CS 319 - Object-Oriented Software Engineering

Analysis Report

Colony Wars

Supervisor: Bora Güngören

İbrahim Taha Aksu

Ahmet Emre Nas

İzel Gürbüz

Kaan Çakmak

Table of Contents

1 Introduction

1.1 Purpose

This document is to introduce an SRS for the game “Colony Wars” by giving an overall description of the software, specifying both functional and nonfunctional requirements, providing various system models such as use case model, dynamic model and class model supported with user interface navigational paths and screen mock-ups.

The intended audience for this particular documentation is us being the developers in the first hand to clarify the requirements of the software system we are going to deal with. Moreover anybody who are interested in our project could get a grasp of what our final project going to be like and check the requirements, models and mockups for having a general understanding of the project.

1.2 Scope

The project that is going to be described throughout this report is a strategy game called “Colony Wars”. Colony Wars is a leisure time entertainment for adults and kids who are into strategy games. The game is not hard or complicated regarding user interference while success in the game highly depends on decisions taken by user. Thus playing the game will provide players with better strategical thinking and if not will entertain users with challenges exposed to him/her.

1.3 Definitions, acronyms, and abbreviations

The project being a game having a specific design of story behind, the SRS for the game uses some particular terms that the reader may not be familiar with. Some of these terms are listed below:

Base: The term base is used for particular and most popular kind of building in the game. It is the building which both of players start by possessing one. Its only trait is its capability to produce soldiers and nothing special other than that.

Strategy game: A strategy game is a game in which the players' decision-making skills have a high significance in determining the his/her success.

Neutral base: The term neutral base is used for bases that do not belong to any of the players and are thought to present before players under possession of local people living there.

Armory: An armory specifies a type of building where weapons are made and kept. In the game the term will be used tto specify a particular type of building which will be further explained in the Overall description.

Tailor: Tailor being a person who provides people with clothings, in the game will be used to specify a particular type of building which will be further explained in the Overall description.

Hospital: Hospital is a place where people are provided with all kinds of health treatments. In the game the term will be used to specify a particular type of building which will be further explained in the Overall description.

Azad: This term not having a real meaning will be used as a name for one of three nations that is included in the game.

Doth: This term not having a real meaning will be used as a name for one of three nations that is included in the game.
Cult: This term not having a real meaning will be used as a name for one of three nations that is included in the game.

1.4 References

This part will be updated later on.

1.5 Overview

The rest of the SRS takes into hand a broad description of the game comparing it with other games on the market those are similar to it. Discussing its contributions and shortcomings to and from these games. The SRS also includes the subjects on requirement specifications, specifying both functional and nonfunctional requirements. It has a part separated for modeling which includes models for dynamic, use case and class models and also for user interface. The document ends with the final part which is Appendix (describing game rules).

2 Overall Descriptions

2.1 Product Perspective and Functions

Colony Wars is a time based strategy game that requires usage of a considerable amount of strategical thinking. Game map basically consists of 2 main bases for each player and several neutral bases that both players seeking to possess. The main purpose in the game is to get control of all the bases in the game. Every base produces number of soldiers by time passing. In order to get control of a base you need to have required amount of soldiers in it if it is a neutral base, if it is an enemy base though you need to send more soldiers than enemy soldiers that currently present in that base. There are several games similar to Colony Wars currently on market to the writer’s knowledge that are popular one of them being Mushroom Wars.

An image from the gameplay of game Mushroom Wars

The game mechanics will be more or less similar to Mushroom Wars meaning that user will need to click on a base of him/her and drag it to the base he/she wants to attack. This action would cause the half of the soldiers currently in the clicked base move to the base where the mouse is released. If the mouse is released on a friendly base then the soldiers would collide with the ones already in that base if it is a neutral base then user will take control of that base under the condition that he sent the needed number of soldiers to that neutral base. This number will be written on that very base. Another state is where the base belongs to the enemy in which case the enemy and allied soldiers will go on a fight and a winning side will be determined based on the number of soldiers both sides have. The side with more soldiers will take control of the base and the surviving soldiers will be left in that base whose number is determined by the difference in allied and enemy soldiers.

Another important point in the game is that it will have various neutral bases with different advantages provided to its owner. One of these being Tailor, which provides its owner soldiers with shoes and clothing making them move faster. Another one being Hospital which provides its owner a greater reproduction by giving better healthcare to its people. The last but not least is a base called Armory which increases the damage of soldiers belonging to a nation by providing them with upgraded weapons. To examine if two players one having access to a armory were to fight, nine soldiers of the nation with armory could take down twelve soldiers of the one who does not have an armory. An important point to be aware of in case of special bases is that it provides the treats it promises not only to soldiers in that base but to all soldiers of its possessors.

The game also has variations of three nations each having advantage in one or more of the soldier traits which are speed, production rate and soldier damage. First nation called Cult has a greater speed and damage compared to other nations thanks to it greatly developed weapons and clothing although the production rate of their species are low. Another nation in the game is called Azad having faster production rates and movement speed compared to others. They have smaller body sizes and are less developed which makes them fast and

increases their reproduction rates since less developed species produce more often. However they have the drawback of having less damage than other species because of their small body sizes. The last nation in the game is called Doth which is a male dominant colony with males having more than one wife and are stronger compared to other species. However the nation is considerably behind in sciences and technologies which come with the drawback to their soldiers being slow compared to others.

