



CS 319 Project Analysis Report Iteration 2

Monopoly Sicilia

Group 2D

Asya Doğa Özer - 21803479

Ufuk Palpas - 21702958

Uğur Utku Seyfeli - 21802239

Yiğit Harun - 21803241

Oğuz Orhun Tüzgen - 21802313

Table of Contents

1.Introduction

2.Overview

2.1. Characters

2.2. Mafia

2.3. Police

2.4. Forex

2.5. Building shortages

2.6. Set your own limit

2.7. Power-ups

2.8. Monopoly Classic Features

3. Functional Requirements

3.1. Play game

3.2. How to Play

3.3. Credits

3.4. Quit Game

4. Nonfunctional Requirements

4.1. Performance

4.2. Usability

4.3. Reliability

4.4. Supportability

5. System Models

5.1. Use Case Model

5.2. Dynamic Models

5.3. Object and Class Model

5.4. User Interface

6. Improvement Summary

7. Glossary & References

1. INTRODUCTION

Monopoly is a classical board game that ages back to 1935. This game represents a journey and rivalry of landowners. The game involves buying lands and improving them by building houses or hotels. If another player lands on another person's land they pay a certain money under the pretext of rent. This price increases with more improvements to the land. The game goes on until all other players are bankrupt with no land. With a basic concept like that the game was open to improvements. There were a lot of iterations of monopoly due to new ideas (monopoly kids, monopoly business tour) and or new technologies (monopoly electronic bank). While some of these were a great success like the electronic bank some were disappointing. Some of the main problems on this game were the snowballing effect and the long game time as well as the subtle cheating which is common in most of the board games.

During this project we were tasked to make a digital version of monopoly and make improvements to it. We made some new game mechanics and improvements to the original game which mainly tasked to stop the snowballing effect and add more options to change the gameplay. While these changes are subject to change, we made the core elements with the intent to stay.

2.OVERVIEW

2.1.Characters

In *Monopoly Sicilia* there are two unplayable characters: the Mafia and the Police, these characters interact with both each other and the players. Similar to the table version, up to four players can play *Monopoly Sicilia*.

2.2.Mafia

The Mafia serves two purposes for a player: speeding up their earning process and slowing down others' earning process. The players can order favours from the Mafia at any point in the game unless they or the Mafia are in the Prison. The available favours are as follows:

- Attack the hotels of the opponents when it steps on them by either requesting protection money parallel with the price of the hotel or if the person does not pay protection money it will destroy the hotel which reverts them back to 3 houses.
- Blackmail a player who forces them to pay a certain amount of money proportional to the player's money in TRY.
- Sell lands, hotels and houses cheaper than what they originally are but in turn, the Mafia will take a share of the earnings from that land.

- Buy chance or community cards at a fixed price.

2.3. Police

The Police serves as a dissuasive force to prevent people from consulting the Mafia too often. When the Police collides with the Mafia after one or more players make deals with the Mafia, the Police sends the Mafia to custody for 5 turns and punishes the players who have made deals with the Mafia. If the Police collides with the Mafia, without any past deals with the players, it sends the Mafia to custody for 2 turns.

2.4. Forex

In *Monopoly Sicilia* there are other ways to earn money other than drawing chance cards or trapping opponents in your real estates. One of them is our Foreign Exchange system. There are four currencies and players can buy and sell from these currencies and the currencies react to these activities.

2.5. Building shortages

In *Monopoly Sicilia*, unlike *Monopoly Classic*, there are not any limitations on the total amount of houses to be built in a game session. E.g. in *Monopoly Classic* there are a maximum of thirty two houses, however in our iteration of *Monopoly*, the players are free to build as many houses as they wish.

2.6. Set your own limit

As we discussed earlier, in *Monopoly Sicilia*, the players can set the maximum cash limit as they wish. This allows for players to have better control over the game length.

2.7. Power-ups

With *Monopoly Sicilia*, we are planning to implement power-ups to the game. These power-ups include:

- Multiply your earnings by an amount specified by the power-up for an amount of turns which are also specified by the power-up (e.g. the power-up says double the earning for 5 turns)
- Slow down the movement of a player of your choice by an amount specified by the power-up. (e.g. the power-up says %50 percent slow down)
- Strike an opponent and send them backwards for an amount of tiles specified by the power-up.
- Manipulate the forex system. (i.e. Increase or decrease *currency* by %5)
- The power-ups can be stacked, also they can be held on the hand of the player who possesses it as long as they wish. However, players have to activate the power-ups within their turn.

2.8.Monopoly Classic features

2.8.1.Objective: The objective of the game is to be the last person standing without resigning or going bankrupt.

2.8.2.Preparation: The chance cards and community cards are placed on the board. Each player chooses a token to represent themselves. Then each player will be given 1500TL.

2.8.3.Bank: The banker role will be handled by the computer. Bank will give the players their deeds and auction the tiles if needed. The bank gives the mortgages and collects taxes if needed.

2.8.4.Play: Starting with player 1, every player throws the dice. Then starting with the mafia player order will be determined by the value of their thrown dice (Higher to lower). Last player will be the police. Every player's token will be placed on "GO". After that each player will roll the dice in their respective turns and will move forward by their dice value. After they land on a square they can or will do what that square is. If you throw a double you move the initial value then you can throw again. If you throw three doubles in succession you are sent to jail.

2.8.5.GO: Each time a player's token lands on or passes over GO, whether by throwing the dice or drawing a card, pays them a salary.

2.8.6.Chance and Community Chest: When players land on a Chance or Community tile, they draw a card from the respective deck. And they follow the instructions on the card. Specifically, there is a card called the "Get Out of Jail Free", which is held by the player until it is used. When used, it frees the user from the jail and is returned to the bottom of the deck.

