Bilkent University

**CS 319 - Object-Oriented Software Engineering**

**Iteration 1 - Project Analysis Report**

**Group 3B - The Settlers Of Catan**

Mustafa Çağrı Güngör

Fatih Sevban Uyanık

Çerağ Oğuztüzün

Talha Burak Çuhadar

İbrahim Elmas

**Instructor**: Eray Tüzün

**CONTENTS**

[**Introduction**](#_heading=h.an92nqjxg6ar) **…………………………………………………………………………………………………………………………………………………………………..4**

[**Overview**](#_heading=h.bp9tvl79nfwf) **…….………………………………………………………………………………………………………………………………………………………………..5**

[Game Component](#_heading=h.7s900esyrhfx)**…………………...……………………………………………………………………………………………………………………………..**[5](#_heading=h.7s900esyrhfx)

[Player](#_heading=h.9uxbqx9lc2rn) **…………………...………………...……………………………………………………………………………………………………………………………..**6

[Settings](#_heading=h.142l983jq3k5) **…………………...…………………………………...………………………………………………………………………………………………………..**6

[Maps](#_heading=h.lmbwrg4tirn5) **…………………...………………………………………………………………………………………………………………………………………………...**6

[Cards](#_heading=h.ce5g52e09ey) **…………………...………………………………………………………………………………………………………………………………………………...**6

[Constructions](#_heading=h.unc58x7zcvnc) **…………………...…………………………………………………………………………………………………………………………………….**6

[Instructions](#_heading=h.yvcfewliys96) **…………………...……………………………………………………………………………………………………………………………………….**7

[**Functional Requirements**](#_heading=h.ug9inhh2a57v) **…………………...………………………………………………………………………………………………………………………...7**

[Player](#_heading=h.6vfzpixetfn9) **…………………...………………………………………………………………………………………………………………………………………………..**7

[Map](#_heading=h.jj39ejm83j10) **…………………...…………………………………………………………………………………………………………………………………………………..**7

[Terrain Hexes](#_heading=h.yrwakacmj38) **…………………...……………………………………………………………………………………………………………………………..**8

[Forests](#_heading=h.ne3she9bbrlb) **…………………...………………………………………………………………………………………………………………………………...**8

[Mountains](#_heading=h.k1s9yle1kha2) **…………………...…………………………………………………………………………………………………………………………...**8

[Fields](#_heading=h.utmbzxh81eyx) **…………………...…………………………………………………………………………………………………………………………………..**8

[Hills](#_heading=h.jp1jqx315j3) **…………………...……………………………………………………………………………………………………………………………………..**8

[Pastures](#_heading=h.69xrxyqfkrx9) **…………………...……………………………………………………………………………………………………………………………….**8

[Desert](#_heading=h.nywgfd112z17) **…………………...………………………………………………………………………………………………………………………………….**9

[Harbors](#_heading=h.63ybv5v1ueo) **…………………...……………………………………………………………………………………………………………………………………...**9

[Vertices](#_heading=h.klgw3ofecbar) **…………………...……………………………………………………………………………………………………………………………………...**9

[Cards](#_heading=h.iw6an833cw8v) **…………………...………………………………………………………………………………………………………………………………………………...**9

[Development Cards](#_heading=h.470j9yntkhny) **…………………...…………………………………………………………………………………………………………………….**9

[Knight Cards](#_heading=h.ncecjvriyjv0) **…………………...……………………………………………………………………………………………………………………….**10

[Victory Point Cards](#_heading=h.10mvlrwj4o9y) **…………………...…………………………………………………………………………………………………………….**10

[Monopol Cards](#_heading=h.9nyqwurgsn8l) **…………………...………………………………………………………………………………………………………………….**10

[Road Destruction Cards](#_heading=h.mdvmlz5j5eof) **…………………...………………………………………………………………………………………………………**10

[Profit Interchange Cards](#_heading=h.90ki4w62ic8y) **…………………...…………………………………………………………………………………………………….**10

[Invention Cards](#_heading=h.t70sdr4048ix) **…………………...………………………………………………………………………………………………………………….**10

[Resource Cards](#_heading=h.qie44n6jc66a) **…………………...………………………………………………………………………………………………………………………...**10

[Entitlement Cards](#_heading=h.5fihepn0917m) **…………………...……………………………………………………………………………………………………………………..**11

[Price Cards](#_heading=h.uyouh7n82c5u) **…………………...………………………………………………………………………………………………………………………………..**11

[Constructions](#_heading=h.3znysh7) **…………………...…………………………………………………………………………………………………………………………………..**11

[Roads](#_heading=h.r2e65pcsws1h) **…………………...……………………………………………………………………………………………………………………………………….**11

[Villages](#_heading=h.ftj18bru8nqj) **…………………...……………………………………………………………………………………………………………………………………..**11

[Cities](#_heading=h.naqqkrjk4unk) **…………………...………………………………………………………………………………………………………………………………………..**11

[Civilizations](#_heading=h.7mss96p58537) **…………………...……………………………………………………………………………………………………………………………….**12

[Interactions Between Players](#_heading=h.rq8h8zdswzxa) **…………………...…………………………………………………………………………………………………………...**12

[Thief](#_heading=h.ktb3foieashn) **…………………...………………………………………………………………………………………………………………………………………...**12

[Trading](#_heading=h.orju0m5cjroc) **…………………...……………………………………………………………………………………………………………………………………..**12

[Settings](#_heading=h.axx9zgc34flh) **…………………...…………………………………………………………………………………………………………………………………………...**12

[Theme Settings](#_heading=h.ck2yxh9xfibm) **…………………...………………………………………………………………………………………………………………………...**12

[Sound Settings](#_heading=h.rvbt1fice7r5) **…………………...………………………………………………………………………………………………………………………….**13

[Threshold Settings](#_heading=h.kjc7nyg1d6sp) **…………………...…………………………………………………………………………………………………………………….**13

[Instructions](#_heading=h.3oopuzrpoep2) **…………………...……………………………………………………………………………………………………………………………………..**13

[Credits](#_heading=h.kbym3dgoq72) **…………………...…………………………………………………………………………………………………………………………………………….**13

[**Non-functional Requirements**](#_heading=h.6f2fu1u51bun) **…………………...………………………………………………………………………………………………………………..13**

[Extendibility](#_heading=h.o8chfza58xdh) **…………………...…………………………………………………………………………………………………………………………………….**13

[User-Friendly Interface](#_heading=h.ty3llyruix0d) **…………………...…………………………………………………………………………………………………………………….**14

[Reliability](#_heading=h.pvvf776l70yg) **…………………...………………………………………………………………………………………………………………………………………..**14

[**System Models**](#_heading=h.obry98plqjg3) **…………………...……………………………………………………………………………………………………………………………………...14**

[Use Case Model](#_heading=h.dw20kls2hxp9) **…………………...……………………………………………………………………………………………………………………………...**14

[Use Case Descriptions](#_heading=h.hpjw7brbnylx) **…………………...……………………………………………………………………………………………………………….**15

[Dynamic models](#_heading=h.9lj3jvot61h6) **…………………...……………………………………………………………………………………………………………………………...**19

[Sequence Diagrams](#_heading=h.sb5eelxgq24m) **…………………...…………………………………………………………………………………………………………………..**19

[Play Game Turn](#_heading=h.8yigkrnmgiqb) **…………………...………………………………………………………………………………………………………………….**19

[Trade](#_heading=h.1e0lpy3pwlmc) **…………………...………………………………………………………………………………………………………………………………….**20

[Make Construction](#_heading=h.32q49cp81wtt) **…………………...……………………………………………………………………………………………………………..**21

[Change Settings](#_heading=h.kuwysjsyxem3) **…………………...………………………………………………………………………………………………………………….**22

[Activity Diagrams](#_heading=h.yut57vyzo4db) **…………………...……………………………………………………………………………………………………………………..**23

[State Diagrams](#_heading=h.t05p6ohrmn39) **…………………...………………………………………………………………………………………………………………………...**25

[Object and Class Model](#_heading=h.d44gexdr1f1x) **…………………...…………………………………………………………………………………………………………………...**26

[User interface](#_heading=h.gzdbgxaq1e3i) **…………………...………………………………………………………………………………………………………………………………….**29

[**Conclusion**](#_heading=h.d826iuwk3hfq) **…………………...……………………………………………………………………………………………………………………………………………..**33

**Glossary & References…………………...…………………………………………………………………………………………………………………………...**33

# Introduction

 Figure 1: The representative picture of The Settlers of the Catan.

