**Trivia Game**

**Software Requirements Specification**

**Version 2.0**

**December 17, 2021**

**Members:**

**Hana Bechara**

**Brandon Rosario**

**Julie ChanJoo Kim**

**Table of Contents**

[**1. Introductio**](#_nxqi3eqlg0o2)**n 3**

[1.1 Purpose](#_7x5exl6y0s3x) 3

[1.2 Intended Audience and Reading Suggestions](#_w9ywcy9wlb7q) 3

[1.3 Project Scope](#_qxf8pke516eg) 3

[**2. Overall Description**](#_poiz1bshdlno) **4**

[2.1 Product Perspective](#_dyo5ua1ue1wq) 4

[2.2 Product Features](#_k6dg16vm3n1) 4

[2.3 Operating Environment](#_fj9qahhy9dzh) 4

[2.4 Software Interfaces](#_dudr0g5azdy) 5

[**3. System Features**](#_46pmrzzddqg0) **6**

[3.1 Play Game](#_sroeoz8ds91) 6

[3.2 Save Game](#_1ba6n8hp1dbm) 6

[3.3 Load Game](#_chl5vixcv4ej) 7

[3.4 Game Play Instructions](#_be1wf2z2cpk1) 7

[3.5 About Us](#_oxivhpx0e8uy) 8

[3.6 Player Movement - Into PassableTerrain](#_wmq8ei8wd899) 8

[3.7 Player Movement - Into ImpassableTerrain](#_md8kq1upaz8t) 9

[3.8 Player Movement - Into Closed Door](#_hriw05u72ryj) 10

[3.9 Player Answers Trivia Questions](#_k7iixg2cvwp9) 10

[3.10 Player Opens Closed Doors](#_o5zjnvfin9jx) 11

[3.11 Player Locks Closed Doors](#_1y2lukjqhozp) 12

[3.12 Cheats / Debugging - Open / Lock Doors](#_50vfjvawns5) 12

[3.13 Win Condition](#_wzoknl30vxd) 13

[3.14 Loss Condition](#_ozd6hs31bhe2) 13

**4. Appendix 15**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| **All Members** | **10/26** | **Draft version, all sections may change** | **1.0** |
| **All Members** | **12/17** | **Final Draft** | **2.0** |
|  |  |  |  |

# **1.** **Introduction**

## **1.1** **Purpose**

The purpose of this Software Requirements Specification, SRS is to provide information detailing the implementation of the game Trivia Maze. The document will include features of the back-end development and the front end, the interface, along with information on how to run the game and the constraints under which it operates. The SRS is divided into 3 main parts which include: introduction, overall description, system features, external interface requirements, and other non-functional requirements.

## **1.2** **Intended Audience and Reading Suggestions**

The intended audience includes game developers, users(players), and testers to get a better understanding of this project. Any developers who want to revise or update (with new features) can use this document as a foundation for developing a better version. For users and testers, this document will provide detailed explanations of the logic behind the trivia maze, allowing users to understand the developer’s thoughts and processes. Readers should read it in order, as it was organized in this sequence for the reader’s purposes - convenience and simplicity.

