
Software Requirements Specification

for

Trivia Maze Game

Version 1.0 unapproved

Prepared by Richard S., Andrew M., Ana K., Kevin R.

CSCD350

May 15, 2014

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Project Scope and Product Features	1
1.3 References.....	1
2. Overall Description.....	1
2.1 Product Perspective	1
2.2 User Classes and Characteristics	1
2.3 Operating Environment.....	1
2.4 Design and Implementation Constraints.....	1
2.5 User Documentation	Error! Bookmark not defined.
2.6 Assumptions and Dependencies	1
3. System Features.....	Error! Bookmark not defined.
3.1 Order Meals	Error! Bookmark not defined.
3.2 Create, View, Modify, and Delete Meal Subscriptions	Error! Bookmark not defined.
3.3 Register for Meal Payment Options	Error! Bookmark not defined.
3.4 Request Meal Delivery	Error! Bookmark not defined.
3.5 Create, View, Modify, and Delete Cafeteria Menus	Error! Bookmark not defined.
4. External Interface Requirements	Error! Bookmark not defined.
4.1 User Interfaces	2
4.2 Hardware Interfaces.....	Error! Bookmark not defined.
4.3 Software Interfaces	3
4.4 Communications Interfaces	Error! Bookmark not defined.
5. Other Nonfunctional Requirements.....	Error! Bookmark not defined.
5.1 Performance Requirements.....	3
5.2 Safety Requirements.....	3
5.3 Security Requirements.....	Error! Bookmark not defined.
5.4 Software Quality Attributes.....	3
Appendix A: Data Dictionary and Data Model.....	3
Appendix B: Analysis Models.....	5

Revision History

Name	Date	Reason For Changes	Version
Group	5/15/14	initial draft	1.0 draft 1

1. Introduction

1.1 Purpose

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

1.2 Project Scope and Product Features

TMG randomly generates a maze. The player must correctly answer trivia questions in order to progress to the exit. The player has a health status and key inventory. The player loses the game if they run out of health, deducted after each wrong answer. The player wins the game if they reach the exit point. Each maze room prompts the player with a question. TMG uses simple windows to display the maze, trivia questions, and answers.

1.3 References

???

2. Overall Description

2.1 Product Perspective

MTG is designed for single player use. It is expected to eventually save player progress and track past performance through a simple leader board. It uses a connection to a small database to retrieve questions and answers.

2.2 User Classes and Characteristics

Insert context diagram?

2.3 Operating Environment

- OE-1: TMG shall operate on Windows OS.
- OE-1: Code shall be written in Java, including the database setup via a JDBC:SQLite driver.

2.4 Design and Implementation Constraints

- DC-1: TMG is designed for single player use.
- DC-2: The maze is square with odd numbered dimensions, and shall be no larger than 13X13.
- DC-3: The trivia database shall contain no fewer than 50 questions and no more than 100 questions.

2.5 Assumptions and Dependencies

- AS-1: The player knows how to read English and is able to answer trivia questions.
- AS-2: The user can run Java programs.
- AS-3: The user has SQLite DB and appropriate drivers installed.

2.6 Playing the Game

2.5.1 Description and Priority

The maze is randomly generated on each attempt. Priority = 1.

The player navigates through the maze using keyboard arrow directions. Priority = 1.

The player can see their progress through the maze.

The player can see their health and key inventory.

The player reads and answers trivia questions in a simple window.

2.5.2 Stimulus/Response Sequences

- Stimulus: Player requests to play the game.
- Response: Maze is randomly generated and displayed.
- Stimulus: Player uses keyboard arrow keys to move .
- Response: Maze updates player location & prompts question.
- Stimulus: Player choses wrong answer.
- Reponse: Player's health decreases.
- Stimulus: Player choses correct answer.
- Response: Player may gain a key & has option to move again.
- Stimulus: Player reaches the exit location.
- Response: The game ends in a victory.

2.5.3 Functional Requirements

3. System Features

3.1 ???

4. External Interface Requirements

4.1 User Interfaces

- UI-1: A simple start/quit menu with active buttons is displayed to the console.
- UI-2: The player's maze and location are displayed to the console.
- UI-3: The user navigates the maze using the keyboard arrow keys.

UI-4: The user sees trivia questions and selects answers in a simple window displayed to the console.

4.2 Software Interfaces

SI-1: TMG shall connect to the SQLite DB via JDBC:SQLite driver.

5. Other Non-functional Requirements

5.1 Performance Requirements

PE-1: TMG shall allow 1 user to play the game.
 PE-2: The user may choose which direction they wish to travel within the maze confines.
 PE-3: The user may use keys to open doors as a shortcut.
 PE-4: TMG shall display trivia in a text box, and the user shall select answers using radio buttons.

