

Software Requirements Specification

for

Trivia Maze

Version 1.1 approved

Prepared by Nathan Graham
Kyle Johnson
Eric Laib
Daniel Moore

Twenty Hats

11/7/14

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	2
2.7 Assumptions and Dependencies	3

3. System Features.....	3
3.1 System Feature 1	3
3.2 System Feature 2 (and so on)	4
4. External Interface Requirements.....	4
4.1 User Interfaces	4
4.2 Hardware Interfaces	4
4.3 Software Interfaces	4
4.4 Communications Interfaces	4
5. Other Nonfunctional Requirements.....	5
5.1 Performance Requirements	5
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
6. Other Requirements.....	5
Appendix A: Glossary.....	5
Appendix B: Analysis Models.....	6
Appendix C: Issues List.....	6

Revision History

Name	Date	Reason For Changes	Version
Kyle Johnson	11/7/14	Initial Draft	1.0
Daniel Moore	12/8/14	Final Draft	1.1

- **Introduction**
- **Purpose**

This is a Software Requirements Specification document for Trivia Maze. This document is intended for use by the team members of Twenty Hats for the purposes of the development process.

- **Intended Audience and Reading Suggestions**

This document will be composed by developers for the reading by the developers.

- **Project Scope**

Trivia Maze is a game to be enjoyed by children and adults alike, given that they enjoy video games and having their knowledge of such tested. This project is intended for the enjoyment of the player and for the education of the developers involved.

- **References**

1. <http://penguin.ewu.edu/cscd350/>
2. www.pivotaltracker.com
3. www.github.com/comebacktomebaby

- **Overall Description**

- **Product Perspective**

Trivia Maze is a new product developed for a software development class at Eastern Washington University.

- **Product Features**

Trivia Maze utilizes a procedural maze generation algorithm to increase replayability value. Mazes are composed of rooms containing one to four doors. As the player navigates the maze, the gender-neutral individual will be asked video game trivia questions. Correct answers will allow the player to pass through the door and incorrect answers will lock the door permanently. The player's objective is to traverse the maze and find the exit. If the maze becomes impossible to complete, the player will lose the game.

- **User Classes and Characteristics**

Player ---> Game <--[maintenance and post-release development]--Developer

Players	People playing the game
Developers	People developing the game

- **Operating Environment**

Trivia Maze is developed for Windows and requires the .NET framework version 4.0.

- **Design and Implementation Constraints**

The given time frame for the development process will limit the content included in the final product.

- **User Documentation**

There will be a help section available to the player that includes all necessary how-to-play documentation and general information about the project.

- **Assumptions and Dependencies**

It is assumed that the player will be able to use a keyboard and mouse and have some knowledge of video game trivia. There are no prior software dependencies.

- **System Features**

- **Create, Save, and Load Maze**

- 3.1.1 Description and Priority**

- Maze creation is essential for gameplay and is a primary priority. The saving and loading of maze games will be implemented later in the development process for the convenience of the player.

- 3.1.2 Stimulus/Response Sequences**

- The option to create, save, and load mazes will be accessible through the file menu of the application.

- 3.1.3 Functional Requirements**

- REQ-1: Maze Generation Algorithm

- REQ-2: Serialization of Maze Data

- **Maze Navigation**

- 3.2.1 Description and Priority**

- The maze can be navigated with the arrow keys or the mouse. As said in the how to play inside of the game.

- 3.2.2 Stimulus/Response Sequences**

- N/A

- 3.2.3 Functional Requirements**

- N/A

- **Trivia Question Interaction**

- 3.3.1 Description and Priority**

- Trivia question will be generated by the database from SQL. We will fill the database with the questions then the program will interact with the database to get the question for the game.

- 3.3.2 Stimulus/Response Sequences**

- N/A

- 3.3.3 Functional Requirements**

- N/A

- **External Interface Requirements**

- **User Interfaces**

- UI-1: Menu

- UI-2: Trivia Question Box

- UI-3: Room Display

- UI-4: Mini Map

- **Hardware Interfaces**

HI-1: Keyboard

HI-2: Mouse

HI-3: Monitor, Speakers, Computer, Chair, User, Mountain Dew

- **Software Interfaces**

SI-1: SQLite Database

SI-2: .NET Framework (version 4.0)

- **Communications Interfaces**

Text boxes will pop up to tell the user information as the game is being played.

- **Other Nonfunctional Requirements**

- **Performance Requirements**

PE-1: The system will accommodate a single user.

PE-2: Given a modern computer, game features will load and execute within 2 seconds.

- **Safety Requirements**

SR-1: Don't cross the streams.

- **Security Requirements**

SE-1: Prevent SQL injection attacks

SE-2:

- **Software Quality Attributes**

QL-1: All trivia questions will have correct answers as determined by the developers.

