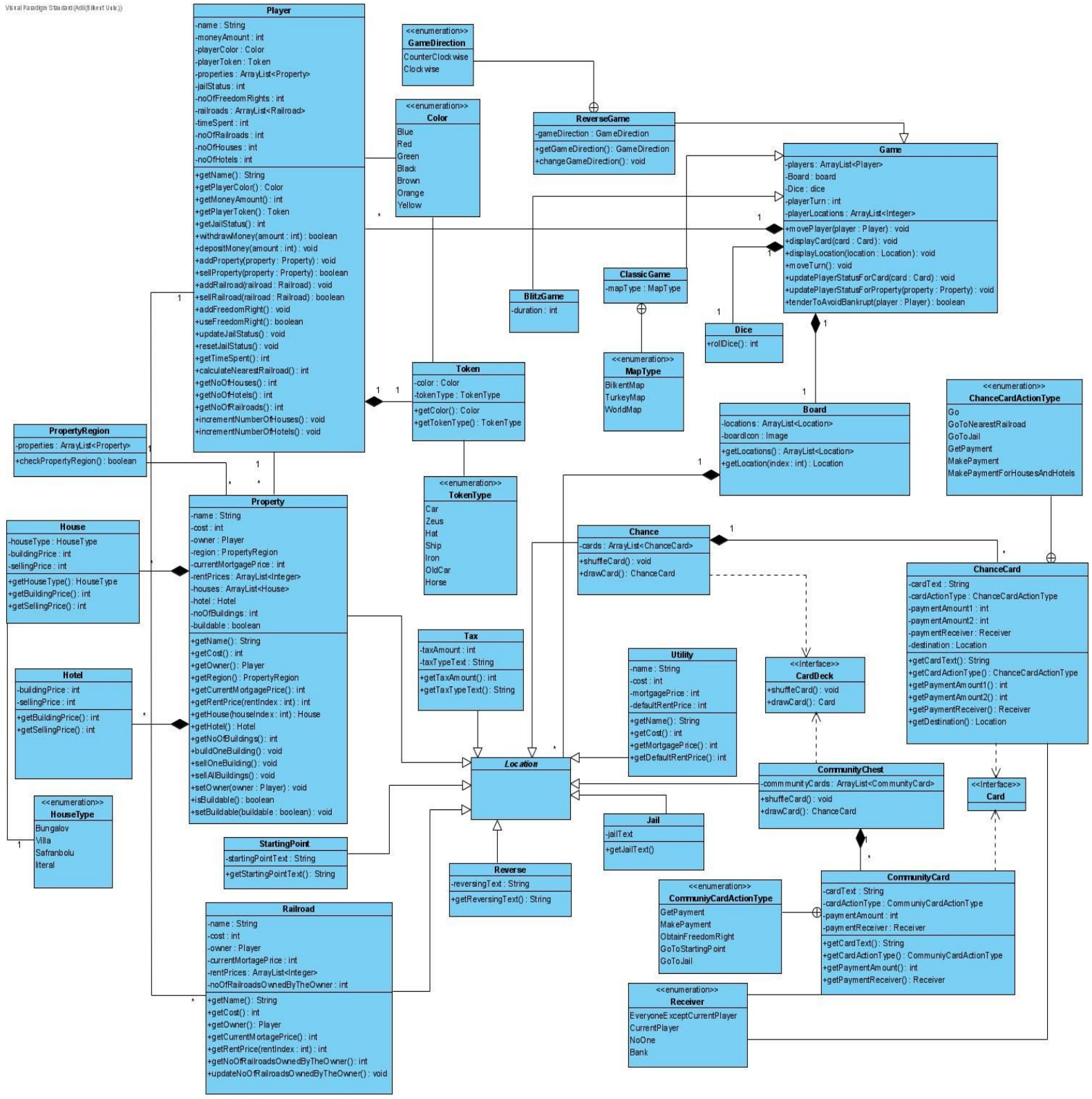


Figure 7

Class Descriptions:

The class diagram of the “Monopoly” game is stated at the Figure above. The “Monopoly” game has 24 classes, 7 enumerations and 2 interfaces.

Classes:

The main class of the Monopoly game is the Game class. Game class is responsible for the functionalities of the game. It contains players, dice, board and player’s locations. The game class has 3 subclasses which represents the different Game modes.

ClassicGame:

ClassGame class is the first subclass of the Game class which is responsible for the classic game mode. It contains 3 existing constant maps which are BilkentMap, TurkeyMap and WorldMap.

BlitzGame:

BlitzGame class is the second subclass of the Game class which is responsible for the blitz mode. It has duration property different from its parent class, it specifies the total duration for the game.

ReverseGame:

ReverseGame class is the third subclass of the Game class which is responsible for the reverse mode. It has a gameDirection property different from its parent class. Also, it contains inner enumeration which specifies the game direction of the class. The GameDirection enumeration can only take two values that specify the game direction which are CounterClockWise and ClockWise.

Player:

As can be deducted from its name Player class is responsible for players and their current status. All players, therefore, player instances have different colors and tokens. This class contains all game instances owned by the player and related with the player. Player’s money, properties, railroads, freedom rights are kept by this class.

Dice:

Dice class is defined for the dice. In order to move, players should roll the dice and they may move through the board with respect to the facing values of the dice. The rollDice method returns an integer number between 2 and 12 inclusively.

Board:

Board class is responsible for the game board which means the construction of the game map and orders of the locations. It contains a location array. The first (0) index of the location array represents the starting point and last index represents the ending point of the board(map).

Token:

This class is used to visually represent current players. The tokens are colored with respect to the player's color. And players may choose one of the provided tokens by the game.

Location:

Location class is an abstract class used to group all locations. It has 9 sub-classes which are the different kinds of locations on the board and the game. The location abstract class does not contain any properties or methods. Board class uses arrays that consist of location classes.

Property:

Property class is a subclass of the location class. It is responsible for the properties that exist in the game. It contains all information about the properties such as its name, price, renting price and current owner.

PropertyRegion:

PropertyRegion class is responsible for the property regions. It keeps the properties with the same region. It is used to check the availability of building houses/hotels and updating the rent prices when all properties that belong to the same region are owned by the same player.

Railroad:

Railroad class is a subclass of the location class, hence, it is a location. It is responsible for the railroads that exist in the game. It contains all information about the railroads such as its price, rent price. Since the rent price of the railroads change according to the number of railroads owned by its owner, it keeps the number of railroads owned by its owner.

Tax:

Tax class is a location. It is responsible for the tax locations that exist in the game. There are several tax locations with different back stories and amounts. Therefore, it keeps the text which represents the back story, and the amount the player should pay.

Utility:

Utility class is a location. It is responsible for the utilities in the game. There are several utilities with different names and amounts. Similar to the property and railroad class, this class contains all information about the utility.

Jail:

Jail class is a location. It is responsible for the jail areas in the game. At default maps, there is only one jail location where players go when they are imprisoned.

StartingPoint:

StartingPoint class is a location. It is the first location and starting location in the game. When users pass or stop at the starting point, they receive money.

Reverse:

Reverse class is a location that exists only in the reverse mode. When a player comes to the reverse location, the direction of the game changes.

Chance:

Chance class is a location and card deck that exists in the classic mode and the blitz mode. This class implements the CardDeck interface. When a player comes to the chance location, the player draws a chance card from the deck and the related actions are made.

CommunityChest:

CommunityChest class is a location and card deck that exists in the classic mode and the blitz mode. This class implements the CardDeck interface. When a player comes to the community chest location, the player draws a community card from the deck and the related actions are made.

ChanceCard:

ChanceCard class is a card because it implements Card interface. It contains specific information about an action which will be made. It keeps the payment amount, destination, payment receiver and card action type which is an inner enumeration that is defined in the ChanceCard class. According to the card action type, the game decides the type of action and makes the action according to the properties which are payment amount, destination and payment receiver.

