

SE 3XA3: Software Requirements  
Specification  
Title of Project

Team #104  
Andrew Hum, 400138826  
Arshan Khan, 400145605  
Jame Tran, 400144141  
William Lei, 400125240

February 9, 2020

# Contents

## List of Tables

- Table 1: Revision History
- Table 2: Work Partitioning Events
- Table 3: Work Partitioning Event Summaries

## List of Figures

- Figure 1: System Use Case Diagram

Table 1: **Revision History**

Date	Version	Notes
2/3/2020	1.0	Project Drivers and Project Issues
2/8/2020	1.1	Completed Functional Requirements

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

For all time, humans have played games for entertainment, as a pass-time, or to relieve stress. In the last forty years, humans have had the ability to create games on the computer as well. Over time, these games have only improved in their complexity and design, as well as generated billions of dollars in revenue for the entire gaming industry. Our focus is not on profit however, we care most about giving the players a seamless, satisfying experience when playing the mini games. Thus, we are focusing on making visual and difficulty enhancements to the pre-existing mini games.

## **1.2 The Stakeholders**

### **1.2.1 The Client**

- The client(s) for the product are Dr. Asghar Bokhari and the SFWRENG 3XA3 Teaching Assistants. As clients, they will administer the development of the product. They will also determine the degree to what the product delivers on the requirements.

### **1.2.2 The Customers**

The intended customers of the product are any individuals with the ability to download the product online. This includes a child, up to an elder. The product is created to entertain its users and give them a joyful experience.

### **1.2.3 Other Stakeholders**

All other stakeholders include all current developers as well as any future developers who wish to improve upon the project themselves.

## **1.3 Mandated Constraints**

The system is to be built under the following constraints

- The product must run successfully on at least Windows and MacOS.

- The product must be able to be installed in less than 1 hour by only installing the executable file.
- The product must not require any specialized hardware which the common user may not have.
- The product must free of cost to all users.
- The product must be sufficiently complete by March 20, 2020.
- The product must contain at least one playable mini-game.
- Development of the product must not exceed \$0.
- The mini-game(s) must have a visual and/or difficulty improvement to be considered "updated".

## 1.4 Naming Conventions and Terminology

\*\*\*\* SET SHORTER/SINGLE NAMES FOR EVERYTHING. E.G. CALLING THE WHOLE SYSTEM/PRODUCT/ETC. ONE NAME \*\*\*

## 1.5 Relevant Facts and Assumptions

User characteristics should go under assumptions.

# 2 Functional Requirements

## 2.1 The Scope of the Work and the Product

The Current Situation:

- Currently no launcher. To run the games you need to install the code from using Ubuntu, and running each game individually.
- Maze: a minimal 2-dimensional maze with one preset maze that is generated that can be navigated through single mouse clicks with little-to-no difficulty.

- Flappy: a simplistic game where the user utilizes the mouse-click to prevent their character from colliding with randomly generated objects, and loses when the collide
- Pong: a very simple version of the classic game Pong. It is a two-player game that has contains minimal physics aspect as well as poor graphics and poor performance.

### 2.1.1 The Context of the Work

- This mini game launcher is developed for the course, SFWRENG 3XA3, professor Dr. Asghar Bokahri and the teaching assistants assigned to our group, Thien Tandin and Oluwaseun Owojaiye.

### 2.1.2 Work Partitioning

Table 2: Work Partitioning Events

Event Number	Event Name	Input	Output
1	Selecting/Launching a Game	Mouse/Keyboard	Selected Game
2	Accessing the Leader-boards	Mouse/Keyboard	Display Leader-board
3	Playing the Maze	Keyboard	Final Score
4	Playing Flappy	Keyboard	Final Score
5	Playing Pong	Keyboard	Final Score

Table 3: Work Partitioning Event Summaries

Event Number	Summary
1	Based on the users input, launch the specified game
2	If the user chooses to access the leader-board, a list of the high scores will be listed and will have the option to be sorted by each game.
3	The user will select the difficulty of the maze and navigate their object through the maze using the keyboard. Once completed the system will provide the user with their final score which is the time it took to complete the maze.
4	The user will start a new game and use the space-bar to avoid obstacles. The system will provide their final score once they collide with an obstacle based on the time elapsed.
5	The user will use the keyboard to play against an opponent controlled by the system to try to score against them. They will play till either they or their opponent reaches a desired score. The system will then display the final score and record their score.

