Software Requirements Specification

for

Unicorn Chronicles

Version 1.0 approved

Contributors: Caroline El Jazmi, Brodi Matherly, JJ Coldiron

University of Washington

Last update: 01/05/2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this document is to describe the required features and functionality of Unicorn Chronicles – a trivia maze game (Version 1.0). This will include intended users, compatibility with various platforms, development constraints, and required software characteristics. In defining these attributes, this document will serve as a roadmap for development of the game's first iteration.

1.2 Intended Audience and Reading Suggestions

The intended audience of this document is the course's instructor and colleagues, as well as any users seeking to understand the design principles being employed in the development of the game.

Section 2 will provide a synopsis of the most important features that the software must carry out, the varying classes of users that will be interacting with the software, and the constraints affecting the development/implementation of the software. Following this synopsis, Section 3 will describe each feature, its priority level, how it responds to user actions, and the requirements it must meet to support full software functionality. Section 4 moves forward in describing which software elements require a GUI for user interaction and what tools/libraries are being employed in producing the final product. Finally, Section 5 will describe the performance and security requirements necessary to maintain both the quality of the software and a pleasant user experience.

For general users of the software, reading this document sequentially will provide you with the most logical explanation of development goals and constraints. For the instructor and colleagues, it is recommended that you allocate most of your attention to Sections 4 and 5, as these will be the most specific to our individual product and you are likely familiar with the assignment specifications and key functions that the software must carry out.

1.3 Project Scope

Unicorn Chronicles places players in a maze that can only be navigated by correctly answering the trivia questions presented to them. This game exists as part of an undergraduate computer science program, so the primary purpose of this software is to both educate the developers while still creating an enjoyable experience for players. The development process emulates that of a real industry project, employing Agile

methodologies, version control between multiple users, and other formal software engineering practices.

2. Overall Description

2.1 Product Perspective

This product is intended as an assignment first and foremost. It exists as its own entity, and no other systems depend on it. The product uses a database to store the trivia questions that will be displayed to the user, and uses Unity, Unity's UI Framework, as well as its Input System.. Figure n illustrates the system dependencies of the application.

