Bilkent University

Department of Computer Engineering

Monopoly Project Group 2J

Bilopoly: Monopoly's Bilkent version

Analysis Report

Asım Güneş Üstünalp, Mohammed S. Yaseen, Turan Mert Duran, Radman Lotfiazar, Mohammad Elham Amin

Instructor: Eray Tüzün
Teaching Assistants: Barış Ardıç, Emre Sülün, and Elgun Jabrayilzade

Contents

Analysis Report

Bilopoly : Monopoly's Bilkent Version

1. Introduction

Monopoly is one of the most popular board games in the 20th and 21th centuries. Monopoly is adopted from The Landlord's Game which was invented in the United states in 1903 by Lizzie Magie. Monopoly was published in 1935 and until now approximately 275 million sets have been sold in 43 languages in 111 countries. It has different editions which leads to increase its popularity including Star Wars, The Lord of the Ring and SpongeBob SquarePants. Based on South China Morning Post Monopoly is popular among people because "it gives them a chance to handle money and make adult investment decisions, practising skills of investment and negotiation that are critical in adult life, but with no danger of real loss" [1]. Therefore, it is a common board game not only for youngsters but also adults who want to practice their different abilities including investment, negotiation and decision making without any danger.

Bilopoly is computer based which we are planning to change and add different rules to the games. However, the main rules and features are not changed, including the fact that it is played with two six-sided dice to move around the game board and players collect rent from their opponents in order to drive them into bankruptcy. In our version of Monopoly we are trying to design Bilkent themed properties instead of the original properties. Furthermore, different features and rules, which are designed in this digital version of the game, lead players to feel different circumstances than the original and other versions of Monopoly.

2. Overview

Our proposed system is a digital version of Monopoly which lets people run it in their Windows machine. After running the game the first page is the menu page. Menu includes buttons for starting a new game, load game, settings and about us. By pressing the Start a new game button the application will ask the users about the number of players and the colors and nicknames which they would choose for representing them during the game. Load game will ask which one of the games you have saved you want to continue. In setting part users are able to increase or decrease the sound. And in about us we will provide concise information about the game and our group.

2.1. Game Play

In our new version of Monopoly, players start the game by pressing the New game button or load a game which they have saved before by pressing the load game button, users have a chance to continue their game from where they left.

If they press the New Game button, they will choose the number of people who will play the game. After that, for each player colors and nicknames should be decided. For initial startings, each player rolls dice and they are arranged according to their dice results. Biggest dice get the first turn and others follow the queue according to their dice results. All of the players start from the beginning location, which is stated on the map. Each player rolls the dice when their turns come. According to their dice results they move and pass places.

When they come to a place and if the place has not been bought by anyone, they need to decide whether they will buy this place or not. Each place has its own market value that was designed before by game makers. The user can buy the place by paying the mortgage value of the place to the bank.

If the place that user came has not been bought by anyone, the user has two options. The user can choose to just do nothing and wait for her/his next round or

reroll dice by paying the rerolling payment value that is determined by game algorithms.

If the user has already bought the place, the player can buy vending machines or cafes(similar to houses and hotels from the original game), which increase the payment value of the location, for their places. Values of each place can be found on places' information cards.

If the place that player came has already been bought by someone, users have two options. One of them is that the user who came on someone's property has to pay money to the owner of the place according to the includings (vending machines and cafe) of the place. Second option is that the player can choose to roll dice. If the result of dice is double, the user escapes the payment and leaves the place without paying any money. However, if the user can not roll double dice, the user has to pay double the price. If the user has not enough money to pay, the user can sell his/her ownings to the bank. All of the places can be turned into the game money by selling them to the bank or other users. If the player has not enough owings and money, the player goes bankrupt. Thus, the others can continue their game with one missing player.

2.2. Map

Map of the game consists of 32 different locations. Of these, 20 of them will be Bilkent buildings. 4 of them will be different cafes that can be found on Bilkent University campus. 4 of them will be chance locations where users draw a chance card, they will be located symmetrically on each side of the map. There will be one prison where users visit prisoners. One "Go To Prison" location where people who come to this location will go to prison and stay there for 3 rounds. One "Start location" where players start the game from and take money from the bank for each passing of this location. And one location where all the taxes are collected, if a

player arrives at this location he/she takes all the money and taxes that have been paid to the bank and accumulated on that location.

2.3. Settings

In this part the user is able to change the sound effects of the game. Games include different sound effects such as the music of the game, the sound of movement when players move around the board and different sound effects belonging to property and chance locations. The volume of all these sound effects are changeable by accessing this part. Furthermore, setting is not only available after running the game but it is also accessible during the game for changing any of these sounds.

2.4. Properties

There are 20 number of Bilkent buildings where players are able to buy these buildings and build vending machines or cafes there for increasing the rent of the place. These buildings are:

1. Electric Electronic Engineering (EEE)
2. Faculty of Engineering (EA)
3. Faculty of Economics, Administrative and Social Science (A)
4. Faculty of Science (S)
5. Library Main Campus
6. Faculty of Art, Design and Architecture (F)
7. Faculty of Law (B)
8. Faculty of Humanities and Letters (H)
9. Faculty of Business Administration (M)
10. Faculty of Music and Performing Arts (P)

11. Vocational School of Tourism and Hotel (R)
12. Nanotechnology Research Center
13. Sport Center Main Campus
14. Register's Office
15. Health Center Main Campus
16. Vocational School of Computer Technology and Office Management (G)
17. Student Dormitories
18. Bilkent ODEON
19. School of English Language (C)
20. Meteksan Bookstore

Moreover there are 4 different cafes' buildings which players are able to buy and earn money by renting them. These cafes are:

1. Speed-Kirac Cafe
2. Starbucks
3. Mozart Main Campus
4. Cafe Break Main Campus

2.5. Chance cards

There are 4 chance locations where players are able to draw a chance card. These cards give different opportunities or punishments to players. For opportunities we can mention, a "Get Out of Prison" card, win an amount of money card, don't pay the rent for this round card etc. Punishments include, lose an amount of money card, go to prison card, pay the double rent for this round card etc.

Furthermore, players are able to use their chance cards whenever they want, they can use them instantly or use them in the later rounds. They can also trade their cards and sell them to other players.

2.6. Transfer money and properties between players

Players in our version of Monopoly are able to transfer money and properties between themselves. However, these transactions will include some taxes which players have to pay. For transferring money only players who receive the money have to pay %8 of the money to the bank and pay back the whole money whenever the players who give the money ask for that. If players, who receive the money, have not that much cash for paying back the other player is able to ask for properties even if it is worth more than that money they borrow. Nevertheless, For transferring properties both plates have to pay 5% of the price of properties which they receive as tax.

2.7. Invest money

Players are able to invest their money in the bank and get the benefit back. Based on the amount of money and number of rounds they decide to invest, the bank will decide how much money will pay back after finishing that period. These percentages will calculate by application and inform the players every round. These percentages might stay unchanged or change during the game based on the decision of application.

3. Functional Requirement

3.1. New game

3.1.1. Selecting tokens and colors

At the start of the game, players will select a token and a color each that will represent them on the board. A player's current location will be displayed

by the token and a player's owned properties will be displayed by coloring those tiles with the owner's color.

3.1.2. Track money

At the start of the game, every player will be given some amount of money. Players will be able to use their money to buy property. They will also pay to the other players and to the bank by using their money. That's why the game needs to keep track of every player's money.

3.1.3. Position

At the start of the game, every player starts at the "Start Location" which is located on one of the corners of the map.

Every turn, players roll the dice and move forward the amount of tiles according to their dice rolls (e.g. 10 tiles for a 6 and 4 roll). The game needs to keep track of the position of every player's tokens for the game to properly function.

3.1.4. Chance cards

The game board has 4 locations in which incoming players draw a random chance card from a predefined deck of cards. The game needs to have a deck of cards for the players to draw from and the game also needs to be able to automatically give a random chance card to any player who landed on a chance location.

3.1.5. Buying Property

The game board features 24 locations which the players can purchase (20 regular Bilkent buildings, 4 special cafes found on the campus). When the players arrive at those locations, if those locations hadn't been purchased by

any other player before, the arriving player should be able to purchase those locations.

3.1.6. Inventory

As described above, the players will accumulate chance cards and properties by arriving at the chance locations and by purchasing respectively. The game needs to hold an inventory for each player to hold his/her belongings. It is needed because the players can trade with each other. This is explained in the next point (3.1.7).

3.1.7. Trade

The players need to be able to trade with each other. They should be able to trade money, property or chance cards in any combination they want with each other. For example, a user should be able to sell his/her chance card to another player for a property + some amount of money.

3.1.8. Paying Rent

The purpose of purchasing property for the players is to collect rents from that property when other players arrive at it. The game needs to be able to automatically transfer the rent money from the arriving user's inventory to the property owner's inventory. The amount of rent is determined by some factors. The game needs to calculate the rent based on those factors which are:

1. The original price of that property. If the property was expensive for the property owner to buy, it should generate more rent.
2. Whether the group of properties that this property belongs to are all owned by the same person. If the same person owns all of them then he/she collects more rent from them.
3. Whether the property has any vending machine or cafe on it. If it does, then the rent would be higher.

4. Whether the property owner used any chance card to increase his/her rent income or whether the arriving player used any chance card to decrease his/her rent expenses. For example, there exists a chance card that says “Pay less next time you need to pay rent”. If the arriving player uses that card then he/she pays less.

3.1.9. Rerolling the dice

In their turns, the players should be able to reroll the dice when they are unhappy with its results by spending money.

3.1.10. Game Turns

The game is a turn based one. That means every player performs all their actions on their turns. They then pass their turns for the next player to play. The only action a player makes outside their turns is to whether accepting or declining a trade deal made by the player who has the turn which is an exception.

3.1.11. Building Vending Machines and Cafes

If a player owns all the properties of a group, they should be able to build vending machines or cafes if they have enough money to do so. Before building a cafe, a player first needs to build 3 vending machines on that property. Vending machines and Cafes increase a property’s rent.

3.1.12. Investment

The players should be able to invest their money in the bank. The bank would have a fixed interest rate and the players would gain money based on that interest rate.

