CS 319 Project Analysis Report Iteration 1

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

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.

Overview

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. **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 claiming protection money parallel with the price of the hotel or if the person does not have protection money it will destroy the hotel which reverts them back to 3 houses.
   - Blackmail a player which forces them to make a decision whether paying money to avoid any consequences.

- Sell lands and houses cheaper than what they originally are but because of that the Mafia will take a share of the earning from that land.
- Buy chance or community cards at a fixed price.

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.

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.

5. **Building shortages**

In *Monopoly Sicilia*, unlike *Monopoly Classic,* there is 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.

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.

7. Power-ups

With *Monopoly Sicilia*, we are planning to implement additional power-ups to the drawing cards of the table version of the game. These power-ups include:

- Multiply your earnings by an amount specified by the card for an amount of turns which are also specified by the card (e.g. the card says double the earning for 5 turns)
- Slow down the movement of a player of your choice by an amount specified by the card. (e.g. the card says %50 percent slow down)
- Strike an opponent and send them backwards for an amount of tiles you choose provided that this amount is within the range specified by the card.
- 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. Also they cannot use the power-ups as they receive them. In other words, players must hold a power-up at least one turn long before using it.

8. Monopoly Classic features

8.1. **Objective:** The objective of the game is to be the last person who didn't go bankrupt.

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

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.

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.

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 $200 salary.

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.

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.

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 pay 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.

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.

8.10. **Jail:** You land in Jail when. ..(I) 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.. .(I) 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.

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.

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.

Functional Requirements

1. Play game

1.1. Play new game

With this feature we would like to present the player the option to start a brand new game session. In addition we want the player to feel themselves at home by providing a couple of customization options. Firstly, the players will have the freedom to rename the tiles on the map as they please before initializing a game session. Secondly, the player can set the end game condition before initializing the game. If they choose not to, the default value will be a million dollars. Then the player will get to choose the game mode. The two options we planned for the release build are the Hotseat mode and the play vs AI mode.

1.2. Hot-seat and vs AI

In the Hotseat mode, a set amount of players (2 to 4 players) play a game on a single computer by taking their turns one by one. Whereas in vs AI mode, the player will play the game against BOTs. The players also get to choose if they want to complete the party with BOTs or they want to be by themselves.

1.3. Continue previous game

This feature allows players to continue a previous, unfinished game session. In addition we would like to add the option to kick players from previous Hotseat sessions which eliminates a lot of hassle on the player's part if a friend of them is missing or simply they don't want to play. Then the game will replace the kicked players with BOTs.

2. How to play

In the main menu, we will have a sub menu for a how to play section. Here we will present the players a few slides to get them familiar with the game without even playing it. The goal of presenting this information here is to teach them the basics of our game and ensure that their first gameplay is as seamless as possible.

3. Credits

In the main menu, we will have a credits screen that shows the names of each group member. It will list each member in alphabetical order so that there will be no misunderstanding.

4. Quit Game

This option will quit the game and close the window. If the game is quitted during a gameplay it will give an information message to inform

that the game will not be saved on exit. And it will give a choice to go back to the game or quit without saving.

Nonfunctional Requirements

1. Extendibility

The organization stage of our project helps us to have a strong foundation for our source code. The software engineering and object oriented design principles we discussed in the class serves as a guideline for the project's roadmap. Our source code is going to be organized in such a way that adding or subtracting features will require minimum consequences.

2. Accessibility

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, and possibly a narrator option implemented in the game.

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. For future releases, we might use cloud services to backup the progress of the players.

4. Maintainability

Since we are going to use JavaFX in our project, we will write the code once and run it on Java Virtual Machine. Thus we will have cross-platform support for linux and mac. On the other hand, we are planning to write our code in such a way that it will be easy to maintain in the long run. For these purposes, we are not planning to use any external libraries.

