

---

# **Software Requirements Specification**

**for**

## **Trivia Maze Game**

**Version 1.0 approved**

**Prepared by Austin Hall, Cody Bafus, Steven Zuelke**

**CSCD350 EWU**

**5/1/2020**

## Table of Contents

<b>Table of Contents</b> .....	<b>ii</b>
<b>Revision History</b> .....	<b>ii</b>
<b>1. Introduction</b> .....	<b>1</b>
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Project Scope.....	1
1.5 References.....	2
<b>2. Overall Description</b> .....	<b>2</b>
2.1 Product Perspective.....	2
2.2 Product Features.....	2
2.3 User Classes and Characteristics.....	3
2.4 Operating Environment.....	3
2.5 Design and Implementation Constraints.....	3
2.6 User Documentation.....	4
2.7 Assumptions and Dependencies.....	4
<b>3. System Features</b> .....	<b>4</b>
3.1 The Maze Internal Components and External Interface.....	4
3.2 Serialization.....	5
3.3 Trivia Questions.....	6
3.4 Cheats.....	7
3.5 Audio/Video.....	7
3.6 Database.....	8
<b>4. External Interface Requirements</b> .....	<b>8</b>
4.1 User Interfaces.....	8
4.2 Hardware Interfaces.....	9
4.3 Software Interfaces.....	9
4.4 Communications Interfaces.....	9
<b>5. Other Nonfunctional Requirements</b> .....	<b>9</b>
5.1 Performance Requirements.....	9
5.2 Safety Requirements.....	9
5.3 Security Requirements.....	10
5.4 Software Quality Attributes.....	10
<b>6. Other Requirements</b> .....	<b>10</b>
<b>Appendix A: Glossary</b> .....	<b>10</b>
<b>Appendix B: Issues List</b> .....	<b>10</b>
<b>Appendix C: References</b> .....	<b>11</b>

## Revision History

Name	Date	Reason For Changes	Version
Austin Hall	5/14/20	Initial Draft	1.0 Draft 1

# **1. Introduction**

## **1.1 Purpose**

This SRS describes the software functional and non-functional requirements for the Trivia Maze Game version number 1.0. This document is intended to be used by the members of the project team that will design, implement, and verify the correct functioning of all software related to the production of the Trivia Maze Game. Unless otherwise noted, all specified requirements and implementations covered in this document are of the Trivia Maze Game committed for release 1.0.

## **1.2 Document Conventions**

This document will follow standard SRS style conventions including 12-point Times font within paragraph bodies, 18-point bold Times font in the headers of each section, each section will include one or more subsections with included header of 14-point bold Times font. When writing this document from this point forward, the project name Trivia Maze Game will be acronymized to TMG.

## **1.3 Intended Audience and Reading Suggestions**

This document is intended for the developers of the TMG to serve as an initiation to open discussion for exploring development opportunities regarding current and future iterations of the software relating to the TMG, as well as outline current project conventions and specifications to be used as a reference to ensure homogenous development practices. The document also serves as required documentation about the TMG within the scope of the CSCD350 Software Development course assigned, and intended for use, by the instructor Thomas Capaul. Section 2 of the document will provide an overall description of the TMG software, including features, characteristics, and constraints. The next section will contain the TMG technical functional requirements for the product by system features which are the major services provided by the product. Then the external interface requirements highlighting the logical characteristics of each interface between the software product and the users are discussed. The section after that will include other non-functional requirements that aren't intended for user use functionality but are still required for the project as a whole. Finally, the document will conclude with references.

## **1.4 Project Scope**

The final product enables the user to answer trivia questions interactively in a fun and engaging way to progress through a maze of contiguous rooms, with wrong answers permanently locking doors and thus preventing advancement to the end of the maze from that room. The goal is to provide the user with an entertaining experience to maintain product usability. This document specifies the TMG version 1.0 and includes all implementations regarding this version intended for final release.

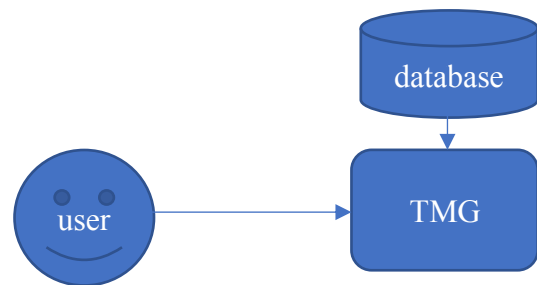
## 1.5 References

Team Coding Standards by Austin Hall, Cody Bafus, Steven Zuelke [1]

## 2. Overall Description

### 2.1 Product Perspective

The TMG is a newly designed implementation of a previously outlined system provided to the development team from the instructor Thomas Capaul. It is a self-contained product featuring a Sqlite database which contains the questions and answers that will be used within the final implementation of the game. The user will interface directly with the TMG through the keyboard and play the game, while the game will be connected to the database to provide through the interface the questions and answers to the user (Figure 1). The system is expected to be finalized and ready for release at the end of production of version 1.0.