The game will also have a campaign mode which includes a series of games all having different objectives. Some examples for these varying objectives are as following:

- Capturing all bases.
- Capturing a particular base.
- Reaching a particular amount of soldiers.
- Having more bases compared to your rival in a specified time.

2.2 User Characteristics

The game addresses any scope of individuals under the condition of being aged enough to be able to use the controllers of the game and understand the basics. There is no upper limit to which the game is appropriate for nevertheless people who have interest in games overall and strategy games in particular would give probably more attention to it.

3 Specific requirements

3.1 Functional Requirements

3.1.1 Start New Game

User can start a new game via the related button in the main menu. After choosing his option player will be directed to a page where he/she would be expected to choose the game mode.

3.1.1.1 Select Game Mode

There are 2 possible game modes: Skirmish and Campaign. If the player chooses Skirmish mode he/she will choose the map, difficulty and the nation. If the player chooses the campaign mode they choose the nation, a series of games will start until they finish it successfully or choose to exit.

3.1.2 Open Tutorial/Help

Player will have the option to see the tutorial/help for the game which will be accessible through main menu or in the pause menu which would be accessible in game by pausing the game.

3.1.3 Load Game/Choose Level

Playing a campaign mode player will be able to continue to the game where they have left some other time via load game button that is located in the main menu. Using this button will direct them to a map menu where they will be able to choose from which map they would like to continue.

3.1.4 Sound on-off

There will be a button for muting the background music in the game in case the user does not want it .The button will be located in both settings that is located main menu and pause menu.

3.2 Nonfunctional requirements

3.2.1 Usability

Aiming a user scope that is so broadly open from children to adults our program should have a user interface that is clearly understandable regardless of the users experience with any software applications or games. Besides having a simple interaction for user the tutorial/help part should be explaining basics and usage of game's features neat enough so that nearly no one would not have any problem using it.

3.2.2 Reliability

Provided with any input regardless of valid or not the game should keep working without any crash and should be well prepared for such cases so that it gives the best service possible to the user.

3.2.3 Performance

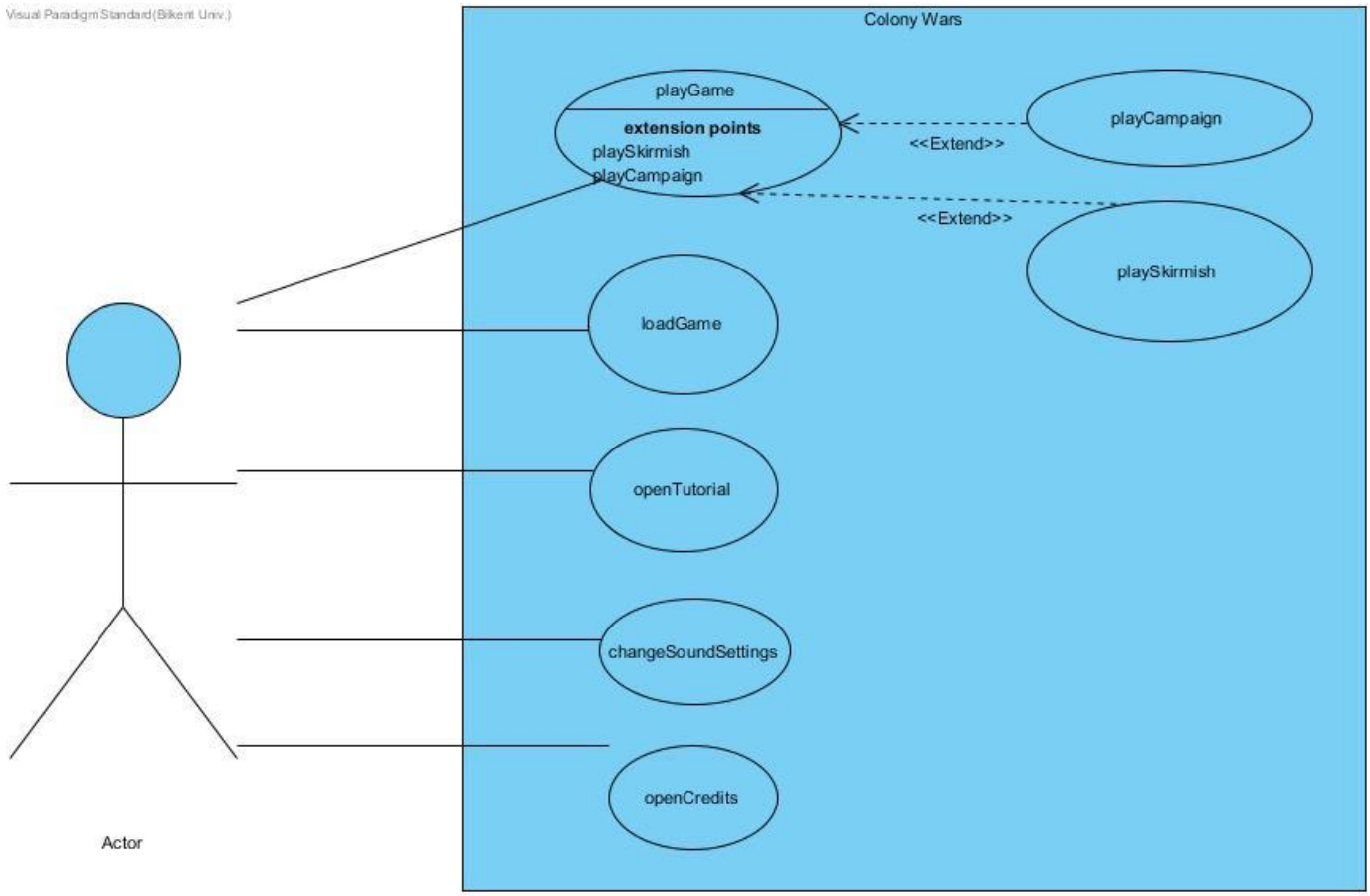
The game is going to be refreshing frame and doing calculations constantly since it's a real time dynamic strategy game. Thus it should have a fast enough response time and should be stable in spite of the length of the game which comes with more workload by time passing.

