



Object-Oriented Software Engineering Group Project

CS 319 Project: Monopoly Bidding War - Group 1D

Analysis Report

Deniz Kasap 21702460, Irmak Çeliker 21702502, Batuhan Budak 21704212,

Kübra Okumuş 21600980, Ömer Faruk Kayar 21602452

Instructor: Eray Tüzün

Teaching Assistant(s): Emre Sülün, Barış Ardiç and Elgun Jabrayilzade

Progress/Analysis Report Iteration 1

Oct 28, 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

1 Introduction	5
2 Overview	5
2.1 Game Play and Control	5
2.2 Map	6
2.2.1 Go	6
2.2.2 Property	6
2.2.3 Utility	6
2.2.4 Chance or Community Chest	6
2.2.5 Pay Tax	7
2.2.6 Free Parking	7
2.2.7 Go to Jail	7
2.2.8 Jail	7
2.2.9 Railroad	8
2.3 Players	8
2.3.1 Host Mode	8
2.3.2 Join	8
2.3.3 Banker	8
2.4 Items	8
2.4.1 Property Cards	8
2.4.2 Chance or Community Chest Cards	9
2.4.3 Pawns	9
2.4.4 Money	9
2.4.5 Utilities and Railroad Cards	9
2.4.6 Houses and Hotels	9
2.4.7 Dice	10
2.5 Buying Properties	10
2.6 Paying Rent	10
2.6.1 For Properties	11
2.6.2 For Utilities and Railroads	11
2.7 Building Houses or Hotels	11
2.8 Trading	11
2.9 Bankruptcy	11
2.10 Mortgage	12
2.11 Surrender	12
3 Functional Requirements	12
3.1 Controlling the Game	12
3.2 Sounds and Music	12

3.3 How to Play	13
3.4 Credits	13
3.5 Exit Game	13
4 NonFunctional Requirements	13
4.1 In-Game and Server Performance	13
4.2 User-Friendly	13
4.3 User Interface	13
4.4 Extendibility	14
4.4.1 More Players	14
4.4.2 Single Player Mode	14
4.4.3 Customization	14
5 System Models	15
5.1 Use-Case Model	15
Use Case Descriptions	15
Use case Name: Play Turn	15
Use case Name: Start Bid Routine	17
Use case Name: Pay Tax	17
5.2 Dynamic Models	18
5.2.1 Sequence Diagrams	18
5.2.1.1 Trading Scenario	18
5.2.1.2 Join Game Scenario	19
5.2.1.3 Change Settings Scenario	20
5.2.2 Activity Diagrams	21
5.2.2.1 Main Menu Activity Diagram	21
5.2.2.2 Play Turn Activity Diagram	22
5.2.3 State Diagrams	24
5.2.3.1 Building House or Hotel State Diagram	24
5.2.3.2 Player in Jail State Diagram	24
5.3 Object and Class Model	25
5.4 User Interface - Navigational Paths and Screen Mock-ups	28
5.4.1 Main Menu Screen	28
5.4.2 Settings Screen	30
5.4.3 How to Play Screen	30
5.4.4 Credits Screen	31
5.4.5 Game Board	31
5.4.6 Places	32
Glossary & References	49

Analysis Report

CS 319 Project: Monopoly Bid Wars

1 Introduction

Monopoly Bidding War is a turn based multiplayer strategy game. Each player will have a distinct pawn which will have the ability to move around the map by rolling dice. According to the squares they land on, they will purchase assets by auction, collect rent from the assets, draw cards, lose, or gain money throughout the game.

The main goal of the game is to purchase as many assets as possible while building houses or hotels on properties and to collect rent from the other players to bankrupt them. Besides owning assets, players can collect money from the chance or community cards, by landing on free parking or by selling their assets. Also, players may lose money in many different ways along with paying rent. Chance or community cards can make players pay money as well as pay tax squares or landing in jail. There is no limit of rounds, the game ends when a player is bankrupt. The player who is wealthier will be the winner of the game.

2 Overview

2.1 Game Play and Control

The game will be online and will have two different player options: player and a host. According to the line-up, each player will roll the dice on their turn to move on the map that contains 10 different types of squares: Go, Property, Utility, Chance, Community Chest, Pay Tax, Free Parking, Railroad, Go to Jail and Jail. By landing on these squares, players will be able to buy properties or utilities and keep on gaining or losing money throughout the game depending on the square they landed. The main goal is to lose a minimum amount of money while owning the maximum amount of properties to gain money and bankrupt all the other players. When there is only one player remaining who has not bankrupted or surrendered, he or she will be the winner of the game.

The main control is maintained by the mouse and the keyboard. Besides entering an amount for a property, the keyboard will not be used. Every other operation will be done by clicking.

2.2 Map

The map will be 11 x 11, containing 10 different types of squares: Go, Property, Utility, Chance, Community Chest, Pay Tax, Free Parking, Railroad, Go to Jail and Jail on the sides. In the middle, Chance and Community Chest cards will be settled, along with a space to pay tax and jail money on.

2.2.1 Go

Go square is the starting point of the game. After the first round, every player who passes or lands on this square will be rewarded with \$200. Passing the Go square means starting a new round for the players.

2.2.2 Property

Properties are meant to be bought and collected by the players. There will be 22 Property squares consisting of 8 different colors and initial prices, the value and the rent of each property will become higher respectively from the start to the end.

If the player lands on an unowned property, an auction screen will pop up automatically, so that every player will have a chance to purchase that property.

If the player lands on an owned property, he or she will pay rent to the owner.

If the player lands on to his or her own property, he or she will have a chance to build a house or to do nothing.

2.2.3 Utility

There will be 2 Utility squares, same as properties they are meant to be owned and the price also increases from start to end. The same rules apply for landing on a Utility; however, players cannot build houses on Utility squares.

2.2.4 Chance or Community Chest

There will be 3 Chance and 3 Community Chest squares. When the player lands on one of these squares, by clicking on the pile found in the middle of the map they will be able to draw a card. According to the content of the drawn card, the player may lose or gain money, be sent to jail, or have a chance to get out from jail without paying any money.

2.2.5 Pay Tax

There will be 2 Pay Tax squares, these squares will show the amount of money to be paid. The person who landed on any of these squares will have to put the specified amount of money on the middle of the map. If the player does not have enough money, he/she will be declared as bankrupt.

2.2.6 Free Parking

If a player lands on Free Parking square, he or she will collect the money found in the middle.

2.2.7 Go to Jail

Go to Jail square is located on the corner of the map and sends the players to Jail where it is located on the cross corner of the square.

2.2.8 Jail

When a player lands on the jail by rolling dice, they are not counted as in jail and keep on playing on the next round.

There are three ways to end up in jail: by rolling doubles successively for 3 rounds, by landing on Go to Jail square or by drawing the Go to Jail Chance or Community Chest card. People who land on this square will also have three options to get out of jail.

After ending up in this square, the player can try rolling doubles on the next round to get out of jail, if it is a success they can keep on playing, if it is a failure they keep on waiting. However, a person can only try rolling doubles for 3 rounds, if it is still a failure in the third round, they need to pay \$50 to the middle and can start playing in the next round.

As a second option they can pay \$50 to the middle any of the next 3 rounds and keep on playing on that round without trying to roll doubles. If the player does not have enough money, he/she will be declared as bankrupt.

The last option is to use the Get Out of Jail Free card. The players can own this card by drawing on the previous rounds or by buying or trading from another player who owns this card. The player can return this card and keep on playing on that exact round.

2.2.9 Railroad

Railroad is also an asset which can be purchased by the players, the rent collecting will be the same as utility. There will be 4 of them and players will not be able to build houses on these assets.

2.3 Players

All the players will join the game online, there will be no computer players actively playing the game. The number of players will be 4.

2.3.1 Host Mode

If a person chooses this mode, they will start a new game. Hosts will also be able to share an id so that the desired players will be able to join.

2.3.2 Join

If the player has a shared id, he/she can use the id to join a specific host's game.

2.3.3 Banker

The banker will not be from one of the players, this role will be played by the computer.

2.4 Items

2.4.1 Property Cards

Property cards will display the rent, house or hotel building and the mortgage value of the particular property. The rents will change according to the number of houses or hotels on that property and it will be displayed by these cards.

Besides the owner of the property, other people will not be able to see the content of the card. However, every player can see the property cards owned by the other players to be able to send trade or buy requests.

2.4.2 Chance or Community Chest Cards

They will be located at the center of the map. People who land on Chance or Community Chest squares will click here to draw a card. Every card will have a different content, such as get out of the jail free, go to jail, pay tax, collect money etc.

2.4.3 Pawns

There will be 6 different pawns. Each player needs to choose a pawn that has not been chosen by another player. Pawns will move across the map according to the output of the dice, so that players will be able to follow where they are located.

2.4.4 Money

Each player will start the game with \$1500. People will be able to track each other's money amount. Collecting rent and purchasing will be done by it.

2.4.5 Utilities and Railroad Cards

Utilities cards will carry almost the same characteristics with property cards. There will be 2 different utilities and 4 different railroad cards, each of the cards will display the rent and the mortgage value. However different from the property cards, the more utility and railroad cards a player collects, the higher rent they will receive since there is no build house option.

Same as property cards, people will be able to see the belongings of other players without seeing the content.

2.4.6 Houses and Hotels

Houses and Hotels will be built by the players to increase the rent of a property. There will be 32 houses and 12 hotels, if all these houses and hotels have been purchased, people will not be able to buy any other.

They will be displayed on top of the property squares, so that every player will be able to follow the game.

2.4.7 Dice

There will be two dice and players will be able to move on the map by rolling them. If it is the player's turn, when the player clicks on the roll dice button, the pawn will move according to the random output given by the dice.

2.5 Buying Properties

The main method for purchasing a property will be made by performing an auction. When a player will land on an unowned property or utility, the auction will start automatically if the player does not choose to veto. A screen for bidding will be displayed so that players will be able to enter their offers. Players will not see each other's bids and after the auction is concluded, the player who has landed on the square initially will be notified if he/she won or lost the auction. A second chance will be given to the player to make another bid. Also, if there are two or more players who have entered the same amount of money and one of them is the player who landed on the square, he/she will be the automatic winner. Else, the players remaining will roll the dice and the player who rolled the highest amount will be the winner. After the auction is concluded, the winner of it will be displayed on every players' screen and the winner will be able to purchase the property by paying the last bid he/she has entered.

Veto Right: Each player will be given 2 veto rights throughout the game. When the players land on an unowned square, the given veto right will let them own that asset automatically without an auction. If the player chooses to use his/hers veto right, he/she will pay the required amount and will be the owner of the asset. A player cannot use his/hers veto right twice in a single round, they will be able to use the remaining one in the next rounds.

2.6 Paying Rent

Players need to pay rent if they have landed on an owned square. If the player who will pay rent has not enough money, he/she cannot send a trade request or become a debtor, they will be declared as bankrupt.

2.6.1 For Properties

Properties' rents depend on the number of the houses. As the house number increases, the rent also increases.

2.6.2 For Utilities and Railroads

The rent of these are affected by two factors: the output of the dice and the number of utility or railroad cards the owner has. Five times the number of utility or railroad cards the owner has and the output of the dice will be multiplied in order to calculate the rent. For example, if the player rolled 9 and the owner has 2 utility cards, the rent will be $2 \times 5 \times 9$, which equals \$90. If the player does not have enough money, he/she will be declared as bankrupt.

2.7 Building Houses or Hotels

As mentioned before, there will be 32 houses and 12 hotels, and when all of them are being used in the game people will not be able to build houses anymore. There are three conditions to build a house on a property: a player needs to be located on that square, has to build only a house at one time and he/she needs to own all the properties with the same color.

After building 4 houses the player will be able to build a hotel on the next rounds, and the other 4 houses are returned to the bank. The maximum number for building a house on a single property is also 4.

2.8 Trading

Players will have the chance to send each other requests to trade or buy the belongings of the other players if it is their turn. The player can offer money or his/her belongings in return of the desired properties or cards. After that the other player will receive the offer in his/her screen and can either accept or deny the request, however if the other player wants to negotiate, he/she can wait for his/her turn and make a new offer to the player.

