Software Requirements Specification

for

Trivia Maze Game

Version 1.0 approved

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

CSCD350

June 11, 2014

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Triploblastic	5/15/14	initial draft	1.0
Triploblastic	6/11/14	Final 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 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 stated, all features covered in this document are to be completed by the first distribution.

1.2 Project Scope and Product Features

TMG randomly generates and displays a maze graphic. The player directs a character through the maze towards the exit by correctly answering trivia questions. The player has a health status, points and key inventory. Health is deducted after each wrong answer. The player loses if they run out of health. The player wins the game if they reach the exit. As the player moves into unexplored squares, there is approximately a 50% chance of encountering a trivia prompt. For every square traversed, either a key is picked up (15% chance), health is regained (30% chance), or points are granted (55% chance). The player may use a key to pass through a gate. TMG uses simple pop-up windows to display the maze, trivia question prompts, and leader board.

1.3 References

Maze generation code modified from http://www.migapro.com/depth-first-search/

2. Overall Description

2.1 Product Perspective

TMG must be installed and run as an executable for Windows OS. TMG is designed for single player use. It tracks past performance through a simple leader board that is serialized between game executions. TMG uses a connection to a small database to retrieve questions and answers that are used to create prompts.

2.2 Operating Environment

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

2.3 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 15X15.
DC-3: The trivia database shall contain no fewer than 50 questions and no more than 100 questions.
DC-4: The leader board will contain no more than the top 10 scores.

2.4 Assumptions and Dependencies

AS-1: The user can properly install the TMG package on a system running Windows OS.
AS-2: The user knows how to read English and is able to answer trivia questions.
AS-3: The user has a keyboard and mouse installed.

2.5 Playing the Game

2.5.1 Description and Priority

Priority values are 1-5, with 1 being highest. Priorities < 3 must be implemented in version 1.0.

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

The player navigates through the maze using keyboard ASWD keys. Priority = 1

The player can see their progress through the maze. Priority = 1

The player can see their health, points and key inventory. Priority = 1

The player reads and answers trivia questions in a pop-up window. Priority = 1

The player can see a leader board at the end of the game. Priority = 2

A fog-of-war conceals unexplored areas in the game. Priority = 3

2.5.2 Stimulus/Response Sequences

Stimulus: Player executes TMG.
Response: Maze is randomly generated and displayed along with character start location, key inventory, and points.

Stimulus: Player uses keyboard arrow keys to move.
Response: Maze checks move validity.
If True: If true, there is a 50% chance that a trivia prompt displays, and one of the following may occur: health increases by 1, a key is added to the inventory, or 100-149 points are added to the total. In the absence of a question, the player is allowed to progress and the location is updated.
If False: Nothing changes & the player does not progress.

Stimulus: Player choses wrong answer.
Response: Player's health decreases by 1.

Stimulus: Player choses correct answer.
Response: Player's location is updated and player has option to move again.

Stimulus: Player moves towards a gate.
Response: Maze checks if a key is available from the inventory.
If True: A pop-up window asks if the player wants to use a key.
If Yes: Key is removed from inventory, player moves to the gate square.
If No: Nothing changes & the player does not progress.

If False: Nothing changes & the player does not progress.

Stimulus: Player reaches the exit location.
Response: Player wins the game & header board is displayed.

Stimulus: Player has a top-ten score.
Response: Input dialog box appears and asks for player's name.

Stimulus: Player health reaches zero.
Response: Option dialog box appears and prompts for a new game.

3. External Interface Requirements

3.1 User Interfaces

UI-1: A simple continue/quit option dialog box is displayed to the console upon game win/lose event.
UI-2: The maze, player inventory, points, and location are displayed with graphics.
UI-3: The user navigates the maze using the keyboard arrow keys ASWD.
UI-4: The user reads trivia questions and selects answers in a customized dialog box.
UI-5: The user inputs their name for the leader board using an input dialog box.