CommunityCard:

CommunityCard class is a card because it implements Card interface. It contains specific information about an action which will be made. It keeps the payment amount, payment receiver and card action type which is an inner enumeration that is defined in the CommunityCard class. According to the card action type, the game decides the type of action and makes the action according to the properties which are payment amount and payment receiver.

FreeParkingLot:

FreeParkingLot class is a location which is responsible for the free parking lot area. When a player comes to this point, the player takes all of the money in the middle

5.4. User Interface

5.4.1. Main Menu

When the user gets into the game, they will see the main menu first. From this screen, the user can select one of the following options they are presented with.

1. **Play**

   This will take the user to the further options of the game mode, and the player count.
2. **Mode Editor**

   This button will take the user to the mode editor.
3. **Options**

   When users click onto the **Options** button, they will go to the **Options** menu.
4. **How to Play**

   This button will open up the **How to Play** instructions.
5. **Credits**

   This button will take users to the **Credits** screen.
6. **Quit**

   The user closes the software and **quits** the game.

Figure 8: Main Menu Screen

5.4.2. Play Game

When the user clicks the **Play Game** button in the **Main Menu**, this screen will come up. In this screen, the user can specify **the player count**, and **the game mode** from the drop down menu right below the name. There will be default placeholders for **the player count**, and **the game mode** fields in their own respective drop down menus. The user may choose the **Back** button to navigate to the **Main Menu,** and choose the **New Game** button to specify the game properties further in the **Player Specification** screen. The **Load Game** button will take the user to the **Load Game** screen.

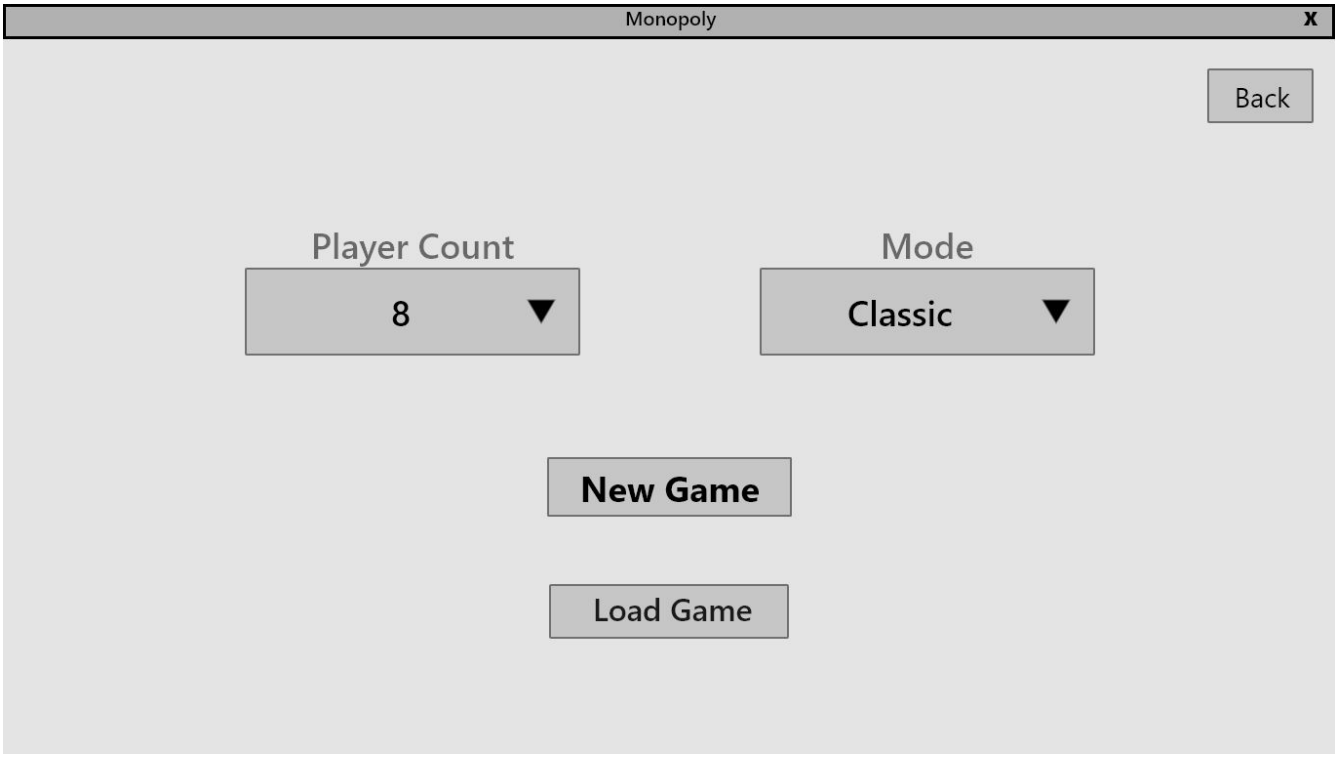


Figure 9: Play Game Screen

5.4.3. Load Game

In this screen, the user can select a previously saved game to continue playing. The saved games have 4 main properties. Name, mode, save time, and time elapsed. The user can delete the saved game by clicking on the **X** button or can continue to play the game by clicking at the green horizontal triangle next to the **X** button. The user may choose to navigate to the **Play Game** screen by clicking the **Back** button.

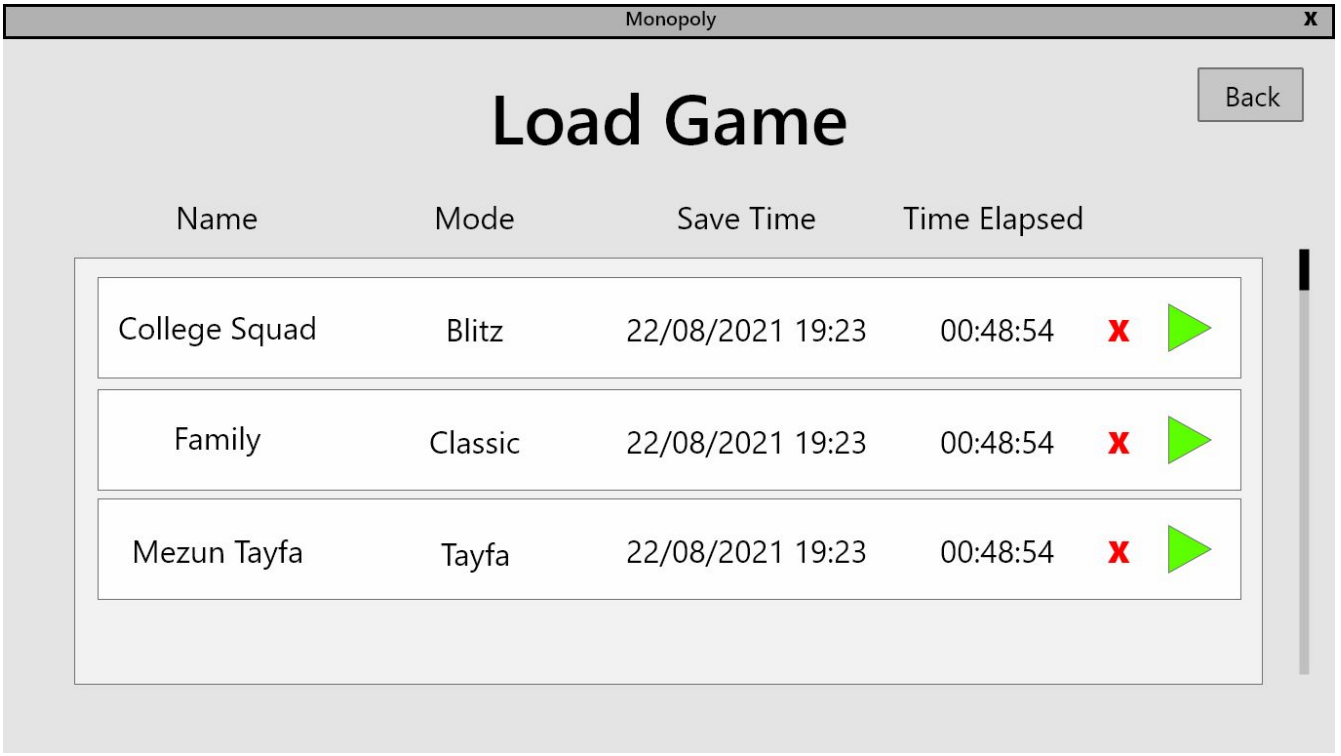


Figure 10: Load Game Screen

5.4.4. Player Specialization

In this screen, the user can specify the names of each player, choose their tokens, and confirm their changes. In this screen, it can be seen that all of the players are not ready, they have not confirmed their names or tokens. In this case, the user cannot **start** the game. When all of the players are ready, the user can start the game. The **start** button navigates to the **in-game** screen. The **back** button navigates to the **Play Game** screen.

