Bilkent University

Department of Computer Engineering

CS319 Term Project

3A - Monopoly

Analysis Report

Ali Taha Dinçer, İrem Ecem Yelkanat, Kutsal Bekçi, Saidcan Alemdaroğlu, Sultan Simge Kürkçü

Instructor: Eray Tüzün
Teaching Assistant(s): Barış Ardıç, Elgun Jabrayilzade, Emre Sülün

Progress/Iteration 2 - Project Analysis Report
Dec 13, 2020

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Object Oriented Software Engineering course CS319.

Contents

Analysis Report

Group 3A: Monopoly

1 Introduction

Monopoly is a square shaped board game where each player has a unique pawn and in each turn according to dice results they move their pawns. The players try to have properties and with time, they are able to build structures on their owned properties. The main purpose of the Monopoly game is to avoid going bankrupt which means losing all the money and the properties. Having structures on the properties will help the players have an income so that they won't go bankrupt easily. The game can be played with at least 2 and at most 8 players.

There exists two same shaped card allocations on the board, which are located for chance cards and community chest cards. These cards are drawn by the players and increases the joy of the game. There are different buyable property squares located along the board. The game starts when the dice are rolled and finishes when every player except one gets bankrupt.

In this project, the purpose is to implement Monopoly into the digital single player and multiplayer platform. The name of the digital version of the game that we implemented is "Yolopoly". The players can play against the computer to enhance their skills and clash with their friends in multiplayer game mode.

2 Game Overview

Monopoly was first designed in 1935 with the aim of pursuing players on investing in properties and managing their money to be the last one standing while pushing the other players to go bankrupt. There are 22 different locations that players can be bought and there are options to build houses or hotels for each of the properties in the name of rules. The game is based on rounds. Each player has to move to a location where the sum of the dice rolled points. Each player gets money by doing one complete turn on the board, hence there is always a possibility to invest on each location on the board. If a player gets bankrupt, he/she will get out of the game until the game completes. This will continue until only one player is left on the game or all of the players decide to end the game.

2.1 Game Board

The board of the game is a directed cyclic graph. The board of the game consists of two sections. The first section is the inner section. This section contains the Chance Cards and Community Chest Cards that players can draw when the pawns of the players land on squares in the second section. The second section contains squares. There are different types of squares in the second section. There are 22 different locations along with 4 railroads, two utilities, three chance spaces, three community chest spaces, a luxury tax space, an income tax space and 4 corner squares.The design of these squares can be changed by theme selection before starting the game.

Figure 1: Game Board

2.1.1 Location Squares

These are the squares that players can build houses and hotels. Each of these squares consist of a group by their color. Only the last and first two squares in this type is in a group of two. All other squares are grouped by three. Each group has a color regarding their price.

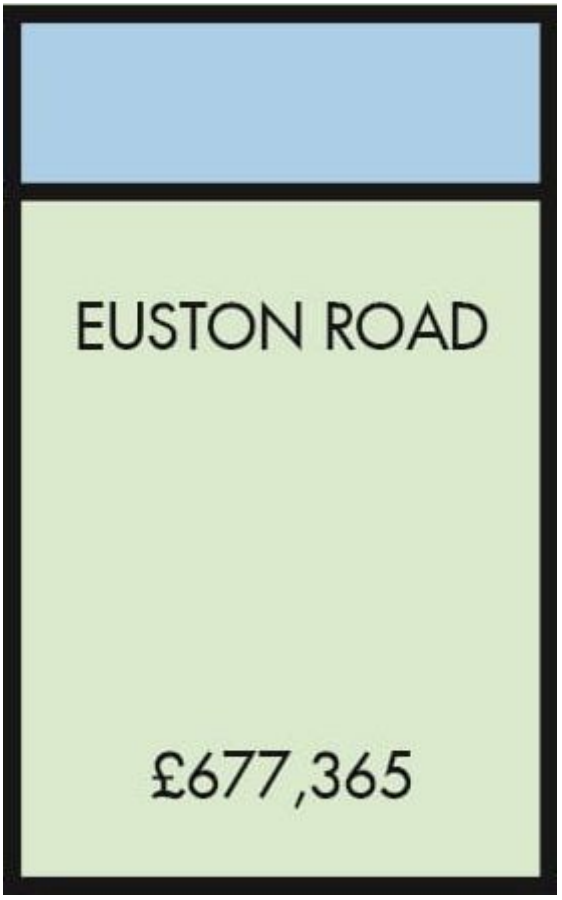


Figure 2: Location Square Example

2.1.2 Railroad Squares

There are only 4 railroad squares on the board. Each player can own these squares, and the rent price is multiplied by each of these locations owned.

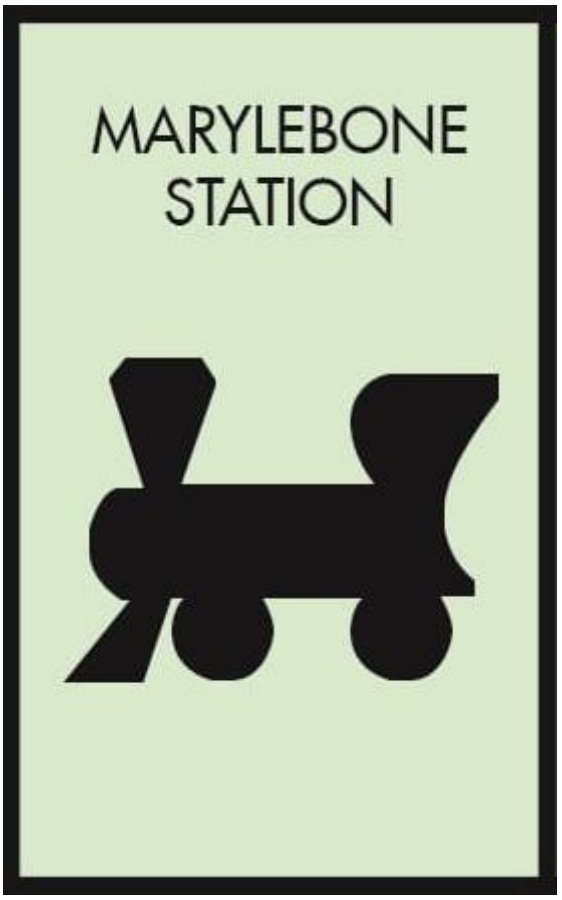


Figure 3: Railroad Square Example

2.1.3 Utility Squares

There are only two utility squares in the game. These squares can be bought by players and each of these squares enhances themselves like railroad squares.

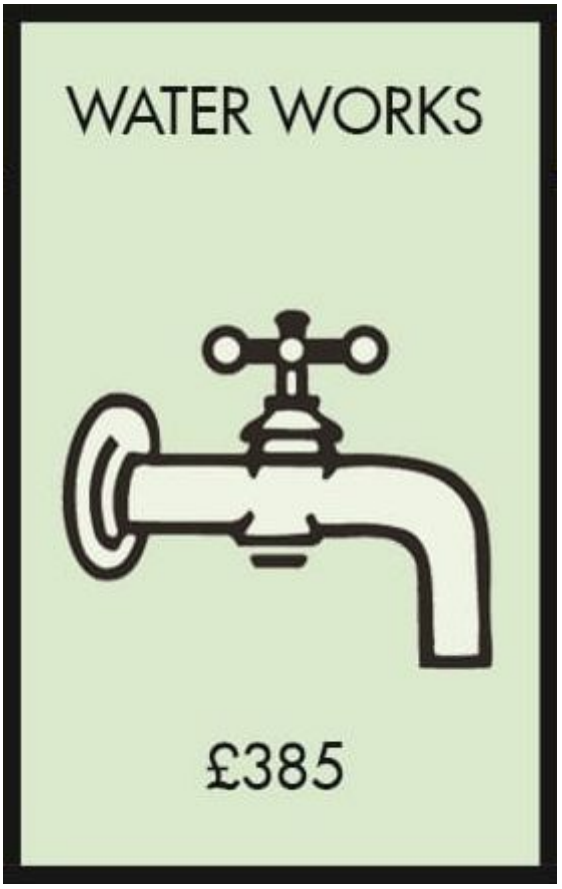


Figure 4: Utility Square Example

2.1.4 Chance and Community Chest Squares

There are three chance squares and three community chest squares in the game. Players have to draw a card according to the square type when their pawns land on these squares. These squares can not be owned by players, hence, players can not build structures or pay rent in these squares. In addition, these squares cannot be put on auction.

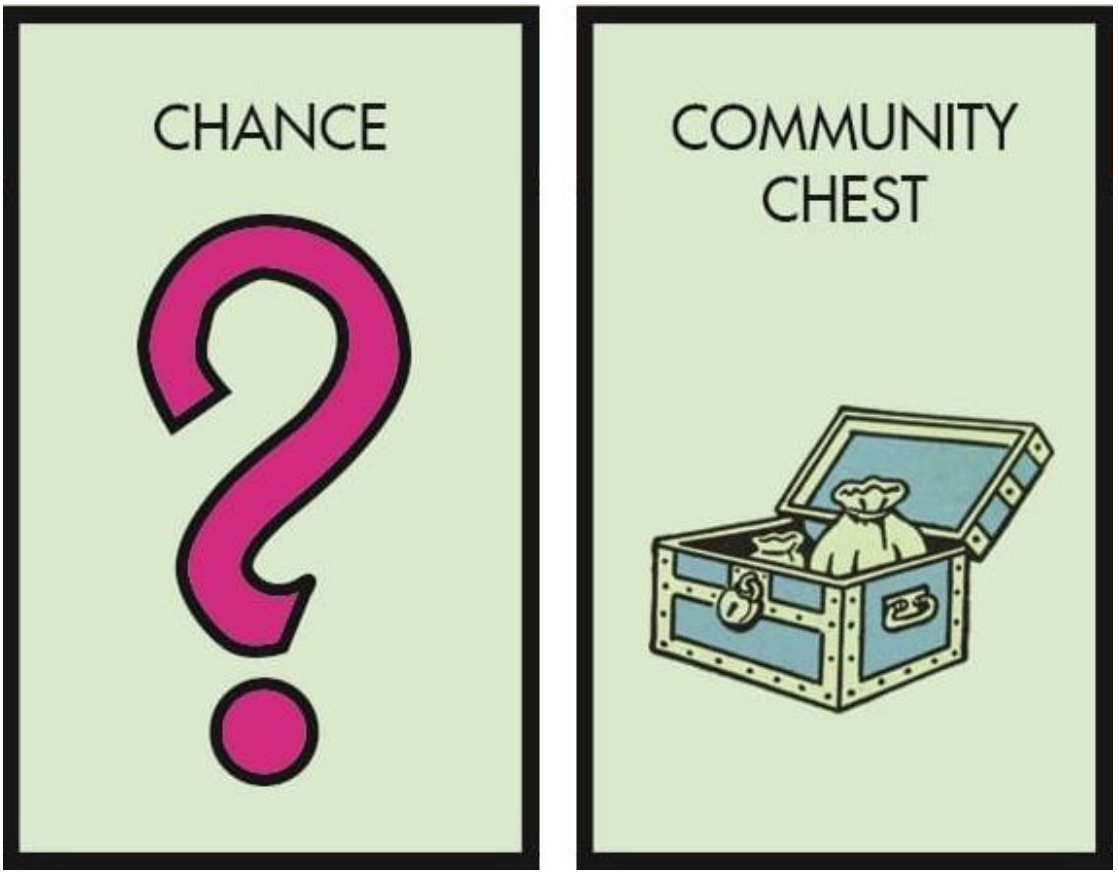


Figure 5: Chance and Community Chest Square Example

2.1.5 Luxury and Income Tax Squares

There are only one luxury tax square and income tax square in the board of the game. Players whose pawn land on these squares has to pay the amount written above in that square. Like chance and community chest squares, players can not own these squares. Hence, players can't build structures in these squares. In addition, these squares cannot be put on auction.

Figure 6: Luxury and Income Tax Squares Example

2.1.6 Corner Squares

As the name suggests and the board is a square itself, there are four different corner squares in the game. Each corner square can not be bought by the players but each of these squares has its own effects for the gameplay.

2.1.6.1 GO! Square

This is the initial point of the board for the players. At the beginning of the game, each player has to put their pawns in this square. During the game, players who pass this square receive some amount of money as income. Number of turns in the board is determined by this square. The money received by this square is determined by the game rules.

Figure 7: GO! Square Example

2.1.6.2 Just Visiting/Jail Square

This is the one of the most important squares in the game. This square has two sections. First section is to put the pawns of the players who are "just visiting" the jail. In other words, players who are forced to move by the dice will put their pawns in the "just visiting" section. Second section is the jail section. Players who have to go to jail by drawing the according chance or community chest card or players who moved to "Go To Jail" square have to put their pawns in this section. In addition, rolling dice and getting doubles thrice is a reason to go to jail square. These rules are explained in the further sections.

Figure 8: Just Visiting/Jail Square Example

2.1.6.3 Free Parking Square

This square is just a resting square for the players whose pawn lands on this square.

Figure 9: Free Parking Square

2.1.6.4 Go To Jail Square

This is one of the worst squares to land on the board during the game. Players whose pawn lands on this square have to go to jail square. Even if the player has a "Get out of jail" card, they have to put their pawns to "just visiting" section of the square and continue the game from there.

Figure 10: Go to Jail Square Example

2.2 Components

There are different types of components in the game. Each of the components are described in the following sections.

2.2.1 Cards

There are two different types of cards in the game. First type is the cards that contain actions. These types of cards are Chance Cards and Community Chest Cards. This type of card can only be obtained by landing a pawn to the corresponding squares in the board. The other type of the card is the

property cards. These cards indicate the attributes of the properties. These types of cards can be owned by buying from the bank when the pawn of the player lands on the according square or getting into an auction. More explanation about different types of cards are given in the sections below.

2.2.1.1 Cards With Actions

2.2.1.1.1 Chance Cards

Chance cards can only be obtained by the players when the pawn of the player lands on a chance card square. There are a total number of 16 chance cards in our game. Each card has its own specific actions. However, these actions can be grouped as stated below.

- Cards that move a player's pawn to another square.
- Cards that pay to players.
- Cards that force players to pay to other players or to the bank.
- Cards that forces player to go to jail
- Cards that help players to get out of jail. These types of cards are save-able.

Figure 11: Chance Card Example

2.2.1.1.2 Community Chest Cards

Community Chest Cards cards can only be obtained by the players when the pawn of the player lands on a community chest card square. There are a total number of 16 community chest cards in our game. Each card has its own specific actions. However, these actions can be grouped as stated below.

- Cards that move a player's pawn to another square.
- Cards that pay to players.
- Cards that force players to pay to other players or to the bank.
- Cards that forces player to go to jail
- Cards that help players to get out of jail. These types of cards are save-able.

Figure 12: Community Chest Card Example

2.2.1.2 Property Cards

Property cards can be bought by players and indicates a corresponding square in the board. Players who own properties can build houses, hotels and get rent payment type from other players. These cards can be put on auction by the players. Different types of Property Cards are explained in the further sections.

2.2.1.2.1 Location Cards

This type of property card indicates the location squares. Each location card has its own color group. Only the last two and first two locations are grouped by two and the other cards are grouped by three in a color. Players have to buy all the property cards of a color group to build houses and hotels. Each property obtained by the same color group and structures built on corresponding squares increases the rent of the property. In addition, mortgage of the property is available on both sides of this type of card.

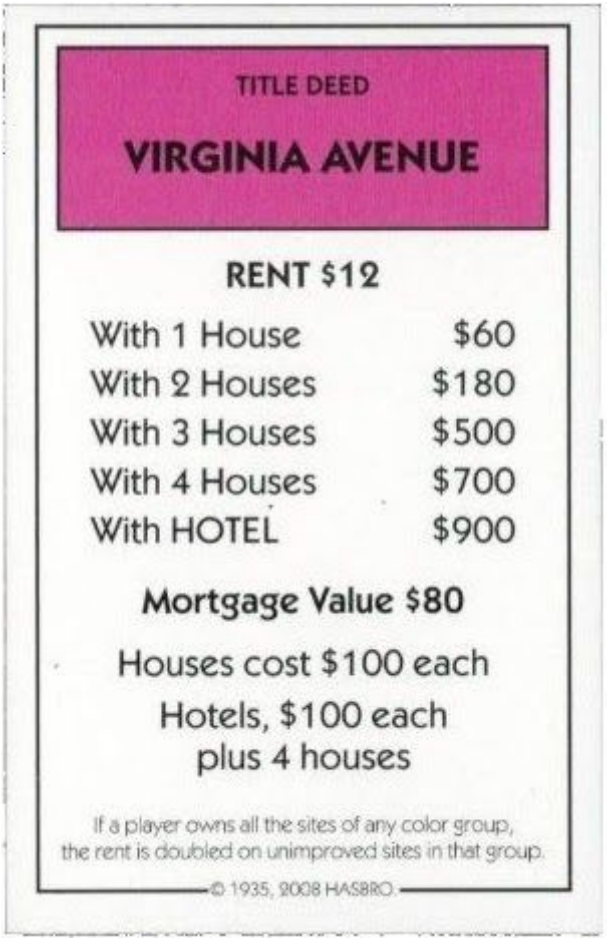


Figure 13: Location Card Example

2.2.1.2.2 Railroad Cards

There are only four Railroad Cards available for each corresponding square type on the board. Players can have all of the Railroad Cards in the game and each Railroad Card that player owns increases the rent that other players pay when their pawns landed on corresponding square type.

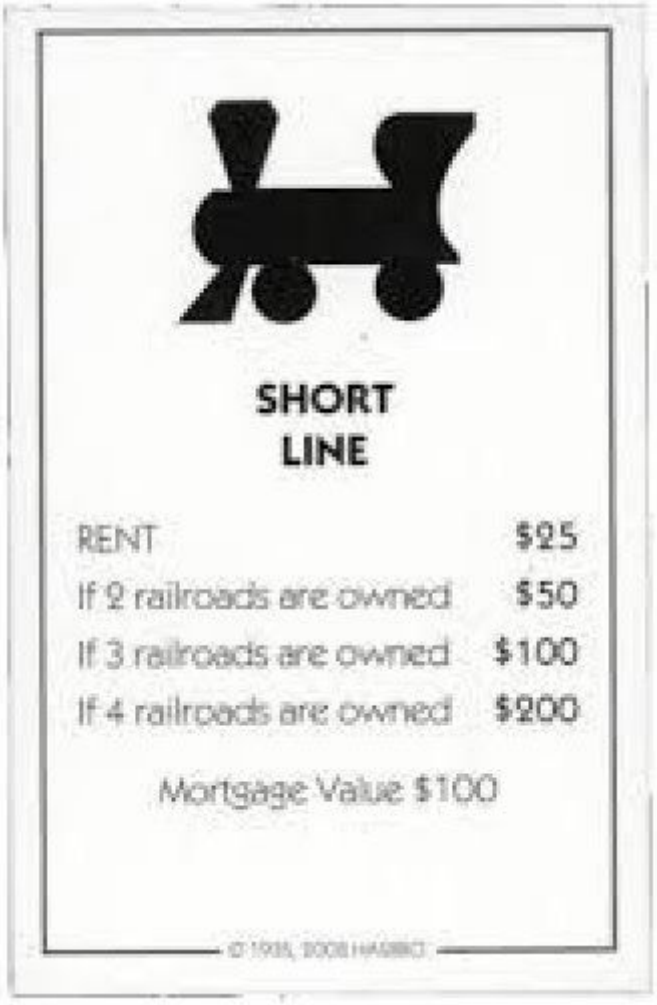


Figure 14: Railroad Card Example

2.2.1.2.3 Utility Cards

There are only two utility cards available in the game. Like railroad cards, rent of the Utility Cards increase when players own more than one of them.