3.2 Software Interfaces

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

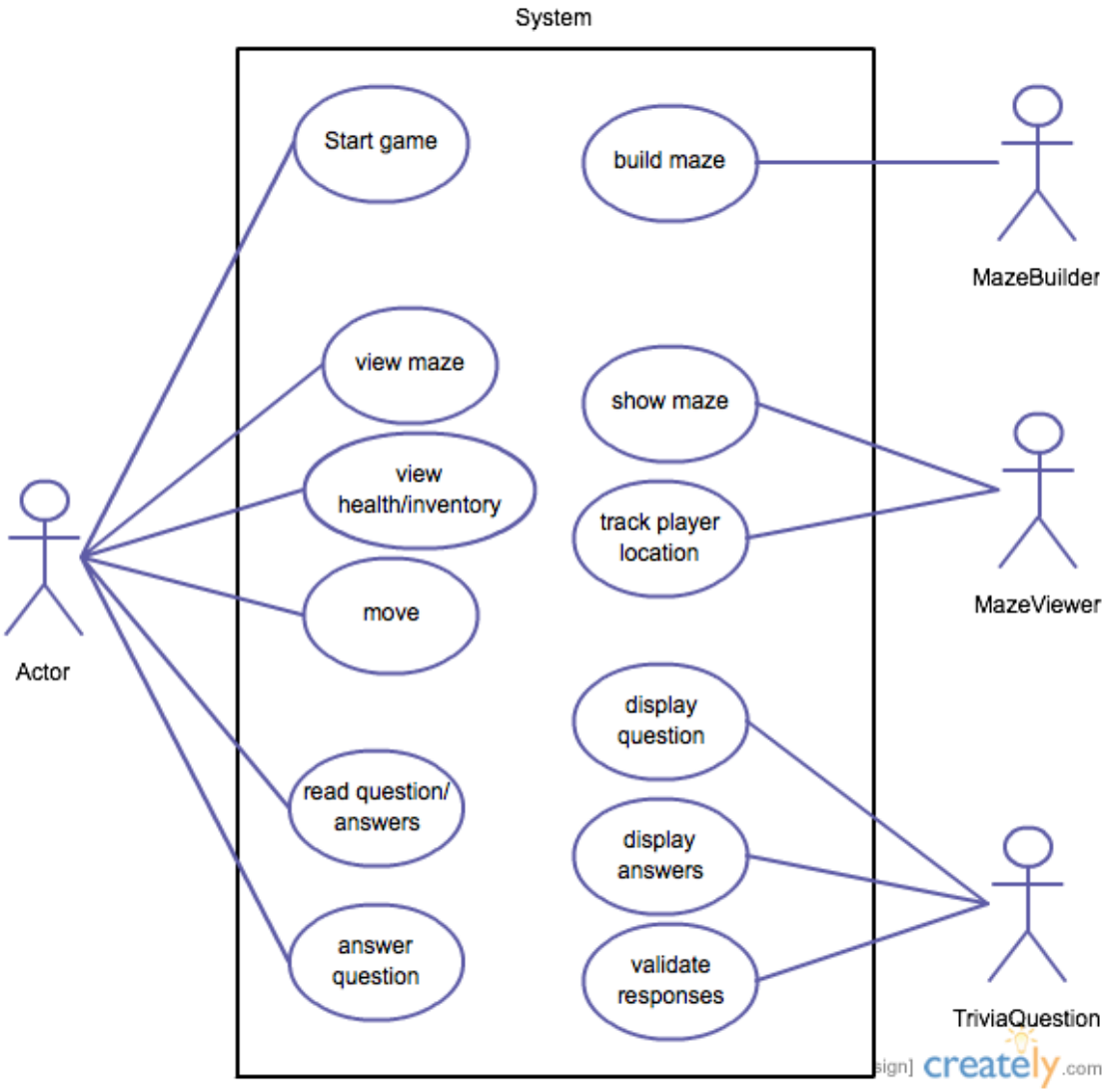
Appendix A: Key Terms

Player	Traverses the maze and answers questions. Depicted as a collection of sprites with N,S,E,W orientations. Has health and key inventory.
Maze	Randomly generated path surrounded by walls and blocked by doors. Begins with a start, ends with an exit.
Path	Divided into rooms, each of which may prompt the player with a question, increase health, or provide a key.
Exit	The player's destination to win the game.
Door/Gate	A path obstruction opened with a key.
Key	Opens a door. Obtained after entering a room that contains it (15% of the time).
Health	Player attribute that starts full at 3, drops by 1 with wrong answers, & rises by 1 if entering a room with a healing effect (30% of the time).
Database	File that stores trivia questions, possible answers, and the correct answers.

