SOFTWARE REQUIREMENTS SPECIFICATION

ECSE 321 Introduction to Software Engineering
Term Project Phase One Deliverable

Simon Ho, Raymond Guo, Henry-Michel Cantave, Jeffrey Tichelman, Laurent Jacob, Frank Luong

Team Name TBD

TABLE OF CONTENTS

1. INTRODUCTION

1.1 Purpose

The purpose of this software is to provide entertainment to its users. The software described in this document is the game bomberman. The game needs to provide an easy to use interface. Consequently, the controls need to be simple and fluid. The game also provides extensive statistics for users to see their progress in skills and their ranking compared to other users. The game also has a creative aspect, since players will be able to create their own maps and save them.

The targeted audience is people of all ages. Because the purpose of this software is to build a game with an easy to use interface, anybody older than the age of 5-6 years old should be able to play. This version of the software will also assume an English-speaking audience.

1.2 Scope

1.2.1 Product Name

The name of this software is Bomberman.

1.2.2 Product Capabilities

Anybody should be able to use this software. People will be able to gain skills and get better at the game. This software provides players with a simple but complete user experience, since statistics and gameplay are built to be as extensive as possible. The game will provide the player with multiple modes such as multiplayer and single player story mode.

1.2.3 Objectives & Goals

The goal of the Bomberman software is to provide a complete Bomberman experience. Players need to be able to be creative and develop their own maps and game types. They also need to be able to level up and face stronger enemies. They need to be able to learn the controls really fast and these should be easy to use. The design and gameplay should combine for an overall pleasant experience for any user within the target demographic.

1.3 Definitions, Acronyms & Abbreviations

- Enemy: Take bomb blasts to destroy; AI “monster” or a boss
- Game State: the mode of gameplay the user is currently happening (either Menu Selection or Gameplay)
- Gameplay: the tactical aspects of a computer game, such as its plot and the way it is played (distinctive from the graphics and sound effects)
- In-Game: It is the state when a player is actively playing the game or navigating through the menu of the game
- Level: A session or iteration of the game defined by map layout and gameplay difficulty, the user should advance through progressively more difficult levels in single-player mode
- SRS: Software Requirement Specification
- User: A player/User-Player

1.4 References

Rules: **http://strategywiki.org/wiki/Bomberman**
Format: **http://www.site.uottawa.ca/~shervin/courses/seg4105/project/SRS.html**

1.5 Overview

The rest of this document will layout the general description and the specific requirements of the project. Section 2 describes the various factors and considerations relevant to the software's design. It will also attempt to summarize the functions of the game and the typical events within the system. Section 3 will list explicit requirements the project should meet. The aim is to encompass all the required functional and performance requirements of the system but this entire document is subject to alteration and possible addition in response to unforeseen circumstances as the project unfolds. It's simply meant to portray our understanding of the project at this initial stage to ensure a harmony with the customer's intentions.

2. OVERALL DESCRIPTION

2.1 Product Perspective

It's a game designed for the Introduction to Software Engineering course (ECSE 321). This game is a continuation of the strategic game Bomberman. It was first developed by Hudson Soft in 1983.The project will have a lot of similarities like its predecessor since the game rules are almost identical. The game will be a self-contained software system.

The behavior of Bomberman, his enemies and the power-ups are replicated in our game. However, the game will be modified in such a way that it will be more challenging and enjoyable. We will implement several additional features such as variable opponent combat stats (health, speed, bombs, etc.), custom map builder, multiplayer, story mode and more. The game will end when all the enemies are killed or if the player fails to complete the mission within the given time limit.

2.2 Power-ups

The game will include several power-ups in order to increase Bomberman's skills. Those special powers can be obtained by destroying a certain brick or killing a specific enemy. Power-ups can effectively help the player to finish a level faster by increasing one or more of their capabilities. The following are some of the powerups we intend to add:

- Speed Boost: Increase Bomberman's speed.
- Double Bomb: Allow Bomberman to drop a second bomb.
- Big Explosion: Increase the ranges of all the bombs.
- Throw-the-bomb: Allow Bomberman to plant a bomb from a far distance.
- Extra Life: Grant Bomberman an extra life.

