Department of Computer Engineering

Bilkent University

CS319 Term Project

Section 1

Group 1D - Risk Takers: Risk Board Game

Analysis Report

1. Esad Burak ALTINYAZAR
2. Burak YENI
3. Burak MUTLU
4. Yigit Kutay GULBEN
5. Nurlan FARZALIYEV
6. Anar HUSEYNOV

Supervisor: Eray TUZUN

Contents

1. Introduction

Risk is a strategy board game of diplomacy, conflict and conquest for two to six players. The standard version is played on a board depicting a political map of Earth, divided into forty-two territories, which are grouped into six continents. Turn rotates among players who control armies of playing pieces with which they attempt to capture territories from other players, with results determined by dice rolls. Players may form and dissolve alliances during the course of the game. The goal of the game is to occupy every territory on the board and in doing so, eliminate the other players.

In our version of the game, this game will have some different features compared to the real one.

At the moment we have the features mentioned below:

⇒ Single Player Mode (with AI)

⇒ Multiplayer Mode

⇒ Animated Graphics

⇒ Background Music

⇒ Different Languages

⇒ In-game Menu

When the game is about to begin, the parts of map, territories, will be distributed among players randomly such that each with a player. The purpose of a player is to invade all territories to win the game. The game is mouse controlled. For each player, there are 3 phases (stages). Which are Draft, Attack and Fortify phase? In each one of these phases, the player has some functionalities in his hand. For example, increase number of players, move players, attack to other territories and etc. Phases will be talked about more detailed later.

During the game player is provided with an in-game menu, so he/she does not need to go to the main menu in order to make changes in settings, like sound, language and etc.

We are using Java to implement the game since it is the most familiar language to all group members and provides us with object-oriented programming

design. Our aim during the implementation of the game is to implement it in accordance with the principles that we have seen in class.

2. Overview

2.1 Gameplay

We will provide users with 2 game modes Singleplayer and Multiplayer. There is an AI for the “Single player” mode, the player plays with AI basically. In general, the player has his/her own territories at the beginning of the game, the rest territories to the AI, controlled by computer. In Multiplayer mode the idea is the same. There will be two to six players that can play the game at the same time in accordance with their turns. Each player in his/her own turn can add units to a specified territory by clicking on the territory and using add unit button. Then, the player can attack the territories which are in the border with the selected territory of the player’s own, and if the selected territory of the player has more than or equal to 2 players. The aim of the player is to capture all the territories at the end of the game in order to win.

2.2 Players’ Initial State

The player is the most fundamental component of the game since the game is not a game without a player. Each player has a unique colour to symbolize their own territories. As stated before at the beginning of the game each player owns some territories where he/she can do some actions, such as add unit.

2.3 Map

In this game, there is only one map (*Figure 1*). The map is also animated. For example, after the play game is pressed and game mode is chosen the game is started withdrawing the map animation wise and the territories are distributed in between players. When a player chooses to attack to another territory, with the usage of animation both territories are driven to the middle of the screen and game continues until the player captures or stop the capturing or cancels the fight.

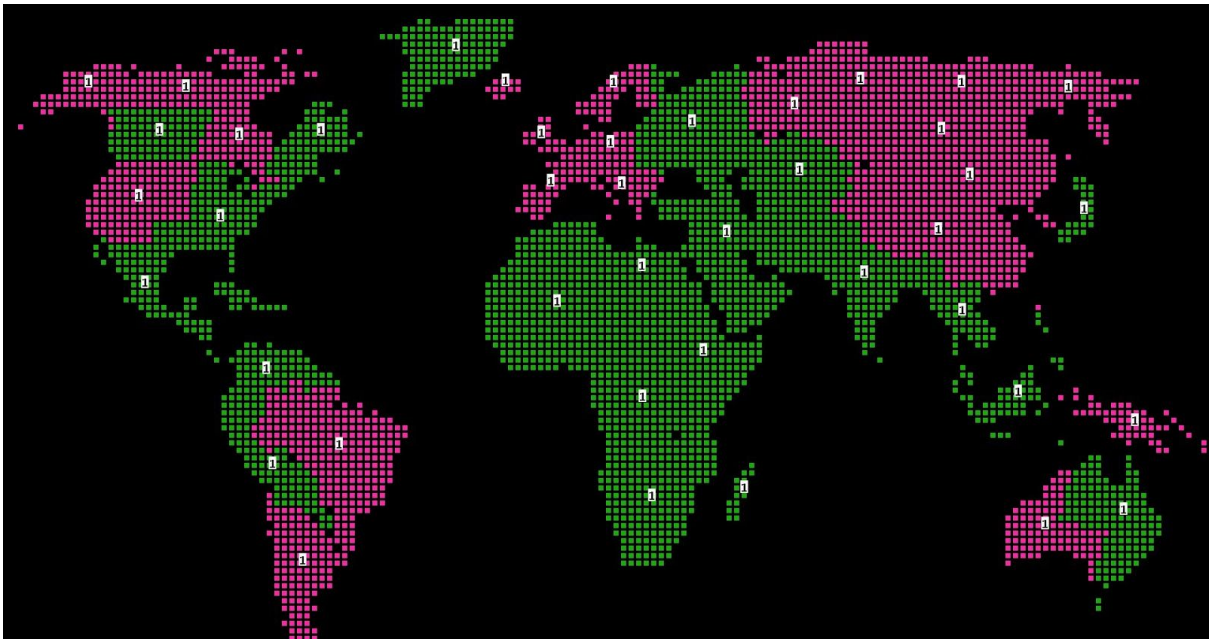


Figure 1

2.4 Territory

The Territory represents the region (*Figure 2*). Each territory has its own card and owner. Territories can be invaded by other players. Each territory possesses some amount of units more than zero. Some actions over territories constrained with the border connections with other territories.

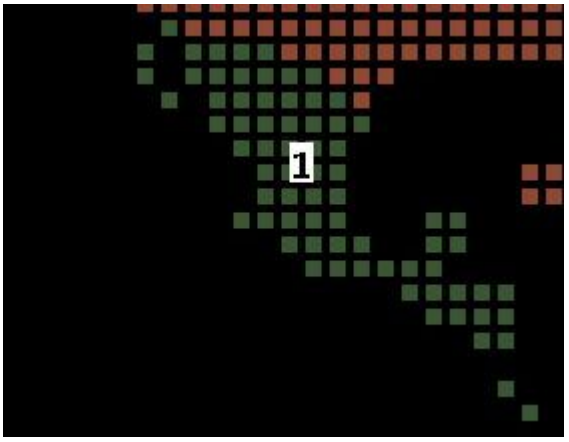
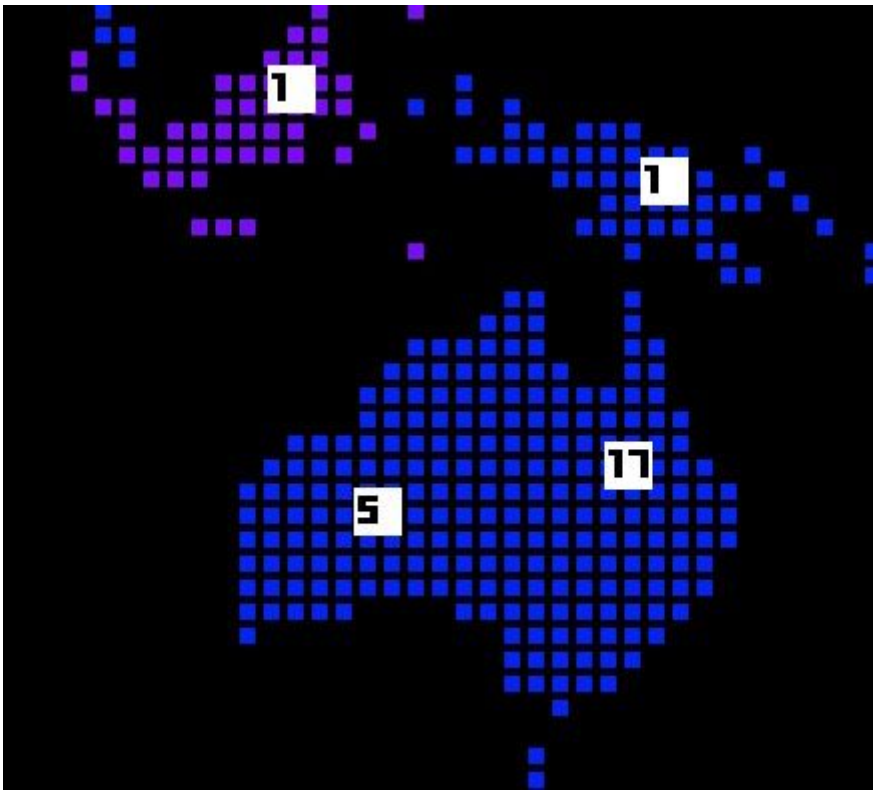


Figure 2

2.5 Unit

Each territory has some amount of soldiers more than zero. The number of the units in the territory is represented by the number on the white square on the territory (*Figure 3*). Soldiers are used for attacking or defending



2.6 Cards

Each territory has its own territory card(*Figure 4*). They can differ in 3 ways that are shown by stars. Stars decide the number of units can be spawned in-game if 3 of that same star-level card are acquired.

Figure 4

2.7 Dice

In order to capture territories player needs to select the territory first to attack under some conditions mentioned in Gameplay(2.1). After selecting territory player has to roll a dice, according to the results of the rolls of players (attacker and defender) the territory will be captured by the attacker if he gets the bigger number in the dice roll than of the defender. The attacker cannot get the territory otherwise. The player can roll a dice(s) in two ways, roll the dice until capture or roll one-by-one.

2.8 Settings

User can go to change settings either from the main menu or from the in-game menu. Inside settings user can change music, increase-decrease volume, change language and difficulty of the game. The difficulty of the game is available only for Single player mode, and it cannot be changed during the game, basically after the game started. Player has to choose difficulty level before he started the game.

2.9 UI Components

2.9.1 Theme

We, as designers, determine to use an unusual theme for our game: The square theme. Today, indie games use pixel theme to decrease the money spend on graphics and make the game prettier. To do so, we use squares to draw something on the screen. These squares make the graphics so cool and basic, also make the game simpler to understand in terms of graphics. As a theme, we use this method across the whole game graphics including map design, menu design, letters and also animations.

2.9.2 Language

In order to make the range of players of our game wider and easier to understand we are adding other languages to the game besides English. So, there will be 4 languages in the game, which are Azerbaijani, English, Turkish and Russian.