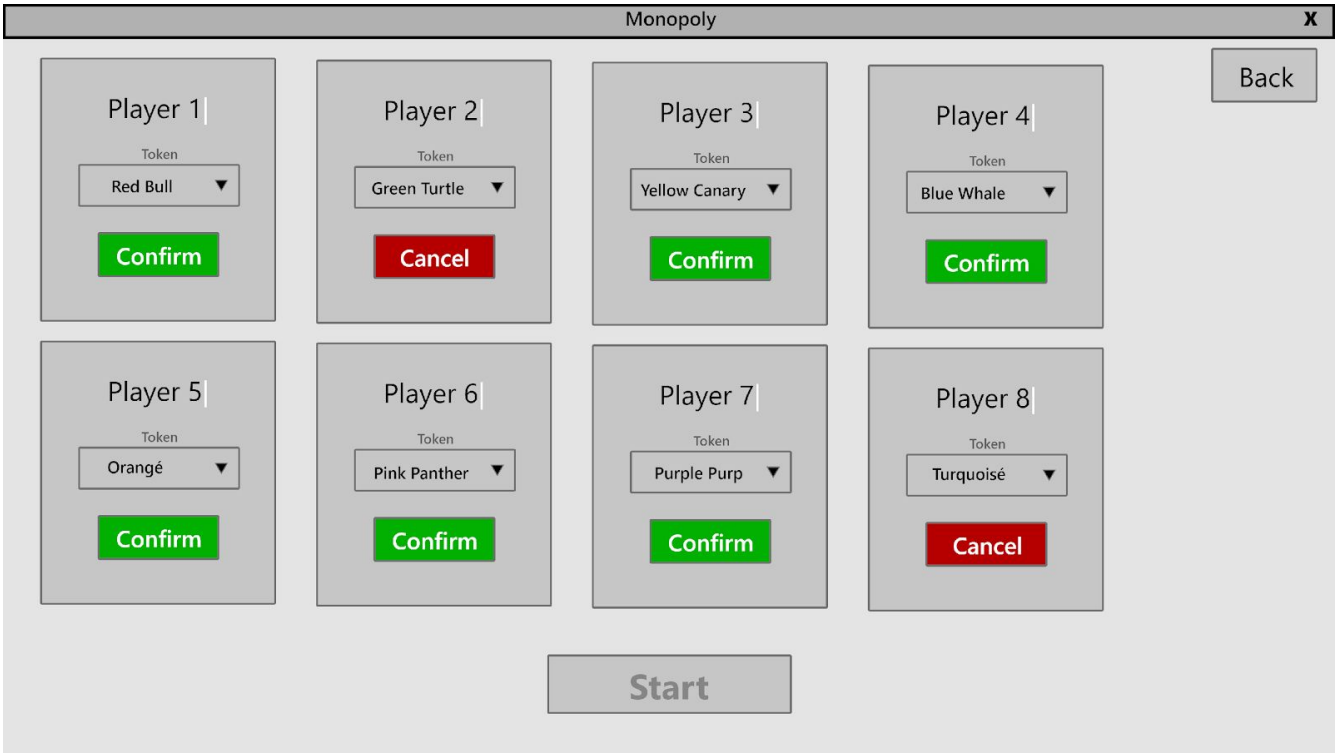


Figure 11: Player Specialization Screen

5.4.5. In-Game Screen

In this screen, the players can see the table, their cash value, who owns the cities, which cities are in the same group and so forth. However, **one must not forget that** this is only a **mock-up**, therefore, this visualizations are not exactly the implementation wire frames, and it is simplified to give the basic understanding of the implementation.

Each player gets a unique color with the tokens they have chosen and it forms the color of their name and cash value. There appears a giant sideways black triangle next to the player whose turn is on. On the right of the screen, there is the pause button, which opens a dialog to close the game, and perhaps some other features we might implement. The center colors of the properties on the table show which property group they belong to. The upper blocks on each of the properties show the owner of the property and have the same color as the owner's token color. Tokens are shown as little disks at the left lower corners of properties.

As we are implementing this game as a multiplayer game on a single local computer, we cannot control who is taking actions, we only assume. The user can roll the two dice and

then continue to take actions such as buying houses. The user must click **end of turn** to end his turn and pass the turn to the next player. The user can make further arrangements as long as he/she does not roll the two dice. The parallelogram on the upper side of the roll button is the **chance** card holder and on the lower side of the roll button is the **community chest** card holder. The places on the board's side are the **chance** and the **community chest** places respective to their symbols. The **tax** place is the place where newcomers pay tax.

In this mock-up, there are only 8 places on each side of the board, however, there will be 10 when we implement the full game. We will further discuss this in the design reports. When the user clicks on a property placed on the board, he/she can access the information about the property, and can take access if he/she has the proper authorization. This will be shown in the **In-Game – Property Screen**. By clicking at the pause button at the center right, the user will navigate to the **In-Game Pause Screen.**