3.2.4 Extendibility

Regarding the limited time we as developers have we are not including some of the functions although we are eager to have them. Thus the game should be extendible further in the future.

4 System Models

4.1 Use Case Model



4.1.1. Play Game

Participating Actors: Player
Entering Condition: Player has already opened the game and clicks the play button and selects one of the game play from campaign and skirmish.
Exit Conditions:
1- Player loses the game OR
2- Player wins the game OR
3- Player leaves by clicking to exit button

Main flow of events:
1- Player starts the game.
2- Player chooses the difficulty

3- Player chooses one of two game modes: Skirmish or Campaign. If the player chooses Skirmish play Skirmish use case is used otherwise play Campaign use case is used.

4.1.1.1 Play Campaign

Participating Actors: Player
Entering Condition: Player has already opened the game and clicks the play button and selects the campaign mode.
Exit Conditions:
2- Player wins all games OR
3- Player leaves by clicking to exit button

Main flow of events:
1- Player starts the game.
2- Player tries to complete all the steps according to missions on each level.
3- Player continuously directs his/her soldiers throughout the gzame from a base to another which he/she wants to attack/support.
4- Player finishes the campaign by accomplishing all the missions given in each level.

Alternative Flow of events:
5- Player fails to accomplish an objective in one of the games and refuses to play again.
6- Player leaves the game by pressing the exit button.

4.1.1.2 Play Skirmish

Participating Actors: Player
Entering Condition: Player has already opened the game and clicks the play button and selects skirmish mode.
Exit Conditions:
1- Player loses the game OR
2- Player wins the game OR
3- Player leaves by clicking to exit button

Main flow of events:
1- Player starts the game.
2- Player chooses the map and nationality.
3- Player plays against the AI according to selected difficulty.
4- Player continuously directs his/her soldiers throughout the game from a base to another which he/she wants to take/support
5- Player tries to conquer all bases before the AI.
6- Player wins the game by taking all bases.
Alternative Flow of events:
7- AI wins the game by taking all bases.

8- Player leaves the game.

4.1.2. Load Game

Participating Actors: Player
Entering Condition: Player clicks the Choose Level button from the main menu.
Exit Condition: Player clicks the back button and goes back to the main menu.
Main flow of events:
1- Player clicks on the Load Game button.
2- The games (maps) will be loaded which player played and did not play yet.
3- Player selects one of the maps which he/she played or the next one which player did not play.

4.1.3. Open Tutorial

Participating Actors: Player
Entering Condition: Player clicks the Tutorial button from the main screen.
Exit Condition: Player clicks the back button and goes back to the main menu.
Main flow of events:
1- Player clicks on the Tutorial button.
2- Tutorial screen will be opened.
3- Player returns to the main menu by clicking to back button.