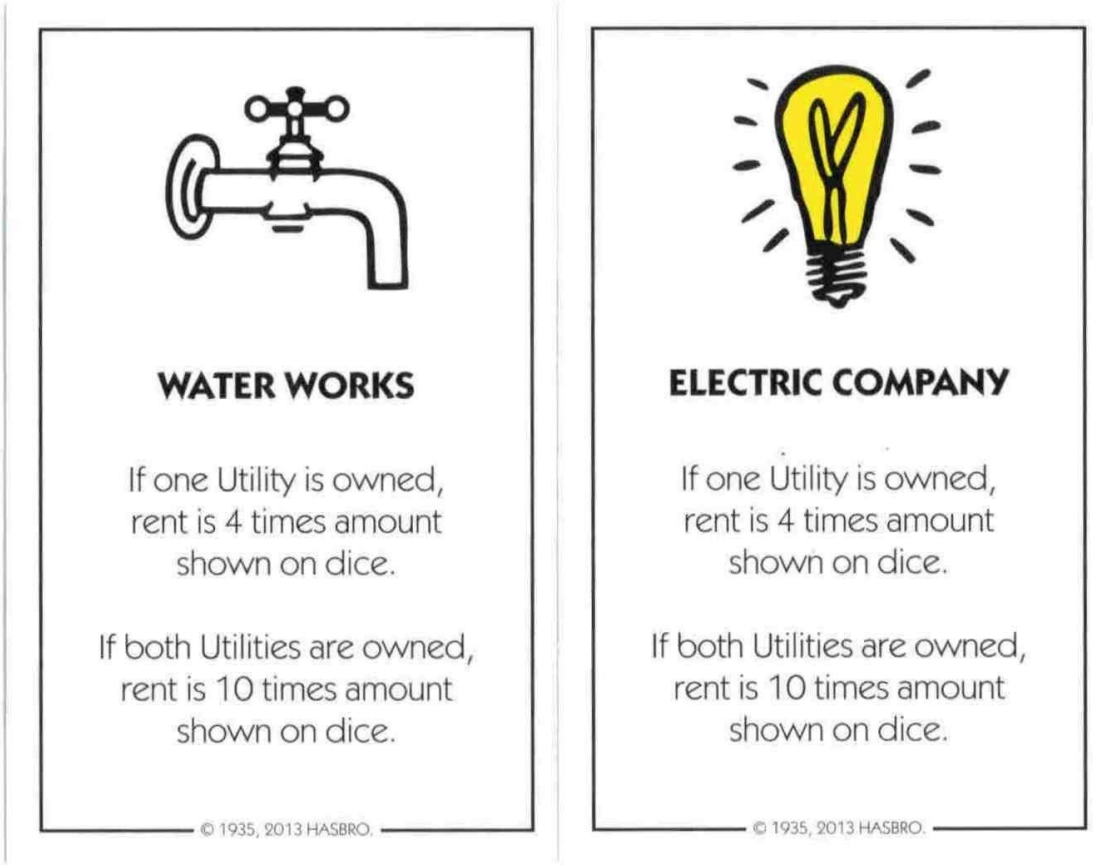


Figure 15: Utility Card Examples

2.2.2 Pawns

As there are a maximum number of 8 players allowed in the game, there are only 8 available pawns in the game. Pawns in the game can be changed by the theme selection before starting the game. Each player has to have a unique pawn to be able to start the game. The pawns for the vanilla theme of the game are given below.

Figure 16: Pawns

2.2.3 Bank

In the physical version of the game, Bank is also a player in the game and the gamer who works as the banker has to separate his/her hand from the components of the bank. In our digital version, Bank is an object and manages itself. In the game, the bank is considered to have unlimited money and Bank will pay money to the players and receive money from them. In addition, all properties that can be bought are first owned by the Bank. Hence, players will buy properties from the bank. Also, if a player gets bankrupt, the Bank will take the responsibility to create the auctions. Players can do mortgage actions with the Bank.

2.2.4 Money

In the physical version of the game, money is divided into different kinds and colors in terms of their value. In our version of the game, there is no division for the money needed. Each player in the game starts with an initial amount of money and money in the game is unlimited. Each player can receive or have to pay money because of the actions stated below.

- Player has passed the GO! Square. In this case, the player will receive money.
- Player has drawn a chance or community chest card regarding to do an action with money. In this case, players can either receive money or lose money.
- Player has bought a property. In that case, the player will lose money.
- Player has landed on one of the tax squares. In this case, the player has to lose money.
- Player has land on a property that another player owns. In this case, the player has to lose money.
- Player has decided to build structures in the properties that he/she owns. In this case the player will lose money.
- Player has decided to deconstruct structures in the properties that he/she owns. In this case the player will gain money.

2.2.5 Deck

In the physical version of the game, there is no real deck system that has been prescribed in the game. Basically, all of the properties and components that players owned are considered in the deck. With this motivation, our version of the game will contain an area for each player that players can see what they owned and do actions with them.

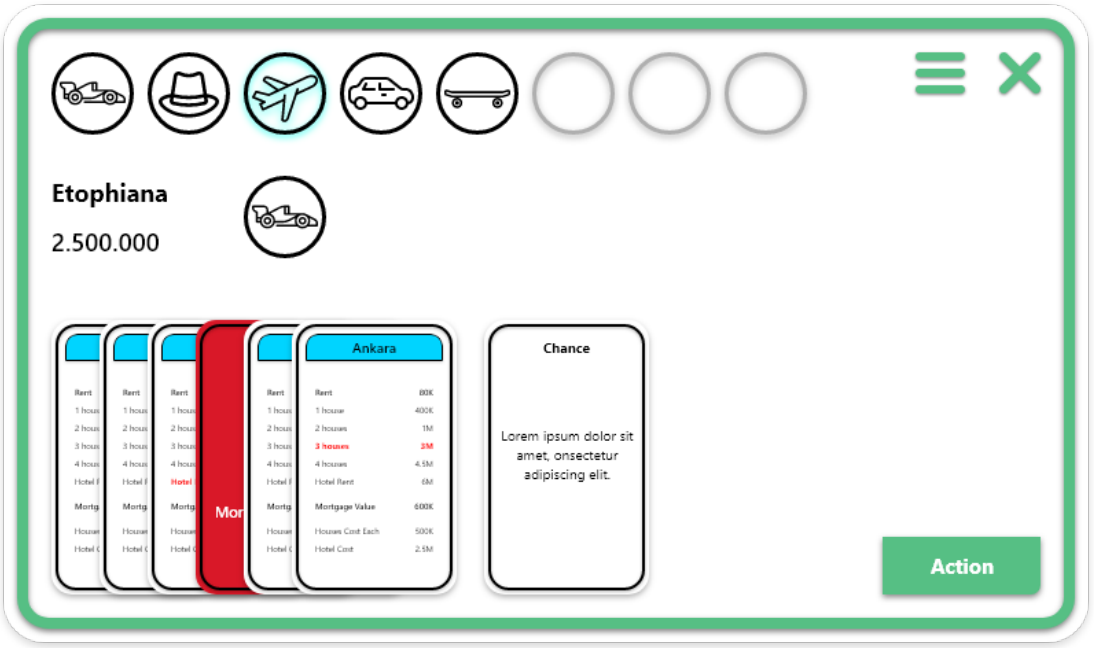


Figure 17: Deck UI Mockup

2.2.6 Buildings

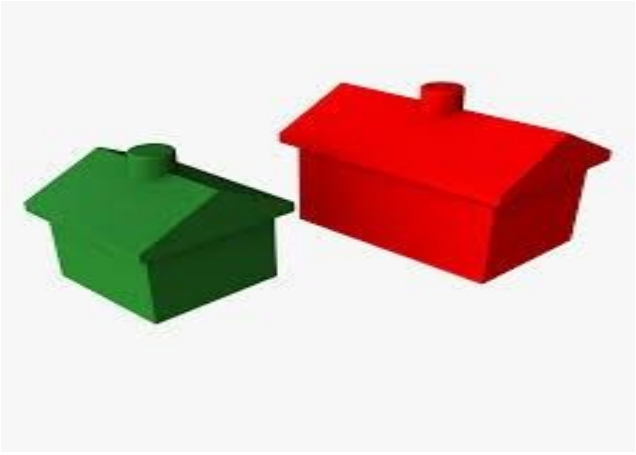


Figure 18: House and Hotel

2.2.6.1 House

Players are able to build houses on the properties that they have owned during their turns. House prices are available on the property cards of each square. Players can start to build houses when they own all the properties in the same color group. Each house increases the rent of the property and players are able to sell their houses with the half of the price they bought the house.

2.2.6.2 Hotel

Building a hotel is one of the hardest things to achieve in the game. After building four houses in each same colored property, then the player is able to exchange four houses in a property with an additional fee in the bank for a hotel. Like houses, hotels can be sold to the bank in exchange for 4 houses or the amount of money given for four houses and hotel fee.

2.3 Game Setup and Controls

There are several options to set up a game. Regarding the single player game option, the game setup requires that the player select his/her nickname, Bot count in the game, game mode and game theme. Regarding the multiplayer game option, there are two choices. The first option is to be a host. The host of a multiplayer game is able to change game mode, game theme, password of the lobby, total player count and bot players in the game. If players select the second option, then the player is able to join a game from the list shown. When a player selects a game from the list, then regarding player has to set a unique nickname. After entering the lobby, the player has to select a pawn and notify himself/herself as ready. Then the host of the lobby will finish the setup of the game. After the game has started, each player will roll the dice in order to select the first player.

The game can be played with a mouse connected to the computer or a touchpad. The player can do any action with the help of a mouse or touchpad by just clicking.

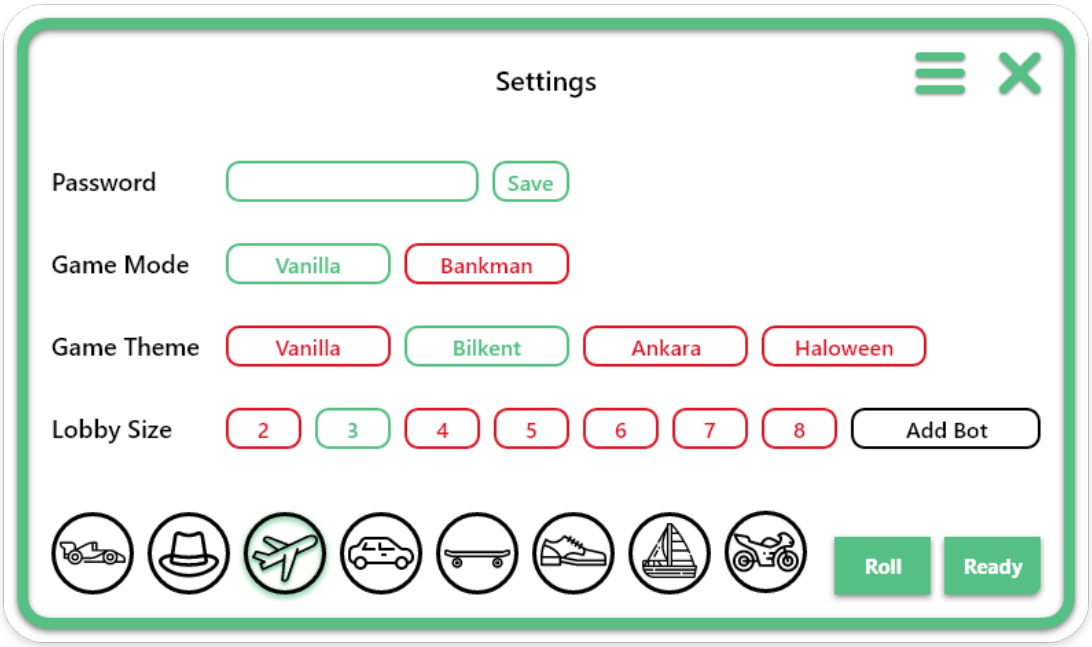


Figure 19: Game Setup Mockup

2.4 Gameplay

2.4.1 Play Turns

Each player has to roll the dice in order to play their turns. After the dice is rolled, the system will move the pawn of the player where the sum of dice indicates. According to the square that the pawn lands on, there are different actions that the player can or has to do. These actions are stated below.

Figure 20: Roll Dice and End Turn Button Mockup

- If the pawn of the player lands on any location squares that no other player has owned, then the player can buy that square. If the square has been owned by another player, then the according player has to pay the rent of the square.
- If the pawn of the player lands on a community chest square or chance square, then the player has to draw a card from corresponding card place.Then, player has to do the action stated on the card unless the

card drawn is a "Get Out of Jail" card. If a player draws a "Get Out of Jail" card, then the player could save the card in order to use it further.

- If the pawn of the player lands on a tax square, then the player has to pay the amount stated under the square to the bank.
- If a pawn of the player lands on the just visiting/jail square, the system will put the pawn automatically to just visiting section of that square.
- If the pawn of the player lands on go to jail square, then the system will automatically move the pawn of the player to the jail section of the just visiting/jail square. If a player has a "Get Out of Jail" card, then the system will use that card and move the pawn of the player to the just visiting section in the corresponding square.
- There are no special effects or required actions to do in the free parking square.
- If a player passes the GO! Square in any actions stated above, except the actions performed for moving the pawn of the player to the jail, then the player will get money from the bank as an income.
- If a player rolls the dice and gets doubles three times in a row, then the pawn of the player will be moved to the jail by the system.

All of the actions stated above will be done by the system automatically, but there are actions that players can do intentionally. These actions are stated below.

- Players can build structures in owned properties. The condition of building a house is owning all the properties in the same color group. If a player has four houses in a property, then the player can build a hotel in the corresponding property.
- Players can do mortgages on the properties that he/she owns.
- Players can create auctions. Creating an auction is left to the player without any condition. But making auctions become necessary when a player gets bankrupt.
- Players can examine the deck of the other players.
- Players can chat with others or read the game log.
- Player can end his/her turn.
- In Bankman mode of the game, players can exchange their money with other currencies for investment purposes.
- In Bankman mode of the game, a player can get a multiplier chance for each payment his/her does. The amount of these chances is unlimited but it is hard to get good chances.

- In the Bankman mode of the game, players can get loans from the bank. However, players need to pay these loans in a specific period of time in the future turns.

2.4.2 Buy Properties

Players can buy and sell properties during their turns. In fact, investing on properties is one of the crucial things for winning the game as it is the easiest way. Players can buy properties as long as they have enough money. In addition to that, players can sell their properties to the bank or mortgage their properties. Also, another way to buy a property is getting into auctions.

2.4.3 Pay Rent

Players have to pay rent of the property if the pawn of the player lands on a square that another player has owned. Base rent prices differ with the color of the properties. In addition, rent of a property changes if the player decides to build structures on the properties. Rent of the properties with hotels is five times of the rent price of the property with houses.

2.4.3.1 Pay Tax

Taxes are some type of rent and players have to pay tax if the pawn of the player is landed on squares. If players are unable to pay tax, then they have to find the money by selling some of the structures built on their properties, mortgage their properties, selling their properties to the bank, creating auctions with their properties or get a loan from the bank if the mode of the game has been set to Bankman.

2.4.4 Build Structures

Players can build structures on the location properties that they have owned. There are two structures that each player can build, houses and hotels. In order to build structures in the property squares, players have to buy all the same colored properties of the according property square. After buying all properties in the same color, players can start to build houses. If houses on a location square reaches 4, then the player is able to build a hotel in the according property square. In addition, players can sell the structures that they have built on location squares in order to get money when needed such

as paying rents or loans. Each structure built on a square increases the rent price of the property.

2.4.5 Go to Jail

There are several reasons for going to jail in the game. When a player is jailed in the game, the only action that player can not do becomes is moving the pawn to a different location. Hence, players may not be able to pay rent, draw a chance or a community chest card and buy properties. All the other actions are still available for the player such as building structures or getting rent from an owned property. In addition there are several options to getting out of jail, which are described in the next section. The reasons for going to jail are stated below.

- Pawn of the player landed on Go to Jail square.
- Player has drawn a "Go to Jail" chance card or community chest card.
- Player successfully rolled the dice and got doubles three times.

2.4.6 Get Out of Jail

As going to jail is a must, getting out of jail is a decision in the game. There are three possible actions that can be done in order to get out of jail.

- Players can roll the dice and get doubles. This option enables all the actions to the player.
- Players can use "Get Out of Jail" cards drawn and saved from chance cards of community chest cards. This option returns the corresponding card to the game and other players become able to draw that card again. To use this option, the player has already drawn the card before.
- Players can pay the fee of getting out of jail before rolling the dice.

2.4.7 Go Bankrupt

Going into bankruptcy is one of the reasons to lose the game. Players have to manage their money in order to not get into bankruptcy. Hence, good decision making is important in the game. A player gets bankrupt if the

player's money goes below zero. To avoid losing the game with bankruptcy, players have to sell some of their buildings, mortgage their properties, sell their properties or create auctions with their properties. In addition to that, passing GO! Square helps players to gain money and avoid bankruptcy. One of the easiest ways to get money to avoid bankruptcy is buying properties and investing in them. In addition to that, in the Bankman mode of the game, players can get a loan from the bank to avoid bankruptcy but remember that, this option is just procrastinating the incoming. If a player gets bankrupt, then the player loses the game.

2.4.8 Auction

Each player is able to create auctions during their turns. During the turn, players can create auctions with the properties that they owned. Like a basic auction, the property will be sold to the player with the highest bid given. In addition to that, when a player gets into bankruptcy, one of the ways to get away from it, is to create auctions with the properties owned. If a player is unable to get out of bankruptcy by auction, or the player chooses to lose the game, then the bank will create auctions with the properties of that player. In addition to that, if the pawn of the player lands on an unowned property square, and the player decides to not to buy that property, then the bank will put that property to the auction.

2.4.9 Winning the Game

There are only two possible ways to win the game. Most used way is to become the wealthiest player in the game when all of the players in the game decide to finish the game. The other way is to force other players to bankruptcy by managing the resources in the best way. Hence, when the other players get into bankruptcy, the player that survived will win the game.

2.5 Game Rules

- All players have to roll the dice to select the first player in the game.
- Each player will start with an amount of 1500 in-game money.
- Each player will gain 200 in-game money when they pass GO! Square.

- Each player has to roll the dice in the beginning of their turns in order to start the turn and each player has to move their pawn to the according square where the sum of the dice shows.
- Each player is able to buy properties where the pawn of the player lands on if the property has not been owned by any other player. In that case, the according player has to pay the rent of the property to the owner. In addition, if the player chooses not to buy the property, then the bank will create an auction and the property will be sold to the player with the highest bid given.
- Players do not have to pay rent if the pawn of the player lands on a mortgaged property.
- Players have to pay taxes when the pawn of the player lands in a tax square.
- Players have to draw a card respective to the pawn of the player that lands on a draw card square.
- If a pawn of the player lands on the just visiting/jail square, then the pawn of the players must be placed on just visiting section of that square. No charges will apply.
- If the pawn of the player lands on Go to Jail square, then player has to move his/her pawn to the jail section of the just visiting/jail square. If a player has a "Get Out of Jail" card, he/she can use it but start from just visiting a section of the square.
- Players can sell their properties to the bank with half of the price of that property or to the other players with the same amount of money.
- Players can only loan money from the bank in the Vanilla mode of the game by only mortgaging their properties. In Bankman mode, players can get a loan from the bank but must pay that loan in the further turns or charges will apply.
- If a player rolls a double, then the according player will be able to play one more turn. If the player rolls doubles third times, then the pawn of the player will be sent to the jail.
- Players can get out of jail if they roll a double but can not play the turn.
- If a player owns all the same colored properties in the game, then rent prices will be doubled.
- Players can build 4 houses and 1 hotel in each property they have owned.
- In the physical version of the game, players can build structures in their properties in any time they want. But we have prohibited this

feature to make better gameplay for everyone. Hence, players can build structures only in their turns.

- Players have to build 4 houses in all the properties in the according color group in order to build hotels.
- Players have to return their houses in the property if the player decides on building a hotel.
- Players can get money from selling their houses, hotels and downgrading their hotels to houses.
- Players can de-mortgage their properties by paying the mortgage price + %10 interest to the bank.
- The rent of the railroad and utility properties will increase with each railroad or utility property bought.

3 Requirements

3.1 Functional Requirements

3.1.1 Create Single Player Game

- The players must be able to create a new single player game.
- The players must be able to choose the game type.
- The players must be able to enter his/her nickname.
- The players must be able to choose his/her pawn.
- The players must be able to determine the number of players between 2 and 8.
- The players must be able to choose the mode of the game.
- The players must be able to choose the theme of the game.
- The players must be able exit from the Create Single Player Game section and return to the Main Menu.
- The game engine must check the validity and the uniqueness of the player’s nickname.
- The game engine must generate the game regarding game theme and game mode.
- The game engine must assign unique pawns and nicknames to the bot players.
- The game engine must immediately start the game after the player enters his/her nickname and chooses a pawn.
- The game engine must check the validity of the player's nickname.
- The game engine must route the player to the Main Menu if requested.

- The game engine must create the bot players according to the number of players determined.

3.1.2 Create Multiplayer Game

- The players must be able to create a new game.
- The players must be able to choose the game type.
- The players must be able to enter his/her nickname.
- The players must be able to choose his/her pawn.
- The players must be able to determine the number of players between 2 and 8.
- The players must be able to choose the mode of the game.
- The players must be able to choose the theme of the game.
- The players must be able exit from the Create Multiplayer Game section and return to the Main Menu.
- The players must be able to generate a password for their game to share with friends.
- The game engine must check the validity of the player's nickname and pawn.
- The game engine must generate the game regarding game theme and game mode.
- The game engine must route the player to the Main Menu if requested.

3.1.3 Join Game

- The players must be able to join a created multiplayer game.
- The players must be able to join a created multiplayer game if he/she has the same network connection with the creator of the game.
- The players must enter a password for the game if the game requires a password to join.
- The players must be able to enter his/her nickname.
- The players must be able to choose his/her pawn.
- The players must be able to roll dice to determine the turn order in the beginning of the game.
- The players must be able exit from the Join Game section and return to the Main Menu.

