Software Requirements Specification

Team 5, Puzzeltraz
Muneeb Arshad, arsham14
Bill Nguyen, nguyew3
Jiaxin Tang, tangj63
Karim El Shenawy and elshenak
Matthew McCracken and mccracm

February 5, 2021

1 Introduction

1.1 Purpose

The software Requirements Specifications document is intended to give a complete overview of the game project including the user interface, functional and non functional requirements. The SRS documents details of all features of the game with reference to the manner and importance of their implementation. The SRS document is intended for Developers, Testers and TAs.

1.2 Scope

The proposed software application is named *Escape from Puzzeltraz* and it is intended to provide an escape room based game that allows users to complete a set of puzzles in order to escape the game. A set of puzzles are encapsulated in a *Room* which the user has to complete in a specific time constraint. Each completed puzzles awards the user with a piece of a *Key*. When the Key is completed, the user has defeated the game. Escape from Puzzeltraz provides an immersive, educational experience, increase problem-solving skills and the ability for the user to interact with visual tasks.

1.3 Definitions, Acronyms, and Abbreviations

- **Game** : The entire experience including several minigames named 'Rooms'.
- **Room** : Consists of a minigame which rewards the user a piece of information that allows them to complete "a Key" and also rewards hints for other minigames.
- **Code** : A code is a small portion of the 'Key' which will be provided for a user who successfully passes a minigame.
- **Key** : A word which allows the user to access the final door and win the game. The Key is obtained by completing all "Room" to collect all 'Code'.
- **Unlockable or Final Door** : The final task to comlpete by the user in order to win the game. The Door is unlocked by the "Key".
- **Clock** : The timer that counts down the time in which the user must complete the game in.

1.4 References

No References used.

1.5 Overview

The first section contains the introduction which identifies the scope of this document, the purpose of this software project, and lists the definitions, acronyms and any external documents that were referenced throughout this document. The second section consists of the overall description of the software, including the product description, major functionality, user characteristics, constraints, all assumptions and dependencies, and finally apportioning of requirements. The third section contains the Use Case Diagram. The forth section identifies all functional requirements in this project based off the Use Case Diagram and specifications listed in section 2. Finally, the fifth section encompasses all Non-Functional Requirement of this project.

2 Overall Description

This section of the SRS should describe the general factors that affect the product and its requirements. It does not state specific requirements; it provides a background for those requirements and makes them easier to understand.

2.1 Product Perspective

The game is a 2-d puzzle game with a visual style similar to old Pokemon games. The mini puzzles players must solve are similar to sudoku, tic tac toe, and traditional sorting challenges. Moreover, the product is independant but individual challenges are similar to existing simple games (ie. tic tac toe).

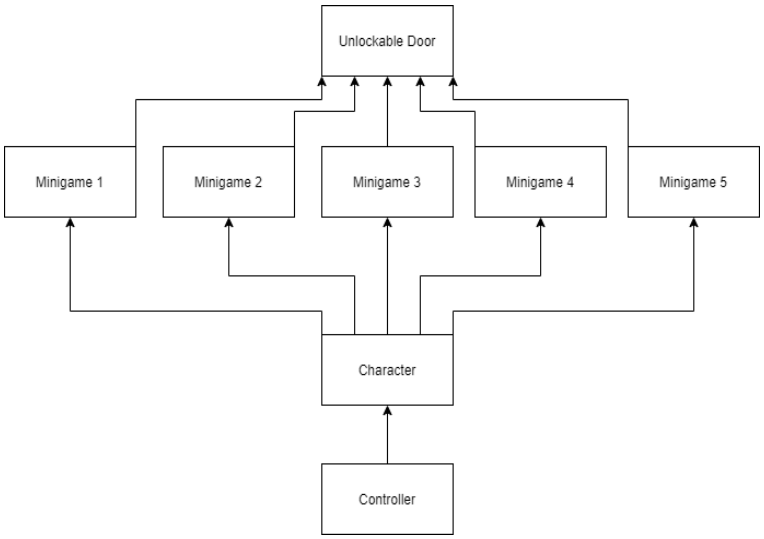

Figure 1: Figure A

2.2 Product Functions

The major functions that will be preformed in this software, is first the mini-games/challenges that the user needs to complete in order to advance to next stage of the game. So to further expand there will be 5 challenges that comprises of simple and fun games such as tic tac toe, Sudoku etc. and after the completion of each challenge the user gets a letter and advances to the next challenge and once all challenges are complete the user enters all the given letters from each challenge to win the game. To add on, each time the user plays the game each 5 challenges will be the same but each time the challenge might change slightly for example if there is a challenge to solve a math problem, each time the user starts a new game the math problem will change as well.

