Software Requirements Specification

Team 9

Matthew Stein	**260355809**
Oruj Ahmadov	**260523568**
Romain Nith	**260571471**
François Parent	**260533275**
Casimir Désarmeaux	**260467441**

McGill University

Department of Electrical and Computer Engineering

February 9, 2014

Table of Contents

1. Introduction

1.1. Purpose and scope

The purpose of this Software Requirements Specification document is to promote a better understanding of the goals of the project being undertaken. This is done by clearly expressing the requirements that need to be met in order to complete the project successfully. It is intended for both the client as well as the software engineers responsible for the project. The client should be able to determine from the document that all his needs are being met and the finished product will be consistent with his expectations. The software engineers will use this document to ensure they are meeting the needs of the client as they design the software.

The aforementioned project is the design of the game *Bomberman.* Bomberman is a classic video game consisting of a wide variety of versions. In its simplest form, the player, represented by Bomberman, has to navigate through a maze while trying to kill enemies, earn power-ups, and destroy blocks through the strategic placement of bombs. A version of this game is to be programmed that includes all the basic features of the game as well as some more novel and interesting components. There will be options for both single and multiplayer game play. The single player will try to pass a number of levels to ultimately win the game. In multiplayer the players will try to win the game by defeating their opponents. The system will communicate solely with the user and will have no connection to other programs on the host computer or any access to a network.

1.2. Definitions

Java Runtime Environment (RTE): A program that allows any software with a Java extension to run on the computer.

Integrated Development Environment (IDE): A program that includes a source code editor, a compiler, and usually a debugger to facilitate easier software development.

Leaderboard: A score board showing the names and current scores of the leading competitors.

Graphical User Interface (GUI): Graphical interface of a computer that allows users to drag and click objects with a mouse rather than entering text at a command line.

1.3. References

1. "Bomberman." - *StrategyWiki, the Video Game Walkthrough and Strategy Guide Wiki*. N.p., n.d. Web. 09 Feb. 2014. <http://strategywiki.org/wiki/Bomberman>.

2. "Play Free Game Space Bomber." *Y8.Com*. N.p., n.d. Web. 09 Feb. 2014. <http://www.y8.com/games/space_bomber>.

3. "RTE." *The Tech Terms Computer Dictionary*. N.p., n.d. Web. 09 Feb. 2014. <http://www.techterms.com/>.

4. *Software Requirements Specification Template*. N.p., 9 Feb. 2005. Web. 09 Feb. 2014. <www.tricity.wsu.edu/~mckinnon/cpts322/cpts322-srs-v1.doc>.

1.4. Overview

This document consists of an Overall Description as well as the Specific Requirements of the product. The Overall Description includes the perspective of the product as well as its functions and use cases. This section also includes a description of the users being targeted, any constraints on the product, any assumptions being made and any dependencies. The Specific Requirements section includes lists of all the functional and non-functional requirements. It also includes any constraints specific to the design as well as a number of additional requirements that are neither functional nor non-functional.