2.9.3 Animations

Animations make graphics live. Moving graphics make players feel the game much more. In our game, we added many animations thanks to our theme. After the game opened, squares come to their correct position on the screen from the borders of the screen and merge into letters. To draw the map, squares come from borders to their correct position and compose the map after merging. Also, when combat mode is activated, squares belonging to all territories except two selected territories go to the borders of the screen and disappear then new squares come to the screen from left and right borders and they are merged in terms of the soldier numbers of territories. Moreover, In the main menu and the extras menu, there are also animations to make players feel the general theme of the game.

2.9.4 Sound effects

People like to put background music when they are doing something. As well as when they play games, of course, if the sound effects of the game do not affect anything in gameplay, like shooter games or some other games, for example, hearing the footsteps of the other players makes it easier for players to know where their opponents are. In our game, there is no such a sound effect that would give a player some kind of strategy, so we decided to provide the player background music. As mentioned above in Settings(2.8) users can adjust sound settings as well.

2.9.5 Game experience

Risk is a strategy game as we all already know. By playing this and these sorts of games players can gain some experience such as some kind of strategies or how to manipulate or control your own territory. Afterwards, if the player plays the same sort of game he will be already got used to how to play it and uses strategies and experience that he gained from our game.

3. Functional Requirements

3.1 Play Game

There are to game modes available for the players, which are Single player mode and Multiplayer mode.

After clicking to the **Play Game** button in the main menu Mode selection menu will be screened. Here as an option player will be able to choose either Single player mode or Multiplayer mode.

3.1.1 Single Player Game

If the player chooses Single player mode in the mode selection menu, the game starts with Single player mode.

The territories will be distributed between the player and AI. There will be only two colours for the territories in the map, the ones belong to the AI and the ones belongs to the player. Rules do not change here for any reason. In order to finish the game and win the player has to capture all the territories that belong to AI.

3.1.2 Multiplayer Game

This is our second mode of the game. There will be two to six players will be able to play the game at the same time. In order to play in Multiplayer mode, the user has to choose Multiplayer mode in the mode selection menu, then the game starts.

Again here the territories will be randomly distributed in between the players. The number of colour in the map will be depended on the number of players. Rules are the same for each player and will not change until the very end of the game. Same rules apply here for every player which is the same as in Single player mode. And again here, in order to finish the game and win the player has to capture all the territories that belong to all other players in the game.

3.2 How to Play

Since not everyone knows this game, like me, all users can click on the **How To Play** button in the main menu and go to How To Play screen. Here players will be informed about:

⇒ Overall rules

⇒ The buttons in the game

⇒ How to control the map and territories

⇒ Controls via Keyboard

3.3 Options

Users can click on the **Options** button in the main menu and go to the Options screen.

Here users can change:

⇒ Music

⇒ Sound level

⇒ The difficulty of the game

⇒ Language

3.4 About Us

Users can click on the **About Us** button in the main menu and go to About us screen. Here there is simply some information about group members, and additional users can go and visit our Facebook, Instagram and Twitter pages, where we will share some new updates about our game.

3.5 Extendibility

3.5.1 Extends on the Multiplayer

Right now when users choose Multiplayer mode they have to play on the same computer. We are planning to make some changes on the implementation so that each player can use different desktops to play while Multiplayer mode is selected.

3.5.2 Extends on the Single Player

Right now when users choose Single player mode game is not that fancy. We have a simple AI but we are sure that it will be changed and maybe the current one can be one of the easy difficulty levels of the Single player mode.

3.5.3 Pause

Right now we have Pause function to pause the game but it is not fully functioned. When the player pauses the game there will be in-game menu provided where the player can adjust settings. Additionally, there will be **exit** button if the player exits during the gameplay the game will be saved and the player will be able to continue from the last state of the game.

3.5.4 New Game

After implementing a pause function we will implement the new game function as well. After the player presses the **Play Game** button in the main menu, there will be two options “**Continue** or **New Game**”. If **New Game** is selected default game will start.

3.5.5 Continue

After implementing pause function we will implement the continue function as well. After the player presses the **Play Game** button in the main menu, there will be two options “**Continue** or **New Game**”. If the player chooses to **Continue** he/she will start from the last state of the game where the player left.

4. Nonfunctional Requirements

4.1 Game Performance

4.1.1 Time efficiency

Time efficiency is one of the most crucial criteria for the game. The program should not wait for a long time in order to execute any main function. In order to satisfy this requirement, appropriate data structure and algorithms are used. Execution of any major function (for example button actions) should not take more than 1 second.

4.1.2 Space efficiency

Space efficiency is also an important point like time efficiency. The program should not occupy a big amount of memory. All files that belong to the program should not take more than 500 MB memory in the Hard Disk.

4.2 User-friendly Interface

4.2.1 Usability

Usability of the program is especially important for users. The program should be used and learned without any difficulty. In order to satisfy it, reasonable (well known) names should be used for buttons and there will be instructions about how to play the game.

4.3 Compatible Interface with real world

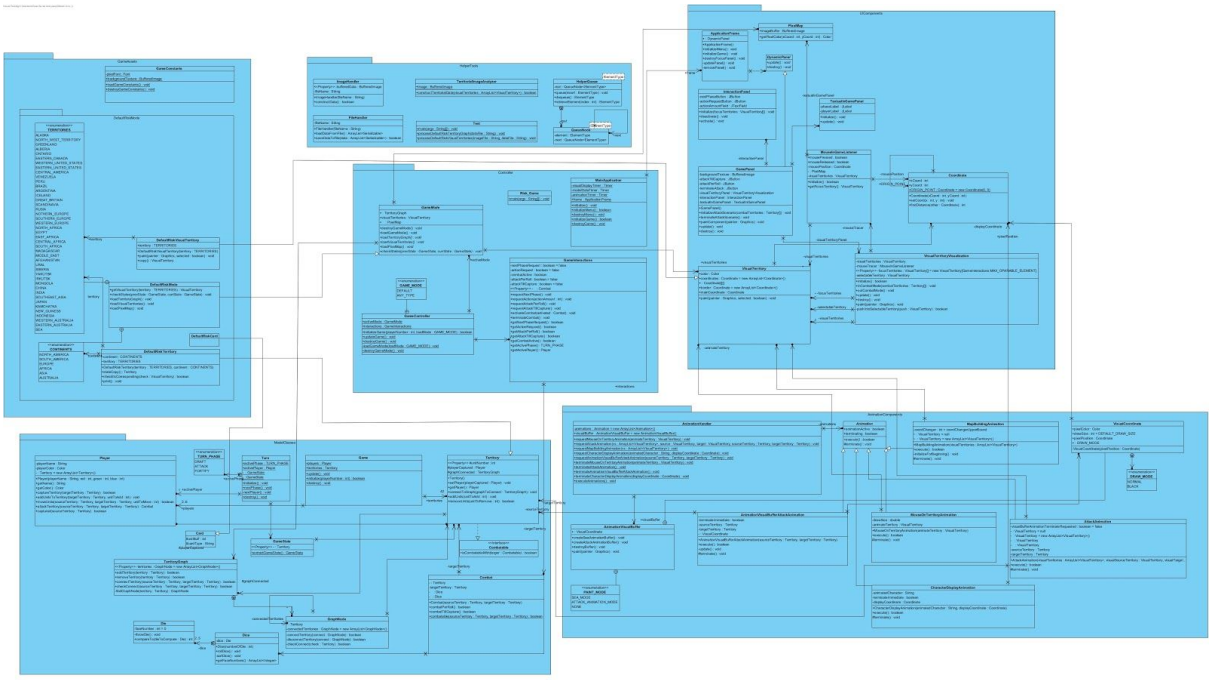
Since the “Risk Takers” game have been derived from the “Risk” game, name of game components and actions should be almost the same with board game type. As a result, it will be easier for players of the board game to adjust to a new software version. To satisfy this requirement, name of actions, territories and in-game phases are the same with board game version.

5. System Models

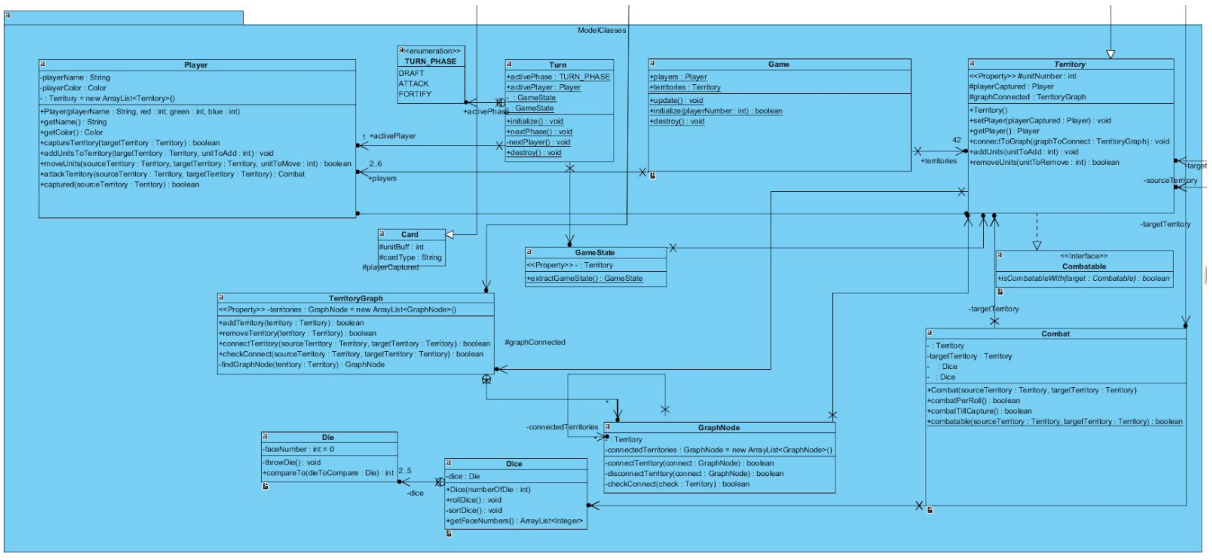
5.1 Use case model

5.2 Object and Class Model

Here, we have 6 different packages for the general appearance of class models. Four of them should be considered as game essential that effectuate the fundamental flow of the game and application; one of them, namely GameAssets, is designed for handling the game constants and mode-specific data. Lastly, HelperTools is mainly purposed for the development phase of the game.

