BILKENT UNIVERSITY

Department Of Computer Science

CS319 - Object-Oriented Software Engineering

Analysis Report

Game: Terra Mystica

Group: 1E

Berdan Akyürek 21600904

Ege Hakan Karaağaç 21702767

Ömer Olkun 21100999

Aziz Ozan Azizoğlu 21401701

Fırat Yönak 21601931

1.Introduction

Terra Mystica is a board game without any luck that rewards strategic planning. The game can be played by two to five players. Each player governs one of fourteen factions trying to develop more successfully than their opponents. Terra Mystica is a magical world: its inhabitants are able to transform the terrain they are living in. Each faction is bound to a specific type of terrain:
It can only build structures on its “Home terrain”. This is why each faction over the centuries needed to develop Terraforming capability.

2.Overview

2.1 Goal of the Game

At the end, the player with the most victory points wins the game. There is a scoring tile for each rounds. Each of them displays how to get victory points in a round via building dwellings, trading houses, the stronghold or the sanctuary. Additional victory points may be awarded for transforming terrain spaces and founding towns.

Three out of the nine bonus cards award victory points when player has completed all the actions player wish to take in the current action phase:
Then, victory points are either awarded for all of your dwellings, all of trading houses, stronghold and sanctuary on the game board.

The game consists of six rounds and each round goes through three phases.

2.2 Overview of a Round

In each round, players will have three phases to execute specific actions. Phases are as follows:

First Phase: Income
Second Phase: Actions
Third Phase: Cult Bonuses and Clean-up

2.2.1 First Phase: Income

In the first round, players first collect character-specific starting income and second for the first round. Players' income depends on the structures that they have built, on the favor tiles they own. On all components, income is shown by an open hand.

2.2.1.1 Basic Income for Structures

Basic income for structures is as follows:

Workers: Take the number of workers from the general supply equal to the number of visible worker symbols on your **dwelling** track.

Coins: Take the number of coins from the general supply equal to the number of visible coin symbols on your **trading house** track.

Priests: Take the number of priests from your supply equal to the number of visible priest symbols on your **temple** track. Another priest symbol will be appeared once you have built your **sanctuary**.

Power: Power is a gained due to the visible power symbols on the **trading house** track. The **stronghold** usually provides additional power.

2.2.1.2 Additional Income for Cards and Tiles

Beside the income collected by **faction** board, add the income depicted on your onus card and **favor** tiles.

The bowls of power: Each player has 12 power tokens that are distributed in three bowls. Power actions require power tokens in Bowl 3. Whenever a player gains power in the game, player does not get new power tokens, but rather move the existing ones from one Bowl to another. Also, when player spends power on an action, player does not lose Power tokens, but moves them around.

Power tokens move according to the following rules:

1. If there are power tokens in Bowl 1 for each Power player gains move one token from Bowl 1 to Bowl 2.
2. Once Bowl 1 is empty, for each power, player gains move one token from Bowl 2 to Bowl 3.
3. Once all power tokens are in Bowl 3, player cannot gain additional power.

2.2.2 Second Phase: Actions

In the second phase, beginning with the starting player and in clockwise order, each player takes exactly one action. It continues until no player wants to take any more actions. There are 8 possible actions to choose from. All actions can be taken multiple times during the same phase.

2.2.2.1 Action #1: Transform and Build

First player may change the type of one terrain space. Then, if player has changed its type to home terrain, player may immediately build a dwelling on that space.

2.2.2.2 Action #2: Advancing on the Shipping Track

Terrain spaces and structures need to be indirectly adjacent to one another if player wants to expand beyond River spaces. In order to transform Terrain spaces or build structures beyond river spaces, as an action, player may move the marker on on shipping track forward one space. Then, faction board displays the cost of this action: 1 priest and 4 coins.
As a reward for taking this action, player gets a number of victory points as indicated on the shipping space that moved to.

2.2.2.3 Action #3: Advancing on the Shipping Track

At the beginning of the game, spades cost 3 workers each. In order to reduce cost down to 2 workers each or less, as an action, player may move the marker on the exchange track up one space. Then, faction board displays the cost of this action: 2 workers, 5 coins, 1 priest.
As a reward for taking this action, player gets 6 victory points.

2.2.2.4 Action #4: Upgrading a Structure

The cost for each upgrade are shown on the faction board to the left of a structure. Upgrading costs are as follows:

Dwelling House > Trading House: Upgrading a dwelling house to trading house costs 2 workers and 6 coins. If there is at least one opponent's structure directly adjacent to that dwelling house, player only needs to pay 3 coins instead of 6 coins. If the current scoring tile depicts a trading house, player gets 3 victory points.

Trading House > Stronghold: Upgrading a trading house to stronghold costs a number of workers and coins depending on the faction. After building stronghold, each faction gains a

specific special ability. If the current scoring tile depicts a stronghold, player gets 5 victory points.

Trading House > Temple: Upgrading a trading house to temple costs 2 workers and 5 coins. Player chooses and takes one favor tile as a reward.

Temple > Sanctuary: Upgrading a temple to sanctuary costs a number of workers and coins depending on the faction. Player chooses and takes one favor tile as a reward. If the current scoring tile depicts a sanctuary, player gets 5 victory points.

2.2.2.5 Action #5: Sending a Priest to the Order of a Cult

There are 4 spaces below each of the cult tracks of Fire, Water, Earth, and Air that each can hold exactly 1 priest. As an action, player may place one of priests on one such space to advance 3 or 2 spaces on the corresponding cult track.

2.2.2.6 Action #6: Power Actions

There are two types of power actions: fully-fledged actions and auxiliary actions. The power actions on the game board may only be taken once per round. Whenever taking one of these actions, move a number of power tokens from Bowl 3 to Bowl 1 equal to its indicated cost. Then, put an action token on its space on the game board to indicate that action may not be taken any more in a specific round.

2.2.3 Third Phase: Cult Bonuses and Clean-up

When all players have passed, the current action phase is over. In rounds 1 to 5, the action phase is followed by a Clean-up phase preparing the next round.

Cult Bonuses: First, the cult bonuses depicted on the current scoring tile need to be awarded. Each player with enough progress in the depicted cults gets the depicted reward, multiple times if need be.

2.3 End of the Game and Final Scoring

Cult Scoring: Score each of the four cult tracks individually:
• 8 Victory points for the player highest on a track.
• 4 Victory points for the player second highest on a track.
• 2 Victory points for the player third highest on a track.