2.8.7.Houses and Hotels: If a player buys every tile in a color group, they may buy houses and place them provided that they have enough money and they have built the houses evenly, i.e. they have to build the second house for every tile in a color group if they want to build a third one on one of these tiles in the same color group. The prices of improvements and the tile rent costs of these tiles (houses and hotels) are specified in the respective cards. In addition to houses, the players may also build hotels on their tiles. When a player has three houses on every tile in a color group, the player can build a hotel in one of the tiles in this color group. Only one hotel can be built on a tile. The players may also sell their tile improvements if they wish provided that they sell the improvements evenly, similar to the buying tile improvements case.

2.8.8.Going bankrupt: A player goes bankrupt if they owe to another player or the bank more than their current money and assets. If the player owes to another player, the game turns all of the money the player in debt has over to that player. In addition, the computer refunds the bankrupt player's houses for half the price and pays the debts to the creditor. If the player owes money to the bank rather than a player, the bank immediately seizes all assets and auctions them to other players.

2.8.9.Mortgage: Unimproved properties can be mortgaged at any time. Before an improved property can be mortgaged, all the buildings on all the properties of its color-group must be sold back to the Bank at half price. The mortgage value is printed on each Deed card.No rent can be collected on mortgaged properties or utilities, but rent can be collected on unmortgaged properties in the same group. In order to lift the mortgage, the owner must pay the Bank the amount of the mortgage plus 10% interest. When all the properties of a color-group are no longer mortgaged, the owner may begin to buy back houses at full price.

2.8.10.Jail: You land in Jail when. ..(1) your token lands on the space marked "Go to Jail"; (2) you draw a card marked "Go to Jail"; or (3) you throw doubles three times in succession. You get out of Jail by.. ..(1) throwing doubles on any of your next three turns; if you succeed in doing this you immediately move forward the number of spaces shown by your doubles throw; even though you had thrown doubles, you do not take another turn; (2) using the "Get Out of Jail Free" card if you have it.

2.8.11.Selling Properties: Unimproved properties, railroads and utilities (but not buildings) may be sold to any player as a private transaction for any amount the owner can get; however, no property can be sold to another player if buildings are standing on any properties of that color group. Any buildings so located must be sold back to the Bank before the owner can sell any property of that color-group.Houses and hotels may be sold back to the Bank at any time for one half the price paid for them.

2.8.12.Income Tax: If you land here you have two options: You may pay 18% of your total worth to the Bank. Your total worth is all your cash on hand, printed prices of mortgaged and unmortgaged properties and cost price of all buildings you own.

3. FUNCTIONAL REQUIREMENTS

3.1.Play game

3.1.1.Play new game. The players can rename the tiles on the map as they please. The player can set the end game condition default will be one million. Then the player should choose the game mode between the two options Hotseat mode and the play vs AI mode.

3.1.2.Two different game modes. In the Hotseat mode, a set amount of players (2 to 4 players) play a game on a single computer.. It is possible to fill the remaining players with BOTS. In vs AI mode, the player must play the game against BOTS.

3.1.3.Continue previous game. The player can load a previous game. They can fill the missing players from the loaded Hotseat game with BOTS.

3.2.How to play

In the menu there is a How to play menu to teach players the basic rules of the game.

3.3.Credits

In the main menu there is a credits menu which has all the team members' names.

3.4.Quit Game

This option will quit the game and close the window. If there is a game going on it will warn the player to save before quitting.

4.Nonfunctional Requirements

4.1. Performance

For the game to feel responsive and fluid, the game should work on at least 30 frames per second. In addition, the game should respond to any user request in under a second.

4.2. Usability

Party games are the types of games which unite people on a couch. These games are usually played on special occasions where various people with

different qualities gather around. What makes party games such as Monopoly Sicilia enjoyable is when people first pick up the game, it is easy to understand and play, it has accessibility options to make the experience as fun as possible for people with disabilities. For this reason, we will have a simple but efficient user interface, a how to play guide within the main menu, colorblind mode.

4.3. Reliability

In game progress will be saved locally every round in case of any system faults. With this provided, the users will not have to start over in case something unexpected happens.

4.4. Supportability

The game will be supported in different platforms such as linux, macOS and windows. On the other hand, it will be easily modified when an error is encountered or when a new feature will be added in future implementations.

- The player moves to a tile which the player has three houses on.

Exit condition:

- Player successfully buys a hotel from the Mafia with a cheaper price.
- Player cancels the purchase sequence.
- Player exits the game.

Success scenario events:

1. Player selects the tile's "Buy hotel" option.
2. Player selects the "Make a deal" option from the buy menu to make a deal with the mafia.

Alternative Event Flows:

1. When player wants to quit the game
 - a. Player opens the pause menu by clicking the pause button on the game screen.
 - b. Player selects the "Return to Main Menu" button.
 - c. System saves the game stage and returns to the main menu.
 - d. Player clicks the "Quit Game" button from the main menu.
2. When the player wants to cancel the purchase.
 - a. Player either closes the purchase window or buys the hotel without dealing with the Mafia.

5.1.2 Use Case Scenario #2

Use Case: Open a power-up crate

Primary Actor: Player

Stakeholders and Interests: Player wants to open a power-up crate.

Precondition: Player must be in the game and it must be the player's turn.

Entry condition:

- Player selects the "Buy power-up crate" button from the screen.

Exit condition:

- Player gets the power-up from the crate.
- Player exits the game.

Success Scenario Events:

1. Player buys the power-up crate
2. Player receives a random power-up.
3. Player uses or keeps the power-up he received.

Alternative Event Flows:

3. When a player wants to buy a crate but does not have enough money for it.
 - a. Player clicks the buy a power-up crate button
 - b. A warning will write at the bottom to warn the player that they don't have enough money.
4. When player is in jail
 - a. Player clicks the buy a power-up crate button
 - b. A warning will write at the bottom to warn the player that they are in jail and they cannot do that without leaving the jail

5.1.3 Use Case Scenario #3

Use Case: Forex Invest

Primary Actor: Player

Stakeholders and Interests: Player wants to invest in a forex currency.

Precondition: Player must be in the game.

Entry Condition:

- Player enters the amount of money that he wants to invest next to the desired currency and clicks buy.
- Player enters the amount of money that he wants to sell next to the desired currency and clicks sell.