The Settlers of the Catan is a strategy game played by 4 people where everyone plays in turns. Our digital adaptation of the game is single player, other players played by AI. Before the game starts, the player can choose the theme of the game, can change the default thresholds for achieving Longest Road Card and Strongest Army Card, can change the winning points that is fixed to 10 in default by entering the settings menu in the main menu. There are different themes that change the music that plays during the game, type of resource cards, map background. New players can enter the instructions menu to be knowledgeable of the game. Music can be turned on/off during the game.

After the game starts, the game automatically shuffles the terrain hexes and their random numbers from 1 to 12, then places them to the map. The player enters their name and chooses their color to play with. Every construction is the color of their player. Construction Price Card is placed to the game screen where the price information about construction such as village, cities or civilization and the points they provide to the players are written. Constructions can be settled to the vertices of the terrain hexes. Only roads can be settled edge of the terrain hexes. The turns of the players are randomly assigned to them by the game. At the beginning of the first turn, the player who has the turn rolls two die and place villages and roads on the map to a vertices/edges he desires. After each player settles their first villages and roads, the competition literally starts.

In each turn, the player rolls the die and can collect sources, buy or use development cards, trade (swap sources), and build constructions. After the die are rolled, players who own constructions located on the vertex of the terrain hex which has the number equal with the sum of the dies gains resource produced by the terrain hex. There are five types of Resource Cards which are brick, wool, ore, grain, and lumber in the default game theme. These Resource Cards are utilized while doing construction and buying Development Cards. If the sum of numbers on the die is seven, each player who has seven or more cards places half of them to the chest, also the player who has rolled the die can place the thief figure on wherever terrain hex on map he/she wishes where the next player won't be able to get source cards from that place. The player who places the thief gets a random source card from each player who owns construction in that terrain hex. The player who owns the turn can trade his/her source cards with other players or the chest. In order to construct a village, the player must reach that part of the game by building roads.

Villages can be upgraded to cities. Cities can be upgraded to civilizations. There are three types of development cards that add more game functionality to the game. If a player collects fixed points from constructions and cards, the player wins and the game ends.

# Overview

## Game Components

* 19 terrain hexes
* 9 harbors
* 18 number signs
* 95 Resource Cards
* 25 Development Cards (14 Knight/Soldier Cards, 6 Progress Cards, 5 Victory Point Cards)
* 4 “Building Costs” Cards
* 2 Special Cards: “Longest Road” and “Largest Army”
* 20 village (4 of each color shaped)
* 20 cities (4 of each color shaped)
* 20 civilization (4 of each color shaped)
* 60 roads (15 of each color shaped)
* 2 dice
* 1 robber

## Player

Catan is a game played by 4 players. In our adaptation, It is adapted as a single player game where the other 3 players are controlled by AI. The players are able to make new constructions, gather Resource Cards, use Development Cards, trade and interact with each other and roll die in each turn. Depending on the constructions and cards, users obtain Victory Points. The player that is the first to have 10 Victory Points wins the game.

## Settings

In the game, there is a settings menu section at the beginning where the player is able to change the theme and sound settings which provide a better user interface quality. In addition, through the settings, users can determine various thresholds to achieve Entitlement Cards and Victory Points.

## Maps

In the map, there are 19 terrain hexesthat consist ofForests, Mountains, Fields, Hills, Pastures and Desert**.** Every terrain hex has a number that refers to the sum of die rolls of the player which to take resource if the player owns a construction on a vertex of that terrain hex. In addition to the terrain hexes, there are 6 harbors that enable easier trading. Constructions can be built on vertices which are the crossings of terrain hexes.

## Cards

There are a variety of cards distributed or exchanged by the players in each turn. To visit the utility of each card type briefly: Resource Cards refer to the resources obtained from each field the player has construction on. Players use these cards to build constructions or buy Development Cards. Development Cards consist of Knight Cards, Victory Point cards, Monopol Cards, Road Destruction Cards, Profit Interchange Cards and Invention Cards which change the power balance between players and make the game more exciting. The Entitlement Cards makes the player gain more Victory Points when a certain threshold of Roads and Knight Cards are obtained, however, the Entitlement Cards are other cards to alter the power balance between players. Lastly, Price Cards are viewed by the player, which gives the information about how many of which Source Cards can be used.

## Constructions

There are constructions in the game which the players can build using Resource Cards. The vertices the construction is built on yields the player with Resource Cards from the adjacent map fields at each turn. Constructions can be roads, villages, cities and civilizations. Villages can be upgraded to Cities and Civilizations, each yield player more Victory Points and yields more Resource Cards. These constructions are built on vertices of the hexagonal map, and in order to build a construction, the player has to reach that part of the map by building roads between vertices.



Figure 2: The villages, cities and roads for red color are demonstrated in the board game.

## Instructions

Instructions function gives access to the player to view instructions on how to play the game. These sections provide the player with a good read to be knowledgeable about the game and the rules, in order to play the game with the greatest extent of fun.

# Functional Requirements

## Player

The game is single player. There are 4 players present in Catan, 3 of them are controlled by AI. At the first round, every player place 2 villages and two roads. After the initial step, the real game starts and, in each turn, players roll the dice and depending on the rolled number, every player gets Resource Cards by the courtesy of their constructions. Based on the collected Resource Cards, players can make new constructions. For example, roads, villages, cities and civilizations. Players can buy or use development cards and these development cards have different functionalities such as bringing victory points, constructing additional roads etc. The main purpose of each player is to collect as many victory points as possible. In the default game, the player first having 10 victory points is the winner of the game.

## Map

The entire map has a hexagonal shape and consists of 19 terrain hexes, harbors and vertices. Each of these units is explained in the following section.

### Terrain Hexes



Figure 3: Terrain hexes from the board game.

There are 6 types of terrain hexes that are Forests, Mountains, Fields, Hills, Pastures and Desert. Every terrain hex has a number on it and if the sum of die rolls is equal to the number on it, the players who have a construction on the vertices of the terrain hexes get corresponding resource cards.





Figure 4: Types of Terrain Hexes, their pictures and their resources.

#### Forests

Forests provide lumber resource cards to players who have construction on its vertices.

#### Mountains

Mountains provide ore resource cards to players who have construction on its vertices.

#### Fields

Fields provide grain resource cards to players who have construction on its vertices.

#### Hills

Hills provide brick resource cards to players who have construction on its vertices.

#### Pastures

Pasture provides wool resource cards to players who have construction on its vertices.

#### Desert

There is a single Desert terrain in the game. The dessert does not provide a resource card to players. At the beginning of the game, the thief is located here. The place of the thief can be changed from the dessert to another terrain hex afterwards by the players.

### Harbors

There are 6 harbors on the vertices of the map that enable easier trading with the chest. In default, to be able to trade with the chest, players need to put four identical resource cards to the chest and hence, the player gets a card of interest. On the harbors, there are ratios written that makes it easier to trade. For instance, the ratio is written as “2 lumber: 1 brick”. In that case, the player can trade bricks easier with lumbers because instead of giving 4 lumbers to the chest, the user gives 2 lumbers away.

### Vertices

Vertices are the crossing sections of each terrain hexes. On these vertices, constructions such as villages, cities, civilizations can be built. Between the vertices, roads can also be constructed. A vertex can be located in between only terrain hexes or terrain hexes and harbors.

## Cards

The game dynamics are consisted of by cards mostly. There are a variety of cards distributed or exchanged by the players in each turn. To visit the utility of each card type briefly: Resource Cards refer to the resources obtained from each field the player has construction on. Players use these cards to build constructions or buy Development Cards. Development Cards add a different dynamic to the game with each card, there are Knight Cards, Victory Point cards, Monopol Cards, Road Destruction Cards, Profit Interchange Cards and Invention Cards which change the power balance between players and make the game more exciting. Road Destruction Cards and Profit Interchange Cards are new features that are added by the programmers. The Entitlement Cards makes the player gain more Victory Points when a certain threshold of Roads of Knight Cards is obtained, however, the Entitlement Cards are other cards to alter the power balance between players, they are dynamic and change owner when the maximum number of cards or roads change by the process of the game. Lastly, Price Cards are viewed by each player, which gives the information about how many of which Source Cards can be used to make which constructions.

