SE 3XA3: Systems Requirements Specification: Revision 1

Team 03, Pongthusiastics
Adwity Sharma - sharma78
Arfa Butt - buttaa3
Jie Luo - luoj3

December 8, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
October 7, 2016	1.0	Created the SRS document
October 11, 2016	1.1	Completed SRS Rev-0 for deadline submission
October 24, 2016	1.2	Uploaded diagrams
October 31, 2016	1.3	Updated requirements
December 7, 2016	1.4	Updated document for Revision 1

1 Project Drivers

1.1 The Purpose of the Project

We have overtaken this project to improve the electronic version of ping pong game we found in an open source project on the Git website. The use of computers and other electronic devices for entertainment has been growing for several years now. The electronic version of the ping pong game used to be automatically installed with the installation of new windows. But for some years now Microsoft has stopped adding this game to the windows operating system. While, there are variations of this game available for download in the App store, there is still a market for new and improved ping pong games. Our game provides the players with a slightly challenging game that does not require any tools other than a computer that has a Java running environment. This game can be played among friends to relieve themselves of boredom. This project will also provide the users with a game that can be played by people of any age group.

1.2 The Stakeholders

1.2.1 The Client

An external entity is regarded the client for this project. This entity is interested in the redevelopment of the project. Though this project is very open ended and can be further developed in endless manners, there still are some restrictions that have been applied on the manner of development. This restrictions, in regards to the development has been applied by the

3XA3 team(3XA3 team defined in Table 2 of Section 1.4), and thus can be looked upon as the client figure.

1.2.2 The Customers

Any person with an access to a computer that has a Java running environment can be considered as a customer for this project. As this game does not have restrictions in terms of age, gender or any other socio-economic factor, everyone can be considered as the customer of this project.

1.2.3 Other Stakeholders

We are also the stakeholders for this project, as the development of this project and its result would be a great effect upon us. The general public can also be considered as the stakeholders as through them we are to gain customer base for our game. The rules and regulations of the redevelopment process have been set by the professor and the TAs, so they are also considered as stakeholders.

1.3 Mandated Constraints

- **Description:** The project is being developed in Java language in Eclipse and has to be run in a Java running environment.
  Rationale: Java running environments such as Eclipse are available online for free for users and thus can be downloaded by any user intending to play the game.
  Fit Criterion: There shall be a link provided with the release of this game to download a Java running environment, for users that do not have it already.

- **Description:** The project is provided online, in an open source environment, for anyone who wants to use it. Internet connection would be required to download the game the first time the player uses it.
  Rationale: The user should be connected to the internet that can assure the downloading of the game.
  Fit Criterion: As the game only needs to be downloaded once for the

user, it isn't very hard to accomplish. It only needs to be downloaded once and can be enjoyed for as long as the user intends to use it.

1.4 Naming Conventions and Terminology

Table 2: **Naming Conventions and Terminology**

Terms	**Definitions**
Users	Players of the game
The Project	The pong game that is being reconstructed.
Product	The game that is being developed.
Java	Java programming language.
Clients	The group for whom who the project is being developed for.
Git	The GitLab website.
Windows	Microsoft windows.
Customer	Anyone who would like to use this game.
3XA3 team	Professor, course coordinators, teaching assistant and any other personnel responsible for running of the 3XA3 course.
Single Player mode	The original ping pong game with all the rules from the original project.
Advanced Player mode	A new mode with a bomb added to the game.
Bomb	Red-colored ball that is to be avoided by the players when playing Advanced mode.
Game scene	The window with the game and in-game options.

1.5 Relevant Facts and Assumptions

Facts: There will be approximately 1000 lines of codes for this game by the time this game is completed. The original code of the game that is being reconstructed can be found in https://github.com/mihneadb/Pong.

Assumptions: It is assumed that the user will have some access to the internet to download the codes that will be used to run this game when the user wishes to play this game for the first time. It is also assumed that the person will either have access to a Java running environment or be willing to download a Java running environment. Since internet is readily accessible this should not be much of a trifle.