2.3 Product Functions & Use Cases

2.3.1 General

This product allows users to play the game, and record a corresponding name and score, as well as comparing scores with previous players on the same account. The software will be a video game playable on any Operating System. It also consists of inputs through the keyboard and outputs through the screen and speakers.

When the game is started, the system will be presented a menu from which a user can prompt their name and navigate into starting a level. When the player finished the level, the system will automatically record their scores. Once the score has been recorded, the system will then update the leaderboards.

The game will consist of two different states which are Menu Selection and In-Game.

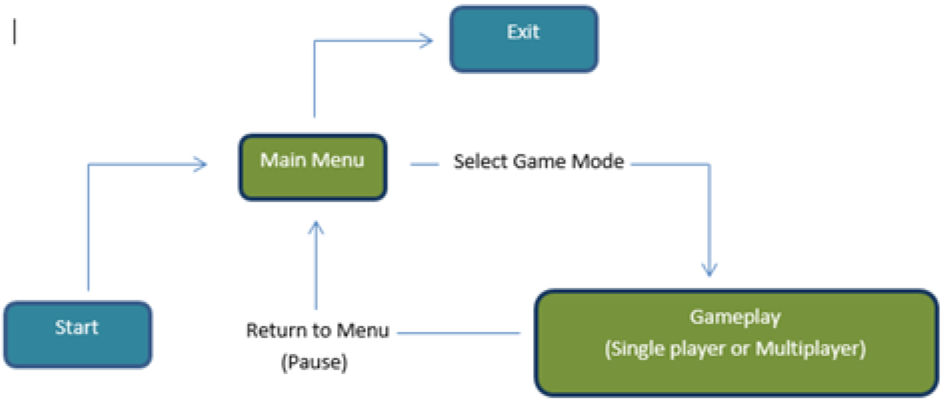

Figure 1: State Transition Diagram

Basically, the figure above describes the behavior of our system. Once the user has started the game, the user can browse the menu and select the one of the options of the interface described in section 2.3.2. After the selection of the game mode, the player can start exploding the monsters In-Game.

2.3.2 User Interface

The software makes it possible to play the game alone or against another user, depending on the mode selected. A graphical user interface will be displayed to the user. This user interface is mainly controlled with a mouse, but can also be controlled with a keyboard.

During the Menu selection state, a user will have the following actions:

- Scroll up
- Scroll down
- Select

By default, these activity will be assigned to the up, down and enter keys, respectively, on the keyboard. The user can alternatively select one of the options graphically using the mouse.

During the In-Game state, the user interface will have a head-up display (HUD) that shows the number of lives remaining, the scores, and the current stage/level of the game.

Graphical User Interface:

- Main menu screen (starting screen/title screen) that contains the button for Play Game and Options
- Create a new user: allow users to register their name.
- Select game mode: Allow users to choose the mode of the game. (Single player/Multiplayer)
- New game: Allow the user to start the game from the very beginning.
- See achievements: Allow users to see the high score list with statistics
- Settings: Allow the user to adjust the settings of the game.
- Load game: the user can load a saved game.
- End game
- Close program