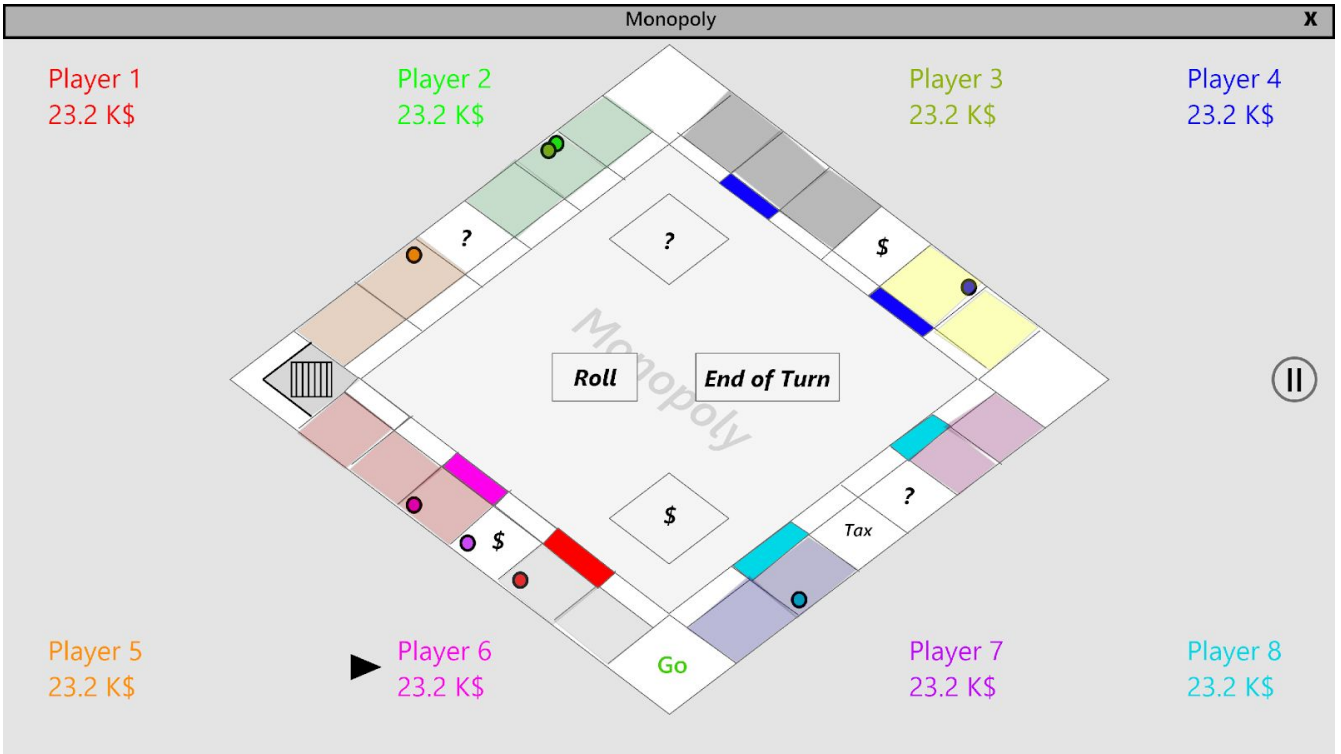


Figure 12: In-Game Screen of the Classic Mode

5.4.6. In-Game – Property Screen

The place's name is on the top row of the property image. The background color of the place name is the color of the owner’s token. The cost of the property, the rent incomes in each different building states, mortgage values and the building costs are shown in this screen. As now the turn is on the **Player 6,** the player can mortgage the property, sell the property, or build buildings on the property.

The **mortgage** option does not require any further confirmation, however, in the **sell** option the buying player must further confirm the transaction. They can also get out of the property screen by clicking the **X** button on the top right of the property card.

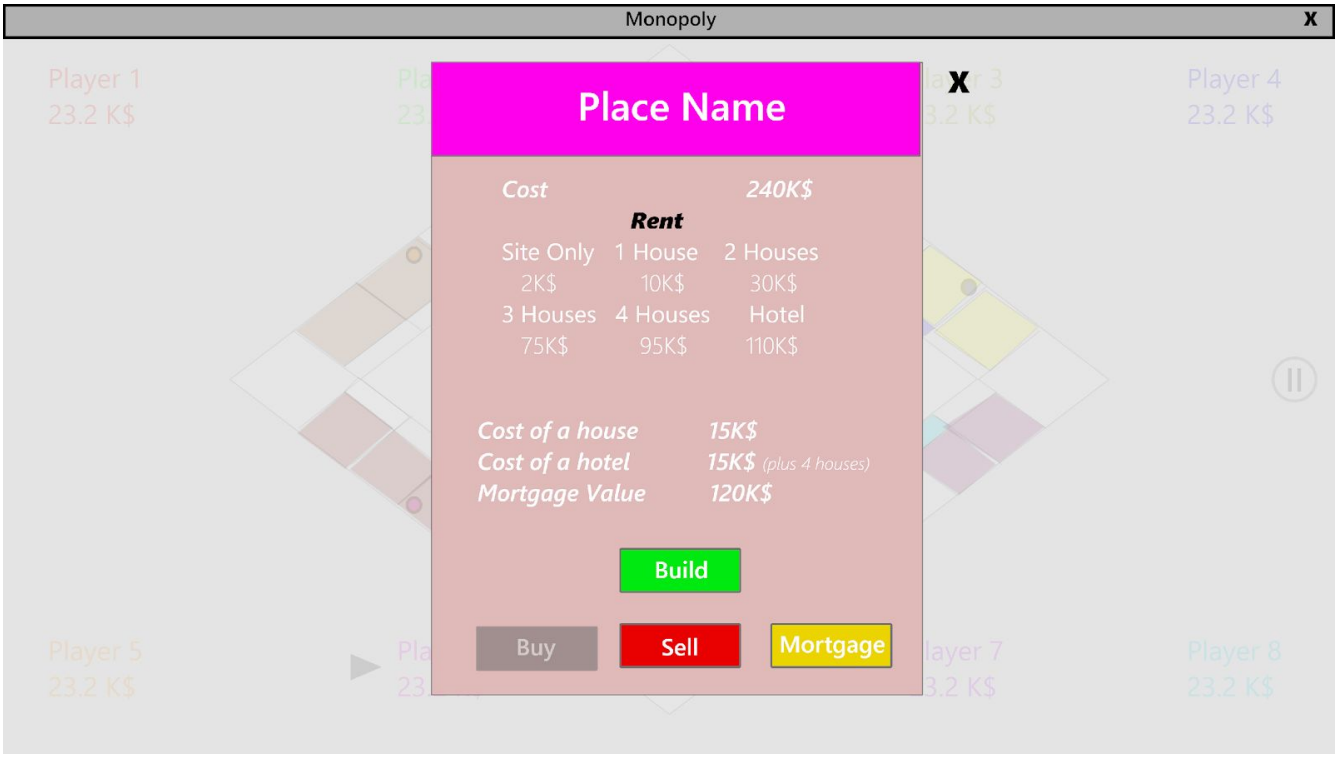


Figure 13: In-Game - Property Screen

5.4.7. In-Game – Build Screen

As there are not any buildings on the current property, **Player 6** can only build a house. After the build **Player 6** can close the build screen by clicking the **X** button on top right of the property card.

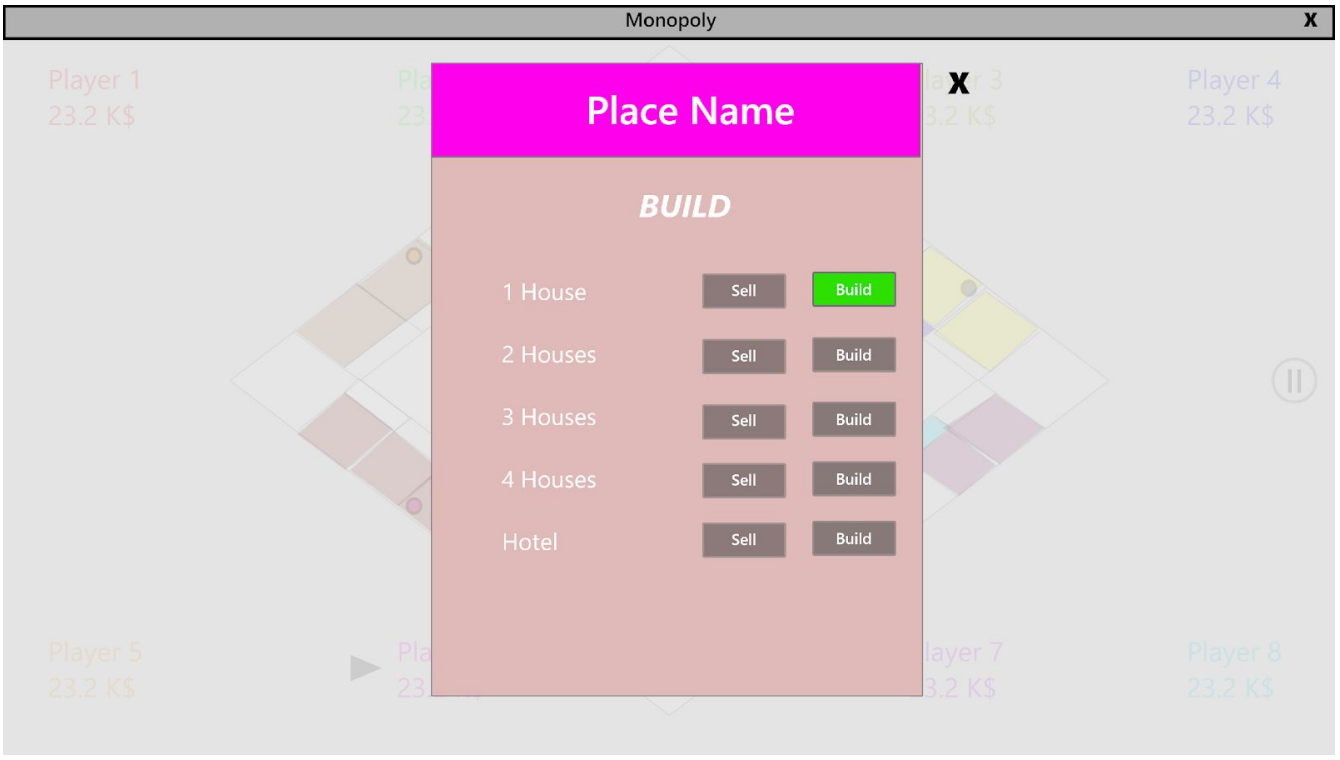


Figure 14: In-Game - Build Screen

5.4.8. In-Game Pause Screen

By clicking at the pause button on the **In-Game** screen, the user will navigate to this screen. In this screen the user can save the game, continue playing, or quit the game without saving.