Exit Condition:

- Player finishes investing
- Player quits the game

Success Scenario Events:

1. Player enters the amount of money he wants to sell or invest next to the desired currency.
2. If the buy or sell button is clicked and if the player has enough money to exchange, then the money is transferred from one currency to the other one.
3. The game will continue.

Alternative Scenarios:

1. When a player wants to invest but does not have enough money for it.
 - a. Player enters the amount of money he wants to sell or invest next to the desired currency.
 - b. A warning will write at the bottom to warn the player that they don't have enough money.
2. When player is in jail
 - a. Player enters the amount of money he wants to sell or invest next to the desired currency.
 - b. A warning will write at the bottom to warn the player that they are in jail and they cannot do that without leaving the jail

5.1.4 Use Case Scenario #4

Use Case: Trade with another player.

Primary Actor: Player

Stakeholders and Interests: Player wants to trade with another player

Pre Condition:

- Player must be in the game.

Post Condition:

- The properties owned by both players are updated according to the trade.

Entry Condition:

- Player selects the trading option on their screen.

Exit Condition:

- Player clicks the "Back" button from the trading screen.
- Player quits the game.

Success Scenario Events:

1. Player selects the trading option on their screen.
2. Player chooses the player they want to trade with.
3. Player selects what they want from the selected player.
4. Player offers what they will give for the selected item/s.
5. Player clicks the "Offer Deal" button.

Alternative Event Flows:

1. Player is in jail.
 - a. Player clicks the trade button.
 - b. A warning will write at the bottom to warn the player that they are in jail and they cannot do that without leaving the jail.
2. Player offers an amount that he cannot give.
 - a. Player selects the trading option on their screen.
 - b. Player chooses the player they want to trade with.
 - c. Player selects what they want from the selected player.
 - d. Player offers what they will give for the selected item/s. The money he offers is an amount that he can't pay.
 - e. A warning will write at the bottom to warn the player that he cannot offer that.

5.2 Dynamic Models

5.2.1 Sequence Diagrams

5.2.1.1 New Single Player Game Scenario

Scenario: Player starts a new single player game.

Player wants to play the game. Player interacts with the system by using the screen. Player clicks the new game button and selects the game mode as a single player game. He/she selects the details such as map, nickname etc. Then, the bot creator creates all the AI characters such as police, mafia and other AI player bots. After that, according to the information given by the players, the map is created. Card deck class creates chance and community cards and adds them to decks. After that the forex manager initializes the forex system and prepares exchange rates for the first run. Sound class activates the sounds of the game such as music and effects. Finally, the game manager gives permission to the system to start the game and the player starts playing the game.

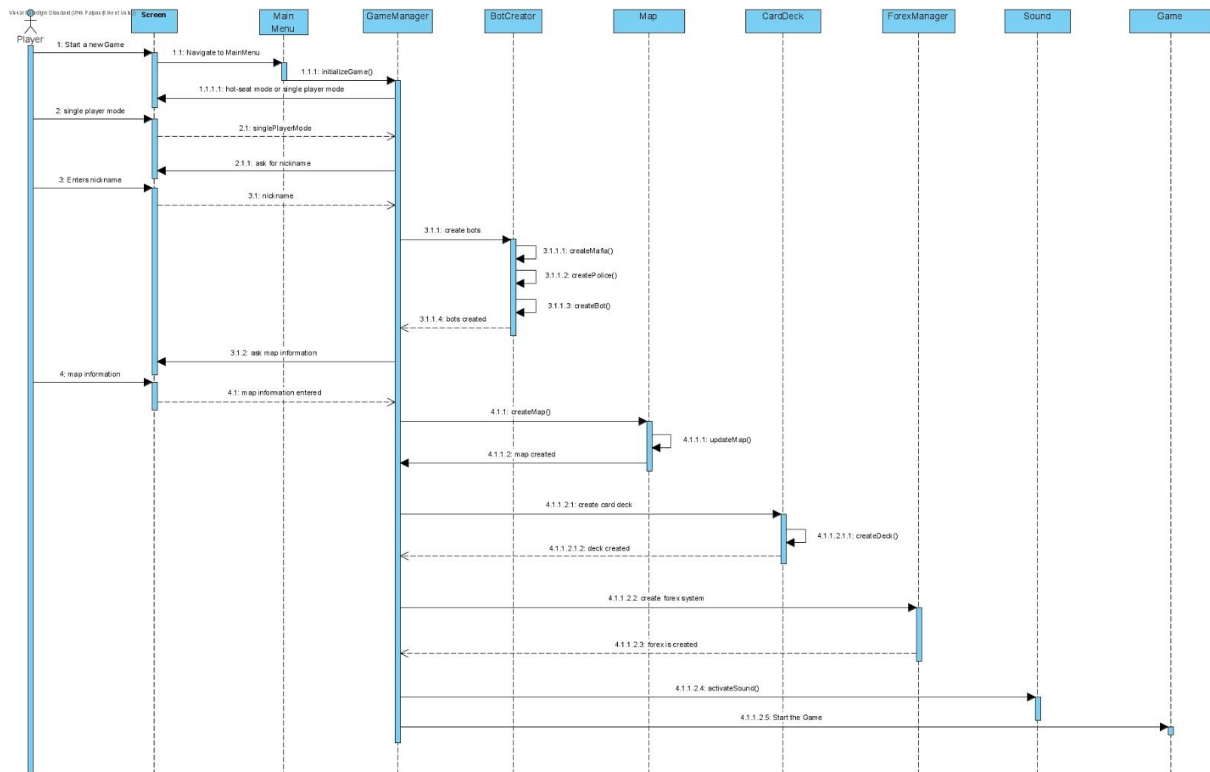


Figure-1b: Sequence diagram for a new game scenario

5.2.1.2 Buying a Power-up Crate Which Returns a Multiply Earning Power-up in the Turn of the Player

Scenario: Player wants to buy a power-up crate in his/her turn and a multiply earning power-up comes out of the crate.