2.9 Bankruptcy

When a player loses all his/her money and owned properties, or has all his/her belongings on mortgage with no money or are not able to make payment, the player will be declared as bankrupt. A

bankrupted player will not be able to play and loses the game, however he or she can keep on watching it.

2.10 Mortgage

If a player does not want to lose his/her property or utility and needs money, he/she can put his/her properties on the mortgage. The player will receive the specified mortgage amount of the property by the bank; however, he/she will not be able to collect rent until the property is unmortgaged.

If the player decides to unmortgage the property, he/she will need to pay the mortgage amount plus %10 interest.

2.11 Surrender

When the players want to quit playing, they can choose the surrender option and stop playing. After a player surrenders, his/her belongings will be returned to the bank, so the properties or utilities he/she has will be opened to purchase.

2.12 Winning

If any of the players is declared as bankrupt, the game will be over and the winner will be decided according to their wealth.

3 Functional Requirements

3.1 Controlling the Game

The game will start with the map. Players will be able to control the game by the mouse and the keyboard. All of the operations in the game require clicking except the operation of entering money. However, the keyboard will be used in order to enter an amount of money in the auction or while sending trades.

3.2 Sounds and Music

Settings option will include Master Volume, BGM Volume, SE Volume. Players will be able to adjust the volume of the game. BGM stands for background music, SE stands for sound effects and Master controls the whole volume of the system.

3.3 How to Play

The players who would like to learn how to play or to check a rule can choose this option. The detailed description and the rules of the game will be available for the user by this option.

3.4 Credits

By this choice, players will be able to see the creators of the game.

3.5 Exit Game

If the player decides not to play, he/she can exit the game. After choosing this option, the game will shut down and the player will be returned to the desktop.

4 NonFunctional Requirements

4.1 In-Game and Server Performance

We are using a database to interact and exchange data between users. Data flow between database and computers are really constricted, without any loss of information. Some parts of the game works only on the host's computer, and it distributes the crucial and necessary information through the database. Joined computers take the information and display them accordingly. This feature makes the game faster and fluent.

4.2 User-Friendly

The objects and details of the game have smooth graphic designs. We are planning to please the eyes of users as much as possible with soft drawings and chirpy colors. We have some options in order to maintain the control of the user as giving them more flexibility. Besides, users will not be trapped in huge amounts of buttons and control units. We will sustain the simplicity so as not to cause confusion.

4.3 User Interface

We adopt the thought that we need to protect the traditional monopoly structure. However, we tried to expand it with some animations and ideas. In game, cities (squares of the board) are demonstrated as 3D objects. Size of a pawn is 40x40 and the size of a dice is 50x50 pixels. Game

window will have 1920x1080 resolution as default. In a turn, when a card is played, as a chance card or as a property, it is displayed fully then the game goes on. Players can trade their cards and money through a trade window. There are some compatible ideas like these in order to give the best experience to the players.

4.4 Extendibility

4.4.1 More Players

In the first release, we will have multiplayer mode that can be played only with shared id. We are planning to add a platform where the player can join some public servers, and meet strangers online to enjoy the game. Having strangers getting together will require communication through the game, so there can be a chat and a voice chat too.

4.4.2 Single Player Mode

With the help of artificial intelligence, we are planning to add a single player mode which can be played with bots. These bots will have at least three levels of difficulties. Users can develop their skills without any frustration due to skill level difference.

4.4.3 Customization

We are planning to add more tables and carry this adventure to another level. Players will enjoy playing in their nation's cities, also they will be able to select their national animals and traditional objects as pawns. We are planning to customize the dice as well.

5 System Models

5.1 Use-Case Model

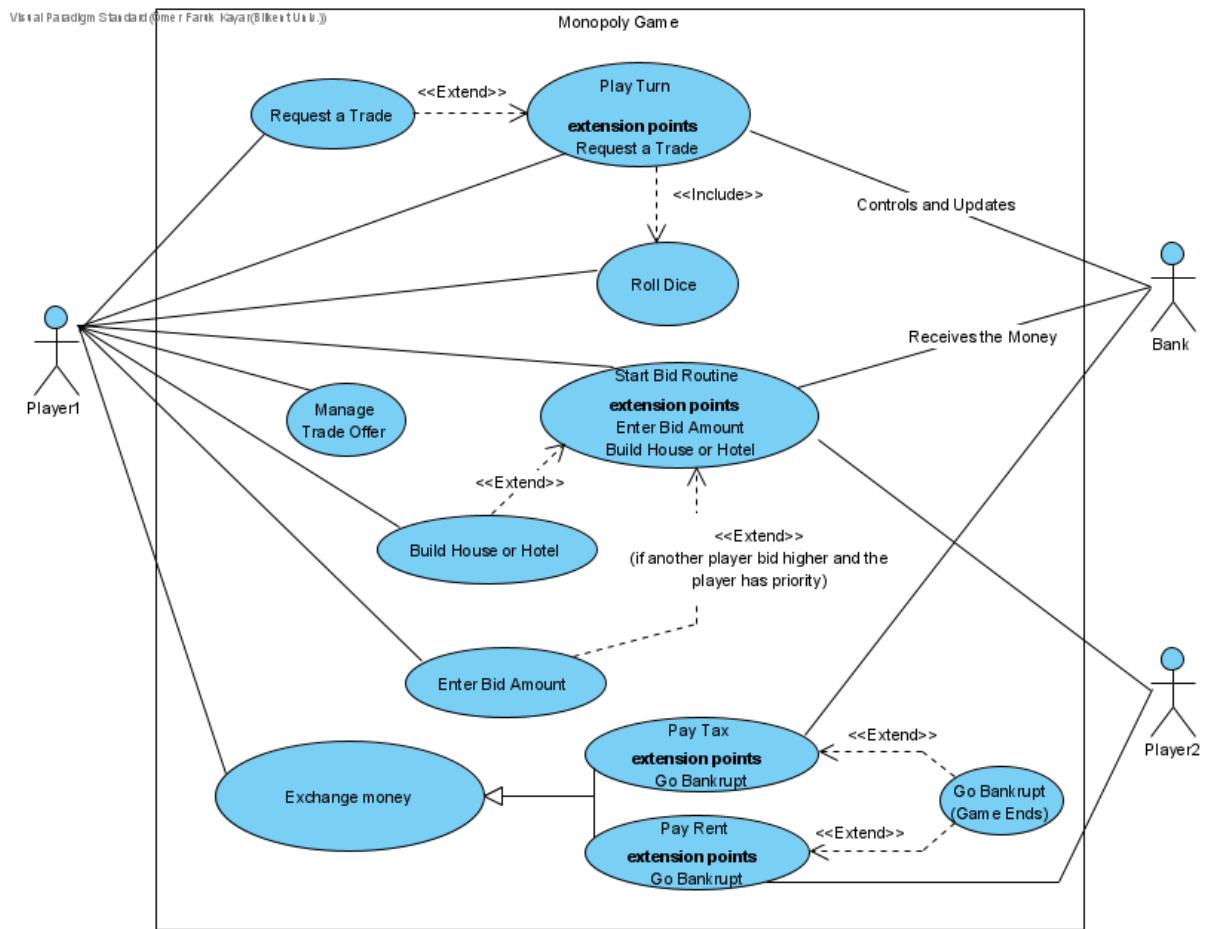


Figure 1: Use-Case Diagram

Use Case Descriptions

Use case Name: Play Turn

Participating actor: Player1, Bank

Entry condition:

- Bank decides whose turn starts and if it is the player's

Exit condition:

- Player clicks the end turn button and the game proceeds.
- Player opens the pause menu and clicks the exit game button.

Main Flow of events:

1. Player1 starts the game or waits for the previous player to end his/her turn.
2. Bank gives the player the permission to play their turn.
3. Player1 rolls dice.
4. His/Her pawn proceeds to the concluded amount of blocks further and lands on a square.

Based on what the square is, the next event will be decided.

5. The pawn lands on an unowned property.
6. The Player1 doesn't buy the property.
7. Start Bid Routine starts.

Alternative flow of events:

5. a. The pawn lands on Player2's property.

1. Player1 exchanges money with the owner(Player2).
2. Player1 clicks the end turn button.

5. b. The pawn lands on a Player1's property.

1. Player1 may Build House or Hotel.
2. Player1 clicks the end turn button.

6. a. If Player1 has a buy right, Player1 buys the property.

1. Player1 clicks the end turn button.

6. b. If Player1 has a buy right, Player1 buys the property. Player1 offers Player2 a trade.

1. Player1 specifies what he/she wants and what he/she offers for those.
2. Player2 manages the trade offer by accepting or declining the offer.
3. If Player2 accepts, the properties will be traded. If Player2 declines the offer, no change will occur.
4. Player1 clicks the end turn button.

Use case Name: Start Bid Routine

Participating actor: Player1, Bank, Player2

Entry condition:

- Player1 lands on an unowned property and doesn't buy the unowned property.

Exit condition:

- Player1 and Player2 enter bid amounts.

Main Flow of events:

1. All players enter bid amounts.
2. Bank informs Player1 if he/she won or lost the auction.
3. Player1 won the auction.
4. Player1 gets the property.
5. Game proceeds.

Alternative flow of events:

3. a. Player1 lost the auction.
 1. Bank will ask Player1 to enter a second bid.
 2. If the second bid of the Player1 is the highest, Player1 gets the property. If the second bid of the Player1 is not higher, the first auction winner gets the property.
 3. Game proceeds.

Use case Name: Pay Tax

Participating actor: Player1, Bank

Entry condition:

- Player1's pawn must land on a pay tax square

Exit condition:

- Player1 clicks the end turn button.

Main flow of events:

1. Bank receives the certain payment.
2. Bank puts the money in the middle.
3. Player1 clicks the end turn button.

5.2 Dynamic Models

5.2.1 Sequence Diagrams

5.2.1.1 Trading Scenario

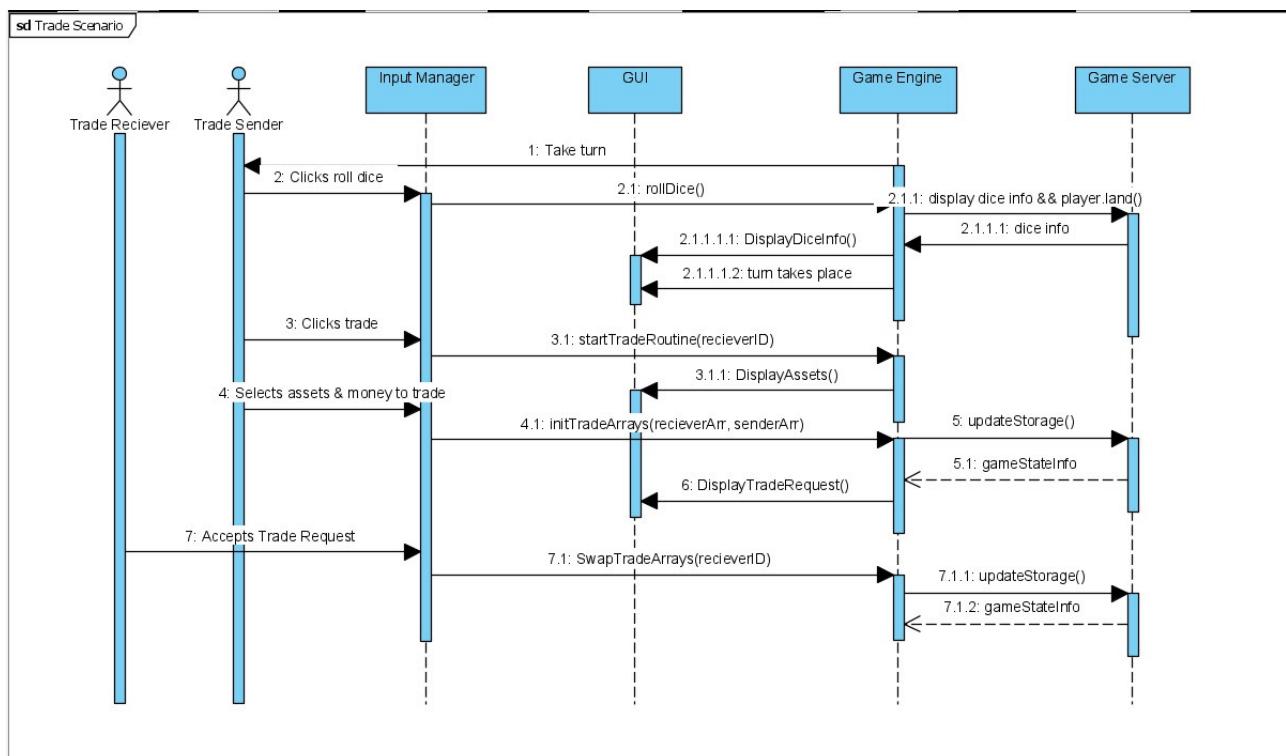


Figure 2: Sequence Diagram for Trading Scenario

In this scenario, the player is currently playing a game and it's her/his turn. Player rolls dice and the location which s/he lands is displayed on the screen. Player wants to trade so s/he clicks the corresponding trade button. The Player's assets and the other player's assets are displayed on the screen in order for him/her to choose. Other player's screen will display the request. After the request response is given, according to the answer the assets will be updated through the game server. New game board information will be distributed to all players.

