SE 3XA3: Software Requirements Specification
Pong Invaders

Team #10, Ben Ten
Rehan Theiveehathasan theivers
Karnvir Bining, biningk
Puru Jetly jetlyp

December 8, 2016

Contents

List of Tables

List of Figures

Table 1: **Revision History**

Date	Version	Notes
October 11, 2016	Puru Jetly	Initial Draft
October 11, 2016	Karnvir Bining	Initial Draft
October 11, 2016	Rehan Theivee-hathasan	Initial Draft
December 8th, 2016	Puru Jetly	Final Revision
December 8th, 2016	Karnvir Binign	Final Revision
December 8th, 2016	Rehan Theivee-hathasan	Final Revision

1 Project Drivers

1.1 The Purpose of the Project

The purpose of Pong Invaders is, at its core, to help cure the boredom of its users. Pong Invaders looks to bring back the best of the pioneer gaming generation with a new twist to bring these classical favourites back up to speedwith todays games. This game will allow anyone with a computer to either relivetwo of their favourite games or for a new generation to be exposed to the origins of gaming. With the completion of Pong Invaders anyone with a computer will be able to get their hands on the remodelled duo of Space Invaders and Pong.

1.2 The Stakeholders

1.2.1 The Client

The client for our project is Professor Spencer Smith, 3XA3 Instructor of McMaster University.

1.2.2 The Customers

The primary customer will be the general public interested in playing Pong Invaders. Any individual with internet access able to download the game would constitute as a viable customer for Pong Invaders.

1.2.3 Other Stakeholders

General Public The general public will constitute our consumer base as a whole will makeup the remaining stakeholders. These stakeholders need not be involved in the re-development process since it is a remodelling of two previously made games with already given specifications and requirements. This allows for the general public to be focussed towards creating a player base for Pong Invaders.

1.3 Mandated Constraints

Description: Pong Invaders will be available for anyone able to download the jarfile and a computer to run it on.

Rationale: The jar file download will allow users to run the game on any computer platform with internet access.

Fit Criterion: Pong Invaders will be available to users that can run and download the jar file.

Description: The Game window will be set to a static 640x480.

Rationale: keeping the game on static window dimensions will allow for less problems when adding the pong overlay.

Fit Criterion: The window size will be kept static to ensure game functionality.

1.4 Naming Conventions and Terminology

Definitions of All Terms, Including Acronyms, Used by Stakeholders Involved in the Project

Table 2: Team Communication Table

Acronym/ Abbreviation	**Meaning**
J	Java
Ex	Example
OS	Operating System
txt	Text file
git	Git Lab
eng	Engineering
Ben Ten	Project Team
Product	The game that is being developed
Program	Code that makes up the game
Client	The group this product is made for
Customer/User	Any individual that utilizes the product after completion
Project	Development ofthe product
ETC	Et cetera
Repo	Repository

1.5 Relevant Facts and Assumptions

Relevant Facts

The facts that are relevant to this project are that there are over 950 lines of code currently present in the original Repo of this game. This game needs to be run on Java Development Kit 1.8 or higher.

Assumptions

The assumptions that are made for this project is that all the text and photo editing software will be free and accessible for all developers. The written code will be encapsulated so it won't be sold and only used for educational purposes. The users will have access to a computer with internet access.

2 Functional Requirements

2.1 The Scope of the Work and the Product

2.1.1 The Context of the Work

The project will focus on interactions between developers and the software and interactions between the software and the user.

2.1.2 Work Partitioning

Table 3: Work Partitioning

Number	Component	Input	Output
1	Space Invaders Base Game	Source Code	Executable Jar
2	Space Invaders Sprites	Graphics Files and Source Code	User Display
3	Collision System	Source Code	Executable Jar
4	Pong Base Game	Source Code	Executable Jar
5	Final Revisions	Source Code	Executable Jar

