
Software Requirements Specification

for

Trivia Maze

Version 1.0 approved

Prepared by Andrey Melnikov

Macrosoft

5.19.2014

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
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	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Features	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	3
2.7 Assumptions and Dependencies	3
3. System Features.....	3
3.1 ASCII Output of a Maze and Player	3
3.2 Output of a Trivia Question and Answer Input	3
3.3 Locking/Unlocking of Doors Upon Answering a Question	4
3.4 Game is Won Upon Reaching Maze Exit.....	4
3.5 Input of Player Name, game difficulty at Game's Start	5
3.6 Randomly Generated Maze size depending on difficulty	5
4. External Interface Requirements	5
4.1 User Interfaces.....	5
4.2 Hardware Interfaces	6
4.3 Software Interfaces.....	6
4.4 Communications Interfaces	6
5. Other Nonfunctional Requirements	6
5.1 Performance Requirements	6
5.2 Safety Requirements.....	6
5.3 Security Requirements	6
5.4 Software Quality Attributes	7
6. Other Requirements	7
Appendix A: Glossary.....	7
Appendix B: Analysis Models.....	7
Appendix C: Issues List.....	7

Revision History

Name	Date	Reason For Changes	Version
Andrey Melnikov	5.19.2014	Initial Draft	1.0

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Trivia Maze, and includes all of its functionality. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system/game. Unless otherwise noted, all requirements specified here are high priority and committed for release 1.0.

1.2 Document Conventions

This document is written in Times New Roman with a font size of 12 for the main text, and font size of 18 for headers to sections. Each header is also bolded. Subsections are bolded, size 14 font. Classes contained in the Trivia Maze are written in a bold, italic font, whereas methods are just in italics. Every requirement statement has its own priority.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by the developers of the Trivia Maze project. This document contains the overall view of what the Trivia Maze is supposed to do and be. Further, it contains specifics about the project's features, hardware interfaces, and hardware/software requirements. It is recommended that this document is read from start to bottom. Or, if the reader has a particular section he wants more details on in mind, the reader should refer to the table of contents and proceed to the specific section from there.

1.4 Project Scope

The Trivia Maze project is a game that will allow people to sit back and enjoy spending time answering trivia questions as they progress through a maze, trying to get to the exit. For release 1.0, a user will be able to start up the game on a computer running the Windows operating system and then interact with a console based window that draws a maze, presents questions, and takes the user's input.

1.5 References

1. Microsoft, *Microsoft Coding Standards, Version 1.0*.
https://github.com/isaiahgrant/MacroSoftRepo/blob/master/Trivia_Maze_Project/Resources/CodingStandards/Microsoft%20Coding%20Standards.pdf
2. Microsoft, *Trivia Maze Class Diagram, Version 1.0*.
https://github.com/isaiahgrant/MacroSoftRepo/blob/master/Trivia_Maze_Project/Resources/UML_Diagram/Trivia%20Maze%20Class%20Diagram.pdf

2. Overall Description

2.1 Product Perspective

The Trivia Maze project is a new, self-contained product. It was originally created in response to a Software Engineering project by Thomas Capaul in the Spring of 2014.

2.2 Product Features

- Console based input/output via keyboard/monitor
- Display of an ASCII Maze and user's location
- Display of Trivia questions
- Input for movement in maze, and trivia answers
- Variety of trivia questions and trivia categories
- Difficulty setting
- Output of player statistics at the end of a game
- Save/Loading a game

2.3 User Classes and Characteristics

The only anticipated user for this product is a person who will play the Trivia Maze game. This user class will interact with the game via keyboard, and is expected to use all of the product's features, including traversing the maze, winning or losing the game, and being asked/answering trivia questions.

2.4 Operating Environment

The Trivia Maze will operate with the Windows 7 operating system, and any newer releases. The minimum hardware requirements are those required to run the operating system itself. In order to operate the Trivia Maze, the user is required to have a computer monitor, keyboard, and mouse.

2.5 Design and Implementation Constraints

- CO-1: The Trivia Maze's code and maintenance documentation shall conform to the *Microsoft Coding Standards, Version 1,0* [1]
- CO-2: The Trivia Maze shall be written in Java 1.7.
- CO-3: All written Java code shall conform to the *Microsoft Coding Standards, Version 1,0* [1]
- CO-4: The database used will be SQLite
- CO-5: The Trivia Maze must take no more than 20 MB of RAM usage.
- CO-6: The Trivia Maze must respond within 1 second to any user input.

2.6 User Documentation

UD-1: The Trivia Maze will include a tutorial on the game's methods of input, interaction, and basic game flow provided as a pdf with the program.

2.7 Assumptions and Dependencies

AS-1: The user/developer has a working computer and mouse/keyboard for input.

AS-2: The user/developer has Java 1.7 or later installed on their computer.

DP-1: This project uses the SQLite database.

3. System Features

<This template illustrates organizing the functional requirements for the product by system features, the major services provided by the product. You may prefer to organize this section by use case, mode of operation, user class, object class, functional hierarchy, or combinations of these, whatever makes the most logical sense for your product.>

3.1 ASCII Output of a Maze and Player

3.1.1 Description and Priority

During the operation of the Trivia Maze, a graphical depiction (in ASCII) of the maze must be provided at the beginning, and every time the user attempts to move within the maze. In addition to the maze itself, the player's current location within the maze must be indicated within the ASCII output. The maze must be redrawn after a user answers a question as well.