Player interacts with the game by using a screen. After the player's turn has come, the player starts his/her turn and rolls the dice by using the rollTheDice() method. Then Dice class returns the diceValue to the GameManager. By using the diceValue setTileNumber() method of the Player class calculates the new position of the player. After that, GameManager updates the map with the help of UpdateMap() method. Then, the Player class calls the openPowerUpCrate() method of the PowerUpCrate class and that method calls getRandomPowerUp() method of the PowerUpManager class. Finally, it returns a multiply earning power up and Player class adds the newly bought power up to the array kept by the Player object by using the addPowerUp() method.

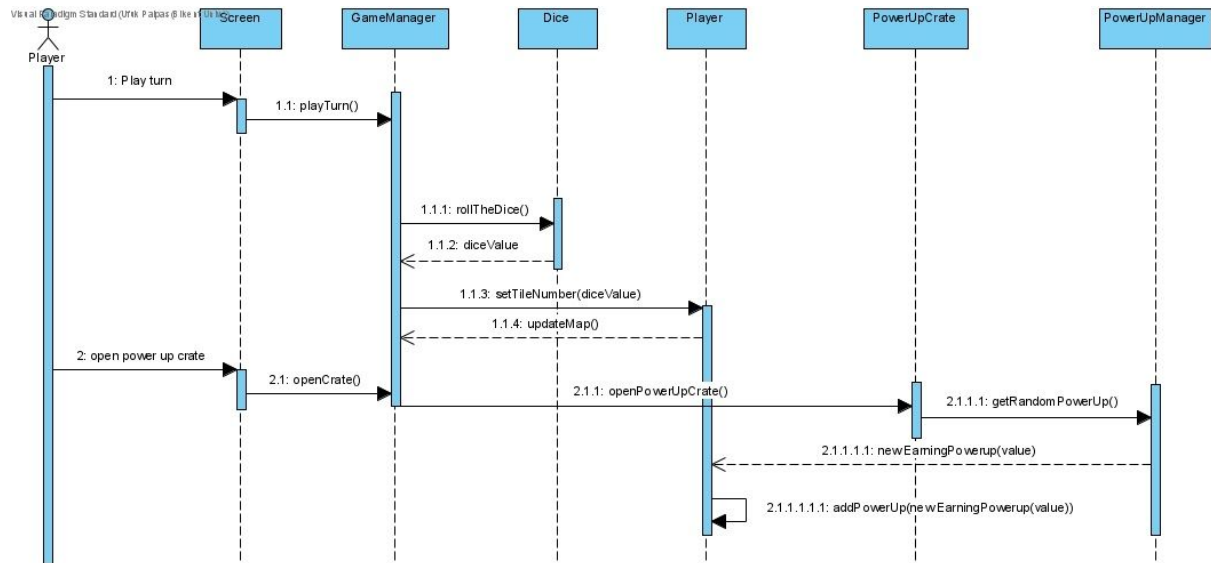


Figure-1c: Sequence diagram for buying a new power-up crate scenario

5.2.1.3 Player Lands on a Community Tile and Draws a Community Card Which Requires 300 Dollars Payment

Scenario: Player starts his/her turn. Then, moves according to the result of rolled dice. Player stands on a community card tile according to the result of the rolled dice. He/she draws a community card from the card deck and the drawn card says that to pay 300 dollars. Finally, player pays the amount.

At a random time during the game, with the help of the playTurn() method the next player gets the turn. Player whose turn has just started starts his/her turn and rolls a dice by using the rollTheDice() method. Then, Dice class returns the diceValue to the GameManager. By using the diceValue, GameManager calls the setTileNumber() method of Player and updates the map. After that GameManager calls the getIsChance() method of CardTile and it returns a boolean value ("false"). GameManager calls CardDeck's drawCard() method and after CardDeck selects a card it uses the getFeature() method of the Card and returns it to the GameManager. Finally, it calls the takeMoney() method of the Bank to pay the amount.