### Development Cards

The Development Cards are obtained by Source Cards. They change the power dynamics of the game between players to provide a more exciting experience to the players. When a development card is bought, be used in a desired, single turn. There is a variety of Development Cards each have different functionality in the game, there are Knight Cards, Victory Point Cards, Monopol Cards, Road Destruction Cards, Profit Interchange Cards and Invention Cards.

#### Knight Cards

When the Knight Card is used, Thief can be replaced by the player who played the card, to the desired terrain hex. The player who plays the Knight Card gets a random Resource Card from a player who owns a construction on an adjacent vertex of the new terrain hex the thief is placed on.

#### Victory Point Cards

When the Victory Point Card is used, the player gets the number of Victory Points that is written on the card. The Victory Points are crucial to win the game.

#### Monopol Cards

When the Monopol Card is used, the player who plays the card indicates a type of Resource Card that every player has to give every Resource Card of that type to the player who played the Monopol Card.

#### Road Destruction Cards

The Road Destruction Cards is an improvement by the developers. When the Road Destruction Card is used, the player who plays the card has the chance the destroy one of the roads constructed by any other player. This situation can alter the owner of the Longest Road Entitlement Card which can change the power balance between players. The player who destroyed the road can also build a road at that place by using Resource Cards in the turn.

#### Profit Interchange Cards

The Profit Interchange Cards is an improvement by the developers. When the Profit Interchange Card is used, the numbers on terrain hexes are switched by the player. One of the numbers must be the number of sums of die rolls of the player at the turn the Profit Interchange Card is played. The number wanted to switch with the terrain hex that is numbered with the sum of die rolls of player, is left to the player’s wish. This functionality changes the power dynamics of the game to a great level.

#### Invention Cards

When the Invention Card is used, the player who plays the card can get two Resource Cards of the types the player desires, from the chest.

### Resource Cards

Resource Cards are the main money-units of the game. The player needs them in order to build constructions and buy Development Cards. They can also be traded among the players with mutual agreement, or with the chest according to the trading rules of the chest. There are a number of Resource Cards each yielded from a certain type of terrain hex. Hills produce the Brick Resource Card, Pasture produces Wool Resource Card, Mountains produce Ore Resource Card, Fields produce Grain Resource Card, Forest produces Lumber Resource Card and Desert do not produce any Resource Card.

### Entitlement Cards

The Entitlement Cards cannot be bought with Resource Cards but can be attained when the player gets a certain entitlement. These cards are attained when the player owns a certain number of consecutive roads or Knight Cards of the Development Cards. If the player has the longest number of consecutive roads in the game, starting from a threshold, the Longest Road Card grants the player with 2 Victory Points. If the player has the greatest number of played Knight Cards among other players, starting from a threshold, the Strongest Army Card grants the player with 2 Victory Points. However, these cards can change hands and the Victory Points earned by these cards can be lost when another player exceeds the related numbers of units in the game.

### Price Cards

The Price Card indicates the costs of constructions for the players to build, in terms of Resource Cards. This prices in this card apply to all players in the game and It is viewed constantly for players to make investment plans according to it throughout the game.

## Constructions

Constructions are the main utilities in the game that make the players gain Resource Cards, which are one of the essential elements of the game. At the beginning of the game, each player is given 2 villages and 2 roads to place on the map to initiate their power. In the game, every construction costs a number of Resource Cards that is indicated on the Price Card.

### Roads

Roads are built between two vertices on the map. Built roads have to be at least one shared vertex.

### Villages

Villages are built on the vertices of terrain hexes. When the sum of numbers on dies on each turn is the number of terrain hex what is adjacent to the village of the user, the player gets a Resource Card produced by that terrain hex, from the chest. Constructing a village gives the player 1 Victory Point. Villages can be upgraded to be Cities.

### Cities

Cities are the upgraded forms of Villages, means the player cannot build cities from nonexistent Villages. The upgrade has a cost of Resource Cards which are indicated on Price Cards. Upgrades include: getting 2 Resource Cards from the related terrain hex. Each City gives the player 2 Victory Points, the Victory Point obtained from the Village is discarded. Cities will be upgraded to Civilizations.

### Civilizations

Civilizations are the upgraded forms of Cities, means the player cannot be built Civilizations from nonexistent Cities. The upgrade has a cost of Resource Cards which are indicated on Price Cards. Upgrades include: getting 3 Resource Cards from the related terrain hex. Each Civilization gives the player 3 Victory Points, the Victory Point obtained from the City is discarded.

## Interactions Between Players

The excitement of the game is on the dynamic power balances between players. The game forces players to interact with other players. At times you hand your cards to them, at times you negotiate with other players to obtain your goals in the game.

### Thief

Thief is a movable figure on the map. When a player rolls the die and 7 comes, the player can place the thief on the desired terrain hex. The player can choose a player who owns a construction adjacent to the terrain hex the thief is placed and get a random Resource Card from that player. As long as the thief is on that terrain hex, Resource Cards that would be yielded from that terrain hex by the players who own a construction adjacent to there will be dismissed.

### Trading

Thief is a movable figure on the map. When a player rolls the die and 7 comes, the player can place the thief on the desired terrain hex. The player can choose a player who owns a construction adjacent to the terrain hex the thief is placed and get a random Resource Card from that player. As long as the thief is on that terrain hex, Resource Cards that would be yielded from that terrain hex by the players who own a construction adjacent to there will be dismissed.

## Settings

In the beginning, there is a main menu containing a settings option. The player is able to choose the game theme from various themes, sounds and determine some thresholds for various events. Furthermore, the sound can be muted and unmuted on every screen. That can be done through the sound button at the right top corner.

### Theme Settings

Theme Settings provide user to choose from a variety of themes. Each theme has a unique music, field and card names, resources corresponding to the theme chosen. The visuals of the UI is also changed according to the theme chosen. The game’s default theme is Catan’s classical theme.

### Sound Settings

Sound of the game can be muted and unmuted throughout the game. Different sounds can also be chosen corresponding to the theme of game chosen from the Theme Settings to create a better user experience.

### Threshold Settings

Through the settings, users can determine the road count threshold to achieve the Longest Road Entitlement Card and the knight count threshold to win the Strongest Army Entitlement Card. In addition, the threshold for 10 victory points can be changed through the settings because some players might want to play the game longer or shorter depending on their times.

## Instructions

The Instructions screen from the main menu. It provides the players with the essential knowledge of the cards, game dynamics, figures and maps, and the rules of the game to provide an easy start and adaptation to the players, for them to play the game without the boredom of struggling to explore the utilities the game provides.

## Credits

This screen is also accessed from the main menu. On this screen, the players will be presented by the hard-working developers who have brought them this game.

# Non-functional Requirements

## Extendibility

Catan is a game that provides extendibility by its structure. The game has a lot of objects such as cards, constructions and figures that can be varied and extended.

We extended the game by:

* Varying Development Cards: Road Destruction Development Card and Profit Interchange Development Cards.
* Varying Constructions: Civilizations that are upgraded forms of Cities.
* Changing themes of the game by changing music, cards, map and figures.
* Ability to change the Victory Point Threshold to alter game time for players’ desire.
* Ability to change the Longest Road Card Threshold and the Strongest Army Card Threshold to alter the rate of change of power dynamics to fit players’ desire.
* Adding AI feature to other 3 players of the game, for the game to be played with single player mode.

According to our creativity and time limits we, as a group plan to extend the game with more fun and user-friendly components.

