**Introduction**

The requirement analysis and development needs for the Walls and Warriors project will be explained in this document that is being implemented as a project for the Object Oriented Programming course by the group of 5.

Document is intended to describe the subsections of the development process. These subsections include overview, functional and non-functional requirements, system models, mockups and improvements list to clarify the status and needs of the project.

Overview and gameplay part will be depicting the final product in mind, its gameplay, attributes and features from the perspective of a user. More detailed, systematic and technical definitions will be given in the proceeding sections.

Functional requirements will include main features of the game that a user will be interacting during the gameplay. These features are consisting of menu options such as starting a new game, selecting levels, options and game instructions and guidelines.

Non-functional requirements will be describing developer needs and responsibilities for a fluid gameplay and responsiveness. This part includes performance and stability, interactivity and responsiveness and compatibility with all the platforms.

System models will be the section explaining the different scenarios and user decisions during the game with different diagrams developing in accordance with the details. Use case diagrams will be explaining user decisions and consequent scenarios. Dynamic models will go further in detail to explain gameplay from a technical perspective. These models include sequence and activity diagrams. The object model explaining the classes, instances and their associations is included in the document describing the game’s structure from a developer perspective.

Mockups are attached to visualize the final picture for better understanding and expectations for the project which will finalize the requirement analysis.

Improvements list will be summarizing the additional features that distinguish this project from the classical board game to increase user affinity and game complexity for a better experience.

**Overview**

This section is intended to explain the general overview and gameplay of the game and key features that is associated with the gameplay. Additionally, main attributes and parts of the gameplay will be explained in subsequent subsections to describe functionalities of these attributes. Map is one of the features that will be explained with added features and main functionality which will be the most frequently interacted part of the game.The objects included in the map, such as the walls and warriors and their functions will be described in the proceeding subsections. Menus that are interacted to play the game will be explained with all the aspects. Finally, settings for the gameplay that are customized by the user will be described in the last subsection.

**Gameplay**

Walls and warriors is a strategic board game that is being implemented in 2D for the desktop environment via this project. Game is consisting of a map that is populated with warriors of different kinds and walls for keeping the enemies outside the borders of the castle of a certain kind of warriors. Main responsibility of the user is to find a way to place the walls in such a manner that they do not overlap and the final castle includes only the warriors of one kind, mainly, blue warriors. The board game has a fixed size map and 2 kinds of warriors, blue and red, and 4 types of walls with different shapeswhich a user can pick one of each at most. The game always has one solution which means there is only one way to place the walls in such a manner that the user will win. This project is based on these key features with more additional ones for increased complexity. These additional features include different sizes of maps, immobile objects and obstacles on the map, different shapes of walls, new functionalities of certain warriors and more which will be explained in more detail in the improvements summary. With the additional features, game may have more than one solution and a certain algorithm will be used to determine if the game is over or not.

**Map**

Map is the fundamental part of the game that all the action is happening on. Map is the base for the walls and warriors to be placed on. With the additional features, in the proceeding levels, map may have natural immobile objects such as lakes, forest or rocks to challenge the user to come up with a solution that circumvents these obstacles.

**Walls**

Walls are the fundamental parts of the final castle which will determine if the user wins or not. With many shapes that satisfies different scenarios, user will be responsible for selecting a wall from a menu to place on the map. The walls will have some strength that may be decreased with the attacks of the enemy warriors which challenges user to think and play faster before they collapse. The strength feature will be included in harder and more complex levels.

**Warriors**

Warriors are the main characters of the game user interacts with. The placement of the warriors is generally fixed and determine the solution of the game. This project will be granting some warriors with new features to walk in certain directions for the blue warriors and attack the walls for the red warriors which adds a time constraint. These new features will increase the complexity in the harder levels.

**Menus**

In the game there are 2 menus that a user interacts with. This subsection will be describing these 2 menus in the following paragraphs.

**Main menu**

Main menu is the first screen that welcomes the player. The menu includes buttons to start playing, reading the guidelines and instructions, customize the game settings, display the credits and to quit the game. User needs to select the play option to start the game and select the level.

**Walls Menu**

In the game, there will be a walls menu that displays the different kinds of walls a user may pick. This menu may present more options in accordance with the complexity of the current level.

**Settings**

Settings menu will allow the user to customize the settings for the desired gameplay. The customizable settings will include adjustment options for the game sounds (SFX) and background music.

**Functional Requirements**

This section includes the requirements and expectations for the key features of the game that a user will be interacting with. These features include main menu, play, how to play, settings, credits and quit options.

**Main menu**

Main menu is the main screen that will be encountered by the user after the execution. This menu will let user to choose one of the options that are being displayed and need to respond accordingly. The main menu will be displaying play, how to play, settings, credits and quit options and will respond with the pre-defined functions for different scenarios.