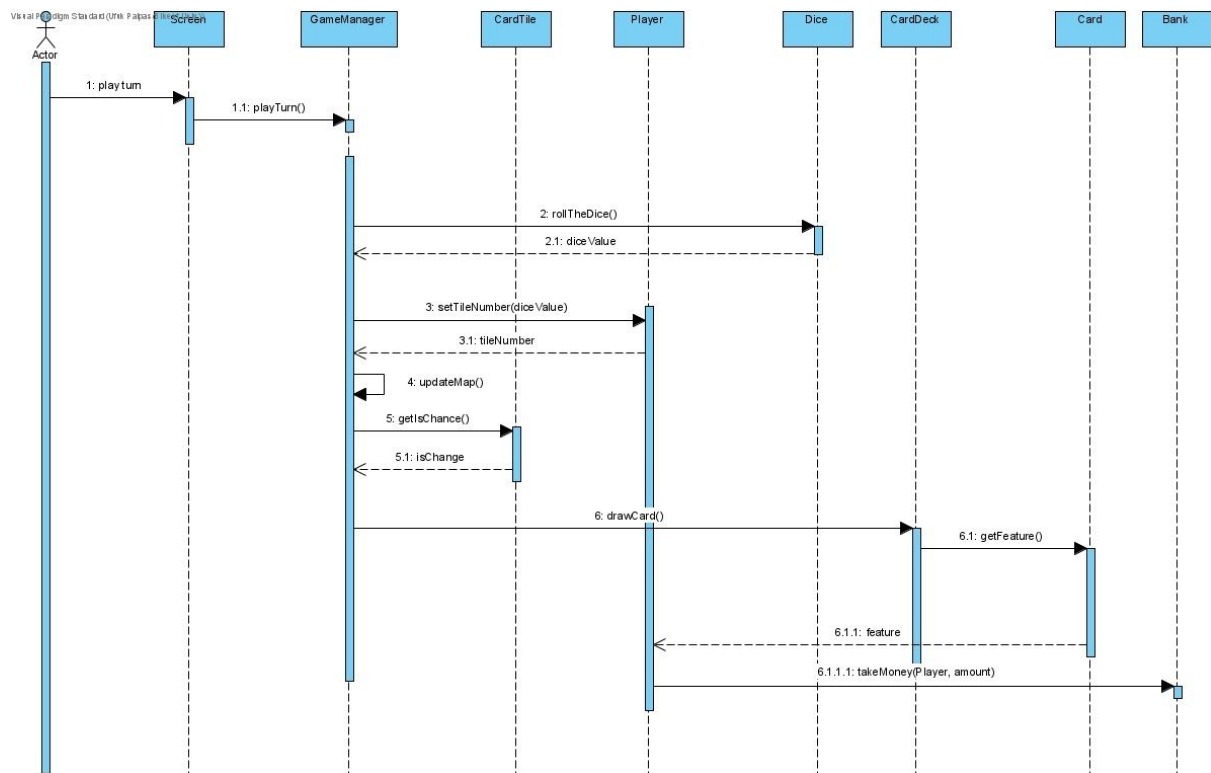


Figure-1d: Sequence diagram for landing on a card tile and drawing a card

5.2.1.4 Buy a Tile from Mafia after Rolling the Dice

Scenario: Player rolls the dice. After that he/she moves the character according to the result of the dice, he/she wants to buy the tile. However, make this by using the mafia.

Dice class rolls the dice with the help of rollTheDice() method. After that User class recalculates the position of the player by using setTileNumber() method and the game manager updates the map. Mafia class sells the tile to the player by using the sellTile() method. isOwned and whoseTile attributes are updated by the CityTile object and with the attribute's set methods. After payment User class updates the account of the player by using the takeMoney() method. And adds the tile to the tileList of Player class or creates a tileList for Player class and finally, the game manager updates the map again.

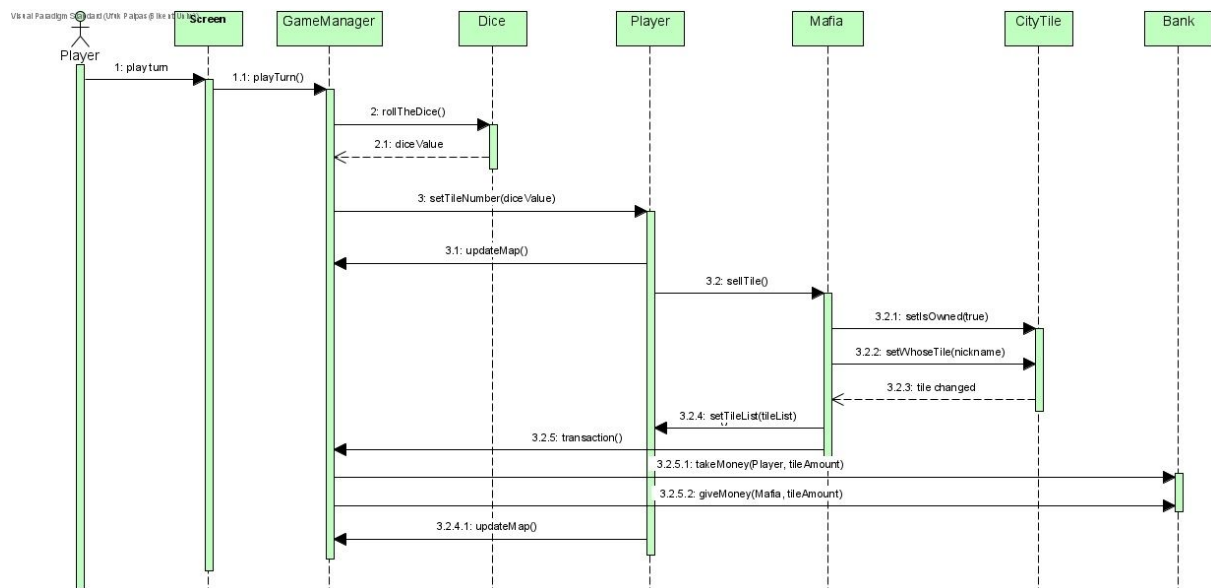


Figure-1e: Sequence diagram for buying a tile from mafia

5.2.1.5 Player Plays the His/Her Turn, Lands on Another Player's Tile and Pays Rent

Scenario: Player plays his turn in time t during the game. He/She rolls the dice and lands on someone else's tile. Thus, the player pays the rent amount to the owner of the tile. Mafia gets its share, if the tile was sold by the mafia.

In the player's turn, the player rolls the dice by using the `rollTheDice()` method. Then, the Dice class returns the `diceValue` to the GameManager. It calculates the new position of the player and then, the GameManager calls the `getIsOwned()` method. True is returned to GameManager. After it learns that it calls `getWhoseTile()` accordingly to learn the owner of the city tile. GameManager gets the rent amount of the tile by calling the `getRentAmount()` method of the CityTile. Now, to pay the rent amount, the game needs to know the tile is sold by whom. There are 2 alternatives here. It might be sold by the Bank or the Mafia. If it is sold by the bank, the game manager calls the `takeMoney()` method of the Bank to take the rent amount of the tile from the player, and the `giveMoney()` method to give the taken amount to the owner of the tile. If it is sold by the mafia, the same process is applied; however, 20% of the rent amount is given to the mafia as a share by using the `giveMoney()` method.