3.2. Save game

While playing the game, players may use this feature to save the current state of the game. Current progress of the game is saved for later use and the players would be able to load the saved game and continue. This action is done automatically in specific intervals. Hence, in case of any crash or error, the players will be able to continue from where they left off.

3.3. Load game

At the beginning of the game, players will be prompted with the options for playing a new game or to continue on with a previously saved game, if any exists. This option is important for giving the user a good experience and flexibility.

3.4. Settings

3.4.1. Change sound volume

Gives the user the ability to increase/decrease or turn off the game sound.

3.4.2. Graphics quality

Gives the user the ability to change the graphics quality between high, medium, and low. This is used to improve the game performance while running on low specifications computers.

3.4.3. Battery Life

Gives the user the ability to prioritize higher performance or better battery life. This option is intended to reduce performance to maximize the battery life for laptops.

3.4.4. Play time while on battery matric

Show the user the estimated time they can play while they are on battery.

3.5. How to play

The players may access information screen about how the game is played while playing the as well as before starting the game. Players will be able to learn:

- How the game is played
- What are the rules
- Buttons explanation
- Some UI elements explanation

This screen is important for new players to learn about the game rules and easily play the game. For example, players can learn how to invest in the bank by using information provided in this screen.

3.6. Quit

Allows the user to exit the game. Prompts the user asking if they want to save the current session if there was a game in progress.

4. Non-functional requirement

4.1. Operational Requirements

4.1.1. Reopening the game after the crashes

Having the game crashing in front of them is already a frustrating experience for the users. When they reopen the game it needs to open the latest open session automatically in order not to frustrate them further.

4.1.2. Game Size

Game size is an important issue for the users who don't have much space on their storage devices. To not crowd the storage devices of our users, the game size needs to be small. The size of the game shouldn't exceed 150 MB.

4.1.3. Save Time

After playing the game and before quitting, users wouldn't want to wait long for the game to save. That's why the game should save in less than 5 seconds.

4.1.4. User friendly UI

Having an user friendly UI (User Interface) is important for the users because otherwise they might feel confused and thus feel frustrated. In order not to make them feel confused, there should be at most 10 pressable buttons on the screen at once.

4.1.5. Screen resolution support

The game should run on screens with resolutions: 1366×768, 1920×1080, 1536×864, 1440×900, 1280×720, and 1600x900. These screen resolutions represent 68.14% of the computer screens available in the market [2].

4.1.6. Performance

The game should be high performant in multiple aspects including:

4.1.6.1. Frame Rendering

The frame rendering is an important aspect since it measures what the player is seeing on the screen. It is measured using frame per second (FPS), especially FPS madian. The goal is to run the highest FPS possible; however, the FPS the game provides should be no less than 30 FPS.

4.1.6.2. CPU Usage

The game should maintain low cpu usage, not to crowd the cpu, which causes the computer to slow down. In general, the game can use between 0-100% of the cpu at any time; however, The game should not use more than 70% of the cpu for longer than 5 mins at once. The frequency of these occurrences when the game is using more than 70% of the cpu should also be infrequent: less than twice each 30 minutes.

4.1.6.3. GPU Usage

GPU is where the game renders graphics and shows them on the screen. The game should maintain low GPU usage because intensive GPU usage can cause the FPS to drop and might cause some sluggishness, which in turn affects the user experience negatively. The GPU usage should be adjusted by possibly reducing the resolution of graphic assets to maintain the minimum frame rate of 30 FPS.

4.1.6.4. Battery Usage

Battery life is an important aspect for users using the game on laptops. The game should not drain the battery very quickly and should maximize the play time while the computer is on battery. In general, the game should not increase the battery drain speed by more than 50%.

4.1.6.5. Load Time

For the sake of better user experience the game should load very quickly. It is shown that users in general expect their computers to load faster than their phones or webpages. Additionally, around 66% of the users are willing to wait more than 6 seconds for the mobile or web page game to load [6]; therefore, our desktop game is required to load in no more than 5 seconds.

4.1.7. Player mode

The game should support multiplayer mode. The number of players allowed should range between 2 and 8.

4.2. Development Requirements

The code, architecture design, and diagrams should have the following qualities:

4.2.1. Scalability

The degree to which to which the system facilitates the addition of new features. This quality needs the system to have a clear architecture and loosely coupled components. Any new feature should have only one logical place to be added to.

4.2.2. Testability

"The degree to which a system or component facilitates the establishment of test criteria and the performance of tests to determine whether those criteria have been met"[4]. This is achieved by having the components being loosely coupled and not dependent on each other that each can be tested individually.

4.2.3. Portability

"The ease with which a system or component can be transferred from one hardware or software environment to another" [3]. The system should be loosely coupled and connected to the exterior systems using templates or adapters to reduce the effort when migrating to other systems.

4.2.4. Maintainability

"The ease with which a software system or component can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment" [5]. The system should be loosely coupled that a bug can happen in only one component that is not dependent on any other components. This facilitates the bug fixing since one one component should change and no other components of the system will be affected.

4.2.5. Understandability

The ease with which the system is understood by other programmers. A new programmer should be able to understand the general architecture of the game in less than half an hour and start working in a few hours.

5. Pseudo requirements

5.1. OOP Language

This project is restricted to using a programming language with reasonable object-oriented programming.

5.2. Game Engine Use

Using any library, framework, or any game engine that forces a particular design on the project is strictly prohibited.

6. System Models

6.1. Behavioral Diagrams

6.1.1. Use-Case Model. Figure (6.1.1)

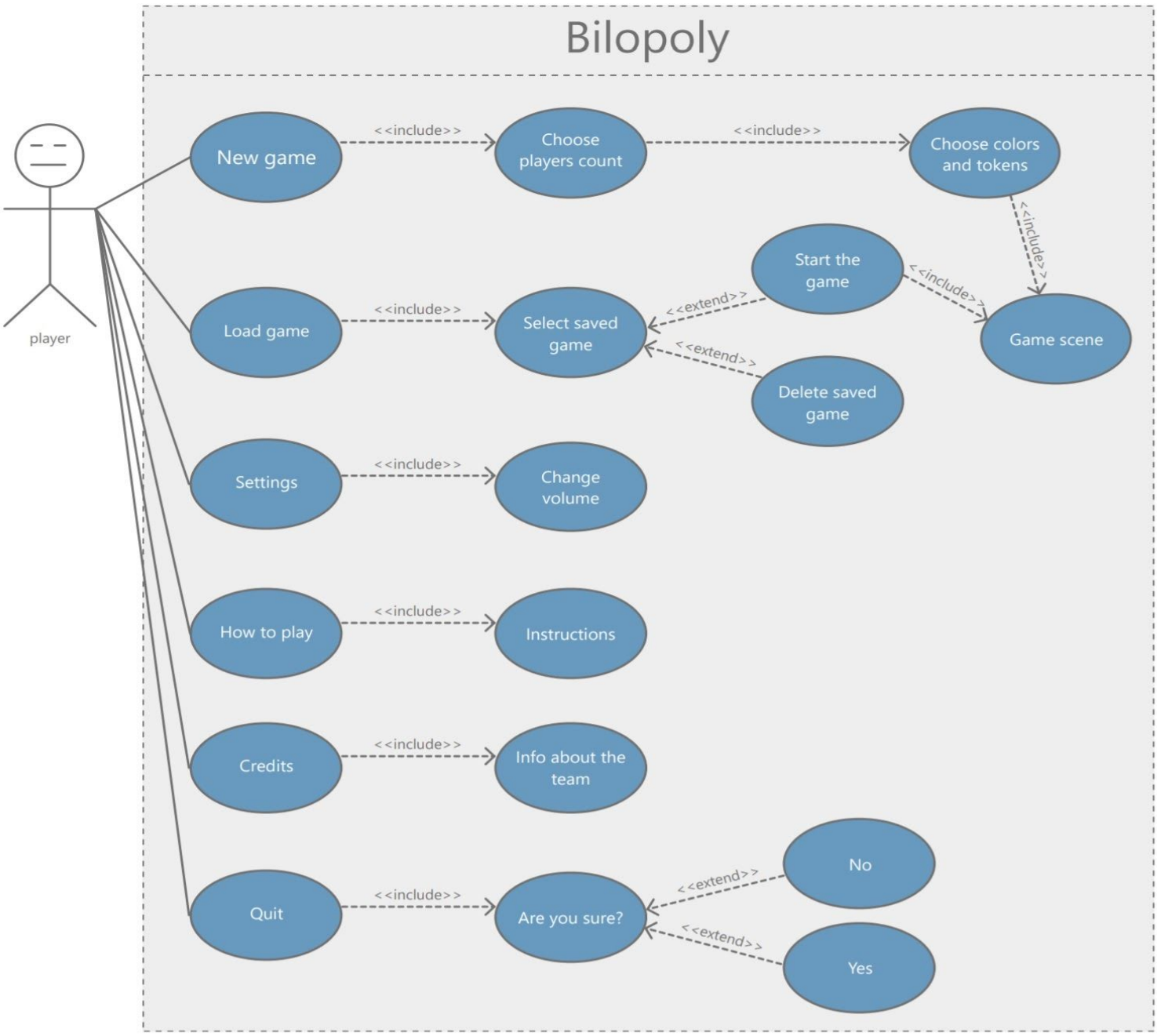

Figure (6.1.1)

6.1.1.1. Play game

Actor: Players

Action:

- Players want to play the game
- Application will create the new game and start it

Pre Condition:

- Players need to be in the main menu and click the “New Game”

Post Condition:

- Players should chose their nickname
- Players should chose their color
- Press the play button for starting the game

Entry Condition:

- Players have to roll the dice to decide in order to decide who is first, second and ect. Player for playing the game

Exit Condition:

- Player can pause the game by pressing ESC button or clicking the icon which is design in the game for quit the game
- One person remains in Bilopoly who is not bankrupt

Success Scenario Event stages:

1. Start the Application
2. Pressing the “New game” in the Main Menu
3. Choosing nicknames and color for players
4. Roll the dice the highest will be first person and other rank will be second, third and ext.