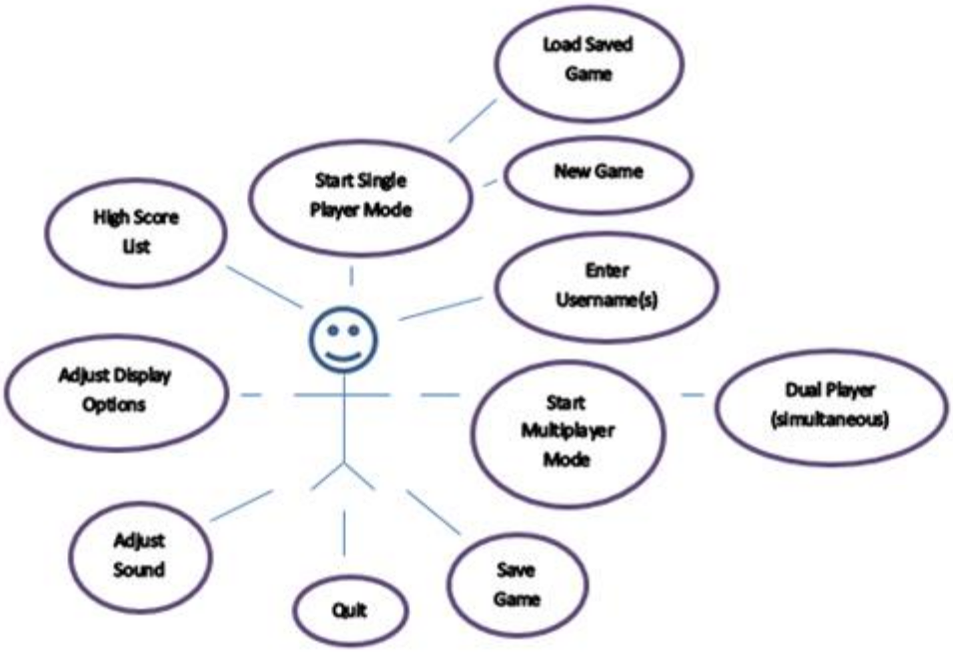

Figure 2: Use Case - Menu

2.3.3 Gameplay & Game Control

Gameplay can be single player or multi-player where several players play the same game alternately. Also, there will be a dual player simultaneous mode where two players can play the game on the same map and at the same time.

Originally, the Player will control Bomberman and eliminate enemy characters in the stage by placing bombs in the passage in the pursuance of concealing them. The player can also blast away the Soft Blocks in order to reach the enemy or obtain special powers. To do this, the player must use the keyboard to issue commands to Bomberman on the game board. Also, when an enemy is destroyed, it might drop a power-up too. A player will lose a life whenever an enemy touched him or a bomb blasted him. It is important to note that all enemy will have an Artificial Intelligence agent to implement the behavior stated above. The game will end when all the enemies are killed or if the player cannot end the mission within a given amount of time.

Specifically, the user stories/controls are:

- Control the game character motions: the player can move the character with the directional pad (Control pad) on the keyboard
- The user can set bombs, perform special abilities and skills by pressing the 'Space' key on the keyboard
- Pick up a power-up: Player can collect the special powers from the enemy or a soft block.
- Kill enemy: Player should eliminate the enemies on the path in order to win. Certain enemy can drop powers-up.
- Pause Menu: Allow the user to pause the game at any time by pressing the pause button whose default binding is 'Esc'. It also allows access to a menu screen.
- Continue game: The user will be able to continue the paused game.
- Save game: the user can save the game.
- When the user finish the level, he will then proceed to the next level with stronger foes.

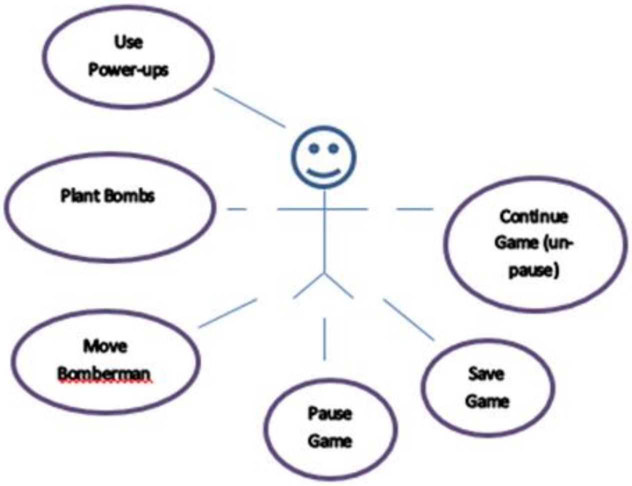

Figure 3: Use Case - Gameplay

2.3.4 Use Cases

2.3.4.1 Menu Use Cases

Use Case Name	Registration
Participating Actors:	User, Interface
Preconditions:	In the menu, user presses “Register”
Flow of Events:	1. Game asks the user to provide necessary information 2. User types his username and password 3. Game checks if username is unique 4. Game stores the user information
Postconditions	System authorizes the registration and notifies the user
Exceptions	If the username is not unique, a message will inform the user to try another one.