2 Functional Requirements

2.1 The Scope of the Work and the Product

The developers would rewrite the code into a structured design. By doing this, the work for future maintenance would be easier.

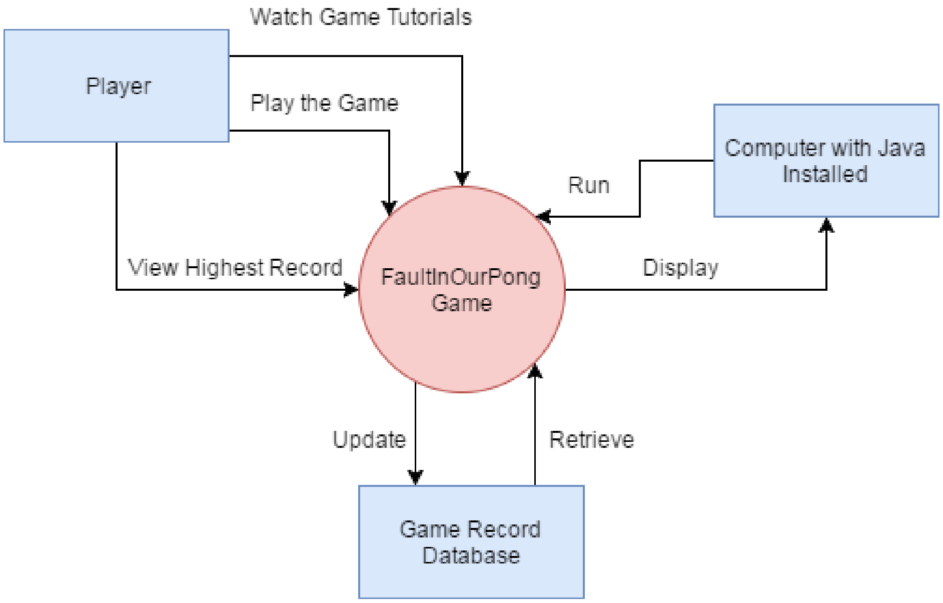

Figure 1: Context Diagram for the Game

2.1.1 The Context of the Work

The software application to be developed will follow the same game rules as the original Pong game application, with the exception of the bomb mode. Bomb mode will have a second red-coloured ball; a bomb, which the players have to miss while not missing the original black ball at the same time.

The interface of the new application would also be improved. There will be additional features added through an in-game options panel. Additionally, the player will be given lives so that there will be a specific termination to the game. Along with the lives, at the end of each game, the player will also be notified of the amount of time they have played.

2.1.2 Work Partitioning

The Pong project will be following a model similar to the Water Flow development process. The project will start with developing a problem statement for the new project. It can help team-mates clarify the objective of the work, such as what to improve on the Pong game, and how much we can improve. Team-mates then would like to create a development plan for the Pong game, to make sure each development process is on track and on time. A requirement specification plan is then created to further break down the project into smaller tasks. This include specifying the requirements for the new software, stating the personnel that is related to this Pong project, and clarifying any concerns and issue related to it. A Verification and Validation Plan would be created in the following, to give team members and other stakeholders a vague idea of what the final product would be. Design Specification Document would be formulated for better clarifications before application implementation. Team-mates would be coding following all the documents mentioned, and the Verification and Validation Plan will test the program when it is completed.

The team of three will be working on the redevelopment of the project. Each team member will be assigned different tasks to work on the project. A Gantt Chart will be used to keep track of working activities for each member in the team, that means each member would have to make a mark on the Gantt Chart when they complete a task.

2.1.3 Individual Product Use Cases

- Case: User performs mouse click on the Start New Game button
  Result: The program would show the New Game page with the option to start a Single Player mode or an Advanced Player mode.