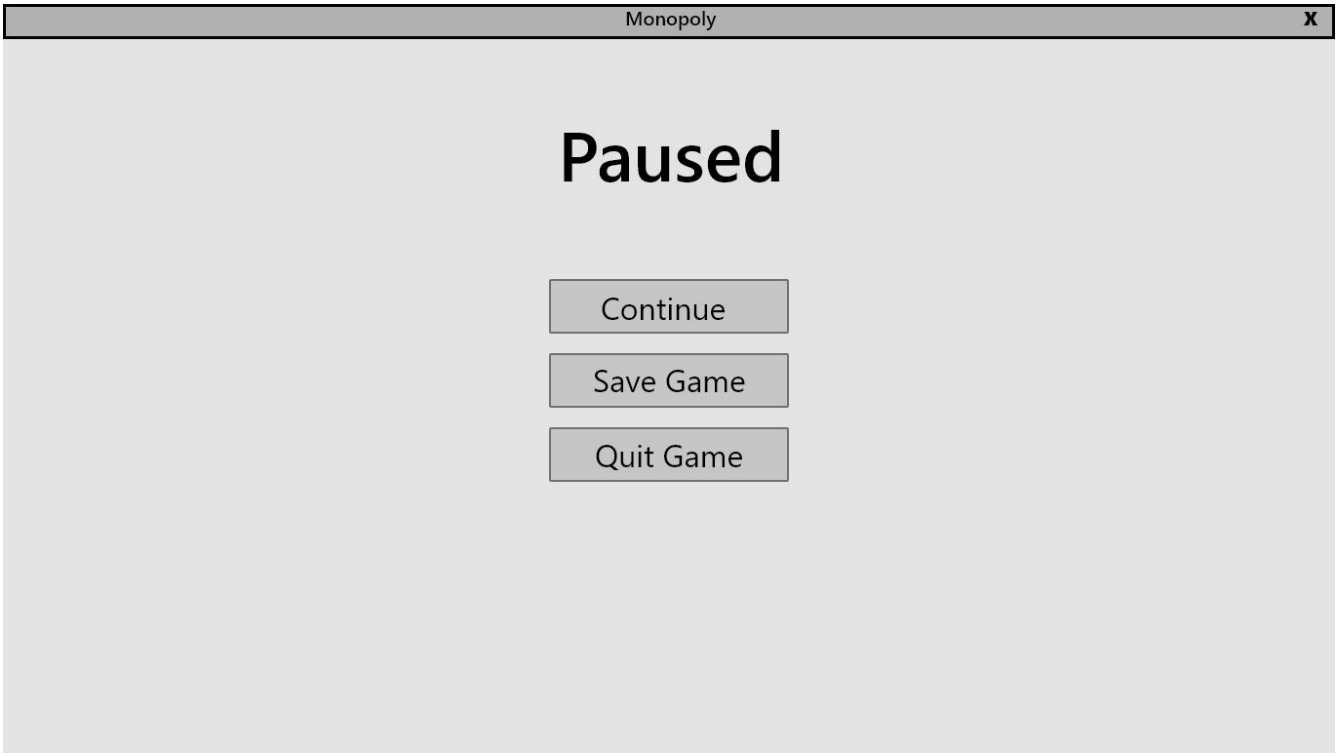


Figure 15

5.4.9. Options Screen

In this screen, the user can adjust the volumes of both the music and the SFX. The **save** button saves the current info from the sliders, and the **back** button returns to the **main menu** screen.

Figure 16: Options Screen

5.4.10. Credits Screen

In this screen, the user can see the credits of the **Monopoly** game. The **back** button takes the user back to the **Main Menu** screen.

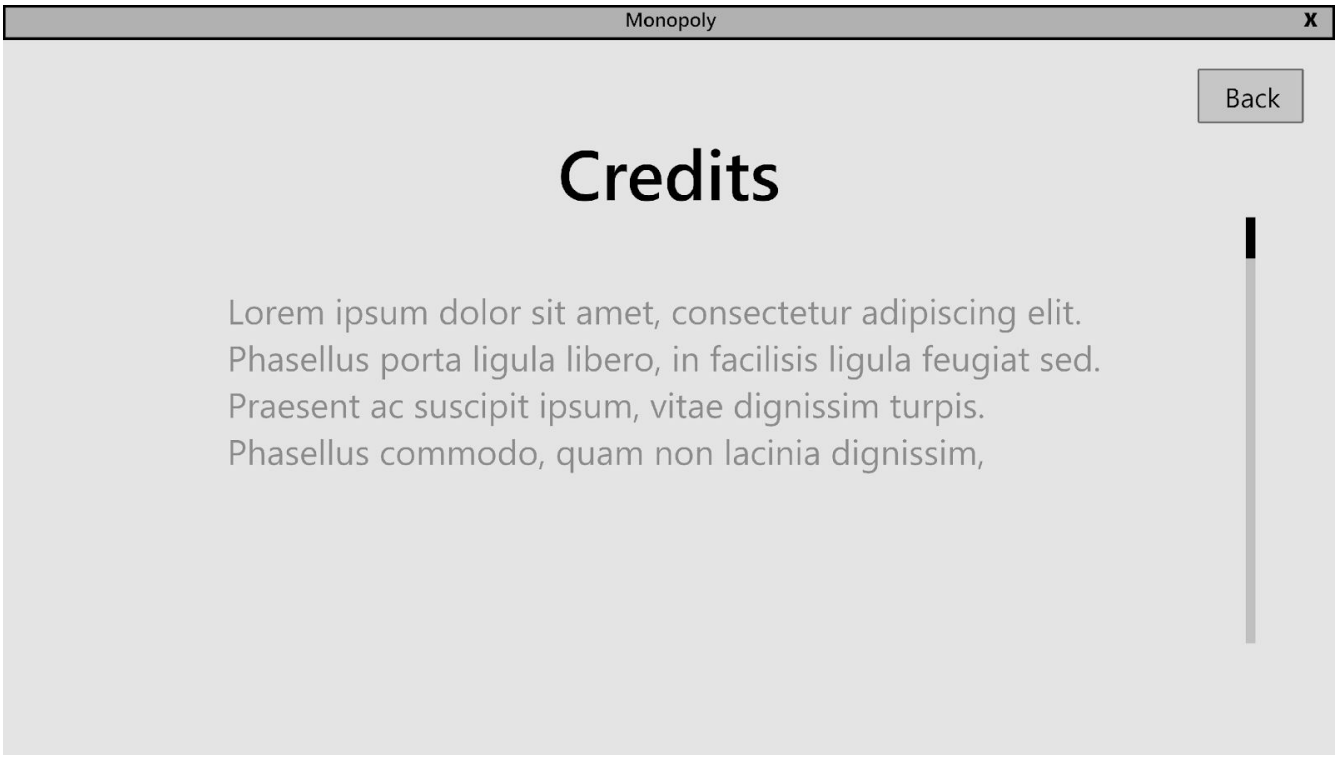


Figure 17: Credits Screen

5.4.11. How to Play Screen

In this screen, the user can see the rules, the general information about the objects in the game and introductions to the different tools. The **back** button takes the user back to the **Main Menu** screen.