5. First person roll the dice again and start the game
6. Going to somewhere in the board based on the dice
7. Decide what to do based on the property
8. Other players do the 5 to 7 stages
9. Players try to bankrupt each other
10. Winner is the last person who is not bankrupt
11. Player finish the game
12. When one person remains in the game it is finished

Alternative Event Stages:

1. Players are able to pause the game
    a. Pause the game by pressing ESC or icon in the game
    b. Select Quit Game
    c. Application will ask do you want to save the game
    d. System quit the game and come to the Main Menu
2. Players are able to save the game
    a. Players pause the game
    b. Select the save Game
    c. They chose the name for the save file
    d. They will continue the game

6.1.1.2. Puse Game

Actor: Players

Action:

- Players want to pause the game
- Players want to change the setting
- Players want to save the game
- Players want to quit the game

Pre Condition:

- Players must be in the game moad

Post Condition:

none

Entry Condition:

- Players must press the ESC on the keyboard
- Or Players can pause the game by clicking on the icon for pausing

Exit Condition:

- By clicking everywhere instead of option on pause menu can quit

Success Scenario Event stages:

1. Pressing ESC on the keyboard or click on the pause icon in the game
2. Player is able to change the sound setting by pressing the setting icon
3. Players are able to save the game by pressing the “Save Game”
4. Player are able to back to game after saving or whenever they want to continue
5. Players are able to quit the game

Alternative Event Stages:

1. Players are able to back to game in two options
    a. By clicking everywhere instead of option on pause menu
    b. By pressing ESC button

6.1.1.3. Load Game

Actor: User

Action:

- User wants to see they saved games
- User wants to continue saved game
- User wants to delete the saved game

Pre Condition:

- User must be in the Main menu

Post Condition:

none

Entry Condition:

- User must click on the “Load Game”

Exit Condition:

- By clicking “Back” button on the screen he can come back to the main menu

Success Scenario Event stages:

1. User press the “Load Game” option on the screen
2. User is able to see all his/her save games’ information including (name, date and time, nicknames of players and ext)
3. User press on one of the saved game and system gives two options
    - Continue the game
    - Delete the saved game
4. User is able to delete or condition the game

Alternative Event Stages:

1. Users see their previous saved game and want to come back to main menu
    a. User press the “Back” button and come to the Main Menu
    2. User delete the saved game and want to come back to main menu
    a. User press the “Back” button and come to the Main Menu

6.1.1.4. How To Play

Actor: User, this can be player or user it does not different in this user case
Action:

- User wants to know how should play Bilopoly
- User wants to know what are the rules in this versions of the Monopoly
- User Want to know what are the difference between Bilopoly and Monopoly
- User want to find out about properties which is used in the game

Pre Condition:

- User must be in the Main menu

Post Condition:

none

Entry Condition:

- user must click on the “How to Play”

Exit Condition:

- By clicking “Back” button on the screen he can come back to the main menu

Success Scenario Event stages:

1. User is in the Main menu and wants to find out about How should play the game
2. User press the “How to play” button on the screen
3. User reads about Rules, Properties and different between Bilopoly and monopoly there
4. User press the “Back” button and come to the Main Menu

Alternative Event Stages:

none

6.1.1.5. Settings

Actor: User, this can be player or user it does not different in this user case

Action:

- User wants to alter the volume
- User want to change the dimensions of the game

Pre Condition:

- User must be in the Main menu

Post Condition:

none

Entry Condition:

- User must click on the “Setting”

Exit Condition:

- By clicking “Back” button on the screen he can come back to the main menu

Success Scenario Event stages:

1. User is in the Main menu and wants to adjust the setting the game
2. User press the “Setting” button on the screen
3. Users are able to increase or decrease the game music
4. Users are able to turn off the default sound for different activities including moving players in the game, starting the game and notifications and ext.
5. Users are able to change the dimension of the game from full screen to half or other option
6. User press the “Back” button and come to the Main Menu

Alternative Event Stages:

1. Users are able to alter the sound setting from “Pause Menu”
    a. Users go to “Pause Menu” by pressing ESC button or clicking icon in the game
    b. They can continue the game by pressing ESC or Clicking anywhere on the screen instead of option on the “Pause Menu”

6.1.1.6. Credits

Actor: User, this can be player or user it does not different in this user case

Action:

- User wants to know about Developers
- User want to know more about application

Pre Condition:

- User must be in the Main menu

Post Condition:

none

Entry Condition:

- User must click on the “Credits”

Exit Condition:

- By clicking “Back” button on the screen he can come back to the main menu

Success Scenario Event stages:

1. User is in the Main menu and wants to know about Developers or application
2. User press the “Credit” button on the screen
3. User reads about Developers
4. User is able to find out link of Github which includes documents and source code
5. User press the “Back” button and come to the Main Menu

Alternative Event Stages:

none

6.1.1.7. Quit

Actor: User, this can be player or user it does not different in this user case

Action:

- User wants to quit the Bilopoly

Pre Condition:

- User must be in the Main menu

Post Condition:

- After asking are sure for quitting press the YES button

Entry Condition:

- User must click on the “Quit”

Exit Condition:

- By clicking “No” after asking “Are You Sure?” user comes back to the Main Menu

Success Scenario Event stages:

1. User is in the Main menu and wants to Quit the Bilopoly
2. User press the “Quit” button on the screen
3. System Asking “Are you sure?”
4. By pressing on the “Yes” button they will leave the Bilopoly

Alternative Event Stages:

1. User press the “Quit” button on the screen
2. System Asking “Are you sure?”
3. By pressing on the “No” user come back to the Main Menu

6.1.2. Scenarios

6.1.2.1. Starting a New Game

Players want to create a new game. From the main menu they press New Game button which leads them to the new game menu. In this new game menu users choose player count, tokens and colors. After that their new game is initialized by the game initializer according to their decisions. Player manager initializes players, the board manager creates the map and controls changings in the map. Card deck manager initializes cards and inventory manager creates inventory for each player and holds all their properties. Game manager controls all these managers and provides persistence of the game. Figure (6.1.2.1)

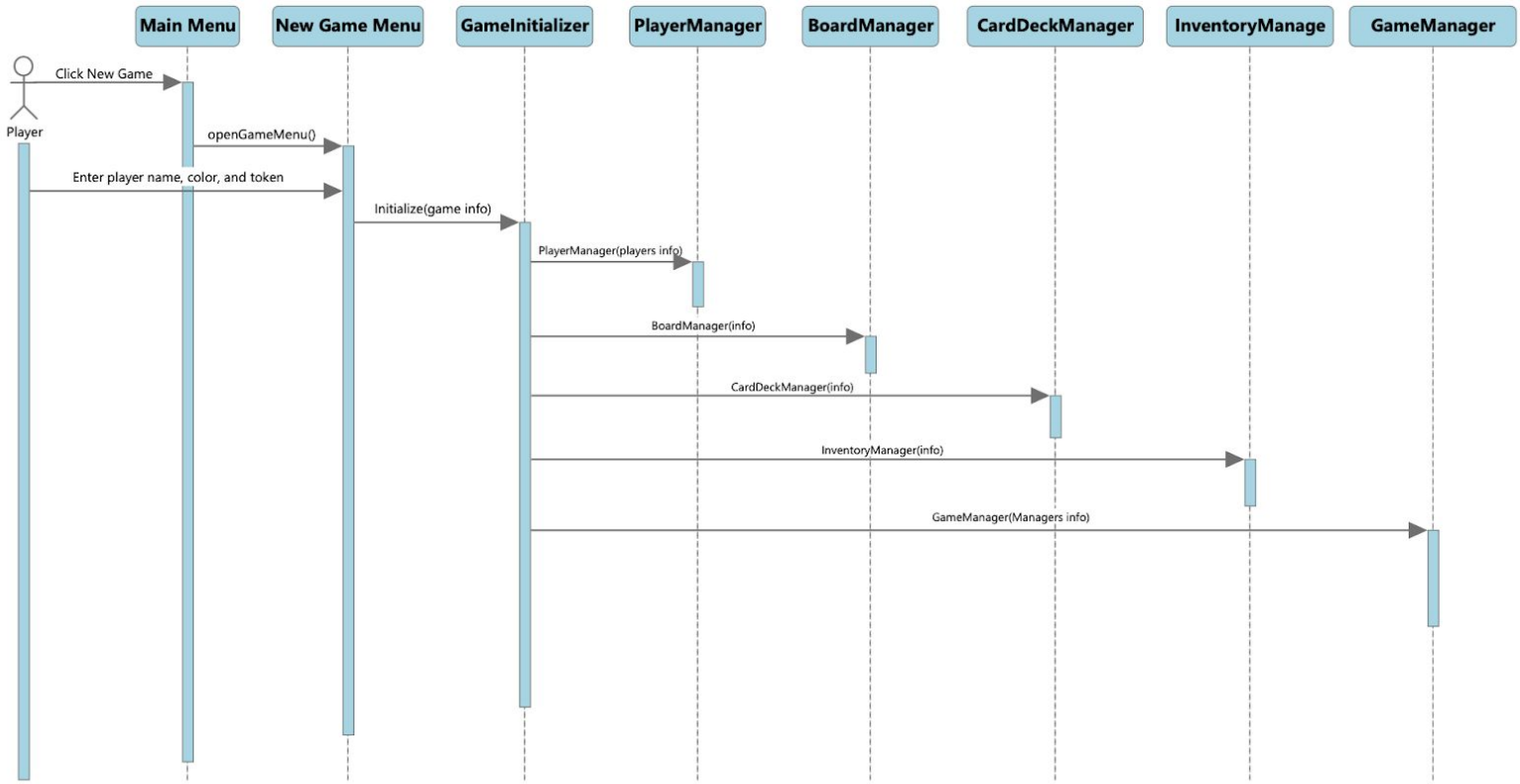

Figure (6.1.2.1)

6.1.2.2. Saving Game

Player wants to save the game to continue from where they left later. User presses the pause button on the game scene. Game stops. By pressing the save button on the pause menu, the save game prompt is opened. Player gives a name to the game which he/she wants to save. All the information is gathered by the game manager. The information is sent to the local data manager to save datas locally. The game is saved by the local data manager thus the scenario ends. Figure (6.1.2.2)