These values are also depicted in the top left corner of the game board. You cannot gain any victory points when on space 0.
In case of a tie, evenly divide the victory points of the respective tiers among the tied players.
Area Scoring: Determine the number of your Structures directly or indirectly adjacent to one another. With the proper shipping value, scattered areas may be indirectly adjacent to one another.
• The player with the highest number of connected structures gets 18 victory points.
• The player with the second highest number of connected structures gets 12 victory points.
• The player with the third highest number of connected structures gets 6 victory points.
These values are also depicted in the top left corner of the game board. In case of a tie, evenly divide the victory points of the respective tiers among the tied players.

Resource Scoring: Finally, get victory points for left-over resources: get 1 victory point per 3 coins.

3.Requirements

3.1 Functional Requirements

- Player shall be able to play game.
- Players shall be able to select the number of players from two to five.
- Players should type their names.
- Players can select their faction from fourteen different kinds.
- At the end of the game, result table should be available for players.
- Player should be able to open How To Play page from the main menu or in game.
- Player should be able to open Credits page from the main menü.
- Player shall quit from the game from main menu via quit button.
- During game play, players shall be able to see their opponents' current situations.
- During a round, a player should be able to execute three phases.
- Every round, situations of players should be updated.
- Players will execute their actions in second phase as long as none of them wants to continue.
- Features of each factions should be specialized for every one of them and players shall be able to learn these features.
- At the end of each round, score table should be available.

3.2 Non-Functional Requirements

User Experience through UI: Graphics of game is the first thing which players care about a game. Because of that reason the UI of the game will be nice and easily understandable to make users experience more pleasant through game.

Performance: The performance of the game is the most significant thing for players that's why this requirement will be considered during the implementation process to make game more attractive. The game will give response as soon as possible and be fast not to wait players.

Extendibility: The game will be implemented extendible to improve its' feature after implementation process. With extendible design, new additions or alterations can be adapted easily without any huge changes on other parts of code.

4. System Models

4.1 Use Case Model

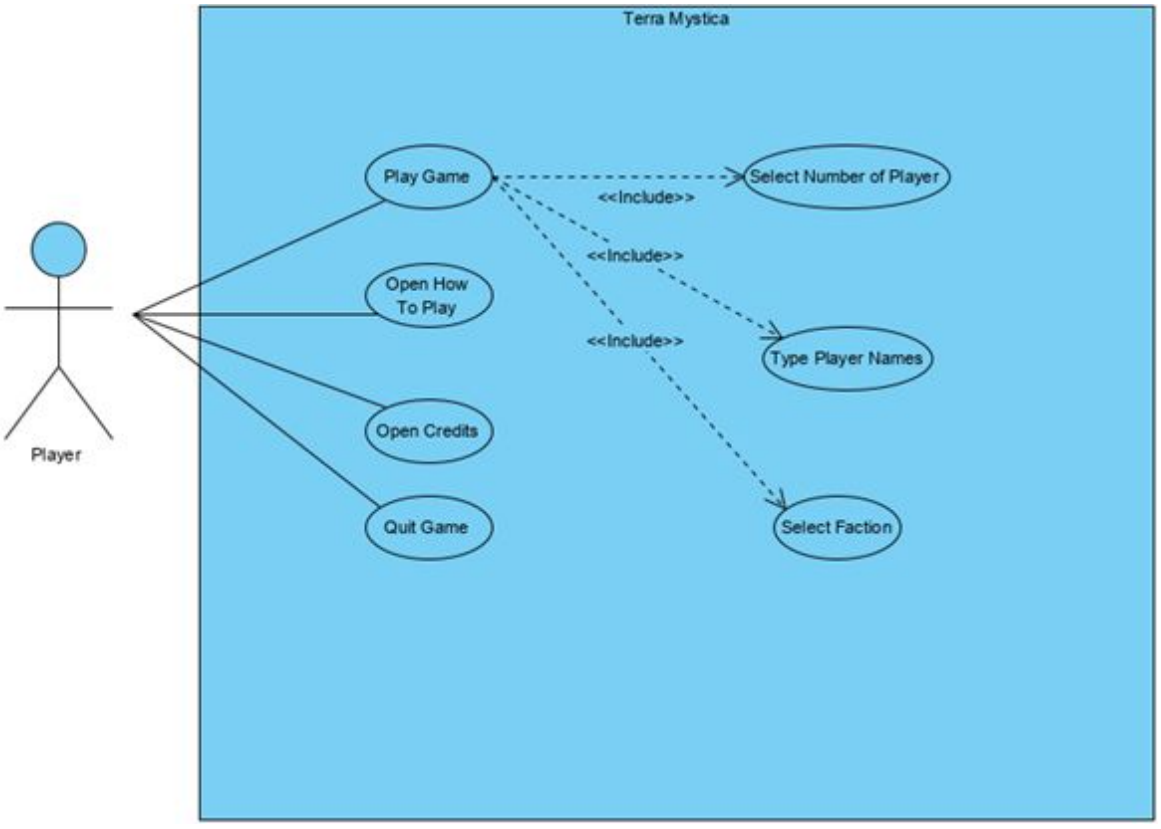

Figure 1:Use case model of Terra Mystica.

4.1.1 Use Case Descriptions

Use Case 1:

1. Name: Play Game

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Player wants to play the game.

4. Entry Condition:

- Player is on main menu screen.
- Player press play game button.

5. Exit Condition:

- Selection page opens.

  OR

- Player exit from game.

6. Flow of Events:

1. Play Game button is pressed by a player.
2. System opens the selection screen.

Use Case 2:

1. Name: Open How to Play

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Main menu page is open.
- Player wants to learn or remember the rules and gameplay.

  OR

- Player is in the game.
- Player wants to learn or remember the rules and gameplay.

4. Entry Condition:

- Player presses “How To Play” button..

5. Exit Condition:

- Player closes the “How to Play” screen.

6. Flow of Events:

1. System displays how to play page.
2. Player see the gameplay and the rules of the game.

Use Case 3:

1. Name: Open Credits

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Main menu page is open.
- Player wants to learn or remember the rules and gameplay.

4. Entry Condition:

- Player presses “Credits” button..

5. Exit Condition:

- Player closes the “Credits” screen.

6. Flow of Events:

1. System displays how to play page.
2. Player see the gameplay and the rules of the game.

Use Case 4:

1. Name: Quit Game

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Player wants to quit game.
- The Main Menu is open.

OR

- Player is in the game.

4. Entry Condition:

- Player selects the “Quit Game” option.

OR

- Player presses exit button from screen.

5. Exit Condition:

- Player will select “Quit Game” option.

OR

- Player will click exit button.

6. Flow of Events:

1. System notice the request of player .
2. System close the game.

Use Case 5:

1. Name: Select Number of Players

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.

- Player has selected the “Play Game” option from main menu.