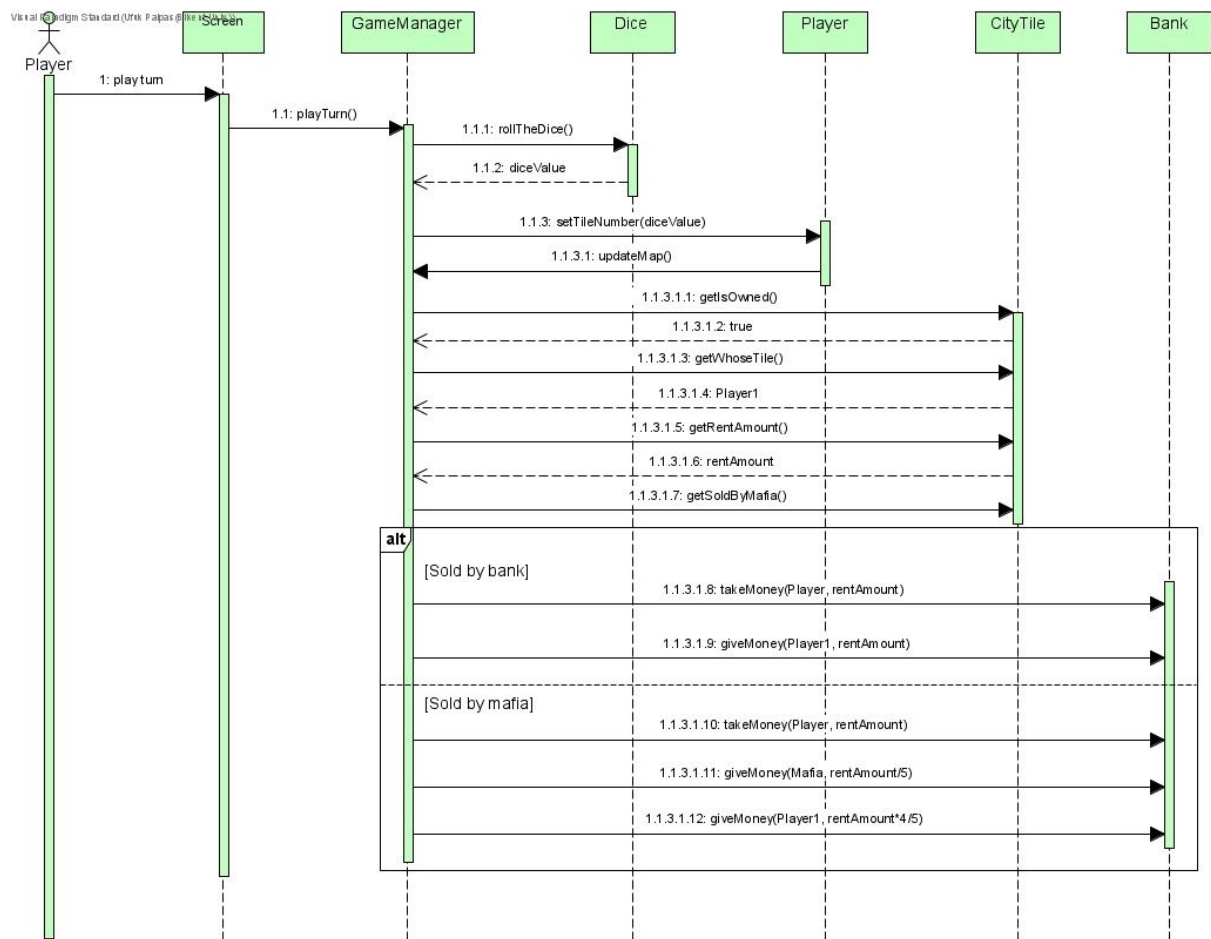


Figure-1f: Sequence diagram for paying rent amount of the tile that player stands on