- The game engine must start the game after the required number of players have joined and all players are ready to play the game, regarding the multiplayer game.
- The game engine must route the player to the Main Menu if requested.
- The game engine must check if the player has entered a valid key for the game.
- The game engine must check that each player chooses different pawns.
- The game engine must check that each player chooses different nicknames.
- The game engine must warn the players if they don't choose unique pawns and nicknames.
- The game engine must connect the players of a multiplayer game by local network.
- The game engine must start the game after all players enter their nickname, choose a pawn and hit the ready button.

3.1.4 Play Game

- The players must be able to play the game.
- The players must be able to play the game as long as one player has not gone bankrupt in a typed multiplayer game.
- The players must be able to play the game as long as the game is not finished by the current player or one player has not gone bankrupt in a typed single player game.
- The players must be able to win the game by being the one that has the most money with the values of his/her properties and buildings at the end of the game.
- The players must be able to play their turn.
- The players must be able to roll the dice and perform actions in their turns.
- The players must be able to chat with each other during the game.
- The players must be able to see the last performed actions in the log part.
- The players must be able to change system settings regarding music and effect during the game.
- The players must be able to save the game and exit from the games that are typed single player and return to the Main Menu.

- The players must be able to exit from the games that are typed multiplayer and return to the Main Menu.
- The game engine must give every player a turn one by one.
- The game engine must start the game after all players have chosen their pawns and unique nicknames have been provided.
- The game engine must save the state of the game if requested.
- The game engine must route the player to the Main Menu if requested.
- The game engine must replace the players that exited from the game with bot players in typed multiplayer games.
- The game engine must change the state of the game and players after each turn.
- The game engine must store the last performed actions and display them to the players accordingly in the game log.
- The game engine must store the chat messages and display them to the players accordingly in the chat area.
- The game engine must change system settings regarding music and effect during the game if requested.
- The game engine must store the players' information about the remaining amount of money, cards, buildings etc.
- The game engine must end the game if the game is finished in both game types or there are no real players left on a typed multiplayer game.

3.1.5 Save Game

- The players must be able to save the game typed single player that they are currently playing to continue from the last state later.
- The players must be able save up to three games.
- The players must be able to play the saved game any time.
- The players must be able to delete saved games.
- The game engine must save the current state of the game if requested.
- The game engine must not allow the players to save more than three games at a time.

3.1.6 Load Game

- The players must be able to load a single player typed game not finished previously.
- The players must be able to see and choose from the not finished games.
- The players must be able to continue to the game reloaded with the last state of the game.
- The players must be able exit from the Load Game section and return to the Main Menu.
- The game engine must reload the last state of the loaded game requested.
- The game engine must display the unfinished games which were saved and single player typed.
- The game engine must route the player to the Main Menu if requested.

3.1.7 Change Settings

- The players must be able to change the settings if they want, or keep going on with the default settings.
- The players must be able to adjust the volume of the game and the volume of the music.
- The players must be able to exit from the Change System Settings section and return to the Main Menu.
- The game engine must change the settings regarding sounds if requested.
- The game engine must route the player to the Main Menu if requested.

3.1.8 Get Help

- The players must be able to get the information regarding the game.
- The players must be able to see the list of the game rules.
- The players must be able to have a tutorial with images about how to play the game.
- The players must be able to exit from the Get Help section and return to the Main Menu.
- The game engine must provide players with the game rules and tutorial of the game.

- The game engine must route the player to the Main Menu if requested.

3.1.9 See Credits

- The players must be able to get more information regarding the developers of the game.
- The players must be able to exit from See Credits section and return to the Main Menu.
- The game engine must display the information about the developers of the game to the players.
- The game engine must route the player to the Main Menu if requested.

3.1.10 Send Feedback

- The players must be able to send their feedback to the developers of the game.
- The players must be able to report the bugs that they encountered to the developers of the game.
- The players must be able to exit from the Send Feedback section and return to the Main Menu.
- The game engine must deliver the players' feedback and bug reports to the developers of the game.
- The game engine must route the player to the Main Menu if requested.

3.1.11 Play Turn

- The players must be able roll the dice and perform actions during their turns.
- The players must be able to end their turns after they have performed their actions if available.
- The players must be able to end their turns without performing any action.
- The game engine must change the turn of the player after the current player ends his/her turn.
- The game engine must warn the players to finish their turns.

- The game engine must update the game state after an action is performed.

3.1.12 Pass by GO! Square

- The players must be able to pass by the GO! square many times throughout the game.
- The players must gain money each time they pass through the GO! square.
- The game engine must correctly calculate if a player has passed by GO! square in order to pay the player.
- The game engine must update the game state after a player has passed through GO! square.

3.1.13 Buy Property

- The players must be able to buy properties throughout the game.
- The players must be able to buy properties if they satisfy the conditions to buy a property.
- The game engine must check if the player is able to buy the property considering the conditions to buy a property.
- The game engine must warn the player if he/she is not able to buy the property.
- The game engine must change the state of game and player after player bought a property.

3.1.14 Build Structures

- The players must be able to build structures, houses and hotels, on the properties that they own.
- The game engine must check if the conditions are held for the player to build a structure.
- The game engine must warn the player if he/she is not able to build structures.
- The game engine must change the state of game and player after player bought a property.

3.1.15 Go to Jail

- The players must go to jail if the necessary conditions are encountered.
- The game engine must check if the conditions are held for the player to go to jail.
- The game engine must warn the player if he/she is not able to perform actions in the jail.
- The game engine must update the game state after a player has gone to jail.

3.1.16 Get out of Jail

- The players must be able to get out of jail, if the conditions are satisfied to get out of the jail.
- The game engine must check if the conditions are held for the player to get out of jail.
- The game engine must not allow the player to get out of jail and play turn if the conditions required for going out of jail are not satisfied.
- The game engine must update the game state after a player has got out of jail.

3.1.17 Draw Card

- The players must be able to draw cards that are chance or community chest from the board.
- The players must be able to draw cards that are chance or community chest from the board if the conditions for drawing a card is satisfied.
- The players must be able to use the "Out of Jail" card for the further turns if they have drawn it.
- The game engine must shuffle the cards when the card is returned to the game board.
- The game engine must keep the "Out of Jail" card in the deck of the player until it is used.
- The game engine must play according to the contents of the drawn cards i.e., increase/decrease the amount of the money that players have.

- The game engine must check if players have enough resources to perform the actions in the cards.
- The game engine must update the state of the game after a chance card or community chest card has drawn.

3.1.18 Pay Rent

- The players must pay rent if the necessary conditions are encountered.
- The game engine must check if the conditions are held for the player to pay rent.
- The game engine must check if the player has enough money to pay rent.
- The game engine must update the state of the game after a player has paid rent.
- The game engine must warn the player if he/she doesn't have enough money to pay rent.

3.1.19 Pay Tax

- The players must pay tax if the necessary conditions are encountered.
- The game engine must check if the conditions are held for the player to pay tax.
- The game engine must check if the player has enough money to pay tax.
- The game engine must update the state of the game after a player has paid tax.
- The game engine must warn the player if he/she doesn't have enough money to pay tax.

3.1.20 Go Bankrupt

- The players must go bankrupt if they don't have enough money to perform an action, i.e., paying rent, paying tax, playing according to the chance and community chest cards.
- The game engine must check if the conditions are held for the player to go bankrupt.

- The game engine must warn the player if he/she has gone bankrupt.
- The game engine must update the game state after a player has gone bankrupt.

3.1.21 Auction

- The players must be able to put property cards in auction.
- The players must be able to participate in auctions except for the player that his/her property is sold.
- The players must be able to offer money to a specific property in the auction.
- The players must be able to own the properties if they offer the most amount of money before the end of the auction.
- The players must offer money for properties in the limit of the amount of the money they have.
- The game engine must start an auction if necessary requirements are satisfied.
- The game engine must not allow the player that his/her property is sold by the auction to participate in the auction.
- The game engine must warn the player that his/her property is sold by the auction if he/she tries to participate in the auction.
- The game engine must control the money that players offer considering the players' amount of money.
- The game engine must warn the players if they bid more money than the money that they have.
- The game engine must keep track of the bid values of each player during auction.
- The game engine must determine the winner of the auction considering the money offered by each player.
- The game engine must update the game state after a property card has been auctioned and bought.

3.1.22 Mortgage Properties

- The players must be able to mortgage their properties.
- The players must be able to mortgage their unimproved properties.
- The players must sell houses/hotels on their properties in the same color group if the player wants to mortgage the property.

- The players must be able to lift the mortgage on their properties.
- The players must be able to lift the mortgage on their properties provided that they have paid to the bank the required money.
- The game engine must update the state of the game after the player mortgaged his/her property.
- The game engine must not allow the player to collect money for rent if the property is mortgaged.

3.1.23 End Turn

- The players must be able to end their turns after they (don't) perform actions.
- The players must be able to end their turns any time during their turns.
- The game engine must update the state of the game after a player ended his/her turn.
- The game engine must pass the turn to the next player after a player ends his/her turn.

3.1.24 Bankman Mode Additions

Bankman is the additional mode that is added to the monopoly game and the mode provides extra features to the game as explained in the following sections.

3.1.24.1 Multiply the Loss or Income

- The players must be able to decide risking the change in the proportion of income or loss in their next turn between 0.5 and 2.0.
- The players must be able to update their loss/income in the turn they choose to set a proportion for their income/loss.
- The game engine must generate a random number between 0.5 and 2.0 if the players choose to set a proportion for their income or loss.
- The game engine must store the generated number to determine the updated value of loss/income after the turn has finished for the players choosing to set proportions.

- The game engine must correctly calculate the income/loss of the players that choose to set a proportion at the end of the corresponding turn.
- The game engine must update the state of the game after the players choose to set a proportion to their income or loss.

3.1.24.2 Currency System

- The players must be able to invest their money in other currencies in the game.
- The players must be able to exchange their money between different currencies within their turns.
- The game engine must store the money of the players belonging to different currencies.
- The game engine must transfer the money of the players between currencies if requested.
- The game engine must update the values of currencies after each turn.

3.1.24.3 Additional Cards

- The players must be able to draw additional cards that belong to the type chance or community chest from the board.
- The players must be able to draw additional chance cards and community chest cards from the board if the required conditions are met.
- The game engine must make the game have the bankman version of the chance and community chest cards.
- The game engine must play according to the contents of the drawn cards i.e., increase/decrease the amount of the money that players have.
- The game engine must check if players have enough resources to perform the actions in the cards.
- The game engine must update the state of the game after a chance card or community chest card has drawn.

3.1.24.4 Get Loan

- The players must be able to get a loan from the bank.
- The players must be able to get a loan from the bank if the necessary conditions are met.
- The players must return the money that they loaned to the bank in a specific number of turns.
- The players must be discarded from the game if they don't return the money back in a specific amount of turns.
- The game engine must keep track of the turn count between the loan and return of the money.
- The game engine must discard the player from the game if the loan is not paid back.
- The game engine must not allow the player to play the game if the loan is not paid back.
- The game engine must update the state of the game after a player got a loan from the bank.

3.2 Non-functional Requirements

3.2.1 Usability

- The players must be able to play the game only with a mouse connected to their computers/laptops or touchpad on their laptops.
- The players must be able create a single player game within 30 seconds including choosing the pawn, nickname, number of players, game mode, game theme and hitting the create game button.
- The players must be able create a multiplayer game within 1 minute including choosing the pawn, nickname, number of players, game mode, game theme and hitting the create game.
- The players must be able to join a multiplayer game within 45 seconds including entering the game password (if specified), choosing a unique pawn, unique nickname and hitting the ready button.
- The players must be able to access all the screens of the game at most 5 clicks.

3.2.2 Reliability

- The game must have an exception handling mechanism to prevent errors.

- The game must display error messages to the players according to the error encountered.
- The game must ensure that no data is lost and save the data in the case of a crush.

3.2.3 Performance

For a fast user experience, the following constraints are required:

- The actions that were performed by the players during the game must update the state of the game within 0.2 seconds.*
- The game must be launched in at most 1 second.*
- The game should be displayed and played at least 30fps.*

*Values were estimated for a computer with a processor having at least 1.8GHz frequency, 2GB of RAM, 1GB of available space and 1GB of video memory.

3.2.4 Security

- The game must store only the game data in the local file system. The system of the game must not hold any private data of the user other than the nickname of the user.

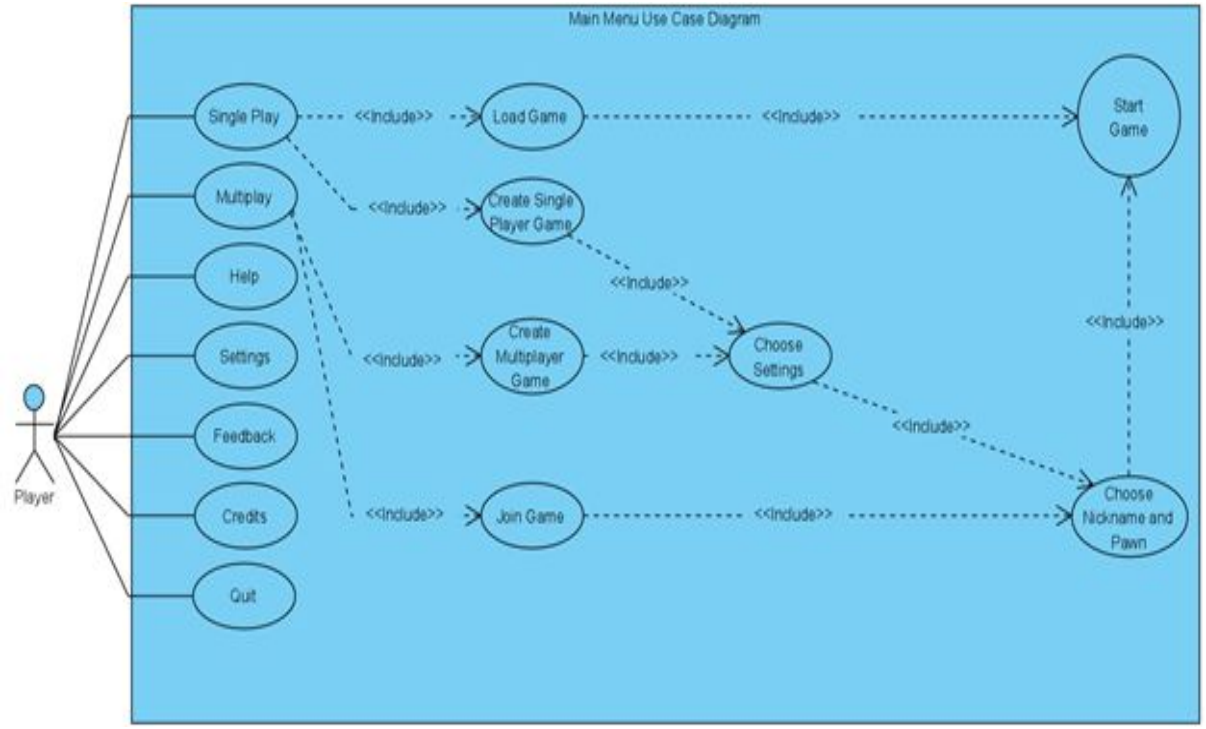
3.2.5 Portability

- The game must be able to run on any environment or platform. It must be executable within all the desktop operating systems.

4 System Models

4.1 Use Case Models

4.1.1 Main Menu Use Case Model



Model 1: Use Case Model of the Main Menu

4.1.1.1 Use Case: Single Play

Use Case: Single Player

Participating Actor: Player

Stakeholders and Interests: The player may want to start playing a single game.

Entry Condition: The player invokes the game by selecting "Single Play" in order to start playing the game alone.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player continues to choose to either load the game or create a new single player game right after clicking on the "Single Play" button.

4.1.1.2 Use Case: Load Game

Use Case: Load Game

Participating Actor: Player

Stakeholders and Interests: The player may want to continue the previously saved game.

Entry Condition: The player continues to the game with the "Load Game" button.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player picks one of the games she/he played before but not finished.

2. The player clicks on "Load Game" and picks one game from the list.

3. The system loads the chosen previously saved game.

4. The player will resume the chosen game.

4.1.1.3 Use Case: Create Single Player Game

Use Case: Create Single Player Game

Participating Actor: Player

Stakeholders and Interests: The player may want to start a new single player game.

Entry Condition: The player initiates a new game by clicking the "New Game" button.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player clicks on the "New Game" button.

2. The player will be forward to the settings section.

4.1.1.4 Use Case: Multiplay

Use Case: Multiplayer

Participating Actor: Player

Stakeholders and Interests: The player may want to start playing the game with others.

Entry Condition: The player creates the game by selecting the "Multiplayer" option.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player chooses the "Multiplayer" option in order to play with other players.

4.1.1.5 Use Case: Join Game

Use Case: Join Game

Participating Actor: Player

Stakeholders and Interests: The player wants to join an already created multiplayer game.

Entry Condition: The player clicks on the "Join Game" button to go to the game list.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player joins a game by selecting the game from the list.

2. If the selected game is private, the player won't be able to join this game without knowing the password.

4.1.1.6 Use Case: Create Multiplayer Game

Use Case: Create Game

Participating Actor: Player

Stakeholders and Interests: If the player doesn't want to join an existing game, they should create their own game with the "Create Game" option.

Entry Condition: The player clicks on the "Create Game" button to generate their game.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. "Create Game" button is clicked.

2. The player is forward to the "Choose Settings" screen.

4.1.1.7 Use Case: Choose Settings

Use Case: Choose Settings

Participating Actor: Player

Stakeholders and Interests: The player may want to determine game settings in "Choose Settings".

Entry Condition: The player creates a new game.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player determines the number of the players.

2. The player chooses the theme of the game.

3. The player chooses the mode of the game.

4. The player may determine a password if the game is a multiplayer game.

4.1.1.8 Use Case: Choose Nickname and Pawn

Use Case: Choose Nickname and Pawn

Participating Actor: Player

Stakeholders and Interests: The player may want to determine his/her nickname and pawn after the player clicks on "Join Game".

Entry Condition: The player is either in "Choose Settings" or "Join Game" sections to set nickname and pawn.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player enters his/her nickname.
2. The player chooses his/her pawn.
3. The player clicks the "Ready" button.

4.1.1.9 Use Case: Start Game

Use Case: Start Game

Participating Actor: Player

Stakeholders and Interests: The player may want to start playing the game.

Entry Condition: All the players in the lobby have clicked on the "Ready" button.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The game is generated according to settings.
2. The players start to play the game.

4.1.1.10 Use Case: Help

Use Case: Help

Participating Actor: Player

Stakeholders and Interests: The player may want to get instructions and learn the rules of the game.

Entry Condition: The player clicks on the "Help" button to get help with playing the game.

Exit Condition: The player returns to "Main Menu".

Success Scenario Event Flow:

1. The player opens the "Help" section.

2. The player reads the instructions and details about the game.

3. The player can return "Main Menu".

4.1.1.11 Use Case: Settings

Use Case: Settings

Participating Actor: Player

Stakeholders and Interests: The player may want to change the settings of the game.

Entry Condition: The player clicks on the "Settings" button to change default settings of the game.

Exit Condition: The player returns to "Main Menu" after changing settings.

Success Scenario Event Flow:

1. The player opens the settings section.

2. The player may see and change the default settings of "Music" and "Effect" audios. Making a change is not obligatory.

3. The settings are changed by the system.

4. The player may return back to the main menu.

4.1.1.12 Use Case: Feedback

Use Case: Feedback

Participating Actor: Player

Stakeholders and Interests: The player may want to report a problem or send a suggestion about the game to the developers of the game.

Entry Condition: The player clicks on the "Feedback" button to write down a problem or suggestion about the game.

Exit Condition: The player returns to "Main Menu" after writing down the subject.

Success Scenario Event Flow:

1. The player clicks on "Feedback".

2. The player writes down the issue to the empty box.

3. The issue written by the player is sent to the related address and success message appears.

4. The player may return back to the main menu.

4.1.1.13 Use Case: Credits

Use Case: Credits

Participating Actor: Player

Stakeholders and Interests: The player may want to learn more about the contributors of the game.

Entry Condition: The player clicks on the "Credits" button to get information about the contributors.

Exit Condition: The player returns to "Main Menu" after getting information.