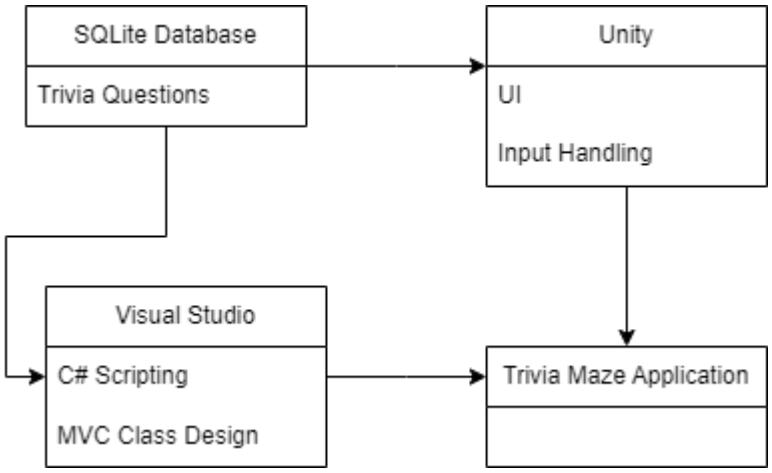
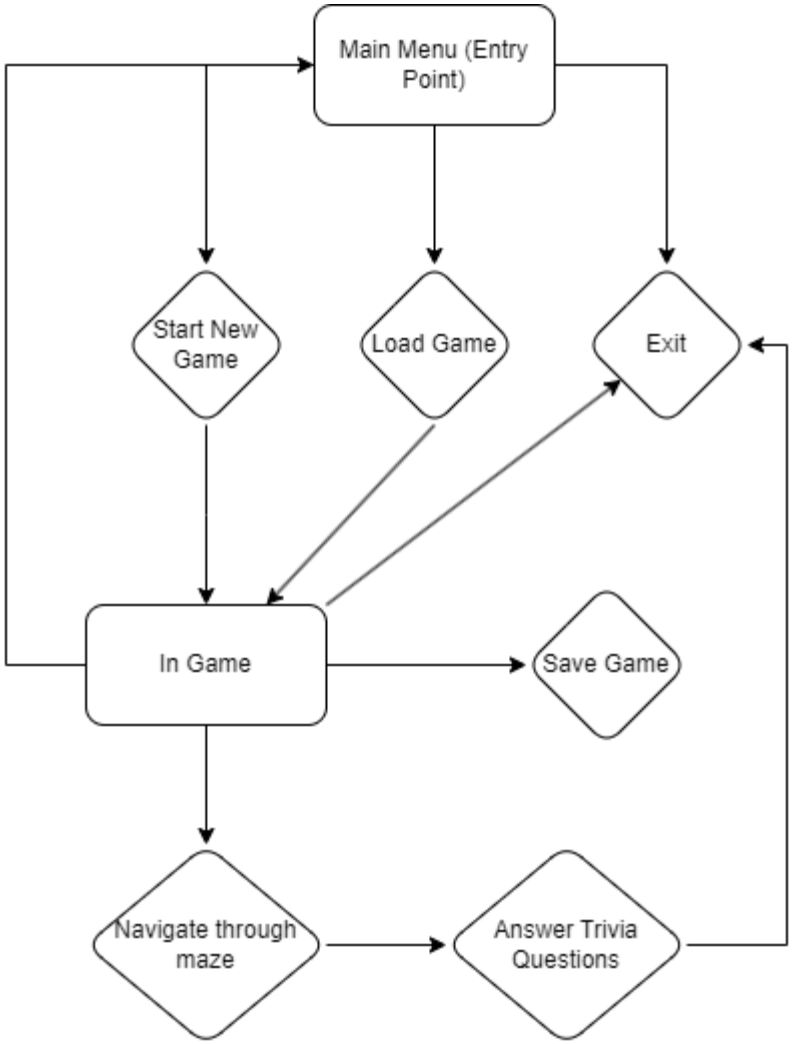


Figure n.

2.2 Product Features

This section outlines the features and significant functions that the user can perform in the application. Details of each feature will be provided in **Sec. 3** of this document. Figure n. shows the general flow of the game, starting from it's entry point, until exiting.



2.3 User Classes and Characteristics

- **Favored Users:**
  - **Professor:** Will interact with the software from an evaluative perspective to gauge developers' proficiency with course concepts. The professor will be able to operate and see the implementation of the product

  - **Colleagues:** May use the game to understand differing practices from their own/gauge their own project correctness. Colleagues will be able to run the application, and will be given access to the underlying implementation if required.

- **Other User Classes:**
  - Any other persons who wish to use the application for purely entertainment. These users do not require the technical knowledge of the above users, and will only run the game.

2.4 Operating Environment

As described in **2.7**, it is assumed that the user has a computer that they will run the application on. It is intended to operate cross-platform, including MacOS and Windows.

2.5 Design and Implementation Constraints

SQLite databases are not directly supported through Unity - an external framework must be imported and added to the project.

2.6 User Documentation

- **README:** A guide explaining the gameplay mechanics, controls, and features of the Unicorn Chronicles

2.7 Assumptions and Dependencies

This section contains any assumptions and dependencies that the application requires to work for its intended use with the intended users.

2.7.1 User Assumptions

Assumption	Reason for Assumption
User's ability to answer trivia questions in the form of multiple choice questions, true/false, and short answers	User is a CS professor or a student at a university level and has likely encountered trivia game questions before.
User's understanding of maze gameplay.	User is a CS professor or a student at a university level and has likely encountered maze game play before.
User's ability to navigate a simple user interface.	User is a CS professor or a student at a university level and has very likely had to navigate a simple user interface before.