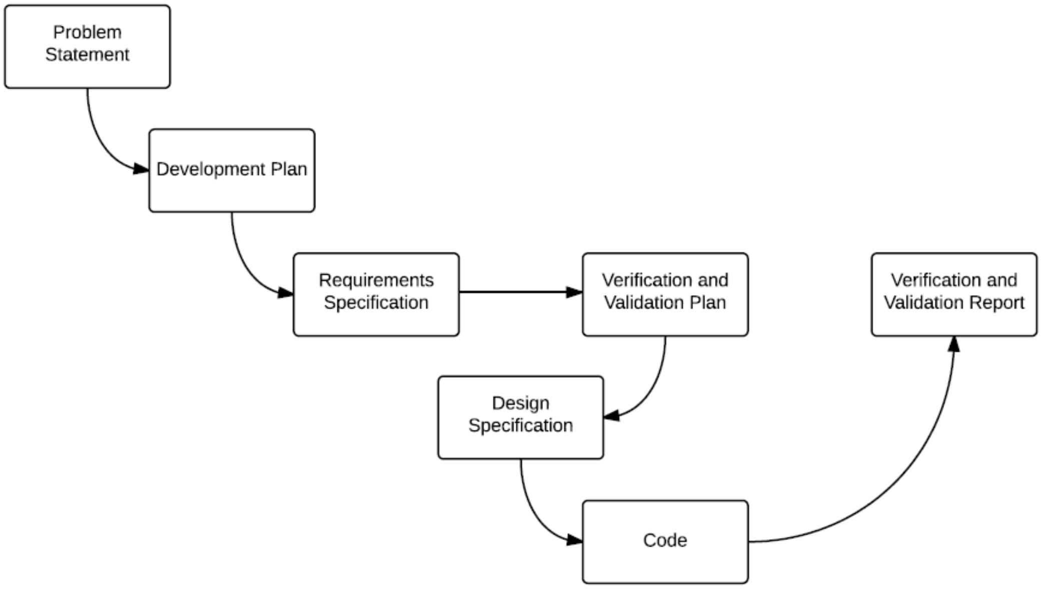
Problem Statement
Development Plan
Requirements Specification
Verification and Validation Plan
Verification and Validation Report
Design Specification
Code

Figure 2: Project Flow for the Redevelopment Process

- Case: User performs mouse click on Single Player Mode button
  Result: The program would start a Single Player mode game and the game scene would be displayed.

- Case: User performs mouse click on Advanced Player Mode button
  Result: The program would start an Advanced Player mode game and the game scene would be displayed.

- Case: User performs mouse click on the Save Game button
  Result: The program would save the players' record, and the game itself would be paused until the user clicks resume.

- Case: User performs mouse click on Pause Game button
  Result: The program would pause the game.

- Case: User performs mouse click on Resume Game button
  Result: The program would resume the game.

- Case: User performs mouse click on Exit to Main Page button
  Result: The program would exit the game scene and go back to the Main Page.

- Case: User performs mouse click on the Load Game button
  Result: The program would load the previously saved game.

- Case: User performs mouse click on High Scores button
  Result: The program would display the top MAX_HS number of high-scores.

- Case: User performs mouse click on the Tutorial button
  Result: The program would display a description of the game and teach the user how to play it.

- Case: User performs mouse click on the Exit button
  Result: The program game would stop, the window of the application would be closed, and the current game result would not be saved.

- Case: User taps on LEFT and RIGHT keys on the keyboard
  Result: The player would be able to move his/her paddle in the game.

2.2 Functional Requirements

Requirement number: FR1

The program should have a button to start a new game.

Rationale: The user should be able to start a new game in order to play the game.

Priority: High

History: Created October 11, 2016

- **Requirement number:** FR2

  The program should have a button to load a previously saved game.

  Rationale: The user must be able to load his/her previously saved game in order to continue the game.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR3

  The program should have a button to save the current game.

  Rationale: The user must be able to save their current game in order to continue playing it at a later time.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR4

  The program should be able to display a list of top MAX_HS high scores (name and their time).

