Software Requirements Specification

for

Computer Science Trivia Maze

Version 1.0

Prepared by Stefan Bostain, Stacy Carlson, and Dan Watt

java.lang.NullPointerException

May 15, 2014

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Stefan Bostain, Stacy Carlson, and Dan Watt	5/15/14	Initial Draft	1.0 draft 1

1. Introduction

1.1 Purpose

This SRS describes the functional and nonfunctional requirements for the Computer Science Trivia Maze software version 1.0. This document is intended to be used by members of the project team to implement and test the software. All specifications listed in this document are assumed high priority and required for the 1.0 release to the client.

1.2 Intended Audience and Reading Suggestions

This document is intended for the developers. The rest of this SRS defines the features of the software, the GUI, and the requirements from the client. The developer should read the SRS in the order provided with particular care being given to the System Features and the External Interface Requirements.

1.3 Project Scope

The Computer Science Trivia Maze will entertain and test the knowledge of users by asking computer science trivia questions in order to navigate through a maze.

2. Overall Description

2.1 Product Perspective

The Computer Science Trivia Maze is a new game that combines the fun of answering trivia questions and navigating mazes.

2.2 Product Features

The Computer Science Trivia Maze features trivia questions in the form of short answer, true or false, and multiple choice. Users can play in endless mode, record their scores on a leader board, and add custom questions and answers.

2.3 User Classes and Characteristics

Player (favored)   A Player is anyone who wishes to play the Computer Science Trivia Maze. All Players have the basic computer skills necessary to install an .exe and interact with the software via a keyboard and mouse. Players will have a more enjoyable experience using the Computer Science Trivia Maze if they have some knowledge of Computer Science, but it is not required.

2.4 Operating Environment

OE-1: The Trivia Maze shall operate on the following operating systems: Windows

2.5 Design and Implementation Constraints

CO-1: All code will be written in Java and the developers will utilize the Eclipse IDE.
CO-2: The GUI will be developed using WindowBuilder and SWT.
CO-3: A sqlite database must be used.

2.6 User Documentation

UD-1: The game will include a pop up screen with instructions on how to play that will be accessed via a help menu.

3. System Features

3.1 Standard Game Play

3.1.1 Description and Priority

A Player answers Computer Science Trivia Questions to navigate through the maze. Priority = High

3.1.2 Stimulus/Response Sequences

Stimulus: Player selects a direction to move (up, down, left, right).
Response: If the direction is invalid due to out-of-bounds or locked door, an error message is displayed. Otherwise a question is displayed with answer options.

Stimulus: Player answers the question correctly.
Response: Door is unlocked and Player's current location is updated. If the game has been won, the Player is informed.

Stimulus: Player answers the question incorrectly.
Response: Door is permanently locked and Player remains in the same location. Player is informed if winning the game is no longer possible.

3.1.3 Functional Requirements

Maze Display:	The Player will be able to see their location in the maze and the possible directions to move. Unvisited doors will be black, closed doors will be red, and open doors will be green.
Movement Options:	The Player will be able to click on GUI directional arrows (up, down, left, right) in order to move. The Player will also be able to use the keyboard arrows. The Player will be informed via popup message of invalid movements.
Question and Answer Database:	The sqlite database will be loaded with questions and their answers.
Question and Answer Display:	The GUI will display the question and answers. The Player will be able to select or type in their answer. Any keyboard input will be scrubbed to protect against SQL injection.

3.2 Endless Game Play

3.2.1 Description and Priority

A Player answers Computer Science Trivia Questions until an incorrect answer is entered. Scores are recorded on a leader board. Priority = Medium

3.2.2 Stimulus/Response Sequences

Stimulus: Player answers the question correctly.

Response: The next question is presented.

Stimulus: Player answers the question incorrectly.
Response: The game is terminated. If the score is larger than the ones on the leader board, the Player is invited to enter their name. If the score is smaller than the ones on the leader board, the Player is invited to try again.

3.2.3 Functional Requirements

Question and Answer Database: The sqlite database will be loaded with questions and their answers. If a Player completes all of the questions correctly they will automatically be entered on the leader board.