Figure 18: How to Play Screen

5.4.12. Mode Editor Screen

The user can specify a name for the previously created mode and can select a base mode to work on. The **Edit Mode** button takes the user to the **Mode Editor – Specialization** screen. The back button takes the user back to the **Main Menu.**

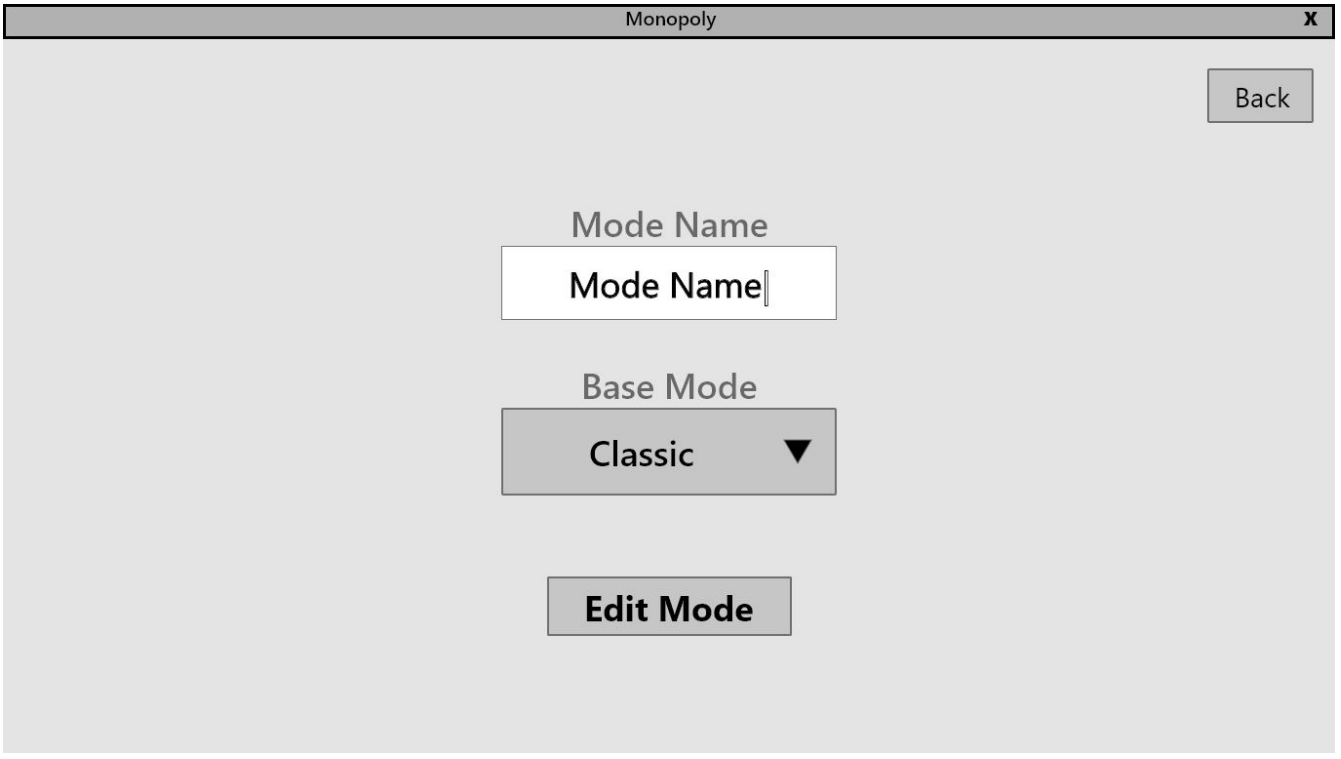


Figure 19: Mode Editor

5.4.13. Mode Editor – Specialization Screen

By choosing different base modes, the user gets more permission to specify the game in the way that he/she wants. This in-game tool enables users to edit the given six modes. Then, the default game board of this mode will appear. On this board, users will be able to modify each location's name and money related numeric values. For example, users may edit a property's name, rent, price and building costs.

As it is seen in Figure 15, the user can specify only salary and start capital in the **Classic** mode, but the user can specify the blitz period in the **Blitz** mode as it can be seen in the Figure 16, and also they can change the tax calculation in the tax mode. However, users cannot change the functionality of a location. For example, they can't convert a property location to a Railway or a Jail, or they can't change the three turn waiting time in Jail.