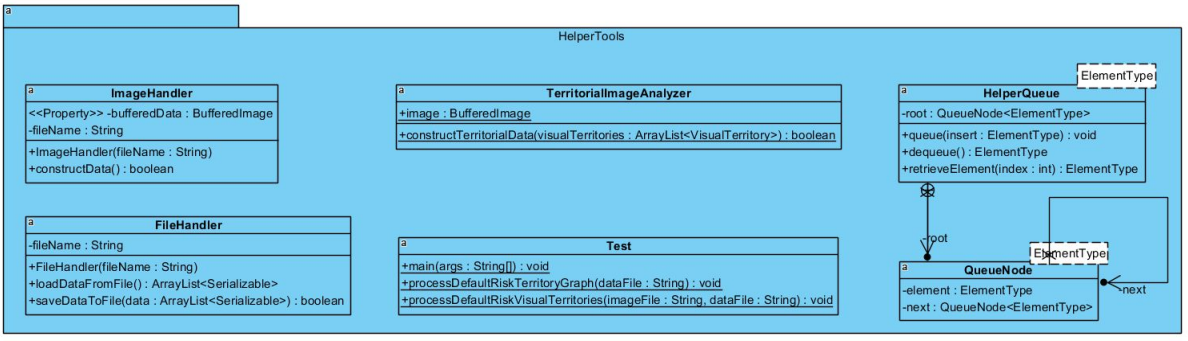


Model Classes



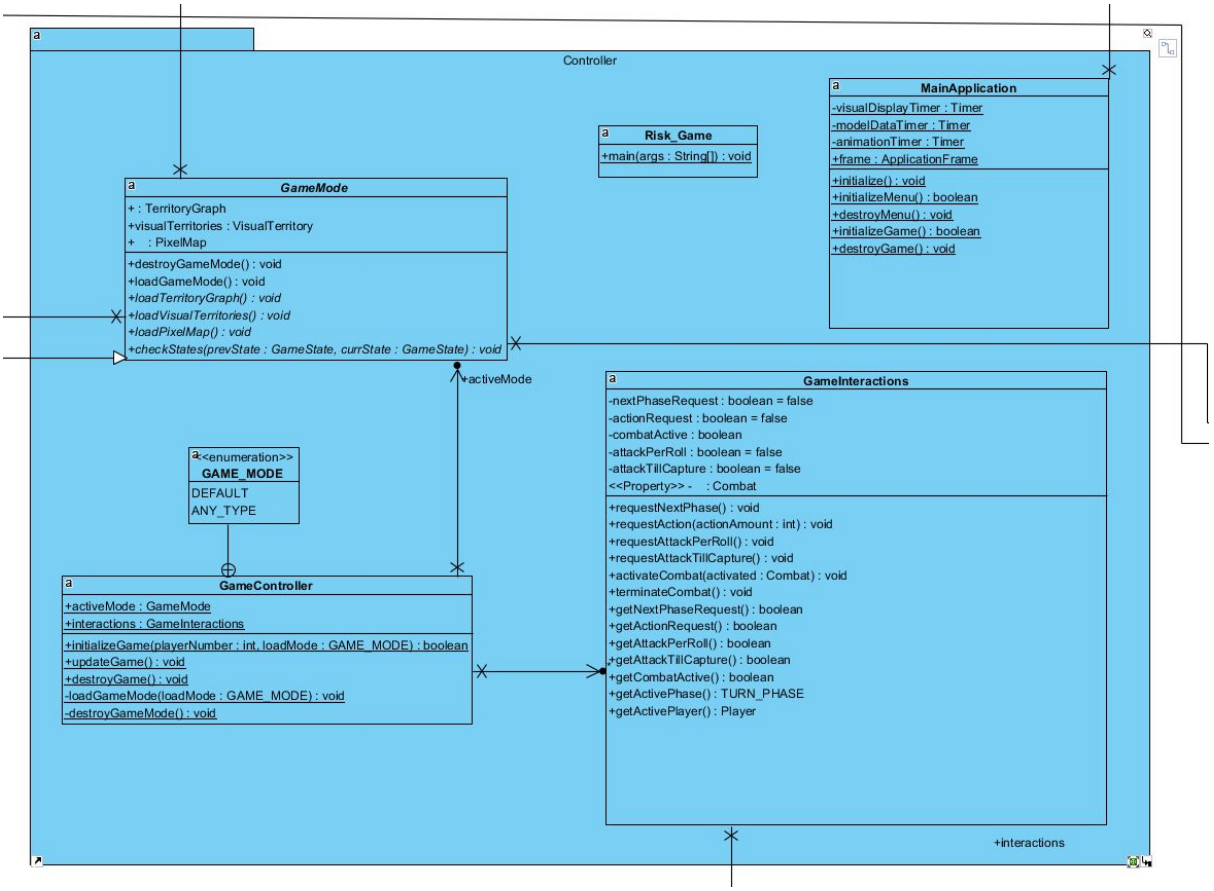
Model Classes package contains the fundamental risk game classes such as territory, combat, player and dice. In other words, this package is the skeleton of the game, actions captured with UI Components package executes the Model Classes package class methods. The visual aspects of the objects are given to monitor by using Model Classes objects properties.

HelperTools



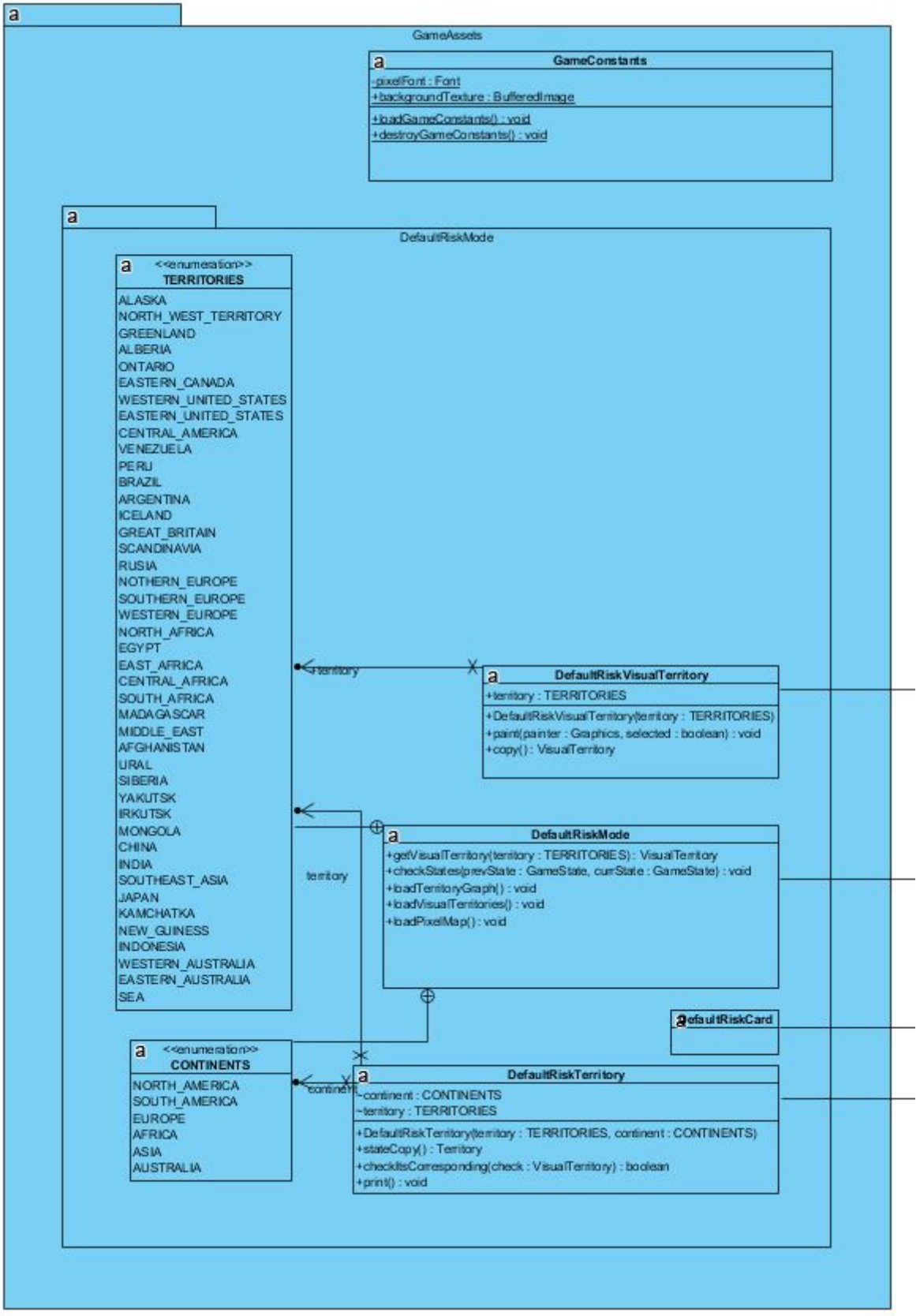
HelperTools package is the package that makes our implementation easier such as ImageHandler, FileHandler. By using these classes the data of pixel map, border paths and card types are obtained.

Controller



Controller Package is the package that controls loading of the games map, properties of the territories and updating the attributes of the relevant objects of the model classes package.