Use Case Name	Load Save Game
Participating Actors:	User
Preconditions:	User enters Menu Selection
Flow of Events:	1. User selects the “Load Game” option 2. User chooses a save file 3. The save file is updated after continuing gameplay
Postconditions	User is In-Game
Exceptions	The user did not save any game previously

Use Case Name	Start Multiplayer Mode
Participating Actors:	User(s)
Preconditions:	User enters Menu Selection
Flow of Events:	1. User selects the “Multiplayer Mode” option 2. Users select “Head-to-Head” otion 3. User selects “Start Game”
Postconditions	Users are In-Game
Exceptions	None

Use Case Name	Dual Player (Simultaneous)
Participating Actors:	Users
Preconditions:	User enters Menu Selection
Flow of Events:	1. User selects the “Multiplayer Mode” option 2. Choose “Simultaneous” option 3. Enters gameplay for Player-1 and Player 2
Postconditions	Users are In-Game
Exceptions	None

Use Case Name	Save Game
Participating Actors:	User
Preconditions:	User presses the “Esc” button
Flow of Events:	1. User selects the “Save Game” option in the Pause Menu 2. Enter save file name 3. Returns to the Pause Menu

Postconditions	Player returns In-Game or stop playing the game.
Exceptions	Save file name doesn't adhere to accepted file naming format

Use Case Name	**Continue Game (Un-pause)**
Participating Actors:	User
Preconditions:	The game is paused
Flow of Events:	1. The player has to press the "Continue Game" Button
Postconditions	User returns In-Game
Exceptions	If the game is not paused, the user cannot select the "Continue Game" key

Use Case Name	**High Score**
Participating Actors:	User
Preconditions:	User open the application (Bomberman Game)
Flow of Events:	1. User selects the "See Achievement" button 2. View Statistics (ranking)
Postconditions	User exits the high score list
Exceptions	None

Use Case Name	Adjust Volume/Controls/Display
Participating Actors:	User
Preconditions:	User enters Menu Selection
Flow of Events:	1. User has to select the “Settings” button 2. User can adjust the volume and other options
Postconditions	User exits “Setting”
Exceptions	User attempts to move the volume out of the allowable range but is denied

Use Case Name	Quit Game
Participating Actors:	User
Preconditions:	User enters Menu Selection
Flow of Events:	User selects the “Exit Game” button
Postconditions	User exits the Bomberman game
Exceptions	None

Use Case Name	Single Player
Participating Actors:	User
Preconditions:	User enters Menu Selection
Flow of Events:	1. User selects the “Single Player” option 2. Choose to start a new game or load a saved game 3. Enters gameplay for Player-1

Postconditions	User is In-Game
Exceptions	None

Use Case Name	**Select Game Mode**
Participating Actors:	User
Preconditions:	User is logged in
Flow of Events:	1. System ask which game mode to play (single player, multiplayer, custom, etc.) 2. User selects game mode 3. Player, Opponents are spawned and map is generated
Postconditions	Game Mode is initialized
Exceptions	None

2.3.4.2 Gameplay Use Cases

Use Case Name	**Move Player**
Participating Actors:	Player
Preconditions:	Keyboard plugged in Player must not be stuck in place Players intended path must not be blocked Player coordinates must be within the map
Flow of Events:	1. User presses an arrow key 2. Player animation switches from Idle to Moving 3. The player moves to the new position
Postconditions	Player position has changed

Exceptions	If the player is out of bounds, player doesn't move

Use Case Name	**Stop Player**
Participating Actors:	Player
Preconditions:	Player was previously moving
Flow of Events:	1. No keys are being pressed 2. Player "Stop" animation sequence plays 3. Player remains in-place
Postconditions	Player remains still at a set position
Exceptions	None

Use Case Name	**Plant bombs**
Participating Actors:	Player, Bomb
Preconditions:	Player is alive Cooldown of previous Bomb planted is over Player intended Bomb placement doesn't overlap an obstacle
Flow of Events:	1. User presses Bomb key 2. Bomb is planted on the direction facing the character 3. Bomb stays Idle for X amount of time
Postconditions	Bomb is planted and cooldown is started
Exceptions	If the player dies during Bomb Idle state, it continues its flow of events.