5.2.1.2 Join Game Scenario

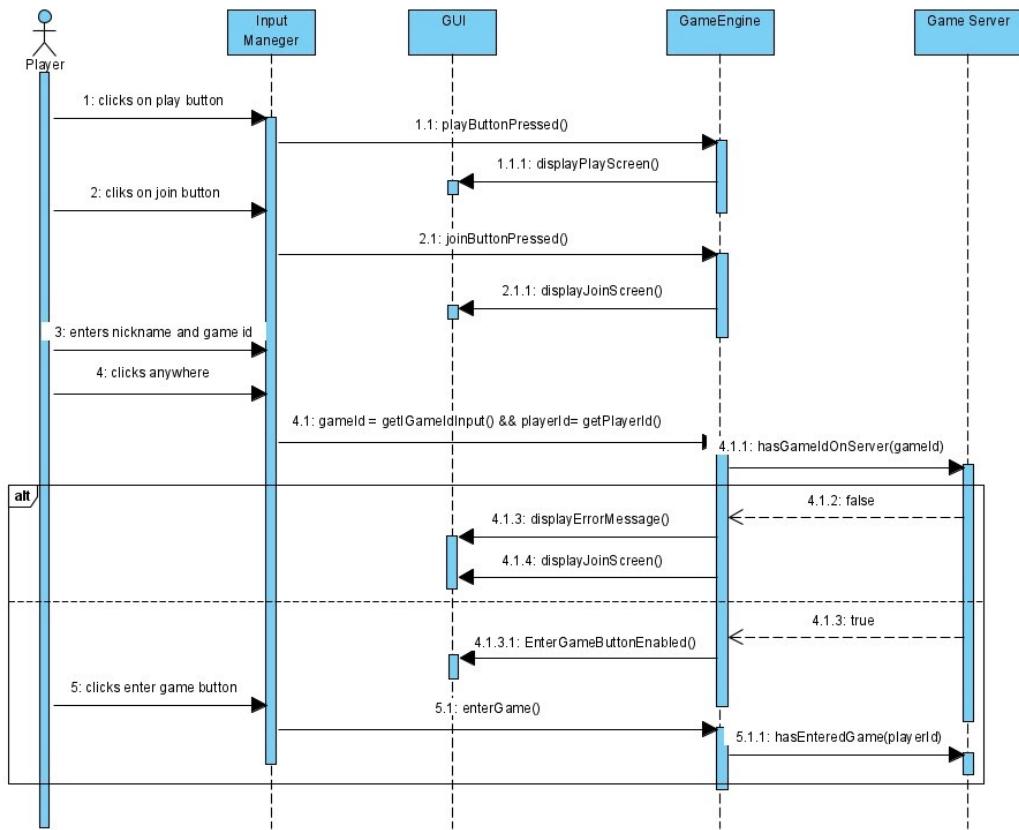


Figure 3: Sequence Diagram for Join Game Scenario

In this scenario, the player wants to join a game which is opened by another player. The user enters a nickname(player id) and a game id on the join screen. S/he must enter the correct game id which is chosen by the host player. When the player presses anywhere on the screen after entering data, the game id is searched on the server. If there is no game with that id, first an error message is displayed and an empty join screen opens. Otherwise, the "Enter Game" button is enabled and the player enters the game.

5.2.1.3 Change Settings Scenario

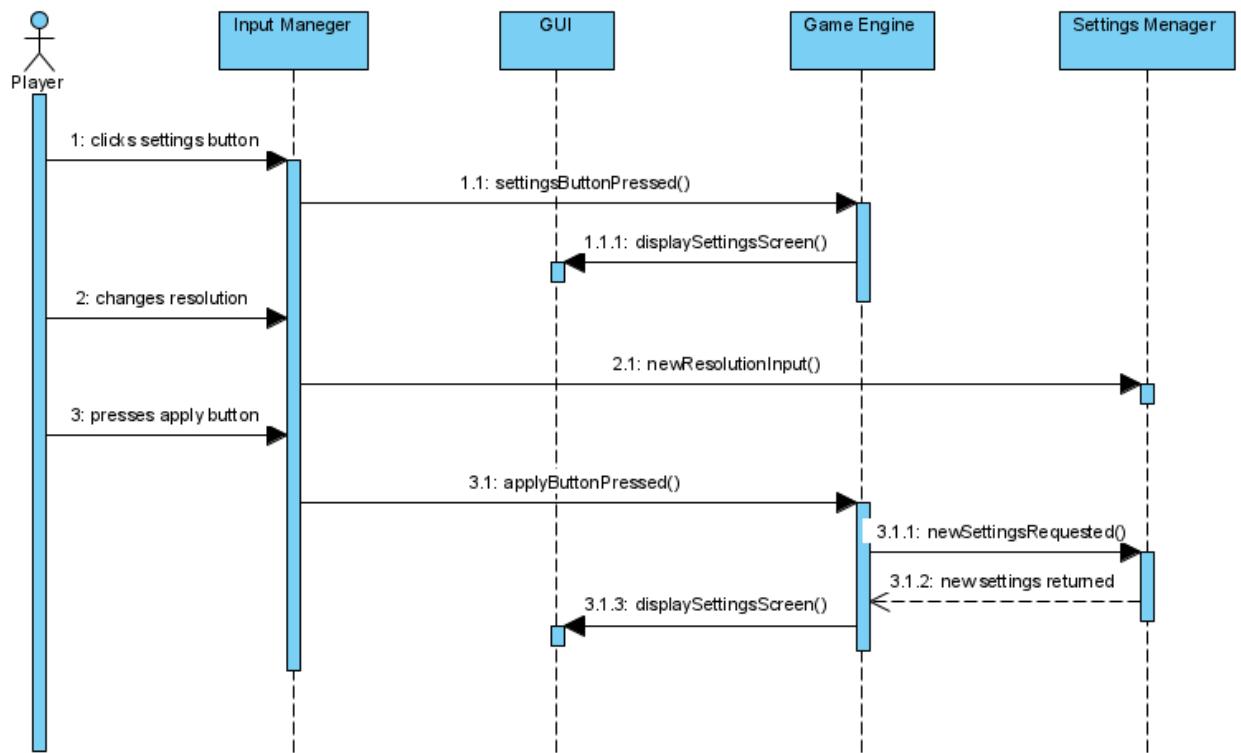


Figure 4: Sequence Diagram for Change Settings Scenario

In this scenario, the player wants to change the resolution of the game window. First, the player clicks on the settings button. Game Engine detects it and executes display command. GUI gives the user the corresponding interface. Player changes the resolution, the settings manager catches the change. When the player clicks the apply button, Game Engine detects it and it requests new settings properties from the settings manager. Settings manager returns those properties and finally, Game Engine sends settings to GUI to display all again.

5.2.2 Activity Diagrams

5.2.2.1 Main Menu Activity Diagram

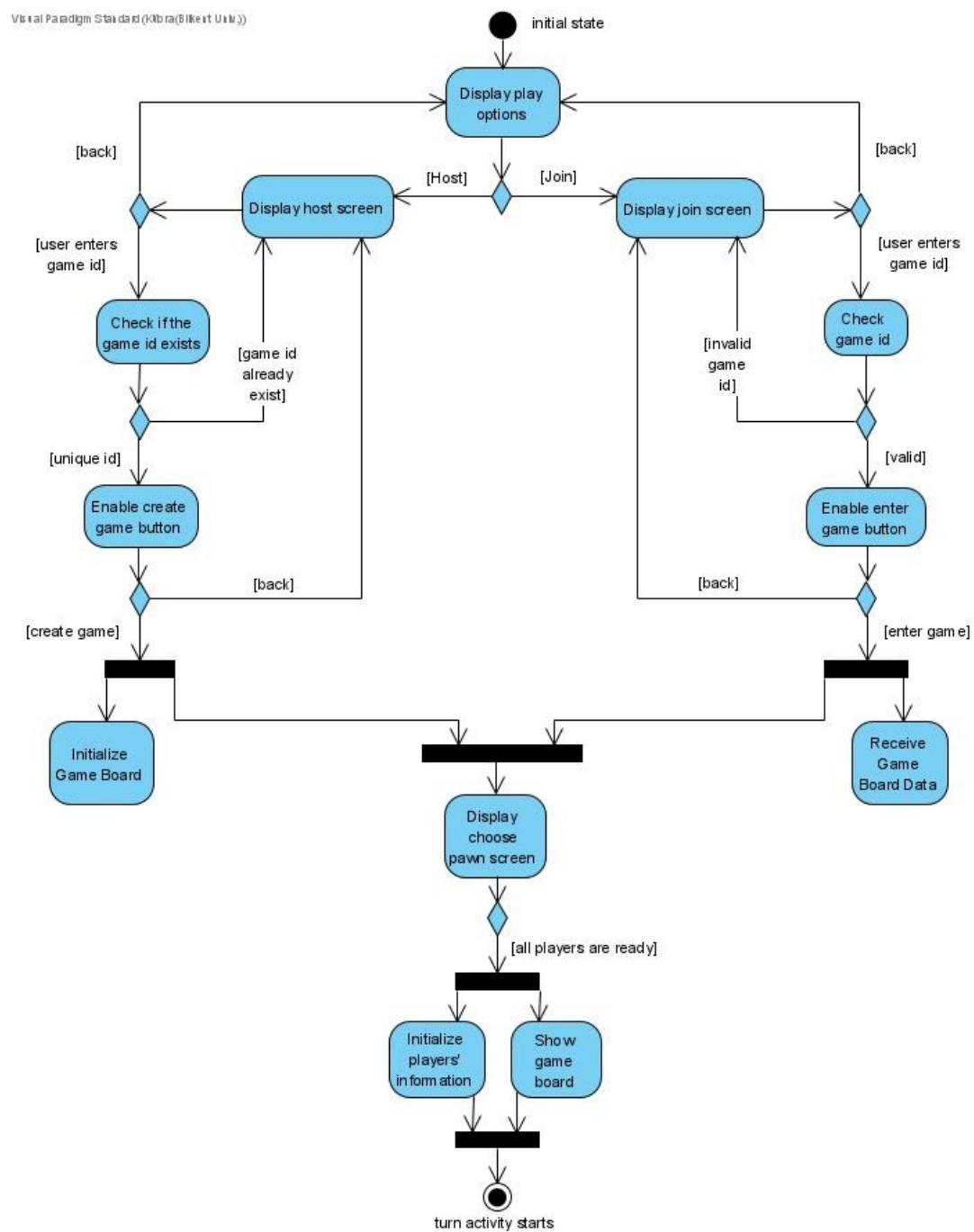


Figure 5: Activity Diagram for Join and Host Game Screens

This activity diagram shows how the system maintains the pre-conditions of the game.

Simply, it starts when the player clicks on the play game button at the main menu. System displays the play options. Different scenarios occur according to the chosen button.

Case 1: If the player selects the join button, he/she will proceed to joining a game with a certain game id. Then the game board data will be taken through the database from the host. After the data transformation, the system will make the same operations in Case 2.

Case 2: If the player selects the host button, he/she will proceed to creating a game with a unique id. Then the game board will be initialized. After the initialization operation of the game board, the system will display the pawn screen where every player has to choose a different pawn from each other. The activity diagram ends with the initialization of the players' information and display of the game board. After this activity, turn activity starts.