The second major function is the aspect of winning. In order to win the game the user must complete all challenges and enter all letters that correspond to each challenge in a given time period. If the user does

not complete all challenges in the certain time slot then the user loses and would be sent back to the main menu. Also, there is no time period for each challenge, the time period would represent the period of time allotted for all tasks to completed. Once the user completes the game the user will be sent back to the main menu, that gives them the option to play again.

The third major function, is the main menu, which will be a simple display giving the user the option to start the game. The main menu will also be used to represent the home or a start place for the user, when for example the user wins or loses they will be sent back to the main menu and when the user launches the game, the first display they will see is the main menu.

2.3 User Characteristics

The intended user will be 10-20 year old people with no technical experience required. Simple challenges will be perfect for this age group and the challenge will come from the time limit rather than the puzzles themselves. Anyone under 10 may struggle with the puzzles. Older people may find the game too trivial.Beneficial previous experience includes having walked around in a game before using WASD keys. Additionally games like Suduko and Tic-Tac-Toe will be named but previous experience playing them would be helpful. Other puzzles that are new or unique will be fully explained for the users.

Later requirements are specified to define the functionality of the system. When requirements for changing difficulty level are written, it is to ensure that the game is playable for markets of all ages. Requirements to support timing functionality are geared towards adding challenge to the game.

2.4 Constraints

On the developer's side, the only constraint this project is facing is the Scope Constraint as this project must satisfy a particular set of requirements, noted in the "Project Outline for Software Design II (SE3A04)".

2.5 Assumptions and Dependencies

The project revolves on the user using a keyboard to move around in the game. Thus, the assumption is that the software system must be available for a desktop environment as a non-web based application.

2.6 Apportioning of Requirements

We do not yet have previous requirements that must be delayed for future versions of the system.

3 Use Case Diagram

The Use Case diagram for the software application entails the several activities the user is given which in this case, the user is considered to be a *Player*. The player begins by starting the game which includes a difficulty selection. This selection decides the amount of time constraint given in each minigame. Moreover, the player then enters a *Room* which will then includes 5 different puzzles. Each puzzle awards a piece of the final *Key* when completed. The player collects each piece in order to enter it to unlock the Room's Door. At this point, the player is considered to have won the game. In addition, the player has the option to restart the Room at any given point which resets the timer and all puzzles. They have also the option to quit the Room which terminates the Room and ends the timer.