### 2.2 Product Features

Figure 1. Relationship of Systems and User

**2.2.1 Maze interface:** The TMG will feature an ascii art style interface to represent the players position within the maze, as well as rooms, questions and answers, and locked or unlocked doors.

**2.2.2 Load and Save Functionality:** The user of the TMG will have the ability to save their position, progress, and statistics into a saved game file. On a later date or time, the user has the option to load a previously saved game and begin playing from the saved state.

**2.2.3 Trivia Questions:** The primary feature of the game is the trivia questions that will be asked. With a variety of questions and answer types, the player will be able to play continuous rounds of the game, with each correct answer allowing furtherment to the end of the maze, and each incorrect answer locking a door to add difficulty in completing the maze.

**2.2.4 Cheats:** The TMG features a secret input that will allow the user to do a variety of things that are otherwise not possible to do in the game, such as unlocking every door, displaying correct answers, and increasing the number of “lifelines.”

**2.2.5 Lifelines:** As with most other trivia-oriented products and entertainment, the TMG will include lifelines that players can find throughout the maze. These lifelines are consumable one time use power ups that may give hints to the correct answer or be a key to a locked door.

**2.2.6 Audio/Video:** The TMG incorporates audio and video components to indicate correct/incorrect answers and a short media experience for winning or losing the maze.

**2.2.7 Database access:** A key component of the interfaced TMG is the Sqlite database that will store the questions and answers. With the correct credentials, the developers have access to this database to add or delete questions and answers from within the game.

## **2.3 User Classes and Characteristics**

The users of the software infrastructure enabling playthrough of the TMG can be differentiated by the developers and managers having been given the confidential access code to enable cheats and make modifications to the database. This user class is designated for use by high level privilege given by the developers on a need-to-have basis. As for all other users, specifically the consumer, there is one class which will be the primary interaction interface of the TMG. This class enable gameplay and presents the questions one at a time as they are encountered with correct answers only visible after answering the question, unless a lifeline is used.

## **2.4 Operating Environment**

This is going to be a .jar executable file so a Java Runtime Environment will be necessary to run this software. Further requirements are TBD.

## **2.5 Design and Implementation Constraints**

The primary challenge in developing this software will be time constraints on the developers. Each of the three members of the development team are full time students with full or near full-time jobs and so the development process will need to be allotted to conform to the schedules of each member. Other constraints include developing the software on different operating systems, which may become problematic in late stages of implementing the installation and driving software. Although, Java can run on any operating system, so it may not become as problematic as prepared for at this time.

## **2.6 User Documentation**

Along with the software product, a user manual in the form of a Readme.txt file would be written to help people understand the working usage of the developed system. It would be written for nontechnical individuals and the level of content or terminology would differ considerably from, for example, a System Administration Guide, which is more detailed and complex. The user manual would follow common user documentation styles capturing purpose and scope of the product along with key system features and operations. In addition to the user portion of this manual, there will be included a manual for use by the instructor of the CSCD 350 class that contains information on accessing the cheats and database system.

## **2.7 Assumptions and Dependencies**

Third-party components that are being used to aid in the development process of the TMG include Github and Pivotal Tracker. Unless there are extraneous circumstances that prevent the development team from accessing, using, and communicating with and through these third-party websites, there are no factors that will affect the requirement stated in this SRS document.

# **3. System Features**

The major services and functional requirements for the product can be illustrated by system features. This section is organized by use cases for major system features. In the following, necessary description is provided for each use case in the system. Each use case description provides information of the associated priority, stimulus condition and response sequences, exceptions and functional requirements (assumptions). Being a major important section of the SRS, this section is expected to go through iterative improvement to make the most logical sense for the intended product.

## **3.1 The Maze Internal Components and External Interface**

### **3.1.1 Description and Priority**

The maze component of this product is the primary component of the entire project. It will regulate the player position within the maze, whether a door is locked or unlocked, provide the question and answer to the user, access the database through other subclasses, and provide the user with an ascii art interface of the size, components, and player position of the interfaced maze. Because of its high dependence on other subclasses, and the fact that this is the driving class of the product, the Maze is the highest priority component.