5. System Models

5.1 Use Case Model

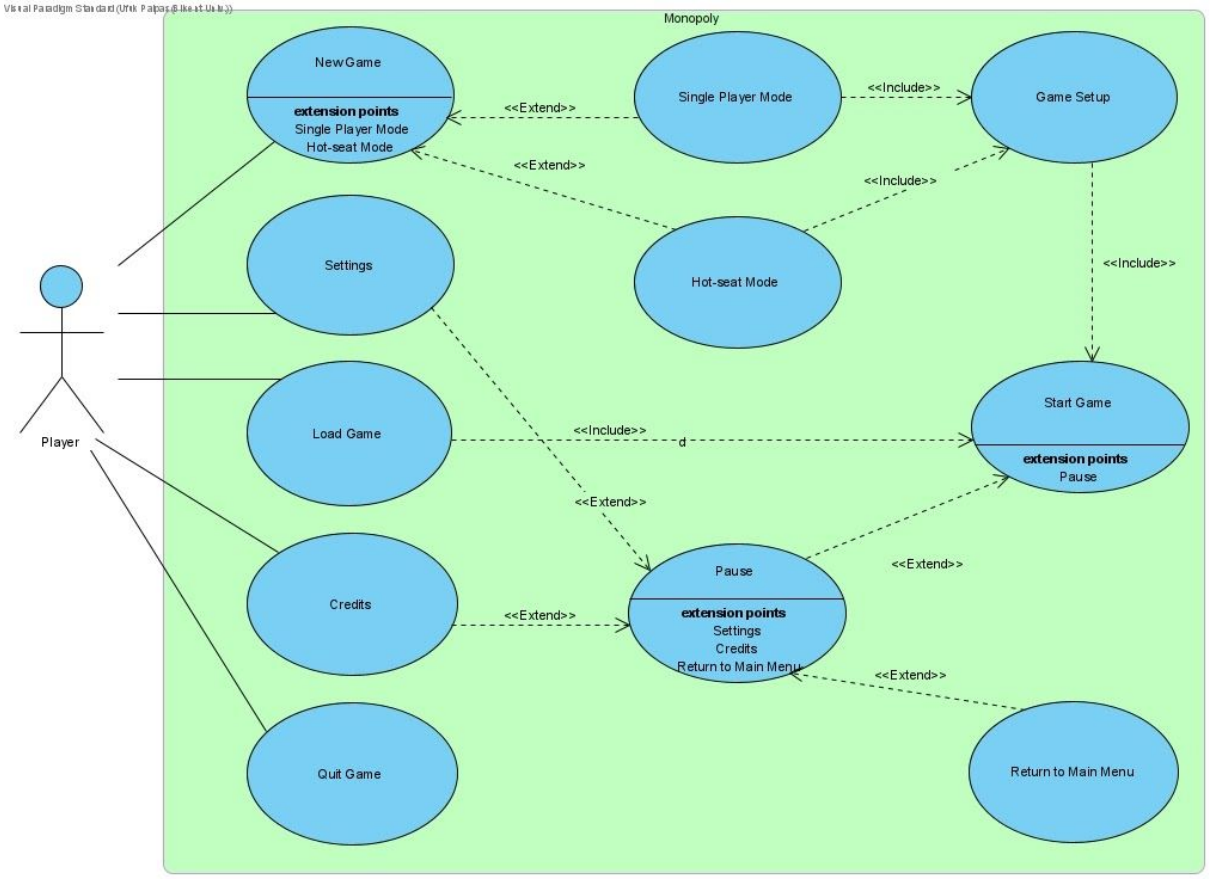

Figure -1: Use Case Model

5.1.1 Use Case #1

Use Case: New game

Primary Actor: Player

Stakeholders and Interests: Player or players want to create a new monopoly game.

Precondition: Player must be in the main menu.

Entry condition:

- Player selects the “New Game” button from the main menu.
- Player should choose a game mode (Hot-seat or Single Player Mode) and make the game setup (Number of players, Nicknames etc.)

Exit condition:

- Player selects pause button and than return to main menu button

- Player finishes the game successfully

Success scenario events:

1. Player selects the “New Game” button from the main menu.
2. Player selects single player vs AI mode.
3. Player arranges the game setup.
4. Player rolls the dice and moves according to the result of dice.
5. Player either ends up in a buying block or card drawing block.
6. Player draws a card, buys the property, pays to the owner of the property according to the situation.
7. Player waits for the round to end.
8. Player repeats the steps 3-6.
9. Player finishes the game when the AI gets bankrupt.

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 player wants to change settings during the game
    a. Player opens the pause menu by clicking the pause button on the game screen.
    b. Player selects the “Settings” button.
    c. Player saves the changes and returns back to the game.

5.1.2 Use Case #2

Use Case: Load Game

Primary Actor: Player

Stakeholders and Interests: Player or players want to load their saved game.

Precondition: Player must be in the main menu.

Entry condition:

- Player has to have a saved game beforehand.
- Player selects the “Load Game” button from the main menu.

Exit condition:

- Player selects the pause button and then returns to the main menu button.
- Player finishes the game successfully.

Success Scenario Events:

1. Player starts the game from the last saved stage.
2. Player follows the same steps defined in the use case of the new game.
3. Player completed the game successfully or unsuccessfully

Alternative Event Flows:

3. 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 quits the game.
    d. Player clicks the “Quit Game” button from the main menu.

4. When player wants to change settings during the game
    d. Player opens the pause menu by clicking the pause button on the game screen.
    e. Player selects the “Settings” button.
    f. Player saves the changes and returns back to the game.

5.1.3 Use Case #3

Use Case: Credits

Primary Actor: Player

Stakeholders and Interests: Player or players who want to see who created the game.

Precondition: Player must be in the main menu.

Entry Condition:

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

Exit Condition:

- Player clicks the “Return to Main Menu” button on the credits screen.