Success Scenario Event Flow:

1. The player clicks on the "Credits" button.

2. The player monitors the information and contact information of the contributors.

3. The player may return back to the main menu.

4.1.1.14 Use Case: Quit

Use Case: Quit

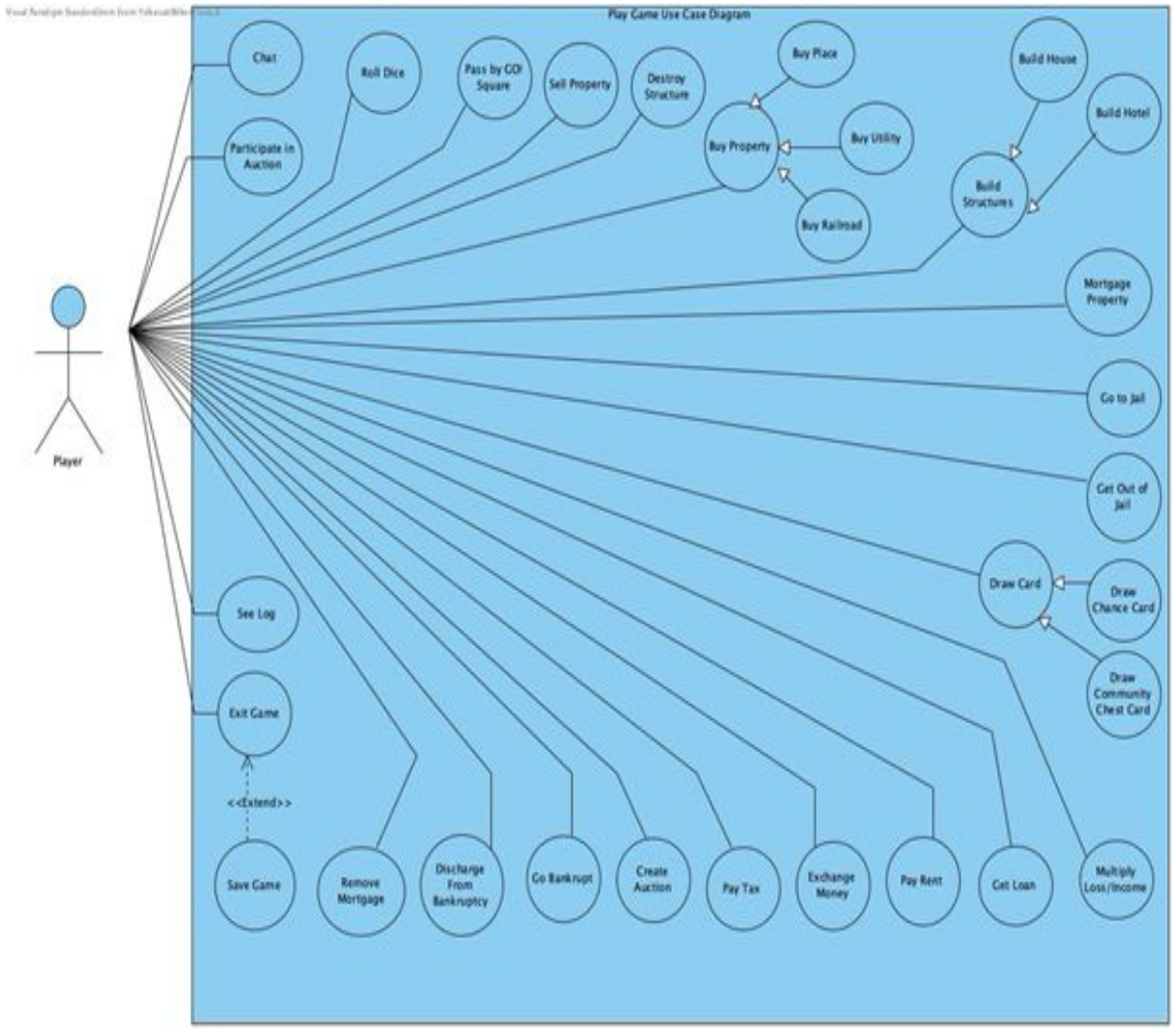
Participating Actor: Player

Stakeholders and Interests: The player may want to quit the game.

Success Scenario Event Flow:

1. The player clicks on the "Quit" button.
2. The game is closed after the click.

4.1.2 Play Game Use Case Model



Model 2: Use Case Model of the Play Game

4.1.2.1 Use Case: Chat

Use Case: Chat

Participating Actor: Player

Stakeholders and Interests: The player may want to chat with other players.

Entry Condition: The player is in the game.

Exit Condition: The player exited from the game.

Success Scenario Event Flow:

1. The player types the message to the chat entry area.

3. The player presses the "Send" button.

4. The corresponding chat message is sent.

4.1.2.2 Use Case: Roll Dice

Use Case: Roll Dice

Participating Actor: Player

Stakeholders and Interests: The player rolls dice in order to calculate the landing position of the pawn.

Entry Condition: The player is on his/her turn.

Exit Condition: The player clicks on the "End Turn" button.

Success Scenario Event Flow:

1. The player clicks on "Roll Dice Button" to roll the dice.

2. The final result of the rolled dice shows on.

3. The pawn moves automatically.

4.1.2.3 Use Case: Pass by GO! Square

Use Case: Pass by GO! Square

Participating Actor: Player

Stakeholders and Interests: The player may pass the "GO! Square" depending on the result of the sum of the dice.

Entry Condition: The player is on his/her turn.

Exit Condition: The player gets game money from the bank.

Success Scenario Event Flow:

1. After the player rolls the dice, he/she passes through the "GO! Square".

2. The player gets 200 amounts of game money for each pass through the "GO! Square".

4.1.2.4 Use Case: Buy Property

Use Case: Buy Property

Participating Actor: Player

Stakeholders and Interests: The player may want to buy property.

Entry Condition: The pawn of the player is landed on a property square that is not owned by another player.

Exit Condition: The player buys the property.

Success Scenario Event Flow:

1. The player has enough money to afford the property.

2. The player pays the bank for the property.

3. The player acquires the card of the property from the bank.

4.1.2.5 Use Case: Create Auction

Use Case: Create Auction

Participating Actor: Player

Stakeholders and Interests: The player wants to sell his/her property.

Entry Condition: The player selects "Auction" option.

Exit Condition: The player with the highest proposal gets the property of the player.

Success Scenario Event Flow:

1. The property of the player is offered different amounts of money by other players.

2. The player with the highest offer gets the property of the player.

3. The player gets the money from the purchaser.

4.1.2.6 Use Case: Participate in Auction

Use Case: Auction

Participating Actor: Player

Stakeholders and Interests: The player wants to participate in the auction created.

Entry Condition: The player selects "Auction" option.

Exit Condition: The auction ends and another player owns the auctioned property.

Success Scenario Event Flow:

1. The player bids money in the auction in the limit of his/her money.

2. The player gets the property if he/she has offered the highest money.

3. Previous owner of the property receives the money.

4.1.2.7 Use Case: Build Structures

Use Case: Build Structure

Participating Actor: Player

Stakeholders and Interests: The player may want to build structures on an owned property.

Entry Condition: The player is the one who plays the turn and bought all the properties with the same color.

Exit Condition: The player builds houses or hotels.

Success Scenario Event Flow:

1. The player has enough money to afford the structure.

2. The player satisfies the conditions to build the structure.

3. The player pays the bank for the structure.

4. The player acquires the corresponding structure from the bank to place on the property square.

4.1.2.8 Use Case: Go to Jail

Use Case: Go to Jail

Participating Actor: Player

Stakeholders and Interests: The player goes to jail.

Entry Condition: The player satisfies one of the requirements of going to jail.

Exit Condition: The pawn of the player is moved to the "Jail" section of the square.

Success Scenario Event Flow:

1. The player rolled double thrice, drew a "Go to Jail" card or the pawn of the player landed on "Go To Jail" square.

2. The pawn of the player goes to jail.

4.1.2.9 Use Case: Get Out of Jail

Use Case: Get Out of Jail

Participating Actor: Player

Stakeholders and Interests: The player should get out of jail in order to roll dice and continue the game.

Entry Condition: The player can roll the dice and get doubles, pay the fee or use the "Get Out of Jail" card drawn before to get out of the jail.

Exit Condition: The pawn moves from the "Jail" section to "Just Visiting" in the corresponding square.

Success Scenario Event Flow:

1. The player satisfies one of the conditions to "Go out of Jail".

2. The pawn moves to the "Just Visiting" section of the square.

4.1.2.10 Use Case: Draw Card

Use Case: Draw Card

Participating Actor: Player

Stakeholders and Interests: The player draws a card when the pawn of the player lands on "Draw Card" square.

Entry Condition: The player is on "Draw Card" square.

Exit Condition: The player draws a card and continues playing.

Success Scenario Event Flow:

1. The player draws a card.

2. The player applies what is written on the card.

4.1.2.11 Use Case: Multiply Loss/Income

Use Case: Multiply Loss/Income

Participating Actor: Player

Stakeholders and Interests: The proportion of income or loss in the next turn is determined by the player's click on "Multiple Loss/Income" button.

Entry Condition: The player wants to increase or decrease the amount of money that he/she pays or receives.

Exit Condition: According to the result of the player's button click, proportion of money changes.

Success Scenario Event Flow:

1. Player clicks on the "Multiply" button.

2. The system generates a random value between 0.5 to 2.

3. The proportion of change in payments or income money is changed by the according result.

4. The resulting money of the player is changed.

4.1.2.12 Use Case: Get Loan

Use Case: Get Loan

Participating Actor: Player

Stakeholders and Interests: The player may want to get a loan from the bank.

Entry Condition: The player loans money from the bank for a specific number of turns.

Exit Condition: The player returns the money to the bank in a specific number of turns.

Success Scenario Event Flow:

1. The player continues playing the game after loaning money.

2. The player pays back the money to the bank in a specific amount of turns.

4.1.2.13 Use Case: Pay Rent

Use Case: Pay Rent

Participating Actor: Player

Stakeholders and Interests: The player pays rent if the property square he/she landed is owned by another player.

Entry Condition: The player lands on an owned square by another player.

Exit Condition: The player has enough money to pay the rent and the owner player receives the rent from the player.

Success Scenario Event Flow:

1. The amount of money that the player decreases.

2. The amount of money that the property square owner increases.

4.1.2.14 Use Case: Exchange Money

Use Case: Exchange Money

Participating Actor: Player

Stakeholders and Interests: The player exchanges money to another currency for investment purposes.

Entry Condition: The player has enough money to exchange.

Exit Condition: The player's invested money exchanges to the selected currency.

Success Scenario Event Flow:

1. The player decides on investing money in other currencies.

2. The player clicks on the "Exchange" button.

3. The player selects a currency and exchanges an amount of money to selected currency.

4.1.2.15 Use Case: Pay Tax

Use Case: Pay Tax

Participating Actor: Player

Stakeholders and Interests: The player pays tax if his/her pawn lands on "Pay Tax" square.

Entry Condition: The player's pawn lands on "Pay Tax" square.

Exit Condition: The player pays the required money to the bank.

Success Scenario Event Flow:

1. The player pays tax if the amount of money on his/her account is sufficient.

2. If he/she can't afford paying tax, the player has to sell some of the structures built on their properties, put the properties to the auction, mortgage their properties or get a loan from the bank

4.1.2.16 Use Case: Go Bankrupt

Use Case: Go Bankrupt

Participating Actor: Player

Stakeholders and Interests: The player runs out of money in the game.

Entry Condition: The player has no money to perform required actions during the turn.

Exit Condition: The player loses the game or player is able to get out of bankruptcy

Success Scenario Event Flow:

1. The player isn’t able to get a loan from the bank, isn't able to sell buildings or mortgage properties.

2. The game is over for the player.

4.1.2.17 Use Case: Save Game

Use Case: Save Game

Participating Actor: Player

Stakeholders and Interests: The player saves up to three games in single play mode in order to continue later.

Entry Condition: The player currently plays a singleplayer game.

Exit Condition: The unfinished game is saved onto the “Load Game” section.

Success Scenario Event Flow:

1. The player clicks on the “Quit Game” button.

2. The player clicks on the “Save and Exit” button.

4.1.2.18 Use Case: Discharge From Bankruptcy

Use Case: Discharge From Bankruptcy

Participating Actor: Player

Stakeholders and Interests: The player can run out of money and can’t afford actions in their turn.

Entry Condition: The player won’t have enough money to afford actions.

Exit Condition: The player will sell property, mortgage property or get loan from the bank.

Success Scenario Event Flow:

1. The player needs to have more money in order not to go bankrupt.

2. As an option, the player sells or mortgages one of the owned properties if there is any. As another option, the player gets a loan from the bank.

4.1.2.19 Use Case: Sell Property

Use Case: Sell Property

Participating Actor: Player

Stakeholders and Interests: The player may want to sell their property in order to earn some money.

Entry Condition: The player owns a property square.

Exit Condition: The player gets the money that the property costs and doesn't own this property anymore.

Success Scenario Event Flow:

1. The player takes the money from the player that they sold the property.

2. The player gives the property card to the receiver player.

4.1.2.20 Use Case: See Log

Use Case: See Log

Participating Actor: Player

Stakeholders and Interests: The player may want to see the last activities in the game.

Entry Condition: The player is in the game.

Exit Condition: The player exited from the game.

Success Scenario Event Flow:

1. The player sees the log area in the game in “In Game Screen”.

4.1.2.21 Use Case: Mortgage Property

Use Case: Mortgage Property

Participating Actor: Player

Stakeholders and Interests: The player may want to mortgage his/her property and the property is unimproved.

Entry Condition: The player has the turn.

Exit Condition: The property has been mortgaged.

Success Scenario Event Flow:

1. The player selects the corresponding card from the deck.
2. The player clicks the “Mortgage” button.
3. The property is mortgaged.

4.1.2.22 Use Case: Remove Mortgage

Use Case: Remove Mortgage

Participating Actor: Player

Stakeholders and Interests: The player may want to remove the mortgage on their property.

Entry Condition: The player has the turn and the required money.

Exit Condition: The property has been taken back.

Success Scenario Event Flow:

1. The player pays required money to the bank.
2. The player gets the corresponding card back.
3. The mortgage on the mortgage removed.

4.1.2.23 Use Case: Destroy Structure

Use Case: Destroy Structure

Participating Actor: Player

Stakeholders and Interests: The player wants to destroy any structure that they previously built on a property.

Entry Condition: The property square which the player demands to destroy the structure on it needs to have at least one structure.

Exit Condition: The structure located on a property square is destroyed and the money is returned to the player.

Success Scenario Event Flow:

1. The structure on the property square is destroyed.

2. The player gets a specific amount of money in exchange for the destroyed structure.

4.1.2.24 Use Case: Exit Game

Use Case: Exit Game

Participating Actor: Player

Stakeholders and Interests: The player wants to stop playing the game.

Entry Condition: The player clicks on the "Exit Game" button.

Exit Condition: The game window is terminated by the player.

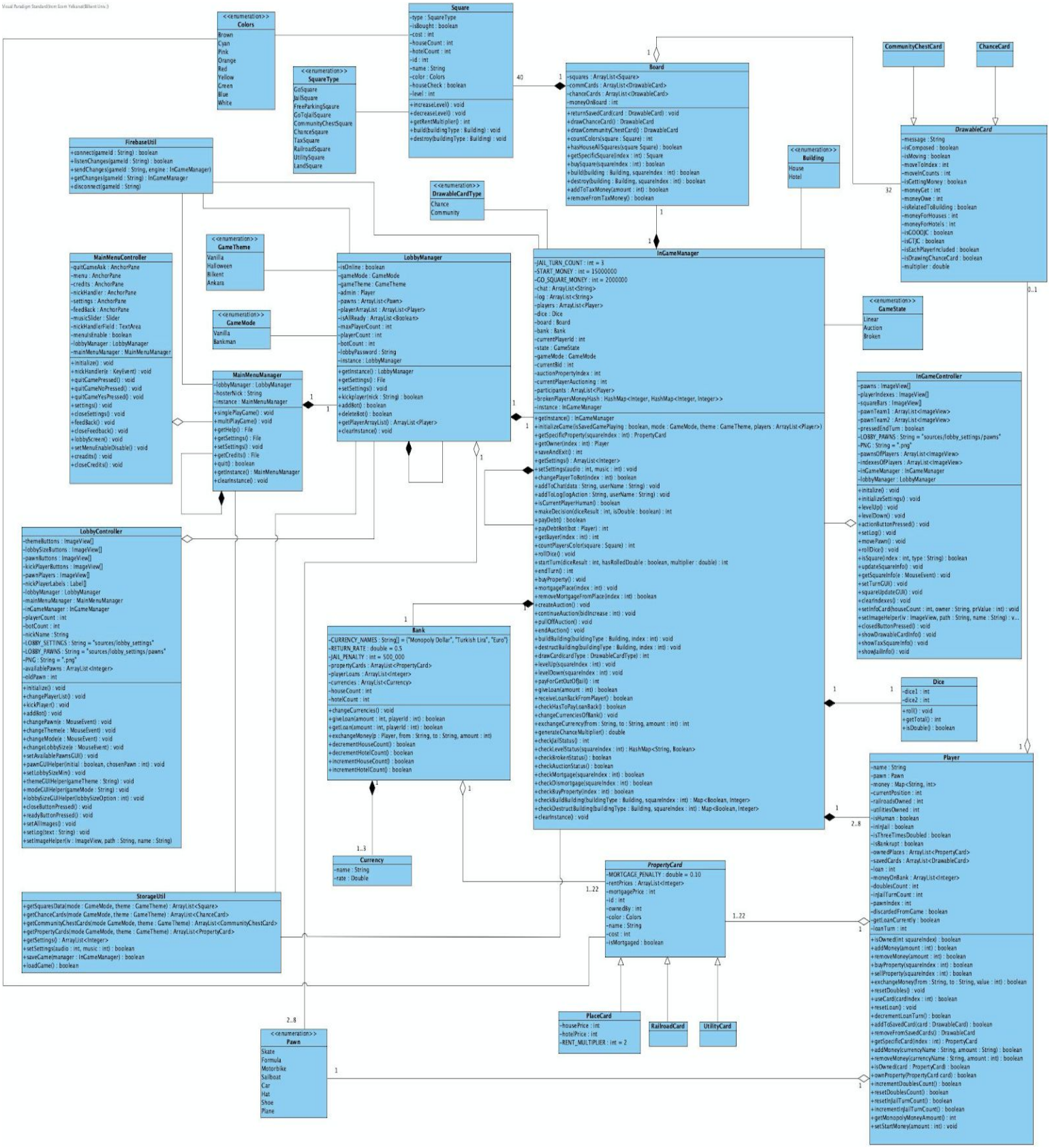
Success Scenario Event Flow:

1. The player is asked whether he/she wants to "Save and Exit" or "Only Exit" if the game is a single player game.

2. The player picks one of them and after, it is asked whether the player is sure of his/her choice if the game is a multiplayer game.

3. The game closes.

4.2 Class Model



Model 3 : Object Class Diagram

MainMenuManager: First engine class that controls the game until the player decides to play a game in either single or multiplayer game modes.

LobbyManager: Second engine class that controls the game until the initial game is started. Hence, while creating a game, when the player is selecting game theme, game mode, pawn etc. this engine controls all the selections.

InGameManager: Third and last engine which controls the initial game. Hence, players are able to play the game with that engine. This engine includes board data, dice data, player data and all the card data along with initial game rules.

Player: Basic player class that initialized during the game by inner engine. All player data hold in that class.

Board: The class that holds the data of the game board in the game.

Square: This class is a part of Board class which contains properties, corners and places that hold actions in the board. This class is a helper class to initialize the board in the game.

PropertyCard: Abstract class that holds the data of all property cards.

PlaceCard: The class that contains data of the property cards that indicate the town squares which can be bought by the player.

RailroadCard: The class that contains data of the property cards that indicate the railroad squares which can be bought by the player.

UtilityCard: The class that contains data of the property cards that indicate the utility squares which can be bought by the player.

Bank: Imaginary, standalone class for the bank object in the game. Handles the buying and selling mechanisms of the properties in the game. In Bankman mode of the game, it handles the loan and currency mechanisms.

Currency: The class that contains the currency data in the game. There are three currencies available in the Bankman mode of the game and this class will be used properly in the Bankman mode.

Dice: The class that contains the dice data in the game. Handles rolling the dice mechanisms.

DrawableCard: Abstract class that contains all the data of the chance cards and community chest cards in the game.

ChanceCard: Class that extends DrawableCard class. Indicates the chance cards in the game.

CommunityChestCard: Class that extends DrawableCard class. Indicates the community chest cards in the game.

StorageUtil: Utility class that handles offline in-game storage operations in the game.

FirebaseUtil: Utility class that handles online in-game storage operations in the game.

MainMenuController: Controller class that controls Main Menu Screen and corresponding FXML file in the game.

LobbyController: Controller class that controls Lobby Screen and corresponding FXML file in the game.

InGameController: Controller class that controls In Game Screen and corresponding FXML file in the game.

GameMode: Enumeration class that contains the game mode information in the game. There are only two game modes available which are Vanilla and Bankman.

GameTheme: Enumeration class that contains the game theme information in the game. There are four themes available which are Vanilla, Bilkent, Ankara and Halloween.

Pawn: Enumeration class that contains the pawn information in the game. There are a maximum number of eight pawns available. Even the name of these pawns will change by theme settings before starting the game, enumerations will remain the same.

Square: Enumeration class that contains square information in the board of the game. There are ten possible enumerations available for squares which are Go Square, Jail Square, Free Parking Square, Go To Jail Square, Community Chest Square, Chance Square, Tax Square, Railroad Square, Utility Square and Land Square.

Building: Enumeration class that contains building information in the game. Used only in functions to indicate which building is going to be inserted in a specific square. Possible building types are house and hotel.

Colors: Enumeration class that contains color info for the game that used in Square and Property Card classes.

DrawableCardType: Enumeration class that contains drawable card info in the game. Only used in InGameEngine class to differentiate between the drawable card input coming from the front-end of the game.