Use Case Name	Bomb Explosion
Participating Actors:	Bomb
Preconditions:	Bomb Idle state time is over
Flow of Events:	1. Bomb explosion propagates any or all of 8 directions 2. Bomb explodes with X squares of radius 3. Bomb decreases 1 lifepoint of any objects the explosion hit except hard blocks
Postconditions	Bomb is gone and cooldown resets Explosion animation is gone Player/Opponent dies depending if explosion hit someone Destroyed obstacle are removed
Exceptions	If the game ends before bomb cooldown, the explosion is not triggered

Use Case Name	Pick up Item
Participating Actors:	Player, Item
Preconditions:	Game mode selected has destroyable crates with items Item is reachable
Flow of Events:	1. Player walks on Item location 2. Item is picked up
Postconditions	Item is gone Player gains a power-up (speed, bomb range, etc.)
Exceptions	If the Player has already reached the maximum amount of power-ups, his stats remain the same

Use CaseName	Win level
Participating Actors:	Player
Preconditions:	Player is alive and all Opponents are dead
Flow of Events:	1. System prompts a “You Win!” text 2. System asks if Player wants to go to next level or return to main menu 3. System records the score
Postconditions	Player/Opponents are spawned in new map Every characters stats are reinitialized
Exceptions	If Player passed every level of the game, System returns automatically to Main Menu

Use Case Name	Death
Participating Actors:	Player, Opponents
Preconditions:	Bomb explosion hit Player/Opponents The character has no lifepoints remaining
Flow of Events:	1. Dead character disappears 2. System prompts a “Game over” text 3. System asks if Player wants to restart or return to main menu 4. System records the score.
Postconditions	Game menu is displayed
Exceptions	If the Player dies after every Opponents are already dead, the Player still wins and passes the level

Use Case Name	High Score
Participating Actors:	User
Preconditions:	User is logged in At least one game has been played beforehand
Flow of Events:	1. User selects “High Scores” in Main Menu 2. High Score and Statistics are displayed
Postconditions	Return to Main Menu
Exceptions	If no game has been played before, nothing is displayed since no scores have been saved

2.3.5 Audio

The game will have an audio resource to enhance the entertainment value of the game. Bomberman theme song will be played as the background music. Furthermore, there will be sound effects for the following actions:

- Menu Scrolling
- Menu Selection
- Bomb Explosion
- Killing an enemy
- Power-ups

2.4 User Classes & Characteristics

The main user class will be the user. The main target users of our product are casual players. Familiarity with a keyboard interface will be required to operate the software as intended. The user interface will be designed to make sure the user’s interaction is as simple as possible therefore, users with very little computer experience are still able to use the application. The game is also designed in a way that the control scheme is intuitive and user-friendly. In other words, we will try to make the game as easy to learn as possible. However, it may take some time to master the game.

2.5 General Constraints

- The project is time constrained due to the fact that the whole project has to be finished on April 13th, 2014.
- The project is done in groups of 5 or 6 persons.
- No game engine are allowed for the creation and development of our game.
- The usage of codes taken from any external resources is not allowed.
- Players of any age should be able to understand and play the game.

2.6 Assumptions & Dependencies

It is assumed that users are all familiar with the basics of using a computer. Also, the game's statistic tracking is dependent on an Internet source. The software can be found on the Internet and all games can be saved online. The game can be loaded from any computer that has access to the Internet or has previously installed the game. The game also assumes an English-speaking user.

3. SPECIFIC REQUIREMENTS

Note: requirements are divided into functional and quality categories and subcategorized by their context. Each requirement is listed and followed by its priority and difficulty in parentheses. There are three possible variations of each; essential, desirable or optional for priority, and easy, medium or difficult for difficulty.

