TRIVIA MAZE: THE GAME

# Software Requirement Specifications

Version 0.1

Prepared by:

Nate Hiblar

Quin Tiller

Wyran Parr

Date Created:

5.15.2020

Table of Contents

[**SOFTWARE REQUIREMENT SPECIFICATIONS 0**](https://d.docs.live.net/6b348535d876d8cd/Documents/EWU/Winter%202017/CSCD%20488/SRS/SRS-Rough-Draft%20v3.docx#_Toc479931547)

1. [INTRODUCTION 3](#_bookmark0)
   1. [PURPOSE 3](#_bookmark1)
   2. [PROJECT SCOPE 3](#_bookmark4)
2. [OVERALL DESCRIPTION 4](#_bookmark6)
   1. [PRODUCT PERSPECTIVE 4](#_bookmark7)
   2. [OPERATING ENVIRONMENT 4](#_bookmark9)
   3. [DESIGN AND IMPLEMENTATION CONSTRAINTS 4](#_bookmark10)
   4. [ASSUMPTIONS AND DEPENDENCIES](#_bookmark11) 4
3. [SYSTEM FEATURES 6](#_bookmark12)
   1. [FEATURE 1 6](#_bookmark14)
      1. [DESCRIPTION AND PRIORITY 6](#_bookmark15)
      2. [STIMULUS AND RESPONSE 6](#_bookmark16)
      3. [FUNCTIONAL REQUIRMENTS 6](#_bookmark17)
   2. [FEATURE 2 6](#_bookmark14)
      1. [DESCRIPTION AND PRIORITY 6](#_bookmark15)
      2. [STIMULUS AND RESPONSE 6](#_bookmark16)
      3. [FUNCTIONAL REQUIRMENTS 6](#_bookmark17)
   3. FEATURE 3
      1. [DESCRIPTION AND PRIORITY 6](#_bookmark15)
      2. [STIMULUS AND RESPONSE 6](#_bookmark16)
      3. [FUNCTIONAL REQUIRMENTS 6](#_bookmark17)
4. [EXTERNAL INTERFACE REQUIRMENTS 12](#_bookmark33)
   1. [EXAM STATES 12](#_bookmark34)

|  |  |  |
| --- | --- | --- |
| [5.1.1](#_bookmark35) | [HIDDEN](#_bookmark35) | [12](#_bookmark35) |

1. [**EXTERNAL INTERFACE REQUIREMENTS 13**](#_bookmark45)
   1. [**HARDWARE INTERFACES 13**](#_bookmark46)
   2. [**SOFTWARE INTERFACES 13**](#_bookmark47)
   3. [**USER INTERFACES 13**](#_bookmark48)
   4. [**COMMUNICATION INTERFACES 13**](#_bookmark49)
2. [**NON-FUNCTIONAL REQUIREMENTS 14**](#_bookmark50)
   1. [**NONFUNCTIONAL REQUIREMENTS 14**](#_bookmark51)

# Introduction

## Purpose

This SRS begins to document known requirements and necessary information for our team to develop a trivia-based game.

## Project Scope

The scope of this project is limited to the playing of a simple desktop-based game. The software will include an installer program. From a desktop icon a user will start the program and be given the option to load an existing game or start a new game. When playing the game, a user can save, quit, or start a new game at any point. When the user completes the maze, or loses a game, there will be options to exit the program or restart the game.

# Overall Description

## Product Perspective

The Trivia Maze game should be intuitive to use. Through as easy to understand GUI, mouse clicks with drive all the game play. Players should be able to easily understand all the game options, the current board state, and receive visual feedback on right or wrong answers.

## Operating Environment

|  |  |
| --- | --- |
| OE-1: | The game will be able to run on Windows 10. |

## Design and Implementation Constraints

|  |  |
| --- | --- |
| CO-1: | The system shall use ­­­­­­­­­­­­­­­\_\_\_\_\_\_\_SQlLite\_\_\_\_\_\_ as its database. |

## Assumptions and Dependencies

|  |  |
| --- | --- |
| AS-1: | Assumptions will be filled in as more information becomes know. |
| DE-1: | Dependencies will be filled in as more information becomes know. |

# System Features GUI

## System Features

3.1.1 Description and Priority

Features will include, saving the game, loading the game, being able to navigate the maze using mouse, will consist of short answer and multiple choice questions, there will be a GUI for all of this and it will consist of 4 multiple choice buttons that you can click to submit your answers to get through the maze. Furthermore, locking and unlocking doors for each room based on answer choice. I believe all these are high priority since this game would not be much of a game without these functionalities.

3.1.2 Stimulus/Response Sequences

Using mouse to click arrows to move from room to room or the maze to move from room to room.

Receiving and answering the question to continue with the game.

Not answering the question correctly will Block your path until eventually there will be no more paths to take and at this point you have lost.

Answering correctly will allow the locked rooms to be unlocked.

Saving the game so you can continue later but also keep your work.

Loading the game so you can possibly have multiple games you want to finish.

Restarting the game if you feel like you have already made to many mistakes.

Quit the game.

3.1.3 Functional Requirements

REQ-1: There will be edge cases that only allow mouse use for navigating through the maze and selecting answers.

REQ-2: Depending on the questions we ask, there should be edge cases for how many and what kind of characters you may use on the short answer questions.

REQ-3: In order for the game to be played later we must implement save game and load game, which will most likely be a drop-down menu at the top.

REQ-4: If we want the player to know where he is in the maze, we must create a game piece and create a method that will continually check the players current location until the game is won.

REQ-5: We also want the player to know when something is answered wrong or right and notify the player as such. If an answer is wrong the room will clearly display a blocked path and if the answered question is correct it will clearly show the game piece moving forward.

REQ-6: Allowing a restart, there must be a way for the player to click the menu and have the option to refresh the game.

## Feature Name Here

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

## Feature Name Here

3.1.1 Description and Priority

<Provide a short description of the feature and indicate whether it is of High, Medium, or Low priority. You could also include specific priority component ratings, such as benefit, penalty, cost, and risk (each rated on a relative scale from a low of 1 to a high of 9).>

3.1.2 Stimulus/Response Sequences

<List the sequences of user actions and system responses that stimulate the behavior defined for this feature. These will correspond to the dialog elements associated with use cases.>

3.1.3 Functional Requirements

<Itemize the detailed functional requirements associated with this feature. These are the software capabilities that must be present in order for the user to carry out the services provided by the feature, or to execute the use case. Include how the product should respond to anticipated error conditions or invalid inputs. Requirements should be concise, complete, unambiguous, verifiable, and necessary. Use “TBD” as a placeholder to indicate when necessary information is not yet available.>

<Each requirement should be uniquely identified with a sequence number or a meaningful tag of some kind.>

REQ-1:

REQ-2:

# External Interface Requirements

## Hardware Interfaces

The game will run on any computer running windows 10.

## Software Interfaces

\_\_\_\_\_\_\_\_\_\_\_\_\_

The system will use \_\_\_\_\_\_\_\_\_\_\_ to store questions and answers.

\_\_\_\_\_\_\_\_\_\_\_\_\_

## User Interfaces

\_\_\_\_\_\_\_\_\_\_\_\_\_

## Communication Interfaces

\_\_\_\_\_\_\_\_\_\_\_\_\_

.

# Non-Functional Requirements

## Other Requirements

## Performance Requirements

<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>

## Safety Requirements

<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>

## Security Requirements

<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>

## Software Quality Attributes

<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>

# Other Requirements

<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>

Appendix A: Glossary

<Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.>