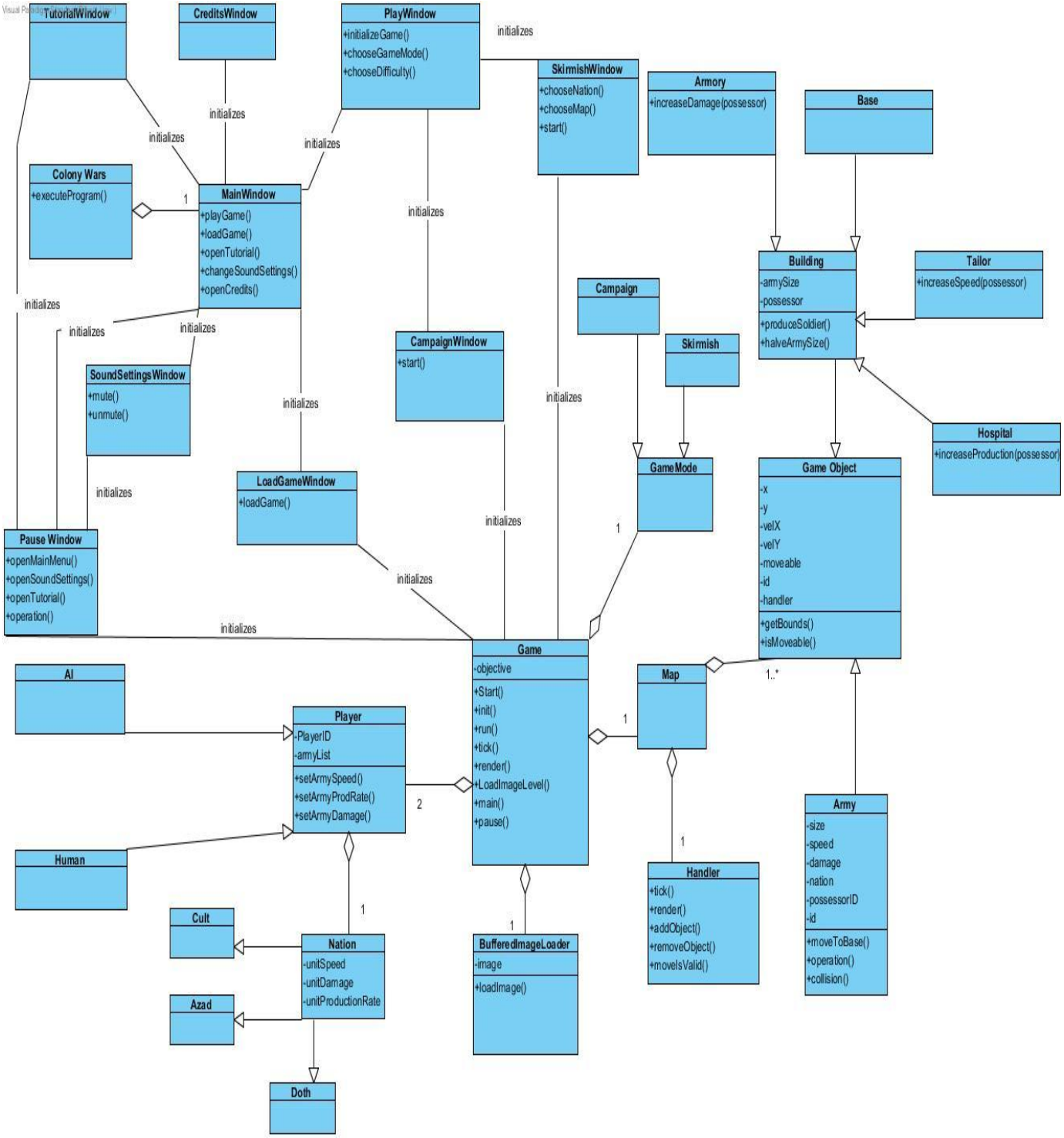
4.1.4. Change Sound Settings

Participating Actors: Player
Entering Condition: Player clicks the Sound button from the main menu.
Exit Condition: Player clicks the back button and goes back to the main menu.
Main flow of events:
1- Player clicks on the sound settings button and enters to sound settings menu.
2- Player can change the game volume as on or off from the sound settings menu.
3- The player clicks the back button and goes back to main menu.

4.1.5. Open Credits

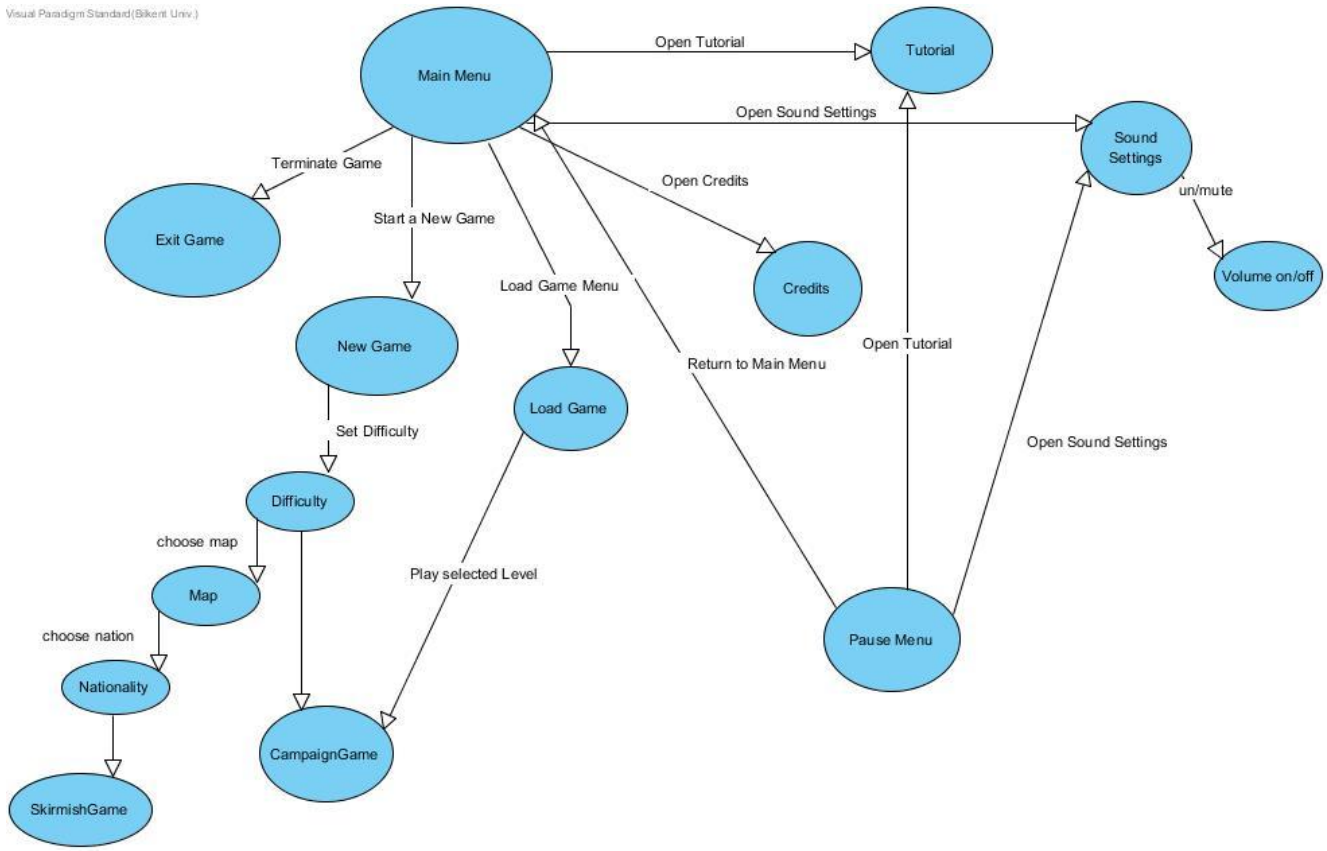
Participating Actors: Player
Entering Condition: Player clicks Credits button from the main menu.
Exit Condition: Player clicks the back button and goes back to the main menu.
Main flow of events:
1- Player clicks on the Credits button
2- The credits page pops up and the player reads the information about developers and the game.
3- The player clicks the back button and goes back to the main menu