2.1.3 Individual Product Use Cases

Use Case Element Description Use Case Number 1 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Execute File Use Case Description Starts the game for the user. Upon startup the game sends the user to the Start Menu case. Primary Actor User Precondition Must have a recent version of java to run the JAR Trigger Execute JAR file by command

line or by double clicking the JAR file. Basic Flow JAR file will run, and a game window is generated on the users display. Sends user to use case 2 on success. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 2 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Start Menu Use Case Description Allows user to select options: Play Now, How to Play, Controls, and Quit Game. On this case the background also shows the title screen saying Pong Invaders. The start menu also shows the current scoreboard of users who have played recently on the current machine. Primary Actor User Precondition Must have successfully ran the JAR file. Requires use case 1. Trigger Source code automatically sends user to this game state once Pong Invaders JAR file runs. Basic Flow Start menu is initialized. User now has options of: Play Now, How to Play, Controls, Quit Game. Where each option respectively takes the user to case 3, 4,5, 8. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 3 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Play Now Use Case Description Sub option of Start Menu case. Allows the user to play the game. Primary Actor User Precondition Must have successfully ran the JAR file. Requires use case 2 to reach this case.Trigger Selecting start menu option of play now will send the user to this state. Basic Flow If user selects Play Now, the game state sends the user to the Game Active case,which is use case 6. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 4 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name How to Play Use Case Description Sub option of Start Menu case. Teaches user how to play the game. Primary Actor User Precondition Must have successfully ran the JAR file. Requires use case 2 to reach this case.Trigger Selecting Start Menu option of How to Play, will send the user to this state. Basic Flow If user selects How to Play, the game state prints to display how to play based on the current control configurations in Controls. User can from here exit to Start Menu (case 2) on user input. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 5 Application Pong

Invaders Executable JAR. (Requires Java) Use Case Name Controls Use Case Description Sub option of Start Menu case. Allows user to edit the current configuration of controls that the game uses. Primary Actor User Precondition Must have successfully ran the JAR file. Requires use case 2 to reach this case.Trigger Selecting Start Menu option of Controls, will send the user to this case. Basic Flow If user selects Controls, the game state displays all current controls. The usercan select and edit any controls they desire. User from here exits to Start Menu(case 2) on user input. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 6 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Game Active Use Case Description The active state of the game upon selecting the Play Now case. Here the user canfully play the game. Primary Actor User Precondition Must have successfully ran the JAR file. Requires use case 2 to reach the Play Now case. If the Play Now case is successful the user is sent to this case. Trigger Selecting Start Menu option of Play Now, will send the user to this game state. Basic Flow If user selects Play Now, the game state sends the user to Game Active. From here user can either exit to Start Menu (case 2) by user input, or upon losing be taken to the Game Over screen (case 7). Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 7 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Game Over Use Case Description A Game Over screen that displays the current sessions high scoreboard, and where the user ranks currently. Upon any user input the Game Over screen takes the user to the Start Menu (case 2) title screen. Primary Actor User Precondition Must have successfully ran the JAR file. Requires user to lose in Game Active (case 6) to reach this case. Trigger Upon losing in Game Active (case 6). Basic Flow Upon losing in Game Active (case 6) the user views an updated scoreboard with their score and is then sent to the Start Menu (case 2). Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

Use Case Element Description Use Case Number 8 Application Pong Invaders Executable JAR. (Requires Java) Use Case Name Quit Game Use Case Description Exits the JAR application window. Primary Actor User Precondition Must have successfully ran the JAR file. Requires user select

the Quit Game option in the Start Menu (case 2). Trigger Upon selecting Quit Game, or closing the window. Basic Flow In Start Menu (case 2) if selected the window closes. Or when the user closes the window/terminates the process. Alternate Flows JAR file fails. User is told to check their java version, and installation via reading an instruction manual.

2.2 Functional Requirements

1. Aim With Pointing Device The user must be able to utilize and aim with their pointing device. The programwill correctly obtain mouse coordinates from the display window.

   FIT CRITERION OR TEST CASE: test if the mouse coordinates are the correct coordinates by displaying them to terminal as a test. Check if user is able to aim with the mouse.

2. Move With Keyboard The user must be able to utilize the keyboard to control, and move the ship while in game. The user must be able to select menu options with the keyboard.

   FIT CRITERION OR TEST CASE: can the user select menu options? Can the user move in the game with the keyboard?

3. Output Game to Display The program must output correct visuals to the users display. The display must have correct up to date graphics depending on user inputs.

   FIT CRITERION OR TEST CASE: is there a display screen? Are user inputs being mimicked onto the display?