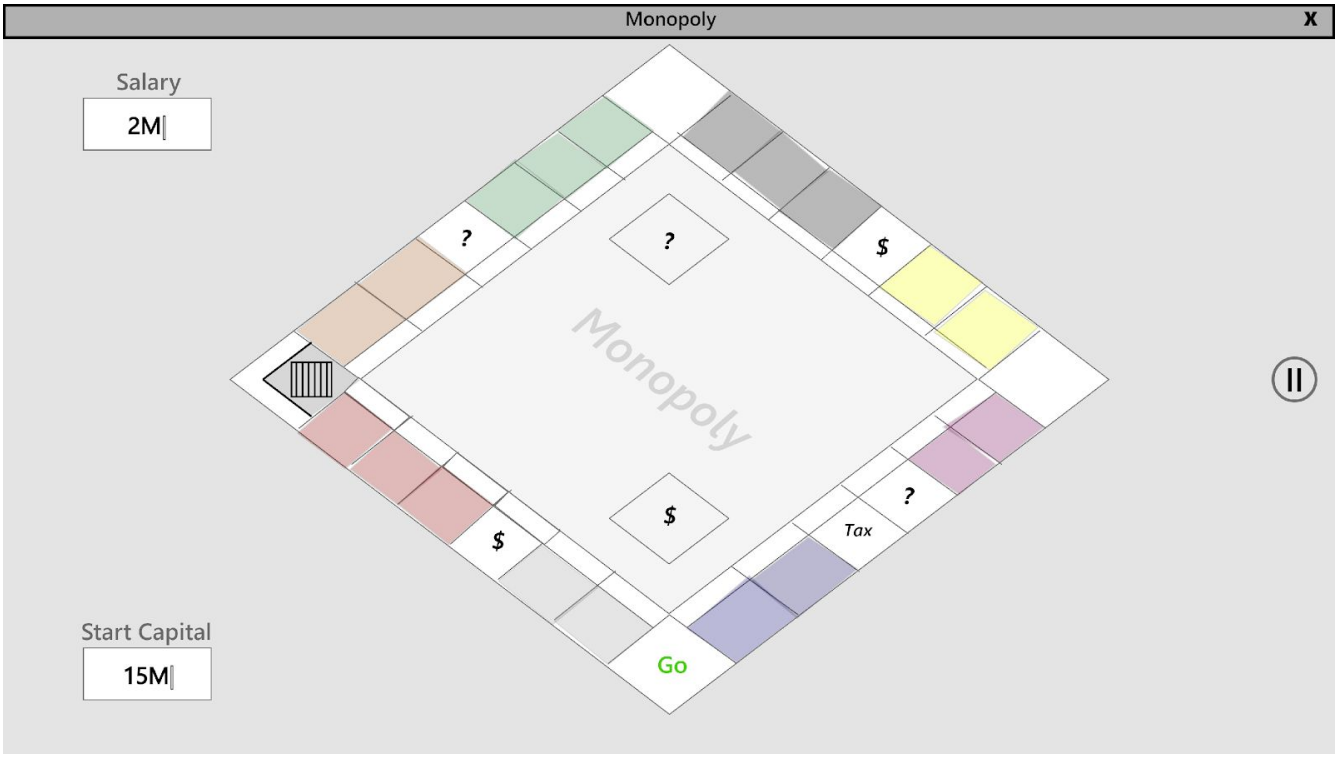


Figure 20: Mode Editor – Specialization – Classic Base Mode

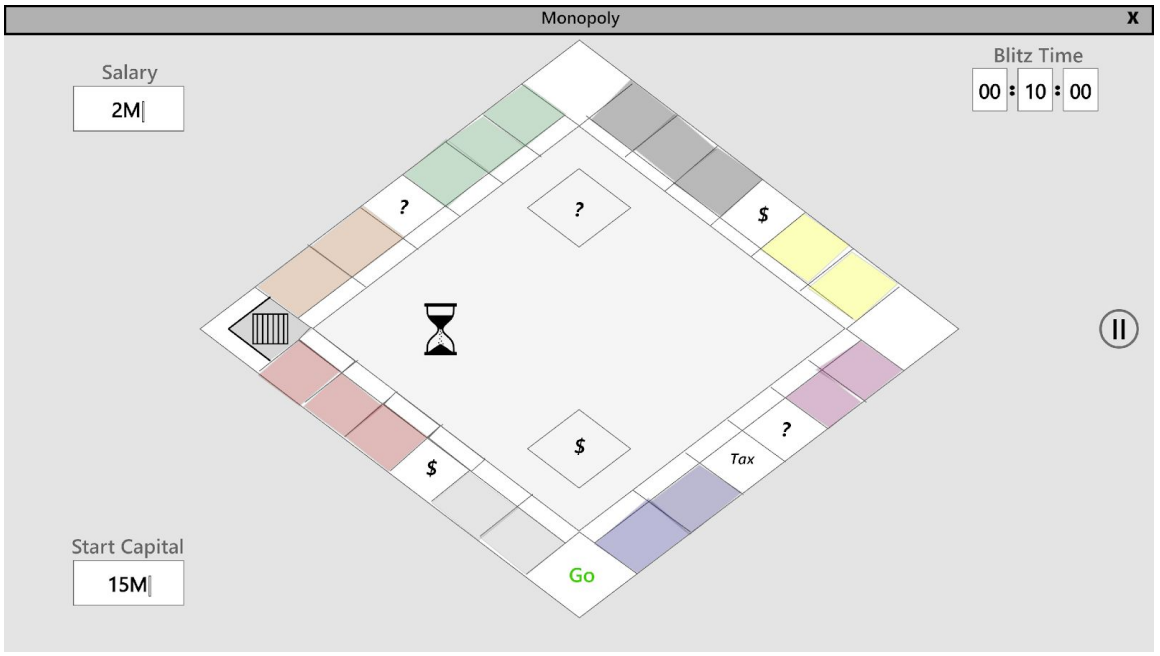


Figure 21: Mode Editor – Specialization – Blitz Base Mode

By clicking on the can-be-specialized properties, the user can change the monetary values, as it opens the property focus screen with text fields. That way the users can interact with the property

values without any further knowledge. By clicking at the **Cancel** button, the user will discard any changes that he/she has made. The **Save** button saves the current status of the property.

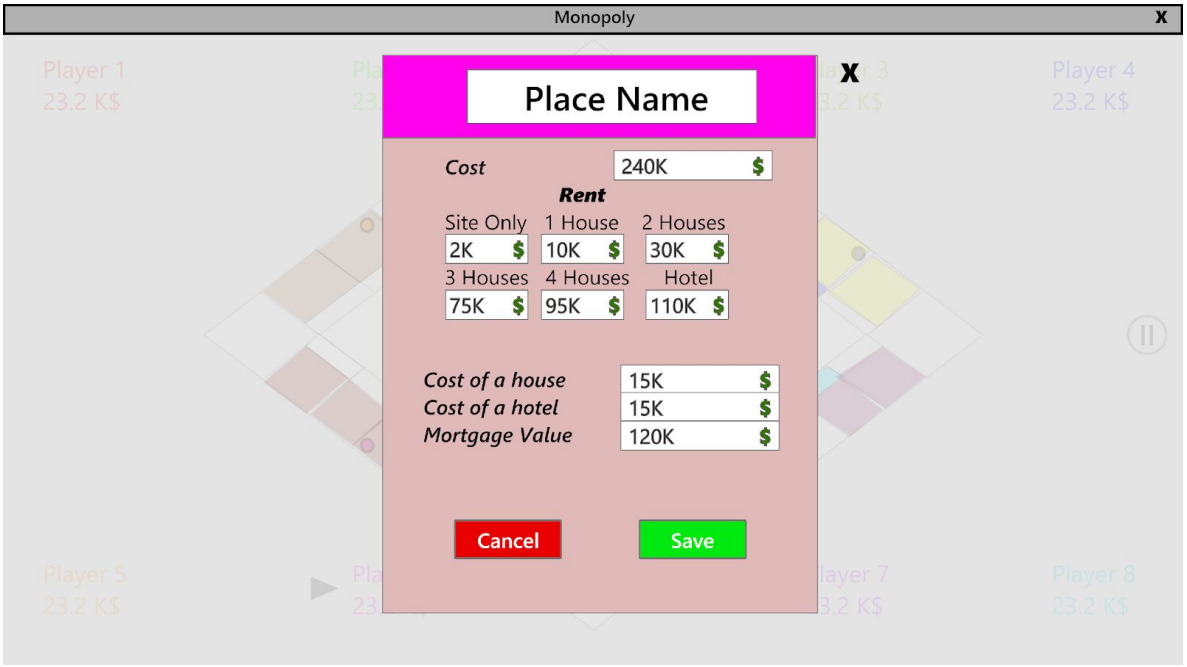


Figure 22: Mode Editor – Property Focus

The pause screen of the mode editor is as below. The user can save the mode or quit the editor. The pause screen is navigated through the pause button on the **Mode Editor – Specialization** screen.

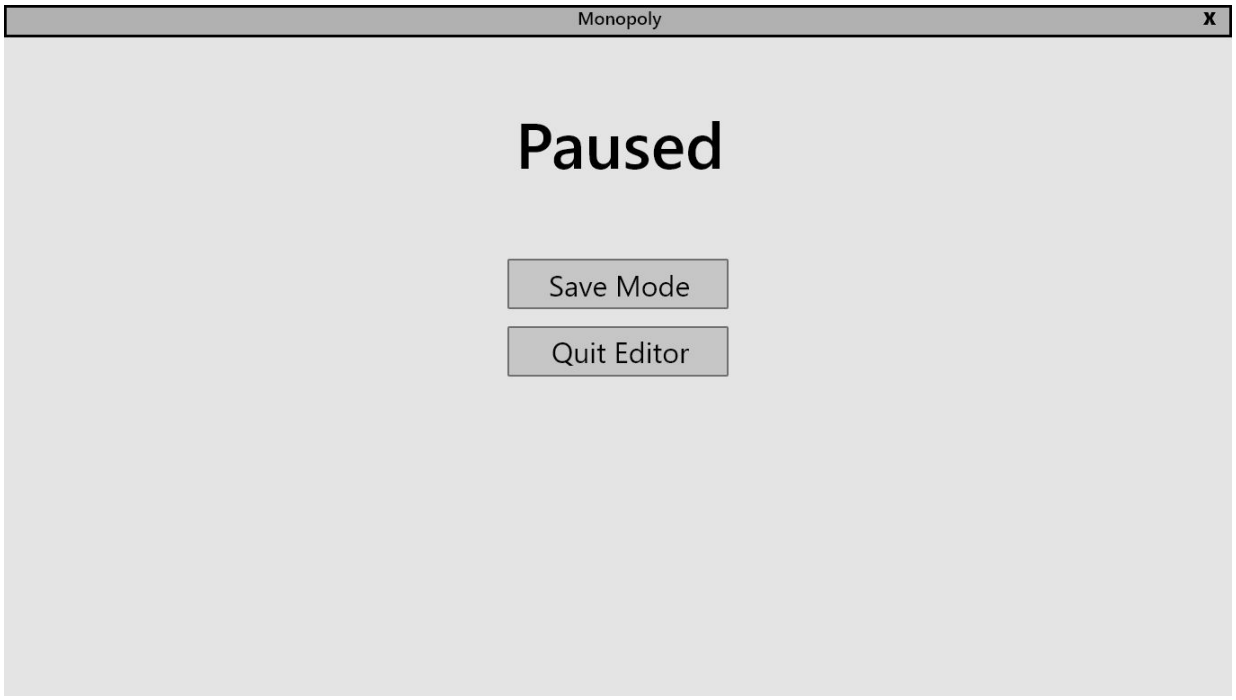


Figure 23: Mode Editor – Pause Focus

5.4.13. Different Mode Mock-ups

Below are some of the modes we are implementing in our term project. These are **Blitz**, **Classic**, and **Reverse**. The mock-ups except the **Classic** mode are almost identical to the **Classic** modes mock-up. However, there are certain visual differences that appear on the screen, as they are different modes indeed and we would like to remind that to the players.