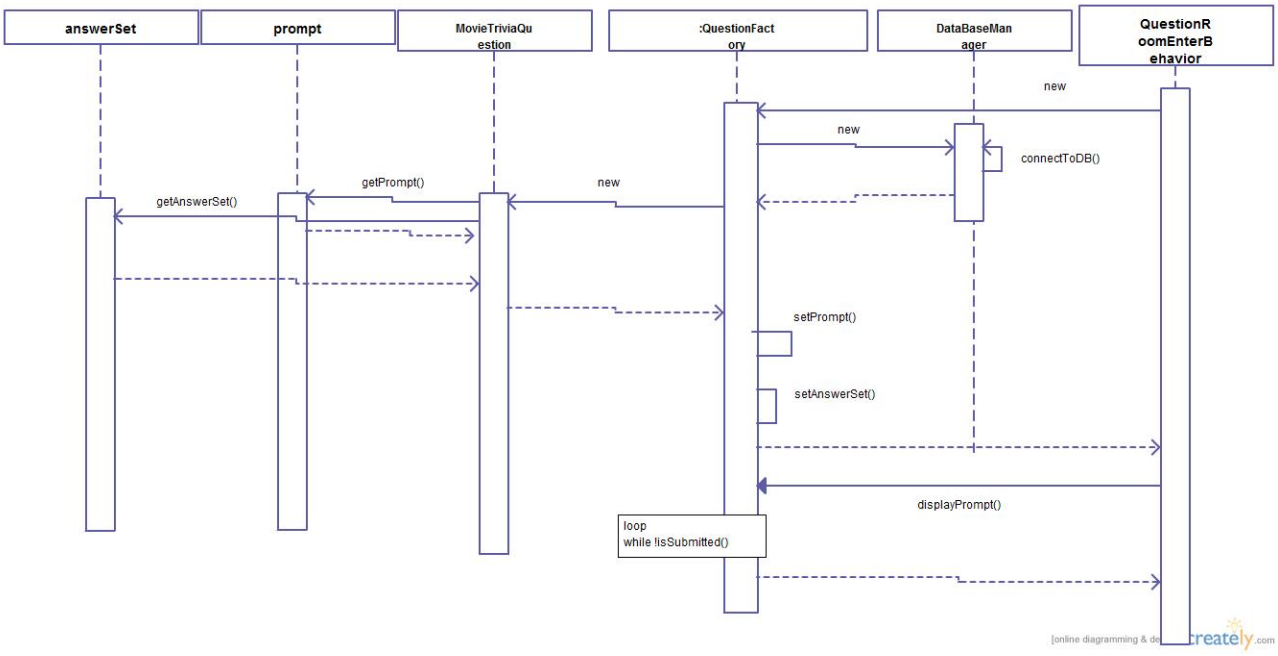
Appendix B : Diagrams & Tables

Table 1MovieTrivia Database schema

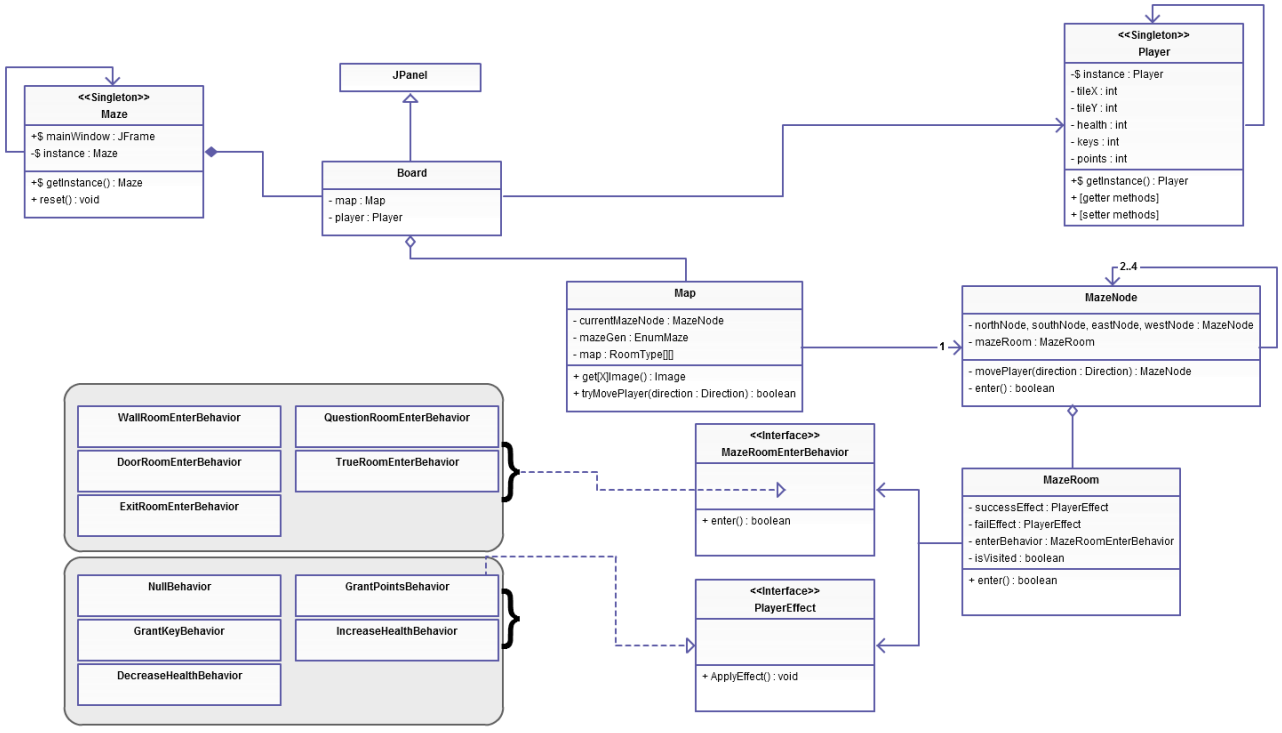
ID : Integer	Question : String	Answer : String	Wrong Option1 : String	Wrong Option2 : String	Wrong Option 3 : String



Sequence Diagram for creating a TriivaQuestion



Maze Interaction Class Diagram



Maze State Diagram for Playing