5.2.2.2 Play Turn Activity Diagram

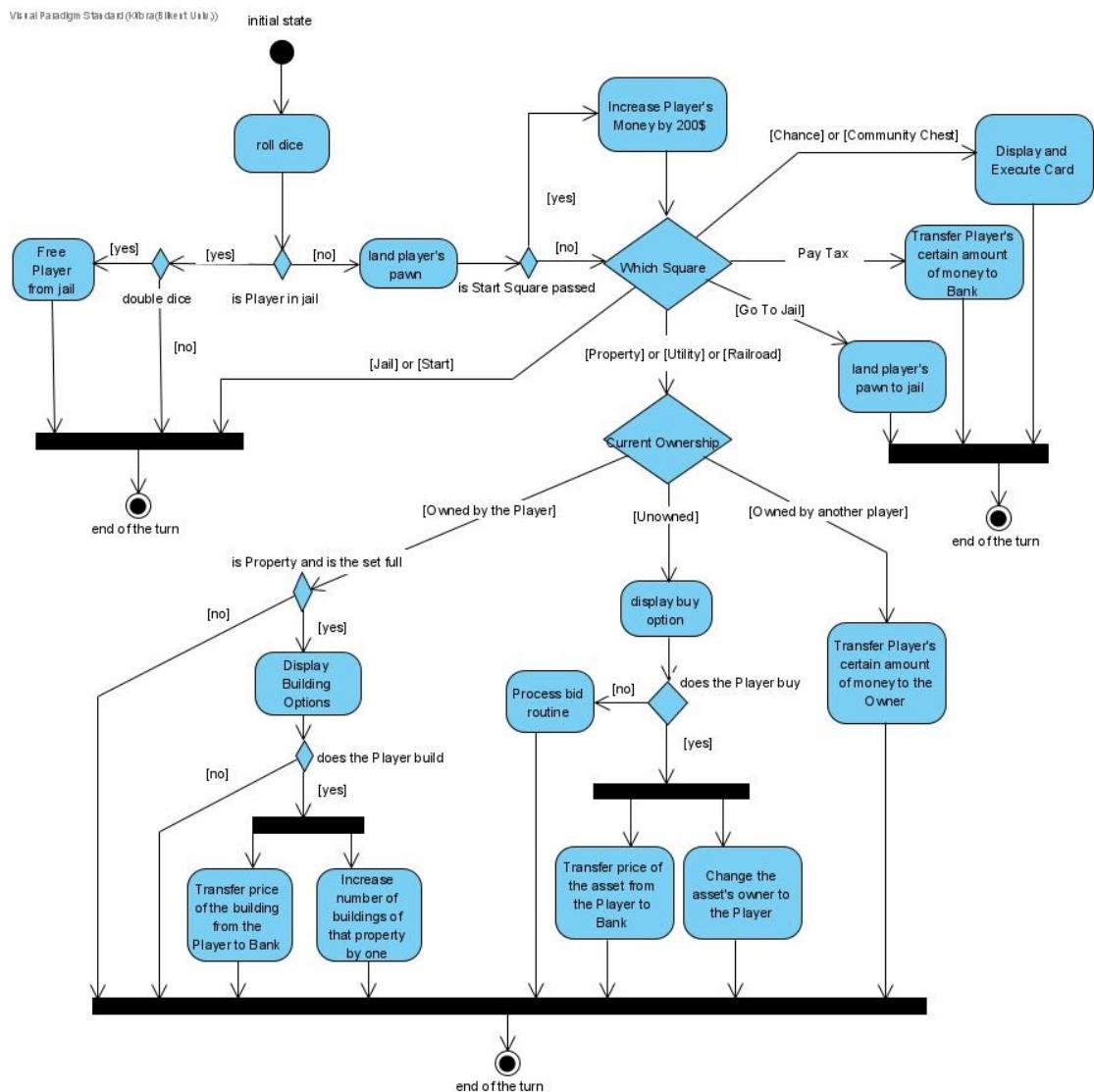


Figure 6: Activity Diagram for Playing a Turn

This activity diagram shows how the system works in a turn. It starts with rolling the dice. If the player is currently in jail, the system checks whether the dice are double. If it is, then the player is set free from the jail and can wait his/her next turn to play. If the dice is not doubled, the turn is over.

In the case of the player not in the jail, his/her pawn is landed according to the result of the dice. Then, the system checks if the Start square is passed. \$200 is given to the player if the condition is satisfied.

After then, different activity flows occur according to the square which the player is on.

Case 1: Player lands on a Chance or a Community Chest square

The card is displayed to the player, and the necessary operations are done according to the content of the card.

Case 2: Player lands on the Pay Tax square

The specific amount of money of the player is transferred to the bank.

Case 3: Player lands on the Go to Jail square

Player's pawn is landed on the jail and the turn is over.

Case 4: Player lands on the Jail Square or the Start Square

In both possibilities, nothing happens and the turn is over.

Case 5: Player lands on a property, an utility or a railroad

In this case, the current ownership of the asset is checked.

1. If it is owned by another player the corresponding money is transferred from the player's account to the owner's account.
2. If the asset is unowned and the player has the right to veto the bid routine, the buy option is displayed. If the player chooses to buy that asset, the money transfer takes place, and the player is declared as the owner of that asset. In case the player does not buy the asset, the bid routine is processed.
3. If the asset is owned by the player, and the certain requirements are met, the system enables the built option. When the player clicks that button, the number of buildings of that property is increased and the price is transferred from the player's account to the bank.

5.2.3 State Diagrams

5.2.3.1 Building House or Hotel State Diagram

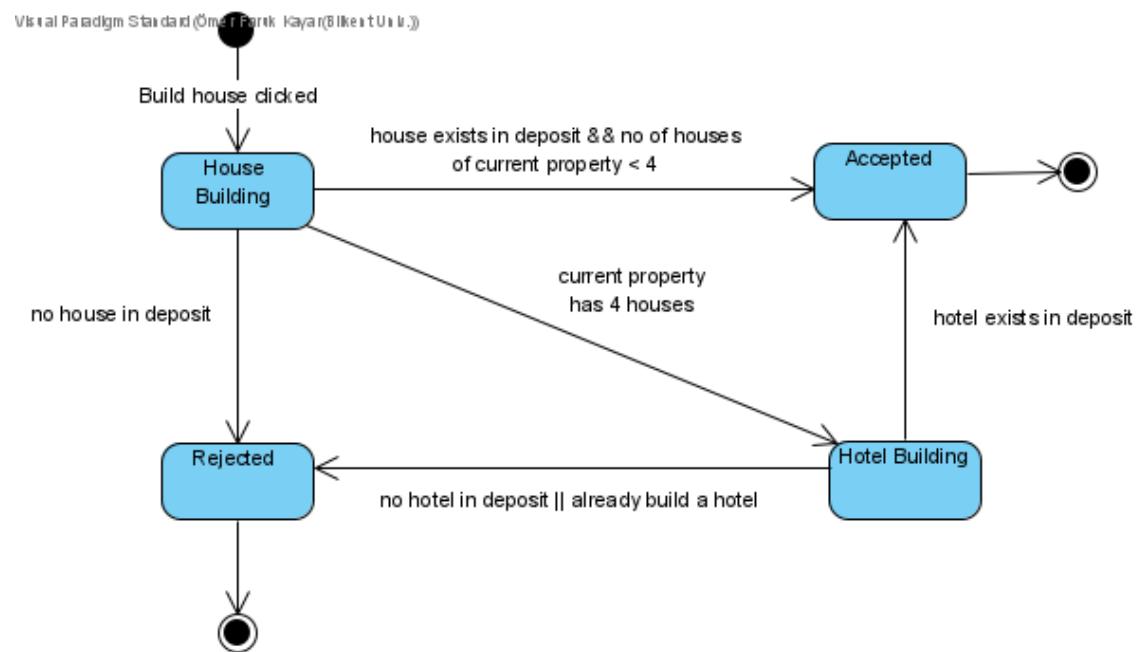


Figure 7: State diagram of building house or hotel

When the player clicks the built house button, House Building state starts. If there are unused houses in the game and that property has less than 4 houses, the next state will be Accepted state and a house is built on that property. If there are no houses to build, the state is changed to the Rejected state. If there are 4 houses in that property, the House Building state will change to Hotel Building state. If there are unused hotels in the game, a hotel is built. If there is no hotel to build or that property already has a hotel the request is rejected.

5.2.3.2 Player in Jail State Diagram

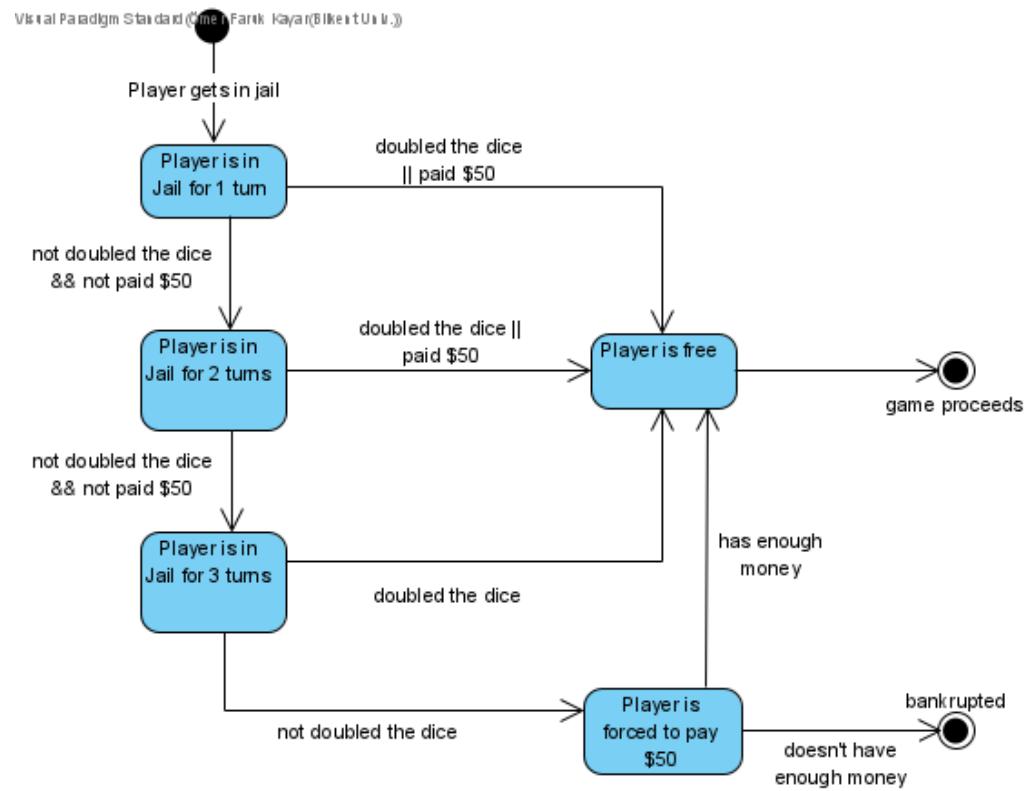


Figure 8: State diagram of player in jail

When the player ends up in jail, he can double the dice or pay the price to get free. If the player could not manage to do either of these, he/she will be in jail for a second time until his next turn comes. Then the player will try doubling the dice or he/she can choose to pay the price. If not, the player will stay in jail until his next turn. If he/she cannot double the dice in this turn, he/she will be forced to pay the price. Not having enough money will make the player go bankrupt. Otherwise, the player will be set free.