4. Entry Condition:

- Player is on selection screen.

5. Exit Condition:

- Player moves other parts of selection screen.

  OR

- Player closes the game.

6. Flow of Events:

1. Player chooses one of the option from two to five.
2. System records it.

Use Case 6:

1. Name: Type Player Names

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Player has selected the “Play Game” option from main menu.

4. Entry Condition:

- Player is on selection screen.

5. Exit Condition:

- Player moves faction selection.

  OR

- Player closes the game.

6. Flow of Events:

1. Player type a name.
2. System records it.

Use Case 7:

1. Name: Select Faction

2. Participating Actor: Player

3. Pre-Condition:

- Player has launched the game.
- Player has selected the “Play Game” option from main menu.

4. Entry Condition:

- Player is on selection screen..

5. Exit Condition:

- Player starts game.

  OR

- Player closes the game

6. Flow of Events:

1. Player selects a faction.
2. System records it.

4.2 Dynamic Models

4.2.1 System Activities

4.2.1.1 Main Menu and Selection Screen Usage

This figure shows the possible process from opening the game to start game. When game is opened, main menu appears first. Using this system can take input to open How To Play or Credits page, quit game and open selection screen game. When player chooses to play game, system also receives input in order to arrange factions, number of players and player names.

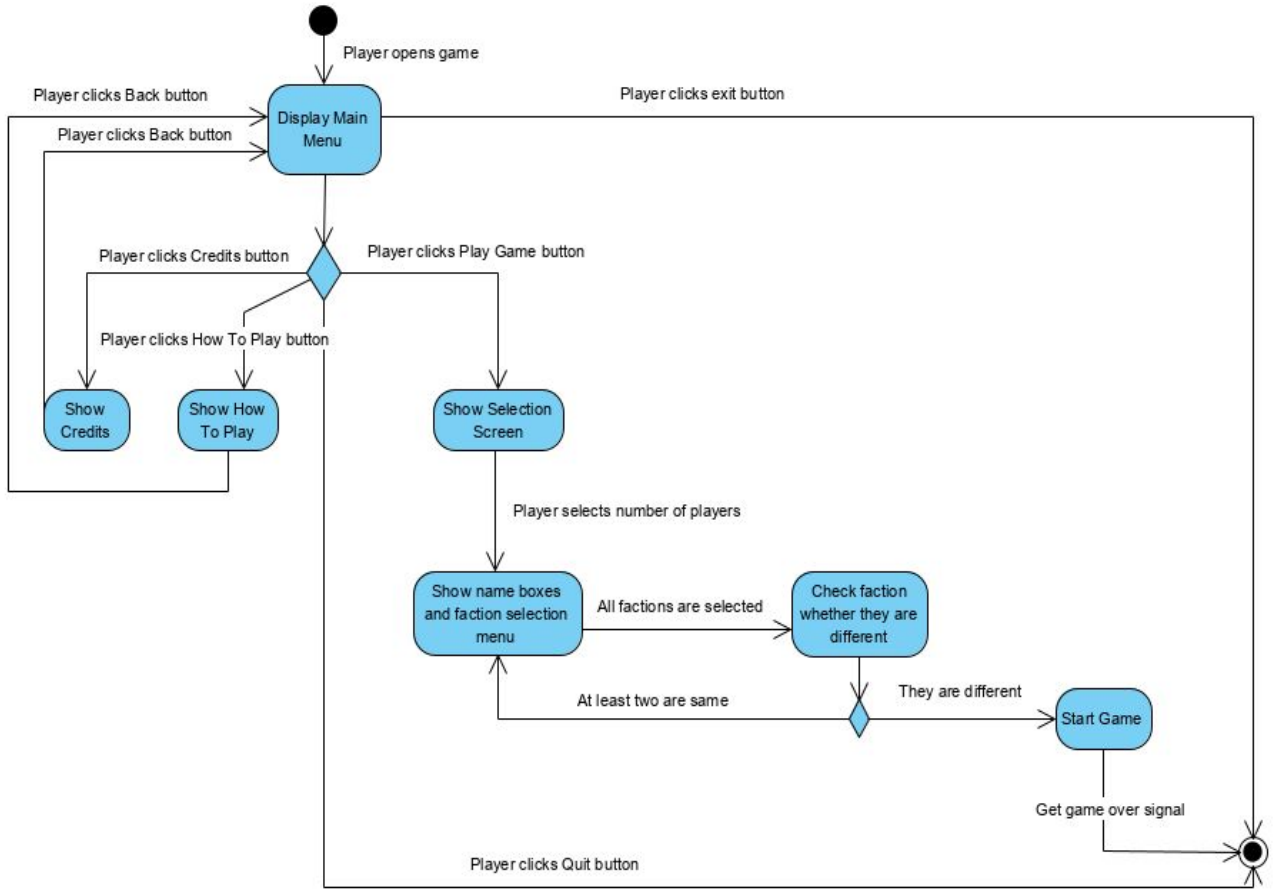

Figure 2: Activity diagram of main menu and selection screen functionalities.

4.2.2.2 Final Points Calculation

This figure shows how system calculates the final points of players. When the game is over, system starts to calculate victory points. Calculation happens according to cult score, total remaining coins and longest connection of players.