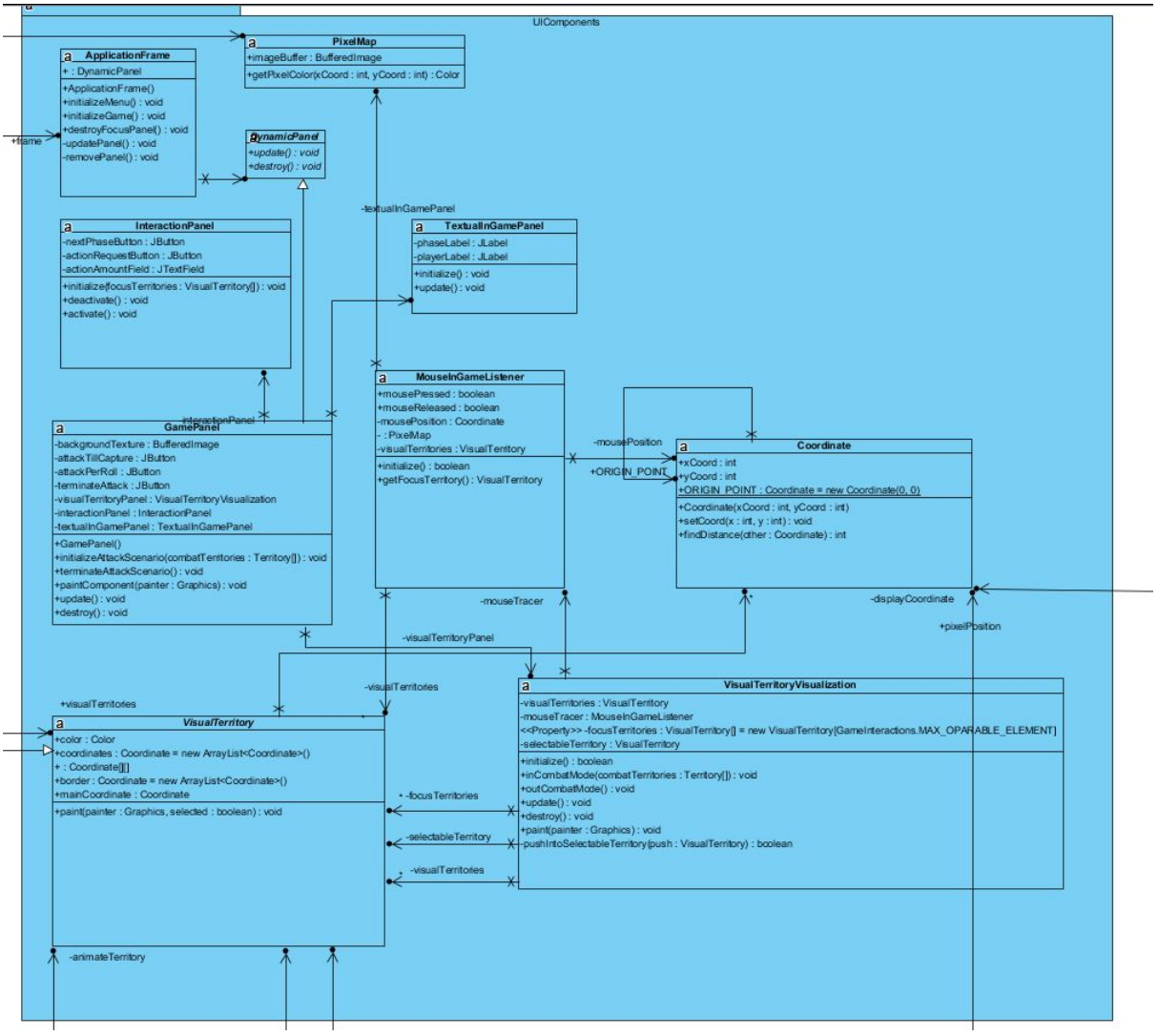
GameAssets



The default risk game is played on the world's map and its territories. And this package contains the Default risk games territory names which are used in the main

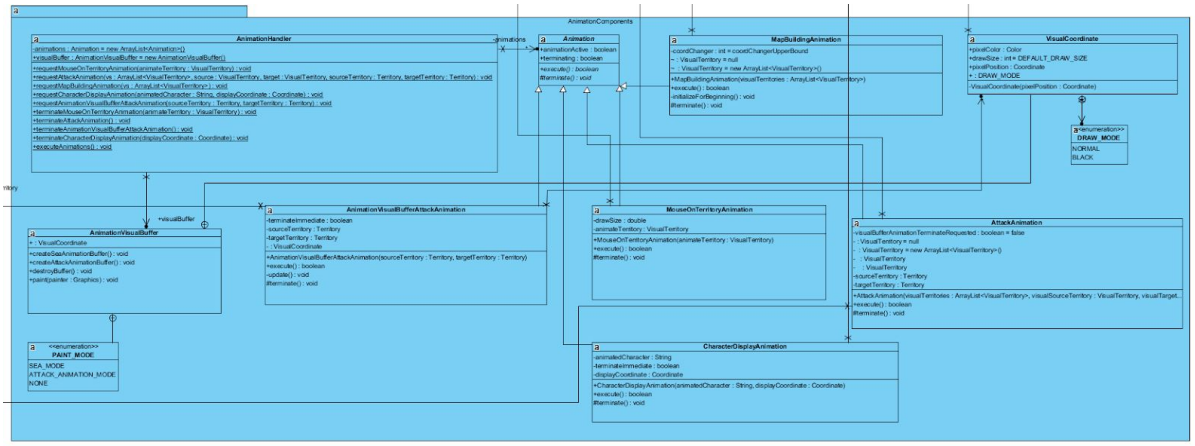
game. In the game number of territories and their properties such as color name and place are loaded by this package.

UIComponents



UIComponent package is the package for all the considered input, they are responsible for all the territory selections, attacking, adding unit and so on. They take the input from the player through the mouse and buttons.

AnimationComponents



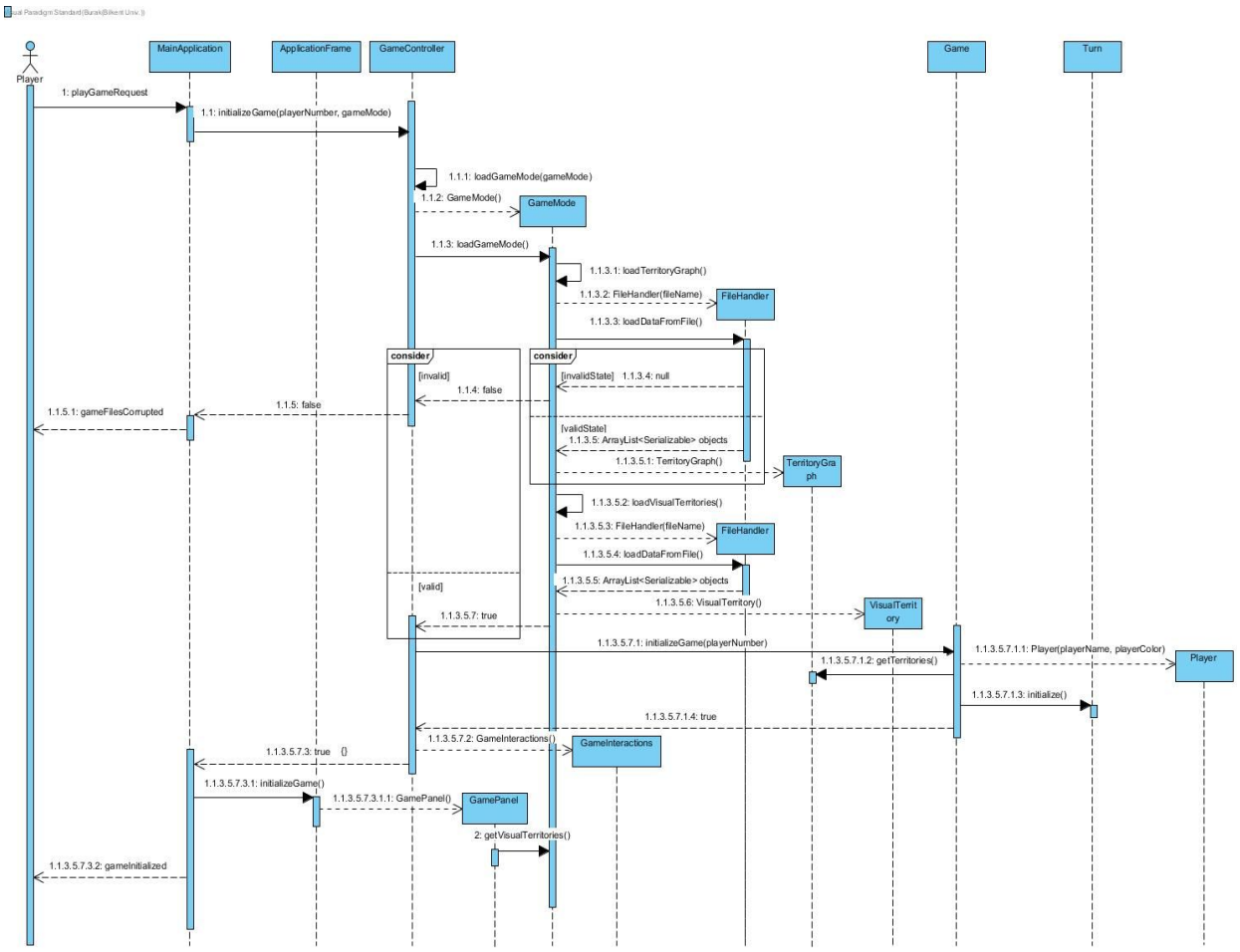
AnimationComponents package is the package that where the animations loke selection animation, starting animation and attack animations are handled.For example, AnimationVisualBufferAttackAnimation class handles the attack animation with corresponding to source territory and the target territory.

5.3 Dynamic models

5.3.1 Sequence Diagrams

5.3.1.1 Game Initialize

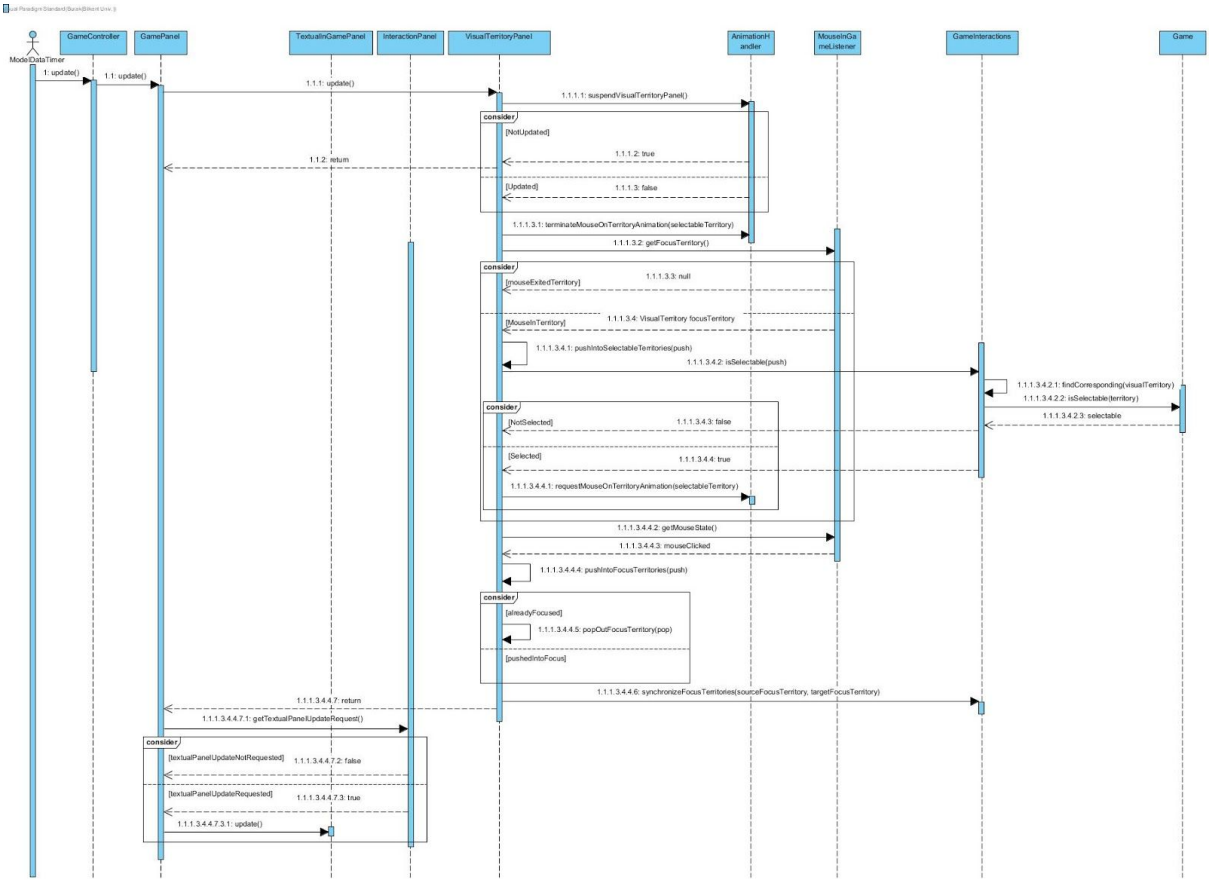
Game initialize action occurs when the user triggered it with “start a new game” button in the menu. Basically, it handles the loading of game-related data like visual and constants into the memory and arranging players and their initial territories.