5.3 Object and Class Model

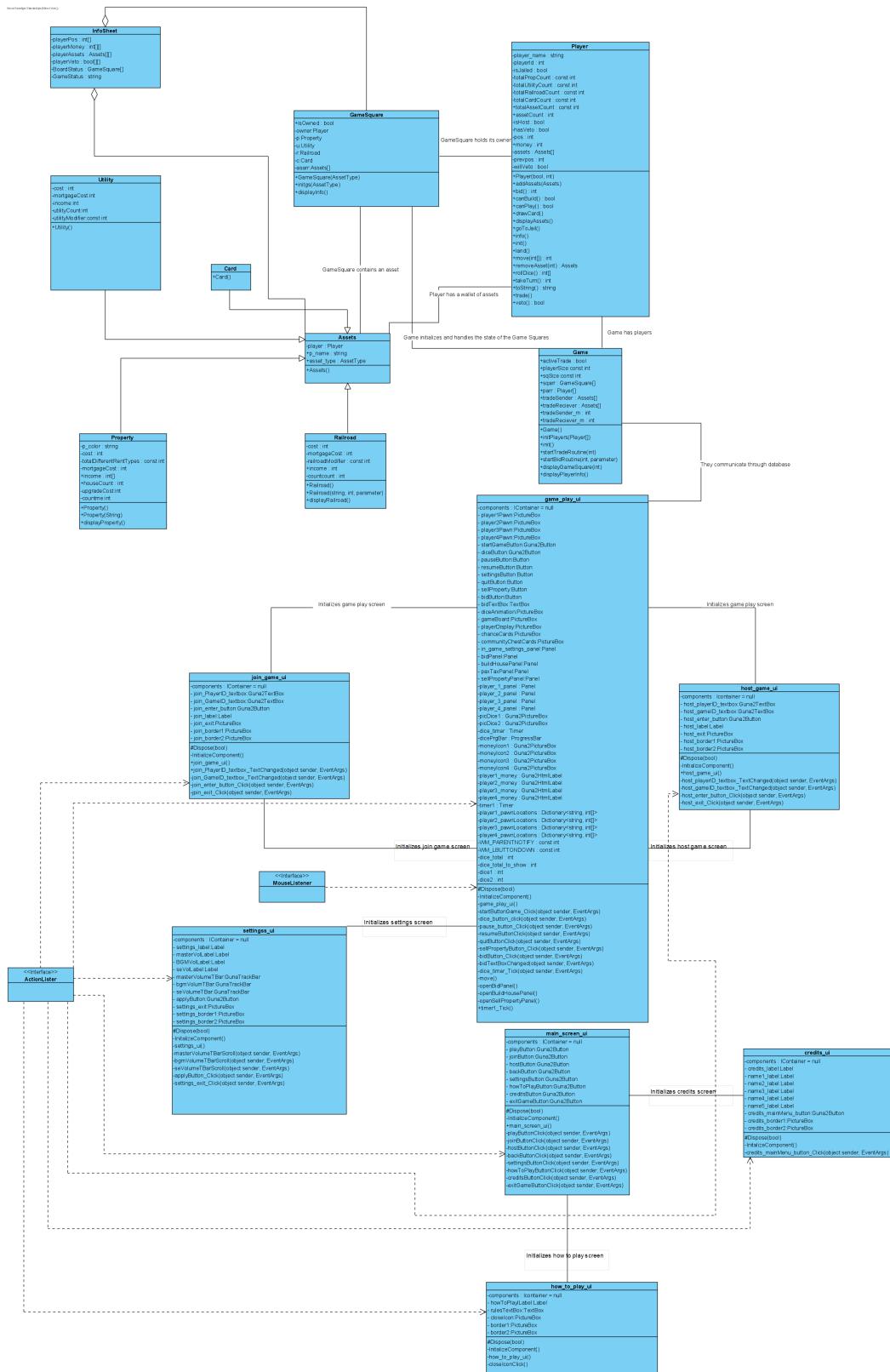


Figure 9: Object and Class Diagram

The class diagram of our game "Monopoly Bidding War" currently has 16 classes. The game starts at the opening screen where players will have options about what they want to do, these visual aspects and their effects on the game will be mainly managed by the GUI classes, the main driver of the game is actually the class "Game". This class takes a lot of information from the other object classes such as "Assets", "GameSquare" and "Player".

Game Class

This class is responsible for all the main components of the game, even though each object is generally responsible for their own actions, properties or changes, this class handles concepts or actions regarding more than one player or object. This class initializes the game, cards, players and assets, handles trading, bidding, buying or selling properties etc.. This class is also responsible for the games communication with both the database and gui. Most of the calculations will be done on the hosts computer, will be sent to the database via this class, and then also will be received by this class on other players computers and the regarding received information will be distributed to the object classes accordingly.

GameSquare Class

This class represents an individual game square on the board, the squares can be a property, utility, railroad, start square or a card square. An array located at the game class will hold 40 instances of this class. This class will be responsible for being the middleman between the GUI, game class , player class and other objects, since this is the one key aspect of the game where all of the pieces come together. Displaying a square will be done through this class. For example if a player wants to buy a property or wants to collect rent from another player, those actions will be done through this class.

InfoSheet Class

This class keeps necessary information of the players such as their position, money, remaining veto rights etc.

Assets Class and It's Children

These classes are here to represent every interactable / tradable object of the game, the assets class is the parent, and other classes that will be listed shortly are its children. They will inherit its methods, but some will be not inherited but individually declared in most of them, since all of these classes are very similar but sometimes act differently, which could be dangerous and buggy if not executed carefully. The children of the assets class are; property, utility, railroad and card. All of these classes have names, and except from the card class, they all have rents, costs, mortgage values. The property class can have houses and hotels built on them, so even though they are all very similar and all are technically assets, some key distinctions are present between them. Thus, inherited or individually declared methods should be planned and written with care.

UI Classes

main_screen_ui Class

This is the first screen the player is prompted with, here the player can host or join a game, it can also view credits, and adjust the settings. The player can also go back to the original start screen from the play game and settings screens via a back button. If the player clicks on play a game, he will be prompted with three options, join a game, host a game, or go back. If s/he presses back, s/he will be taken back to the original starting screen, otherwise s/he will either go to the host_game_ui or join_game_ui class.

join_game_ui Class

This is a page containing two text boxes and an enter button, the player can enter his/her id, and the game id he/she wants to join.

host_game_ui Class

This page is identical to the join_game_ui page, only difference being instead of entering an id for the game, the player will set the id for others to join.

settings_ui Class

Here the player can adjust the sound level of the overall voice of the game, the sound effects voice level, or the music of the game. It can also change the resolution of the game.

how_to_play_ui Class

This is a simple page with a text field containing instructions, rules and tips & tricks for the game. This page is for new players, it will explain the basic concepts of monopoly, also the rules relating to our version of the game. Since this is a new concept, new players can have questions relating to the rules, and they can view this window to answer those questions.

credits_ui Class

This class also contains a single text field which has credits written in it.

game_play_ui Class

This is the main GUI class, this is what the players will see most of the time, this is where the whole game takes place. All of the information sent to the database from the Game class will be displayed here. This class will contain the map, the game squares, the icons for the players, their money & assets, the chance cards, community chest cards, money collected for taxes, dice animations, bidding screens and so on. The player can open the settings window from here, it can concede, or quit the game.

5.4 User Interface - Navigational Paths and Screen Mock-ups

5.4.1 Main Menu Screen

There are five buttons for the player to select what to do.



Figure 10: Main menu screen

If the player presses the "play" button "settings", "how to play", "credits", "exit game" buttons disappear, and the "play" button becomes disabled. Also "join", "host", "back" buttons appear.



Figure 11: Main menu screen after play button pressed

If the player presses the “exit game” button a popup screen comes up to verify.



Figure 12: Main menu screen after exit button pressed

5.4.2 Settings Screen

In the settings screen there are four different options to adjust which are resolution, master volume, BGM volume, and SE volume. Also, a button to apply changes and a close button.

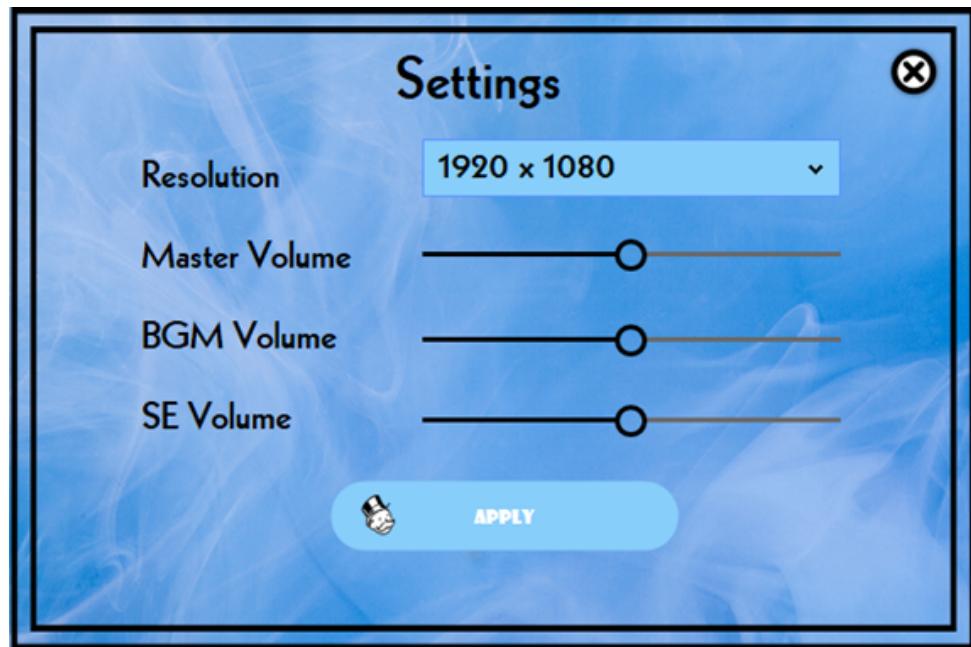


Figure 13: Settings screen

5.4.3 How to Play Screen

How to play screen contains game rules.