## User-Friendly Interface

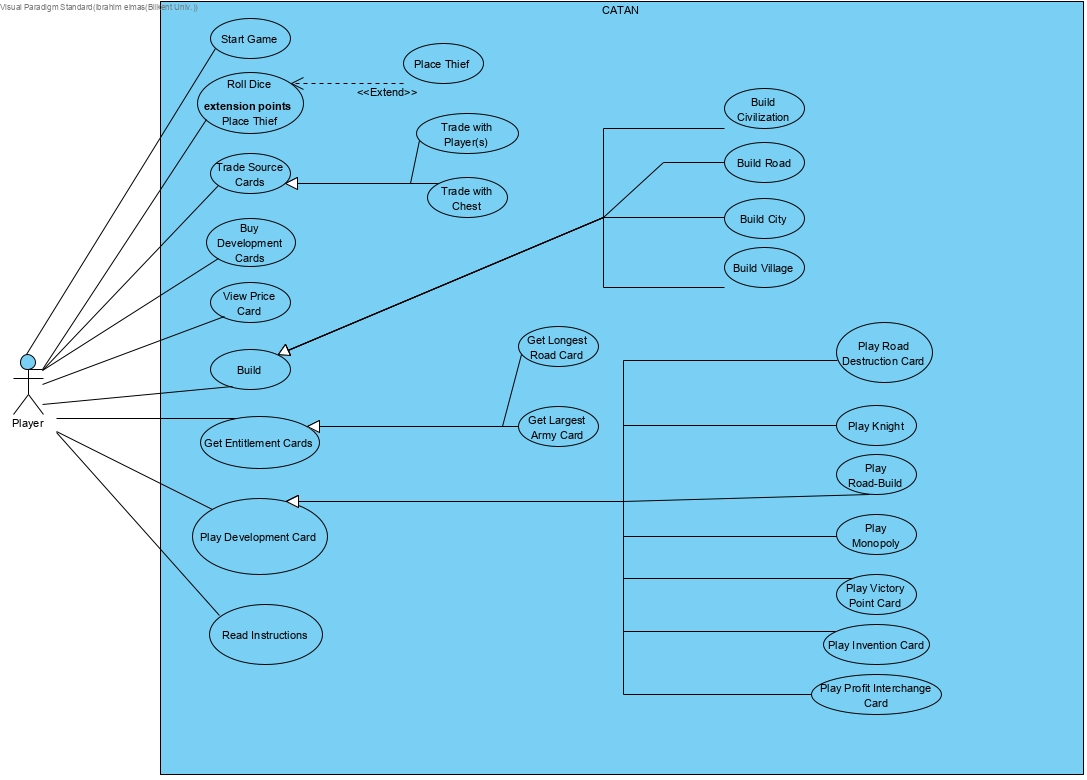
The Catan game user interface is designed as understandable as possible. At the main menu, there are play game button, instructions, settings and exit options. In the Instructions option, the game is explained in detail. In Play Game section, the map is shown, and every player is explicitly directed. Every turn of a player is labelled at the game. At settings, the themes can be changed easily as desired and the corresponding sounds are played. Various thresholds can be changed easily in the settings section, to improve game experience quality as far as possible.

## Reliability

The most important aspect of applications is the stability fact. The application should not terminate in the middle of a game because of having an error or exception. If that would be the case, it would be a terrible user experience. As the developers of the game, this application will be designed and programmed as durable as possible against the errors and exception that might occur.

# System Models

## Use Case Model

Figure 5: Use Case Model of the Catan Game.

### Use Case Descriptions

**Use Case Name: Get Longest Road Card**

**Participant Actor: Inherited** from **Get Entitlement Card** use case**.**

**Flow of events:** The player who has biggest number of roads gets **Longest Road Card.**

**Entry Condition: Inherited** from **Get Entitlement Card** use case.

**Exit Condition: Inherited** from **Get Entitlement Card** use case.

**Use Case Name: Get Largest Army Card**

**Participant Actor: Inherited** from **Get Entitlement Card** use case.

**Flow of events:** The player who has biggest number of knights gets **Largest Army Card.**

**Entry Condition: Inherited** from **Get Entitlement Card** use case.

**Exit Condition: Inherited** from **Get Entitlement Card** use case.

**Use Case Name: Get Entitlement Card**

**Participant Actor:** The Player who is playing the turn.

**Flow of events:**

1. If the player has the biggest number of the roads, he gets the Longest Road Card.
2. If the player has the biggest number of knights, he gets the Largest Army Card.

**Entry Condition:**

1. The Player builds a road and becomes the player who has the biggest number of the roads.
2. The player plays the Knight Card and becomes the player who has the biggest number of knights.

**Exit Condition:** The Player gets entitlement card.

**Use Case Name: Play Road Destruction Card.**

**Participant Actor: Inherited** from **Play Development Card** use case.

**Flow of events:** The player who is playing the turn chooses a enemy road and destroy it.

**Entry Condition: Inherited** from **Play Development Card** use case.

**Exit Condition: Inherited** from **Play Development Card** use case.

**Use Case Name: Play Knight.**

**Participant Actor: Inherited** from **Play Development Card** use case.

**Flow of events:**

1. The player who is playing the turn takes the thief from its location.
2. The player puts the thief where ever he wants.
3. The player wants a source from a player he chose from the players who has edge to the hexagon that the thief is located.

**Entry Condition: Inherited** from **Play Development Card** use case.

**Exit Condition: Inherited** from **Play Development Card** use case**.**

**Use Case Name: Play Road Build.**

**Participant Actor: Inherited** from **Play Development Card** use case.

**Flow of events:** The player who is playing the turn builds two roads without any source.

**Entry Condition Inherited** from **Play Development Card** use case.

**Exit Condition Inherited** from **Play Development Card** use case.

**Use Case Name: Play Monopoly.**

**Participant Actor: Inherited** from **Play Development Card** use case**.**

**Flow of events:**

1. The player who is playing the turn choose a source type.
2. Then, the players take all this type source from all other players.

**Entry Condition: Inherited** from **Play Development Card** use case.

**Exit Condition: Inherited** from **Play Development Card** use case.

**Use Case Name: Play Victory Point Card.**

**Participant Actor: Inherited** from **Play Development Card** use case.

**Flow of events:**

1. The player who is playing the turn shows this card.
2. The players earn 1 victory points.

**Entry Condition: Inherited** from **Play Development Card** use case.

**Exit Condition: Inherited** from **Play Development Card** use case.

**Use Case Name: Play Invention.**

**Participant Actor: Inherited** from **Play Development Card** use case.

**Flow of events:** The player who is playing the turn takes 2 any resource cards from chest.

**Entry Condition: Inherited** from **Play Development Card** use case**.**

**Exit Condition: Inherited** from **Play Development Card** use case**.**

**Use Case Name: Play Profit Interchange.**

**Participant Actor: Inherited** from **Play Development Card** use case**.**

**Flow of events:**

1. The player who is playing the turn chooses two hexagons from map.
2. The player swaps the number of hexagons.

**Entry Condition: Inherited** from **Play Development Card** use case.

**Exit Condition: Inherited** from **Play Development Card** use case.

**Use Case Name: Read Instructions.**

**Participant Actor:** The player.

**Flow of events:** The player read the instructions shown in the screen.

**Entry Condition:** The player clicks read instructions.

**Exit Condition:** The player exit the instructions sections.

**Use Case: Build**

**Participating Actor:**  Initiated by Player

**Flow of Events:**

1. Player decides on a construction type (village, city, road etc.) to build.

**Entry Conditions:**

1. It must be the turn of the player who wants to build.
2. Player must have enough source cards to build.

**Exit Conditions:**

1. Player does not want to build.
2. Player does not have enough source cards to build.

**Use Case:** **Build Civilization**

**Participating Actor:** **Inherited** from **Build** use case.

**Flow of Events:**

1. Player gives the corresponding source cards to build what he wants.
2. Player must choose a viable spot to place the construction.

**Entry Conditions: Inherited** from **Build** use case. The civilization must be built in place of a city.

**Exit Conditions: Inherited** from **Build** use case.

**Use Case:**  **Build Road**

**Participating Actor: Inherited** from **Build** use case.

**Flow of Events:**

1. Player gives the corresponding source cards to build what he wants.
2. Player must choose a viable spot to place the construction.

**Entry Conditions: Inherited** from **Build** use case.

**Exit Conditions: Inherited** from **Build** use case.

**Use Case:** **Build City**

**Participating Actor: Inherited** from **Build** use case.

**Flow of Events:**

1. Player gives the corresponding source cards to build what he wants.
2. Player must choose a viable spot to place the construction.

**Entry Conditions: Inherited** from **Build** use case. The city must be built in place of a village.

**Exit Conditions: Inherited** from **Build** use case.

**Use Case:** **Build Road**

**Participating Actor: Inherited** from **Build** use case.

**Flow of Events:**

1. Player gives the corresponding source cards to build what he wants.
2. Player must choose a viable spot to place the construction.

**Entry Conditions:**

1. **Inherited** from **Build** use case.
2. The player cannot have more than 5 villages.