Game initialization involves the loading data from the game files; firstly extract constant visual and model data into gameMode objects of GameController class with the help of FileHandler objects. Then, distributes such data among the Game class and GamePanel object. In that way, initial constant data and some dynamic data like playerNumber of classes are instantiated.

5.3.1.2 GamePanel Update

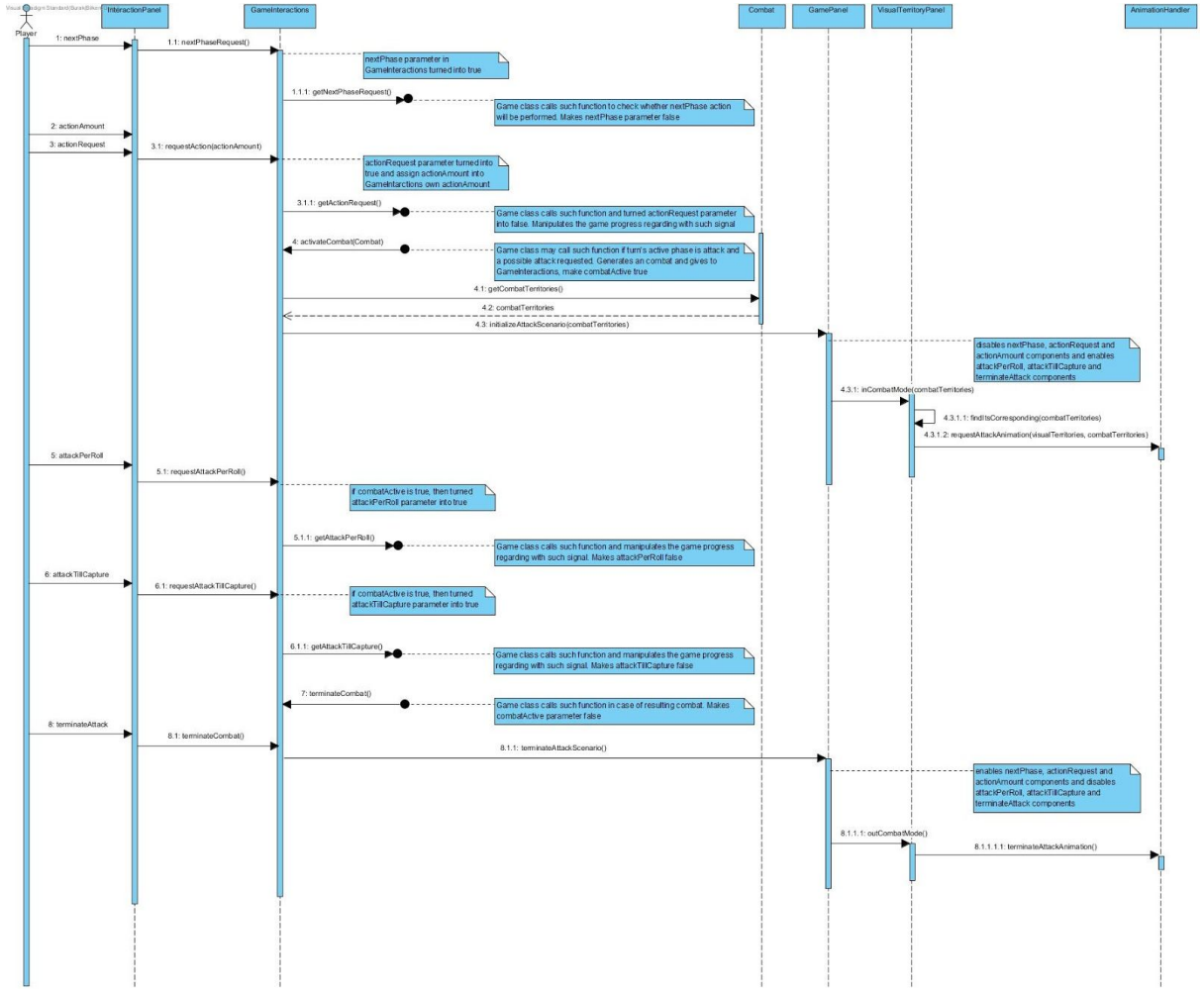
GamePanel update action involves the progress of updating the UI-related data like selectedTerritories and on which phase the game currently is. Regarding with such data, it updates the GameInteractions' signal for manipulating model data.



There are two basic factors for the change in the visual data of the panel. These are MouseInGameListener and InteractionPanel. According to incoming signals from such classes, the required actions will be executed, in which required actions may be triggering some animations or TextualPanel data update. Also, it includes synchronizing user input signal with GameInteractions' outgoing signal to the model data, where the real manipulation of the game model data occurs.

5.3.1.3 User Interactions

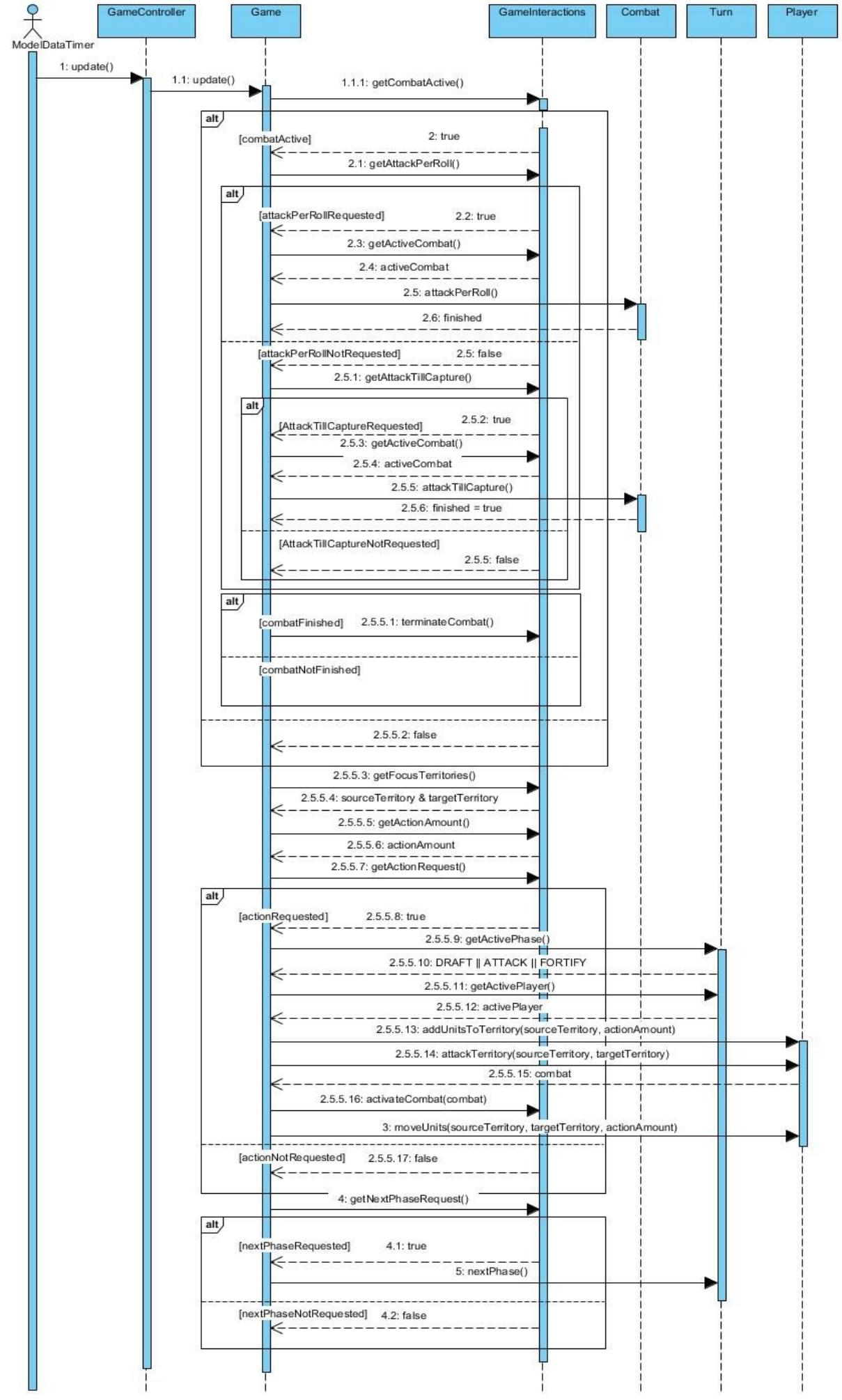
User interactions refer to the effect-response interactions among the game. These are required inputs from the user for going next phase or performing an action.



In the current state of the game, there are 6 possible options, in which you can 3 of them is disabled since they are attack-scenario specific interactions. Attack actions may trigger some animation for attack scenario and may terminate them. In the result of the player's requesting nextPhase or any action, the required parameters in GameInteractions' class will be prepared with these interaction mechanisms for the next update wave of Game.