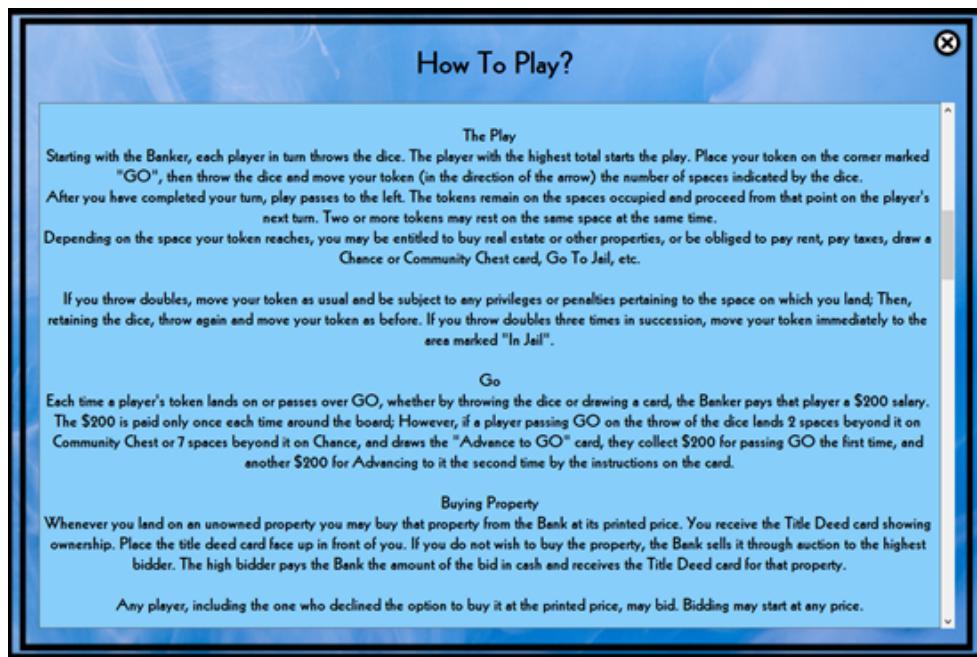


Figure 14: How to Play Screen

5.4.4 Credits Screen



Figure 15: Credits Screen

5.4.5 Game Board

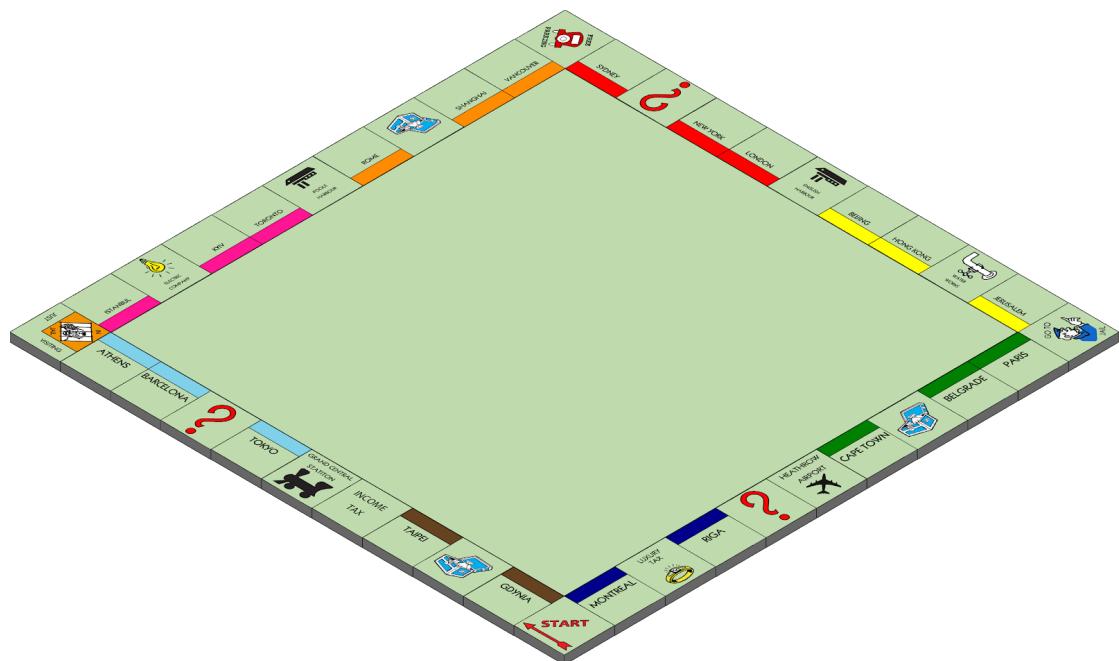
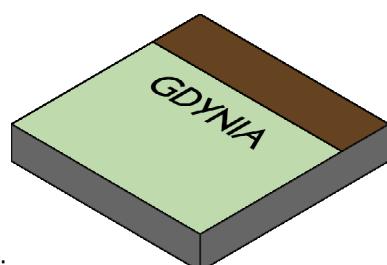


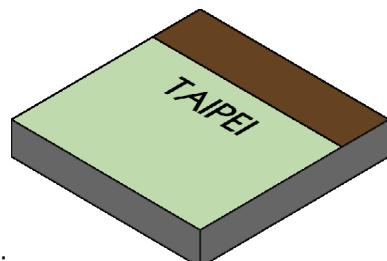
Figure 16: Game Board

5.4.6 Places

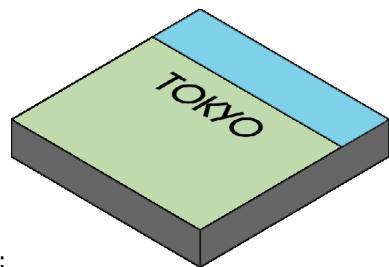
- **Properties**



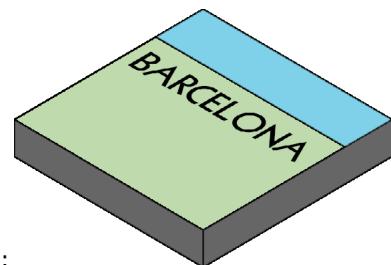
Gdynia:



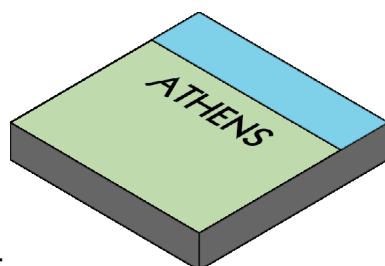
Taipei:



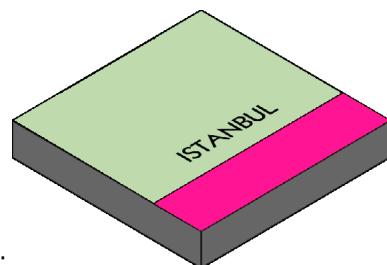
Tokyo:



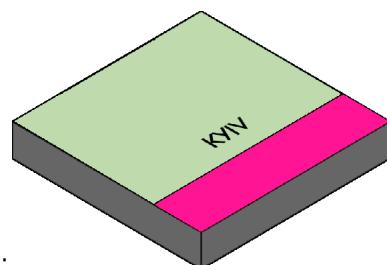
Barcelona:



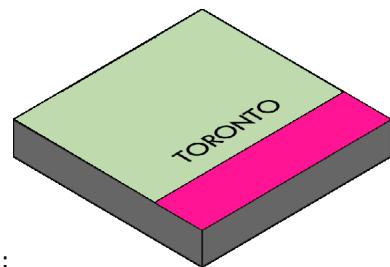
Athens:



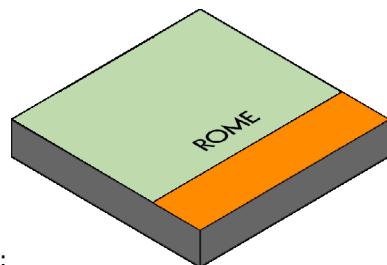
Istanbul:



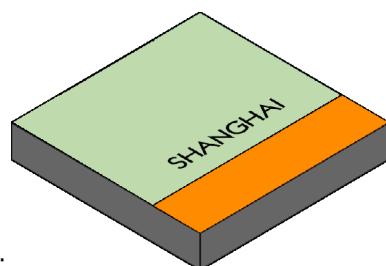
Kyiv:



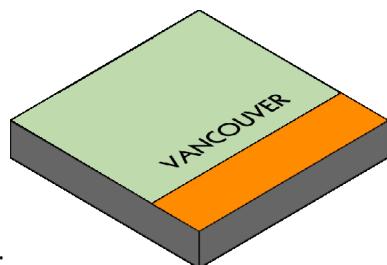
Toronto:



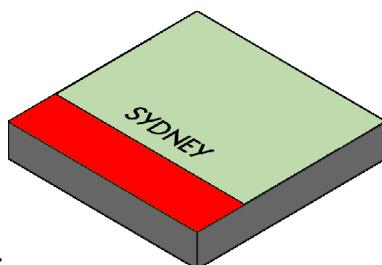
Rome:



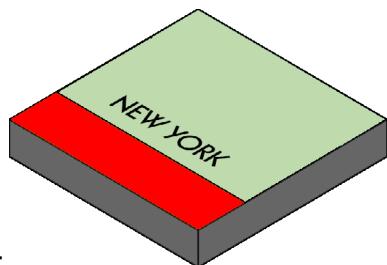
Shanghai:



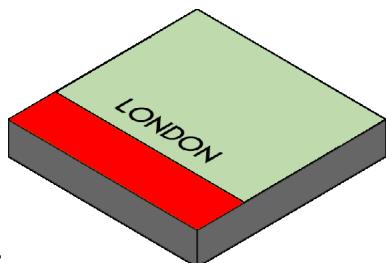
Vancouver:



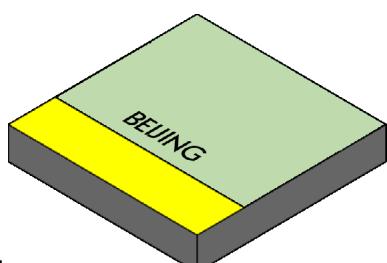
Sydney:



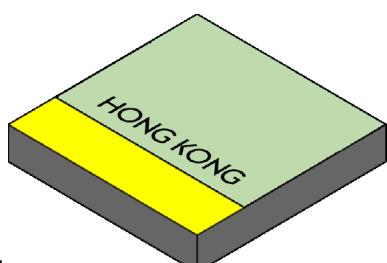
New York:



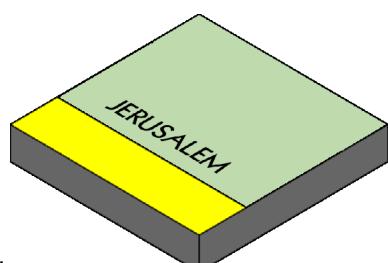
London:



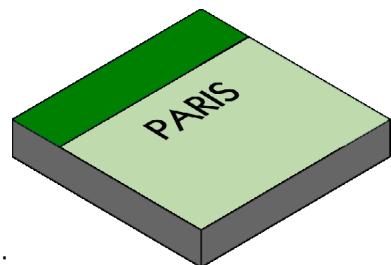
Beijing:



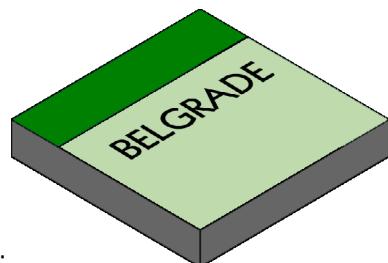
Hong Kong:



Jerusalem:



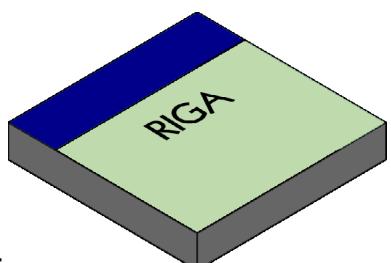
Paris:



Belgrade:



Cape Town:



Riga:

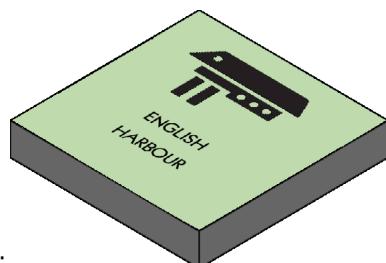


Montreal:

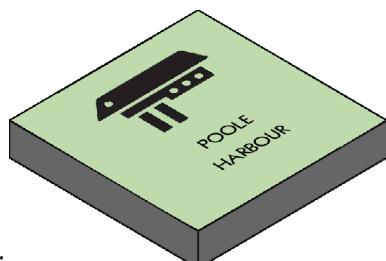
- **Utilities**



Grand Central Station:



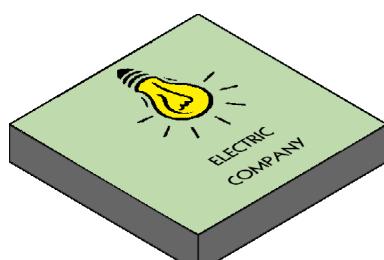
English Harbour:



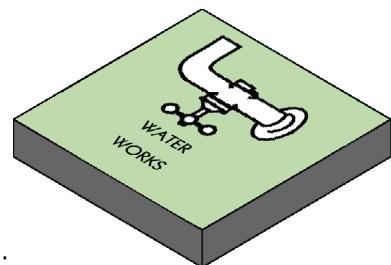
Poole Harbour:



Heathrow Airport:

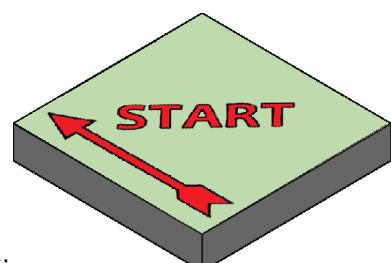


Electric Company:

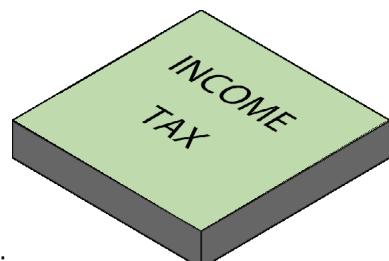


Water Works:

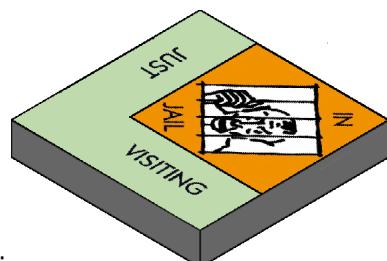
- **Special Places**



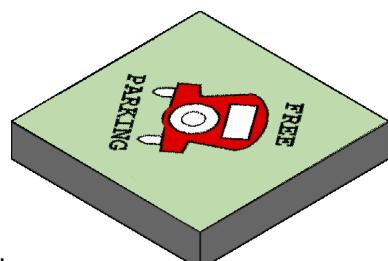
Start:



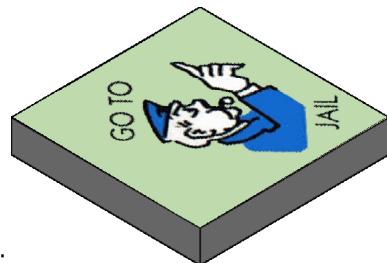
Income Tax:



Visitor:



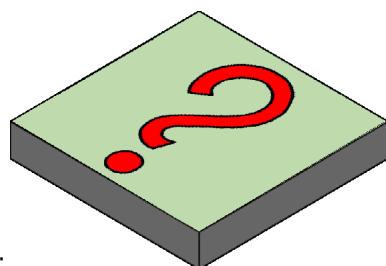
Free Parking:



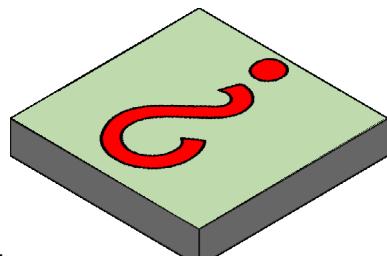
Go to Jail:



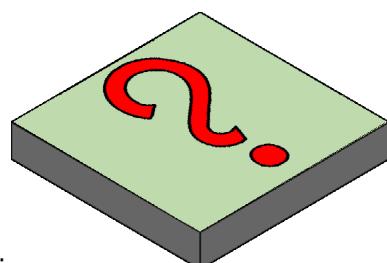
Luxury Tax:



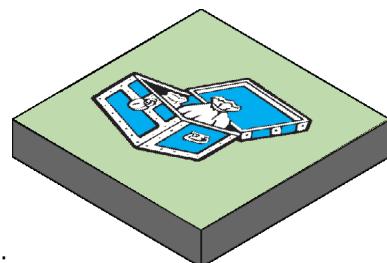
Lower Left Chance:



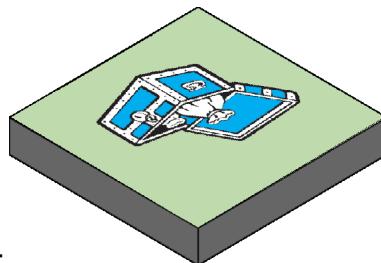
Upper Right chance:



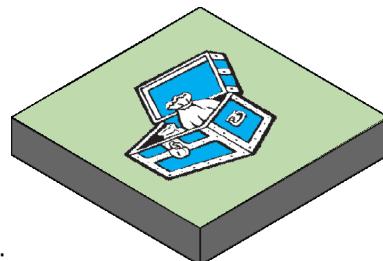
Lower Right Chance:



Lower Left Community Chest:



Upper Left Community Chest:



Lower Right Community Chest:

- Property Cards**

GDYNIA	
PRICE \$ 60 RENT \$ 2	
With 1 house	\$ 10
With 2 house	\$ 30
With 3 house	\$ 90
With 4 house	\$ 160
With HOTEL	\$ 250
One house costs \$ 50 Mortgage value \$ 30	

TAIPEI	
PRICE \$ 60 RENT \$ 4	
With 1 house	\$ 20
With 2 house	\$ 60
With 3 house	\$ 180
With 4 house	\$ 320
With HOTEL	\$ 450
One house costs \$ 50 Mortgage value \$ 30	

TOKYO	
PRICE \$ 100 RENT \$ 6	
With 1 house	\$ 30
With 2 house	\$ 90
With 3 house	\$ 270
With 4 house	\$ 400
With HOTEL	\$ 550
One house costs \$ 50 Mortgage value \$ 50	

BARCELONA	
PRICE \$ 100 RENT \$ 6	
With 1 house	\$ 30
With 2 house	\$ 90
With 3 house	\$ 270
With 4 house	\$ 400
With HOTEL	\$ 550
One house costs \$ 50 Mortgage value \$ 50	

ATHENS	
PRICE \$ 120 RENT \$ 8	
With 1 house	\$ 40
With 2 house	\$ 100
With 3 house	\$ 300
With 4 house	\$ 450
With HOTEL	\$ 600
One house costs \$ 50 Mortgage value \$ 60	

ISTANBUL	
PRICE \$ 140 RENT \$ 10	
With 1 house	\$ 50
With 2 house	\$ 150
With 3 house	\$ 450
With 4 house	\$ 625
With HOTEL	\$ 750
One house costs \$ 100 Mortgage value \$ 70	

KYIV	
PRICE \$ 140 RENT \$ 10	
With 1 house	\$ 50
With 2 house	\$ 150
With 3 house	\$ 450
With 4 house	\$ 625
With HOTEL	\$ 750
One house costs \$ 100 Mortgage value \$ 70	

TORONTO	
PRICE \$ 160 RENT \$ 12	
With 1 house	\$ 60
With 2 house	\$ 180
With 3 house	\$ 500
With 4 house	\$ 700
With HOTEL	\$ 900
One house costs \$ 100 Mortgage value \$ 80	

ROME	
PRICE \$ 180 RENT \$ 14	
With 1 house	\$ 70
With 2 house	\$ 200
With 3 house	\$ 550
With 4 house	\$ 700
With HOTEL	\$ 900
One house costs \$ 100 Mortgage value \$ 90	

SHANGHAI	
PRICE \$ 180 RENT \$ 14	
With 1 house	\$ 70
With 2 house	\$ 200
With 3 house	\$ 550
With 4 house	\$ 700
With HOTEL	\$ 950
One house costs \$ 100 Mortgage value \$ 90	

VANCOUVER	
PRICE \$ 200 RENT \$ 16	
With 1 house	\$ 80
With 2 house	\$ 220
With 3 house	\$ 600
With 4 house	\$ 800
With HOTEL	\$ 1000
One house costs \$ 100 Mortgage value \$ 100	

SYDNEY	
PRICE \$ 220 RENT \$ 18	
With 1 house	\$ 90
With 2 house	\$ 250
With 3 house	\$ 700
With 4 house	\$ 875
With HOTEL	\$ 1050
One house costs \$ 150 Mortgage value \$ 110	

NEW YORK	
PRICE \$ 220 RENT \$ 18	
With 1 house	\$ 90
With 2 house	\$ 250
With 3 house	\$ 750
With 4 house	\$ 875
With HOTEL	\$ 1050
One house costs \$ 150 Mortgage value \$ 120	

LONDON	
PRICE \$ 240 RENT \$ 20	
With 1 house	\$ 100
With 2 house	\$ 300
With 3 house	\$ 750
With 4 house	\$ 925
With HOTEL	\$ 1100
One house costs \$ 150 Mortgage value \$ 120	

BEIJING	
PRICE \$ 260 RENT \$ 22	
With 1 house	\$ 110
With 2 house	\$ 330
With 3 house	\$ 800
With 4 house	\$ 975
With HOTEL	\$ 1150
One house costs \$ 150 Mortgage value \$ 130	

HONG KONG	
PRICE \$ 260 RENT \$ 22	
With 1 house	\$ 110
With 2 house	\$ 330
With 3 house	\$ 800
With 4 house	\$ 975
With HOTEL	\$ 1150
One house costs \$ 150 Mortgage value \$ 130	

JERUSALEM	
PRICE \$ 280 RENT \$ 24	
With 1 house	\$ 120
With 2 house	\$ 360
With 3 house	\$ 850
With 4 house	\$ 1025
With HOTEL	\$ 1200
One house costs \$ 150 Mortgage value \$ 140	

PARIS	
PRICE \$ 300 RENT \$ 26	
With 1 house	\$ 130
With 2 house	\$ 390
With 3 house	\$ 900
With 4 house	\$ 1100
With HOTEL	\$ 1275
One house costs \$ 200 Mortgage value \$ 150	

BELGRADE		CAPE TOWN		RIGA	
PRICE \$ 300	RENT \$ 26	PRICE \$ 320	RENT \$ 28	PRICE \$ 350	RENT \$ 35
With 1 house	\$ 130	With 1 house	\$ 150	With 1 house	\$ 175
With 2 house	\$ 390	With 2 house	\$ 450	With 2 house	\$ 500
With 3 house	\$ 900	With 3 house	\$ 1000	With 3 house	\$ 1100
With 4 house	\$ 1100	With 4 house	\$ 1200	With 4 house	\$ 1300
With HOTEL	\$ 1275	With HOTEL	\$ 1400	With HOTEL	\$ 1500
One house costs \$ 200		One house costs \$ 200		One house costs \$ 200	
Mortgage value \$ 150		Mortgage value \$ 160		Mortgage value \$ 175	

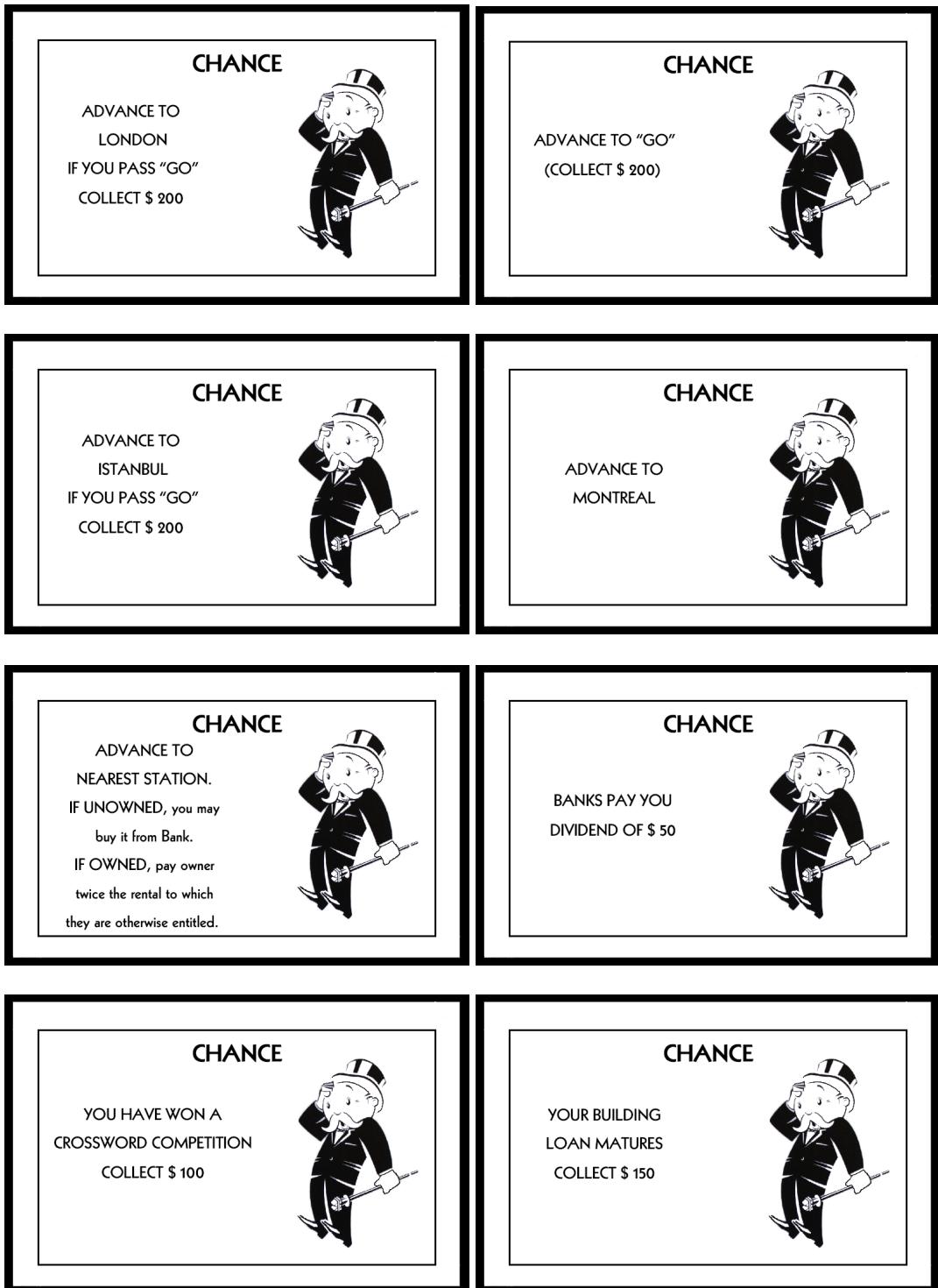
MONTREAL	
PRICE \$ 400	RENT \$ 50
With 1 house	\$ 200
With 2 house	\$ 600
With 3 house	\$ 1400
With 4 house	\$ 1700
With HOTEL	\$ 2000
One house costs \$ 200	
Mortgage value \$ 200	