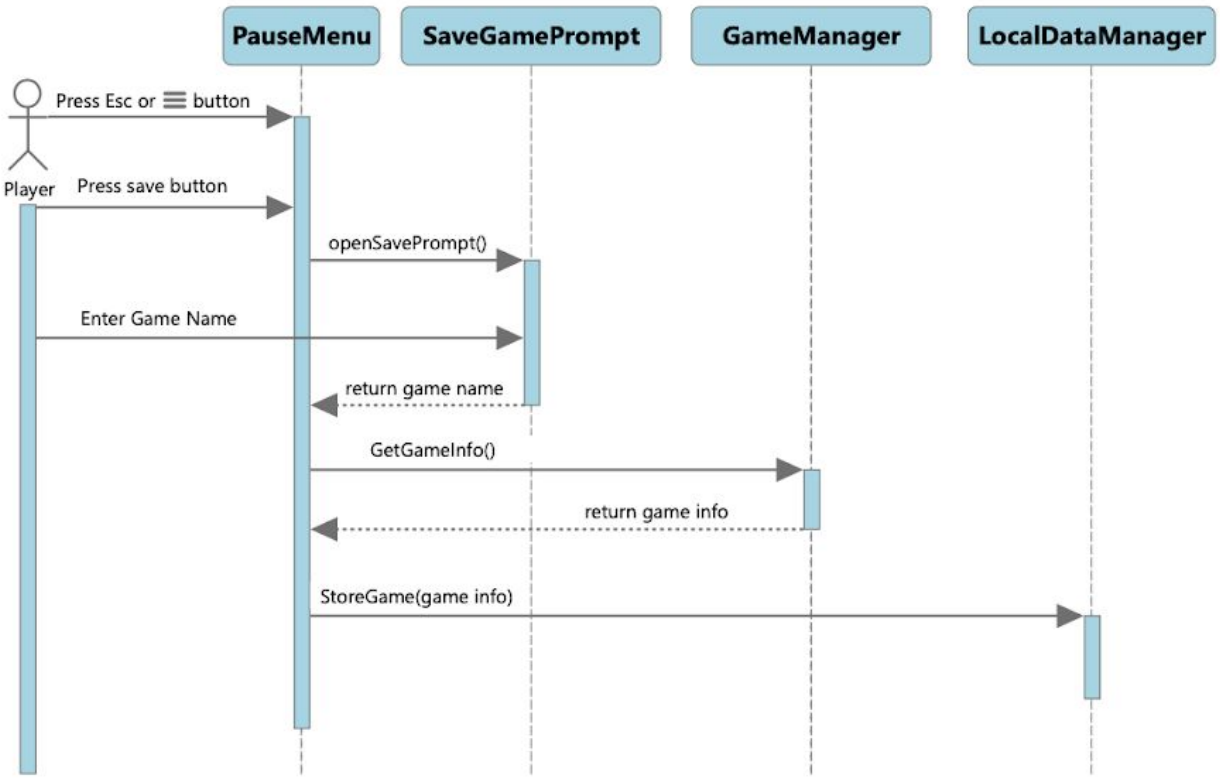

Figure (6.1.2.2)

6.1.2.3. Loading game

Player wants to continue the game which he/she saved before. User opens the game and from the main menu opens the load game menu by pressing the load game button. Local game menu goes to the local data manager and gets all stored games and lists them to the user. Thus, the user chooses the game which he/she wants to continue. By selecting the game according to the name that is given during the saving scenario and pressing the play button, the game is initialized by the information that is gathered from the local data manager. For initialization, the game initializer creates all the managers (player manager, board manager, card deck manager, inventory manager, game manager) that are needed. Thus, the user continues from where he/she left and the scenario ends. Figure (6.1.2.3)

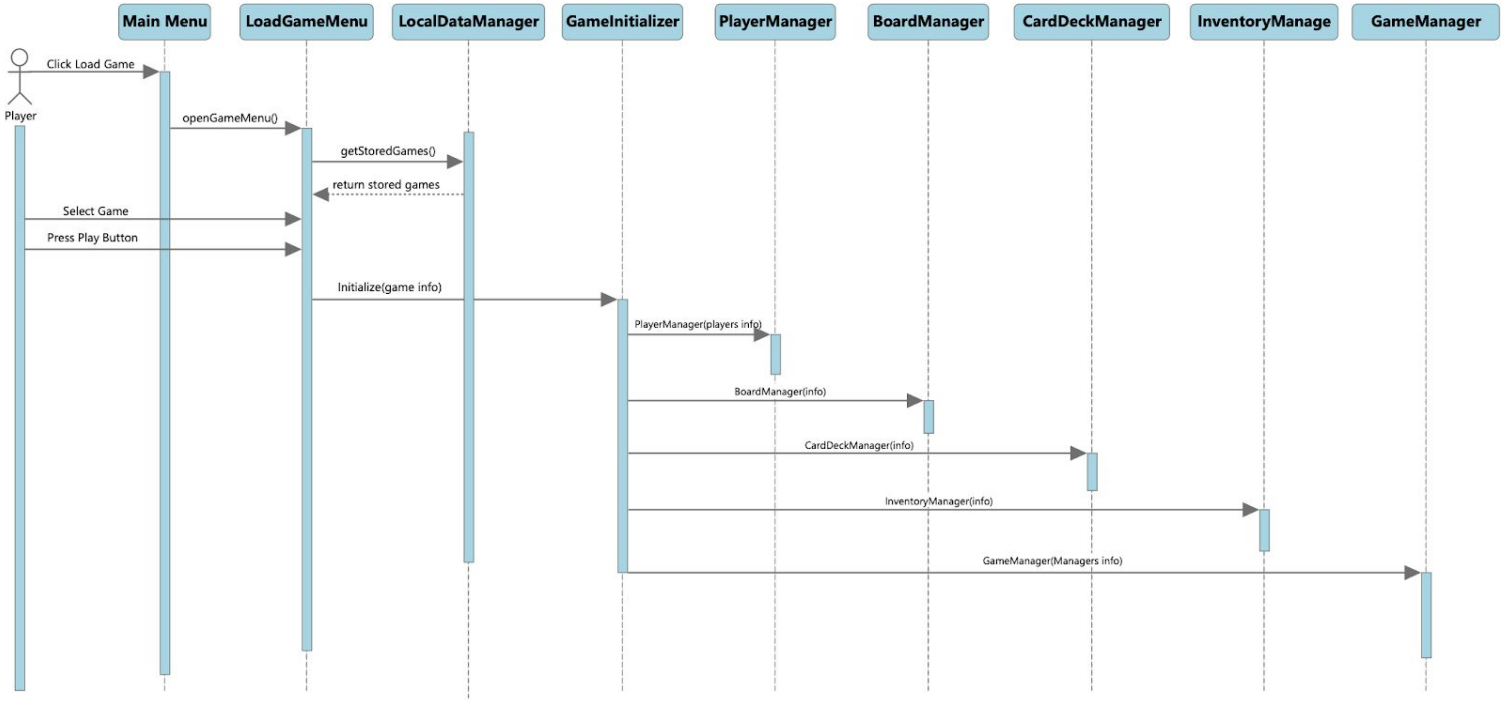

Figure (6.1.2.3)

6.1.2.4. Increasing Sounds Volume

Player wants to increase the volume of effects and background music of the game. From the main menu user opens the settings menu by clicking the settings button. By dragging the volume slider which can be found on the settings menu, the user increases the volume. Settings menu sends changings in volume slider to the sound manager. Thus, the sound manager increases the volume and the scenario ends. Figure (6.1.2.4)

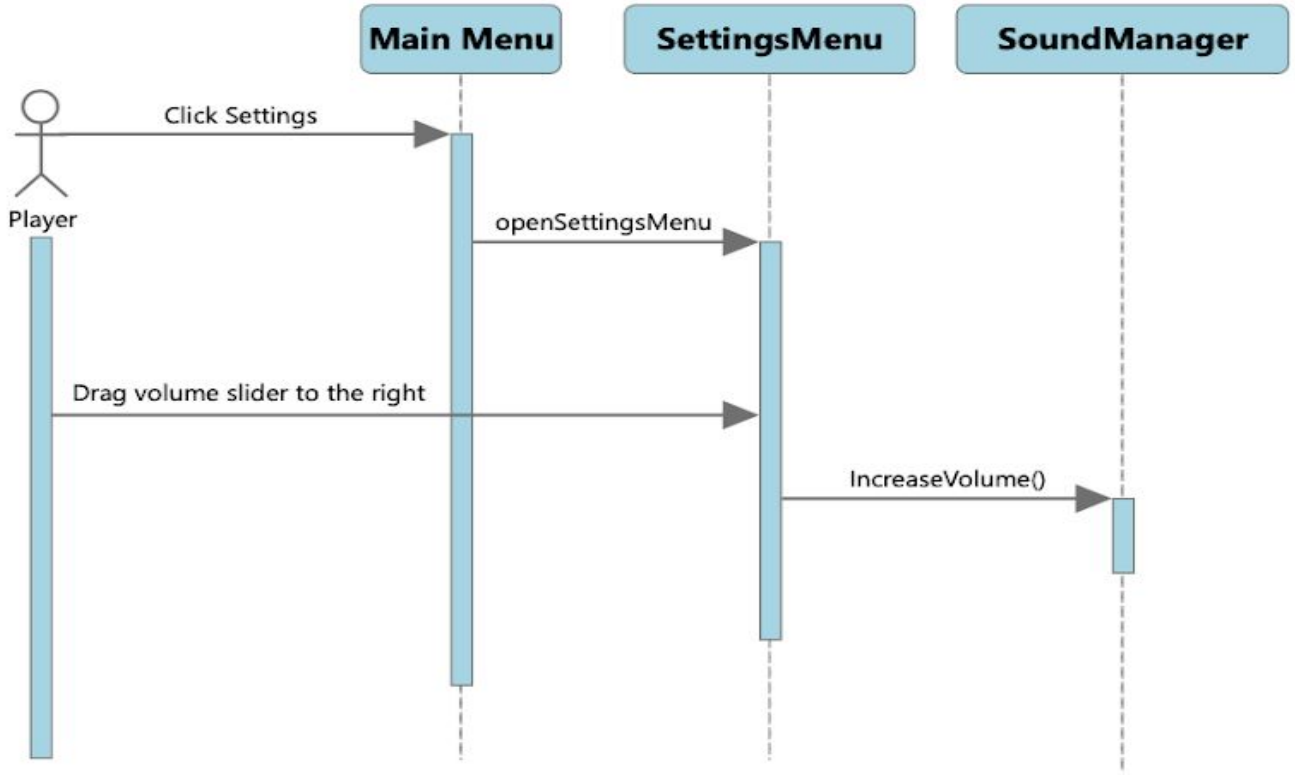

Figure (6.1.2.4)