5.2.2 Activity Diagram

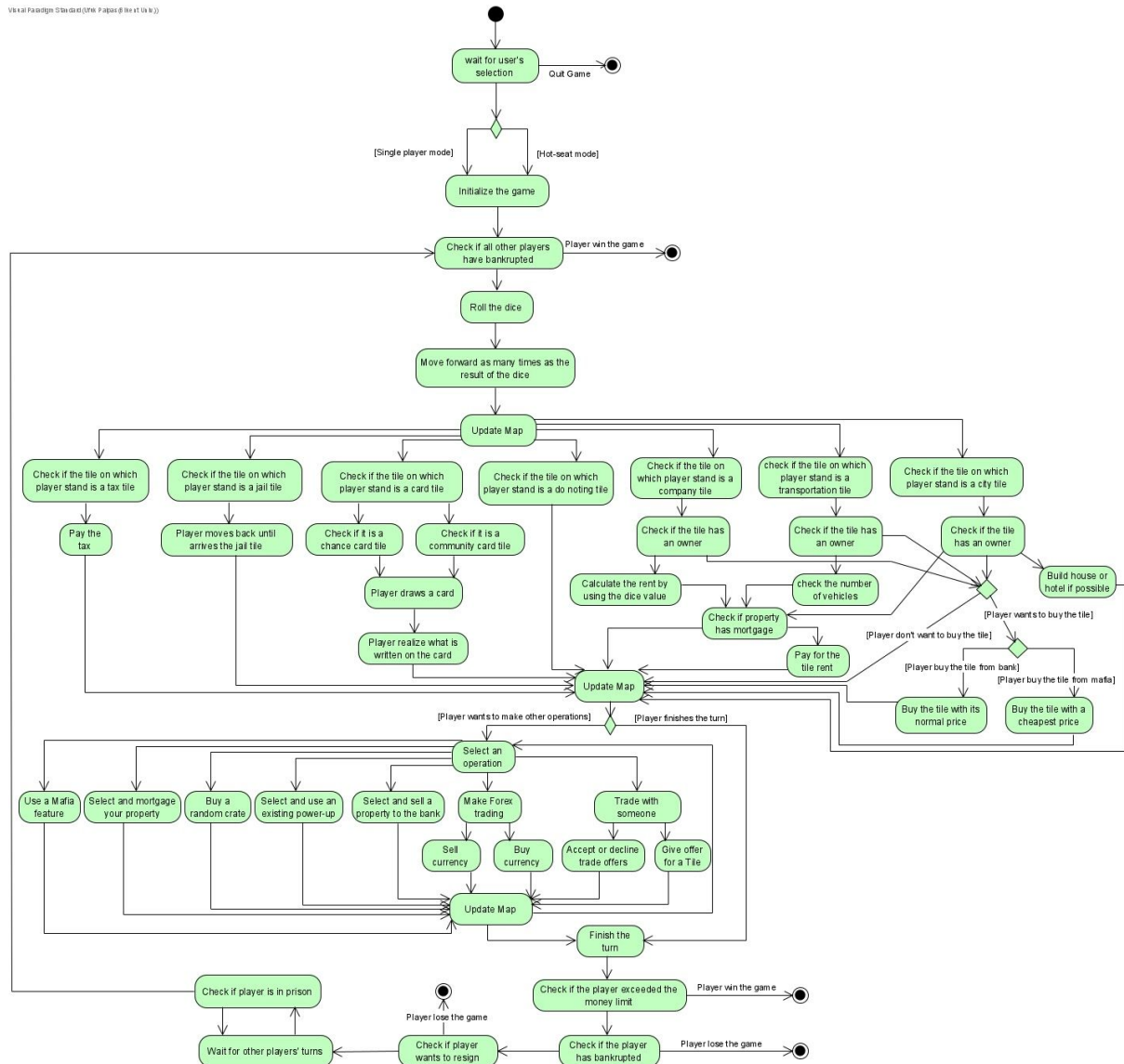


Figure-1g: Activity diagram which contains both tile operations and in-game operations

This activity diagram shows the general processing of the game.

First of all, the game gets an input from the player so that player chooses a game mode, single player mode or hot-seat mode. At this point, the player has a chance to quit the game before starting a new game. Then, the system initializes the game. Player rolls the dice and moves forward according to the result of the rolled dice. Then, the system updates the map accordingly. There will be four different cases at this point.

Case one: If the tile is a jail tile, the player moves back to the jail tile and stands on it.

Case two: If the tile is a do-nothing tile, players do not need to do anything for this turn.

Case three: If the tile is a card tile, the system checks the type of the card and the player draws a card according to its type. Player performs the content of the card.

Case four: If the tile is a city tile, the system checks whether the tile has an owner or not. If yes, the player pays the owner for the rent. If not, the player gives a decision to buy or do nothing. If the player wants to buy it, he/she needs to decide where to buy it, mafia or the bank. If the owner of the tile is the player, he/she can build a house or hotel.

Case five: If the tile is a tax tile, the system automatically gets the amount of tax which is determined before from the player.

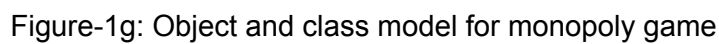
Case six: If the tile is a company tile, the system checks whether the tile has an owner or not. If yes it calculates the rent amount using the dice value and checks if the property has a mortgage. if it has no mortgage on it, the player pays the rent amount. Else, Player gives a decision to buy or do nothing. If the player wants to buy it, he/she needs to decide where to buy it, mafia or the bank.

Case seven: If the tile is a transportation tile, the system checks whether the tile has an owner or not. If yes it calculates the rent amount by looking at the number of vehicles that the owner has and checks if the property has a mortgage. if it has no mortgage on it, the player pays the rent amount. Else, Player gives a decision to buy or do nothing. If the player wants to buy it, he/she needs to decide where to buy it, mafia or the bank.

Then, the system updates the map. Then player has a chance to finish the turn or make any other operation such as using mafia feature, mortgaging a property, buying a random crate, using a power-up selling a property, trading or making some forex operations and then, the system updates the map again.

After that System checks if the player has exceeded the money limit where the player wins the game. System checks if the player has bankrupted or not to check whether the player lost the game. At that point, the player has a chance to resign; however, if the player resigns he/she loses the game. After that other players play their turns. After the player's turn comes, the system checks whether the player is in prison or not. If the player is in prison other players play their turns again. When the player is proper to continue the game, the system checks whether all other players have bankrupted. In that case the player wins the game. Otherwise the player rolls the dice again and starts a new turn.

Visual Paradigm Standard (oguztuz)



5.4 User Interface - navigational paths and screen mock-ups



Figure-2a: Main menu

This is the main menu screen of *Monopoly Sicilia*. Firstly, here you can navigate to the set a new game screen where you can choose whether you want to play the Hotseat mode with your friend(s) or you want to play vs AI. Secondly, you may toggle the colorblind mode, set the volume for the background music, toggle the narrator. Thirdly, you can load your last saved game and continue playing on that. There you will be greeted with an option to kick another player from the previous game session if the previous game mode is Hotseat mode. Else, the game will load the previous vs AI session and the player then may start playing. Other than those cases, the player may see the contributors by entering the Credits scene or they may simply quit the game.

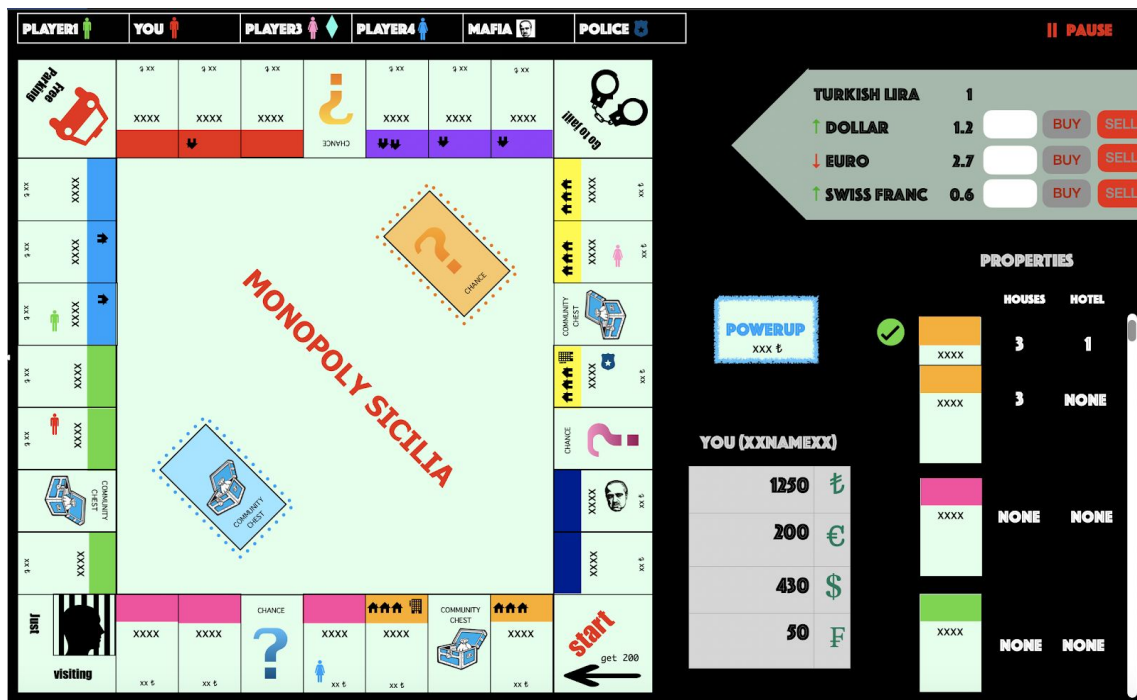


Figure-2b: Game play screen of monopoly.