2. Overall description

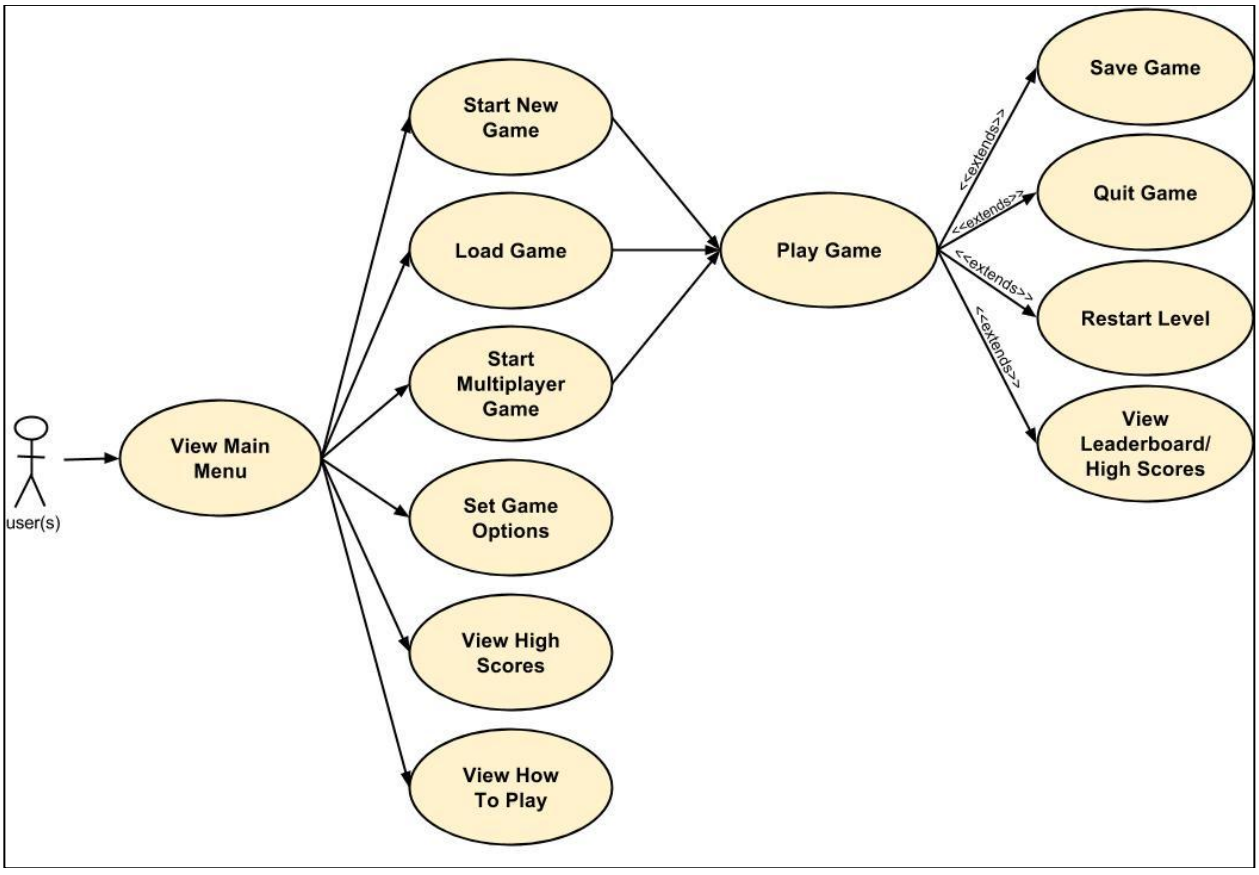
2.1. Product Perspective

This product will be a desktop application with a graphical user interface (GUI). It will be written entirely in Java and will require a Java Runtime Environment. It can be executed from a Java IDE or from the command line. It is a system on its own, so no communication with participants other than the user is needed.

The User Interface: The user will be able to directly interact with the system via the GUI. The user will be able to perform actions using the mouse, keyboard, and track-pad.

2.2. Product Functions and Use Cases

The game will be playable in 2 modes: single player and multiplayer. Each mode will have its own features and options. The single player mode will provide a saving/loading option, and the multiplayer will include a leaderboard after a finished game. The product functions are explained throughout the following Use Cases.



Use Case 1	View Main Menu (Launch Game)
Participating actor(s)	User(s)
Entry conditions	• User launches the application from the desktop • User quits a game • User has pressed 'main menu' from the post game screen • User has pressed 'back to main menu' from the 'view high scores', 'how to play', or 'set game options' screens
Flow of events	1. System displays main men 2. User has the choice to either start a new game, load a previously saved game, start a multiplayer game, set options, view high scores, or view a tutorial
Exit conditions	• Dependent on the users choice, the system displays the appropriate screen

Use Case 2	Start New Game
Participating actor(s)	User
Entry conditions	• User selects 'new game' from the main menu
Flow of events	1. System asks the user to enter a username 2. System creates a temporary 'account' for the user's future saved game and high scores 3. System displays a screen with gaming options 4. User chooses the interface 5. User chooses difficulty (easy, medium, difficult)
Exit conditions	• User presses 'play' and the game starts