For the **Blitz** mode, each user can see their own remaining time right over their player names or under their cash values

In the **Reverse** mode, there are two reverse icons in the middle of the board, pointing clockwise and counter-clockwise respectively with different colors to indicate the current flow direction of the game. Also, there are reverse icons on some of the places, on which the users change the flow direction of the game.

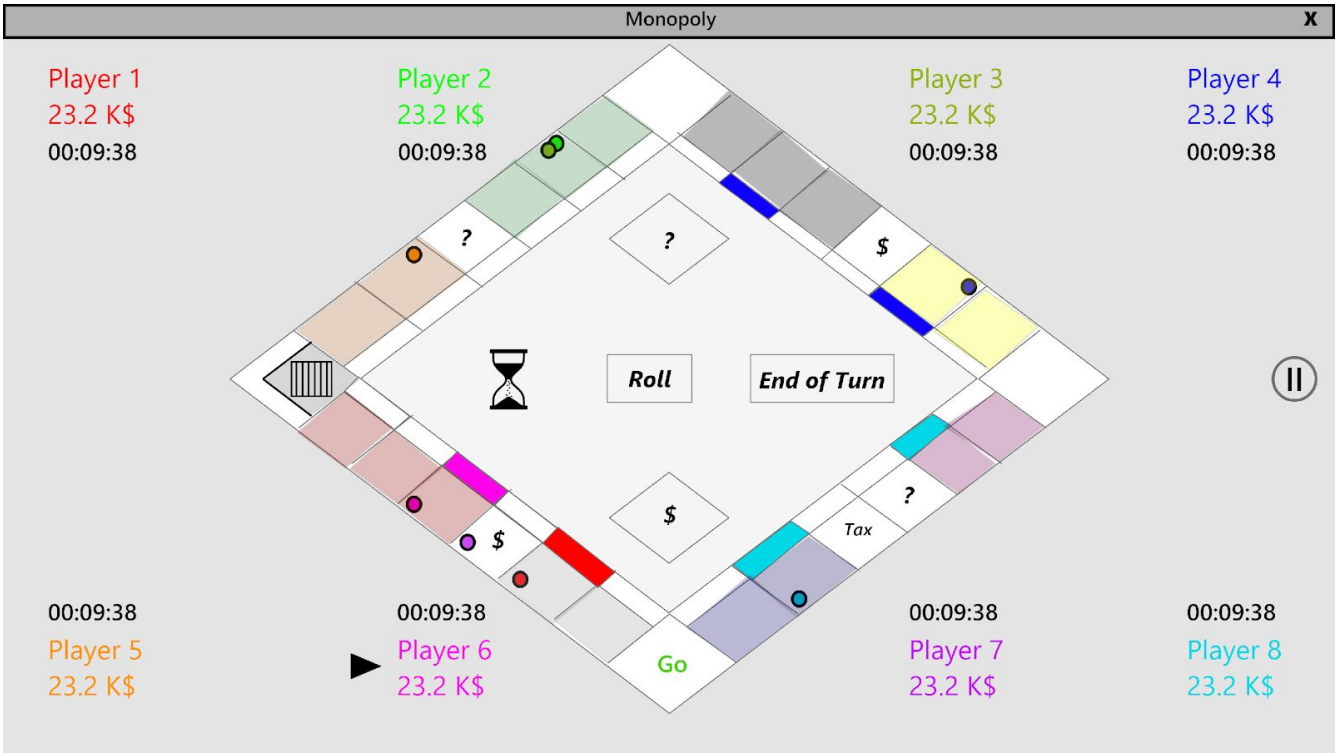


Figure 24: Blitz Mode

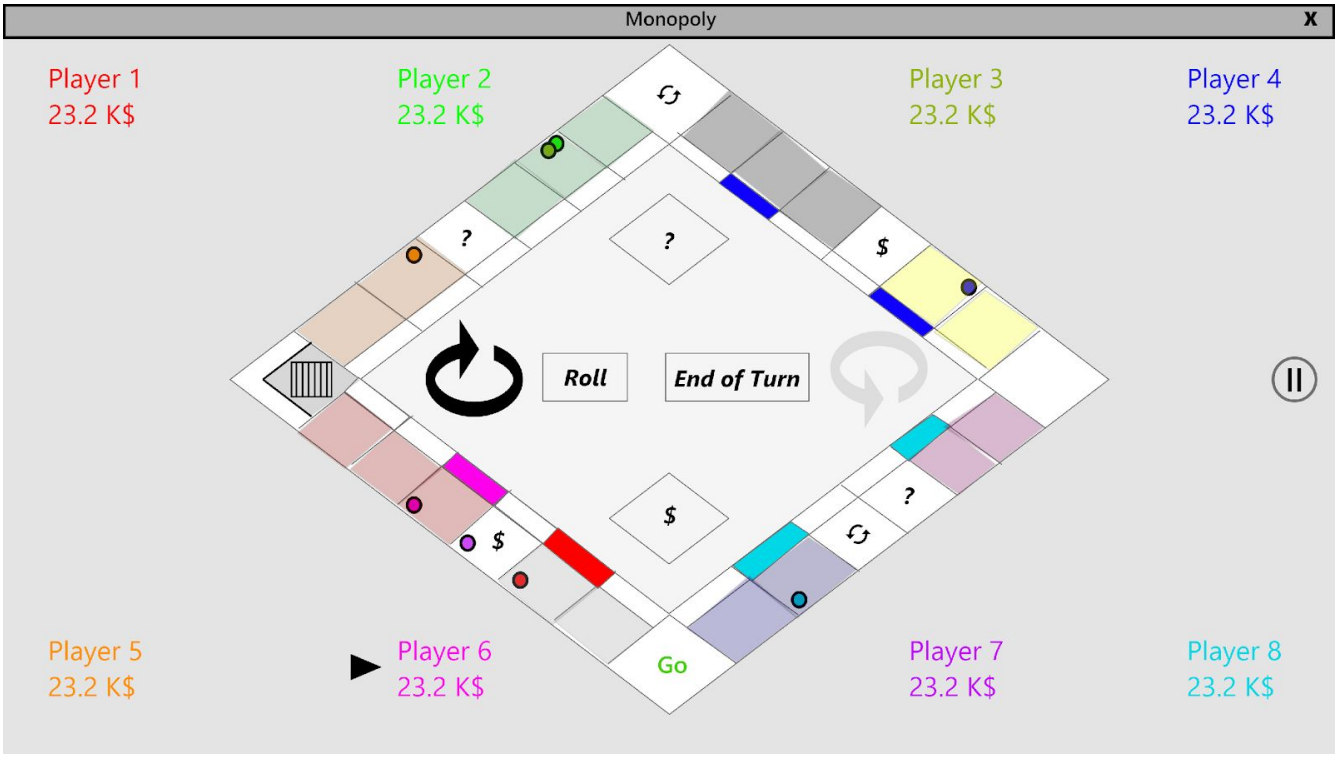


Figure 25: Reverse Mode

6. GLOSSARY & REFERENCES

[1] Monopoly.fandom.com. 2020. *Monopoly Wiki*. [online] Available at: <https://monopoly.fandom.com/wiki/Main_Page> [Accessed 23 October 2020].