The game screen for “You” when it is Player 3’s turn in Singleplayer mode. On the top right corner, Player 3 may buy or sell foreign currencies. In the middle as a grey table, the account of the current player (“You” in this case) including all currencies is shown. The properties section represent the owned properties of the current player. This section is scrollable to show all the properties. If the player collects all the tiles of a colour group, a green tick appears next to that group. If the green tick appears then houses or a hotel can be bought. The blue powerup button lets the player buy a powerup. If the player has power ups at his hand, then they are shown below that button and can be used by clicking on them. On the left hand side of the screen there is the board. The board shows the location of the tiles, the positioning of the players and other characters on the light green space inside each tile with the symbols next to their name. The colored bar on top of each tile contains any houses or hotels built on it with small icons. The drawing cards are located in the middle section of the board. Above the board names of the players can be seen. The blue diamond next to the name shows which player’s turn it is. The players may navigate to the pause menu if they press the “Escape” button on their keyboard.



Figure-2c: Buy pop-up

When a player decides to buy a house or a hotel in their turn, a pop-up similar to the following shows up. Here the player may deal with the Mafia to get a cheaper purchase or may choose the peaceful option of paying the full price.



Figure-2d: Pause screen

This screen comes when the game is paused while playing. In this screen, the players have the option to navigate to the settings screen, see the credits or go back to the main menu.

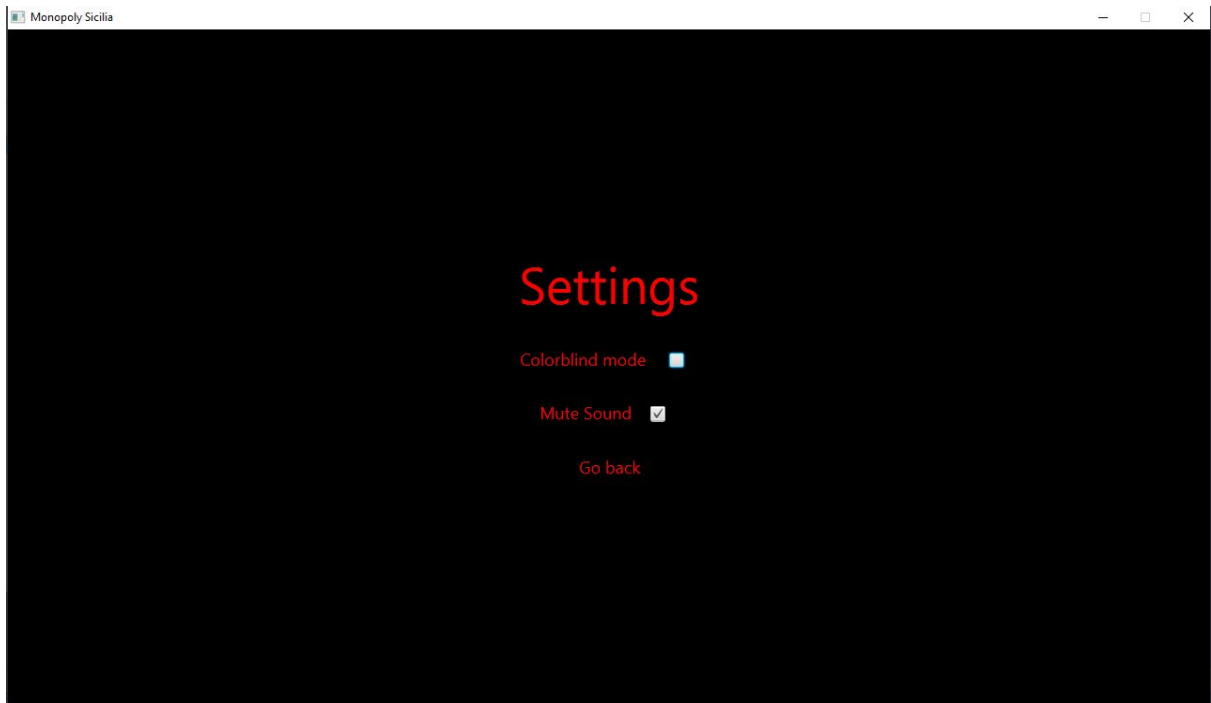


Figure-2e: Settings screen

This screen shows the available settings in our game. Here the player may either toggle the SFX or the colorblind mode.



Figure-2d: Credits screen

This screen shows the people who contributed to the game.

6. Improvement Summary

We have reduced the explanations of the functional requirements to only explain what they do without going into the specifics. Also we reworded it to have different verbs as given in the feedback. In our overview, we narrowed the scope of our StrikePowerUp. We tweaked the features of the Mafia to make him more fearsome. We made the figure fonts bigger. We redesigned the use case diagram with better scenarios. We changed the use case scenarios to have more game related scenarios. We explained them with entry conditions and different outcomes. We changed the activity diagram to use negative space better. We updated the class diagram to show the changes we made. Most of the content was updated. Functional requirements and non-functional requirements no longer contain implementation details such as the platform, language and libraries. The requirements are renamed to better represent our purposes.

7. Glossary & References

Hasbro. (n.d.). Monopoly Classic Rulebook. Retrieved November 29, 2020, from <https://www.hasbro.com/common/instruct/00009.pdf>