Success Scenario Events:

1. Player clicks to “Credits” button
2. Credits screen open.
3. System shows the credits in a text format

Alternative Scenarios:

1. When player wants to main menu
    a. Player clicks the “Return to Main Menu” button,
    b. Player returns to the main menu.

5.1.4 Use Case #4

Use Case: Settings

Primary Actor: Player

Stakeholders and Interests:

- Player wants to change sound level and sound effects.
- Player wants to change the colour options of the game.

Pre Condition:

- Player must be in the main menu.
- Player must be on the pause screen.

Post Condition:

- Sound settings of the game are updated.
- Colour settings of the game are updated.

Entry Condition:

- Player selects the “Settings” button in the main menu.
- Player selects the “Settings” button in the pause screen.
- Player arranges the sound settings by using the volume bar or mute buttons.
- Player arranges the color settings by using the predefined color buttons.

Exit Condition:

- Player clicks the “Back” button after reviewing or changing the game settings.

Success Scenario Events:

1. Player selects the “Settings” button from the main menu or pause screen.
2. Player changes the settings.

3. System applies the modified settings.
4. Player returns to the pause screen or main menu.

Alternative Event Flows:

1. Player does not change the game settings and returns to the main menu.
    a. Player clicks the “Return” button.
    b. System opens the main menu.
2. Player does not change the game settings and returns to the game.
    a. Player clicks the “Return” button.
    b. System opens the pause screen back.

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. He/she 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 game manager initializes the game and then, the bot creator creates all the AI characters such as police, mafia and other AI player bots. After that map is created and updated again. 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.

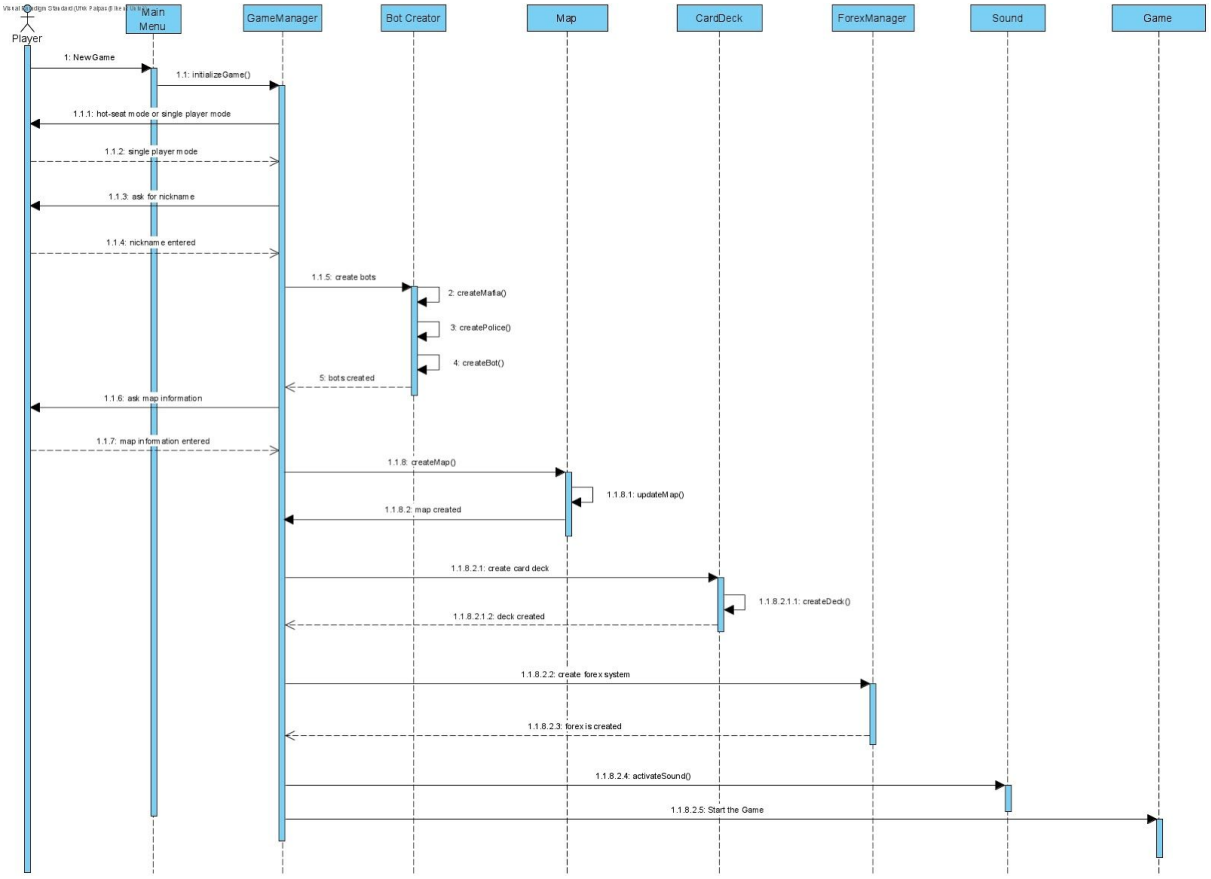

Figure-1b: Sequence diagram for new game scenario

5.2.1.2 Settings Scenario

Scenario: Player wants to mute the game's sound by entering settings from the main menu.