QL-2: Short answer trivia questions will be straightforward

QL-3: The graphical user interface will be intuitive for user interaction

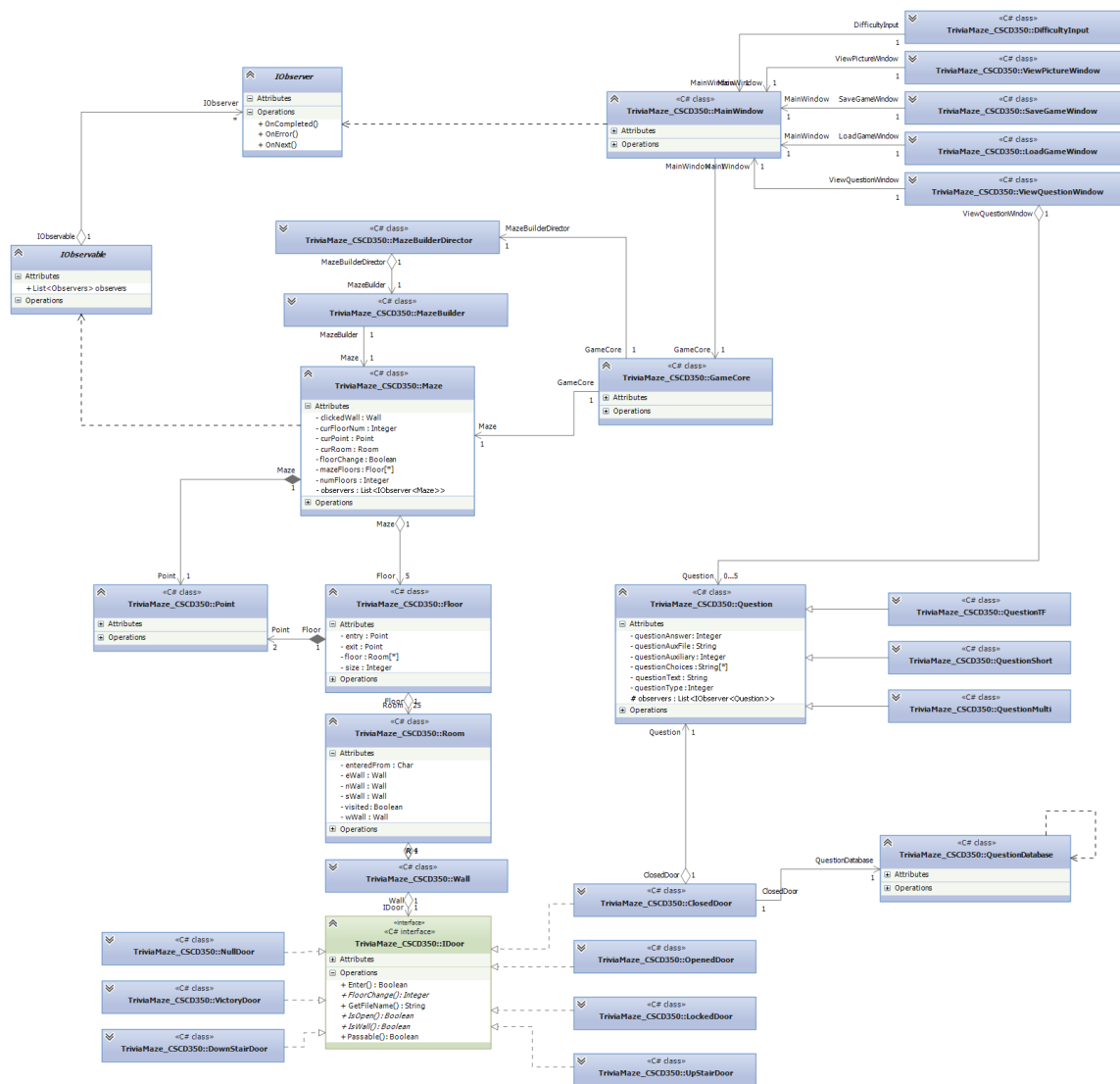
- **Other Requirements**

ET-1: TBD

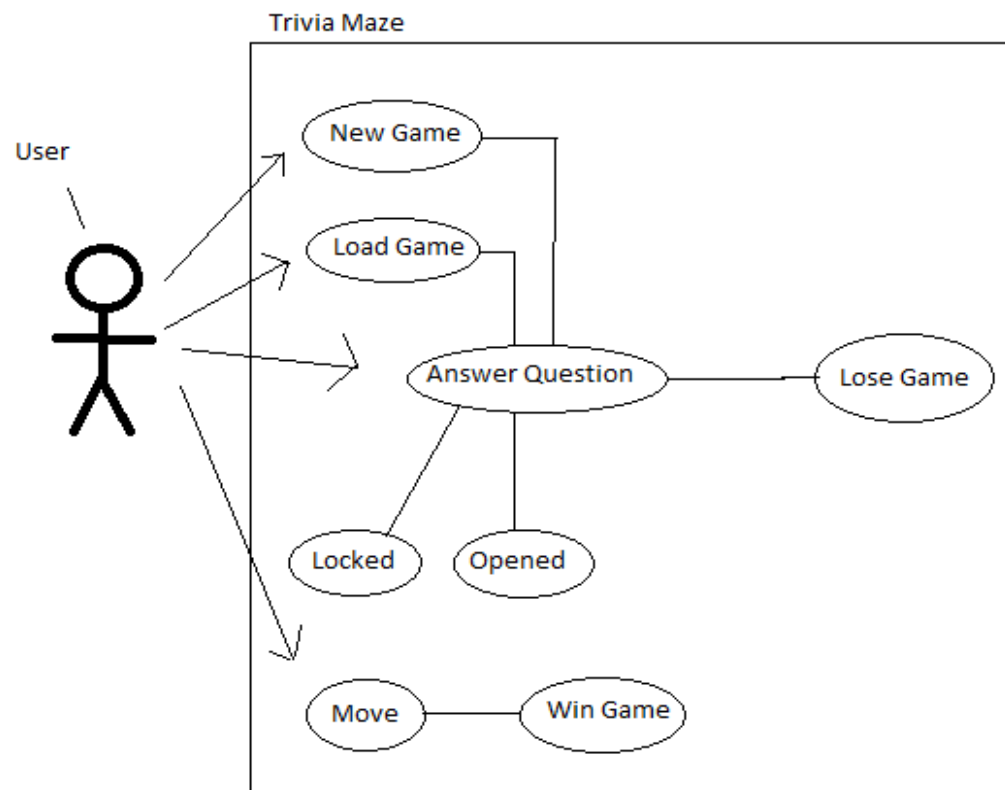