Question and Answer Display: The GUI will display the question and answers. The Player will be able to select or type in their answer. Any keyboard input will be scrubbed to protect against SQL injection.

Leader Board Entry: The sqlite database will be loaded with the leader board. The Player will be able to type in their name. The input will be scrubbed to protect against SQL injection.

4. External Interface Requirements

4.1 User Interfaces
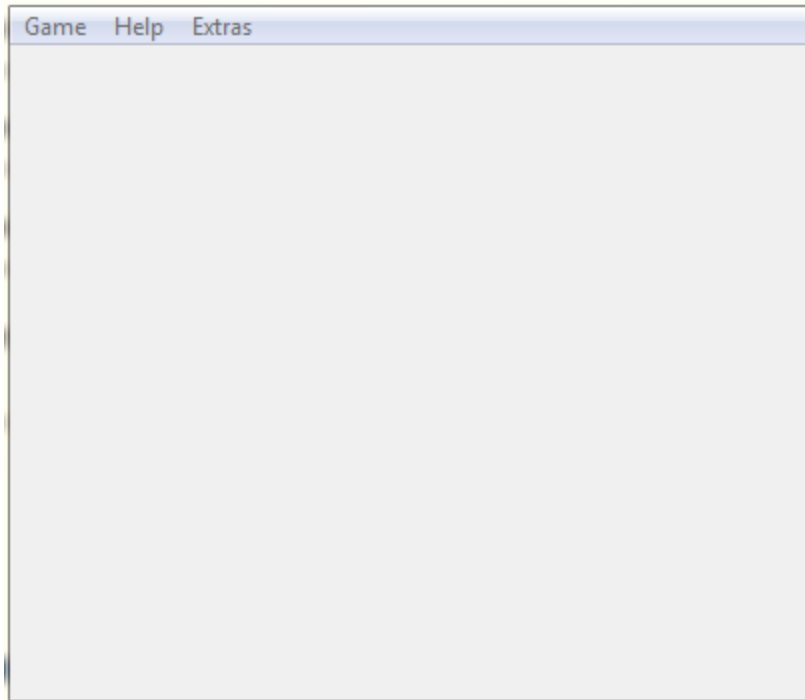
UI-1: The Computer Science Trivia Maze will always provide a menu bar that provides the Player the option to start a game in standard or endless modes, quit the program, get help with playing the game, view information about the software, view the leader board, and add custom questions and answers.

4.2 Hardware Interfaces

No hardware interfaces have been identified.

4.3 Software Interfaces

SI-1:   Question and Answer Database
SI-1.1:   The Question and Answer Database will store the preloaded questions and answers.
SI-1.2:   The Question and Answer Database will store user's custom questions and answers.
SI-1.3:   The Computer Science Trivia Maze will programmatically access the database in order to display questions and answers.
SI-1.4:   The Computer Science Trivia Maze will programmatically access the database in order to determine the correct answer to a question.

SI-2:   Leader Board Database

SI-2.1:   The Leader Board Database will store the scores and names of the top ten players.

SI-2.2:   The Computer Science Trivia Maze will programmatically access the database in order to determine if a Player has earned a high score. If a high score has been obtained, the database will store the score and the Player's name.

4.4 Communications Interfaces

No communications interfaces have been identified.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

No performance requirements have been identified.

5.2 Safety Requirements

SR-1:   The Player will be warned that preloaded question and answers have been compiled by the developers with every attempt at accuracy, but errors still may exist.

SR-2:   The Player will be warned when entering custom questions and answers that no error checking has been done and they are responsible for the accuracy.

5.3 Security Requirements

SE-1:   The Computer Science Trivia Maze shall prevent Player access to the Question and Answer Database and the Leader Board Database due to tampering concerns.

5.4 Software Quality Attributes

Correctness-1:   The Computer Science Trivia Maze should contain 100% correct content.

Usability-1:   The Computer Science Trivia Maze must function 100% correctly in Standard Game Play for release 1.0.

6. Other Requirements

This section might be fleshed out later with database 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: Analysis Models

UML will go here….

<Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.>