Use Case 3	Load Game
Participating actor(s)	User
Entry conditions	• User selects load game' from the main menu
Flow of events	1. System display one or more previously saved games to be continued 2. User chooses the desired saved game
Exit conditions	• User presses 'continue game' and the game starts from the moment the game was previously saved • If no games have been saved, user is redirected to the main menu

Use Case 4	Start Multiplayer Mode
Participating actor(s)	2, 3 or 4 users
Entry conditions	• User selects 'multiplayer game' from the main menu
Flow of events	1. User chooses a number to determine the number of players 2. User selects difficulty (easy, medium, difficult) 3. User selects interface
Exit conditions	• User presses 'play' and the game starts

Use Case 5	Play Game
Participating actor(s)	User(s)
Entry conditions	• User selects 'Play' from either the new game screen, the multiplayer screen or loads a game
Flow of events	1. The game starts at stage 1 if the user starts a new game or is playing multiplayer 2. If the user has loaded the game, then the game starts at the stage corresponding to the last time the user played 3. User attempts to complete all the levels of the game 4. User is lead to a post-game screen at the end of the game (the user either completed all levels or ran out of lives)
Exit conditions	• User completes all the levels of the game and presses 'view high scores' if he has a high score and was playing in single mode • User plays in multiplayer mode and presses 'view leaderboard' • User presses 'restart game' and the game restarts from stage 1 • User presses 'main menu' and the user is redirected back to the main menu

Extension 5.1	Save Game
Participating actor(s)	User
Entry conditions	• User is playing in single player mode and selects 'save game' from the Pause menu
Flow of events	1. A brief menu is displayed and the user can either choose, to 'save', 'save and quit' or 'cancel'
Exit conditions	• User presses 'save', the game is saved and the current game continues from the moment it was paused • User presses 'save and quit' and the game is saved. The user is then directed to the main menu • User presses 'cancel' and the current game continues from the moment it was paused, without being saved

Extension 5.2	Quit Game
Participating actor(s)	User(s)
Entry conditions	• User presses 'quit' from the Pause menu
Flow of events	1. A menu appears asking if the user is sure about quitting
Exit conditions	• User presses 'yes' and is directed to the main menu • User presses 'no' and the current game continues from the moment it was paused

Extension 5.3	Restart Level
Participating actor(s)	User(s)
Entry conditions	• User presses 'restart level' from the pause menu
Flow of events	1. A brief menu appears asking if the user is sure about restarting
Exit conditions	• User presses 'yes' and the game restarts the level from the beginning • User presses 'no' and the current game continues from the moment it was paused

Extension 5.4	View Leaderboard / High Scores
Participating actor(s)	User(s)
Entry conditions	• User was playing in single player mode and has completed all the levels of the game and has a high score • Multiplayer game has ended
Flow of events	1. If the game was played in multiplayer mode, the system displays the leaderboard of all the participants in the game 2. If the game was played in single player mode, the system displays the high score list, with the user's just achieved score highlighted
Exit conditions	• User presses 'back to main menu' and is directed back to the main menu • User presses 'restart game' and the game starts back at Stage 1

Use Case 6	Set Game Options
Participating actor(s)	User(s)
Entry conditions	• User selects 'game options' from the main menu
Flow of events	1. System displays the game options 2. User chooses desired game options
Exit conditions	• User presses 'back to main menu' and is directed back to the main menu

Use Case 7	View High Scores
Participating actor(s)	User
Entry conditions	• User selects 'view high scores' from the main menu
Flow of events	1. System displays a list of the users previous high scores
Exit conditions	• User presses 'back to main menu' and is directed back to the main menu

Use Case 8	View How to Play
Participating actor(s)	User
Entry conditions	• User selects 'how to play' from the main menu
Flow of events	1. System displays a tutorial of the game, showing the rules and the controls
Exit conditions	• User presses 'back to main menu' and is directed back to the main menu

2.3. User Characteristics

This game should be appealing to a variety of users of any age. Therefore it should be relatively easy to understand the core principles of the game. The controls must be simple to get use to (e.g. the arrows of the keyboard are control the movements of the character). Those who are familiar with the game should feel comfortable playing it, and those who have not should be able to do so after playing one game.