User's ability to run the program in their chosen IDE	User is a CS professor or a student at a university level and has very likely had to run a program in an IDE on their local machine.

2.7.2 Software System Assumptions

Assumption	Reason for Assumption
Game design has enough trivia questions to avoid repeated questions during a single game.	Software system designers will assess the max possible moves in the maze and will store enough trivia questions to ensure doubles are unlikely to happen during a single playthrough.
Game design depends on external libraries and/or frameworks compatible with the Unity game engine and C# programming language, such as UI and Input handling systems.	Software system designers will be required to implement external libraries and/or frameworks in order to create a playable 3D maze trivia game.
Game design will relies on the features and capabilities provided by Unity Game Engine and C# Programming.	Software system designers have agreed on using Unity Game Engine and C# programming which will require the use of features and capabilities of those technologies.

2.7.3 Dependencies

Dependency	Reason for Dependency
SQLite Database	Requirement for storing trivia game questions to be accessed by the game.
Windows/MacOS	Requirement for running the game.
Local Machine	Requirement for running the game.
IDE	Requirement for running the game.
C#	High-level language used to create the game.

Unity	Game engine technology used to create the game.

3. System Features

3.1 System Feature 1

- **User Navigation (high priority):** Users can navigate through the maze, moving from one room to another by choosing a door to pass through.
- **UI Map (high priority):** Users can see their position within the maze at all times on the screen.
- **Question-based Doors (high priority):** Each door in a room is associated with a trivia question. To pass through a door, the user must correctly answer the question.
- **Question Types (high priority)**: The software supports multiple choice, true/false, and short answer questions, providing a varied gameplay experience.
- **Question Storage (high priority):** Trivia questions and answers are stored in a SQLite database, allowing for easy management and customization of the question bank.
- **Progress Tracking (high priority)**: The software keeps track of the user's progress, including the number of questions answered correctly, rooms visited, and locked doors.
- **Game Loss Condition (high priority)**: If the user fails to answer a question correctly and exhausts all attempts, the corresponding door is locked permanently. If the user gets trapped and cannot reach the exit, the game is lost.
- **Game State (Save, Load) (high priority)**: The software allows users to save and load their game progress, enabling them to resume the game from where they left off.

3.1.2 Stimulus/Response Sequences

3.1.2.1 Start Game

Stimulus	User launches software
Response	Software displays the main menu with options.

3.1.2.2 New Game

Stimulus	User selects “Start New Game” from the main menu options.
Response	Software initiates new games with default stats, and displays maze with trivia questions prompting the user to navigate.

3.1.2.3 Save Game

Stimulus	User selects “Save” from menu options.
Response	Software saves the current game state and notified the user the game has successfully been saved.

3.1.2.4 Load Game

Stimulus	User selects “Save” from menu options
Response	Software checks if a game has previously been saved, if yes then saved game is displayed on UI, else, software notifies users that there are no saved games available.

3.1.2.5 Exit Game

Stimulus	User selects “Exit” from menu options.
Response	Software closes the game without saving.

3.1.2.6 Help Menu Option

Stimulus	User selects “Info” from menu options
Response	Software displays relevant game instructions and other information

3.1.2.7 Navigation Door

Stimulus	User walks up to a door.
Response	A question window is instantiated and displayed on the screen.

3.1.2.8 Navigation Collectible

Stimulus	User picks up a magic key.
Response	The user's key count goes up and is able to use a key on a door to bypass answering a question.

3.1.2.9 Trivia Quiz

Stimulus	A window with a question is presented to the user.
Response	The user answers the question correctly/incorrectly and is presented with a window stating the outcome.