### 2.1.3 Individual Product Use Cases

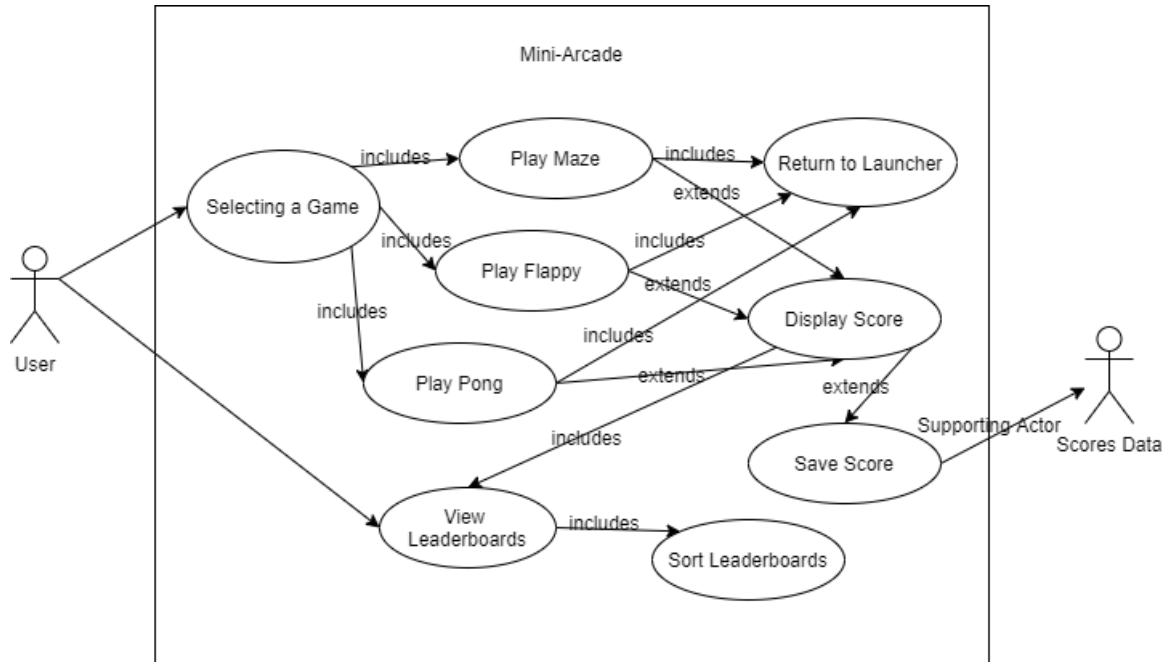


Figure 1: System Use Case Diagram

## 2.2 Functional Requirements

BE1. The user wants to play a game

- FR1. The system shall provide the user with several different mini-game options
- FR2. The system shall launch the desired mini-game in the same window
- FR3. The system shall provide an option to close the application

BE2. The user wants to view their scores

- FR4. The system shall provide the user with the ability to view the leader-boards from the launcher

- FR5. The system shall provide the user with the ability to sort their scores based on the game
- FR6. The system shall provide the user the ability to view the current games leader-boards from the game home screen

BE3. The user wants to play Maze

- FR7. The system shall provide the user with different difficulty levels
- FR8. The system shall record the users current score based on the time elapsed
- FR9. The system shall randomly generate mazes based on difficulty
- FR10. The system shall provide a home button to exit the game at any time
- FR11. The system shall display the users current score and high score once the complete the maze
- FR12. The user shall be able to control the objects movement with the keyboard
- FR13. The system shall provide an option to proceed to another maze once completed
- FR14. The system shall provide an option to return to the launcher

BE4. The user wants to play Flappy

- FR15. The system shall provide the user with the option to start a new game
- FR16. The user shall use the space-bar to control their character
- FR17. The system shall randomly generate objects to interfere with the character
- FR18. The system shall increase speed as the time elapsed increases
- FR19. The system shall generate more objects as the time elapsed increases