• **Utility Cards**

GRAND CENTRAL STATION		ENGLISH HARBOUR		HEATHROW AIRPORT	
PRICE \$ 200		PRICE \$ 200		PRICE \$ 200	
If 1 owned	\$ 25	If 1 owned	\$ 25	If 1 owned	\$ 25
If 2 owned	\$ 50	If 2 owned	\$ 50	If 2 owned	\$ 50
If 3 owned	\$ 100	If 3 owned	\$ 100	If 3 owned	\$ 100
If 4 owned	\$ 200	If 4 owned	\$ 200	If 4 owned	\$ 200
Mortgage value \$ 100		Mortgage value \$ 100		Mortgage value \$ 100	

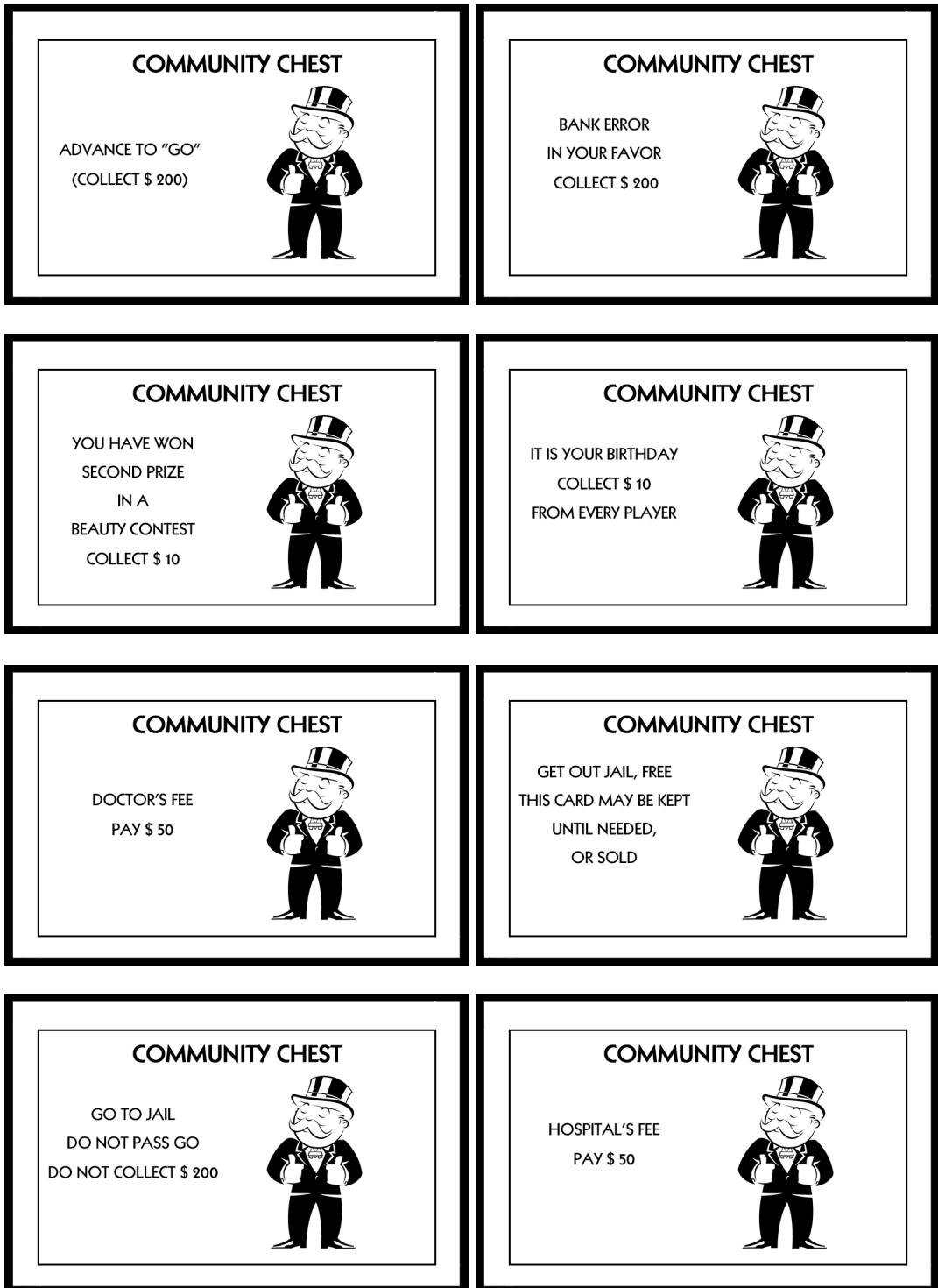
POOLE HARBOUR		ELECTRIC COMPANY		WATER WORKS	
PRICE \$ 200		PRICE \$ 150		PRICE \$ 150	
If 1 owned	\$ 25	If 1 owned, rent equals 4 times dice roll		If 1 owned, rent equals 4 times dice roll	
If 2 owned	\$ 50	If 2 owned, rent equals 10 times dice roll		If 2 owned, rent equals 10 times dice roll	
If 3 owned	\$ 100				
If 4 owned	\$ 200				
Mortgage value \$ 100		Mortgage value \$ 75		Mortgage value \$ 75	

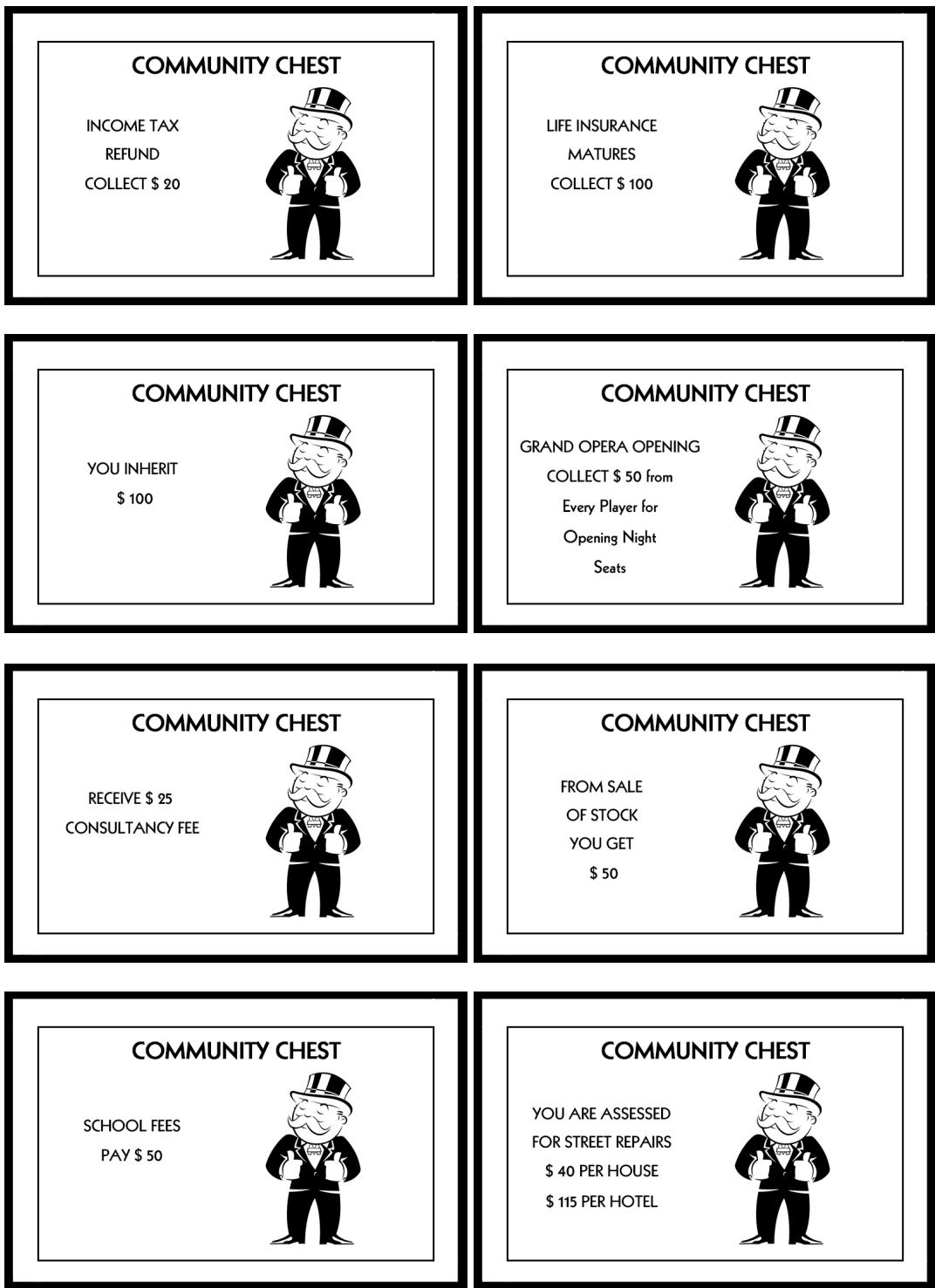
- **Chance Cards**





- **Community Chest Cards**



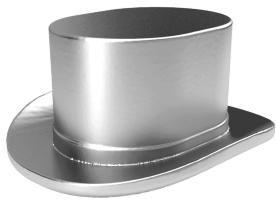




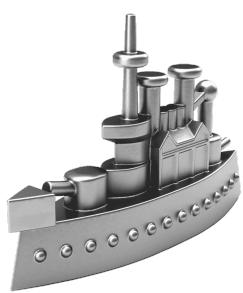
- **Pawns**



Car:



Hat:



Ship:



Shoe:

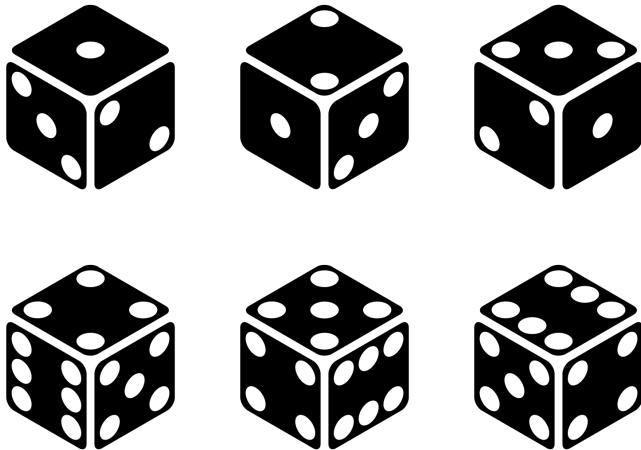


Iron:



Thimble:

- **Dice**



6 Improvement Summary

- 1) Class Diagram was updated, a new class was added and explained in the explanation part.
- 2) Use case diagram is updated.
- 3) New activity diagrams are added, new state diagrams are added.
- 4) Property lands have changed concerning the feedback. Rotated texts on property lands changed to straight texts to improve users' experience.
- 5) Six different pawns were added.
- 6) Resolution option has been removed from the settings screen.
- 7) New dice images added.
- 8) The veto right rule is changed in order to maintain simplicity for the users.
- 9) Players are not able to sell their properties anymore due to the increment in complexity of the operations and their orders.
- 10) There is no dice roll operation to determine the line-up of the players due to the complexity it may cause. The players line-up will be determined according to the order they enter the game.
- 11) The hosts are able to share an id to start a game.
- 12) The rules for auction are changed to maintain simplicity.
- 13) The ending of the game is changed to save time and to keep players engaged in the game.
The game will end when one of the players is bankrupt.

Glossary & References

- [1] Object-Oriented Software Engineering, Using UML, Patterns, and Java, 3rd Edition, by Bernd Bruegge and Allen H. Dutoit, Prentice-Hall, 2010, ISBN-10: 0136066836.
- [2] <https://www.hasbro.com/common/instruct/monins.pdf>
- [3] Applying UML and Patterns - An Introduction to Object-Oriented Analysis and Design and Iterative Development, by Craig Larman, Prentice Hall, 2004, ISBN: 0-13-148906-2.
- [4] <https://monopoly.fandom.com>