Priority: HIGH

3.1.2 Stimulus/Response Sequences

Stimulus: User attempts to move or answers a question

Response: The Maze is redrawn with the player inside it

3.1.3 Functional Requirements

REQ-1: Game must have been started

REQ-2: Player name, and chosen difficulty obtained from player

REQ-3: Maze created according to difficulty

3.2 Output of a Trivia Question and Answer Input

3.1.1 Description and Priority

When the player attempts to move to a door he has not yet visited, he needs to be presented with a trivia question. The player should then be able to type in his answer (a single letter), whereupon he will find out if his answer was correct or not.

Priority: HIGH

3.1.2 Stimulus/Response Sequences

Stimulus: Player attempts to move to an unvisited door.

Response: Trivia Question is displayed, waiting for the player's answer.

Stimulus: Player provides an answer to the question as a single letter.

Response: Game informs player if his answer was correct or not.

3.1.3 Functional Requirements

REQ-1: Trivia Questions and their answers must be obtainable from doors.

REQ-2: User must be able to provide input, and the game process that input as a correct, or incorrect answer.

3.3 Locking/Unlocking of Doors Upon Answering a Question

3.1.1 Description and Priority

After a player answers a Trivia Question, if he is correct, the door is unlocked, and will not ask him a question again. If he is incorrect, the door is locked and the player will not be able to go past that door in the maze.

Priority: HIGH

3.1.2 Stimulus/Response Sequences

Stimulus: Player answers a Trivia Question correctly.

Response: Door is unlocked and player may proceed through it.

Stimulus: Player answers a Trivia Question incorrectly.

Response: Door is locked and player may not proceed through it.

3.1.3 Functional Requirements

REQ-1: Doors must be able to lock/unlock themselves.

REQ-2: Game must detect locked/unlocked doors.

REQ-3: Player must be able to input his answer to a Trivia Question.

3.4 Game is Won Upon Reaching Maze Exit

3.1.1 Description and Priority

Upon reaching the Maze's exit, the game must end and inform the player that he won.

Priority: HIGH

3.1.2 Stimulus/Response Sequences

Stimulus: Player moves into the exit in the maze.

Response: Game ends, informing player he won.

3.1.3 Functional Requirements

REQ-1: Maze must know its exit coordinate

REQ-2: Game must detect when player is at exit

REQ-3: Game must provide game won functionality.

3.5 Input of Player Name, game difficulty at Game's Start

3.1.1 Description and Priority

Upon starting a game, a player needs to input his name and chosen game difficulty. The difficulty will be used in constructing the maze, and the name is to be able to refer to the player throughout the game.

Priority: MEDIUM

3.1.2 Stimulus/Response Sequences

Stimulus: Player starts a new game.

Response: Game requests player's name, then difficulty level.

3.1.3 Functional Requirements

REQ-1: Game must store player's name/difficulty.

REQ-2: Game must be able to detect a new game

REQ-3: Game must be able to request a player's name/difficulty.

3.6 Randomly Generated Maze size depending on difficulty

3.1.1 Description and Priority

Given a difficulty of EASY, MEDIUM, HARD, or EXTREME, the maze needs to create itself accordingly, becoming bigger the higher the difficulty level.

Priority: MEDIUM

3.1.2 Stimulus/Response Sequences

Stimulus: Player inputs difficulty

Response: Maze is generated according to that difficulty

3.1.3 Functional Requirements

REQ-1: Game needs to be able to take player difficulty input

REQ-2: Maze Builder must be able to create different sized mazes.

4. External Interface Requirements

4.1 User Interfaces

The console output of the game must be able to fit the drawn Maze, prompt for the user asking them what they would like to do, a list of available actions, and a line for player input. A Trivia Question must also be able to fit in this same console size, as well as a line for player input.

All input for release 1.0 will be through the keyboard, and the output in a console based window.

4.2 Hardware Interfaces

The Trivia Maze needs to support any PC running Windows 7 or later. No special Hardware Interfaces are needed.

4.3 Software Interfaces

The Trivia Maze interacts with a SQLite data base that is stored locally on a user's computer. The SQLite database is used to store and retrieve the Trivia Questions/Answers used by the Trivia Maze.

For release 1.0, the Trivia Maze depends on access to the Windows operating system's Command Line as that is where all of its output goes, along with input.

4.4 Communications Interfaces

The release 1.0 of Trivia Maze has no capability to connect to any network, and is completely self contained.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

At any given player input, the Trivia Maze must respond within 1 second, and be ready to take further input. This means that all operations of the Trivia Maze must be completed within 1 second to ensure enjoyable game play. Drawing the maze, getting player input, validating input, determining if the maze is winnable, all of these must not take more than 1 second at the very most.

5.2 Safety Requirements

This Trivia Maze is required to be played in a safe, dry, relatively cool environment. Under no circumstances should the user play Trivia Maze while in a bathtub or other aquatic environment. Performance is not guaranteed in this situation, and might result in death.

5.3 Security Requirements

No security requirements have been identified as the product does not store any private information, or have connection to network capabilities.

5.4 Software Quality Attributes

The Trivia Maze must be easy to learn and use, and reliable. Rarely should the Trivia Maze crash. The Trivia Maze must consume little memory and be available for all versions of Windows including and past Windows 7.

6. Other Requirements

TDB

Appendix A: Glossary

TDB

Appendix B: Analysis Models

For UML Class Diagram, see [2].

Appendix C: Issues List

- Player movement validation needs to be fixed.
- Maze Builder needs to create different sized mazes depending on difficulty.
- Drawing maze needs to be fixed – currently doesn't draw player or entire maze.