6.1.2.5. Quitting

Player wants to quit the game. User presses the quit button from the main menu and the confirm prompt is opened. If the user still wants to quit and he/she is sure, by pressing the yes button on the confirm prompt the main menu calls the quitGame() method and the game is closed and the scenario ends. However, if the user presses no button on the confirm prompt, the user returns the main menu and the scenario ends. Figure (6.1.2.5)

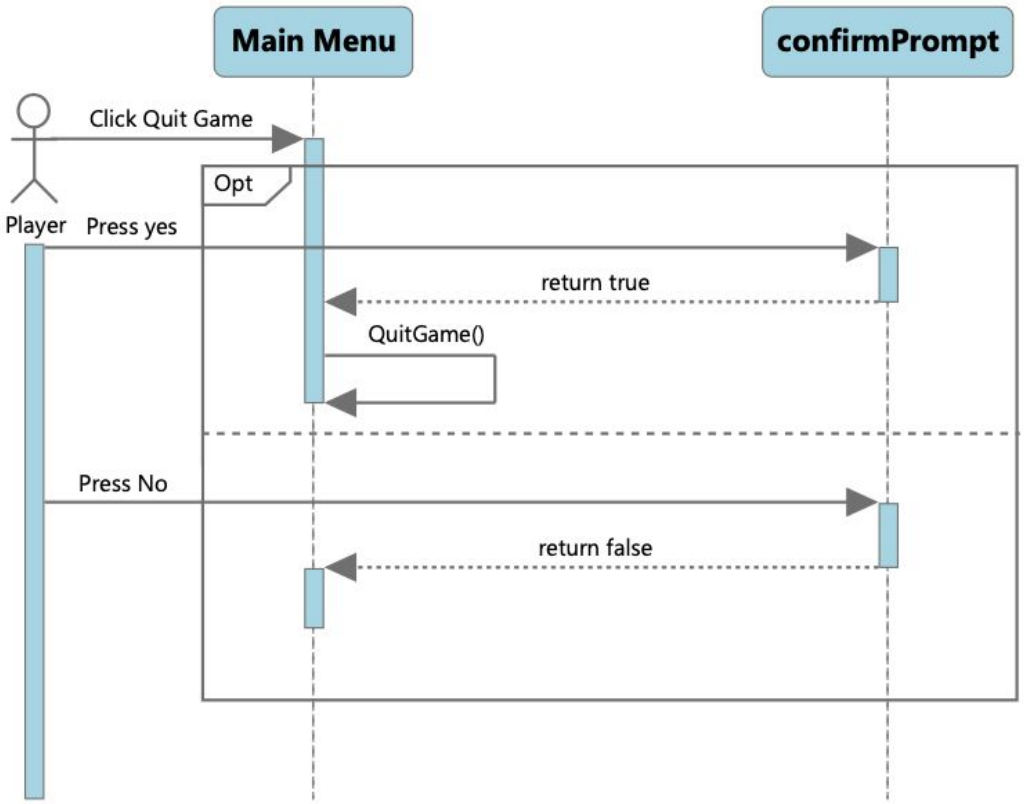

Figure (6.1.2.5)

6.1.2.6. Deleting Saved Game

Player wants to delete one of the games which he/she saved locally before. From the main menu, the user presses the load game button. Main menu opens, load game menu and load game menu calls the local data manager to get all stored games before. The Load game menu lists all these stored games to the user to choose which one of them he/she wants to delete. User selects the game from the list and presses the delete game button. Load game menu calls confirm prompts and ask for user confirmation. If the user presses enter, confirm prompt returns true to the load game menu and load game menu goes to the local data manager and calls delete game method with appropriate id of the game. Local data manager returns true and deletes the game, this means that the game is deleted and the scenario ends. If the user presses false on confirm prompt, none of the games are deleted and returns to the main menu thus, scenario ends. Figure (6.1.2.6)

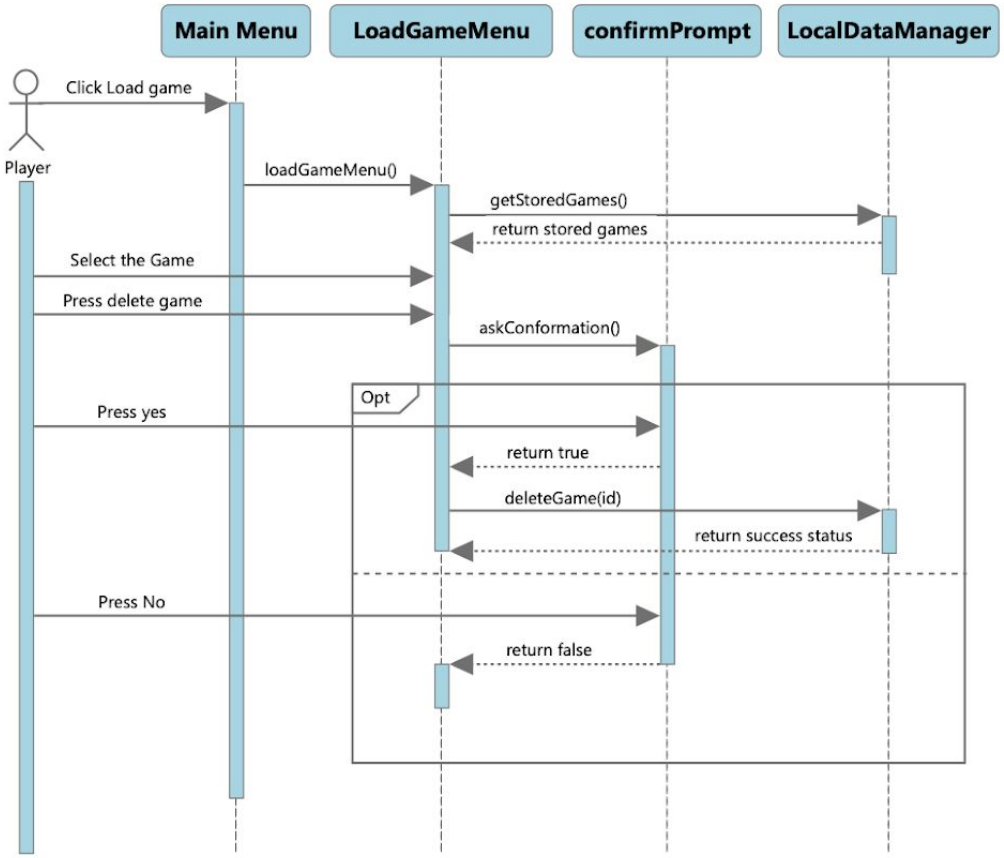

Figure (6.1.2.6)

6.1.2.7. Playing a Turn

Player wants to play his/her turn. From the game scene, the user firstly presses the rolling dice button. This button returns dice results and according to them the player moves on the board. This movement is handled by the board manager. When these are happening, the game manager triggers the sound manager to play movement sounds. New location of the player is sent to the player manager by the game manager. Inventory manager checks whether this place has been sold or not. If the place is still not sold, the inventory manager triggers a prompt and asks

the user for whether the user will buy this place or not. User makes his/her choice then after all these sale transactions are done the user ends his/her turn by pressing the end turn button and the scenario ends. Figure (6.1.2.7)

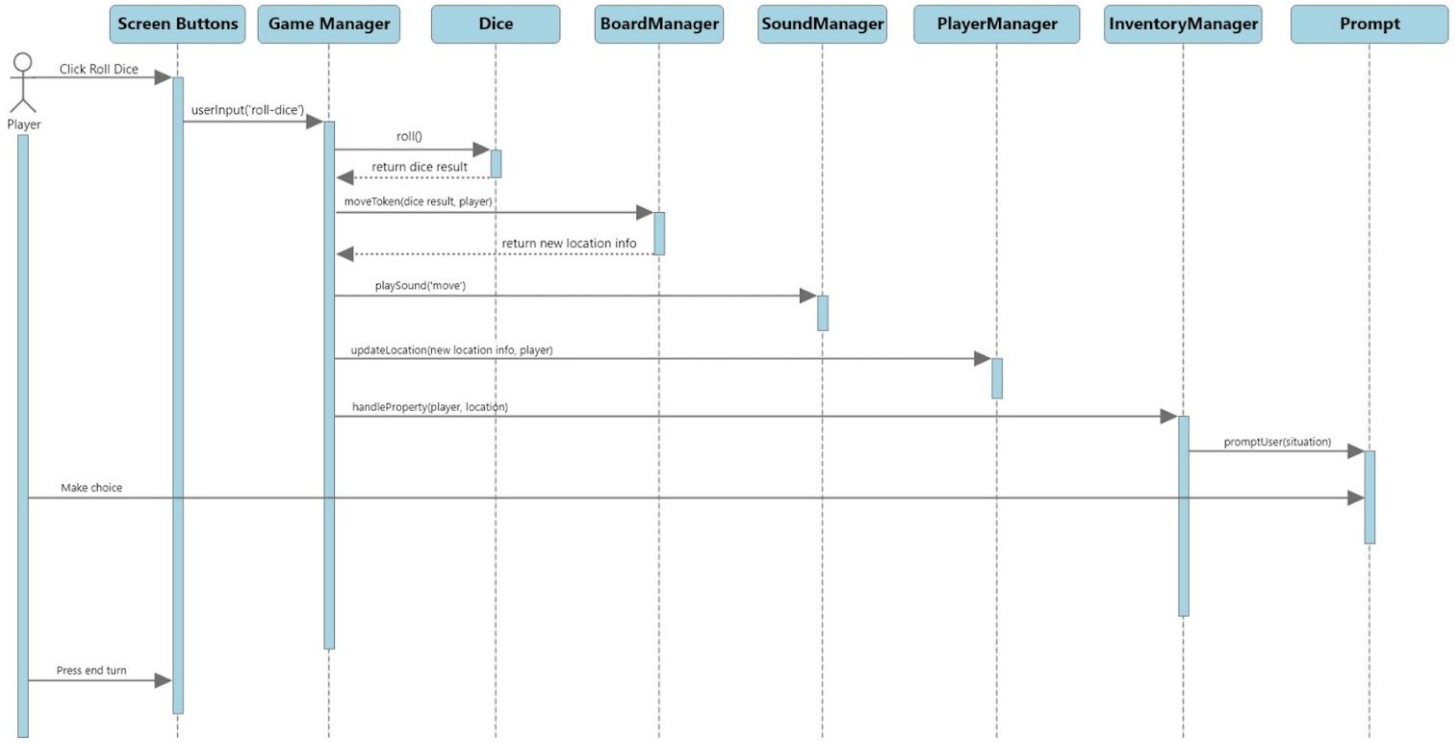

Figure (6.1.2.7)