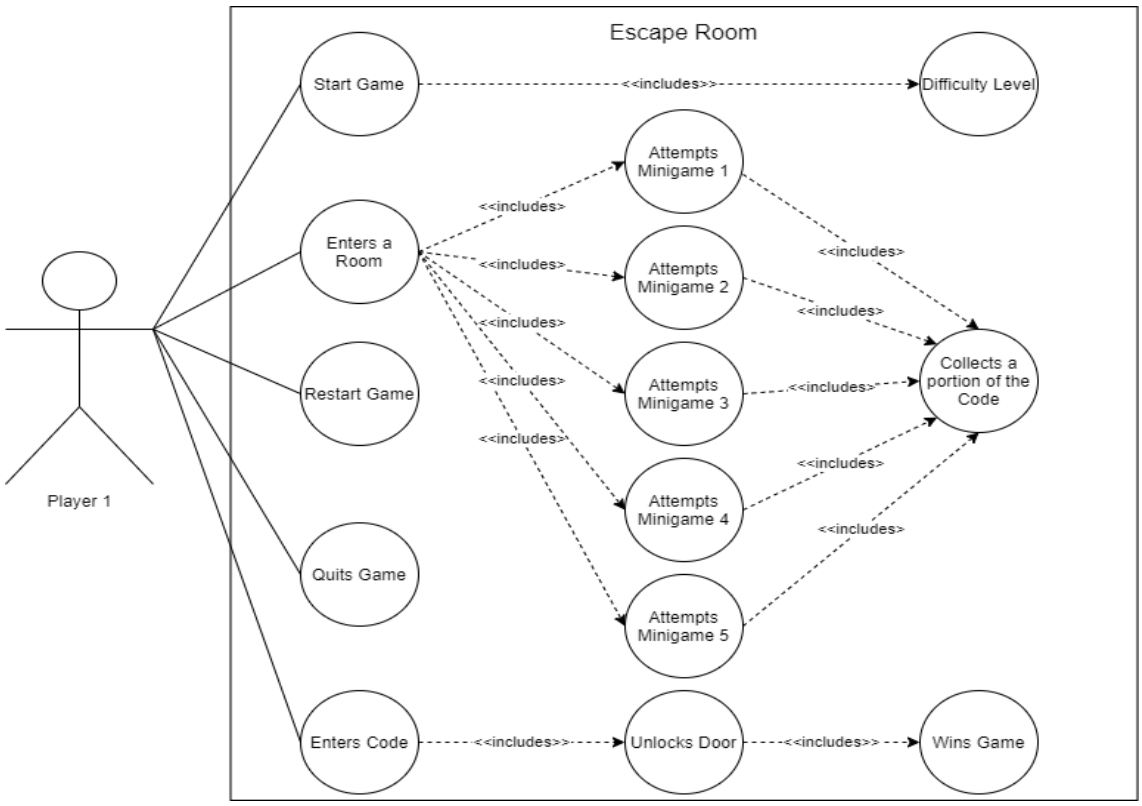

Figure 2: Use Case Diagram

4 Functional Requirements

VP1. Viewpoint: Game Player

BE1 Business Event: Player wants to start the game.

1 The user starts from the main menu and selects/clicks the button/option that is labelled "Start Game".

2 The system must provide a secondary menu to take in a players difficulty preference with a choice between easy, medium, and hard.

3 The timer must be set in accordance with the set difficulty level.

4 The system/game shall then begin with and the timer must commence.

BE2 Business Event: Player wants to begin a challenge.

1 Upon starting the game, the player must enter a room that has 5 challenges, which the player can move to the desired challenge they want to start

2 Once the player decides what challenge they want to begin, the player must click on the challenge in order to begin the chosen mini-game

3 The game should provide the player with a 1 digit code upon completion of the challenge

4 All challenges must be completed under the time limit

BE3 Business Event: Player wants to exit a challenge.

1 The player must navigate to the back button in the bottom of the screen.

2 The user must then click on the back button to return back to the room with all the 5 challenges.

BE4 Business Event: Player wants to escape the room

1 The system must provide an interface to accept a 5 digit code.

2 The system must store and recognize the correct code and will open the door if it the correct code is entered.

3 The player must enter the code before the time limit ends to win the game.

4 The system must provide a congratulatory game over screen upon escaping the room and then bring the player back to the main menu.

BE5 Business Event: Player wants to change the difficulty.

1 The system with provide players with the ability to change difficulty between easy, normal, and hard.

2 The system will scale game difficulty to correspond with current selected difficulty. Specifically, timing allotted to players to escape the room will be shorted for higher difficulty and lengthened for lower difficulty.

BE6 Business Event: Player wants to move around the room.

1 The system must move the player model in relation to the WASD key being pressed in the respective directions Up, Left, Down, and Right.

BE7 Business Event: Player wants to end game.

1 The system must have a pause button that, when clicked, brings players to a pause menu containing options to resume game, or quit.

2 The system must return a player to the main menu when they click the quit button.

3 The main menu must have an option labeled, close game, which must close out of the game when clicked.

BE8 Business Event: The player completes a challenge.

1 The system must analyze if individual challenge requirements have been met and throw pass or fail notifier when appropriate.

2 On challenge passed, the system must remove a player from the challenge interface and reveal one of 5 key characters needed to unlock the door.

3 On challenge failed, the system must throw a failed warning and reset the game state for the challenge being attempted and then give the player a chance to do the challenge again.

BE9 Business Event: The player times out.