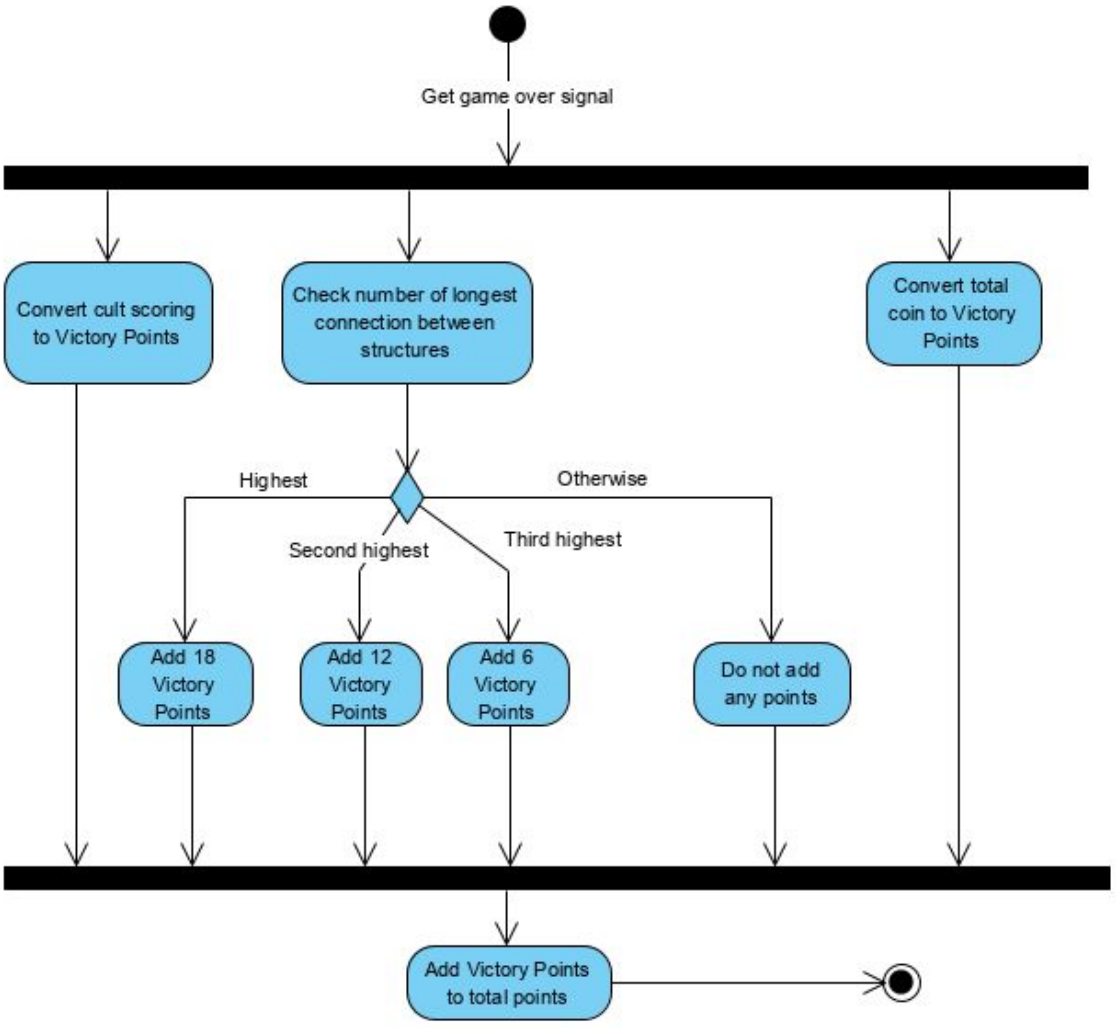

Figure 3: Activity diagram of calculating final points of players

4.2.2 Sequential Methods

4.2.2.1 Open Selection Screen

In this stage, the user is in the menu. By a left click, the program goes to Game Manager which is the component of the design to be responsible for mandatory processes in the game. After that space, according to the mouse action display screen is opened. displaySelectionScreen() enables the user to select starts to game creation. SelectionScreen component responses to Display component. After all processes, program is updated by using uptadeScreen().

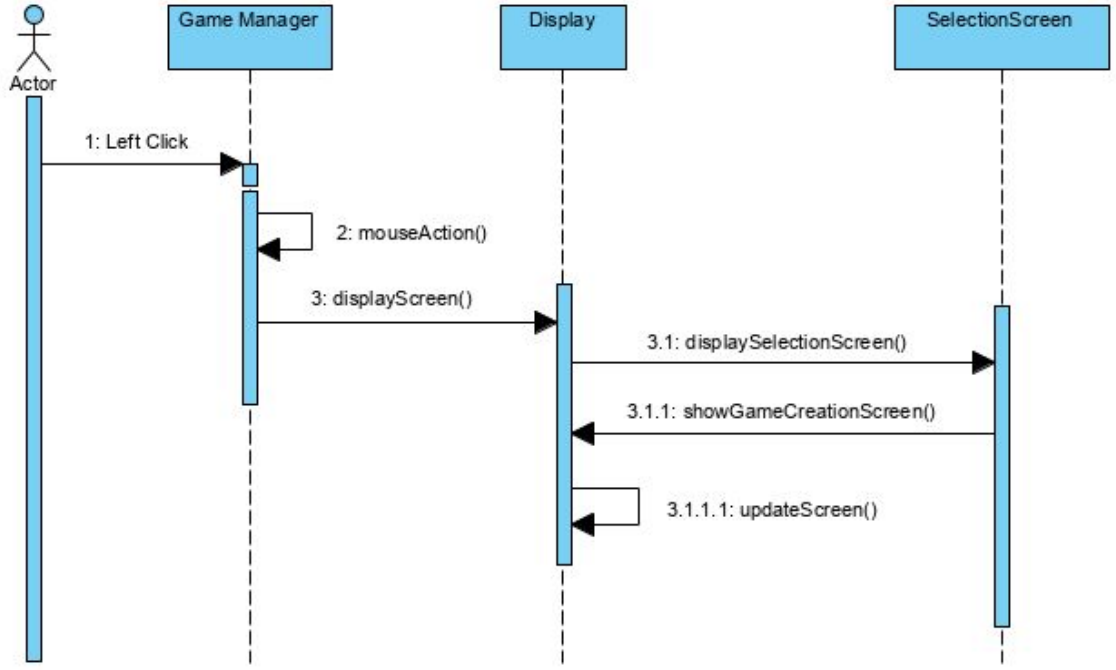

Figure 4: Sequence diagram of open selection screen

4.2.2.2 Open Game

Game Manager component automatically creates the objects which are BonusCards, ScoringTiles, TownTiles, FavorTiles, and Faction. That process is over by updating the screen in the action of updateScreen(). Display component send messages to display the menu via displayMenu() and after Menu component responses to Display. After updating the screen by updateScreen() the process is over.