  Rationale: The user must be able to open up the high scores list, so that they can see what rank they are on. This will also build healthy competition among the players.

  Priority: High

  History: Modified December 7, 2016

- **Requirement number:** FR5

  The program should have a button to open up the tutorial page, which will give instructions to the user on how to play the game.

  Rationale: If the user is playing the game for the first time, they must be able to view the instructions to the game before they can play it.

Priority: Medium

History: Created October 11, 2016

- **Requirement number:** FR6

  The program should have a button to have the option to play a single-player mode, with the computer. The list of high scores will be kept for this mode.

  Rationale: The user must choose which type of mode of the game they want to play, before they can play the game. This type of mode is for users who want to play alone, or want to beat a certain high score.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR7

  The program should have a button for the option to play an Advanced player mode, with the computer.

  Rationale: The user must choose which type of mode of the game they want to play, before they can play the game. This type of mode is for users who want to play the single-player mode, but with a little added difficulty.

  Priority: Medium

  History: Modified December 7, 2016

- **Requirement number:** FR8

  At the start of every game, the player should haveMAX_LIVES number of lives.

  Rationale: The user must have a specific number of lives at the start of the game, as that is the standard that we have decided to keep for the game.

Priority: High

History: Modified December 7, 2016

st

Requirement number: FR9

At the start of an Advanced player mode, the computer should have MAX_LIVES number of lives.

Rationale: The computer must have the same number of lives as the player to keep the game fair.

Priority: High

History: Modified December 7, 2016

- - **Requirement number:** FR10

    The user should be able to move their paddle left and right using the keyboard.

    Rationale: The user must be able to move their paddle left and right in order to play the game.

    Priority: High

    History: Created October 11, 2016

- **Requirement number:** FR11

  Whenever a player misses a hit, their life should be decreased by 1.

  Rationale: The number of lives of each player should decrease by 1 each time the player misses the ball, to keep track of the number of lives the player has left. This is to make sure that the player does not go over MAX_LIVES.

  Priority: High

  History: Modified December 7, 2016

- **Requirement number:** FR12

  The program should have a button to pause the current game.

  Rationale: The user must be able to pause the current game so that if they need a short break, they can take one without it affecting the progress of their current game.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR13

  The program should have a button to resume the current game (if the game is paused).

  Rationale: The user must be able to resume the current game so that if the game is paused, it can be resumed from the point where the player left the game.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR14

  The user should have the ability to exit back to the main page without terminating the whole program.

  Rationale: The user must be able to go back to the main page from the current game so that if they are not satisfied with their game, and would like to play a new game, they may do so by just going back to the main page and starting a new game.

  Priority: High

  History: Modified December 7, 2016

- **Requirement number:** FR15

  Whenever one player misses the ball MAX_LIVES times, the game should end as that players last lost all of their lives.

  Rationale: The standard for this game has been set so that each player gets MAX_LIVES lives, therefore the game needs to end once the player has used up all of their lives.

Priority: High

History: Modified December 7, 2016

- **Requirement number:** FR16

  Once a game ends, the user should be taken back to the main menu page.

  Rationale: The user must be taken back to the main menu page so that they can choose which mode of game they want to play, or if they want to access the high scores list.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR17

  In single-player mode if the player made it into the high scores list then, at the end of the game before the user is taken back to the main menu, (s)he will be prompted to enter their name so that they could be added into the list.

  Rationale: The user must be prompted for their name if they break one of the high scores, so that the list of high scores is kept up to date.

  Priority: High

  History: Created October 11, 2016

- **Requirement number:** FR18

  At the end of the game, a window should pop up to indicate the winning and losing state.

  Rationale: The user need to be notified the end of the game.

  Priority: High

History: Created October 31, 2016

3 Nonfunctional Requirements

3.1 Look and Feel Requirements

- **Requirement number:** NFR1

  The game interface should be colorful and vivid.

  Rationale: The user would be interested in the game if its interface is attractive.