- FR20. The system shall provide the user with the final time elapsed once the game is over
- FR21. The system shall provide the option to view the score boards for Flappy
- FR22. The system shall provide the user the option to play again
- FR23. The system shall provide the user with the option to return to the launcher

BE5. The user wants to play Pong

- FR24. The system shall provide the user with the option to play alone or with a friend
- FR25. The user(s) shall use the keyboard to navigate their paddle(s)
- FR26. The system shall add a point to the side that scores
- FR27. The system shall require a final score to play up to
- FR28. The system shall end the game once the final score is reached by either side
- FR29. The system shall provide the option to play again
- FR30. The system shall provide the option to return to the launcher

## **3 Non-functional Requirements**

### **3.1 Look and Feel Requirements**

### **3.2 Usability and Humanity Requirements**

### **3.3 Performance Requirements**

### **3.4 Operational and Environmental Requirements**

### **3.5 Maintainability and Support Requirements**

### **3.6 Security Requirements**

### **3.7 Cultural Requirements**

### **3.8 Legal Requirements**

### **3.9 Health and Safety Requirements**

This section is not in the original Volere template, but health and safety are issues that should be considered for every engineering project.

## **4 Project Issues**

### **4.1 Open Issues**

The following describes issues that has been raised , but have not yet solved:

1. The mini-games are currently using turtle as the graphic library, but the functionality turtle provides is very basic as it is designed as a introduction library for python beginners. Therefore a different graphic library is needed and the codes must be adjusted to use the new graphic library, preferably before any further development is done (this reduces the complexity of switching libraries).

### **4.2 Off-the-Shelf Solutions**

As some of the mini-games are based on commonly spread ideas, there is a decent amount of similar games out there, such as pac-man, flappy bird,

pong, etc.

### 4.3 New Problems

Our project provide a all-in-one package to some of the common mini-games, with a standalone executable launcher providing access to multiple common mini-games without the need of external libraries and internet connection on computers.

### 4.4 Tasks

Task	Timeline
Launcher Model Implementation and Partial Graphic Library Adaptation	February 13 <sup>th</sup>
Adopt New Graphic Library	February 23 <sup>th</sup>
Launcher Development and Game Remake	March 15 <sup>th</sup>
Testing and Revision	March 29 <sup>th</sup>

Task 1: Basic model for launcher and partially modify the mini-game to adapt with new graphic library (at least modify the main menu).

Task 2: Finish the adaptation to the new graphic library.

Task 3: Finish development of the launcher and remake the mini-games to provide more functionality to each mini-game.

Task 4: Test the launcher and mini-game, fix bugs found during testing and optimizes the mini-games.

### 4.5 Migration to the New Product

Not applicable.

### 4.6 Risks

Risks are inherent in the development of any product. The following information highlights possible risks for this project:

- Complexity of Graphic Library
- Over-estimated Project Size

- Difficult Testing
- Bad Programming Practices
- Improper Git Usage

## 4.7 Costs

There is no monetary cost to this project as all resources are free of charge. The original code this project is based on is a open-source project, and all technologies required in the development of the project such as graphic libraries and programming tools are all free, resulting in no monetary cost. The labour cost required for this project is relatively high, as each team member will require to dedicate around 5 to 7 hours minimum per week in the development of this project.

## 4.8 User Documentation and Training

As the product of this project is planned to be compiled into a standalone executable, there will be no standalone user's manual due to the assumption that user understand how to open a standalone executable.

With each mini-game having different controls and playing method, there will be a help section integrated into each mini-game containing help information.

## 4.9 Waiting Room

As this project only re-made part of the mini-games in the original open source project, the future plan will be remaking all mini-games in the original open source project and integrate them into the launcher.

## 4.10 Ideas for Solutions

The following are purposed solution to some of the issues stated in section 4.1:

- Two libraries, tkinter and pygame, have been proposed to be used as the replacement of the original turtle library. Both libraries are suitable as they provide enough functionality for current and future (planned) developments. Currently those two are being debated on the trade-offs between functionality and complexity of the library.

## 5 Appendix

This section has been added to the Volere template. This is where you can place additional information.

### 5.1 Symbolic Parameters

The definition of the requirements will likely call for SYMBOLIC\_CONSTANTS. Their values are defined in this section for easy maintenance.