Player is on the main menu and he/she clicks the "Settings" button and opens the settings menu. The settings menu loads the sound setting data from the sound class by using the getCurrentSettings() method. If he/she does not want to change any setting he/she returns to the main menu. Otherwise, the player clicks to the mute button. So that game manager deactivates the sound with the aid of the deactivateSound() method and the settings are being saved automatically when the player returns back to the main menu.

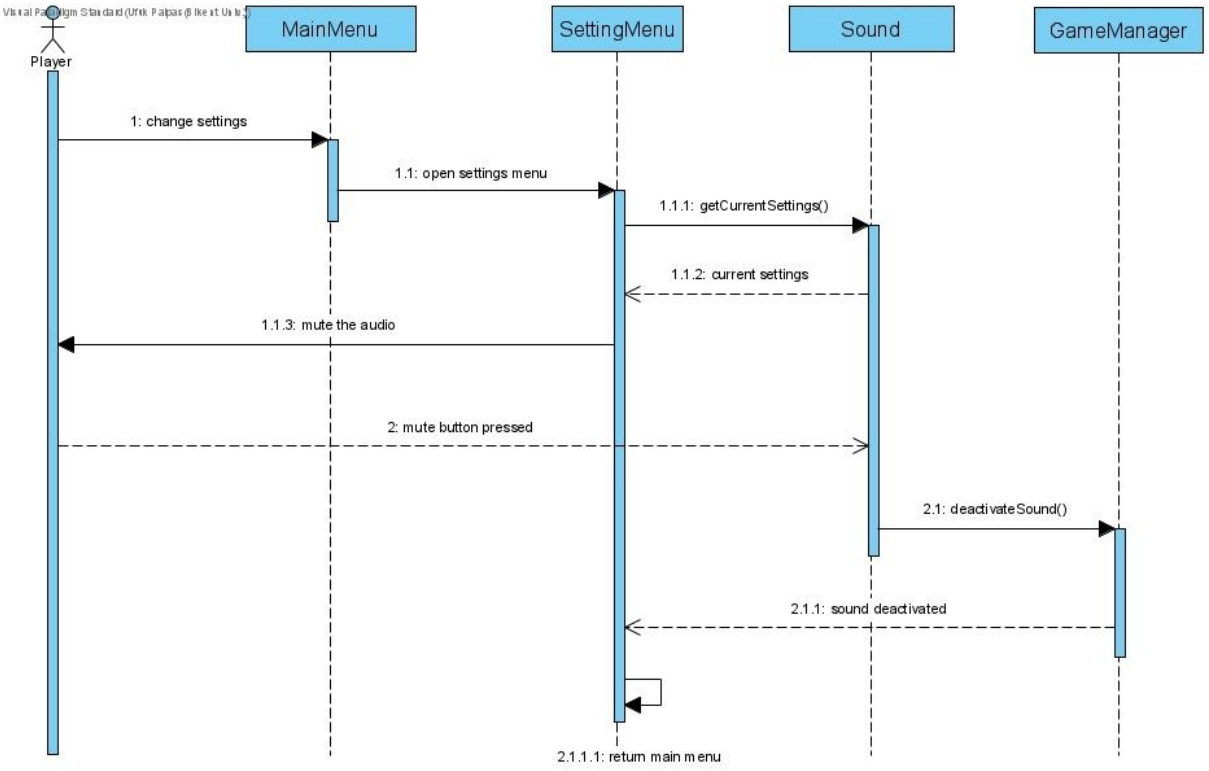

Figure-1c: Sequence diagram for settings scenario

5.2.1.3 Performing the Task Written on a Card that is Drawn When Standing on a Card Tile

Scenario: Player starts his/her turn and moves according to the result of rolled dice. Player stands on a card tile according to the result of the rolled dice. He/she draws a card from the card deck and the drawn card says that go to prison then, the player moves back to the jail tile.

Player has started the game and this is his/her turn. Player wants to roll the dice. Game manager gets the command and rolls the dice with the help of the Dice class and its rollTheDice() method. Then, the Player class recalculates the position of the player according to the result of the dice with the help of setTileNumber() method. After that the game manager updates the map accordingly. Tile class checks the type of the tile and reports that it is a card tile. drawCard() method of CardDeck class gets a card and card object shows the feature of the card. Then, the card is added to the card list in Player class. After that Player class recalculates the position of the player to prison tile and sets the attribute isArrested as true by using the method setIsArrested(true). Finally, the game manager updates the map again.