6.2. Structural Diagrams

6.2.1. Object and Class Model. Figure (6.2.1)

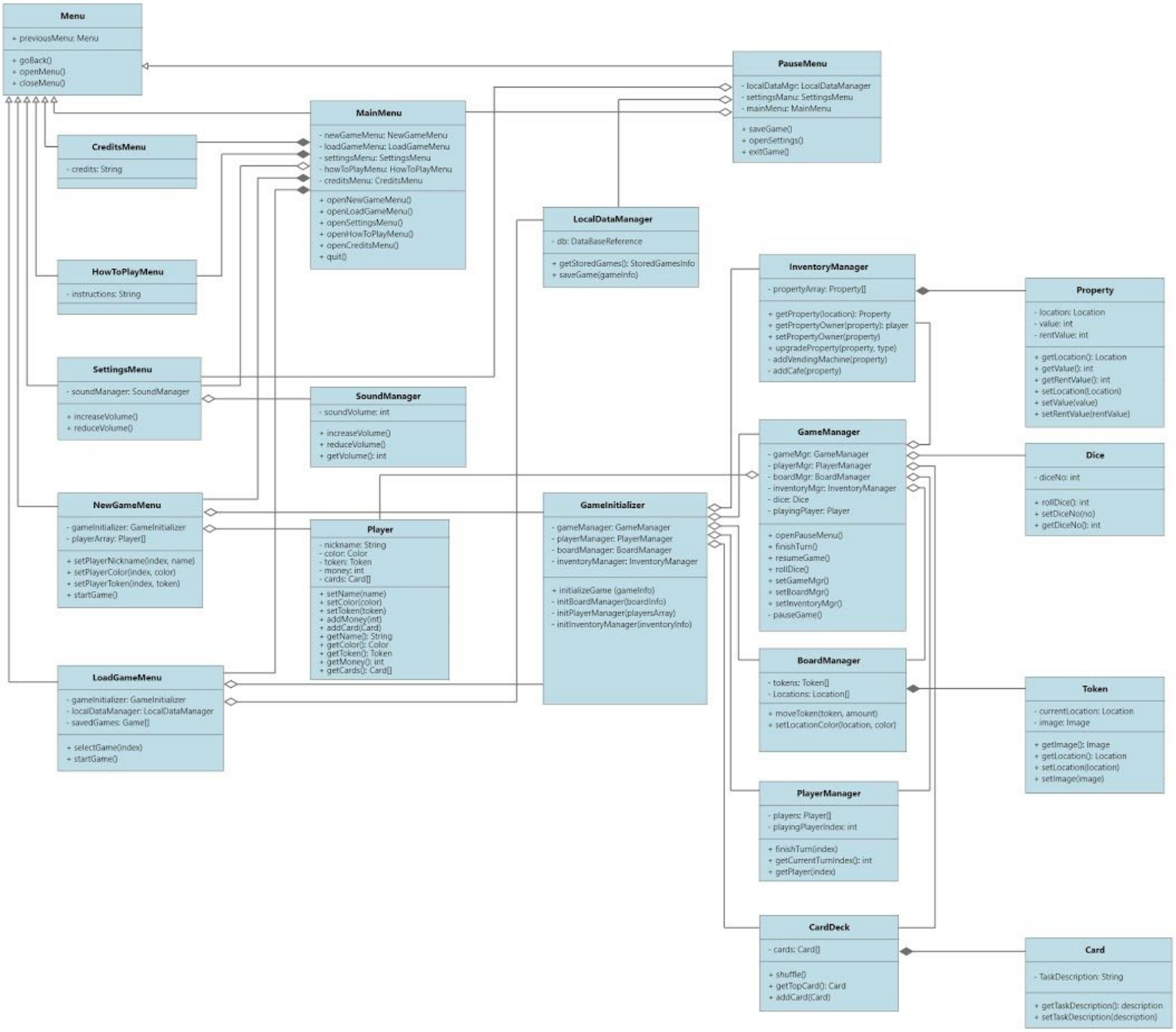

Figure (6.2.1)

6.3. User Interface

6.3.1. Main Menu

This is the first page which users or players see in the Bilopoly. This page lets the user start the New Game or load his/her previous game and continue his game or delta them. Moreover, players or users are able to learn how they should play and what are the rules for this version of the game. Furthermore, there is an option in this page Credits which gives information about our team. Setting and quit also there are in this page which users are able to use. Image (6.3.1)

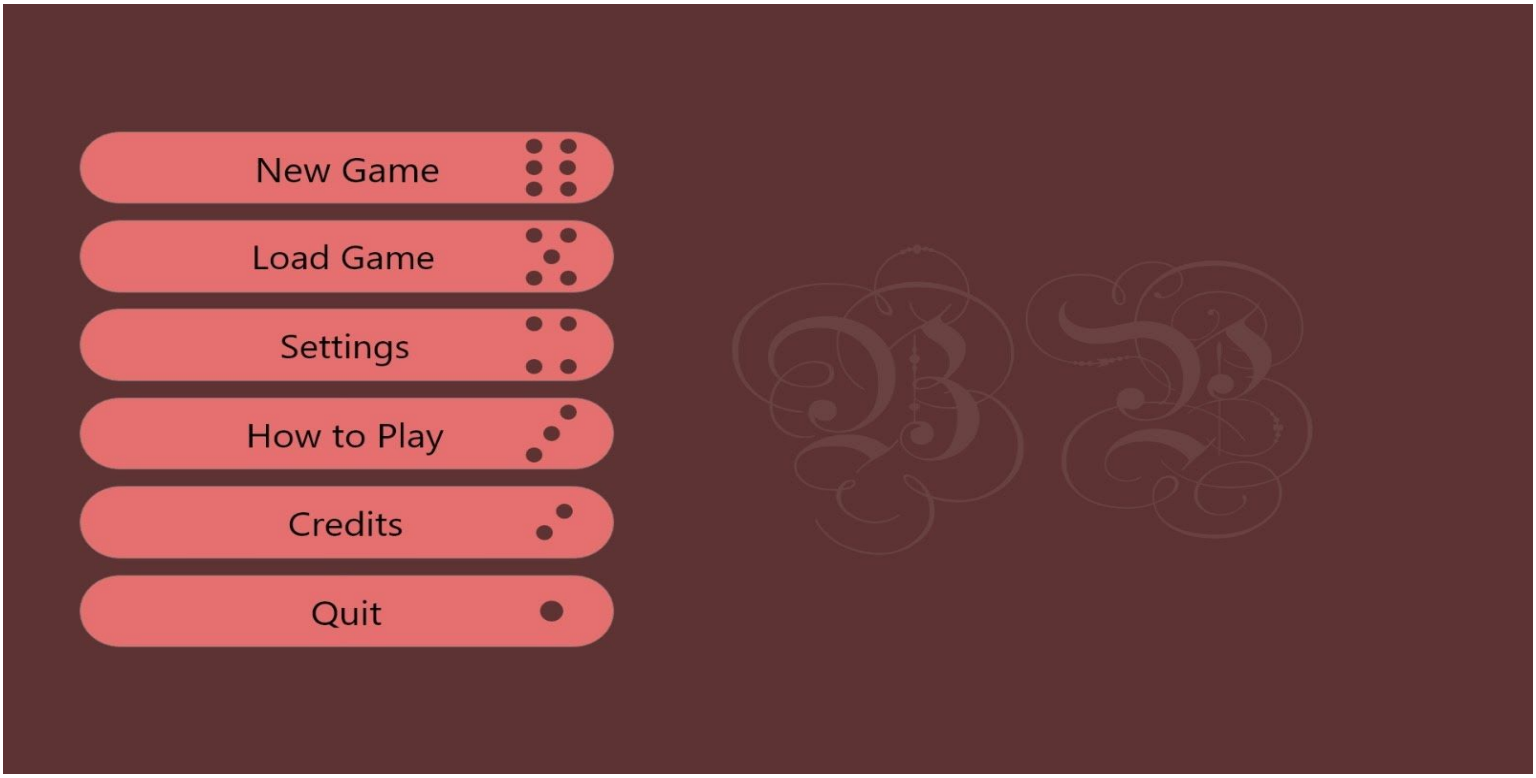

Image (6.3.1)

6.3.2. Load Game

In this page users are encountered when they want to delete their previous saved game or continue that. In this page shows the name which user has chosen for his/her saved game. Moreover, more details about the game is seeable including date. Image (6.3.2)

