Software Requirements Specification

for

Trivia Maze

Version 1.0 approved

Prepared by Jeff Clark
Cody Demianew
JD Hall
Beto Lopez

May 15, 2014

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Cody Demianew	05/15/14	initial draft	1.0 draft 1
Cody Demianew	06/10/14	Final Draft	1.1 draft 2

1. Introduction

1.1 Purpose

This SRS describes the software functional and nonfunctional requirements for release 1.0 of the Trivia Maze. 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

The Trivia Maze will allow users to test their knowledge by navigating a maze and comparing their scores to others. The section in that document titled "Scope of Initial and Subsequent Releases" lists the features that are scheduled for full or partial implementation in this release.

2. Overall Description

2.1 Product Perspective

The Trivia Maze is a new game that allows users to have fun while testing their knowledge. The game is expected to evolve over several releases.

2.2 User Classes and Characteristics

Player — A Player is a end users working their way through the maze with the intent to get finish the maze.

Maze — The Maze provides a static collection of rooms and doors. It also provides an interface for the Player to move between them.

Question — The Question provides the Player a randomly selected question from our database, for which the Player must answer correctly to unlock the door to move on to the next room. If answered incorrectly, they door is locked.

2.3 Operating Environment

OE-1: Windows 7 and newer system is required.

2.4 Design and Implementation Constraints

CO-1: The game's code and comments shall conform to the *Coding Standards Documentation.*

CO-2: The program will be written in C# using the .Net framework *v*4.5.

2.5 User Documentation

UD-1: The game will have a how play page that they may access on the main screen.

2.6 Assumptions and Dependencies

AS-1: The user has a mouse that is properly working and plugged in to the computer.
DE-1: The operation of the game depends on having a database of questions and answers.

3. System Features

3.1 Play a Game

3.1.1 Description

Creates a maze for the Player to click on doors, answer questions, and traverse the maze.

3.1.2 Question/Answer Sequences

Question:	Player clicks on adjacent room to receive new question.
Answer:	Maze prompts Player a question and waits for an answer.
Question:	Player selects a answer an submits it.
Answer:	If correct player moves to the new room, if incorrect the door is locked.

3.1.3 Functional Requirements

Play:	Initializes the maze upon clicking the play button.
Play.Initials:	Prompts the user for their initials before starting the game.
Play.Initials.Start:	Displays maze and places player in the start position in the maze.
Game.Question:	Prompts the Player for a question that they need to answer move to new room.
Game.Question.Answer:	If question is answered correctly, the player will advances to the next room. If the question is answered incorrectly, the door is locked.

4. External Interface Requirements

4.1 User Interfaces

UI-1: Using a mouse or keyboard.

4.2 Software Interfaces

SI-1: Using a SQLite for storage of questions,and answers.