### **3.1.2 Stimulus/Response Sequences**

1. The user will execute the game file and either load a saved game or begin a new game.
2. The user will then be prompted and given several options, in the case of the maze, the user has selected to begin the game.

3. The user is interfacing with the game through the console window which displays the questions, movement options, and other stats of the game. Depending on what option the user selects, the maze will move the player, lock a door, unlock a door, or display a win or loss screen, depending on the final outcome of the game.

### 3.1.3 Exception:

If the user provides an unaccepted input or unsupported format with regard to the maze (such as incorrect movement or answer) then an error message will be displayed on the screen, prompting the user to attempt their input again.

### 3.1.4 Functional Requirements

- REQ-1: Maze Class
- REQ-2: Room Class
- REQ-3: Question Class
- REQ-4: Methods to correctly set up rooms
- REQ-5: Methods to correctly move between rooms
- REQ-6: Methods to check if the user won or lost.
- REQ-7: TBD

## 3.2 Serialization

### 3.2.1 Description and Priority:

Serialization of the TMG will provide the user with the ability to save a game and load a saved game to resume play from a previous state. Since this feature is not essential to the running/maintenance of the TMG, it will be classified as low priority.

### 3.2.2 Stimulus/Response Sequences:

1. During main menu, select load game for a browser to pick which file to load from and immediately resume playing that game.
2. During any game, select save game for the current game file to be overwritten or a browser if there is no current game file.

### 3.2.3 Functional Requirements:

- REQ-1: Every class must be serializable
- REQ-2: Save and Load game methods within Maze class
- REQ-3: Extra menu options to load and save game.
- REQ-4: A class member to keep track of if the game has a current file.

### 3.3 Trivia Questions:

#### 3.3.1 Description and Priority:

The TMG's main feature that accompanies the maze is the trivia questions asked to the user, with correct responses opening a locked door, and incorrect responses forever locking the door. There will be a variety of questions asked, ranging from multiple choice, true/false, and short answer. Because this is such an integral component of the system as a whole, since without trivia, the maze would be contiguous rooms, this is a high priority feature.

#### 3.3.2 Stimulus/Response Sequences

1. The game will prompt the user for an input related to movement, whether North, South, East, or West, and if a locked door is encountered, a trivia question will appear on the interface.
2. The question and question type will be random and consist of one of three aforementioned types, depending on this type, the user will type the answer.
3. The answer type may be multiple choice, in which case the user will enter a digit 1-4 to correspond to that answer provided on screen.
4. If the answer type is true/false, the user will type either y/n, t/f, or 1/2, the final format is to be determined.
5. If the answer type is short answer, the user will type a one-word answer to address the question.
6. If the user correctly answers the question, there will be a positive sound (addressed in section 3.5) and a door will be unlocked in the direction associated with the question (section 3.1).
7. If the user incorrectly answers the question, there will be a negative sound and the door will remain unlocked for the remainder of the game, although it can be unlocked if the user possesses an unlock lifeline.

#### 3.3.3 Exceptions:

In the event that the user provides an incorrect method of input, an error message will be displayed on the screen prompting them to try again.

#### 3.3.4 Functional Requirements

- REQ-1: Multiple-Choice Question class
- REQ-2: True/False Question class
- REQ-3: Short Answer Questions class
- REQ-4: An Abstract Question Class
- REQ-5: A list of all questions and which one has been taken



## **3.4 Cheats**

### **3.4.1 Description and Priority:**

The TMG features the ability for developers and privileged users to enter a specific sequence of numbers or characters in order to activate components of the game that would otherwise be inaccessible to non-privileged users. These cheats include, but are not limited to: unlocking every door, locking every door, displaying the correct answer to the question, and playing the win/loss/lock/unlock sounds. Privileged users will be given the access sequence to use these cheats within the game. Creating the cheat methods is a low priority.

### **3.4.2 Stimulus/Response Sequences:**

TBD

### **3.4.3 Functional Requirements:**

- REQ-1: Adjust the CheckCorrect method to allow the cheat-code instead of the real answer.
- REQ-2: Add to README.txt so Tom knows.
- REQ-3: TBD

## **3.5 Audio/Video**

### **3.5.1 Description and Priority**

The TMG will incorporate audio and video components to play during locking and unlocking of doors/ correct and incorrect answer guesses, as well as a win/loss video to played at the end of the game. These media components are not a vital part of the finished system and are thus of low priority.