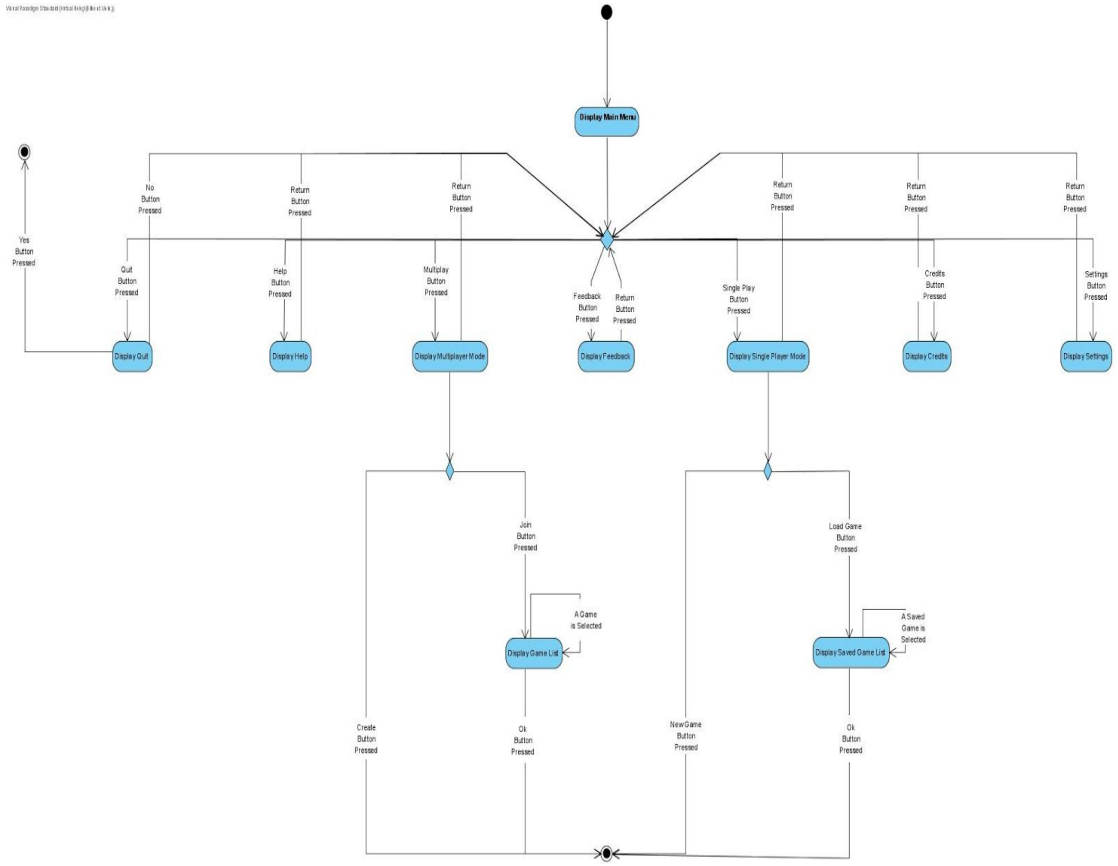
GameState: Enumeration class that contains game state info in the game. There are three possible game states available in the game which are Linear, Broken and Auction. Linear corresponds to normal game-play. Auction states, as the name suggests, breaks the linearity of the game, hence a state. Broken state pauses the game when a player gets broke.

4.3 Dynamic Models

4.3.1 Activity Diagrams

4.3.1.1 Main Menu Activity Diagram

In the Main Menu, players can choose seven activities. They can quit the game after pressing the Quit button then the Yes button. If players need help, they can click the help button to learn about game rules and how the game works then return to the main menu with the Return button. Players can learn about developers when they click Credits after that they can return to the Main Menu with the Return button. Players can view and change settings. Players can start a single player game with pressing Single Player. After that, players can load a saved game from the saved list via Load Game Button. They can select a saved game and start it with pressing the Ok button. Players can also create a new game with New Game Button. If players choose Multiplay, they have two options: Join and Create. If players click the Join button, a game list appears. They can choose a game and press the Ok button to join. Players can choose the Create Game option to create a lobby.



Model 4: Main Menu Activity Diagram

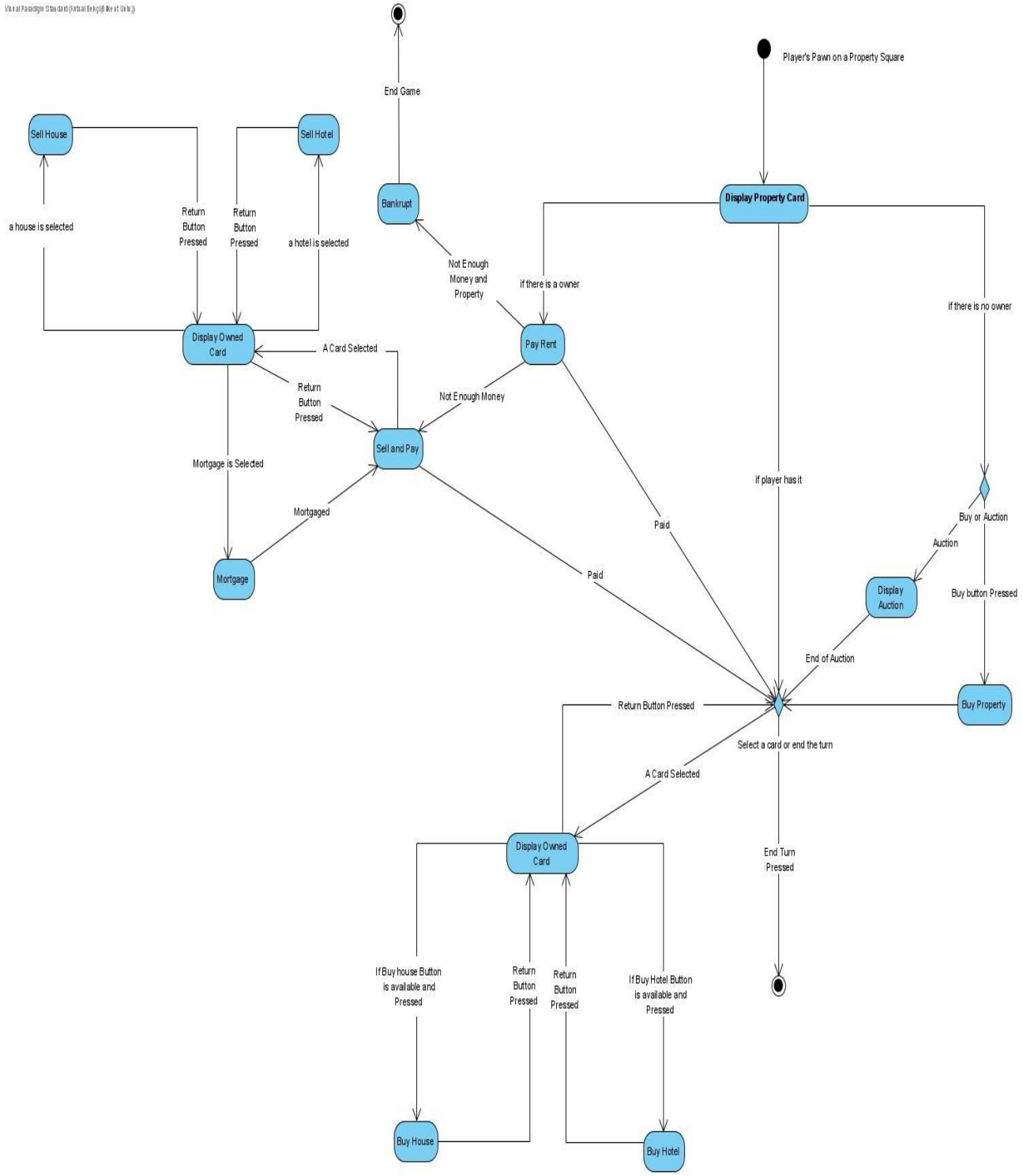
4.3.1.2 Property Activity Diagram

When a player's pawn on a Property Card, several options occur. If there is an owner of that card, the player playing the turn has to pay rent to the owner of the card. However, the player playing the turn has not enough money, he/she has to sell a house and hotel or mortgage the property. Player can choose a card from his card list then he/she chooses a house or hotel and sells it until he/she can pay the rent. If the player has not enough money and property, he/she goes bankrupt and the game is over for the player playing the turn.

If the rent is paid, the player moves to the End turn or Buy Property stage.

If there is no owner of that property card, player playing turn can buy that property with pressing the Buy Button. If the player does not buy the property, there will be an auction and he/she can join the auction. After auction or purchase, the player moves to the End turn or Buy Property stage.

After all these options, the player can select a card to buy a house or hotel if he/she has enough money then return the card list with the Return button or he/she can end the turn with the End Turn Button.



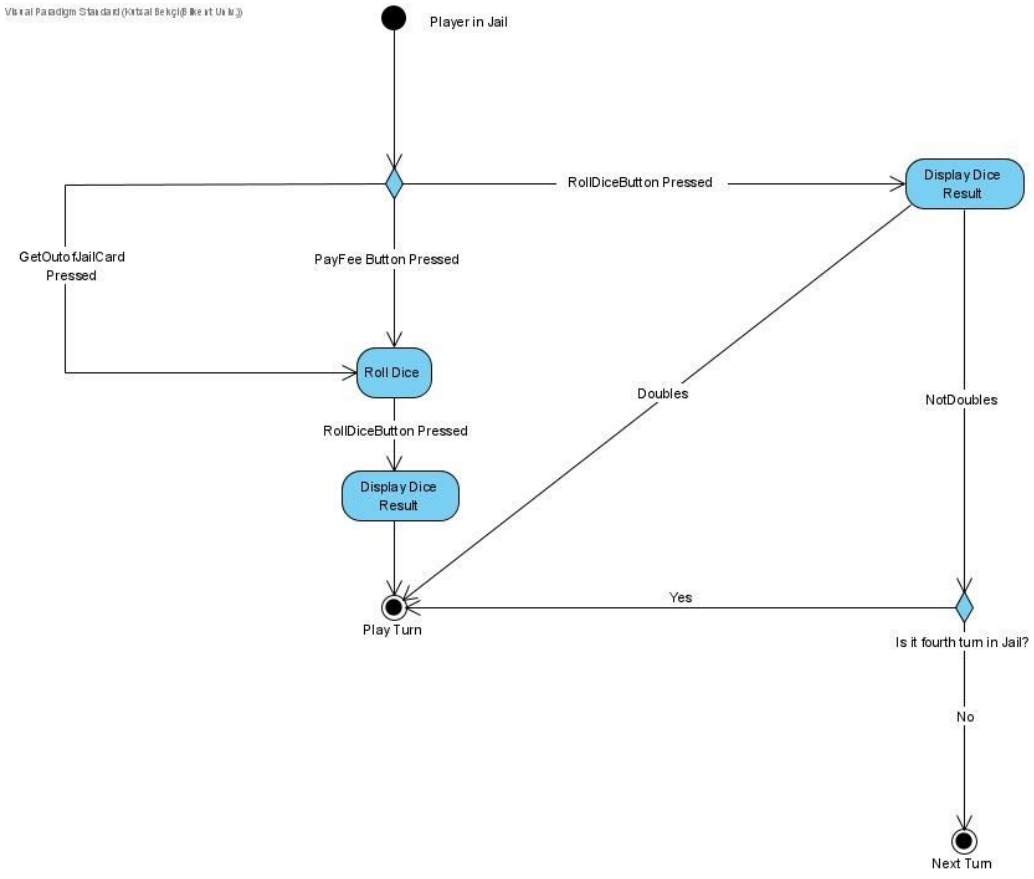
Model 5: Property Activity Diagram

4.3.1.2 Jail Activity Diagram

If Player who plays the current turn is in jail, there are three options that the player can choose: Get out of the jail with "Get Out of Jail Card", get out of the jail with paying "jail fee" and get out of the jail with dice. If the player has the "Get Out of Jail card", he/she can use it to get out of jail with pressing "GetOutOfTheJailCard Button". Then he/she can roll dice with the "RollDice" button and play his/her turn.

He/She can pay the fee with "PayFee" Button. Then he/she can roll dice with the "Roll Dice" button and plays his/her turn.

The player can choose rolling dice with the "RollDice" button and try his/her luck. If the result of the dice rolled is a double, the player can play his/her turn freely. If the player is in the jail for four turns, the player can play his/her turn freely.

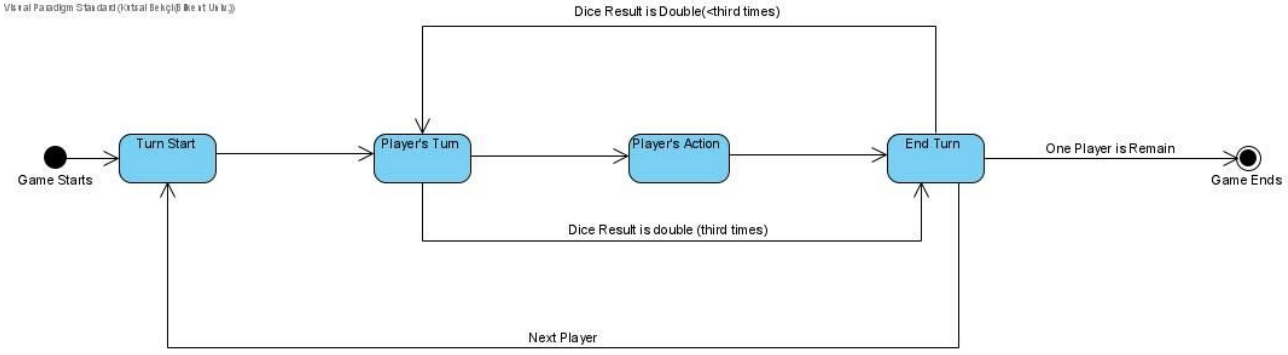


Model 6: Jail Activity Diagram

4.3.2 State Diagrams

4.3.2.1 Turn State Diagram

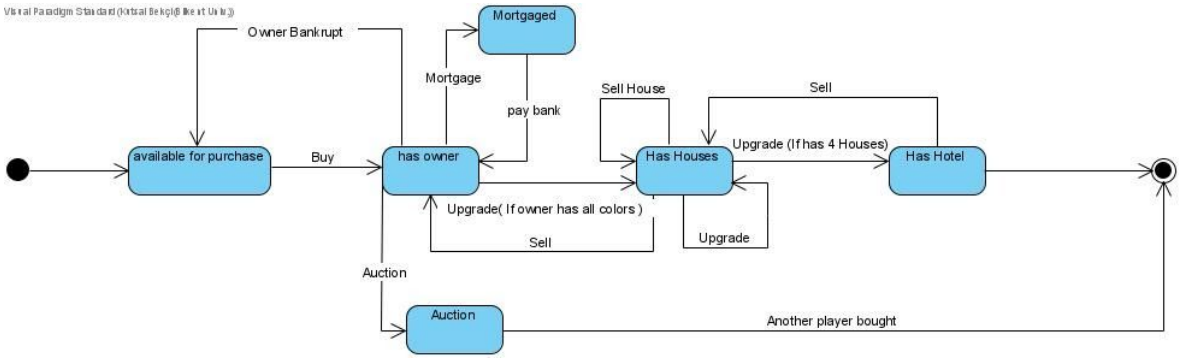
This is the Turn state diagram. the player chooses his actions when it is his/her turn. If the result of the dice rolled is a double, he/she can play one more turn. After the player ends his/her turn, the next player's turn is started. If one player is left, the game ends.



Model 7: Turn State Diagram

4.3.2.2 Property State Diagram

A property card can be bought by a player if it is available for purchase and the player's pawn is on the corresponding square. A player can upgrade his properties by buying houses. If the property has 4 houses, it can be upgraded by buying a hotel. The player that has the property card with houses and hotel can sell his hotel then his houses if he/she needs. He/she can mortgage his/her property and get back by paying the bank. The player can make an auction to sell his/her card.

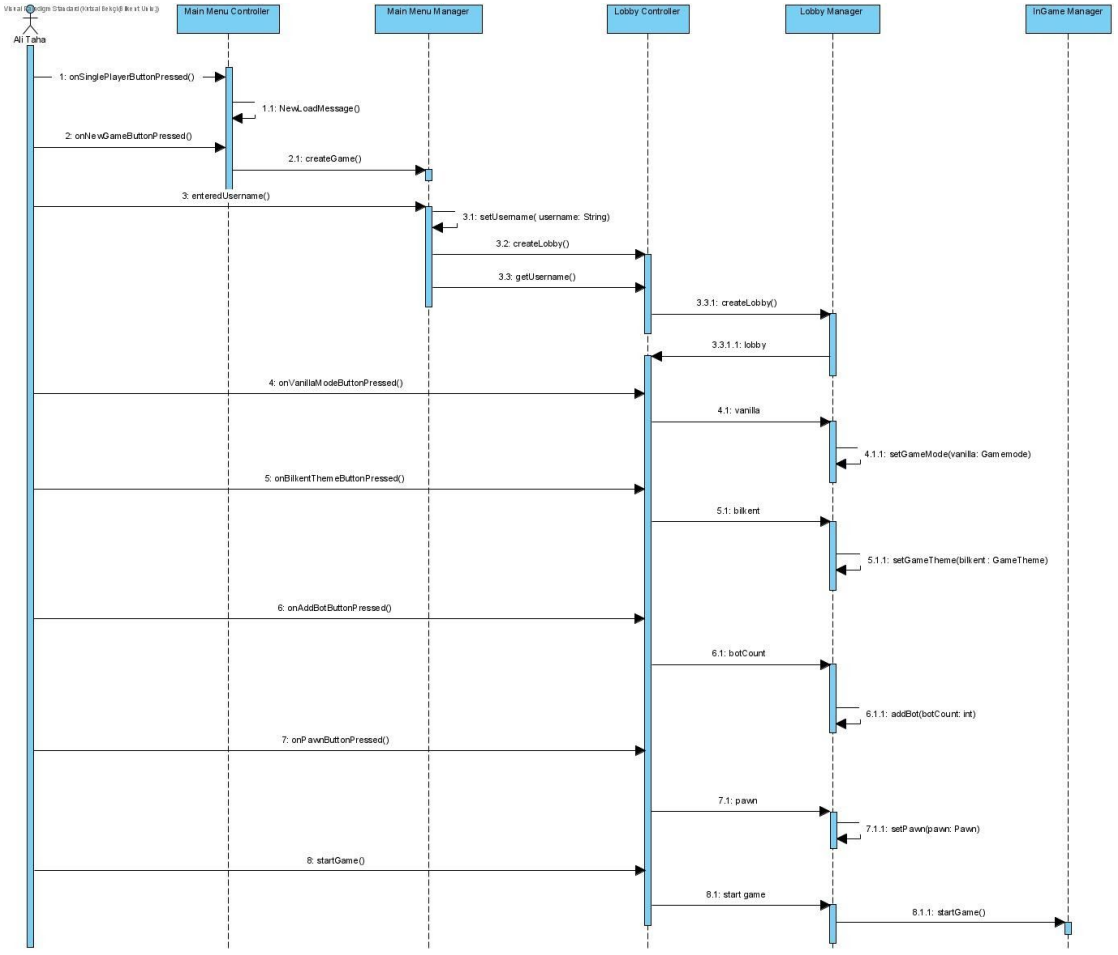


Model 8: Property State Diagram

4.3.3 Sequence Diagrams

4.3.3.1 Start Game (Single Player) Sequence Diagram

Scenario: Ali Taha wants to play Yolopoly. He starts the game. He wants to play single with Vanilla mode, and he clicks Single Player, then he clicks New Game Button. He is in the lobby. He chooses Vanilla mode. After he chooses the mode, he chooses the Bilkent theme. He can adjust the number of bots. He decides the number of the bots and his pawn. Then he starts the game with the ready button.