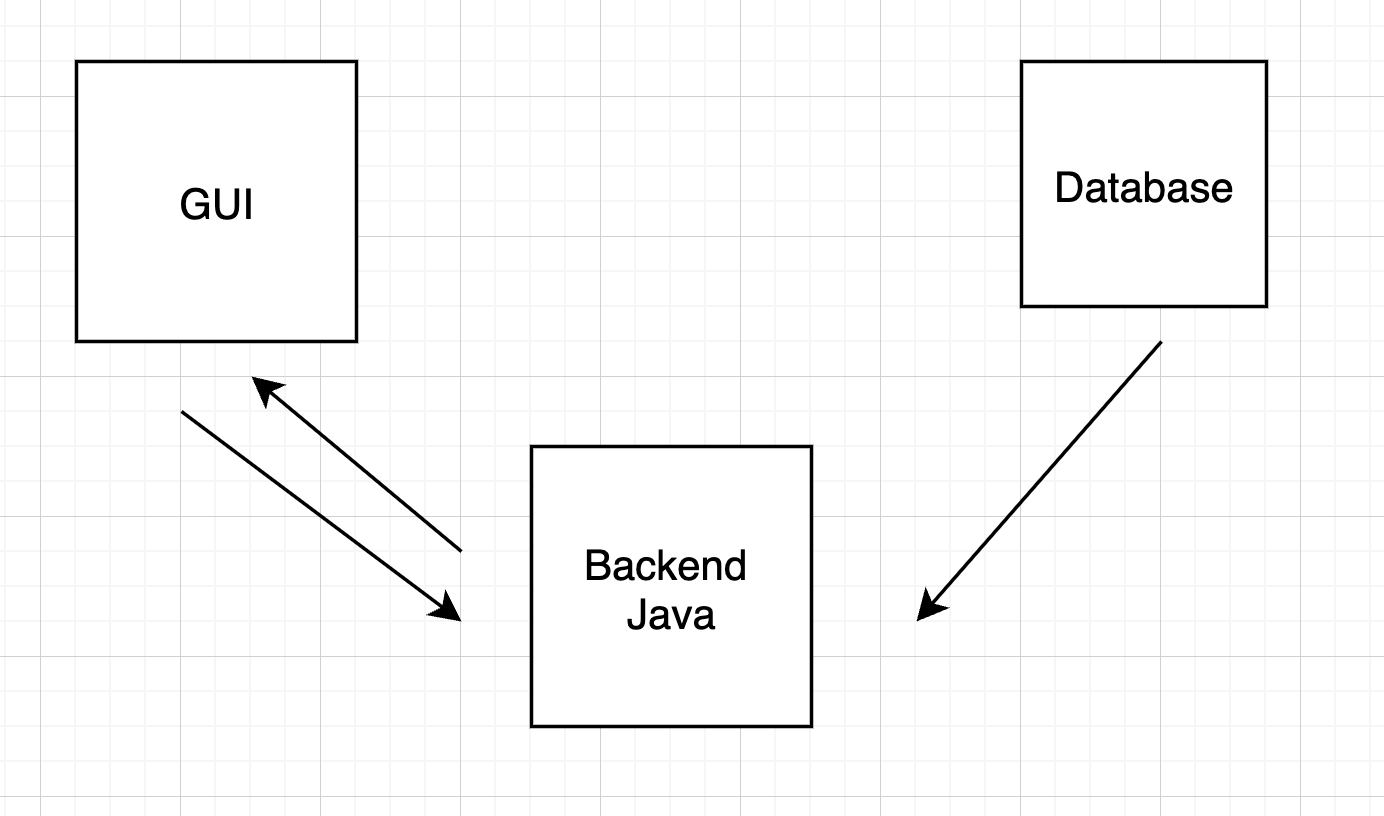
## **1.3** **Project Scope**

The game, Trivia Maze, is developed using the programming language Java, to create the back-end and front-end, and the graphical user interface(GUI). The GUI implementation will ask the user or player, questions if answered correctly the player is allowed to move through the maze, to the desired direction, if answered incorrectly the user is unable to proceed and a direction will be closed off.

# 

# **2.** **Overall Description**

## **2.1** **Product Perspective**



Trivia Maze is a new system and will be a unique implementation of the game, the user will be questioned if answered correctly the user is able to traverse the game. The interface that the user will interact with is implemented through the Java graphical user interface. The trivia questions will be stored on a database that the game will need to access.

## **2.2** **Product Features**

The main functions of Trivia Maze will include: save, load game, exit, “about us” text box, gameplay instructions, cheats, and play. The play function will begin with the user at a starting point, to begin the user needs to click on a direction to proceed by correctly answering the trivia questions they will proceed through a door. If the user does not answer the questions correctly, the door is locked permanently and the player needs to find another way to exit the maze. The save function saves the user’s records(current place, score, etc.), while the load function reloads a saved game. Exit function will terminate the program, and if the user wants to play the game again, then the user needs to run a new program.

## **2.3** **Operating Environment**

The user will be able to run the trivia maze program on all distributions of Linux as well as all versions of Windows and Mac operating systems. As long as the user has Java version 17.0.1, along with running on Intellij version Idea 2021.2 installed on their computer, and is able to open a JAR file as an executable there will be no issues running the program. On Linux distributions and Mac operating systems, configuration settings may need to be adjusted to run JAR files. No internet connection will be required. This program will not be developed for mobile devices.