5.3.1.4 Game Update

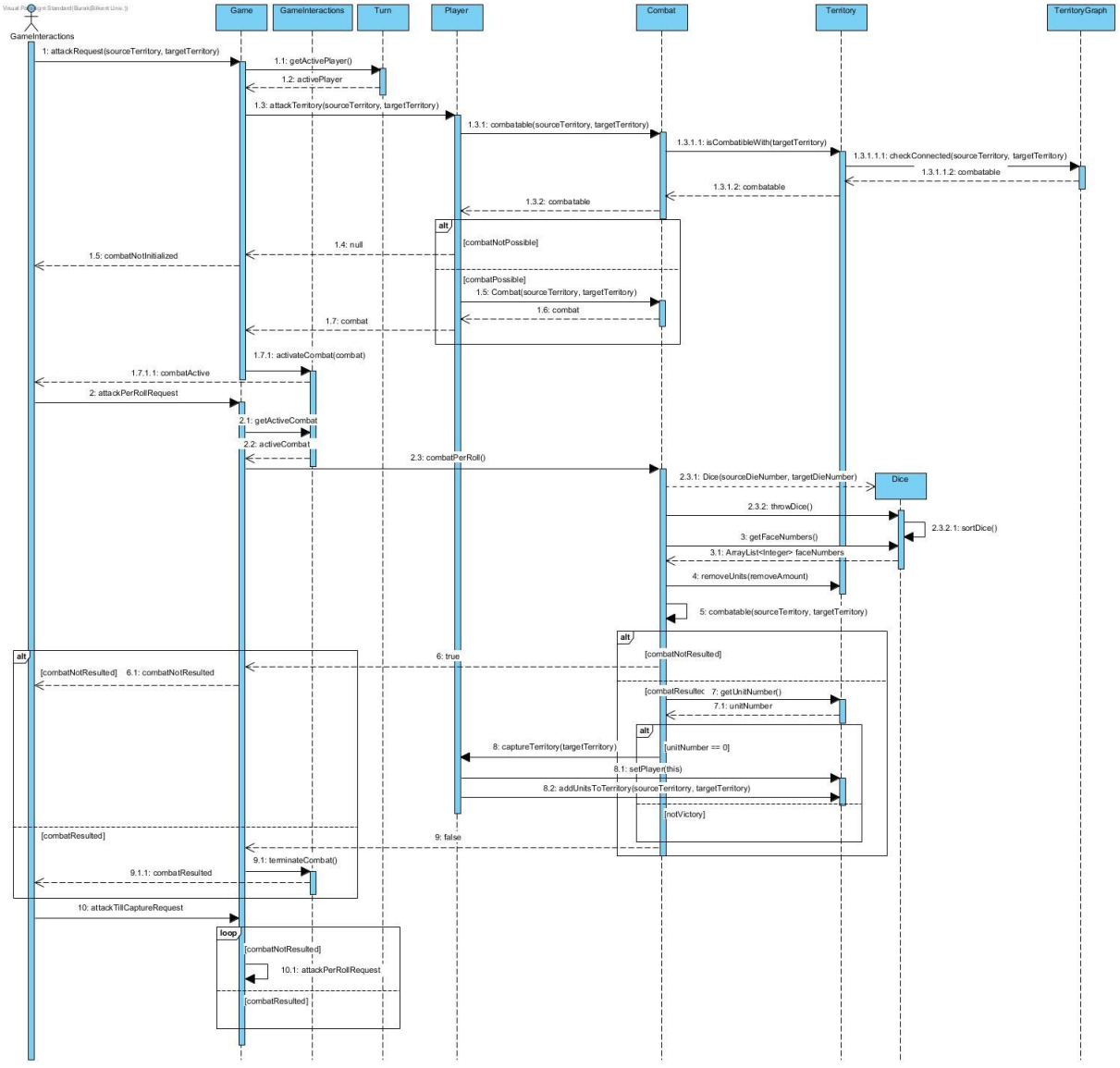
GameUpdate involves the manipulation of game's model data regarding with extracted signals from GameInteractions' parameter, that is, actually UI-originated inputs.



Here, a update wave of the game may include only particular actions at once. These can be listed as nextPhase or action (regarding with activePhase of Turn class, adding unit or attack scenario or transferring units) or performing attack-specific actions. Since modelDataTimer calling such update function, this action occurs many times and no possible way to perform more than one action mentioned above in one update wave. Also, it disables nextPhase and actionRequest when attackScenario initialized and enables attackPerRoll, attackTillCapture and terminateAttack, that is, reversely when attack terminated or resulted. So, there is no way to contradict effectuating of such signals with each other.

5.3.1.5 Attack

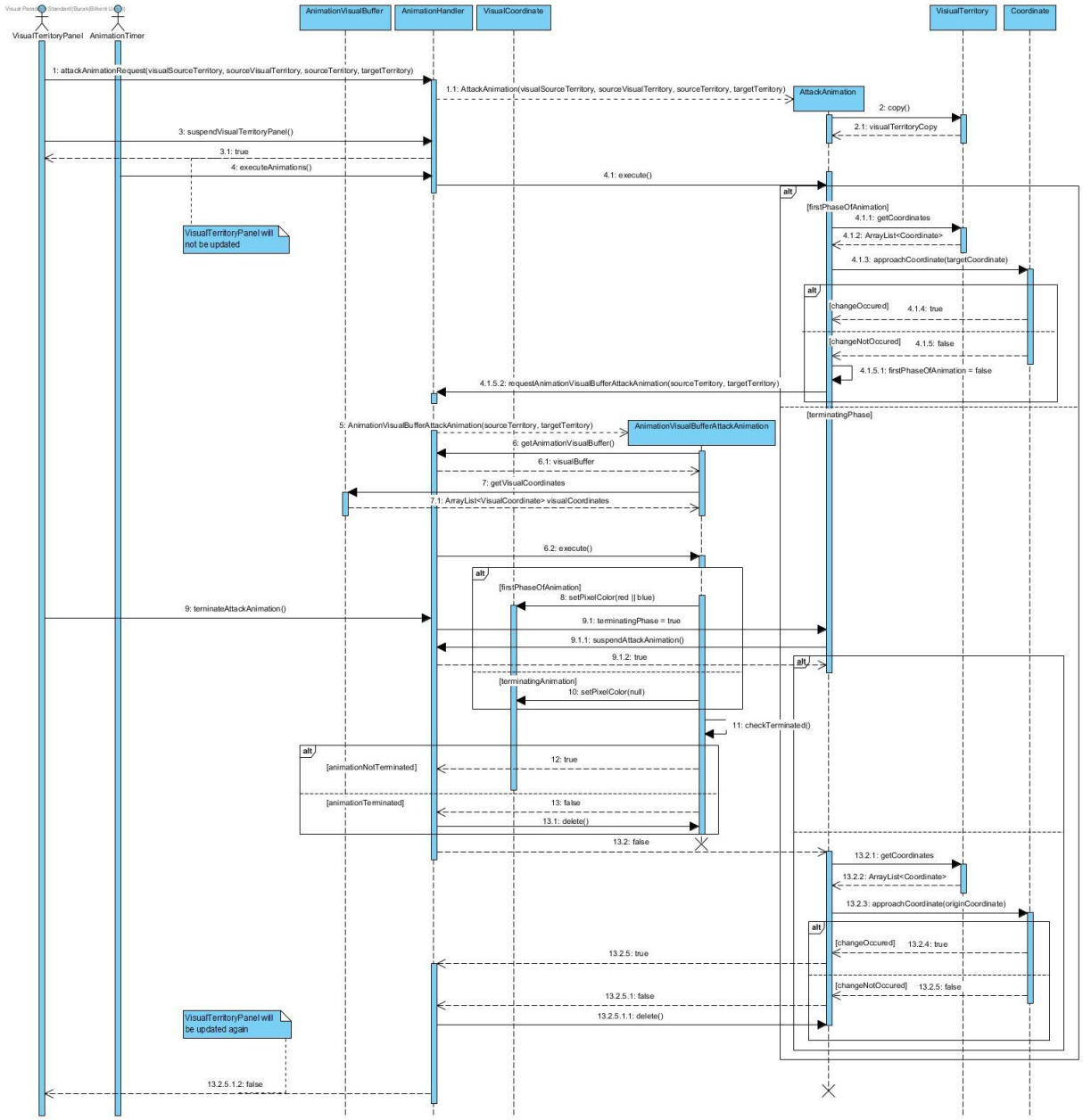
Attack action involves the combat simulation of two territories regarding with their unit numbers and their corresponding dice. Removing units from territories and capturing territories is performed according with combat result.



Any actions related with territories is not directly performed over Territory objects. Instead, such relations is operated over Player and Combat objects. The result of these actions will be transferred to GameInteractions over Game class to reflect their effect on the UI components.

5.3.1.6 Attack Animation

Attack animation implies the reasoning of Animation abstract class which denotes the actual philosophy about how an animation initializes, effectuates and reversely terminates. In the matter of implementation of such animation mechanism, all animations controlled by AnimationHandler with a proposed design.

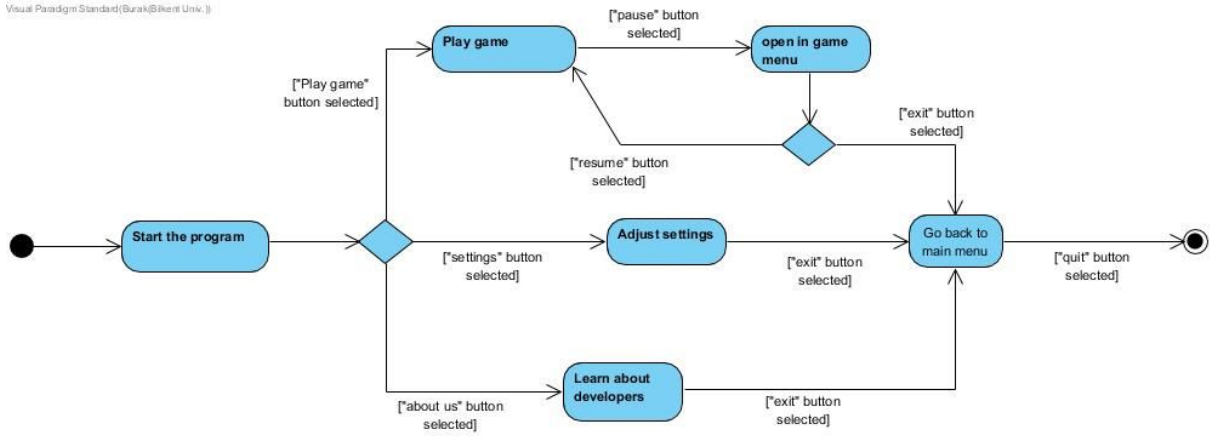


As an animation, attack animation will be requested and terminated over AnimationHandler. Attack animation handles its responsibility by changing the

coordinates of visualTerritory on the screen. Also, it includes an inner animation request and termination. There is the mechanism of suspending an animation' progress, in which animation keeps its current state without manipulating any visual data and waiting for such suspension to be eliminated. Also, such suspension is avaialable of some panel classes to be eliminate some update-related hazard and disruption.