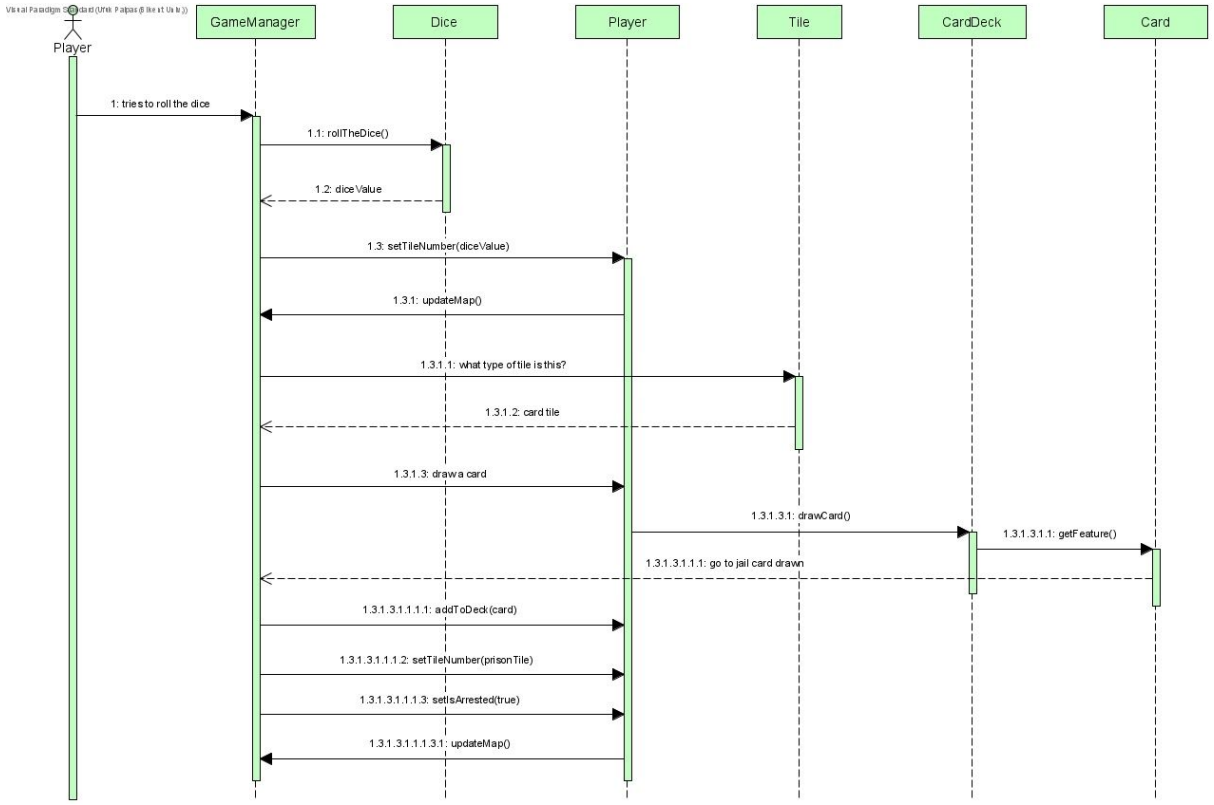

Figure-1d: Sequence diagram for playing scenario

5.2.1.4 Buy a Tile from Mafia after Rolling the Dice

Scenario: Player rolls the dice. After 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.

Game manager initializes the game. Then, 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 setAccount() 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.