Appendix A: Glossary

SETM9001 = Trivia Maze

Appendix B: Class Diagram

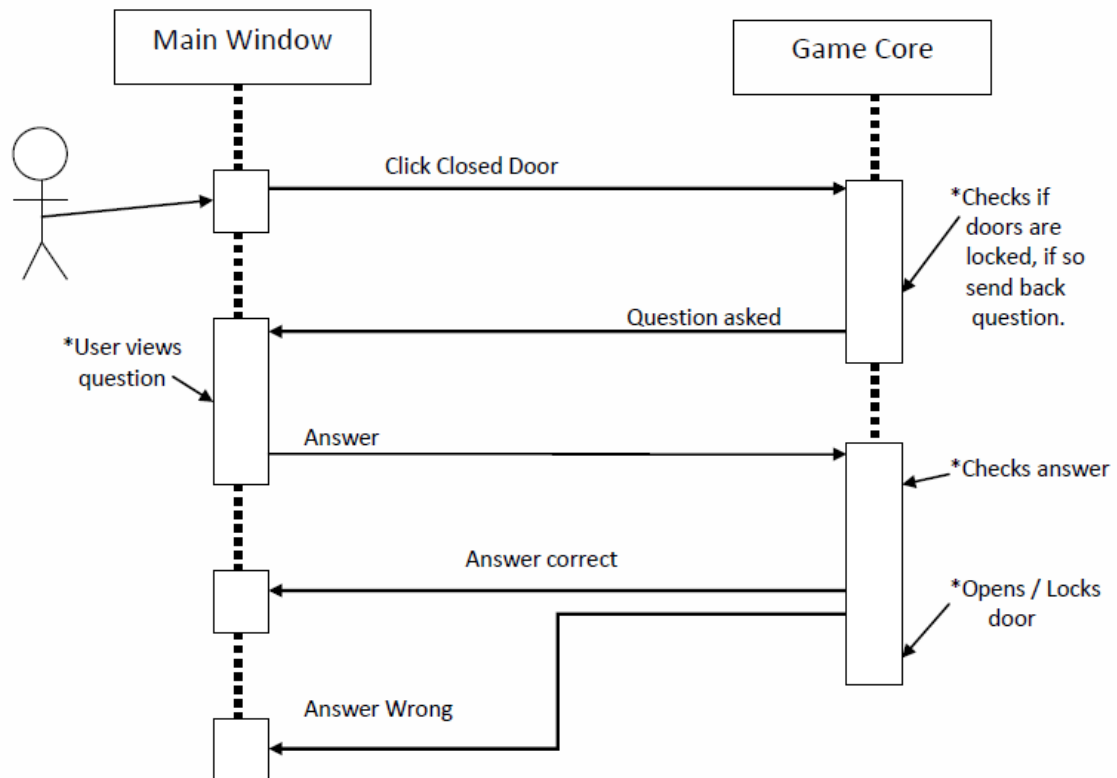


Appendix C: Use Case Diagram



Appendix D: Sequence Diagram

Sequence Diagram



Appendix E: State Diagram

STATE DIAGRAM