4. Graphics Load Correctly The program must correctly load all relevant graphics contained in the JAR file.

   FIT CRITERION OR TEST CASE: on user input, does the correct sprite/visual load correctly?

5. Correct Return Values The program must correctly return all relevant values from called methods.

   FIT CRITERION OR TEST CASE: makefile with several test cases will run through the source code doing unit testing to see if the correct return values are made.

6. Collision System The game must output the correct events upon unit collisions of ingame sprites and visuals.

   FIT CRITERION OR TEST CASE: does the game end when user collides with an object?Do sprites behave as they should upon collision? Are hitboxes of visuals correct?

7. Title Screen Upon executing the game, the display must output the title screen with a list ofoptions.

   FIT CRITERION OR TEST CASE: Does the title screen show up when the game is initialized? Is it possible to correctly select title screen menu options?

8. Play Now Option Upon entering the title screen the user must be able to select the start menu option of Play Now which correctly runs the game.

   FIT CRITERION OR TEST CASE: does the Play Now option correctly change the game state to the active game state?

9. How to Play Option In the title screen the user must be able to select the How to Play option whichoutputs to terminal correctly the current control scheme of the game and how to play.

   FIT CRITERION OR TEST CASE: does the program print to terminal the correct results?

10. Controls Option The user must be able to select the Controls option from the title screen. The user must be able to select and edit controls that they want to. The edited controls must update to the new controls correctly.

    FIT CRITERION OR TEST CASE: can the user enter the control options? Does the program allow for controls to be edited? Does the program update and save the control scheme correctly?

11. Quit Game Option The user must be able to quit the game in the active game state, and in the title screen via user input.

    FIT CRITERION OR TEST CASE: does the game successfully save and exit when the user prompts to exit the game? When the user closes the window / terminates the process are settings saved properly?

12. Game Over Screen Upon user collision with an enemy in game, the program must take the user to a Game Over screen showing the current high scoreboard and where they rank amongstit if they do at all.

    FIT CRITERION OR TEST CASE: on user collision does a Game Over screen correctly display? Does the high scoreboard display properly? Does the scoreboard update correctly with the current user score?

13. Error Messages On invalid user inputs, the program must output a relevant error message to terminal correctly.

    FIT CRITERION OR TEST CASE: if user enters improper input does the terminal output the correct error message?

14. Winning Game Screen Upon all enemies being defeated and pong AI losing the game, a victory screen will be displayed to the user.

    FIT CRITERIOMN OR TEST CASE: Manually test that the game does indeed end once all alien enemies have been destroyed and the pong AI has let the ball go past itself.

3 Non-functional Requirements

3.1 Look and Feel Requirements

Appearance Requirements

Pong Invaders is going to stay authentic to the color pallets used by both Pong and Space Invaders. The pong paddles will be white the ball also. However the graphics for the Space Invaders component of the game, are going to be licensed by Creative Commons. The program will be displayed on desktop with the resolution of 640x480.

Style Requirements

The style Pong Invaders is trying to illustrate to the user is the retro-arcade style of the games.

3.2 Usability and Humanity Requirements

Ease of Use Requirements

The game shall be able to be played by persons with appendages.

Personalization and Internationalization Requirements

Not applicable.

Learning Requirements

The user shall be able to operate a keyboard.

Understandability and Politeness Requirements

The user shall be able to operate a keyboard.

Accessibility Requirements

The game shall be able to be accessed and executed on users machine with Java Development Kit of 1.8 or higher.

3.3 Performance Requirements

Speed and Latency Requirements

The game shall respond to user input immediately

Safety-Critical Requirements

The game will not compromise the users data or machine.

Precision or Accuracy Requirements

The game shall use floating point values.

Reliability and Availability Requirements

The game shall be available wherever there is a desktop running Java DevelopmentKit 1.8 or higher.

Robustness or Fault-Tolerance Requirements

The game shall be able to operate without consideration of the physical state ofthe end users.

Capacity Requirements

N/A

Scalability or Extensibility Requirements

The code will allow for scalability.

Longevity Requirements

The product shall be relevant for the lifetime of support of Java Development Kit 1.8.

3.4 Operational and Environmental Requirements

Expected Physical Environment

The product will be playable on any computer that has the jar file downloaded. It is technically playable in any environment where you can take a computer.