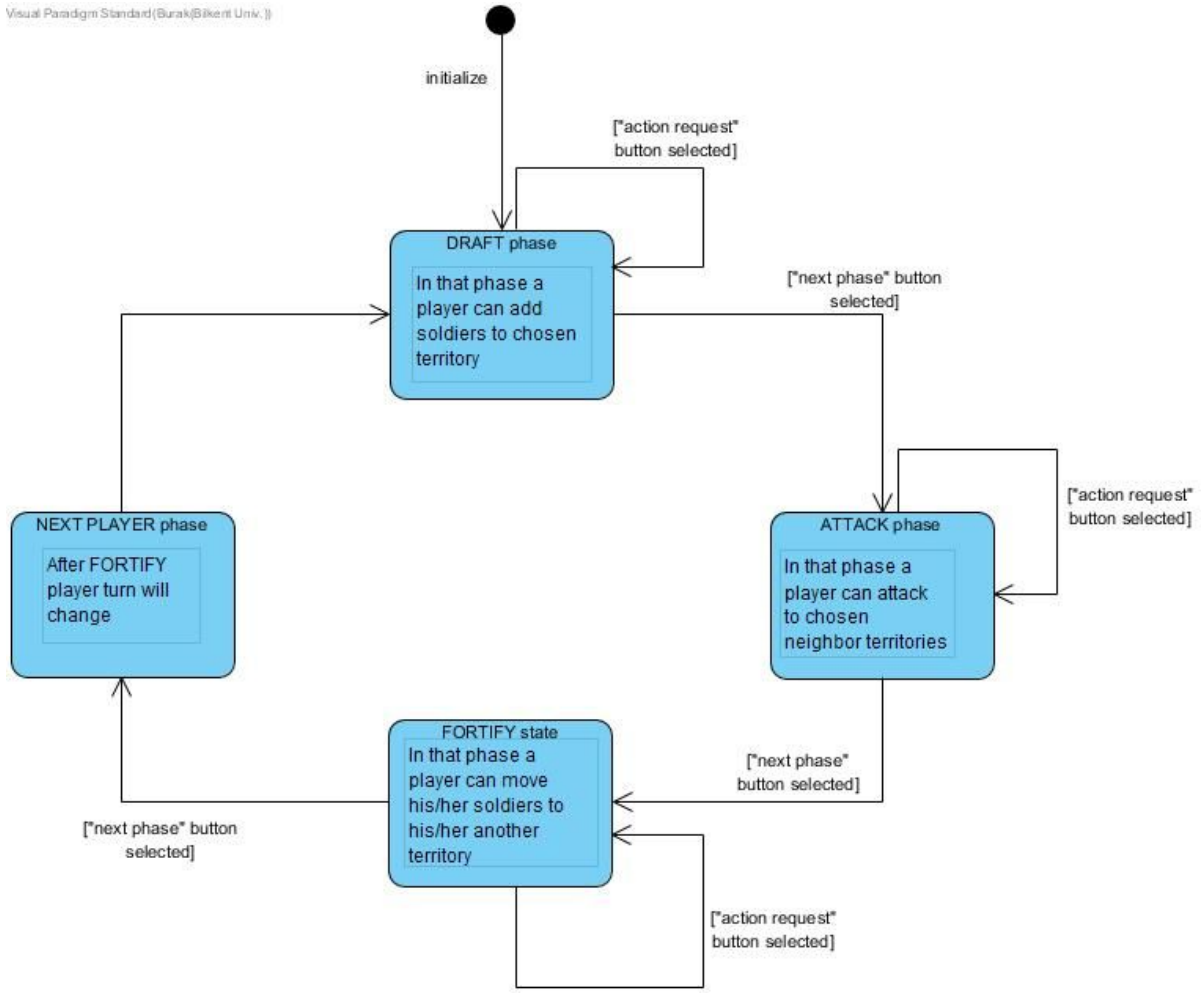
5.3.2 Application Mainflow Activity Diagram

This activity diagram simple shows the event flow starting with opening the program and ending with quitting from the program.



5.3.3 Turn State Diagram

After player starts the game, it will actually cycle between 3 different phases. Draft, Attack and Fortify phase. In addition, if a player passes all these 3 phases, turn will change to next player. So there will be 4 states in total and transition from NEXT PLAYER state to DRAFT state will be automatic.



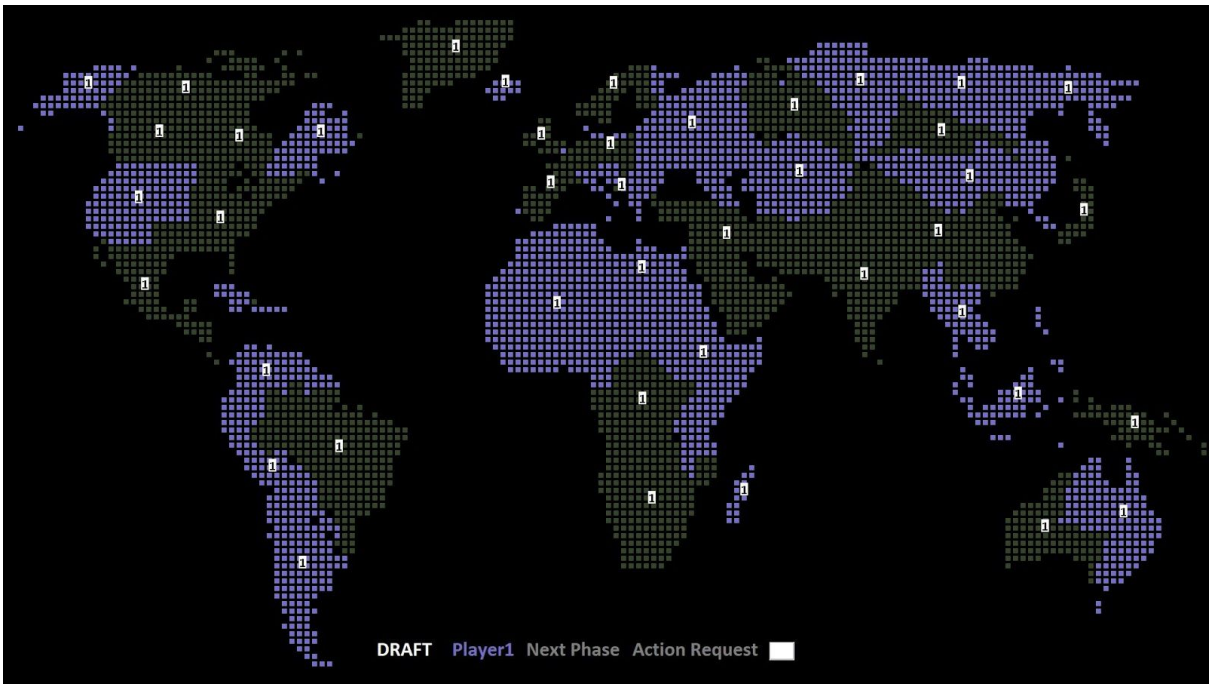
5.4 Screen mockups

These are not not the latest version of the UI. And these are the pictures, there are actually animations running on the screen.

5.4.1 Menu Panel

This is the panel when the game starts. There are some buttons on the left corner which helps the player to change settings, play the game check the rules and see some information about us, and quit the game.

5.4.2 Game Panel



This is the panel players see when they start playing the game, meaningly press the play game button in main menu.

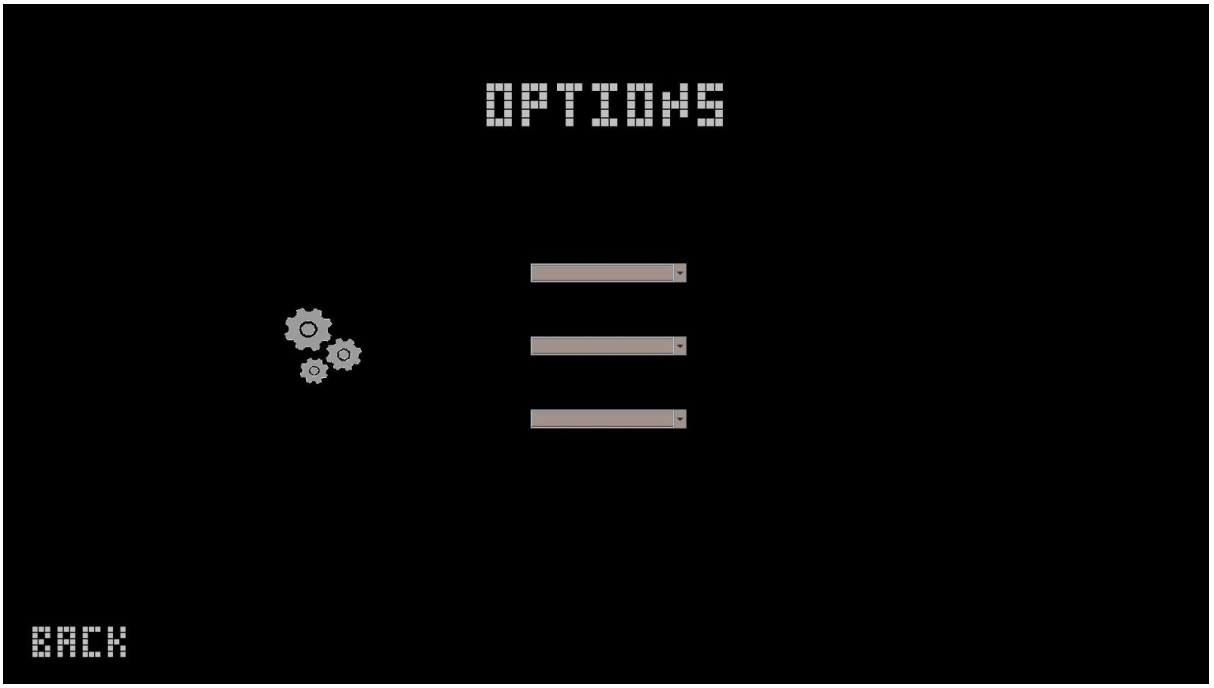
In the place where “**DRAFT**” is shown all the phases of the game will be shown each time the player presses the **Next Phase** button.