4.2 Object and class model



4.3 User Interface

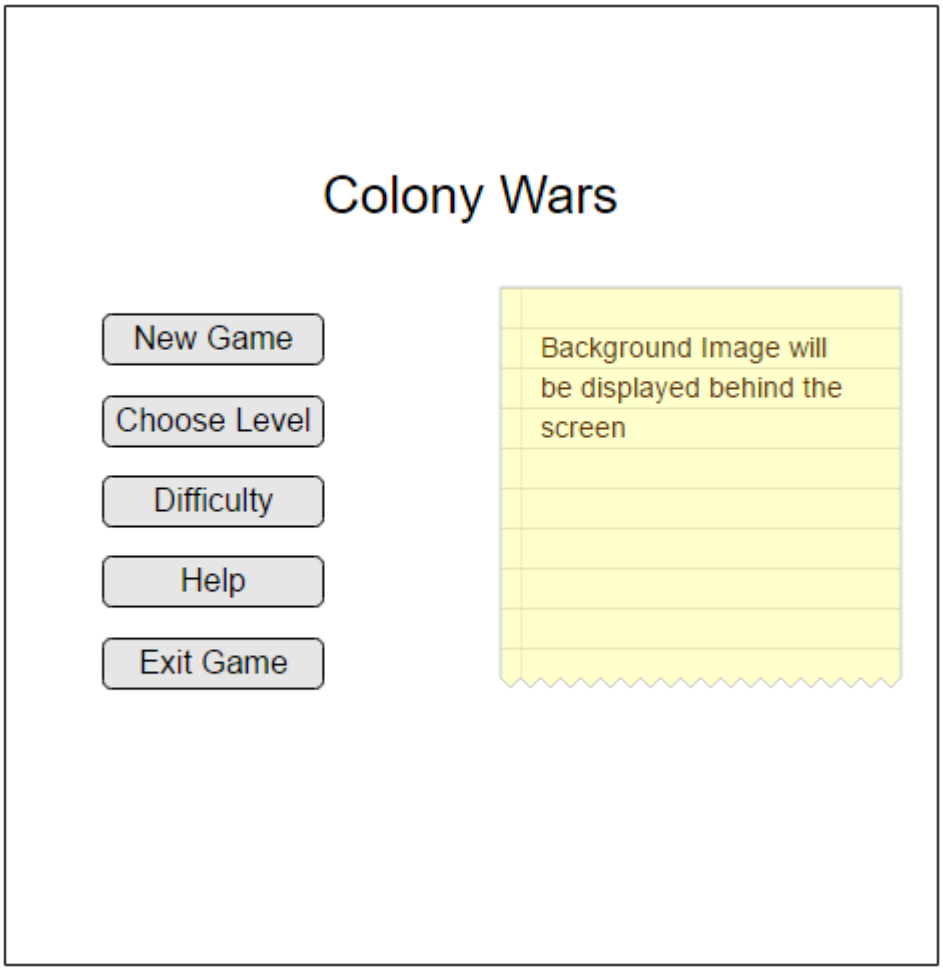
4.3.1 Navigational Path



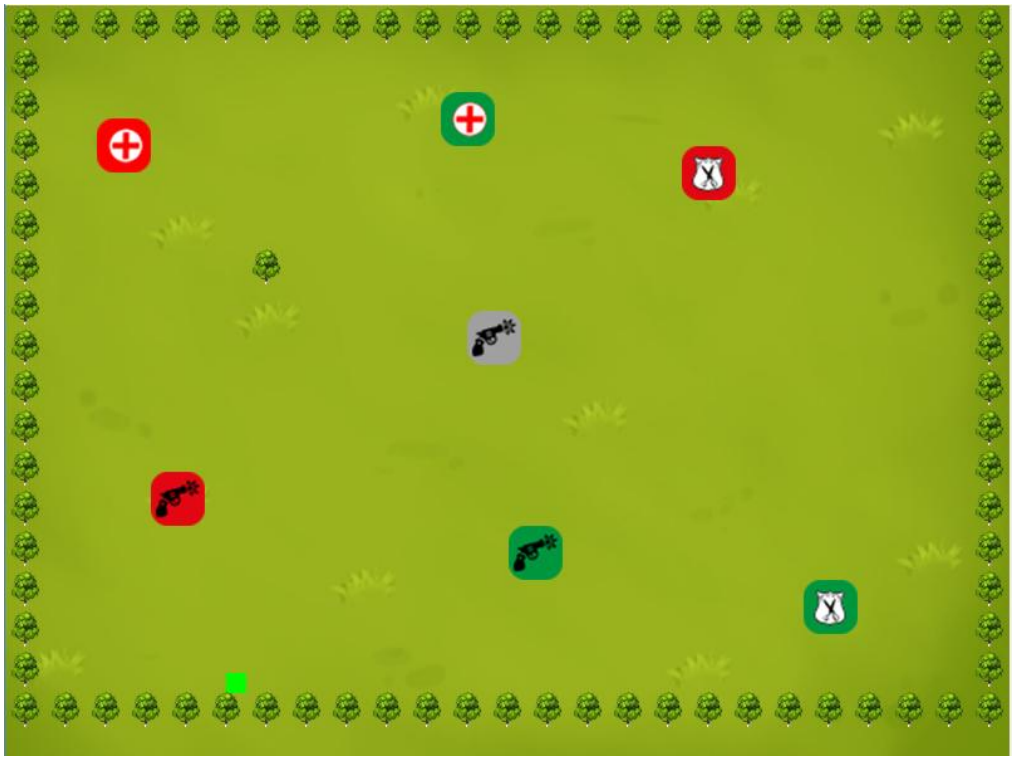
4.3.2 Screen Mock-ups

4.3.2.1 Main Menu

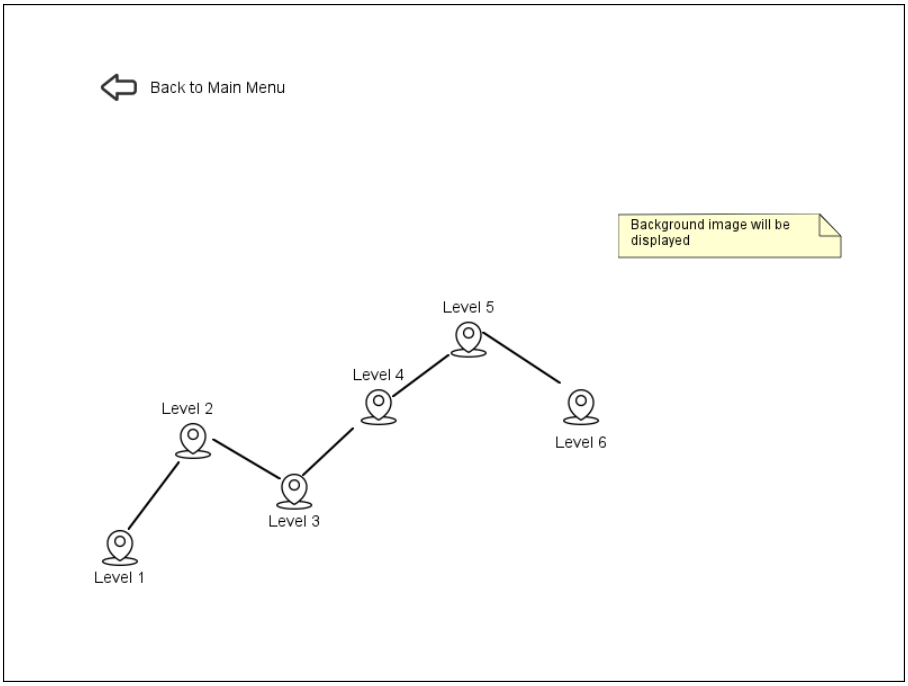
When the game stars, main menu will come and player can choose 5 different option to proceed. Which are New Game, Choose Level(only active if player played game first), difficulty, sound and help.



-New Game : When this button pressed a new game(level 1) or skirmish game will start with selected difficulty level.



-Choose Level : If player played the game before then he/she can start the game from list of levels he/she played but cannot choose the levels that has not been played yet.



-Difficulty : This button will bring a new menu where they are 3

options of difficulties(easy, medium, hard) and depends of the selected setting game will be adjust to that difficulty level.

-Help: Instructions about the game will be displayed here.

-Sound: When this button clicked it will whether turn on or off the music

4.3.2.2 Bases

There will be 9 base image that will be used for this Project. There are 3 main bases(Hospital-Tailor-Armory) and each can have 3 different color according to who occupied it. If no one occupied these bases the main color of the building will be gray, if player have the base then green will be coloured and if computer has it the base will be red.

Computer Hospital: 

Idle Hospital Base: 

Player Hospital Base: 

Computer Tailor Base: 

Idle Tailor Base: 

Player Tailor Base: 

Computer Armory 

Idle Armory Base: 

Player Armory 

4.3.2.3 Soldiers

Only two identical soldier will be on the game with different color.

4.4 Dynamic model

Player has two different main way to initialize the game ; loading the game saved previously or initializing a new game. If player chooses to start a new game , there are two options to select the game mode ; vsBot and campaign. Player have to choose difficulty and nationality if vsBot or campaign game mods are chosen. Also the player is able to choose map if he has chosen not campaign but vsBot mode. After initialization , game starts. During game , bases of the players are check at every time unit. If the player has no bases and other player (the bot) has the total number of bases , game is over - lost-. Then he is back to main menu. If neither the player not the other player(-bot) haven't collected all the bases yet , players try to take over all the bases . Who has taken all bases, wins the game. Any time game is over , player goes back to main menu.