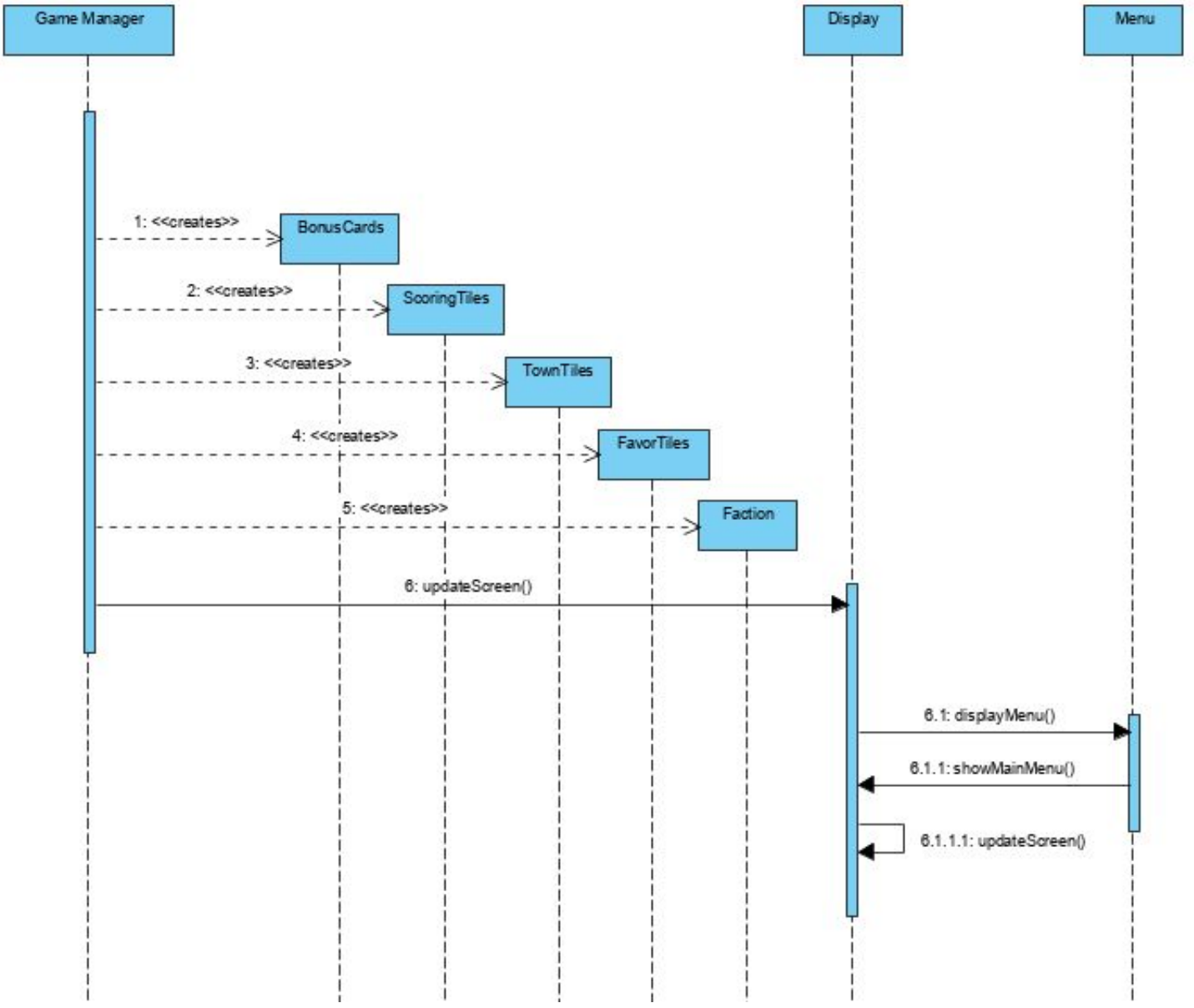

Figure 5: Sequence diagram of opening game

4.2.2.3 Start Game

In the SelectionScreen component the user left clicks and Game Manager is ready to check the next mouse action. According to the mouse action, Game Manager sends messages to Game Play component. In this stage, there is a loop that creates Player objects and updates them as their accordingly faction and makes their level on Cult class. After the loop ends, Game Manager screen updates the screen by updateScreen() function. Display class closes selection screen and sends a request to GameScreen class by displayGameScreen() function. This class uses PlayerTab and Map classes to output the game screen. At last, display updates itself.