Since the software supports a multiplayer mode, two players at most can play in a cooperative way throughout the levels. They will be able to do so by assigning each movement and action key to any key on the computer's keyboard.

2.4. Constraints

As asked by the client, the game must be appealing to any users. Therefore, no additional knowledge is required to use the interface.

The software must be able to run on any modern computer; no special hardware requirements are needed. Since the game must run on any average computer, it should use as little as possible memory and processing power.

As recommended by the client, the higher-order language selected for this project is Java, due to its strict language that can minimize any bugs in the code.

2.5. Assumptions and dependencies

One assumption about the product is that the computer will require a Java Runtime Environment in order to execute the game. If this is not the case, the game will fail to launch.

Due to the time dependency of the software development, some features may or may not appear in the game. However, the core features are certain to be implemented.

3. Specific Requirements

3.1. Functional Requirements

1. A user interface (main menu) that allows people to start a new game, play multiplayer, control sound (on and off options), select different game interfaces (to be defined), see high scores (only for single players), load saved games and to exit.

    1.1. The options of the interface can be selected using the arrow keys and enter key, the track-pad, or the mouse. (Essential, easy)

2. If a user selects New Game, the user should enter a username in the new interface (at least 1 character is required to continue). The difficulty level option should then appear and the user must choose easy, normal or hard. Then the game should start. (Essential, medium)

3. When multiplayer menu is selected, the user must select the number of players (2,3 or 4). Then, each player should enter a name. Afterwards, the difficulty levels should appear and one level should be chosen for all (easy, normal or hard). (Desirable, easy)

4. In the multiplayer tab, the game will start at level one. If a player uses all his lives he is removed from the game. If a player completes the stages it is the next player's turn to play the stage. Once every player completes the stage or is killed, the next stage will start. (Essential, medium)

5. In multiplayer mode, the score of each player is tracked separately. When all the stages are completed or every player is dead, a special leaderboard appears, showing the scores and showing the winner. (Desirable, medium)

6. Graphical implementation of the game including Bomberman, enemies, blocks, bonus drops, weapon drops, movements, explosion animations and environment wraparounds. (Essential, medium)

7. Game board should consists of a long rectangle containing stone blocks in such a way, that each stone block is separated by one tile, and is at least one tile away from the border of the rectangle. (Essential, medium)

8. Game controls to allow player(s) to move (up, down, left, and right), to plant a bomb, to pause the game and to use power-ups. (Essential, easy)

9. Implementation of Bomberman's movements, bomb planting, impact detection, dropping power-ups and movement of enemies. (Essential, medium)

10. Enemies must die when bomb explosion radius partially or completely covers them. When all enemies are dead, the portal is activated. Once the player goes through the portal, the level is completed. (Essential, difficult)

11. Bomberman losses one life if an enemy touches him or if he is caught in a bomb explosion. (Essential, easy)

12. The user can pause the game at any time by pressing the ESCAPE button. When paused, the player can choose to restart the current stage, go back to the main menu, save the game or continue with the current game from where he stopped. (Essential, medium)

13. After the player finishes a game, in single player mode, the program should give him the option of going to the leaderboard or the main menu. (Desirable, easy)

14. If the "exit" tab in the main menu is selected, a pop-up should appear asking the user if they really want to quit. (Essential, easy)

3.2. Non-functional (Quality) Requirements

1. Bomberman starts each round in the upper left corner the stage. (Desirable, easy)

2. It should take 5 seconds or less to load each stage. (Essential, easy)

3. An increase in stage number should correspond to an increase in the number of regular enemies, causing the game to become more difficult. In the first stage there will be 2 regular enemies when the difficulty level is easy, 3 regular enemies when the level is medium and 4 regular enemies when level is hard. There will be an increase of 1 regular enemy after each three stages for all difficulty levels.