### **3.5.2 Stimulus/Response Sequences**

TBD

### **3.5.3 Functional Requirements**

- REQ-1: Add sounds to the abstract question class
- REQ-2: Add video clip on the win/loss methods.
- REQ-3: The users system must play audio and have the volume turned up.

## 3.6 Database Access

### 3.6.1 Description and Priority

The TMG features an external database in the form of a Sllite server to store the question/answer combination and the incorrect answers for multiple-choice and true/false. There will be a sequence of characters or numbers that can be entered at the start of the game so that the developers or other privileged users can examine and modify the entries contained in the database. Access to the database is an essential process that is of high priority. The trivia portion of the game, and thus the game itself, is completely dependent on this access.

### 3.6.2 Stimulus/Response Sequences

1. The game itself will access the database on a need to basis as the player enters different rooms. Once a player enters a room, the database will query a randomly selected question and the corresponding answer (which will not be displayed until an answer is inputted to the TMG).
2. Once a question and answer are queried, these will be put into a data structure to ensure that the user is not displayed the same question twice.
3. On the main screen, there will be a confidential access code for developers and privileged users to access the data to make modifications or view entries, this code is TBD.
4. When the code is entered, there will be an option to view the entries and an option to modify entries.

### 3.6.3 Functional Requirements

- REQ-1: Database must be created
- REQ-2: DatabaseAccess class must connect and retrieve the data.
- REQ-3: Extra menu options to access the database.

## 4. External Interface Requirements

### 4.1 User Interfaces

**UI-1:** The user will interface with the system through the monitor. There will be either a pop-up window or simply a console interface to display the questions, maze, and options; however, the specific details of this interface are not yet known at the time of this document and are TBD.

**UI-2:** There will be audio and video components incorporated into the final system that play corresponding to user actions (section 3.5).

**UI-3:** Further user interfaces and details related to listed interfaces will be added to subsequent versions of this SRS document.

## **4.2 Hardware Interfaces**

No hardware interfaces have been identified.

## **4.3 Software Interfaces**

### **SI-1: TMG**

**SI-1.1:** The TMG will interface with either the console window or a popup window (TBD) to display all pertinent information about the game, such as player position, stats, and questions (section 3.1).

**SI-1.2:** The TMG will have an internal storage for the audio/video files (section 3.5).

**SI-1.3:** The TMG will take user input from the keyboard for movement, options, and answering questions.

### **SI-2: Sqlite database (version 3.31.1)**

**SI-2.1:** The TMG will connect to a Sqlite database via internal methods and the internet to access questions and answers (section 3.6).

**SI-2.2:** The TMG will have the capability to view and modify the database from within the game.

## **4.4 Communications Interfaces**

Since Sqlite is a self-contained library linked into the application program of the host system, the Sqlite engine has no standalone processes with which the application program communicates. For this reason, there are no communication interfaces.

# **5. Other Nonfunctional Requirements**

## **5.1 Performance Requirements**

**PR-1:** Computer

**PR-2:** Monitor

**PR-3:** Keyboard

**PR-4:** Java runtime environment (version 8.0)

## **5.2 Safety Requirements**

Always wear a helmet while operating the software.

### **5.3 Security Requirements**

Only the developers and privileged users should have knowledge of the code that will enable access and manipulation of the Sqlite database.

### **5.4 Software Quality Attributes**

*TBD*

## **6. Other Requirements**

Depending on the copyright legality of some audio and/or video clips used in this program, the developers are responsible to address all legal concerns associated with the use of copyrighted material. This section will be populated with all requirements regarding the use of copyrighted or fair use material used in the TMG.

## **Appendix A: Glossary**

TMG: Trivia Maze Game

PR: Performance Requirement

REQ: Requirement

SI: Software Interface

TBD: To Be Determined

UI: User Interface

## **Appendix B: Issues List**

Section 2.4: Update Operating System Requirements

Section 3: Give More Detailed Information of Classes Used in Each Feature

Section 3.1.4: Functional Requirement of Maze

Section 3.4.2: Stimulus/Response Sequences of Cheats

Section 3.4.3: Functional Requirements of Cheats

Section 3.5.2: Stimulus/Response of Audio/Video Components

Section 3.6.2: Stimulus/Response of Accessing the Database to Make Modifications

Section 4.1: User Interface of Actual TMG

Section 4.3.SI-1.1: User Interface of TMG

Section 5.4: Software Quality Attributes

Section 6: Provide Sources of Audio/Video Components

## **Appendix C: References**

Bafus, Cody; Hall, Austin; Zuelke, Steve. 2020. "Team Coding Standards".