  Priority: Medium

  History: Modified December 7, 2016

- **Requirement number:** NFR2

  The game frame should be clear for user to play.

  Rationale: The user should know where the border of the game frame is, in order to move his/her paddle to the proper prositions

  Priority: High

  History: Created October 31, 2016

3.2 Usability and Humanity Requirements

3.2.1 Ease of Use Requirements

Requirement number: NFR3

The game can be played by nighty-nine percents of people who know how to use a computer with hands.

Rationale: This game utilizes only mouse and keyboard.

Priority: High

History: Modified October 31, 2016

- - **Requirement number:** NFR4

  The game is easy enough to for a child greater than x years old to play. Some practice on the part of the users might be required to play and excel at this game.

  Rationale: This game should be available for most of the children because the game is focused on the children.

  Priority: High

  History: Modified December 7, 2016

3.2.2 Accessibility Requirements

Requirement number: NFR5

The game can be accessed and executed on majority of computing devices.

Rationale: Computers with Java installed should be able to open this game. Meanwhile, as long as there is an internet connection, this game is available for download.

Priority: High

History: Modified December 7, 2016

3.3 Performance Requirements

- ### 3.3.1 Speed requirement

  Requirement number: NFR6

  There has to be some space in the user's computer to download the game and a Java running environment. The game should be played in a fast computer as it would decrease the entertainment factor if it is lagging.

  Rationale: As long as the computer with this game has enough space for caching for Java, the speed of the game should be fast enough for playing.

Priority: High

History: Modified December 7, 2016

- ### 3.3.2 Safety critical requirement

  Requirement number: NFR7

  The program should be able to finish in reasonable amount of time.

  Rationale: Over-playing the game might cause the user to encounter some strain in the user's eyes.However, this is true for all games that require an electronic screen to play. Other than this, there are no other health concerns.

  Priority: Medium

  History: Modified December 7, 2016

- ### 3.3.3 Precision requirement

  Requirement number: NFR8

  The movements of the paddle should be as accurate as the actions performed on the keyboard.

  Rationale: Actions performed on the keyboard would directly affect player scores and satisfiability of the game.

  Priority: Medium

  History: Modified December 7, 2016

- ### 3.3.4 Reliability and availability requirement

  Requirement number: NFR9

The game shall take less than five seconds to start up on 90 percents of computers.

Rationale: The program does not take up a lot of memory spaces or cache on the computer, so the startup should be rather quick.

Priority: Medium

History: Modified October 31, 2016

- **Requirement number:** NFR10

  The program shall take less than two seconds to perform actions after a button is pressed.

  Rationale: Similarly, since the program does not take up a lot of memory spaces or cache on the computer, it should not take a long time for responding.

  Priority: Medium

  History: Modified December 7, 2016

3.3.5 Capacity requirement

Requirement number: NFR11

The game shall not have memory explode due to huge player record storage.

Rationale: The game records at most MAX_HS player records on a file, therefore, it should not take up much storage on the computer.

Priority: Medium

History: Modified December 7, 2016

3.4 Operational and Environmental Requirements

- ### 3.4.1 Expected Physical Environment

  Requirement number: NFR12

  The application should be able to run on a computer whenever the computer functions properly.

  Rationale: The program should be able to run when the computer is running; otherwise the program would be considered a failure.

  Priority: High

  History: Created October 11, 2016

3.5 Expected Technological Environment

-
  - **Requirement number:** NFR13

    The application should be able to run on any computer with Java version 1.7 or higher installed

    Rationale: The application is developed under Java version 1.7 environment, so the user should be able to run it under the same environment.

    Priority: High

    History: Created October 11, 2016

3.5.1 Partner Application

Requirement number: NFR14

The application should be able to run on any Java IDE with Java version 1.7 or higher

Rationale: The application is developed on a Java IDE with version 1.7 environment, so the user should be able to run it under the same environment.

Priority: High

History: Created October 11, 2016

3.6 Maintainability and Support Requirements

- ### 3.6.1 Maintenance Requirement

  Requirement number: NFR15

  The application should be under minimal maintenance.