Activity diagram of the game is below:

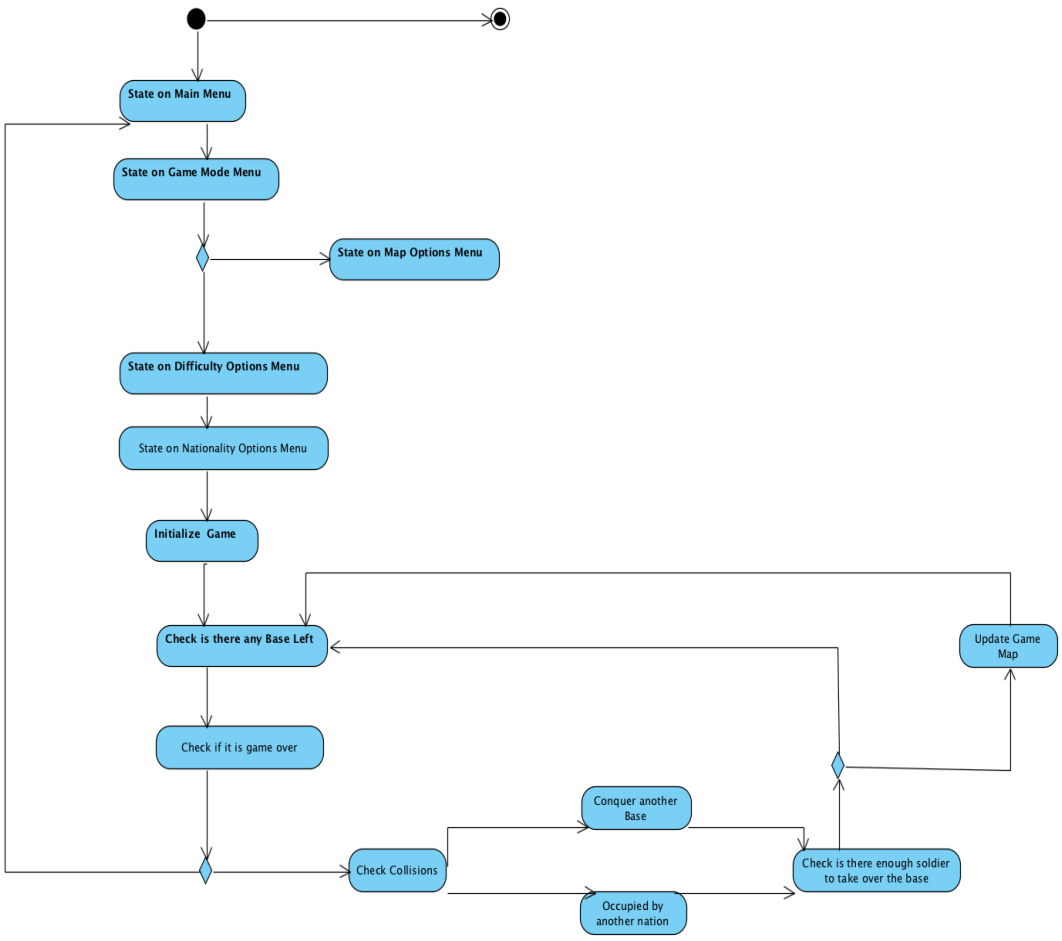


Diagram 1 : Start Skirmish Game

This diagram describes start game scenario. Player(Actor) selects start game in the main menu. When the player presses “Start Game” button , ‘playGame()’ method in MainWindow class is invoked. Player then choses difficulty and selects the game mode as skirmish which invokes skirmishWindow. Skirmish Window creates the game and the game creates Player, GameMode and Map according to given attributes. Game class then calls render and tick mehods on map refreshing it constantly. Map creates Handler class and Gameobjects. Game calls its run method recursively to keep the game going.