**Exit Conditions:** Inherited from Build use case.

**Use Case:**  **View Price Card**

**Participating Actor:** Initiated by Player.

**Flow of Events:** Player clicks on the price card button to see the prices.

**Entry Conditions:** The player must click on the price card button.

**Exit Condition:** The player must click on the X to close the price card.

**Use Case:** **Buy Development Card(s)**

**Participating Actor:** Initiated by Player

**Flow of Events:**

1. Player wants to buy development card(s).
2. Player gives the corresponding source cards to buy development card(s).
3. Player receives development card(s).

**Entry Conditions:** Player clicks on the Buy Development Card button.

**Exit Conditions:**

1. Player does not have enough source cards to buy a development card.
2. Player cancels the buying process.

**Use Case:** **Trade Source Cards**

**Participating Actor:** Initiated by Player

**Flow of Events:**

1. Player chooses a player or the chest to trade with.
2. Player gives source cards to receive his desired source card.

**Entry Conditions:** Player clicks on the trade button.

**Exit Conditions:** Trade is done or cancelled.

**Use Case:**  **Trade with Player(s)**

**Participating Actor:** Inherited from Trade Source Card(s) use case.

**Flow of Events:**

1. Player chooses a player to trade with.
2. Players agree on what to trade.
3. Players choose the cards to trade.
4. Chosen cards are exchanged between players.

**Entry Conditions: Inherited** from **Trade Source Card(s)** use case.

**Exit Conditions: Inherited** from **Trade Source Card(s)** use case.

**Use Case: Trade with Chest**

**Participating Actor:** **Inherited** from **Trade Source Card(s)** use case.

**Flow of Events:**

1. Player chooses the chest to trade with.
2. Player places 2/3/4 of the same cards to receive one card.
3. Player receives the card.

**Entry Conditions:** Inherited from Trade Source Card(s) use case.

**Exit Conditions:** Inherited from Trade Source Card(s) use case.

**Use Case: Roll Dice**

**Participating Actor:**  Initiated by Player

**Flow of Events:**

1. The player whose turn it is, rolls the dice.

**Entry Conditions:** It must be the start of the turn.

**Exit Conditions:** Dice has been rolled.

**Use Case:**  **Place Thief**

**Participating Actor:** Communicates with Roll Dice

**Flow of Events:** ….

**Entry Conditions:** This use case extends Place Thief use case. It is initiated if result of rolling dice is 7.

**Exit Conditions:**  ….

**Use Case:**  **Start Game**

**Participating Actor:** Initiated by Player

**Flow of Events:**

1. A player clicks on the Start Game button.
2. Players name inputs are filled.

**Entry Conditions:**

1. Game has not started.
2. Player clicks on the Start Game button.

**Exit Conditions:**

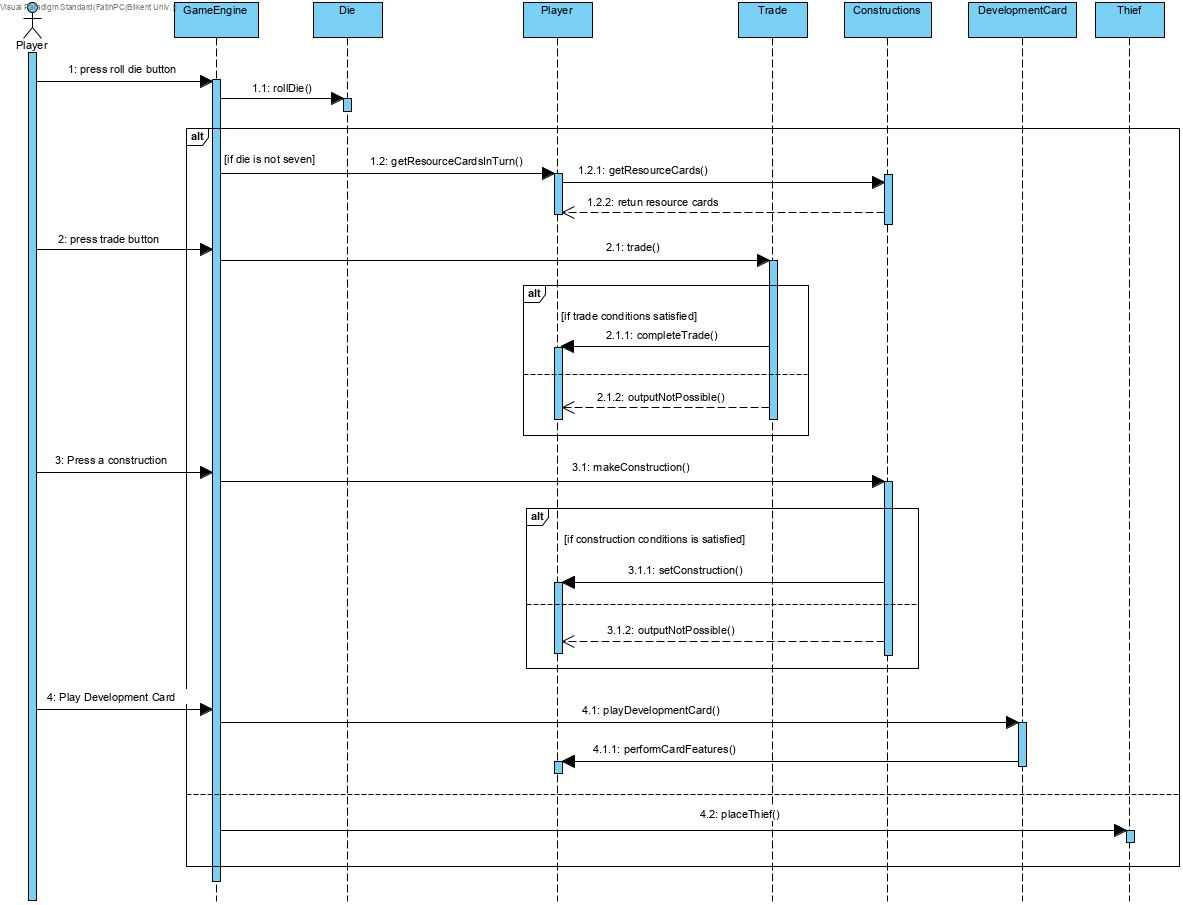
1. Game has started.

## Dynamic models

### Sequence Diagrams

#### Play Game Turn

**Scenario 1:** The game starts with pressing roll die button and the Game Engine rolls the dies. If the sum of numbers on die is not seven, the Game Engine distributes Resource Cards gained in the tour to Players, this happens by Player object’s communication with Construction object. After, the user can press trade button, press a construction, or play development cards. When the user presses trade button the Game Engine initiates a Trade object which swap resources of related Player objects if the trade condition is satisfied. If the user presses the button to build a construction, a Construction object is initiated by the player if conditions are met. When the user plays a development card, a Development Card object is initiated, and its’ effects are interacted with the Player. If the sum of die rolls is seven, Thief object is created, and the Player can place it on the map.

**** Figure 6: The sequence diagram of the Playing Game Turn that is described in Scenario 1.

#### Trade

**Scenario 2:** When the user pressed the trade button, the player chooses the trade of interest: chest or another player, the player also enters the amount of which Resource Cards he wants to trade. The Game Engine initiated a Trade object and If the trade is possible, and if the trade of interest is a player, the Trade object requests trade to the Player object and the Player object replies to Trade Request. If the trade of interest is with a Chest object, the Trade object exchanges the Resource Cards according to specified ratio. If the trade is not possible due to Player’s lack of Resource Cards, the Trade object outputs this situation to the Player and trade does not happen.