3.1.3 Functional Requirements

3.1.3.1 Unity User Interface

REQ-1:	The software must have a user interface with a menu system when the game is loaded.
REQ-2:	The software must display a maze when a game is active.
REQ-3:	The software should include the menu systems option "Start Game".
REQ-4:	The software must include menu system options such as "Save Game", "Load Game", "Exit", "Help".
REQ-5:	The software must be able to display the current room information when a game is active.

REQ-6:	The software must display user navigation options when a game is active.
REQ-7:	The software must display one trivia question at a time when a game is active.
REQ-8:	The software must allow only valid navigation options when a game is active.

3.1.3.2 Game Format

REQ-9:	The software must be able to create a maze of at least 4x4 rooms.
REQ-10:	The software should present one or more doors in each room.

3.1.3.3 Game Logic

REQ-9:	The software must allocate an entry point for the user at the beginning of a game.
REQ-10:	The software must allocate an exit point for the user at the beginning of a game.
REQ-11:	The software should present the user a question before passing through a door.
REQ-12:	The software should allow the user to pass through the door if they answer the trivia question correctly.
REQ-13:	The software should not allow the user to pass through the door if they answer the trivia question incorrectly.
REQ-14:	The software should lock the door if the user answers the trivia question incorrectly.

3.1.3.4 Game Win-Lose

REQ-15:	The software should notify the user that they have won if they have successfully traveled from the entry point to the exit point.
REQ-16:	The software should notify the user that they have won if they have successfully traveled from the entry point to the exit point.

3.1.3.5 Extra Features

REQ-15:	The software should contain objects that help the user bypass doors, i.e. keys.

4. External Interface Requirements

4.1 Software Interfaces

This product is being developed using Unity (Version 2022.3.4), accompanied by any necessary C# code. All C# code is being in Microsoft's Visual Studio (version 17.6), using plugins to manage the SQLite database containing all of the trivia question data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The performance requirements for the system include ensuring fast and seamless communication between the user and the user-interface, regardless of the size of the maze or the presence of multimedia elements. This means that the system should provide a smooth and responsive user experience, enabling efficient interaction even with complex mazes or multimedia content. Additionally, the software needs to efficiently load trivia questions from the SQLite database to the user interface, ensuring quick and efficient retrieval of data for a seamless trivia experience.

5.2 Security Requirements

All data stored in the SQLite database must have proper access control to prevent unauthorized access or modification of trivia questions. Within the game code itself, proper usage of access modifiers for all classes, methods, and variables is necessary to ensure that the code style guidelines are upheld and that data is not manipulated in ways not intended by the developers. All security concerns are to be addressed by developers using proper software engineering practices, as no security or privacy certifications are required to interact with the finished product.

5.3 Software Quality Attributes

Usability, fluidity, and reliability are being prioritized to maintain a pleasant and consistent user experience. The experience should be uniform for all users equipped with reasonably modern machines. To uphold this, the development process will always favor reliability and usability over complexity. Any implemented features, such as 3D graphics and sound, must uphold these principles. Going hand-in-hand with reliability, developers should produce modular, testable code to ensure that proper functionality is guaranteed across any possible edge cases. This will be verified through repeated unit testing throughout all stages of the development process.

6. Other Requirements

To add SQLite integration, this repository will be required:

https://github.com/robertohuertasm/SQLite4Unity3d

Appendix A: Glossary

Technical

Term	**Definition**
SQLite	A C-language library that implements a small, fast, self-contained, high-reliability,full-featured, SQL database engine. https://www.sqlite.org/index.html
Unity	A cross-platform game engine developed by Unity Technologies. Their products give content creators the tools to not just entertain but to create innovative RT3D experiences and deliver better processes for almost every industry. https://unity.com/

Other

Term	**Definition**
Maze	A confusing network of intercommunicating paths or passages; labyrinth. https://www.dictionary.com/browse/maze

Trivia Quiz	A trivia game or competition is one where the competitors are asked questions about interesting but unimportant facts in many subjects. https://www.collinsdictionary.com/us/dictionary/english/trivia-quiz