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Software Quality Attributes

Availability-1: TMG shall be available for Windows OS.

Appendix A: Key Terms

Player	Traverses the maze, answers questions
Maze	Randomly generated path surrounded by walls and blocked by doors. Begins with a start, ends with an exit.
Path	Divided into rooms, each prompting the player with a question.
Exit	The player's destination to win the game.
Door	A path obstruction opened with a key.
Key	Opens a door. Obtained after entering a room that contains it and answering a question correctly.
Health	Player attribute that starts full, decreases with wrong answers, potentially increases with correct answers
Database	File that stores trivia questions, possible answers, and the correct answers.

Appendix B : Diagrams & Tables

Question : String	Answer1 : String	Answer2 : String	Answer3 : String	CorrectAnswer : String
-------------------	------------------	------------------	------------------	------------------------

Table 1: Database Table Schema

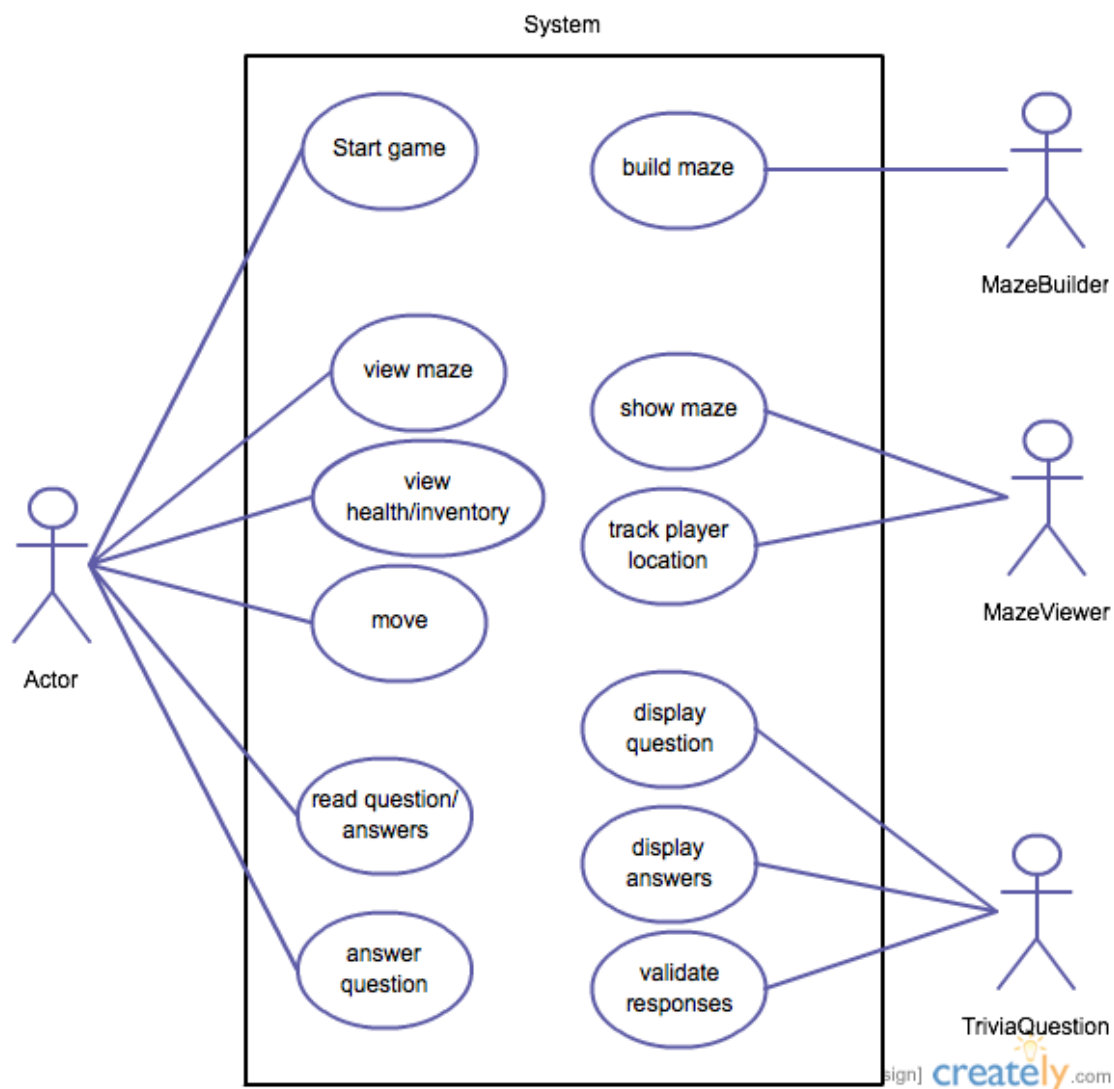


Figure 1 Use Case Diagram