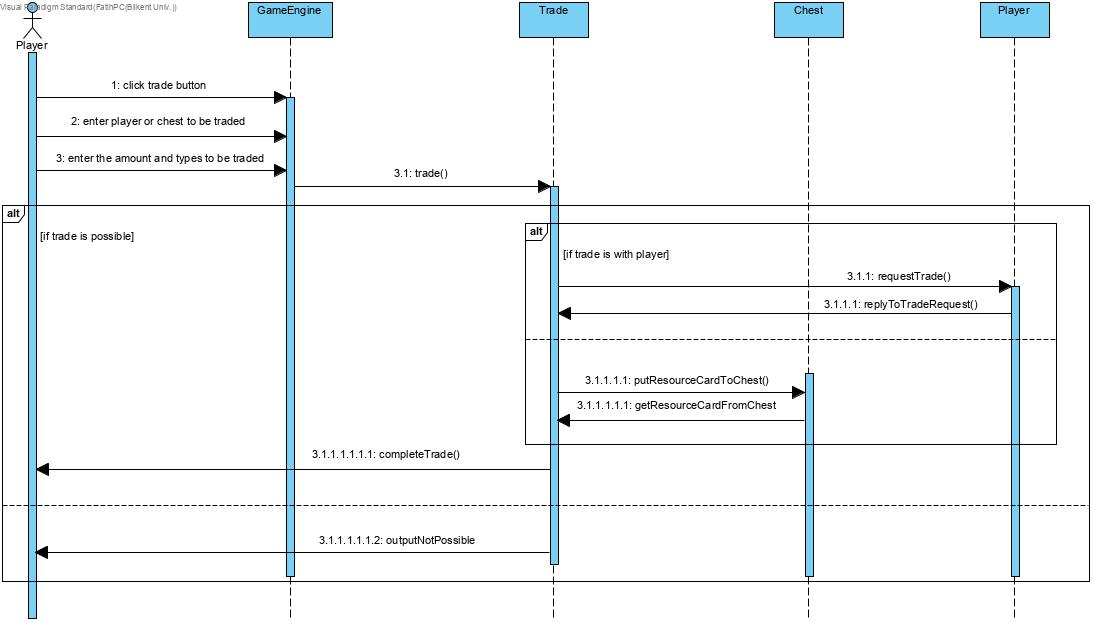


Figure 7: The sequence diagram of the Trade part that is described in Scenario 2.

#### Make Construction

**Scenario 3:** The construction making functionality initiated by the User clicking to a construction and then choosing a vertex to construct it. If the player has enough resources to build the desired construction, the Game Engine makes a Construction object and a Vertex object is set to that Construction object. Also, the Game Engine puts required Resource Cards as the cost of construction to the Chest object. If the player does not have enough Resource Cards to construct, the Game Engine outputs this situation and construction is not built.

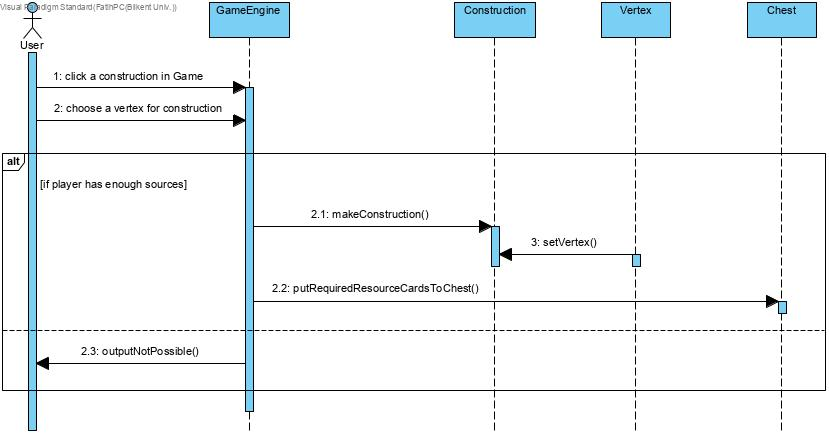


Figure 8: The sequence diagram of the Building Construction that is described in Scenario 3.

#### Change Settings

**Scenario 4:** The User opens the game and Program object is initiated, the user then presses the Settings button to activate the Change of Settings functionality. The user changes the theme from the Settings object and the Settings object changes the theme by changing images, music and card names within the Settings object. The user changes thresholds again by initiating the Settings object and this object changes the Longest Road, Victory Points and Strongest Army thresholds. The user can return to the main screen from settings and the Settings object return to the Program Object.

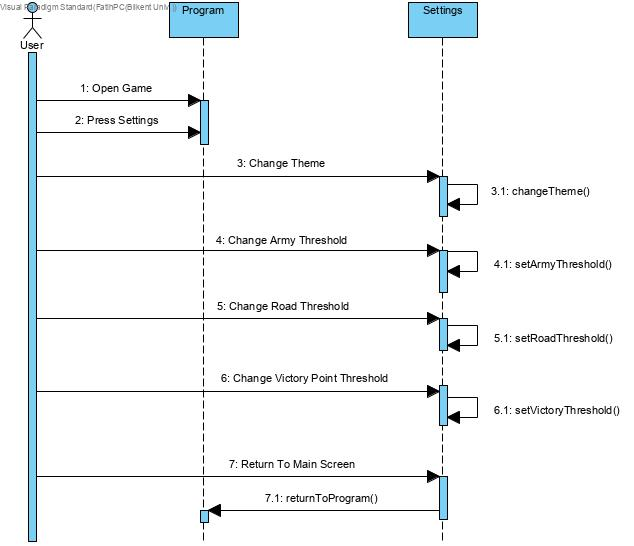
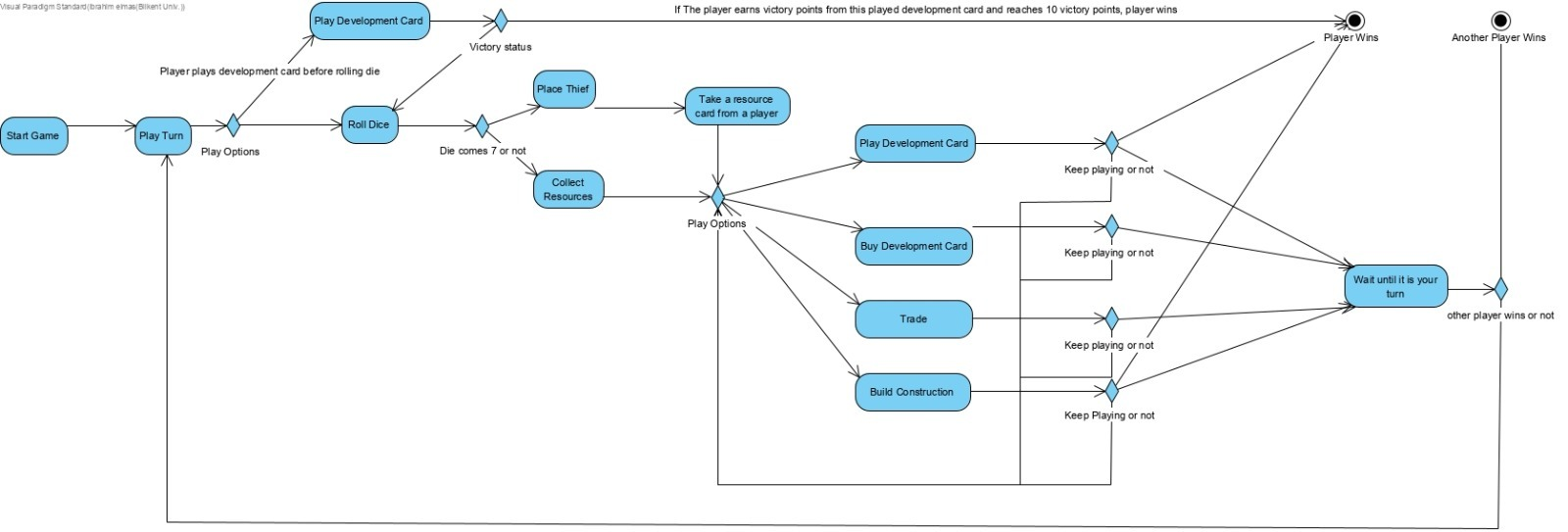


Figure 9: The sequence diagram of the Change Settings that is described in Scenario 4.

### Activity Diagrams

****Figure 10: The activity diagram of the whole Catan Game.

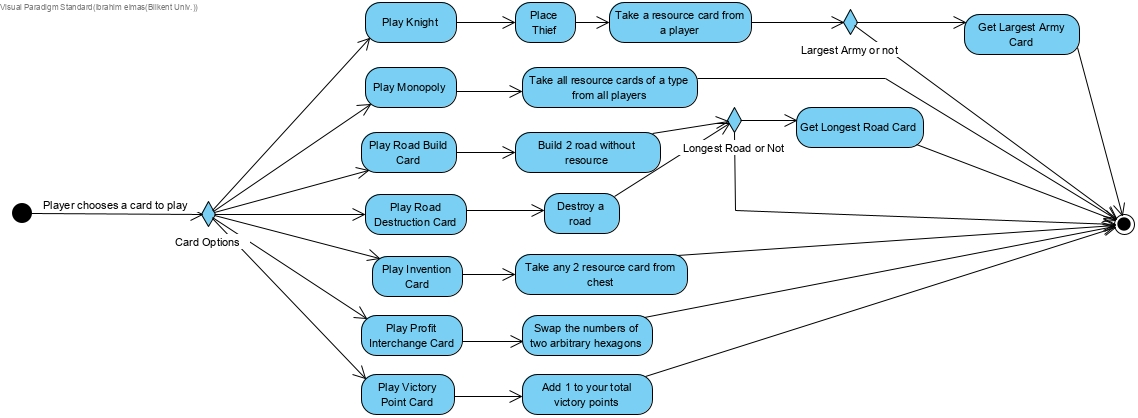
****

Figure 11: The activity diagram of the Playing Development Card process.