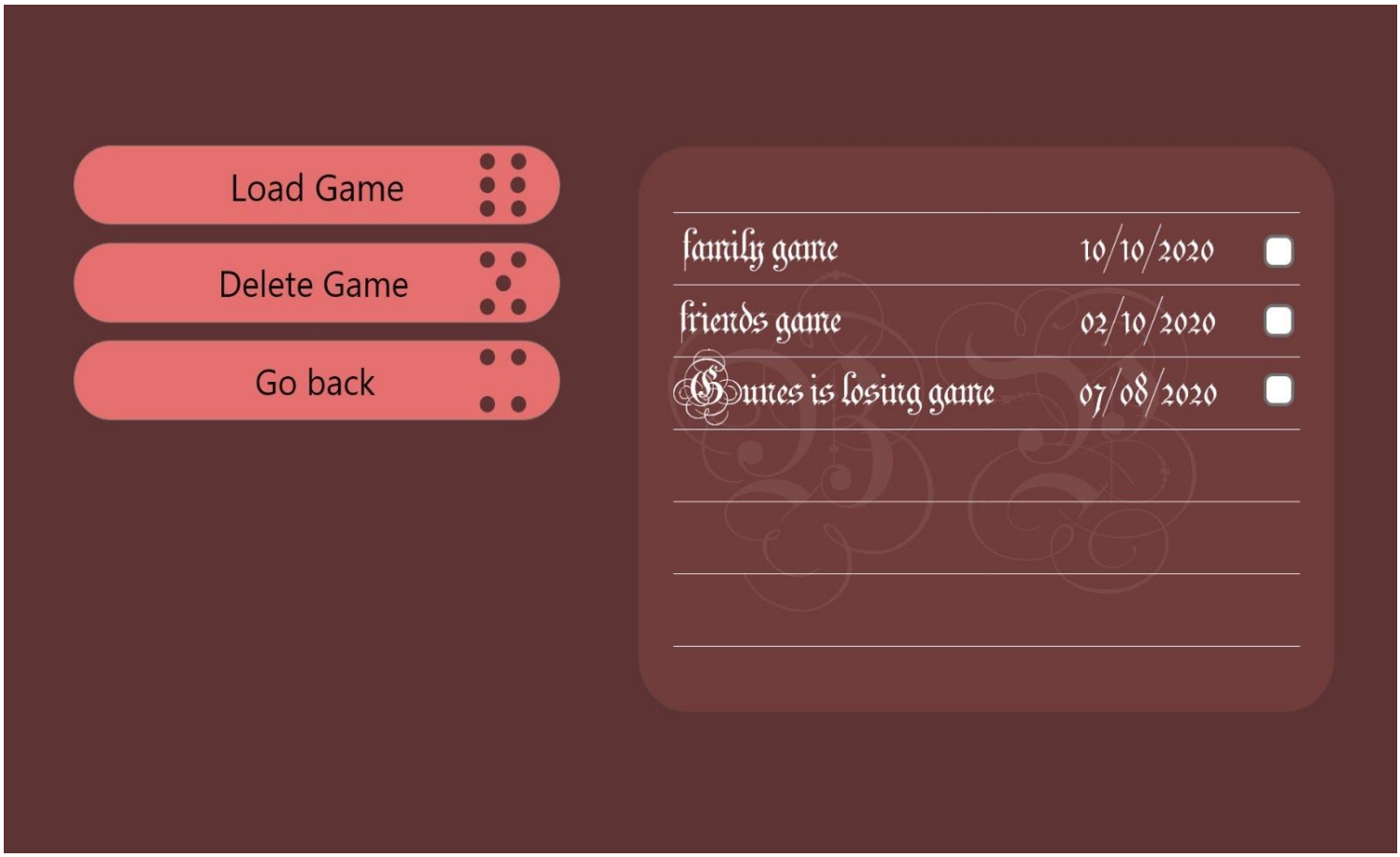

Image (6.3.2)

6.3.3. Settings

The third option from the main menu is the setting which lets the user alter different parts of the game specially sound. In this picture only mentioned the sound but other options will be included in this part which helps the user feel more customized for the game. Image (6.3.3)

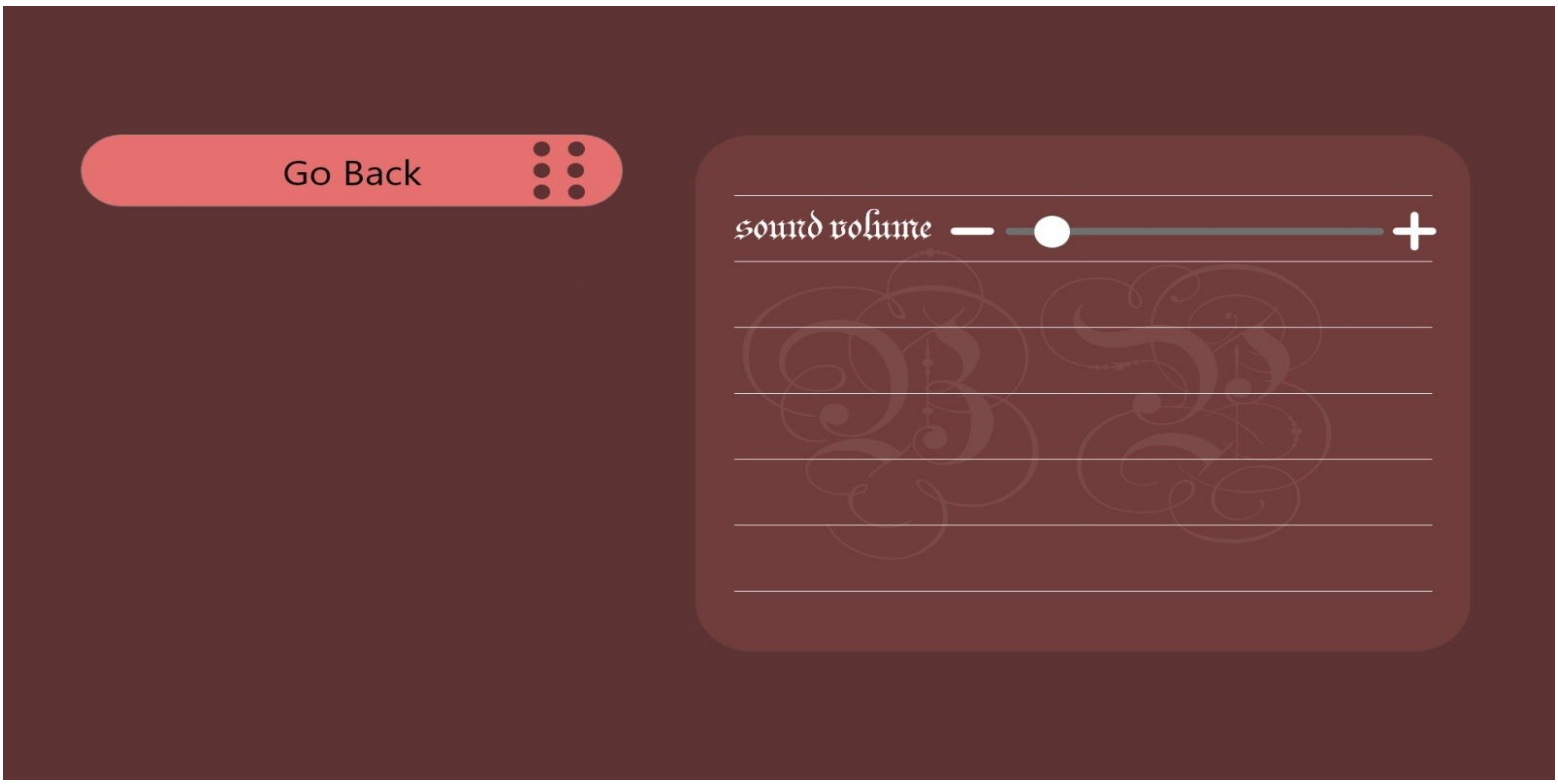

Image (6.3.3)

6.3.4. How to play

In this part we will provide the user with information about the rules of Bilopoly and how they should play the game. Image (6.3.4)

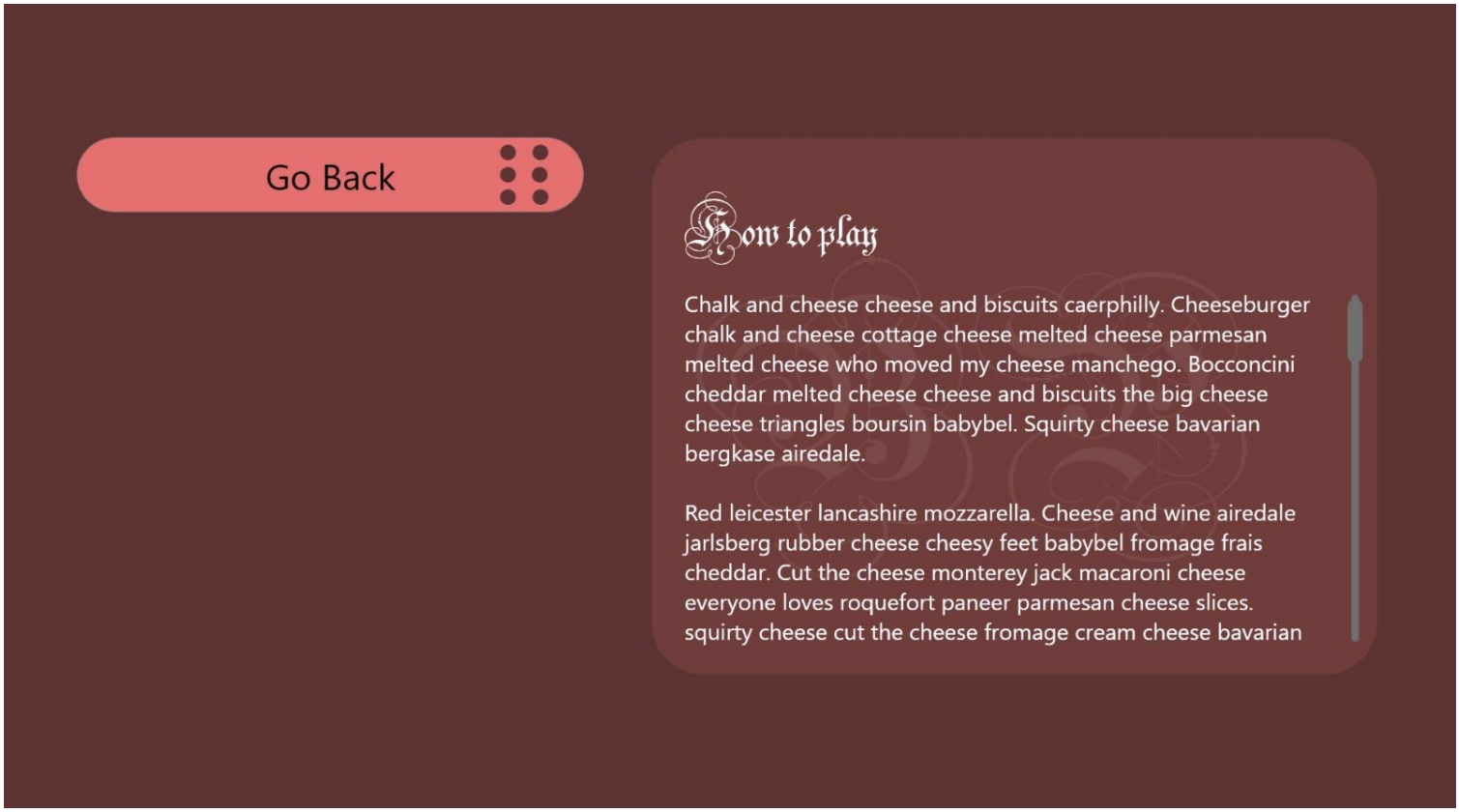

Image (6.3.4)