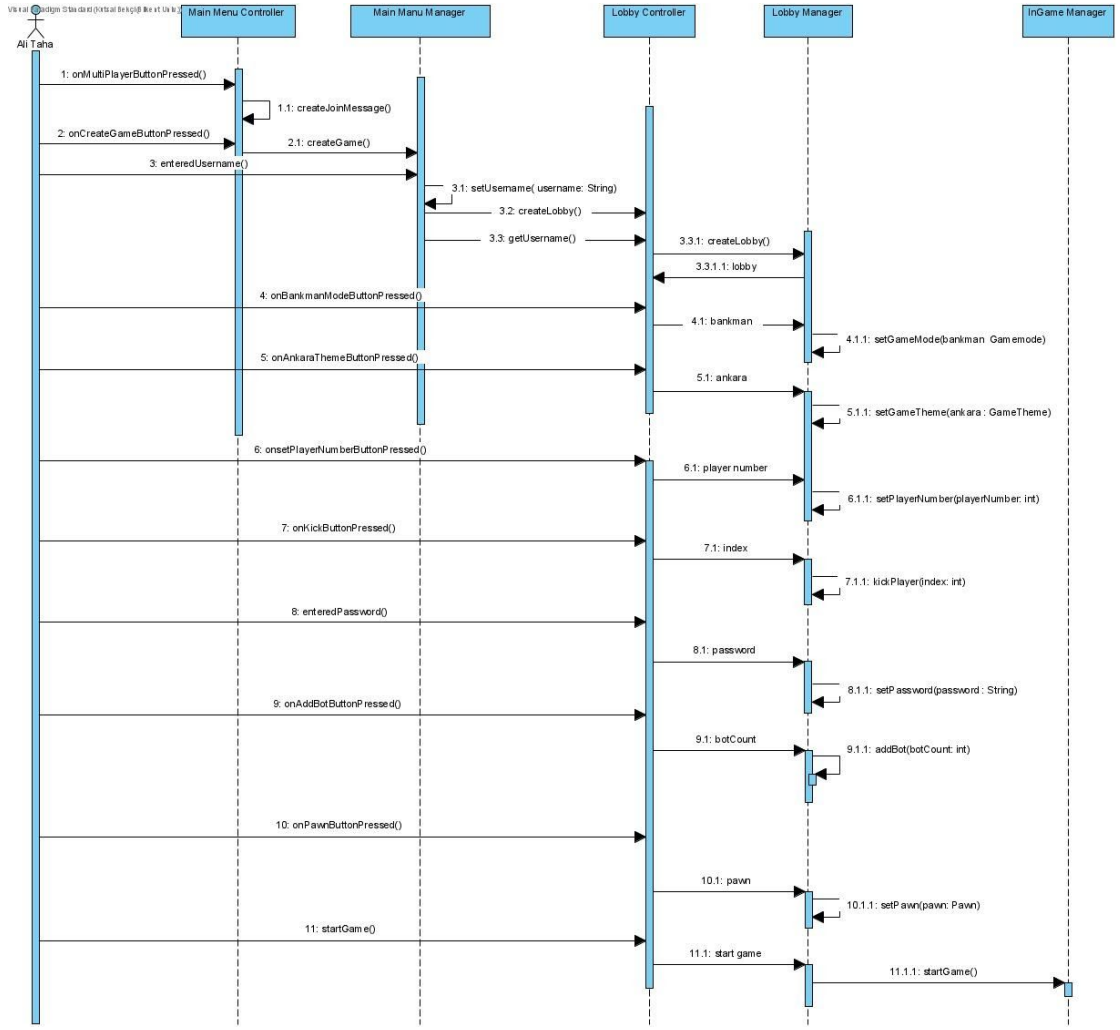


Model 9: Single Player Start Game Sequence Diagram

4.3.3.2 Start Game (Multiplayer) Sequence Diagram

Scenario: Ali Taha wants to play Yolopoly. He starts the game. He wants to play with other players, he clicks Multiplayer. After he clicked, he can choose Create Game or Join Game. If Ali Taha chooses Create Game, he can adjust

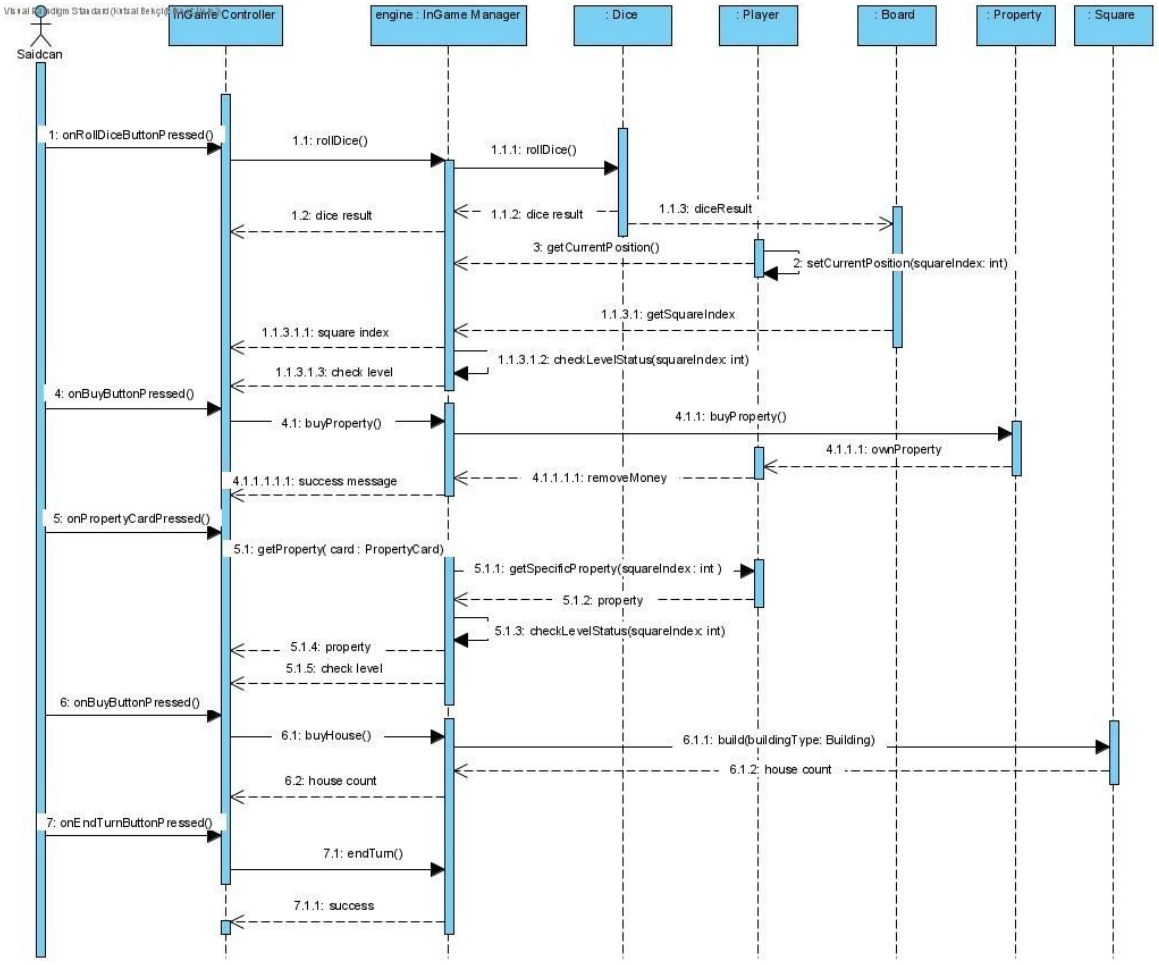
lobby settings. Ali Taha wants to adjust lobby settings and he clicks Create Game. While Ali Taha is waiting for his friends to enter his lobby, he changes the lobby's settings. He chooses Bankman mode. He chooses the Ankara theme. He sets the number of players which is six because Yolopoly can be played with 2-8 people. He realized that he did not set a password when a stranger player joined his lobby. Ali Taha kicks the stranger player and sets a password to his lobby. Two of his friends join his room. He adds three bots to the room. Ali Taha and their friends choose their pawns and then they click the roll dice button to determine the order. After that they click the ready button. If all of them click the ready button, the game will start.



Model 10: Multiplayer Start Game Sequence Diagram

4.3.3.5 Turn Sequence Diagram

Scenario: Saidcan and Ali Taha play Yolopoly. It is Saidcan's turn. Saidcan clicks roll dice button to roll the dice and eight comes. Saidcan's pawn moves eight blocks. His pawn is on a property that no one has that and Saidcan wants to buy this property. He clicks buy button and takes the property. After he bought the property, he wants to add a house to one of his cards. He chooses a card and buys a house to that card. He chooses to end turn, and his turn is over.



Model 11: Turn Sequence Diagram

4.4 User Interface

The display in the game can be divided into four sections. The explanation of each section is explained below.

4.4.1 Main Menu Screen

This is the screen that welcomes the player when the player clicks on the game icon. This screen contains Single Player, Multiplayer, Help, Settings, Feedback, Credits and Quit Game buttons.

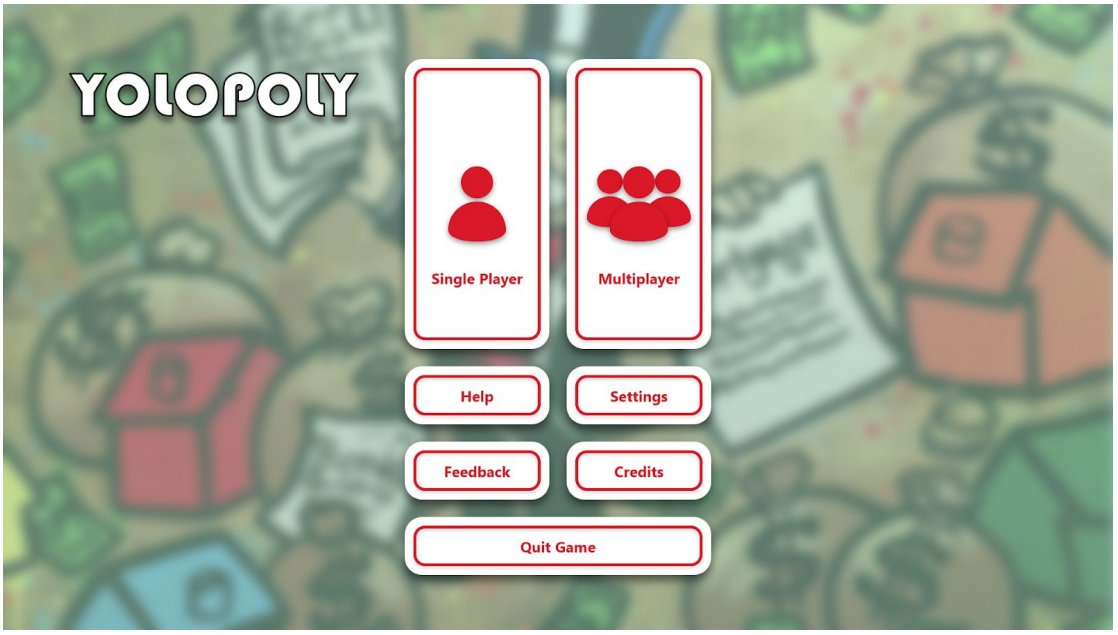


Figure 21: Main Menu Screen Mockup

Single Player button will open a dialog box and asks the player to either start a new game or continue from the previously saved game. If the player clicks on the New Game button, then the player will be navigated to the choose settings screen. If the player clicks on the Load Game button, then a dialog box including saved games with their game mode and game theme will be shown. After selecting a saved game, the player will be navigated to the Game Board screen.

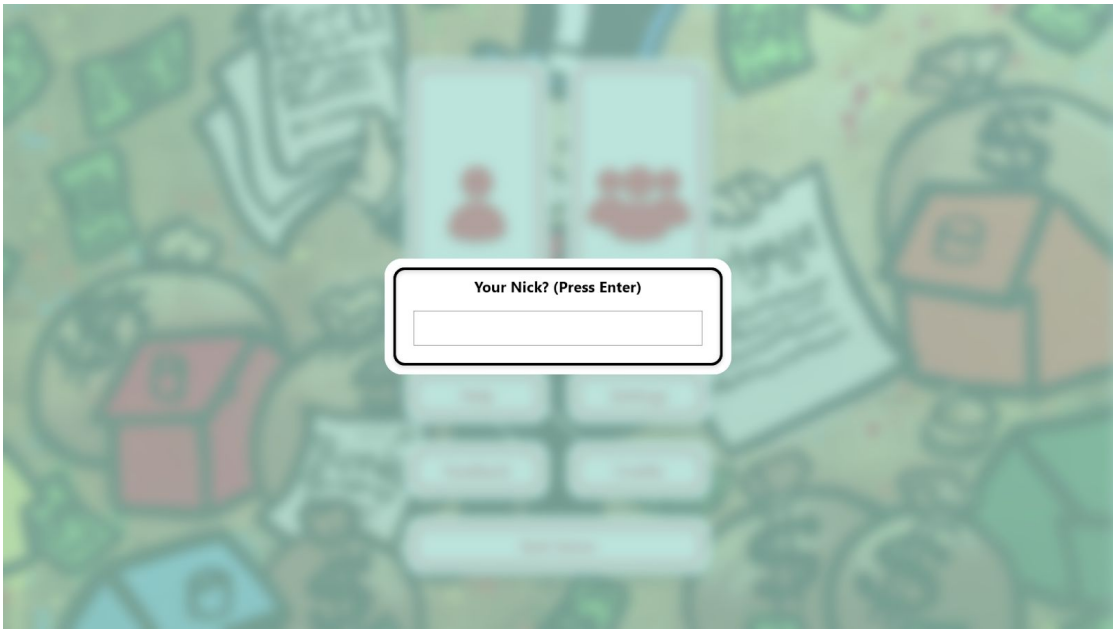


Figure 22: Get Users Nick Dialog Box Mockup

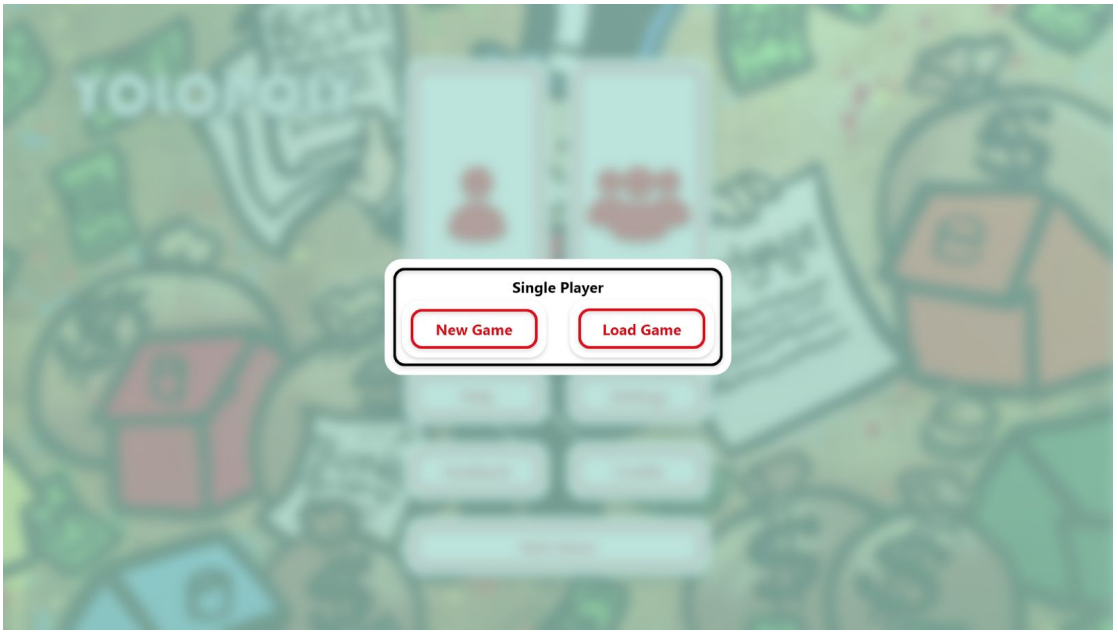


Figure 23: Single Player Dialog Box Mockup

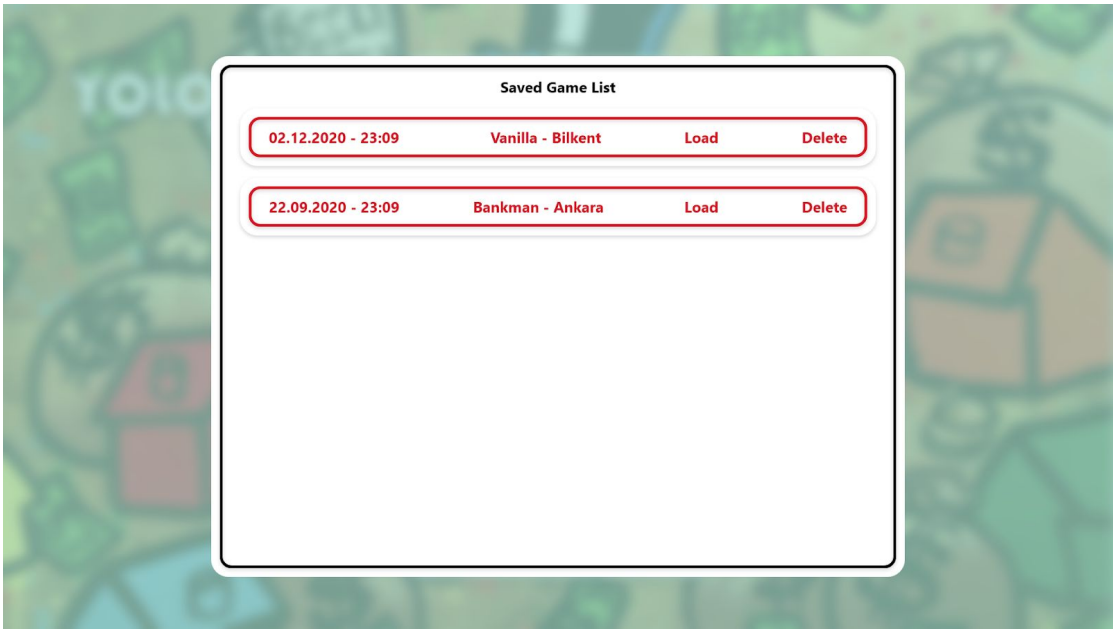


Figure 24: Load Game Dialog Box Mockup

If the player clicks on the Multiplayer button, then another dialog box will ask the player to either select create a new game or join to a game. If the player clicks on the Create Game button, then the player will navigate to another formation of Choose Settings screen which we named Lobby. If the player chooses to click on the Join button, then another dialog box with available online games will be shown. In this dialog box, players can select online games with different game modes and different game themes. In addition, if the created game is secured with a password, then another dialog box to enter the password of the room will be shown before entering the lobby.