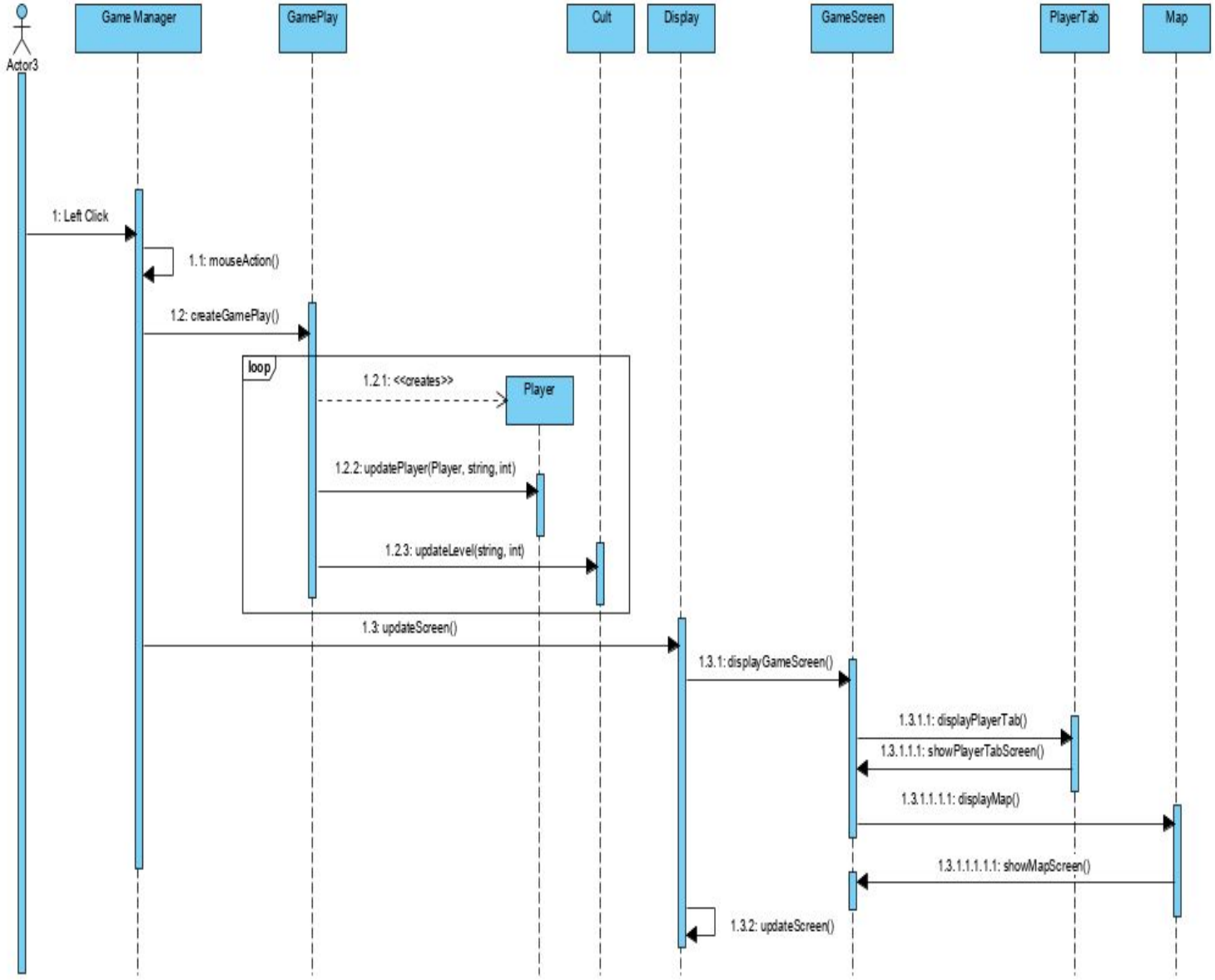

Figure 6: Sequence diagram of starting game

4.3 Class Model

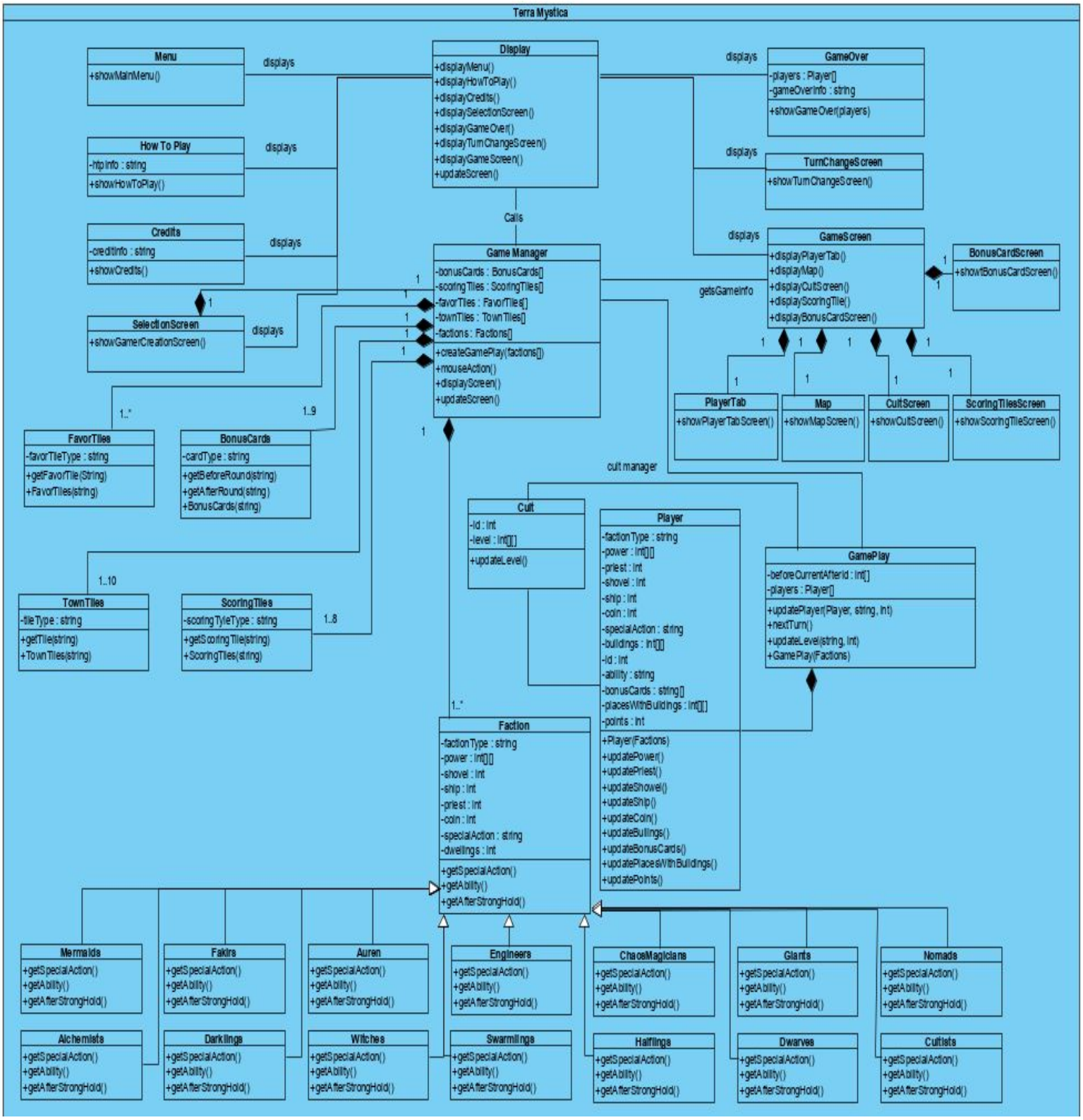

Figure 7: Class diagram of Terra Mystica

4.4 User Interface

The purpose of figure 8 to show what players will see after open the game. Players can see that there is four different option.

Figure 8: Mockup of Main menu.

The purpose of figure 9 is to make users familiar with game properties menu. After they prefer to click play game button. There is 3 different actions which are selecting number of player, faction and typing players' names. With continue button players can move on to game.

Figure 9: Mockup of game properties menu. It appears before game starts.

The purpose of figure 10 is to make users familiar with credits.

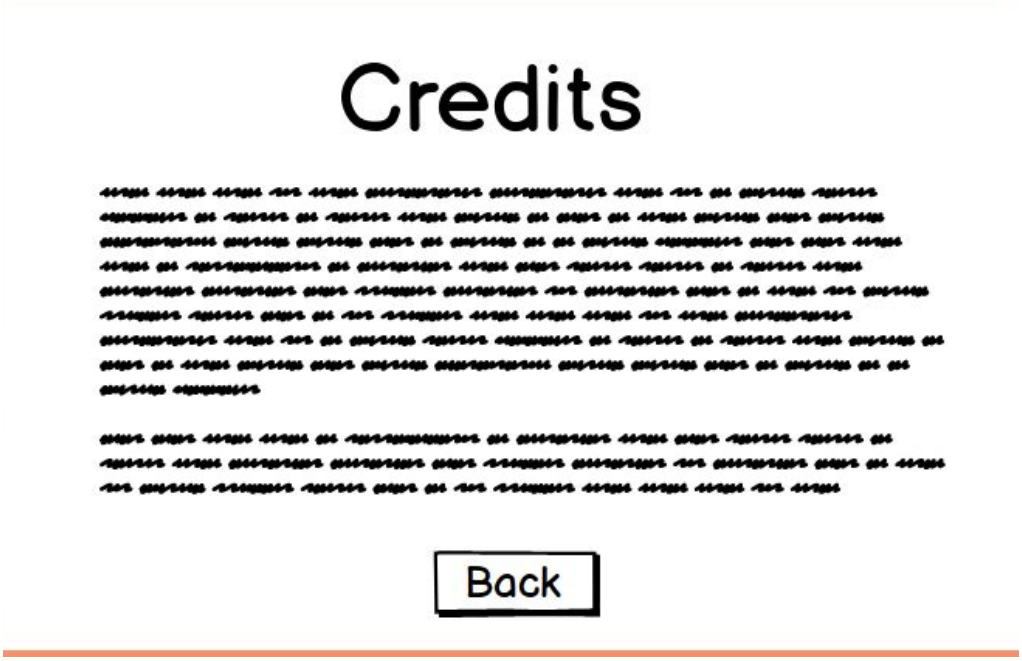

Figure 10: Mockup of Credits screen.