3.1 Functional Requirements

3.1.1 Menu Screen

3.1.1.1 Main Menu

The Main Menu is the first interface presented to users after running the application. (Essential, Easy)

Main Menu state must include access to user statistics, load games, settings and game play selection. (Essential, Easy)

The Main Menu interface will include login username and password forms and a log-out button. (Essential, Easy)

Main Menu can only be re-accessed when game has finished or when players have agreed to quit current game. Otherwise, users can only access the In-Game Menu. (Essential, Easy)

The In-Game Menu must be implemented as a partial Main Menu, whose function is to provide users with saving options, settings and a quitting option during a game. (Essential, Medium)

3.1.1.2 Saved Games

Users must be able to save current game using the “Save Game” option from the In-Game Menu. (Desirable, Easy)

Saving games can only be done via the In-Game Menu, when the game is paused. (Desirable, Easy)

Saved games must only be accessible by selecting the “Load Game” button from the Main Menu. (Desirable, Easy)

3.1.1.3 Statistics

Users must be able to access high scores and user statistics by pressing the “Statistics” button displayed in the Main Menu. (Essential, Easy)

There must be a Highscore section which displays a ranking list of all users in ascending order. (Desirable, Medium)

Number of games played, games lost, games won, games left and K/D values must be listed under each user in the Personal Statistics section. (Desirable, Medium)

AI’s do not have statistics. (Essential, Easy)

3.1.1.4 Settings

Users can change audio, character, and control settings by clicking the “Settings” button in the main menu. (Essential, Medium)

Audio, including music and sound effects can be muted and unmuted. (Essential, Easy)

In-Game character colors can be changed for players and AIs. There will be a selection of 8 different colors. (Desirable, Easy)

User controls can be changed. Players can change the directional keys and bomb-drop key. (Desirable, Easy)

Users can optionally control the game with the mouse. (Optional, Medium)

3.1.1.5 Quit

Users are able to quit current game by pausing the game and clicking the “Quit button”. (Essential, Easy)

Once the game is quit, it cannot be re-accessed, unless if it was previously saved. (Essential, Easy)

3.1.1.6 Game Mode Selection

Users must be able to choose 1-Player game mode by selecting the “Single Player Mode” option. (Essential, Easy)

Users must be able to choose 2-Player game mode by selecting the “Multi Player Mode” option. (Essential, Easy)

Once “Multi Player Mode” is the selected the user must be prompted to choose between “Simultaneous,” or “Head-to-Head” play. (Essential, Easy)

3.1.1.7 Login Information

Users must log in to play the game. (Essential, Easy)

Users can log out when game is finished. (Desirable, Easy)

Users must create valid usernames: 15 characters or less, no symbols, single space allowed. (Essential, Medium)

Users must be assigned unique usernames and be forced to create such. (Essential, Medium)

Users must create valid passwords: 6 characters or more, must include a number, no spaces allowed. (Desirable, Easy)

Password form must show typed characters in symbols which hide the content. (Desirable, Medium)

Users cannot log in more than once at a time. (Essential, Difficult)

2 login forms will be shown. One for Player 1, the other for Player 2. (Essential, Medium)

All user credentials must be saved in a database. (Essential, Difficult)

3.1.2 Gameplay

3.1.2.1 Game Mode

1-Player mode must include multiple levels, for which users are appropriately challenged to pass. (Essential, Medium)

Game Levels will be characterized by their difficulty, which is affected by Enemy ability and map layout. Each subsequent level should be made progressively more difficult. (Essential, Difficult)

2-Player mode must allow simultaneous play for users working together against the enemy. (Desirable, Medium)

2-Player mode must allow two users to play head to head on a local machine. (Desirable, Medium)

Once the game is begun, the "Game Played" statistics will be increased by one for all Players on the field. (Essential, Easy)

3.1.2.2 Winning and Losing

All players and AIs will have 3 lives. (Essential, Easy)

Once out of lives, the character will be removed from the map. (Essential, Easy)

The round is finished when a single character remains on the field. (Essential, Easy)

Players can win the game by being the only character on the field. The winner will have an updated "Game Win" in their statistics. (Essential, Easy)