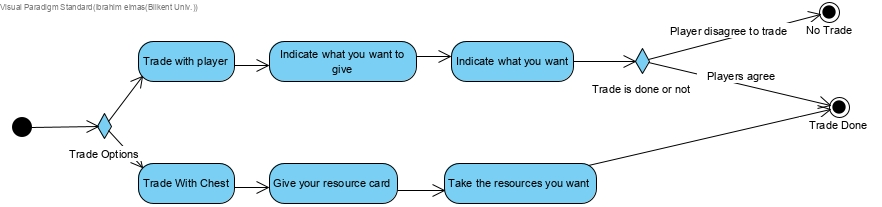
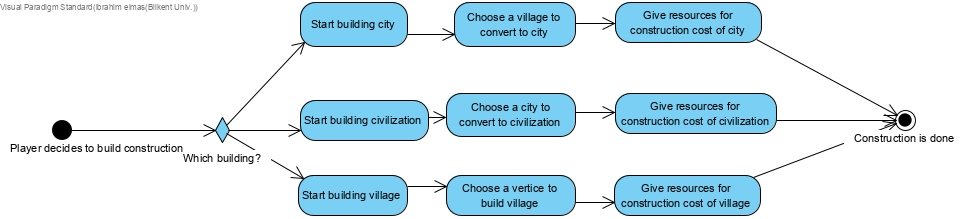
****

Figure 12: The activity diagram of the Trade process.

 Figure 13: The activity diagram of the Building Construction process.

There are 4 activity diagrams. First one shows the steps of the whole game without details like play development card options. For these details, there is another activity diagram. The other 3 activity diagrams are Play Development Card, Trade and Build Constructions activity diagrams. These 3 activity diagrams show the detailed steps which is not shown in the main activity diagram.

### State Diagrams

Figure 14: The state diagram of the Catan game.

Game logic starts with the state “Game is to Start” then proceeds to the state “Next Player’s Turn” when game is initiated. This state is the start of a loop which iterates between 4 players. Afterwards, “Roll Dice” state is entered to roll dice to determine the next step. If dice shows up 7, next state is “Place Thief” or else “Give out Source Cards”. In the state “Place Thief”, player puts the thief in his desired location. In the state “Give out Source Cards”, source cards are handed out to players accordingly. Then, player goes into a state which he can do the following; play development card, buy development card, build construction or trade. With the ending of this state, “End of Turn” state is proceeded to end the turn. Then, If player exceeds the win points threshold, game is ended by going through the state “End of Game”, or else “Next Player’s Turn” state is entered to reiterate this process for the next player.

## Object and Class Model

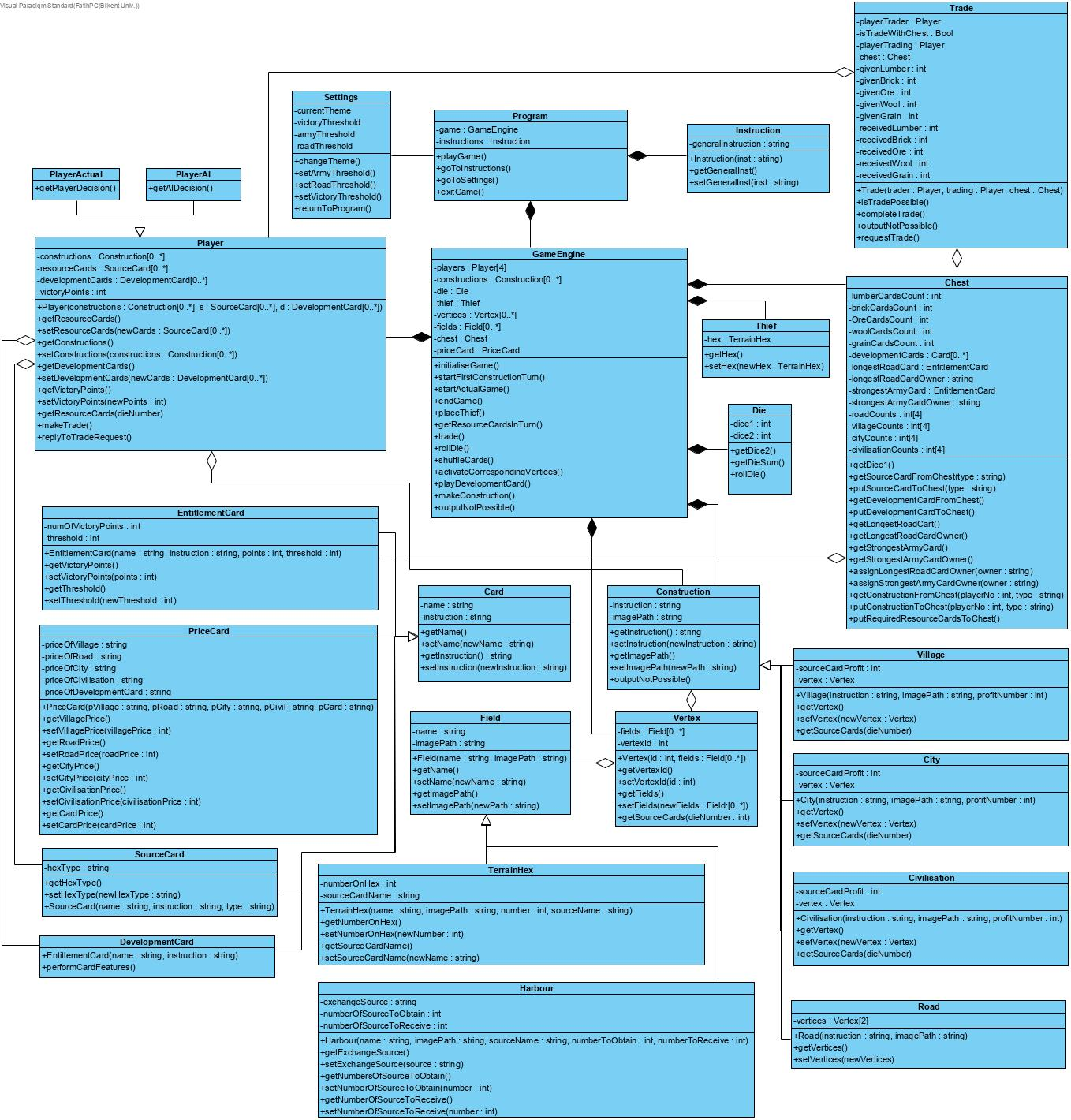
****

Figure 15: The object and class model of the Catan game.

**Player Class:** This class represents each player in the game. The properties of this class keep constructions, source cards, development cards and victory point that player has. While interacting with Game class by created its objects as properties of the Game class, player objects updates its properties and sends the information of its properties by its getter and setter methods. Moreover, the objects of this class in the Game object starts the trade interacting Trade object in the Game object.

**PlayerAI Class:** This class inherits all attributes from Player class. Decisions are made by computer and properties of the objects of this class are regulated according to these decisions.

**PlayerActual Class:** This class inherits all attributes from Player class. Decisions are made by actual players and properties of the objects of this class are regulated according to these decisions.

**Game Class:** This class is the main class of the program. The objects of all related classes are created in the object of this class declaring them as property. Game management is provided by Game class. It initializes, starts, and finishes the game.

**Chest Class:** This class represents the actual chest in the board game version. It keeps number of cards and constructions inside the chest. It updates these values while game continue by getter and setter methods through the interaction with Game class. Total number of cards have certain value. While the cards are reduced in the chest, players’ cards increase. This cards flow is provided by Game class.