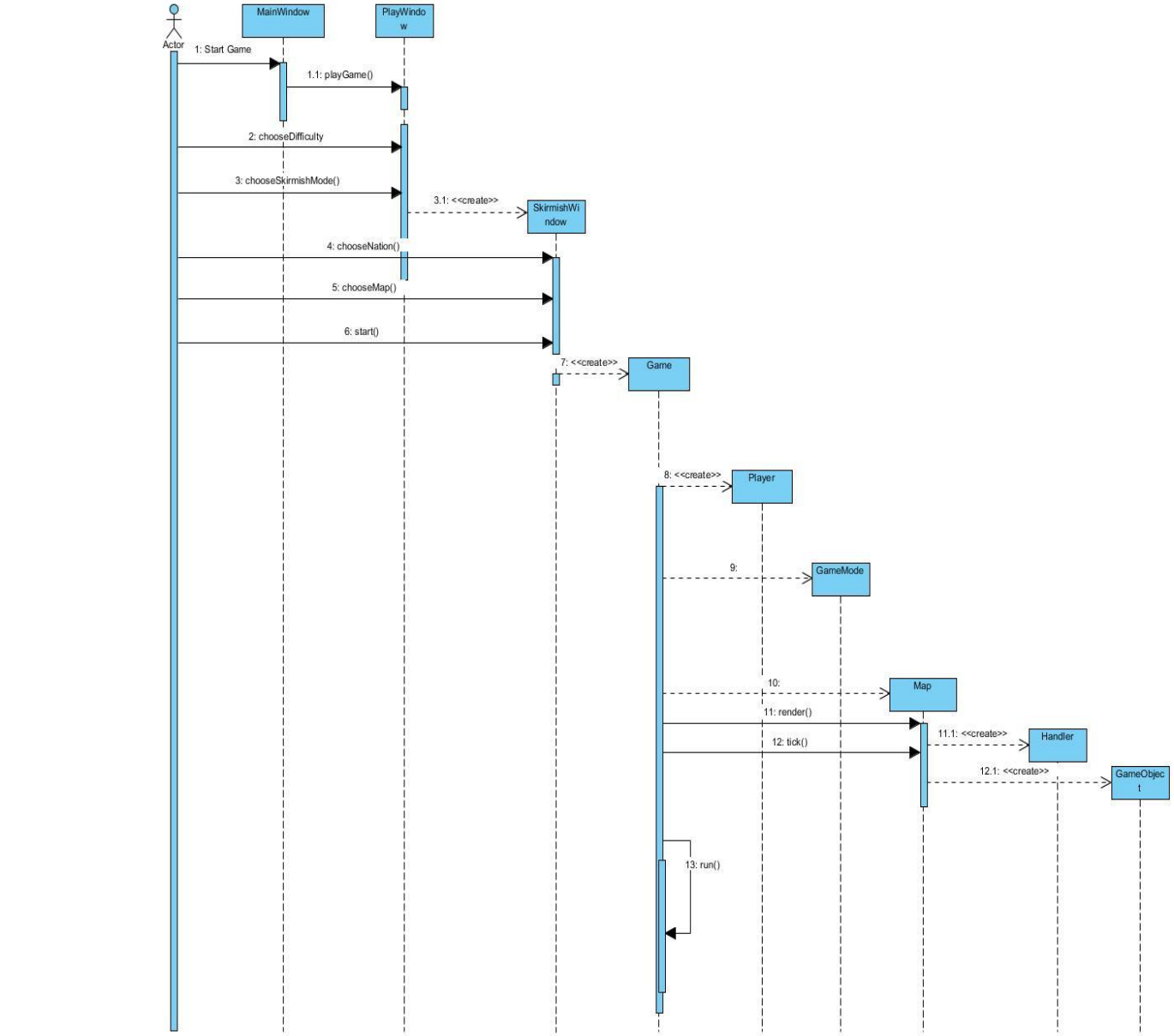
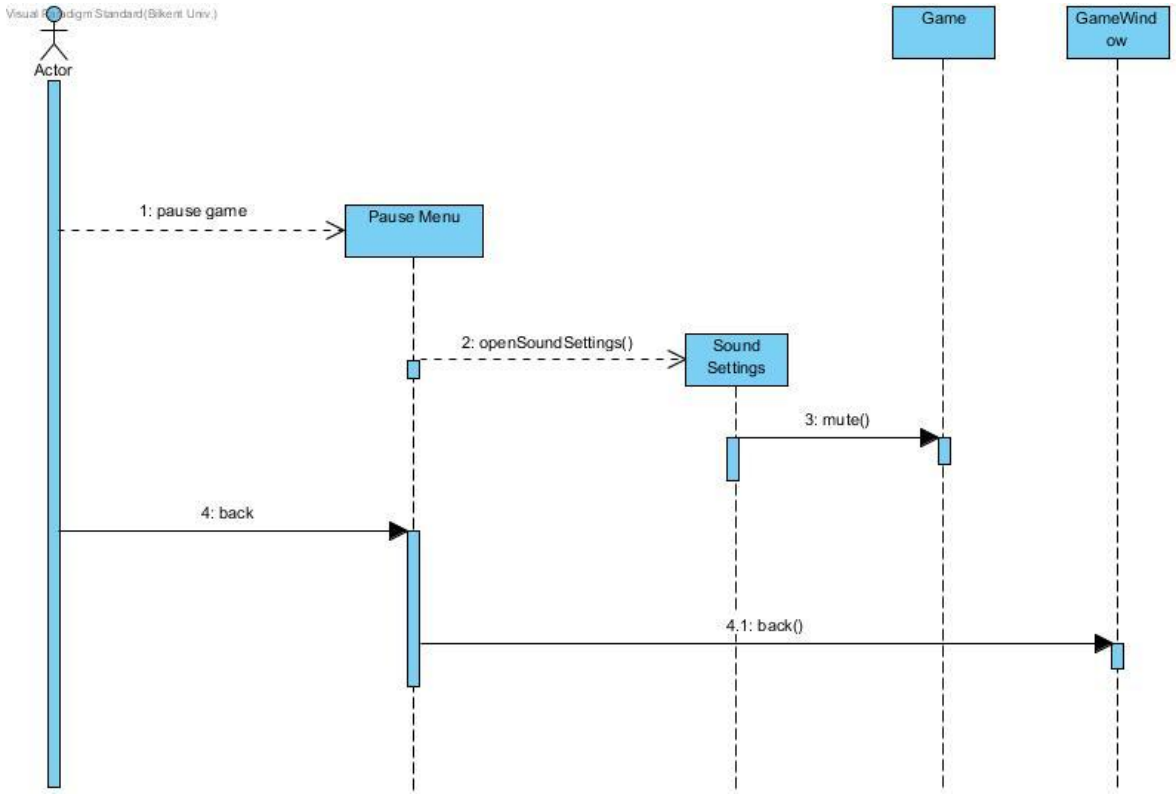


Diagram 2 : Change Sound Settings

This diagram describes the scenario that the player mutes the sound of the game. The Player presses the specific key to open Pause Menu. This move triggers ‘pauseGame()’ method in Game. Then Game invokes ‘openPauseMenu()’ method. The player selects “Sound Settings” from menu and then mutes the sound. After all, he selects “Back” from Pause Menu. It invokes ‘back()’ method in PauseMenu. Pause menu closes leaving the GameWindow open.



5. Supporting Information

5.1 Appendixes

TBD