## 

## **2.4** **Software Interfaces**

The tech stack of the program will consist entirely of the Java programming language and will use SQLite for database needs. The development team will use JDK version 17.0.1 for backend code, SQLite version 3.36.0 for database integration, and JavaFX version 17.0.1 for GUI implementation. The user will interact with the JavaFX GUI to perform actions within the trivia game. The GUI will be updated through the backend code, developed in Java. All questions that are given to the user as they open doors will be supplied from the SQLite database. All questions will be hardcoded into the SQLite database and the user will not have to write privileges to the database. All read privileges of the SQLLite database will be handled through the Java backend and supplied to the user through the JavaFX GUI.

Last additional step to compile file: In IntelliJ configuration in the IDE modify options in Java there is an add VM (short is alt-V) paste the follwoing: --module-path "C:\Program Files\Java\javafx-sdk-17.0.1\lib" --add-modules javafx.controls,javafx.fxml

# 

# **3.** **System Features**

## **3.1** **Play Game**

3.1.1 Description

*The user will be able to start the game via a dropdown menu accessed at the top left of the window panel.*

3.1.2 Priority

*Essential - 10/10*

3.1.3 Stimulus/Response Sequences

*The user will click the “Start New Game” button from the dropdown menu and the game state will build itself, displaying the trivia maze, and all actors and props. If a game is already in progress a prompt will be displayed to the user asking them if they would like to start a new game and lose their progress on the current game.*

3.1.4 Precondition

*None*

3.1.5 Postcondition

*A new game starts.*

3.1.6 Exception Path

*None*

## **3.2** **Save Game**

3.2.1 Description

*The user will be able to save the game state via a dropdown menu accessed at the top left of the window panel.*

3.2.2 Priority

*Optional - 5/10*

3.2.3 Stimulus/Response Sequences

*The user will click the “Save Game” button from the dropdown menu and the game state will be saved via serialization. Only one game state can be saved per machine (TBD). This means that if the user tries to save when there is already a saved game found, the user will be prompted to overwrite the previous saved game.*

3.2.4 Precondition

*3.1 - User must have initiated a new game.*

3.2.6 Postcondition

*The game state is saved and is able to be retrieved via the “Load Game” button located in the top-left dropdown menu.*

3.2.5 Exception Path

*None (Note: Would having a previous save state within memory be considered an exception?)*

## **3.3** **Load Game**

3.3.1 Description

*The user will be able to load the game state via a dropdown menu accessed at the top left of the window panel.*

3.3.2 Priority

*Optional - 5/10*

3.3.3 Stimulus/Response Sequences

*The user will click the “Load Game” button from the dropdown menu and the game state will be loaded via serialization. This means the game will build itself with all serializable actors and props relocated and adjusted according to the saved game state. Gameplay will then resume.*

3.3.4 Precondition

*3.2 - The user must have saved a game.*

3.3.5 Postcondition

*The game state is loaded and the trivia maze is rebuilt with all serializable data loaded into various actors and props. The gameplay is then resumed.*

3.3.6 Exception Path

*If there is no game state currently saved within the program on any given machine, the user will be displayed a prompt indicating that there is no game save data.*

## **3.4** **Game Play Instructions**

3.4.1 Description

*The user will be able to read detailed instructions on how to navigate and complete the trivia maze. The button to access these instructions will be located in the top toolbar.*

3.4.2 Priority

*Optional - 3/10*

3.4.3 Stimulus/Response Sequences

*The user will be able to click a button labeled “Instructions” located in the top toolbar. When this button is clicked, a popup will appear that will give detailed instructions on how to navigate and complete the trivia maze. All actors and props will be shown and described to allow the user to have knowledge of all systems and interactions within the program.*

3.4.4 Precondition

*None*

3.4.5 Postcondition

*A popup will be displayed with detailed instructions on how all actors, props, and interactions within the program function.*

3.4.6 Exception Path

*None*

## **3.5** **About Us**

3.5.1 Description

*The user will be able to read about who made this program, any relevant details about it’s creators, and the process of making the program.*

3.5.2 Priority

*Optional - 3/10*

3.5.3 Stimulus/Response Sequences

*The user will be able to click a button labeled “About Us” located in the top toolbar. When this button is clicked, a popup will appear that will give the names of the program's creators, any relevant details about the creators, and the process and procedures used to create the program.*