    Regular enemy:

    Balloom(pts. 100, 1 block/second, smart low, wall pass no)

    3.1. The first super enemy will appear in the 4th stage and will be replaced by the following super enemy after each 4 stages according to the order given below for all difficulty levels.

        Super enemies are:(essential, medium)

        Oneal (pts 200, sped mid, smart mid, wallpass no)

        Doll (pts 400, speed mid, smart low, wallpass no)

        Minvo (pts 800, speed fast, smart mid, walpass no)

        Kondoria (pts 1000, speed slowest, smart high, wallpass yes)

4. The player cannot pass the stage until it has been completed. (Essential, medium)

5. When the space bar is pressed, bombs should drop to the center of the quadratic tile on which Bomberman stands or on the tile that he occupies most. (Essential, medium)

6. The bomb should explode 5 seconds after it has been planted. Bomberman cannot plant another bomb unless the previous one has exploded. When the bomb explodes, its blast will cover each adjacent block forming a "+" sign. Exceptions will be made to all the above with power-ups. (Essential, medium)

7. The bomb must also kill Bomberman if he is in the explosion radius (exceptions on power-ups). (Essential, medium)

8. Bomberman's speed should be (2) block per second (exceptions on power-ups). (Essential, medium)

9. Enemies' speed should be (2) block per second (exceptions on special enemies). (Essential, medium)

10. The bomb's explosions should last (1.5) seconds and every enemy or player who makes contact with that explosion during that time dies. (Essential, medium)

11. At the beginning of each level, Bomberman should be safe from immediate encounters with enemies. Meaning, there should be no less than 1 block, in all directions leading to Bomberman, between Bomberman and an enemy. (Essential, easy)

12. Bomberman should move in the defined directions (up, down, left and right). Bomberman should move as long as the player holds the appropriate button, but he should stop as soon as the button is released or an obstacle (blocks or bombs) is in his way (exception on power-ups). Bomb planting should be bound to a key and an animation should follow. (Essential, difficult)

13. Enemies should only move in the defined directions (up, down, left and right). They should follow a defined path, unless an obstacle blocks their way (exceptions on super enemies). If blocked, they should return to the initial position of their path along the same path and restart the movement. (Essential, medium)

14. There should be 2 kinds of blocks, "tough" and "soft". Bombs will affect the "soft" blocks only. The "soft" blocks should disappear when covered by a bomb explosion. (Essential, difficult)

15. The smart enemies (mid or high) will not follow a predetermined path but aim for the player if he is within a certain distance of them. The high smart enemies will have a greater detection range. (Desirable, difficult)

16. Some enemies can pass through brick walls. If the wall is destroyed while they are in it, they are also destroyed. (Desirable, medium)

17. At the beginning of levels that have a super enemy, a pop-up screen appears showing an image of the new monster and explaining its particularities. This happens only if the option "tooltip" has been selected in the option menu. (Optional, easy)

18. The possible power-ups will be. (Essential, medium)

    Bombs (+1 bomb can be on the stage / bomb power-up) (-1 if dies)

    Flames (+1 square of range / fire power-up) (-1 if dies)

    Speed (only 1) (Bomberman moves (4) blocks per second)

    Wallpass (lose on death) (can pass through brick walls)

    Detonator (lose on death) (Can detonate oldest bomb)

    Bombpass (lose on death) (can pass through placed bombs)

    Flamepass (lose on death) (Immune to explosions)

    Mystery (time limit) (immunity to all, cannot die)

Design constraints

1. Should appeal to users of all ages (Essential, easy)
2. Should have well defined and simple controls promoting easy use, even for someone with little to no computer experience. (Essential, easy)
3. There should be 3 different kinds of interfaces that the user can choose from. (Optional, medium)
4. Nominal interface should be desktop application with GUI developed using Java swing components (Essential, easy)

Other requirements

1. To be handed in by April 13th (Essential, easy).
2. Conduct all communications with the customer in a professional manner. (Essential, easy)
3. Keep agendas, record meeting dates and times and keep minutes (Essential, easy).