Each time the turn of the players change it will be seen in the place where “**Player1**” is written.

Action Request controls all the actions in the game right now. For example, during the Draft phase if the player chooses one of his territories and adds number of units to the empty JTextArea and press the Action Request button the number of players in the specified territory will be increased.

This is for now, it will be changed, more fancy, more creative.

5.4.3 Options Panel



This is the panel players see when they press the options button in the main menu. Back button here takes the player to the previous menu, which is menu panel. There are some ComboBoxes here which one of these are used to represent Language, Music Selection, Difficulty level and etc.

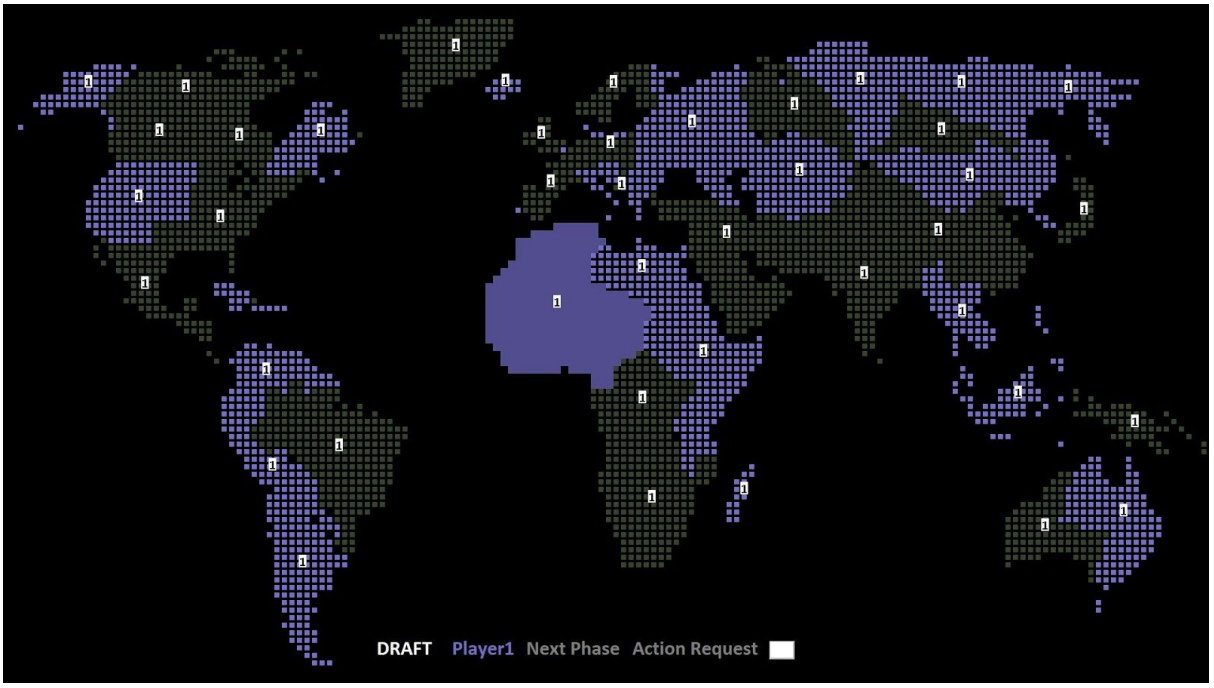
5.4.4 How To Play Panel

In this panel, players will be provided with the rules of the game. They can go to this panel from the menu panel by clicking on the **How To Play** button.

5.4.5 About Us Panel

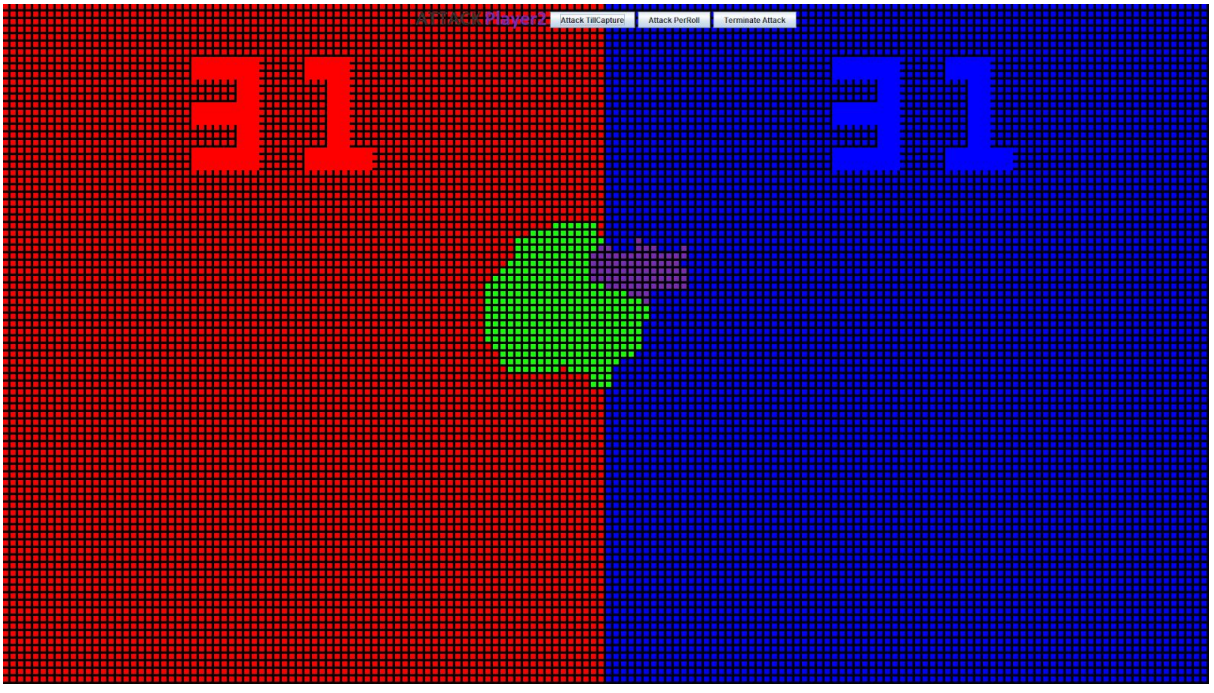
If users press the **About Us** button this panel will be screened, here are some information about us.

5.4.6 Selected Territory

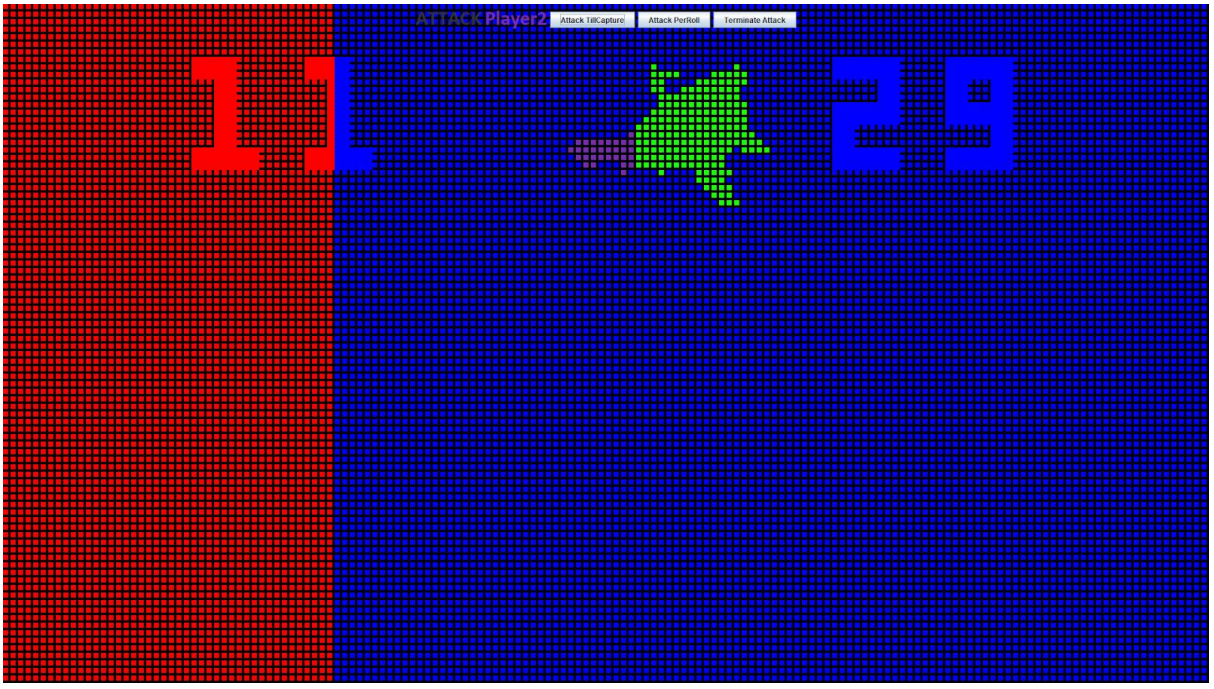


During the gameplay, if the player will select one of his territories it will be screened this way.

5.4.7 During Attack



This is screened when the player is in attack phase and selected a territory to attack.



This is after multiple dice rolls.

6. Conclusion

After the first iteration analysis and design, we come up with more inspirational ideas about implementation and game design. Also, we had a more progressive implementation on the project. As a result, Model Classes are fully designed and implemented, that is, these are will not need any further changes. The middle layer classes between UI components and ModelClasses are structured regarding with MVC. Also, such middle layer classes controls the mainflow of the application and especially game. Lastly, UI components implemented in such way that they currently have a uniquely characteristic appearance and animations are included regarding with the purpose of not deviating from such characteristic appearance. As a combination of these animations, visual components and sound effects, it would be an enjoyable game experience for the players.

7. Improvement Summary

Animations, briefly, is about moving squares across the screen to their previously determined correct coordinates. They are working with timers; if each square is in their correct position animation is acquired as done and timer of the animation is stopped. Some animations may get input to start, combat and territory have animations waiting for the input to start, however, some animations work in the

loop and do not require input such as main menu animation which restarts after done.

The theme is made with squares to make the game looks good and to make the animations easier.

Language is necessary for players to understand the game in their native language.

Multiplayer and single player modes enhance the whole experience.

8. References

[1] ht t p://www.ult rabo ardga me s.co m/r is k

[2] https://en.wikipedia.org/wiki/Risk_(game)