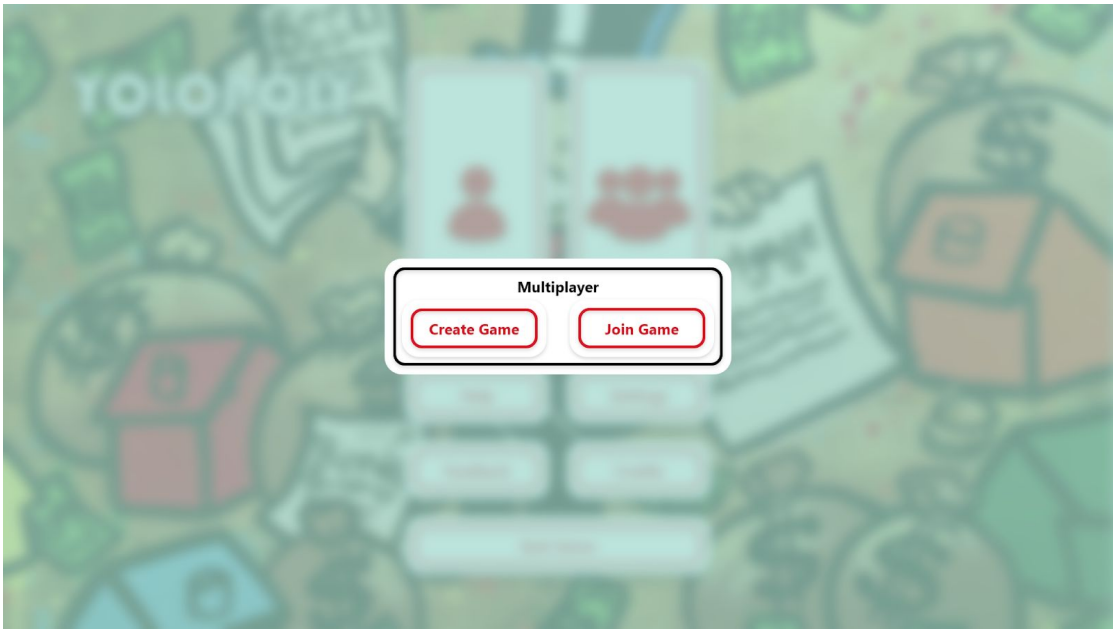


Figure 25: Multiplayer Dialog Box Mockup

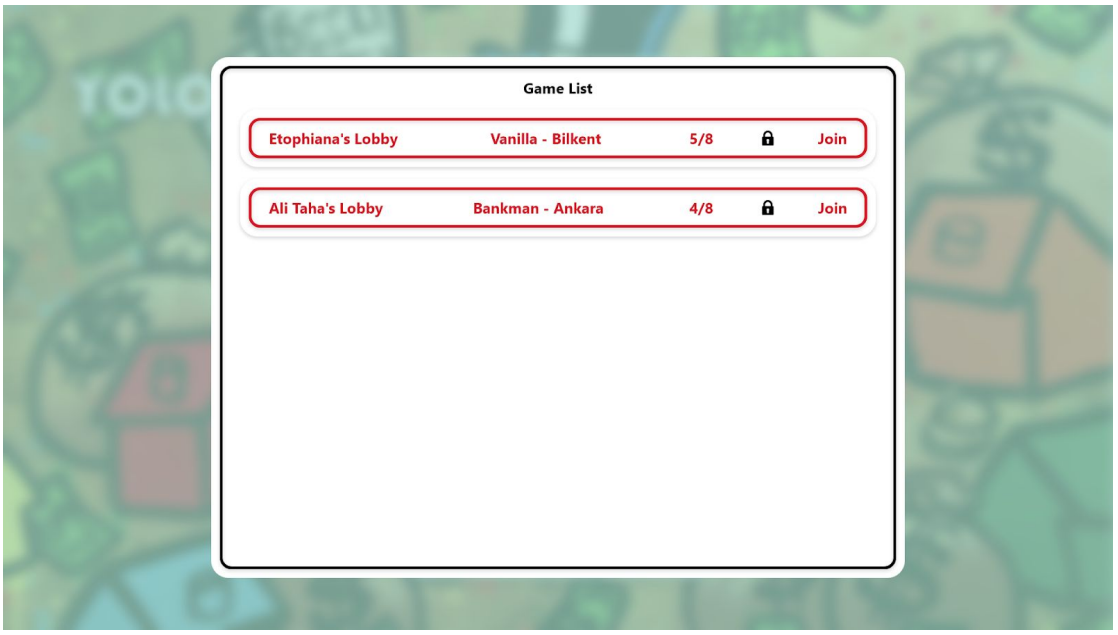


Figure 26: Listed Games Dialog Box Mockup

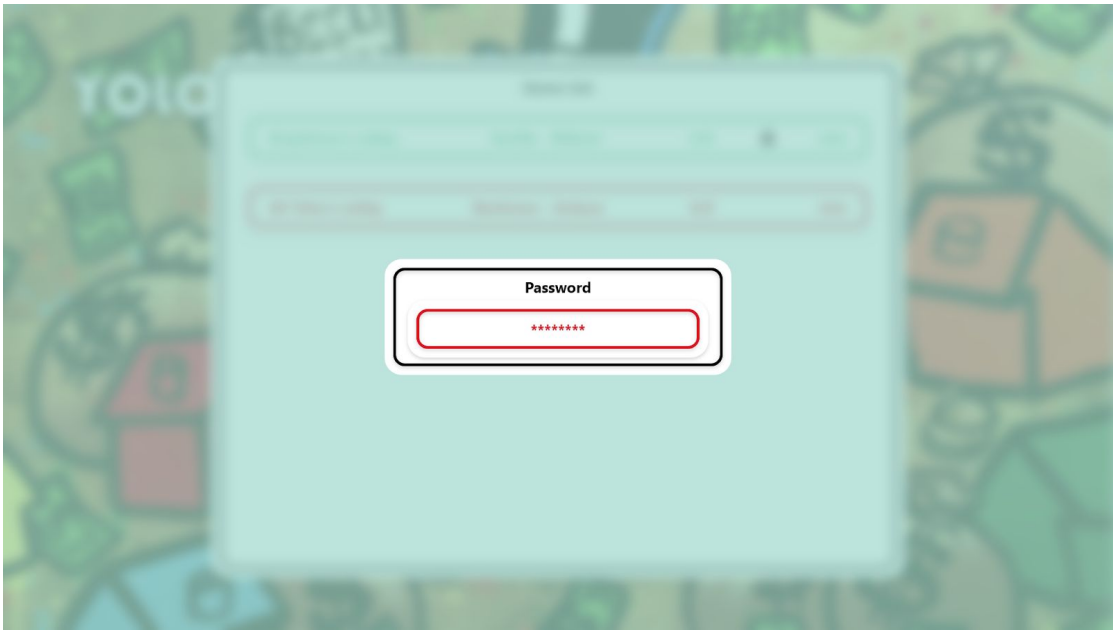


Figure 27: Password Entry Dialog Box Mockup

If the player clicks on the Help button, then a dialog box which includes tutorials about the game will be shown.

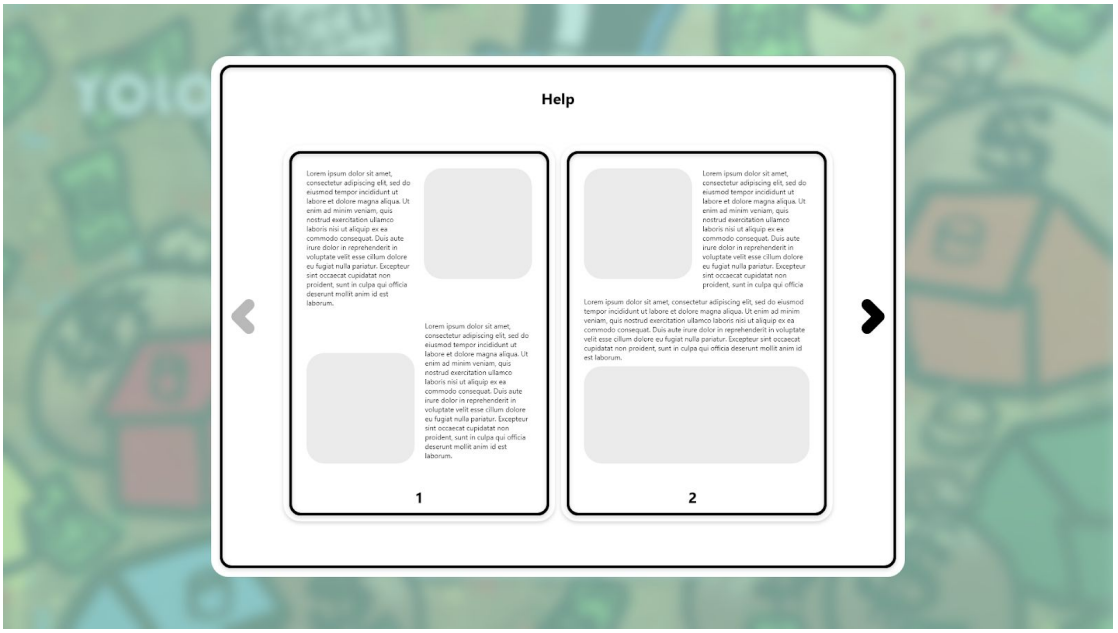


Figure 28: Help Dialog Box Mockup

If the player clicks on the Settings button, then a dialog box which includes two sliders to adjust the volume of the in-game music and effects will be shown. In addition, the player can click on the icons to close and open the sound respectively.

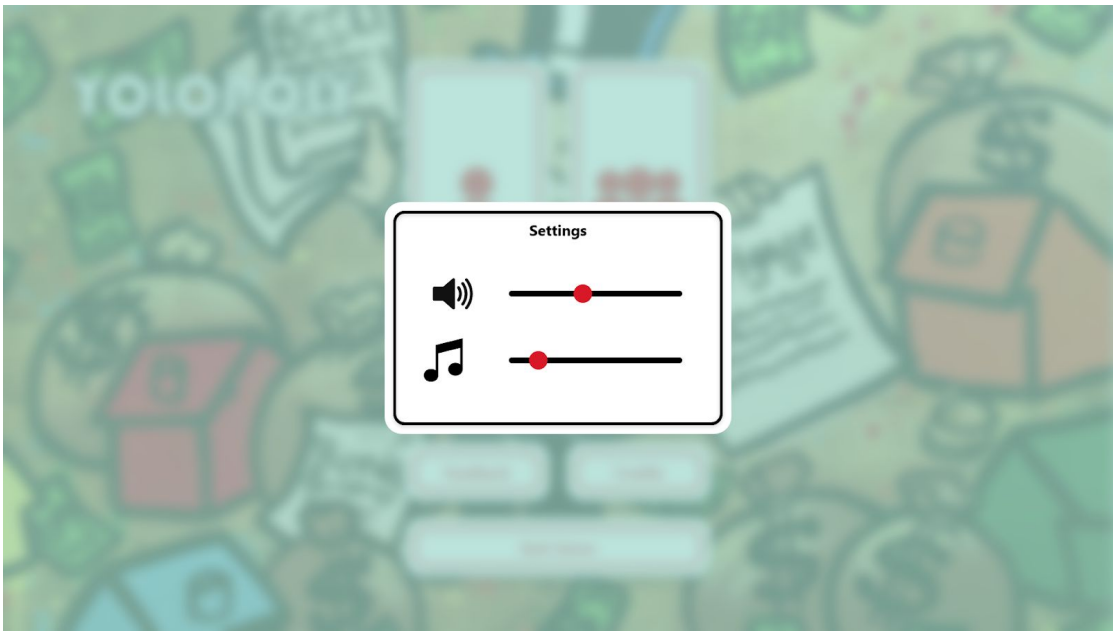


Figure 29: Settings Dialog Box Mockup

If the player clicks on the Feedback button, a dialog box which includes a text field and a send button will pop up. Players can send bug reports and suggestions by using this button.

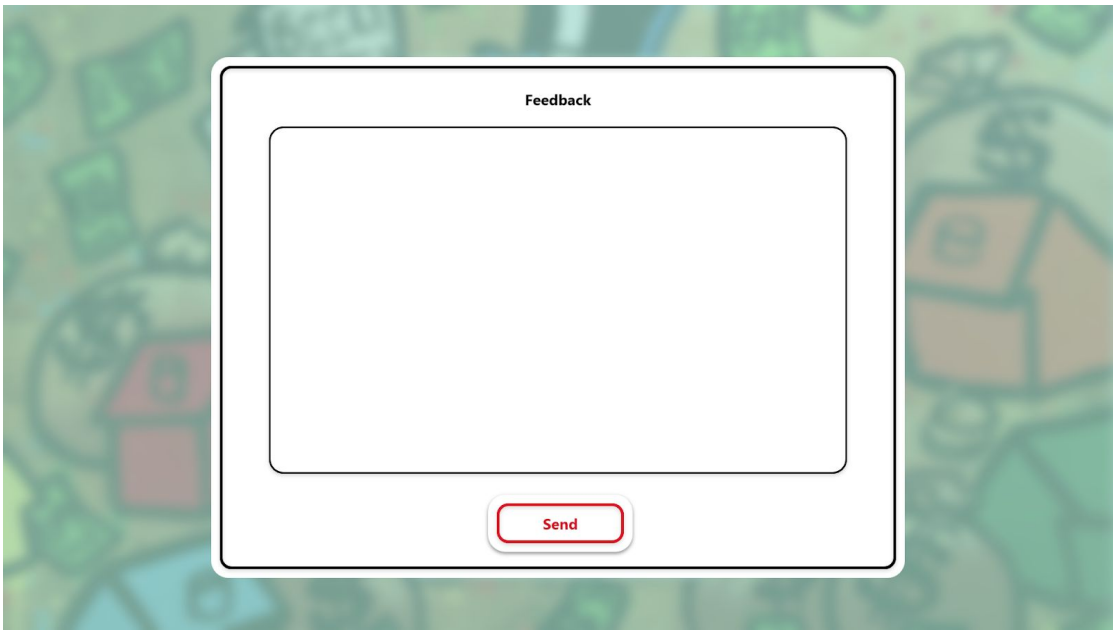


Figure 30: Feedback Dialog Box Mockup

If the player clicks on the Credits button, a dialog box including the information about the creators of the game will be shown.

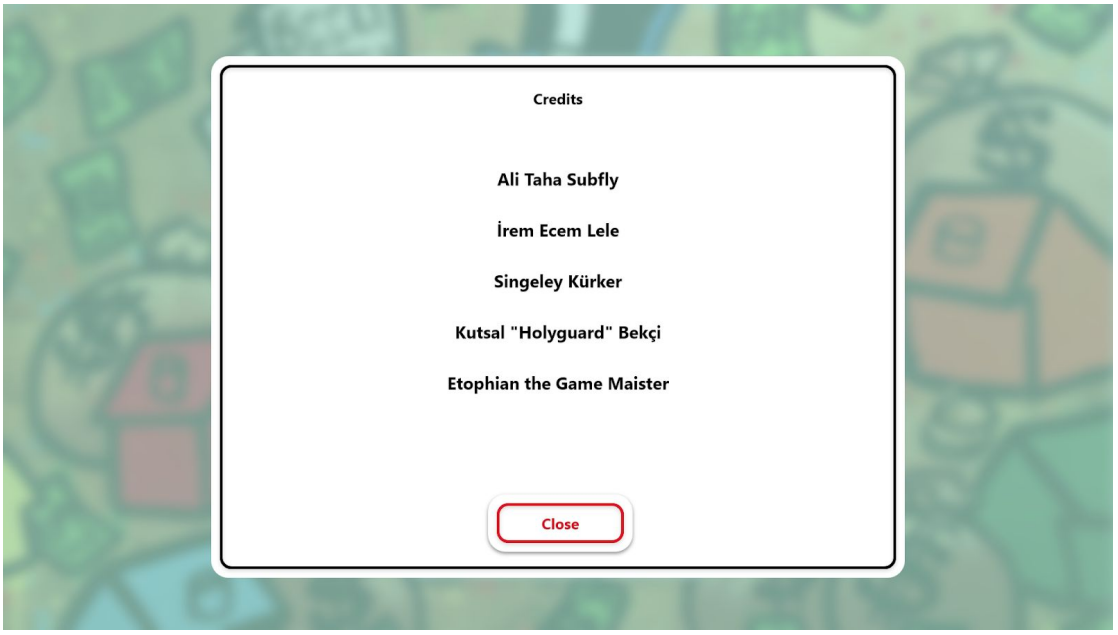


Figure 31: Credits Dialog Box Mockup

And finally, if the player clicks the Quit Game button, the game will be closed.

4.4.2 Choose Settings Screen

Basically, the player can see this screen when the player chooses to create a new game as a Single Play game. This screen is the refined version of the Lobby screen. The player can choose the mode, theme, bot count and the pawns in the game from this screen.

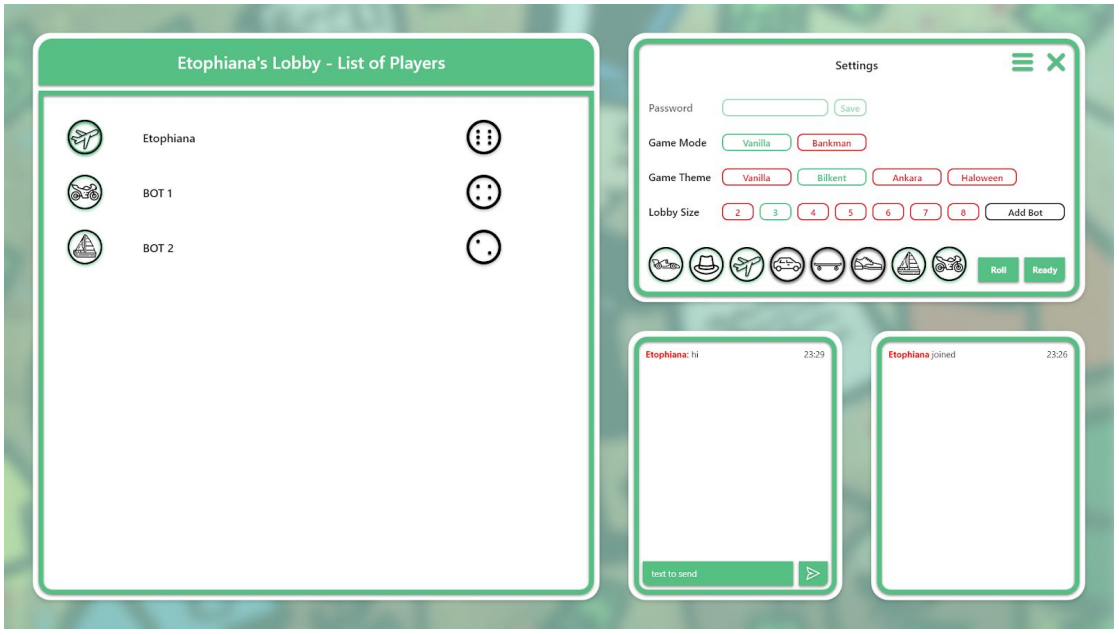


Figure 32: Choose Settings for Single Play Game Screen

4.4.3 Lobby

This is the screen that each player will see if the player has selected to play a Multiplayer game. All the features on that screen are enabled for the player who created the game including changing lobby password, game mode, game theme, number of players and kicking the players who joined the game. On the other hand, players who join that game are only able to choose a pawn and a nickname from that screen. After rolling the dice to select who will start the game and all the players have stated themselves as ready, the game will be automatically started.

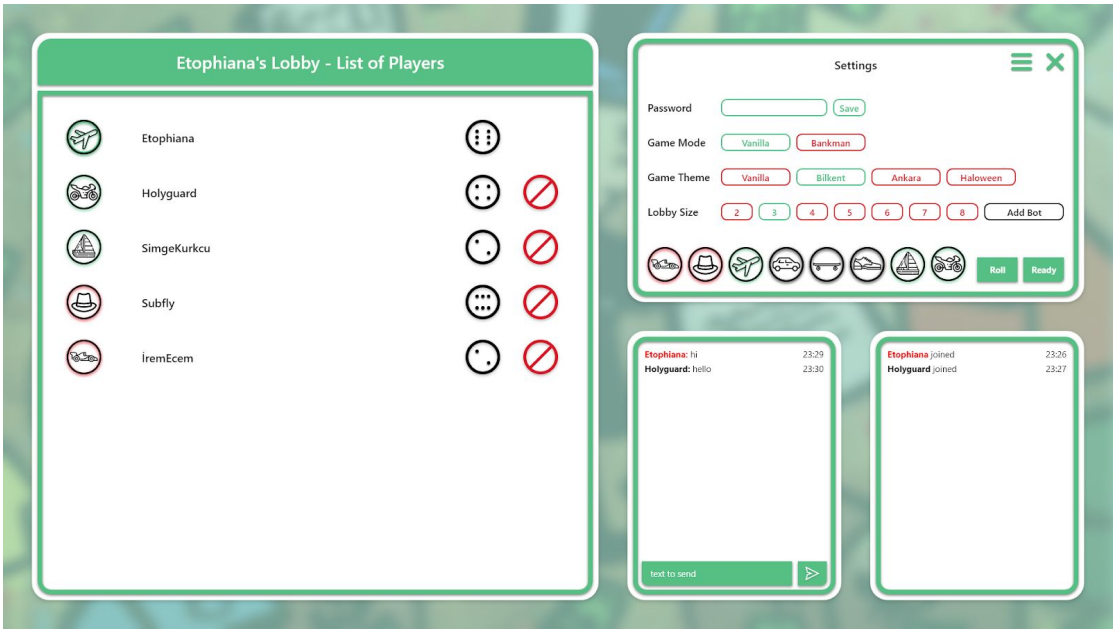


Figure 33: Lobby Screen for the Host

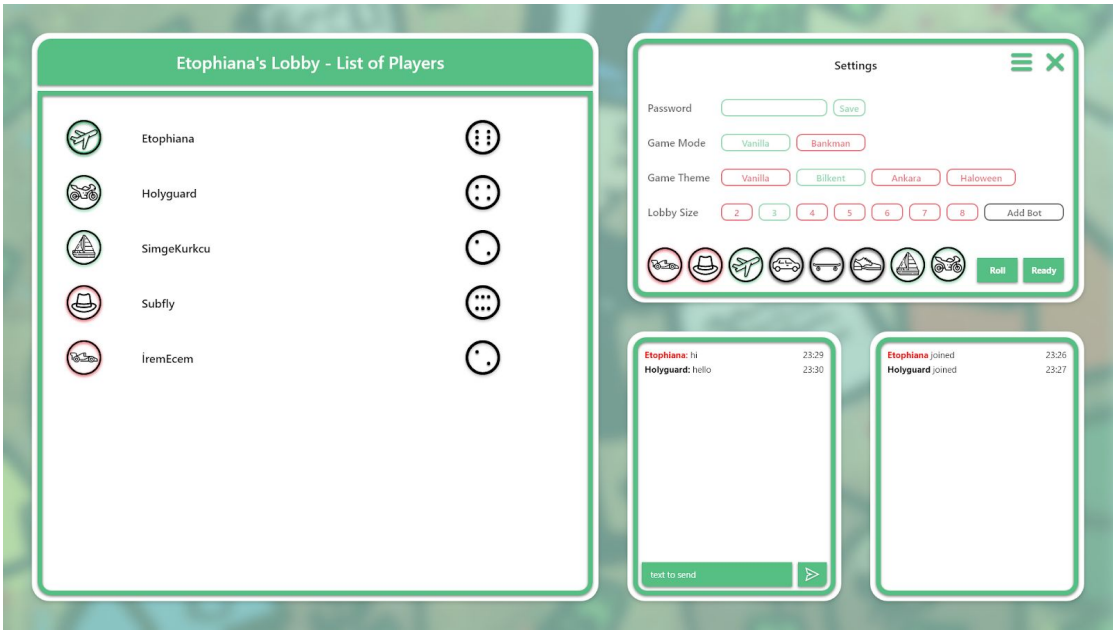


Figure 34: Lobby Screen for the Joined Players

4.4.4 In Game Screen

This is the screen that each player will see when they finally start to play the game, after selecting the settings. This screen contains three sections. Each section is explained below.