Players who are not on the field when the round is finished automatically lose. The losers will have an updated "Game Lost" in their statistics. (Essential, Easy)

Players quitting a game will have an updated "Game Left" in their user statistics. (Desirable, Easy)

3.1.2.3 Controls

Users must be able to control his/her character with keyboard controls. (Essential, Easy)

Users have the option to play with a mouse. (Optional, Medium)

Player 1 and Player 2 must have his/her own set of controls. (Essential, Easy)
User controls will be modifiable and saved through the In-Game settings when game paused. (Desirable, Easy)

3.1.2.4 Power-Ups

Power-ups must present themselves creatively and appropriately on each level. No level should have a surplus or shortage of power-ups that make it trivially easy or difficult. (Essential, Hard)

3.1.2.5 AI

AI must be implemented with an autonomous behaviour. (Essential, Difficult)

AI characters must have the same properties and constraints as the player's character. (Essential, Medium)

The AI should get progressively harder to defeat as one moves through the game. (Essential, Difficult)

3.1.2.6 Game Map

An in-game map must be generated to delimit the bounds of the field. (Essential, Medium)

Obstacles must be auto-generated to provide different possible layouts for each level, maintaining roughly the same difficulty. (Essential, Difficult)

Game maps will have different sizes. Ranging from 2x2 to 10x10. (Desirable, Medium)

Users can create and generate custom maps. (Desirable, Easy)

3.1.2.7 Collision

Collision detection must be handled for all in-game characters, blocks and bombs. (Essential, Medium)

Out-of-bounds detection must be handled for all characters with respect to the game map. (Essential, Easy)

3.1.2.8 Timer

The game must have a timer in order to manage the duration of a round. (Essential, Easy)

When the timer runs out the game must end with the user losing. (Essential, Easy)

3.1.2.9 Graphics & Sound

Graphics and animation must be used for characters, bombs, explosions, blocks, crates, power ups and map for entertainment and identification. (Essential, Medium)

Music and sound effects for explosions, power ups, and character deaths must be implemented. (Essential, Medium)

3.2 Quality Characteristics

3.2.1 Security

The log-in information must be encrypted. A user's password should never be compromised. (Essential, Easy)

To track one's statistics the user must first enter a valid username and password at the login screen. (Essential, Easy)

3.2.2 Usability

The game interface must be intuitive. The learning curve a new user needs to go through in order to enjoy playing the game must be kept to a minimum. (Essential, Easy)

The game should have a "help" section. This section will contain the answer to some frequently asked questions and instructions for how to play the game. (Desirable, Easy)

The user should be given a walkthrough before their first game which teaches them how to play the game. (Optional, Medium)

3.2.3 Reliability

In the case of failure, the statistic must not be corrupted. The reason why the system failed must be display to the user via a pop-up message box by mean of an error message and an error code. (Essential, Easy)

The saved game data must be kept for six months and remain uncorrupted. (Essential, Medium)

The user must progress fluidly between levels and the game must end conclusively. No exceptions should be met at any of these transitions. (Essential, Easy)

3.2.4 Performance

If the background music is ON, it must not slowdown the response time when the user enter a command (ex: plant bomb, move up, etc..). (Essential, Medium)

The map should generate in less than 5 seconds when the user starts a game. Every subsequent map for each level should also meet this requirement. (Desirable, Medium)

3.3 Design Constraints

Three notable constraints on the design of our game are the following:

- The game must fit and play well on a regular laptop screen between 11 and 17 inches.
- We cannot use any game engine. All functionalities must be built from scratch.
- The interface must be intuitive for any English speaker, regardless of age or technical ability.

3.4 Other Requirements

3.4.1 Portability

The game must be able to run on any modern operating system and web browser. (Essential, Easy)

The game must run equally well on most prevalent modern hardware. (Essential, Medium)

3.4.2 Legality

All pictures, effects and music used must not be used without explicit permission from the owner and customer. (Essential, Easy)

All the code used must be original unless explicitly permitted by the customer. The only exception, for now, is the Java Swing library. (Essential, Easy)