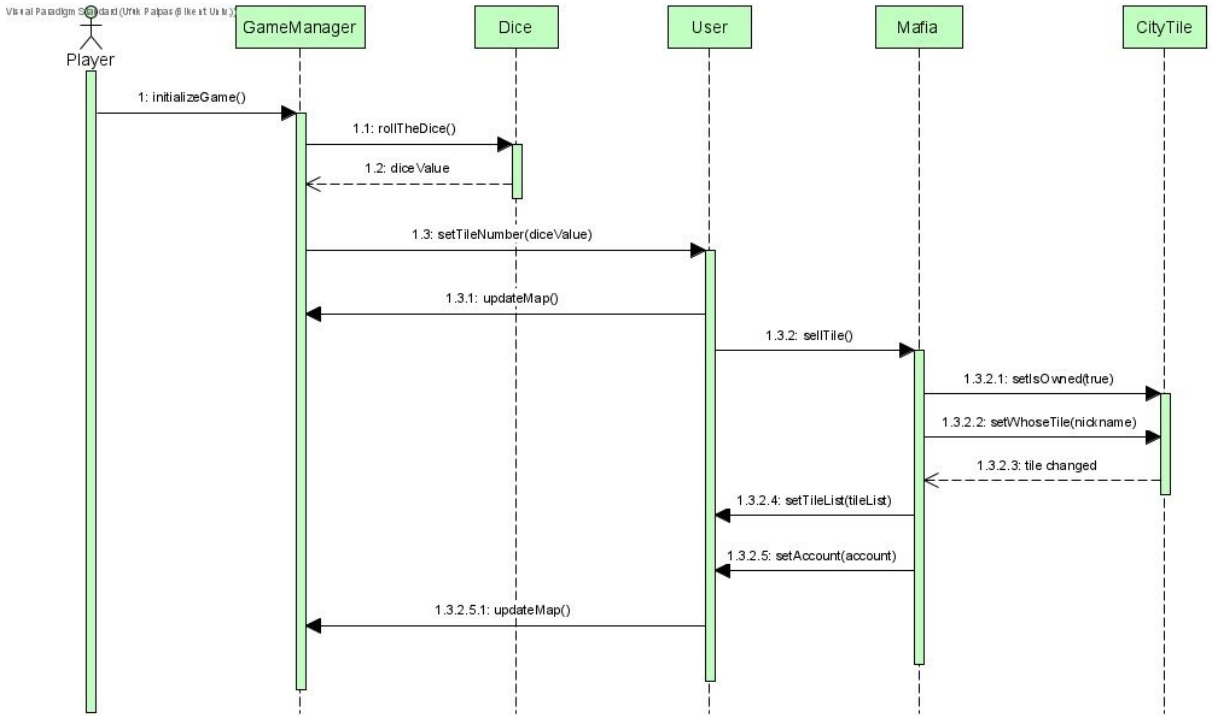

Figure-1e: Sequence diagram for buying a tile.

5.2.2 Activity Diagram

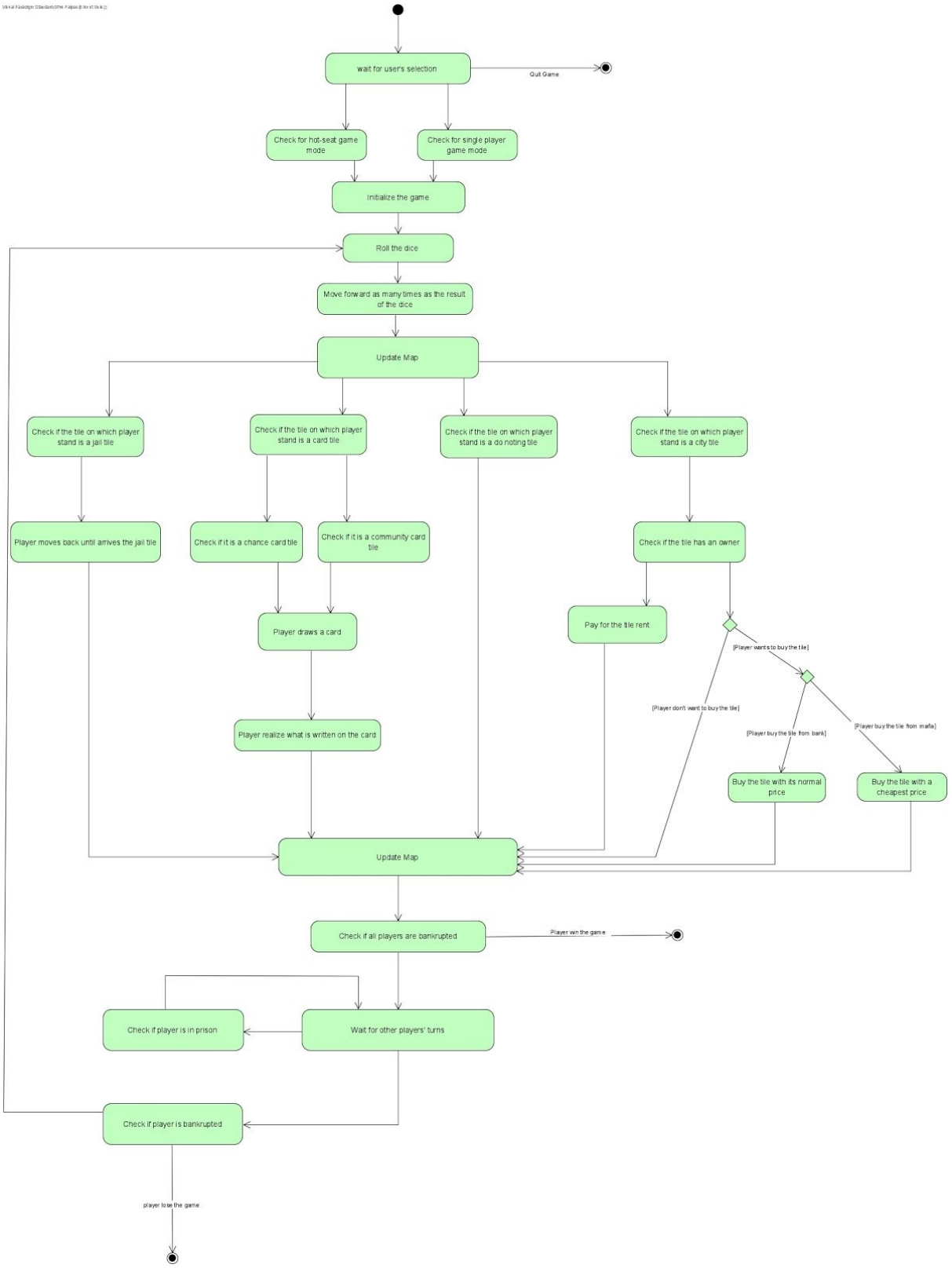

Figure-1f: Activity diagram of monopoly game

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 want to buy it, he/she wants to buy it from mafia or police