6.3.5. Credits

Our group name of developer and content developer information is available there which users will find out more about our group. Image (6.3.5)

Image (6.3.5)

6.3.6. New Game

After clicking on the New Game system will show this page which players are able to choose their nicknames, tokens and colors for representing them in the game. Image (6.3.6)

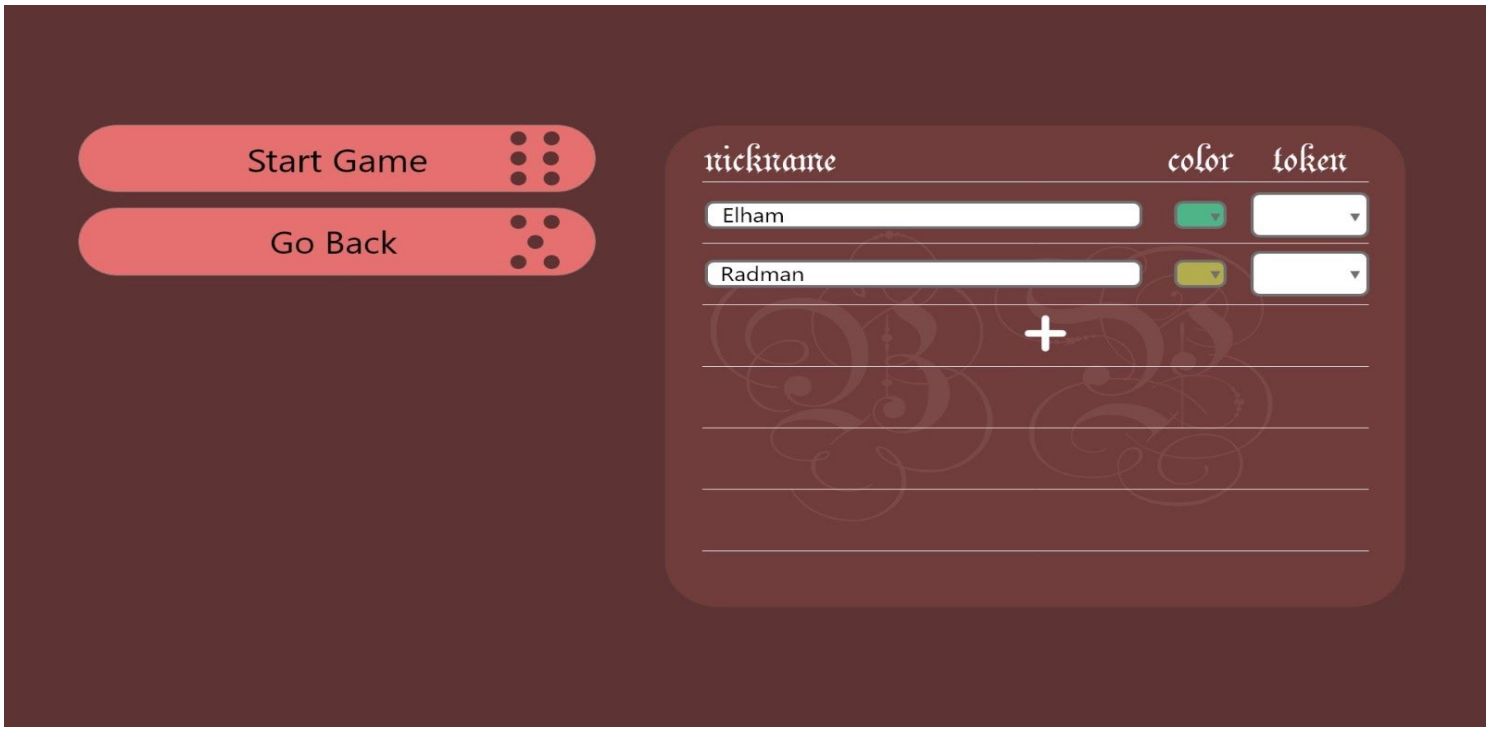

Image (6.3.6)

6.3.7. Pause Game

Players during the game are able to change some setting or save their game which is not mentioned in this picture because we are discussing its position. Moreover, for leaving the game players must use this page. Image (6.3.7)

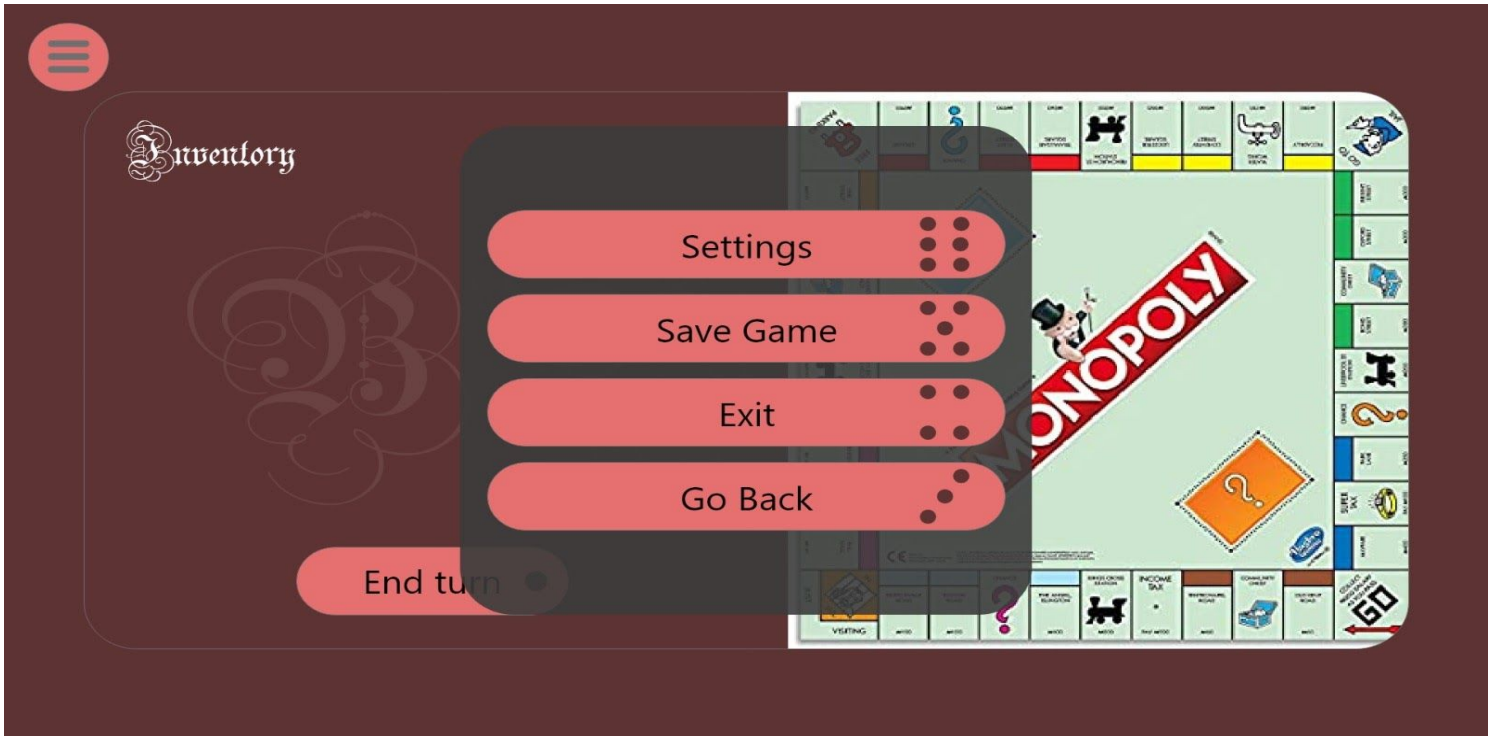

Image (6.3.7)

7. References

[1] “How Monopoly came to dominate board games”, https://www.scmp.com/lifestyle/arts-culture/article/1596064/how-monopoly-came-dominate-board-games [accessed: Oct 16, 2020].

[2] “Desktop Screen Resolution Stats Worldwide” https://gs.statcounter.com/screen-resolution-stats/desktop/worldwide [accessed: Oct 22, 2020].

[3] “Glossary of Supplementary Definitions” https://pubs.opengroup.org/architecture/togaf9-doc/arch/apdxa.html [accessed: Oct 22, 2020].

[4] “Software Testing and Quality Assurance Glossary” http://www.aptest.com/glossary.html [accessed: Oct 22, 2020]

[5] “Why Is It Important to Measure Maintainability and What Are the Best Ways to Do It?” https://ieeexplore.ieee.org/document/7965364 [accessed: Oct 22, 2020]

[6] “How Loading Time Affects Your Bottom Line” https://neilpatel.com/blog/loading-time [accessed: Oct 25, 2020]