1 The system must provide a failed game over screen upon timing out and then bring the player back to the main menu

5 Non-Functional Requirements

5.1 Look and Feel Requirements

5.1.1 Appearance Requirements

LF1. The appearance of each challenge shall be attractive to users.

LF2. The transition from challenge to another challenge shall be smooth and discrete.

LF3. The system shall have simple design for users to use.

LF4. The timer shall display on screen clearly.

LF5. The background shall not overpower any colors of objects in the game.

5.1.2 Style Requirements

LF1. The product shall appear to be a challenging and entertaining game.

LF2. The overall design of the room and challenge shall be consistent and attractive i.e. choice of theme and colors shall match in each challenge.

5.2 Usability and Humanity Requirements

5.2.1 Ease of Use Requirements

UH1. The system shall have clearly labelled buttons for users to start or quit the game.

UH2. The system shall be easy for users with little gaming experience to use.

UH3. The system shall be easy for users with no technical background to use.

5.2.2 Personalization and Internationalization Requirements

UH1. The system and challenges shall be provided in English.

5.2.3 Learning Requirements

UH1. The system shall provide instructions of each mini game before the user's first attempt.

UH2. The system objective should be easily understood and explained in a concise and brief manner.

UH3. Each user should be able to understand each challenge with the least amount of explanation.

5.2.4 Understandability and Politeness Requirements

UH1. The system shall use an average level of vocabulary.

UH2. The system shall be built with logical rules.

5.2.5 Accessibility Requirements

UH1. The system shall be available for mobile devices for Android Phones.

5.3 Performance Requirements

5.3.1 Speed and Latency Requirements

PR1. The system shall respond to a user within 10 seconds after a user selects a challenge.

PR2. The system shall go back to the main menu within 10 seconds if a user presses the back button while playing a mini game.

PR3. The system shall go back to the main menu within 10 seconds after a user runs out of time.

5.3.2 Safety-Critical Requirements

PR1. N/A

5.3.3 Precision or Accuracy Requirements

PR1. N/A

5.3.4 Reliability and Availability Requirements

PR1. The system shall never crash.

PR2. The system shall be available for user at any time.

5.3.5 Robustness or Fault-Tolerance Requirements

PR1. The system shall be usable in a local mode.

5.3.6 Capacity Requirements

PR1. N/A

5.3.7 Scalability or Extensibility Requirements

PR1. N/A

5.3.8 Longevity Requirements

PR1. N/A

5.4 Operational and Environmental Requirements

5.4.1 Expected Physical Environment

OE1. N/A

5.4.2 Requirements for Interfacing with Adjacent Systems

OE1. N/A

5.4.3 Productization Requirements

OE1. The system shall be able to be installed by users without training.

5.4.4 Release Requirements

OE1. N/A

5.5 Maintainability and Support Requirements

5.5.1 Maintenance Requirements

MS1. The system shall be provided with clear documentation.

5.5.2 Supportability Requirements

MS1. N/A

5.5.3 Adaptability Requirements

MS1. The system shall be functional for any versions of Android within 3 years of version.

5.6 Security Requirements

5.6.1 Access Requirements

SR1. The system shall not require passwords, emails or any accounts to function.

5.6.2 Integrity Requirements

SR1. N/A

5.6.3 Privacy Requirements

SR1. The system shall not implement any invasive software.

SR2. The system shall not require any personal information from users.

5.6.4 Audit Requirements

SR1. N/A

5.6.5 Immunity Requirements

SR1. N/A

5.7 Cultural and Political Requirements

5.7.1 Cultural Requirements

CP1. N/A

5.7.2 Political Requirements

CP1. N/A

5.8 Legal Requirements

5.8.1 Compliance Requirements

LR1. N/A

5.8.2 Standards Requirements

LR1. N/A

A Division of Labour

Team Member	Tasks
Jiaxin Tang	Section 5(Non-functional Requirements)
Karim El Shenawy	Section 1, Section 3 (Use Case Diagram)
Mathew McCracken	Section 2 and Section 4(Functional Requirements)
Muneeb Arshad	Section 1(Introduction) and Section 4(Functional Requirements)
Bill Nguyen	Section 2 and Section 4 (Functional Requirements)

Table 1: Division of Labor