Then, the system updates the map. If all users are bankrupted the player wins the game. Otherwise, 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. Otherwise the system checks whether the player is bankrupt or not. If the player is bankrupt he/she loses the game. Otherwise, the player rolls the dice again and the game continues.

5.3 Object and class Model

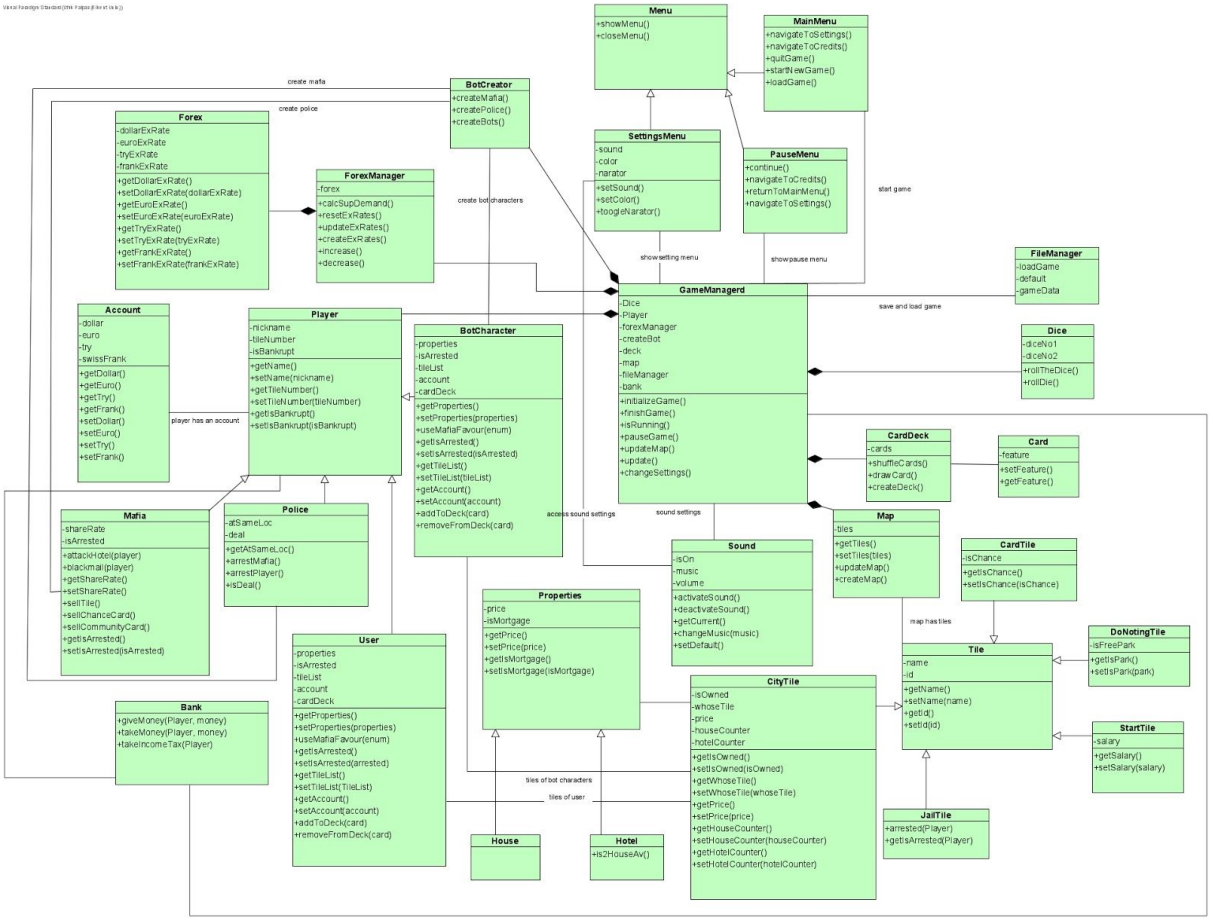

Figure-1g: Object and class model for monopoly game

5.4 User Interface - navigational paths and screen mock-ups

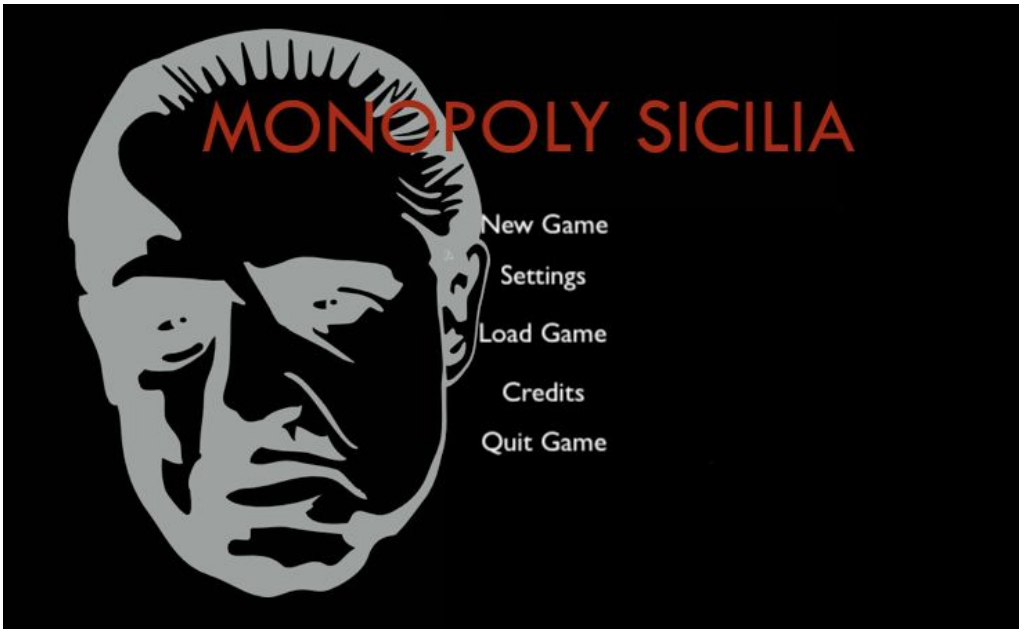

Figure-2a: Main menu

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 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.

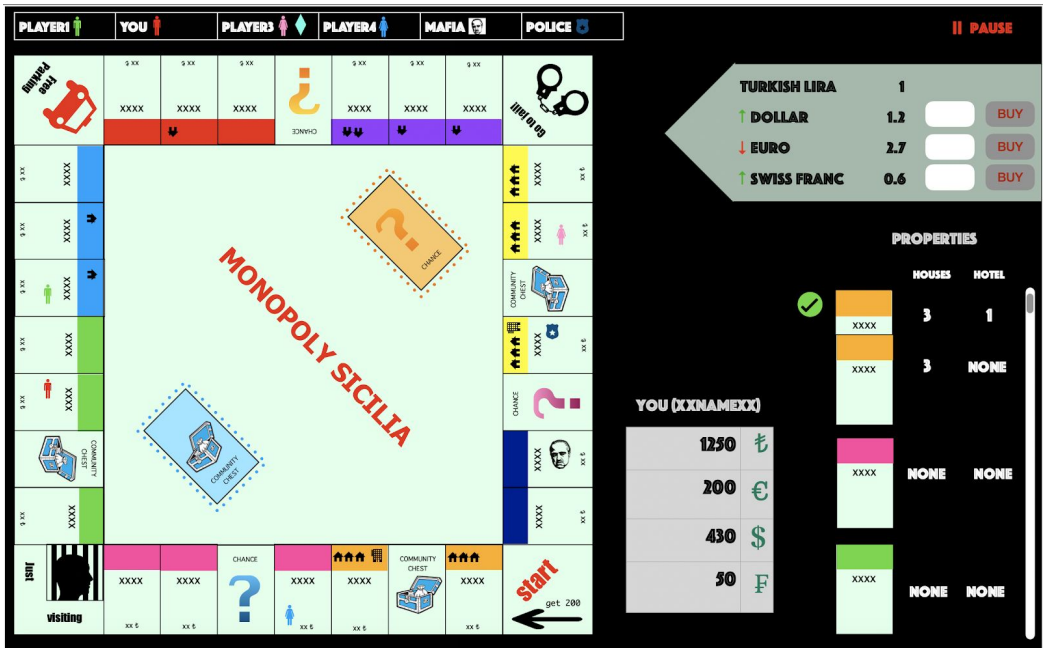

Figure-2b: Game play screen of monopoly.

The game screen when it is Player2's turn. On the top right corner, Player 2 may buy foreign currencies and between properties screen and the table, there is the currency accounts for the current player (Player 2 in this case). The properties section represent the owned properties of the current player. This section is scrollable to show more than three properties at once. When a player decides to buy a house or a hotel, 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-2c: Buy pop-up

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 white space inside each tile. 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. The players may navigate to the pause menu if they press the "Escape" button on their keyboard.

Figure-2d: Pause screen

In this screen, the players have the option to navigate to the settings screen, see the credits or go back to the main menu.