**Program Class:** This class represents the entrance screen of the game. It directs user to the start game, instructions, settings and provides users to exit game.

**Settings Class:** This class lets the user change the game settings which are theme, threshold for victory, threshold for having army card and threshold for having long distance road card.

**Instructions Class:** This class lets user to how to play the game by showing the instructions.

**Trade Class:** The object of this class is activated during the game after necessary conditions are provided. After trade is happened between certain player with certain products, this class interacts with Player and Chest object which are properties of Game class to update their source information.

**Thief Class:** The object of this class is property of the Game class. It holds the certain terrain hex number which cannot produce sources due to having thief.

**Die Class:** The object of this class is created in the Game class as property. After each turn, the die is rolled by players and according to the sum of the die, Game class regulates the game.

**Card Class:** It is the class that represents the cards in the game by including the name and instructions of the cards.

**Entitlement Card Class:** This class inherits all attributes from Card class. It gives the player extra victory point when the player achieves some goals like longest road.

**Price Card Class:** This class inherits all attributes from Card class. It includes information about price of constructions and this information is placed to the corner of the game screen in the Game class.

**Source Card Class:** This class inherits all attributes from Card class. It represents the source cards that player uses it to build construction and trade. This class interact with Player class.

**Development Card Class:** This class inherits all attributes from Card class. It represents the development cards that player uses it to take some advantages to other Players. This class interact with Player class.

**Vertex Class:** Thisrepresents allvertex of terrain hex in the game. This class keeps the adjacent terrain hexes as its properties, and it is used to build constructions. This class interacts with Game class, its objects are created in the Game class as its properties.

**Construction Class:** This class represents the all constructions in the game. It includes the instruction about the construction and the image path of related construction. Its objects are created in the Game class as its properties.

**Village Class:** This class inherits all attributes from Construction class. The class represents the smallest construction in the game. This class includes the vertex that is built upon and the value of source card profit that the player gains points through the construction.

**City Class:** This class inherits all attributes from Construction class. The class represents the middle size construction in the game. This class includes the vertex that is built upon and the value of source card profit that the player gains points through the construction.

**Civilization Class:** This class inherits all attributes from Construction class. The class represents the biggest size construction in the game. This class includes the vertex that is built upon and the value of source card profit that the player gains points through the construction.

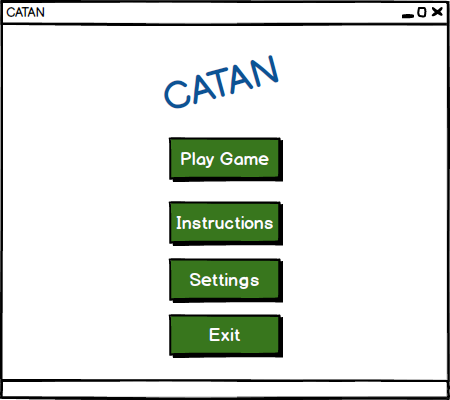
**Road Class:** This class inherits all attributes from Construction class. The class represents the roads in the game. This class includes two vertex which are the beginning and ending point of the road.

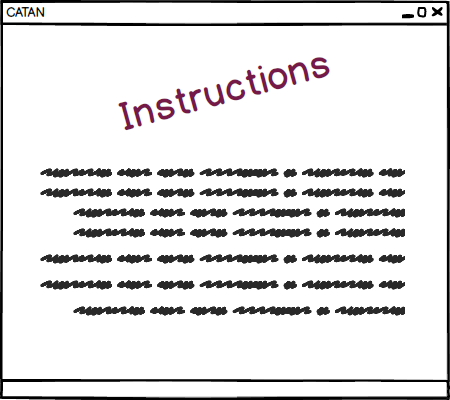
**Field Class:** This class represents the terrain hexes and harbours in the games. Its objects are created in the Game class as its property.

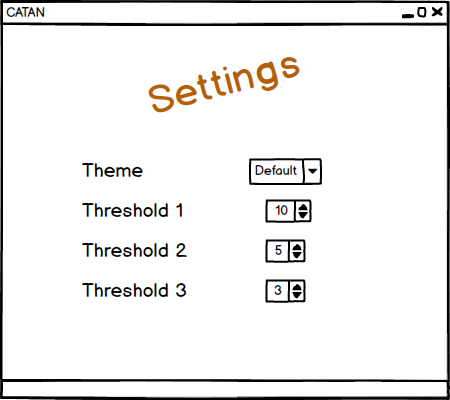
**Terrain Hex Class:** This class inherits all attributes from Field class. It keeps a value as its property that represents its number and keeps the name of which area is represented that produces source.

**Harbour Class:** This class inherits all attributes from Field class. It keeps the source name to change condition of the trade and the required number of cards to obtain and receive.

## User interface

Figure 16: The mockup of the entrance screen of the game.

Figure 17: The mockup of the instructions screen of the game.

Figure 18: The mockup of the settings screen of the game.

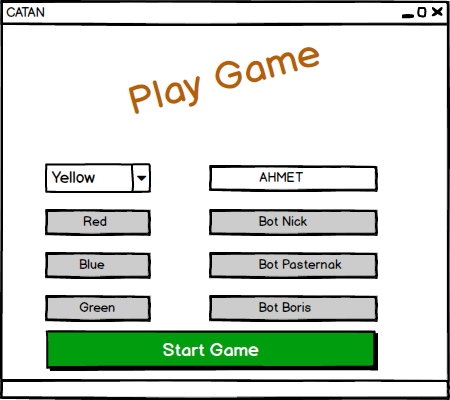


Figure 19: The mockup of the player information screen of the game.

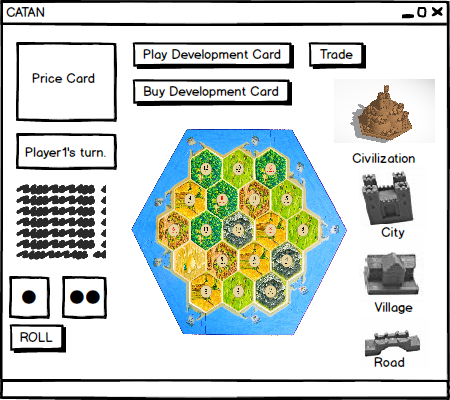
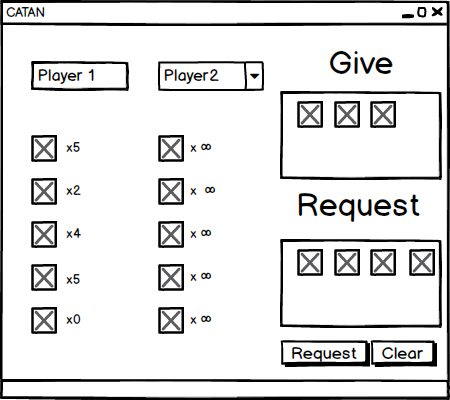
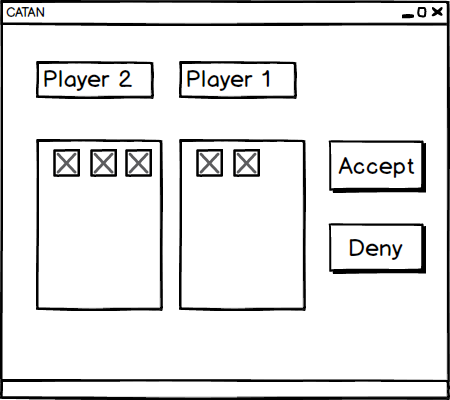


Figure 20: The mockup of the main screen of the game.

Figure 21: The mockup of the trade condition screen in the game.

Figure 22: The mockup of the trade decision screen in the game.

## Conclusion

As a result, The Settlers of the Catan is a strategy game played by 4 people where everyone plays in turns. It is single player game that user plays the game to AI which controls other 3 player. After the user enters his/her name, choose color and select the settings, the game is started. Each player builds their constructions in the first turn, then the usual flow of turns starts. In each turn, the player can roll the die to collect sources, buy or activate development cards, trade with other players or chest and build constructions. The special condition that the sum of die comes 7, then players can manipulate the thief to take some advantage. The ways of collecting points are building constructions and enhancing them such as village to city, achieving the longest road and strongest army cards and buying victory points development card. The player can win the game by reaching the victory points which is determined at the beginning of the game.

1. **Glossary & References**

https://www.catan.com/service/game-rules

https://www.catan.com/game/catan-universe