The purpose of figure 10 is to make users familiar with credits.

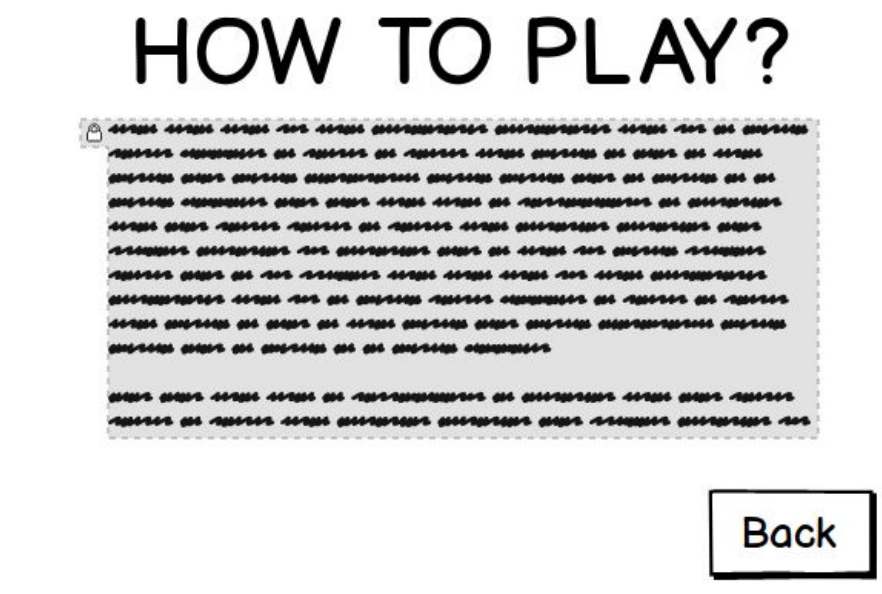

Figure 11: Mockup of How To Play screen.

Figure 12 shows the basic view of game screen of Terra Mystica. The purpose is to make users familiar with game before playing it. Left part of the screen displays the current situation of current player. Middle part displays the map.

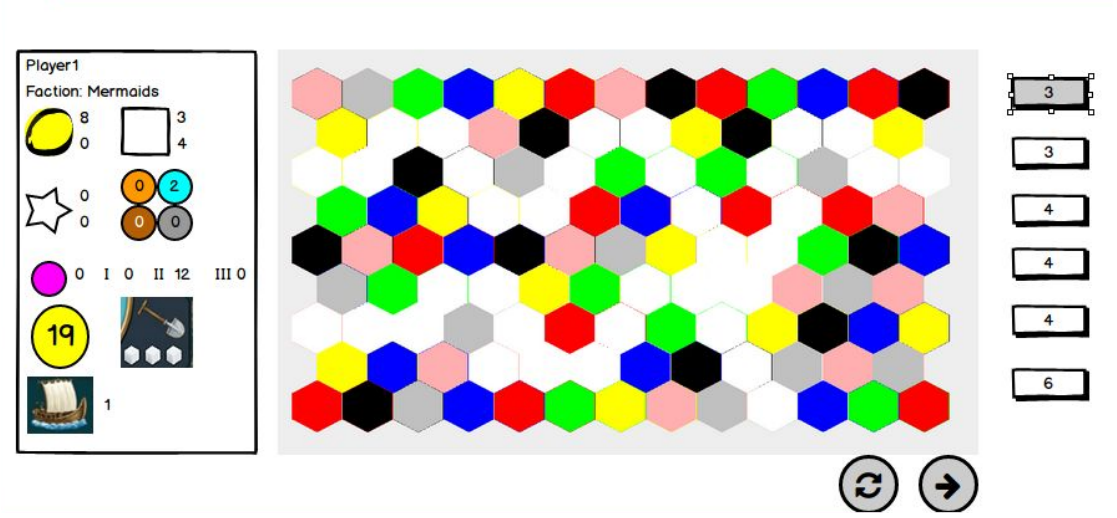

Figure 12: Mockup of main gameplay screen.

The purpose of figure 13 is to display basic view of cult screen.

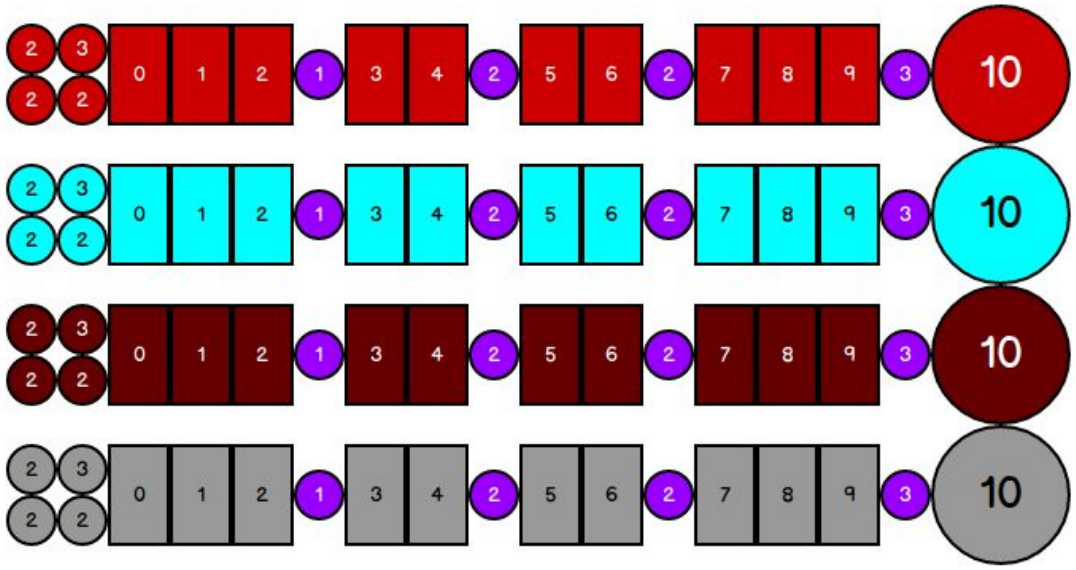

Figure 13: Mockup of Cult screen.