  Rationale: Most of the bugs that occur during the gaming process should be discovered during or after testing, and should be corrected and fixed.

  Priority: High

  History: Modified December 7, 2016

- ### 3.6.2 Supportability Requirement

  Requirement number: NFR16

  The application should run on the majority of users' computers.

  Rationale: Most of the potential players should be able to play the Pong game.

  Priority: High

  History: Created October 11, 2016

- ### 3.6.3 Portability Requirement

  Requirement number: NFR17

  The application should have their file size less than MAX_MB.

  Rationale: This game should be easy to transfer between users for easy access. Players could use any storage devices to transfer files.

Priority: High

History: Modified December 7, 2016

3.7 Security Requirements

- ### 3.7.1 Confidentiality

  None applicable to this project.

3.7.2 File Integrity Requirement

Requirement number: NFR18

The files for the application should have the extension .JAVA.

Rationale: Files need to be recognized as Java files before executions.

Priority: High

History: Created October 11, 2016

- ### 3.7.3 Audit Requirement

  None applicable to this project.

3.8 Cultural Requirements

- **Requirement number:** NFR19

  The language that shall be used in this game is English.

  Rationale: English is the language that all students and faculty members of McMaster University are expected to understand.

  Priority: High

History: Created October 11, 2016

- **Requirement number:** NFR20

  The program shall not contain any imagery or language that can be viewed as offensive to users by mocking, insulting, or appropriating their culture, race, or religion.

  Rationale: User satisfaction will be affected if the material and contents of this game are perceived as offensive to the consumers.

  Priority: Medium

  History: Created October 11, 2016

3.9 Legal Requirements

- **Requirement number:** NFR21

  The program will adhere to the MIT Open License.

  Rationale: The program will adhere to the MIT License as it is the standard that is set for this project.

  Priority: Medium

  History: Created October 11, 2016

4 Project Issues

4.1 Open Issues

So far there aren't any open issues.

4.2 Off-the-Shelf Solutions

The game that this project is attempting to reconstruct would act as an initial model for this project that we have overtaken. There are also other similar games available and the way certain parts are executed can provide a model for us to base our design on. There are also guidance and directions available in Java tutorial sites for any programming problems that are likely to occur during the development of this game.

4.3 New Problems

4.3.1 Problems in the Current Environment

None applicable in the project.

4.3.2 Existing User

Is there a way to clear user record?

4.3.3 Limitations in Implementation Environment

Is it possible to run on every device that has Java?

4.4 Tasks

Tasks Deadline
Model Development October 14
View Development October 14
Control Development October 17
Project Revision October 31
Project Improvement November 19

4.5 Migration to the New Product

None applicable in the project.

4.6 Risks

There are no risks associated with this project.

4.7 User Documentation and Training

The user documents will be created as per the SE 3XA3 guidelines. There will be an in-game tutorial that will display the instructions on how to play the game. Other than that, no further training should be required..

4.8 Waiting Room

There are many factors in the game that has to be improved upon, as has been discussed throughout various parts in the document. The improvement of functionality of the game is to be looked forward towards. The addition of sound to the game can also be added to this.

4.9 Ideas for Solutions

Proper documentations for the game must be made. The coding for the game should be well commented and easy to follow. Other then this, for the current period there isn't any other ideas for solutions. This section would be updated at a later period.

5 Appendix

5.1 Symbolic Parameters

Section 1.4 provides a list of terms and its definition of any symbols that are used throughout the document. Some parameters that are not included in Table 2 are as follows:

Table 3: **Symbolic Parameters**

Terms	**Definitions**
MAX_HS	Maximum number of highscores - set to 20 for this project
MAX_LIVES	Maximum number of lives given to the user and the computer - set to 3 for this project
x	People above this age can play this game - assumed to be 10
MAX_MB	Maximum storage needed to download/play this game - assumed to be 50MB