Expected Technological Environment

Pong Invaders will be written in java and exported as a jar file to be run on any OS (operating system). Meaning it will be runnable on any computer.

Partner Applications:

None.

3.5 Maintainability and Support Requirements

How easy is it to maintain?

Pong Invaders is relatively easy to maintain as it requires little updates afterit is initially launched and might require some accessory downloads if there is more content added. The application will be patched based on user feedback for the game after release.

Special maintenance conditions

In a special case where there is a critical error found in the code the current jar may need to be deleted and completely re-downloaded to ensure proper execution of the game.

Portability

The jar file will be available anywhere a user can access the internet and the game can be taken anywhere the user is willing to take their computer.

3.6 Security Requirements

All individuals are authorized to use this product, and fork their own repository from this project to modify and redevelop as they see fit.

3.7 Cultural Requirements

The product under no circumstances can have offensive icons or images depicted in the final release.

3.8 Legal Requirements

The project is to be licensed and distributed under a Creative Commons license such as the GNU General Public License (GPL) 3.0.

3.9 Health and Safety Requirements

Too much screen time without a break can be harmful to user vision.

4 Project Issues

4.1 Open Issues

Not applicable to this project.

4.2 Off-the-Shelf Solutions

For the product to be executed the following off-the-shelf software is required:- Java Development kit 1.8 or higher

All software can be received for free off the Internet as it is opensource.

Effects on the Current Environment None.

Potential User Problems:

Some user problems that can occur from the product is eye soreness/eye strain from extended playing. Users may not be able to correctly follow installation instructions due to inexperience with Java.

Limitations in the Anticipated Implementation Environment That May Inhibit the New Product:

Computers that are old may not be able to run the product at optimal quality. Meaning sluggish performance and drop in frames per second.

Follow-Up Problems

If one of the developers are not able to hand in major deliverables on time, this can delay the release of the product for an indefinite time. If Java Development Kit 1.8 is not supported in the near future by Oracle.

4.3 Tasks

1. Task
2. Task Completer
3. Timeline
4. Model Implementation
5. Developers
6. October 18th
7. Model Revision

8. Client
9. October 19th
10. Java Implementation
11. Developer
12. November 3rd
13. Testing
14. Developer
15. November 6th
16. Revision
17. Client
18. November 28th
19. Maintenance
20. Developer
21. Yearly

4.4 Planning of the Development

Phases Phase 1 Model Implementation and Revision: This phase is important as this will be the structure and backbone of the project. This is when the game will be set up and how each module interacts with each other. This is crucial tocomplete before any coding is started. The product shall be based on this model.Phase 2 Programming Implementation: This phase is where construction of the product begins. Here the game will be created and tested. The program shall run on desktop. The client will check if all requirements are met for the game and decide if game will be released. Phase 3 Maintenance: At this point for optimal game experience the game shall be maintained at a high standard. This shall be completed and reviewed regularly or upon notice. The program will be continuallyplayable throughout years to come. The program will remain functional as long asthe client deems fit.

4.5 Migration to the New Product

Not applicable to this project.

4.6 Risks

Risks for this project center mainly around productivity and time constraints. Since Pong Invaders will consist of a meshing of two games it requires strong time commitment and high productivity. Failure to do so may result in forced downscaling or extension of the project.

4.7 Costs

Costs for this project center mainly around utility costs to run developer machines, and labour costs. The labour costs are not monetary as this is an opensourced project where the labour costs are in time spent developing for the developers.

4.8 User Documentation and Training

The user documentation is in the form of a README file in the project repository. This file details all installation requirements, and instructions pertaining to installation and running the file. The user requires no training to use this project.

4.9 Waiting Room

V2.0 Pong added to the base game as an additional layer to the display. Pong also functionally reacts with the Space Invaders base game.

V3.0 Audio implemented to all interactions of Pong Invaders. BGM is will now be functional. Audio settings can be manipulated through the title screen in AudioSettings.

4.10 Ideas for Solutions

Implement a class in Java that handles collisions with any object with a designated return type of hitbox. This way the project can scale from Space Invaders to Pong Invaders by allowing Pong to interact functionally with Space Invaders.

5 Appendix

N/A

5.1 Symbolic Parameters

N/A