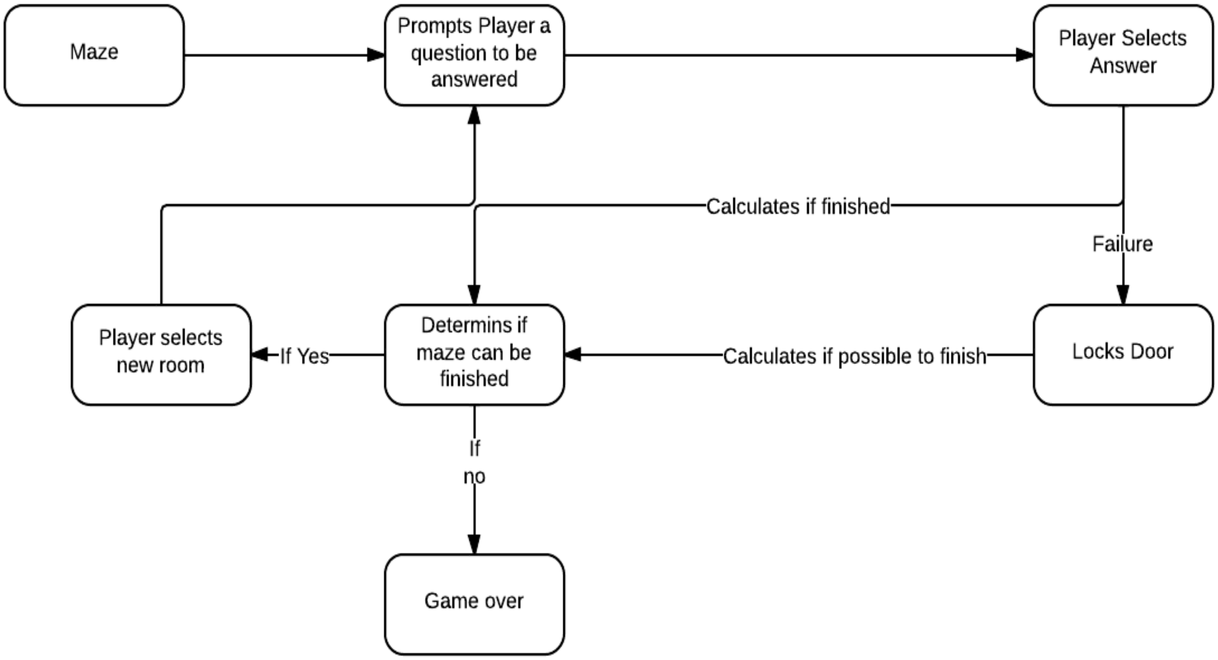
5. Other Nonfunctional Requirements

5.1 Security Requirements

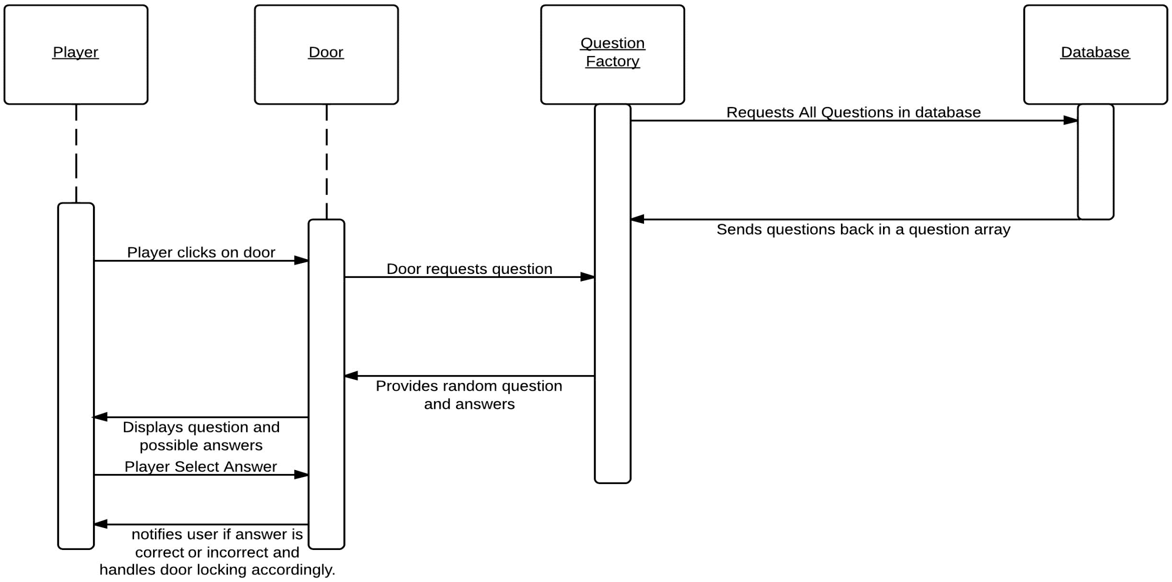
SE-1: User has no access to the database if at all possible.

Appendix A: Data Dictionary and Data Model

State:



Sequence:


Player
Door
Question Factory
Database
Requests All Questions in database
Sends questions back in a question array
Player clicks on door
Door requests question
Provides random question and answers
Displays question and possible answers
Player Select Answer
notifies user if answer is correct or incorrect and handles door locking accordingly.