3.5.4 Precondition

*None*

3.5.5 Postcondition

*A popup will be displayed containing information about the program’s creators and the development of the program itself.*

3.5.6 Exception Path

*None*

## **3.6** **Player Movement - Into PassableTerrain**

3.6.1 Description

*The user will be able to move the main actor sprite up, left, down, or right if the main actor will enter passable terrain. Passable terrain is defined as: open space, open doors, and the exit flag.*

3.6.2 Priority

*Essential - 10/10*

3.6.3 Stimulus/Response Sequences

*The user can attempt to move the main actor sprite in one of the four cardinal directions by using the W / A / S / D keys or the Up / Left / Down / Right keys or using buttons displayed within the main window (TBD). When the user uses one of these actions, the main actor sprite will attempt to move in the corresponding direction. If the main actor sprite would pass through empty space, an open door, or the exit flag, the main actor sprite will successfully move in that direction and the game state will be updated.*

3.6.4 Precondition

*3.1 - The user must have initiated a new game.*

3.6.5 Postcondition

*The location of the main actor sprite will be updated corresponding to which cardinal direction the user wishes to move.*

3.6.6 Exception Path

*None*

## **3.7** **Player Movement - Into ImpassableTerrain**

3.7.1 Description

*The user will not be able to move the main actor’s sprite up, left, down, or right if the main actor will enter impassable terrain.*

3.7.2 Priority

*Essential - 9/10*

3.7.3 Stimulus/Response Sequences

*The user can attempt to move the main actor sprite in one of the four cardinal directions by using the W / A / S / D keys or the Up / Left / Down / Right keys or using buttons displayed within the main window (TBD). When the user uses one of these actions, the main actor sprite will attempt to move in the corresponding direction. If the main actor sprite would pass through a wall or a door that has been locked, due to this movement, the main actor sprite will not move and instead remain stationary. The game state will not be updated.*

3.7.4 Precondition

*3.1 - User must have initiated a new game.*

*Alt: 3.10 - User must have previously locked a door.*

3.7.5 Postcondition

*The location of the main actor sprite will remain stationary. The game state will not be updated.*

3.7.6 Exception Path

*None*

## **3.8** **Player Movement - Into Closed Door**

3.8.1 Description

*The user will be able to attempt to move the main actor sprite up, left, down, or right into a closed-door that is not locked.*

3.8.2 Priority

*Essential - 9/10*

3.8.3 Stimulus/Response Sequences

*The user can attempt to move the main actor sprite in one of the four cardinal directions by using the W / A / S / D keys or the Up / Left / Down / Right keys or using buttons displayed within the main window (TBD). When the user uses one of these actions, the main actor sprite will attempt to move in the corresponding direction. If the main actor sprite would pass through a closed door that has not been locked, the user will be prompted to answer a trivia question giving them an attempt to open said door.*

3.8.4 Precondition

*3.1 - User must have initiated a new game.*

3.8.5 Postcondition

*3.9 - A prompt displaying the question associated with the door will be displayed allowing the user to attempt to answer the question correctly.*

3.8.6 Exception Path

*None*

## **3.9** **Player Answers Trivia Questions**

3.9.1 Description

*The user will be able to answer questions associated with corresponding doors if they have not already attempted to answer said question.*

3.9.2 Priority

*Essential - 9/10*

3.9.3 Stimulus/Response Sequences

*When the user attempts to move their main actor sprite into a door that has not been previously accessed the user will be prompted to answer a trivia question that is associated with the said door. Each door will only have one question associated with them. Each question can be one of three types: short answer, multiple-choice, or true/false. The user will be able to attempt to answer the question via different methods for different question types. A text box where a string can be entered for a short answer question. A radio button selection for multiple-choice questions. A true / false button selection for true / false questions. The user may also select a button that exits the question prompt and leaves the door closed and not locked, allowing the user to select a different door if applicable (TBD).*

3.9.4 Precondition

*3.8 - User must have moved into a closed-door that is not locked.*

3.9.5 Postcondition

*The user is displayed a prompt with a question that is either: short answer, multiple-choice, or true/false.*

3.9.6 Exception Path

*None*

## **3.10** **Player Opens Closed Doors**

3.10.1 Description

*The user will be able to open closed doors if they answer the corresponding question associated with the door correctly.*