**Play**

To start the game, user needs to select the play option which will bring the available levels in response. Levels will be available consecutively after a successful gameplay in the previous level. After choosing a level from the available ones, user will be welcomed with the map that comes with the warriors placed beforehand. The size of map, placement and functionalities of the warriors, features and available shapes of the walls will be determined by the game in accordance with the level complexity. For that reason, features and availability of the walls, functions of the warriors and obstacles on the map will be calculated beforehand to guarantee at least one solution. User will be interacting with the walls via a mouse or other tracking device and drag and place the walls in a manner that they do not overlap and all the obstacles are ignored. Game will detect whether the selected wall can be placed on a certain location that user desires with considering conditions and limitations explained above. Game will be dynamically checking if the user has won yet or not and will respond immediately if the game is over either by a win or because of collapse of the walls. Finally, proceeding level will be made available to play if user won.

**How to play**

How to play section will describe the gameplay, rules and instructions as its name suggests. Basic rules of the board game and features of the walls and warriors with the additional ones will be explained thoroughly in this section.

**Settings**

This settings section will allow the user to customize the game setting for a desired experience. These settings will allow the adjustment of the game sounds (SFX) and background music volume. Additional options may be included such as brightness or graphics.

**Credits**

Credits will display the information about the developers and the contributors of the project. Their names will be displayed on the screen. Some additional information about the developers may also be included in addition to the names such as linkedin accounts or github repository of the project.

**Quit**

As the name suggests, quit option will allow the user to quit and terminate the game from the main menu. This option will appear only in main menu but a terminate button will also work any time to close the window and terminate the process immediately.

**Non-functional requirements**

Main intention of this section is to explain the non-functional requirements which serve the developer responsibilities. These responsibilities include performance, interactivity and compatibility issues of the game.

**Performance**

The game should not bottleneck the CPU and use many resources of the system. Lags and performance decrease or FPS drops should be solved for the final product. Main reason to choose the Java language for the implementation is to use tools and features of the JVM to tackle these issues with the help of garbage collector, for example. Screen tearing should be avoided. In other words, game should maintain a FPS rate accordance with the display in use to not to respond with a higher refresh rate that is used by the displaying device. An ideal 30-60 FPS rate should be maintained to fix any probable bugs related to this issue as it is the case with most of the monitors in use and considering the idleness of the game. CPU usage should also be reasonable for a smooth experience and sound production. Finally, game should not use many data resources and be as compact as possible for a smaller and more desirable disk and memory usage.

**Interactivity**

Game’s responsiveness is another challenge to be tackled for a smooth user experience. User interface also should be as simple as possible to increase satisfaction of the customer. Response time of the game for the monitor and mouse should be reasonable for these purposes.

**Compatibility**

The game should be compatible with all the desktop platforms and operations systems in use. Java’s JVM tackles this issue as it comes with its own VM to run the Java programs.

**System Models**

**Mockups**

**Improvements summary**

To challenge the user and increase the complexity of the game, many additional features have been added to the classical board game. These features will be explained in the proceeding subsections which will describe the additions to the map, walls and the warriors.

**Map**

**Different Sizes**

Unlike the board game, this game comes with different sizes of the maps to increase the complexity. Levels will start easy and advance accordingly. One of the advancements include the increase in the map’s size which will allow it to house more warriors and therefore, more walls and obstacles.

**Immobile Objects**

As the map expands, immobile objects may be generated accordingly to increase complexity and challenge the player. These objects include lakes, forest and rocks that cannot be moved and need to be ignored to build the castle. Lakes may house ships of different warriors. Blue ships should be in the castle’s area while the red ships should remain outside. In that scenario, walls may contain the blue ship by the chains that are attached to the walls.

**Walls**

**New Shapes**

In the board game there are 4 wall shapes and they all can be placed on a single fixed-size map. Because of the varying sizes of maps in this game, new wall shapes are also introduced accordingly.

**Chains**

Like the walls on surface, there may be chains available to replace the walls’ functionality in the water. In certain scenarios where there is a lake on the map and that houses 2 different kinds of ships, chains may be used to house the blue ship in the castle’s field. These chains will appear in these scenarios on the walls menu and should be attached to the walls by the both ends.

**Strength**

Walls in the harder levels will have some strength levels that can be decreased by attackers on the red ships. This same rule applies to the chains as well which may be damaged and collapsed by the attackers. This rule adds a virtual time constraint and challenges the user think and decide faster to win the game. If the strength of the walls decreases to 0, game is over and player loses.

**Warriors**

**Attackers**

As mentioned in the previous subsections, in the maps that house a lake with red ships, there will be red warriors, in other words enemies, who will attack and try to collapse the walls or the chains. These warriors will be seen in the harder levels.

**Walkers**

Rather than being in the same location and fixed place, some blue soldiers will be walking in certain direction. Player needs to place the wall accordingly to contain these walking warriors. Like the attackers, these warriors will appear in the more complex levels.

**References**

Balsamiq – Mockup Software

<https://balsamiq.com>

Visual Paradigm – Diagram Design Software

<https://visual-aparadigm.com>