Figure 14 displays what players will see when they want to pick a bonus card.

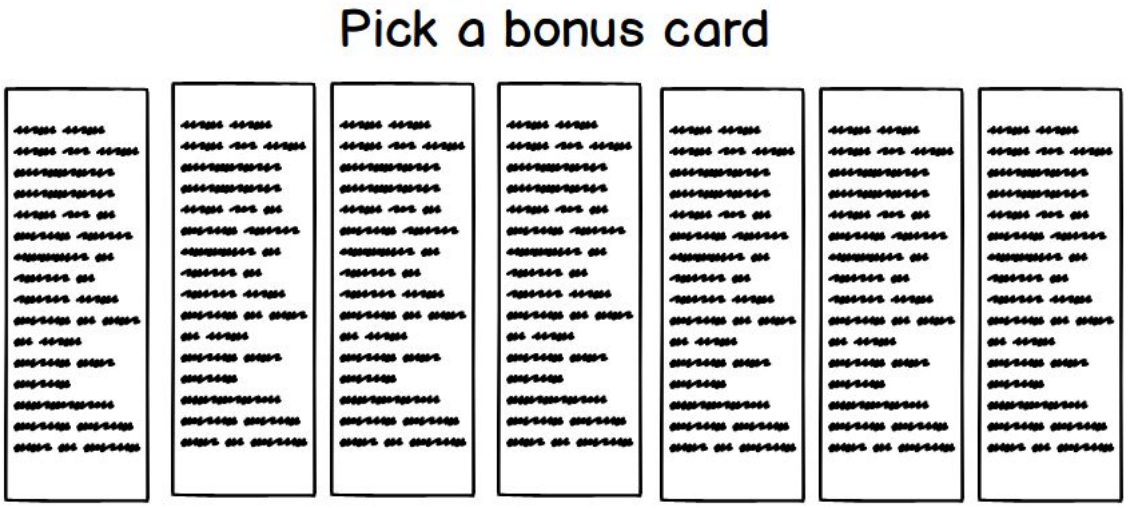

Figure 14: Mockup of bonus card selection screen.

Figure 15 displays what players will see when they want to pick a bonus card.

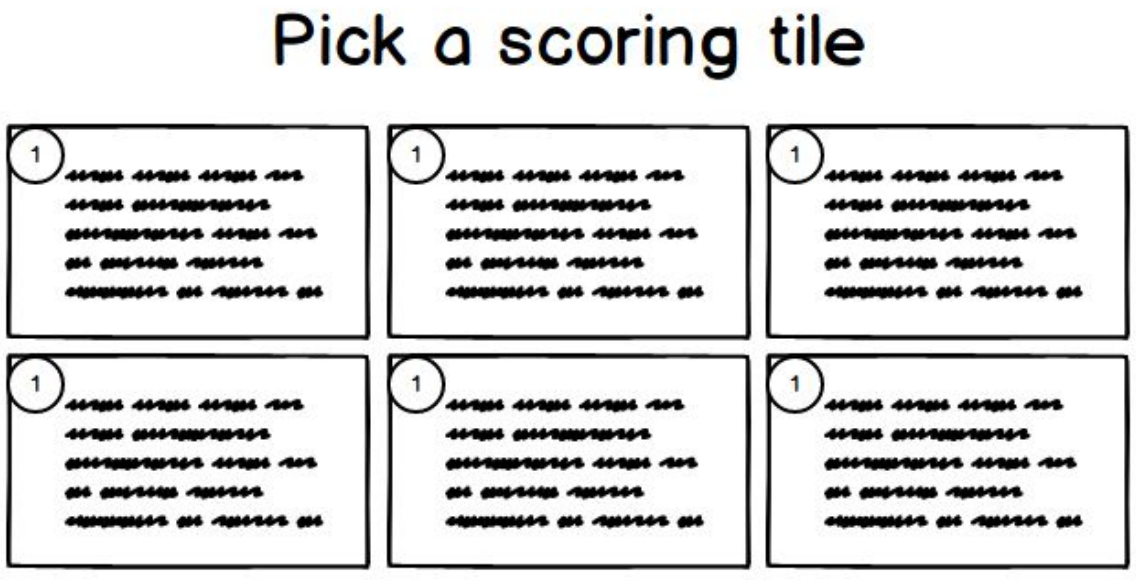

Figure 15: Mockup of scoring tile selection screen.

The purpose of figure 16 is to make familiar players with game over screen. When the game is end, they will see results of the game. They can also go directly to main menu with using button.

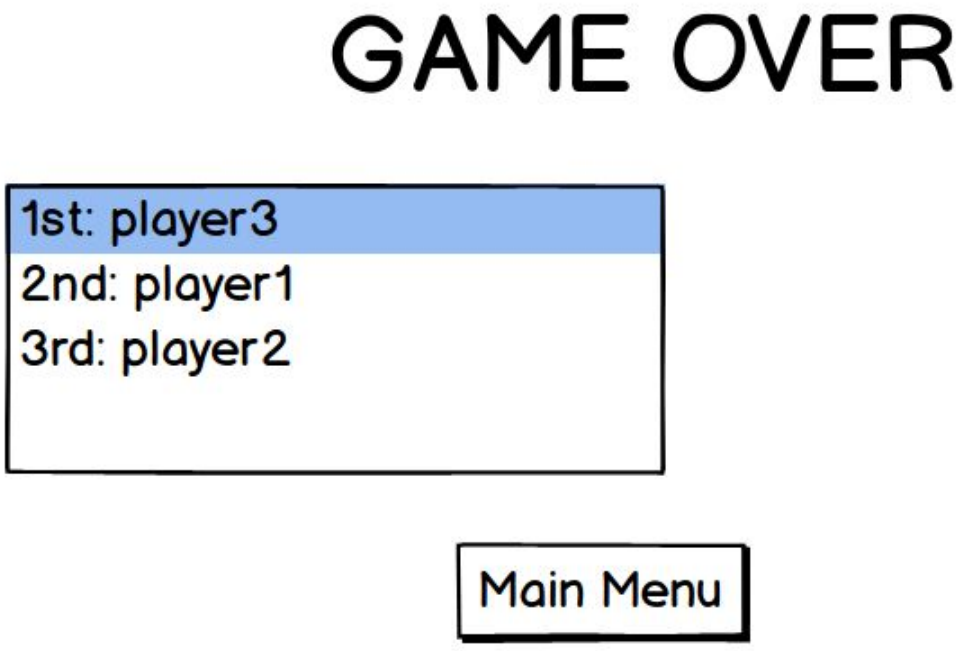

Figure 16: Mockup of game over screen.