3.10.2 Priority

*Essential - 9/10*

3.10.3 Stimulus/Response Sequences

*When the user is prompted with a trivia question when attempting to move through a closed-door they can answer the trivia question correctly. When this occurs, the user will be notified that the question was answered correctly and the door sprite will change to an open door sprite. The main actor will then advance through the door and the game state will be updated.*

3.10.4 Precondition

*3.9 - The user must have been prompted to answer a trivia question and correctly answer said question.*

3.10.5 Postcondition

*3.6 - The player is notified that they have answered the trivia question correctly, the door sprite is updated to an open door sprite and the main actor sprite advances through the door to the next room. The game state is updated as such.*

3.10.6 Exception Path

*None*

## **3.11** **Player Locks Closed Doors**

3.11.1 Description

*The user will be able to lock closed doors if they answer the corresponding question associated with the door incorrectly.*

3.11.2 Priority

*Essential - 9/10*

3.11.3 Stimulus/Response Sequences

*When the user is prompted with a trivia question when attempting to move through a closed-door they can answer the trivia question incorrectly. When this occurs, the user will be notified that the question was answered incorrectly and the door sprite will change to a locked door sprite. The main actor will then remain stationary and the game state will be updated.*

3.11.4 Precondition

*3.9 - The user must have been prompted to answer a trivia question and incorrectly answered said question.*

3.11.5 Postcondition

*3.7 - The player is notified that they have answered the trivia question incorrectly, the door sprite is updated to a locked door sprite and the main actor sprite remains stationary. The game state is updated as such.*

3.11.6 Exception Path

*None*

## **3.12** **Cheats / Debugging - Open / Lock Doors**

3.12.1 Description

*The user will be able to access an option from the toolbar located at the top of the window that will allow them to open or lock doors at will.*

3.12.2 Priority

*Optional - 6/10*

3.12.3 Stimulus/Response Sequences

*The user is able to select an option in the top toolbar under the cheats dropdown labeled “Janitor’s Keys” that will allow the user to lock or unlock any door when prompted to answer a trivia question (Note: Implementation details need to be discussed. Should we make this an option permanently in the trivia question prompt to allow for key pickups after the base game is in a stable build?). This will allow for easy debugging and traversal of the trivia maze.*

3.12.4 Precondition

*3.8 - User must have attempted to enter a closed door.*

3.12.5 Postcondition

*The closed-door will become open or the door will become locked depending on what the user wishes. The door sprite will reflect this decision.*

3.12.6 Exception Path

*None*

## **3.13** **Win Condition**

3.13.1 Description

*The user will be able to “win” the game by entering the exit flag. The exit flag will be located at the opposite end of the trivia maze from where the main actor sprite is placed initially.*

3.13.2 Priority

*Essential - 10/10*

3.13.3 Stimulus/Response Sequences

*The user will be able to move the main actor sprite into the exit flag thus signaling the program that the user has completed the trivia maze. The user will then be prompted that they have “Won the game” and will be asked to either exit the program or start a new game.*

3.13.4 Precondition

*3.6 - User must have attempted to move into the exit flag.*

3.13.5 Postcondition

*The user will be prompted that they have won the game and asked to either exit the game or to start a new game.*

3.13.6 Exception Path

*None*

## **3.14** **Loss Condition**

3.14.1 Description

*The user can “lose” the game if too many doors have been locked and there is no accessible path to the exit flag*

3.14.2 Priority

*Essential - 9/10*

3.14.3 Stimulus/Response Sequences

*If the user answers too many questions incorrectly they can lock doors in a way that the exit flag may become inaccessible. If this is the case the user will be prompted that they have “lost the game” and will be asked if they would like to exit the program or if they would like to start a new game. To ensure that this condition is accounted for, every time the player answers a trivia question incorrectly and a door is locked a check will occur where a pathfinding algorithm will run within the program, making sure there is a path to the exit flag of the trivia maze.*

3.14.4 Precondition

*3.11 - User must have attempted to answer at least two trivia questions and guessed incorrectly, locking those doors and making the exit flag inaccessible.*

3.14.5 Postcondition

*The user will be prompted that they have “lost” the game and will be asked if they would like to exit the program or start a new game.*

3.14.6 Exception Path

*None*