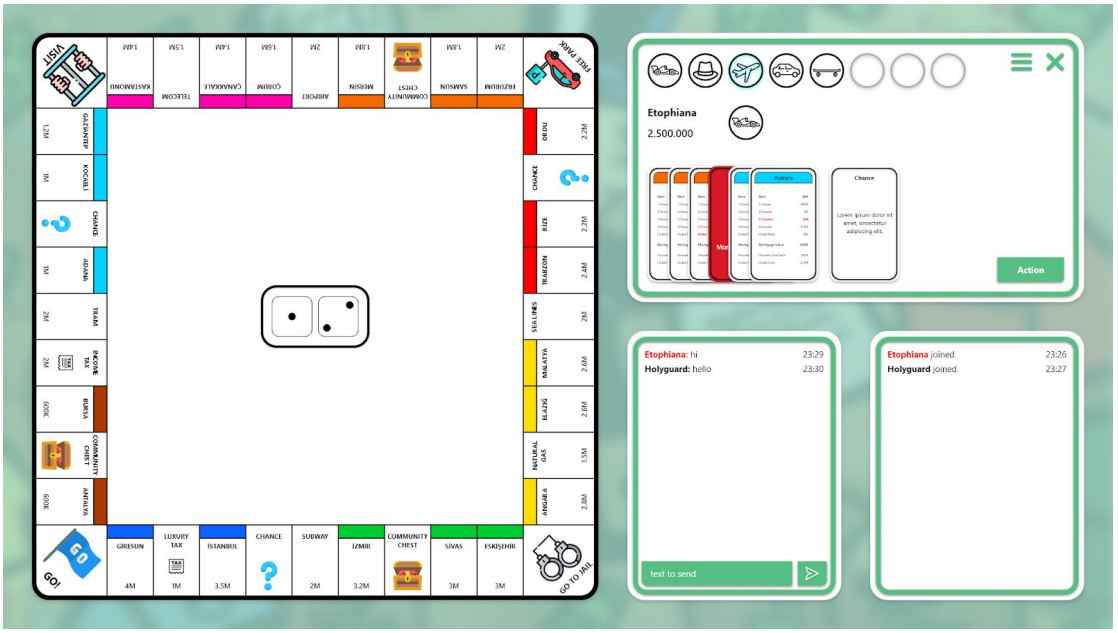


Figure 35: In Game Screen Mockup

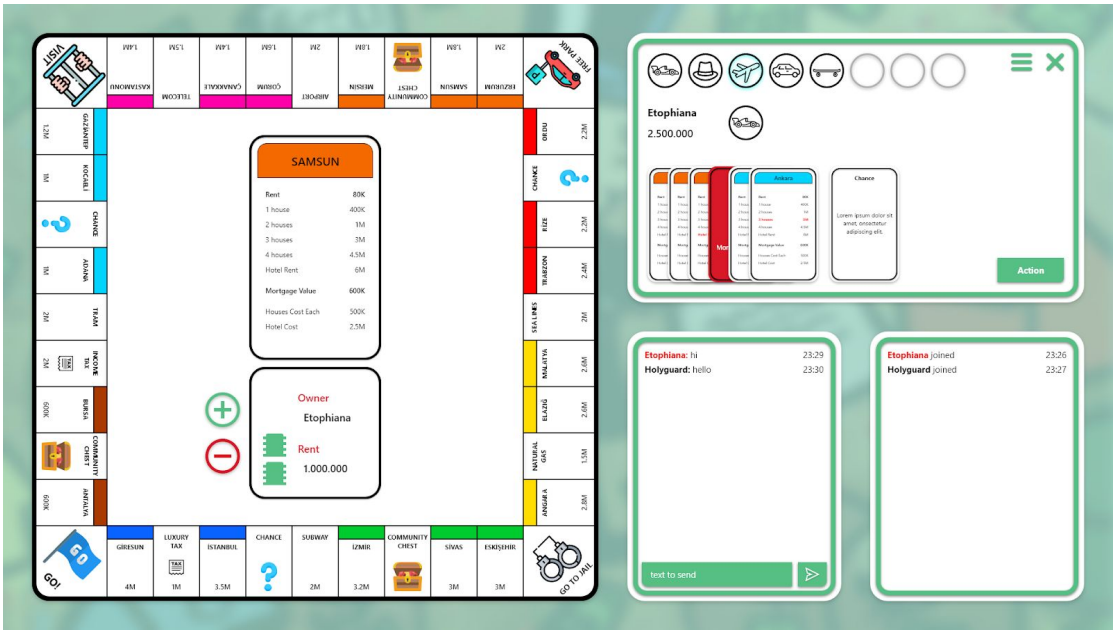


Figure 36: In Game Screen Mockup Property Info

4.4.4.1 Board

This is the section of the in-game screen where the board of the game is shown to the players. The Roll The Dice and End Turn buttons are available in the middle of that section. Only clickable area is that button in this section as the other features are automatically handled by the game engine. Pawns of the players will be moved clockwise in the squares of that area. Each property

group with colors and other types of squares including corners are displayed in this section.

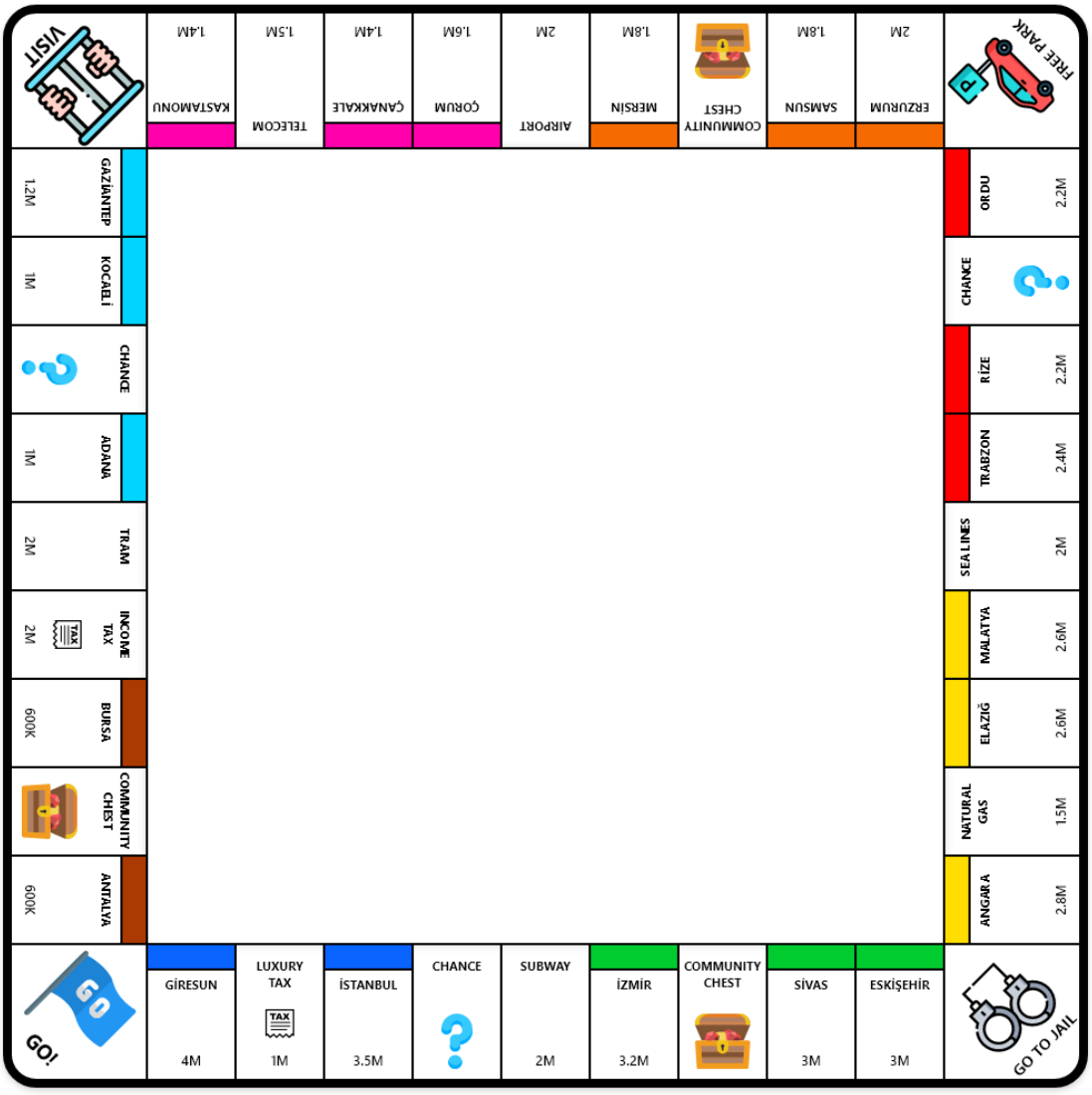


Figure 37: Game Board Mockup

4.4.4.2 Deck

This is the section that contains players in the game and the hands of each player. Initially, this area shows the deck of the player-self but the player can see other players' decks but clicking their pawns. In Bankman game mode, the currency rates and the loan of the player is also shown in that section. In addition, save-able cards drawn from Community Chest Cards and Chance cards are shown in this section. Players can select an owned property in this section in order to sell and mortgage their properties, build structures on them and see the rent prices in that section. Also, information about

nickname and the money of the player and grouped buttons of Settings and Quit are shown in this section. Players can adjust the sound of the game or quit/exit the game from these buttons with the dialog boxes shown.

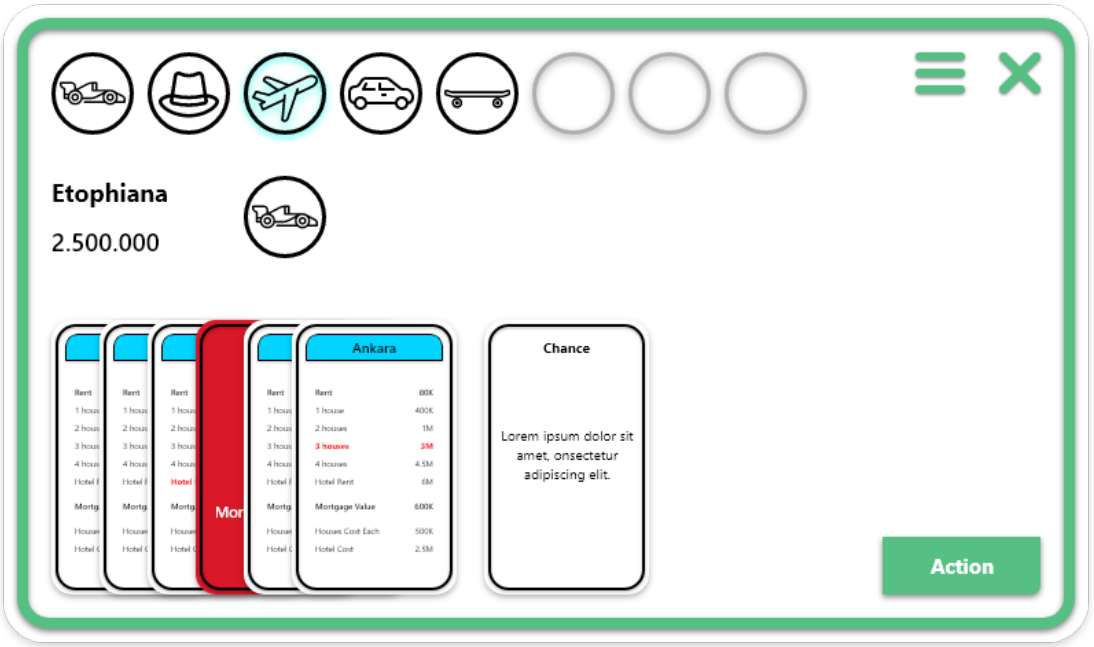


Figure 38: Deck Mockup

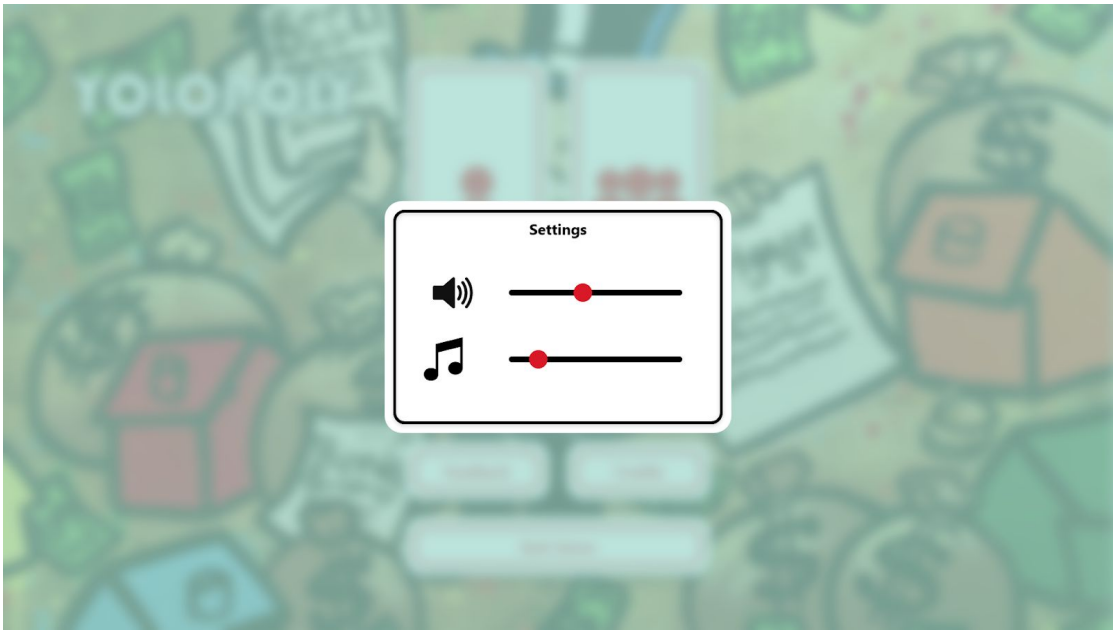


Figure 39: Settings Dialog Box

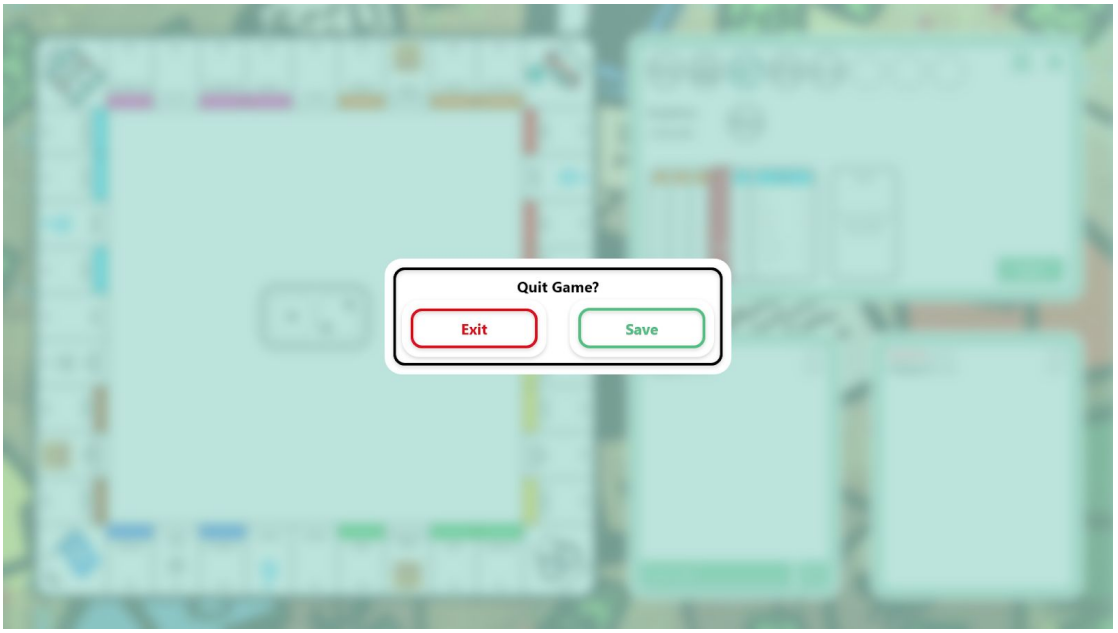


Figure 40: Quit Dialog Box in Single Player Game

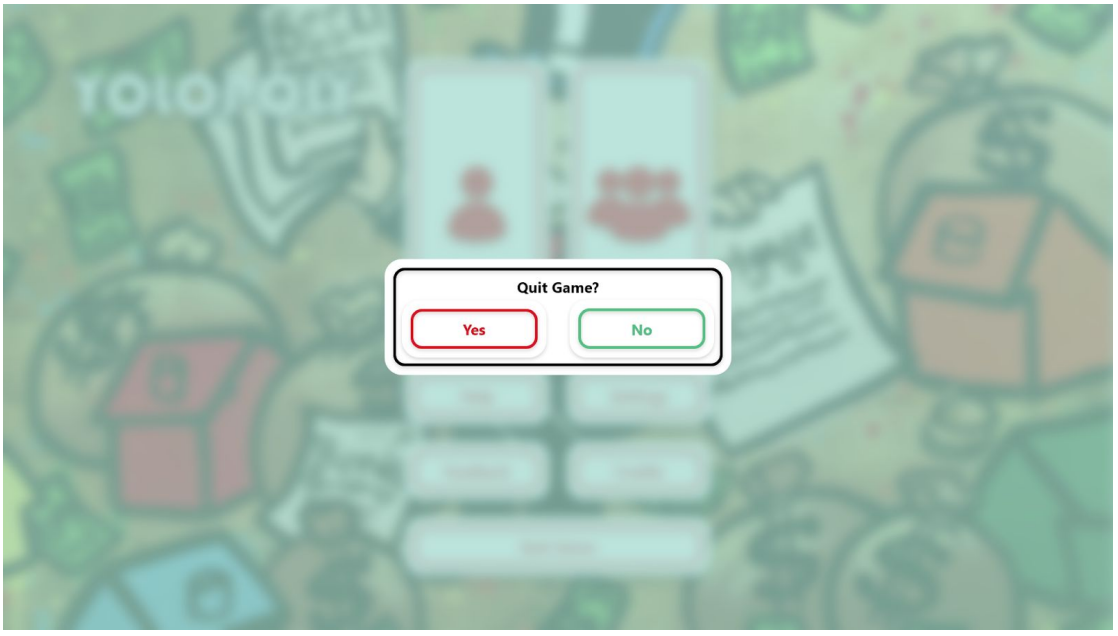


Figure 41: Quit Dialog Box in Multiplayer Game

4.4.4.3 Chat and Log

This section of the screen contains Chat and Log of the game. During the game, players can send messages by using the Chat area. This area contains a textfield to enter the message and a Send button to send the message. In addition to that, players can keep up with the game and see the game history through the Log area. Each move done by the players will be written in that area.

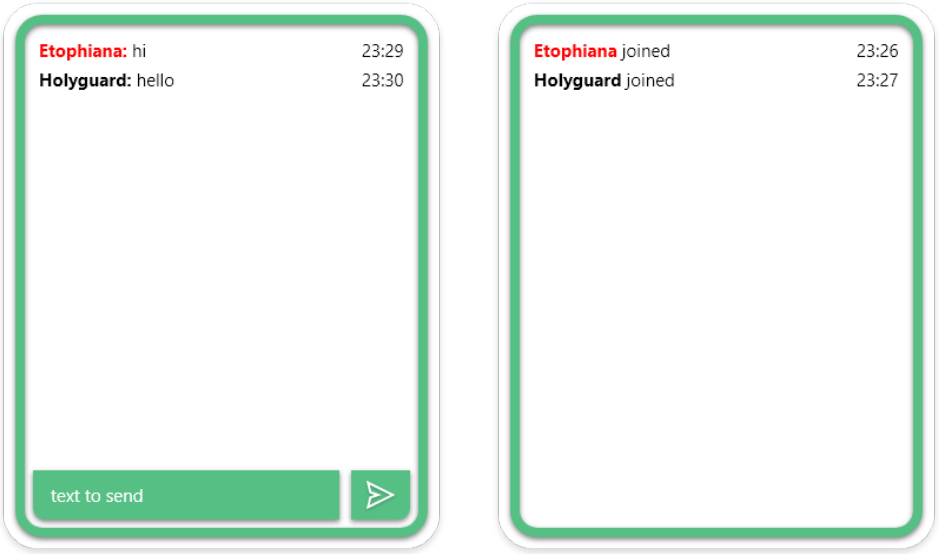


Figure 42: Chat and Log

5 References

[1] https://monopoly.hasbro.com/en-us/product/monopoly-classic-game:7EABAF97-5056-9047-F577-8F4663C79E75

[2] https://www.hasbro.com/common/documents/A0AFE3A69EC745EBA77B9A7950BBCA44/AD7742057B1D43609B53D24D75E9CA9B.pdf

[3] https://www.visual-paradigm.com/guide/