**Software Requirements Specification**

**for**

Cybersecurity Exploration Game

**Version 1.1 approved**

**Prepared by Julian Urban & Brian Hitchcock**

**Hood College**

**1/29/2023**

**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 Product Scope 1

1.5 References 1

**2. Overall Description 2**

2.1 Product Perspective 2

2.2 Product Functions 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. External Interface Requirements 3**

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

**4. System Features 4**

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

**5. Other Nonfunctional Requirements 4**

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

**6. Other Requirements 5**

**Appendix A: Glossary 5**

**Appendix B: Analysis Models 5**

**Appendix C: To Be Determined List 6**

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
| Julian Urban | 2/2/23 | We need to mention something that allows for the game to terminate or say cleared.  Explain how the journal stuff equates into the questions into things.  If the user gets a question wrong, will we tell the player that it is wrong as well as why  Added assumptions and dependencies | 1.1 |
|  |  |  |  |

# **Introduction**

## **Purpose**

The purpose of this document is to define the requirements of a 2D exploration game that incorporates cybersecurity trivia and puzzles. The further the player gets in the game, the harder the questions will get.

## **Document Conventions**

None.

## **Intended Audience and Reading Suggestions**

This SRS document is intended for the developers of this software in order to construct and deliver the content.

## **Product Scope**

The scope of this software system is for incoming students to practice problem-solving and interactivity while having them learn about cybersecurity. It focuses more on teaching incoming Computer Science students while being a way to entertain any who plays the game.

This SRS document is aimed to inform of the details of the contents and requirements for the software to be implemented and developed. This is meant to be software that can potentially run on any Windows PC.

## **References**

None.

# **Overall Description**

## **Product Perspective**

A 2d Game that enables a player character to:

* Challenge their knowledge regarding cybersecurity by answering questions
* Learn more information regarding cybersecurity by exploring the world
* Interact with environments and dungeons to practice problem-solving and learn about cybersecurity

## **Product Functions**

* A 2D environment that the player can navigate
* A player character that the user can move around the screen using their keyboard
* The game is split into different areas
* The user navigates through the areas by walking through doors or stairs
* The user must answer a question about cybersecurity to unlock the next area
* The questions will get more difficult as the player progresses through the areas of the game
* There will be 3 dungeons for the three difficulties of questions
* The player will reach the end of the game after clearing the last dungeon

## **User Classes and Characteristics**

Users of this system should be able to enter the program and be able to interact and control the player character via keyboard inputs that will be listed in a tutorial. The tutorial will be displayed initially upon entering the game and will be accessible afterward should the user wish.

The users will be considered Players, where they will be able to create a game data save file and be able to interact with the game internally.

Players can:

* Play the game.
* Manage and Delete their save data.

## **Operating Environment**

The software will run on Windows operating systems, specifically Windows 10 and 11. We will be primarily focusing on preparing a Windows build of the game, but we will be able to create and test Mac and Linux builds if we so choose.

## **Design and Implementation Constraints**

None

## **User Documentation**

A user manual will be provided along with the software. There will also be a tutorial within the game.

## **Assumptions and Dependencies**

We are somewhat rusty on development in Unity and C# and will have to review tutorials to get started with development. We may need to spend more time figuring out how to develop certain aspects of the game. We will also need to find free assets to use for the game’s graphics.

# **External Interface Requirements**

## **User Interfaces**

Front end software: DirectX, Unity Player

Backend Software: MongoDB, C#

## **Hardware Interfaces**

Hardware Interfaces: Windows 10/11

A system that can support C#/C++

## **Software Interfaces**

We will be developing the software using Windows. We will use Unity Editor, which is a fully-featured game engine. All code for the game will be written in C#. A MongoDB database will be used to store the trivia questions and cybersecurity facts.

## **Communications Interfaces**

None.

# **System Features**

## Movement

4.2.1 Description and Priority

This feature allows the user of the system to be able to move around a playable character. With this, the character will be able to move upwards, downwards, left, and right. High priority.

4.2.2 Stimulus/Response Sequences

The user will be able to press the arrow keys to move the character within the environment. By pressing the up arrow key, you can move up. By pressing the down arrow, down, and so on and so forth.

4.2.3 Functional Requirements

REQ-1: Ensure that the keys are all bound to the player character

REQ-2: When the key is pressed, the character should be able to move in that direction within the game environment

## Interact Command

4.1.1 Description and Priority

High priority. The user should be able to interact with certain objects by pressing a button on the keyboard when they are in close proximity to an object.

4.1.2 Stimulus/Response Sequences

When the player is near an interactable object, they should be able to press a button and it will trigger some form of reaction in the system.

For example, they see a text prompt, they click on the mouse in response to the text prompt, the next one loads, and this continues until the text prompt ends.

4.1.3 Functional Requirements

REQ-1: Objects or automated sequences that require or grant the opportunity for the player to interact with.

REQ-2: A way to determine whether or not the player interacts with the situation in question

REQ-3: A response or in-game reaction to said interaction.

## Saving Progress

4.1.1 Description and Priority

Medium priority. This feature will give the user the ability to save their progress in the game into a file. This file will include information such as what stage the player is on and the area they were last in. Users can also load a progress file to resume the game from where they were previously.

4.1.2 Stimulus/Response Sequences

To save progress, the user will access a menu in the GUI and select the option to save. They will then be prompted to give the file a name and select the save location. Players can load save files from the main menu when they first start the game. They will be given a dialog box to select the file they want to load.

4.1.3 Functional Requirements

REQ-1: System to track the stage the player is on and the player’s location

REQ-2: A menu system

REQ-3: Writing progress to files

REQ-4: Loading progress from files

## Map

4.1.1 Description and Priority

Medium priority. A map that shows all of the locations in the game. Locations that have not been visited by the player will not be visible to the user. The map feature will allow the player to instantly travel to any location they have previously visited in the overworld or have cleared.

4.1.2 Stimulus/Response Sequences

The player will access the map from the menu. They will be able to see all the visited areas. The player can select a location from the list, which will take the player there.

4.1.3 Functional Requirements

REQ-1: Tracking which areas have been visited

REQ-2: Moving the player to a selected area

REQ-3: Adjusting which areas the user can view and select on the map screen

## Journal Pages

4.1.1 Description and Priority

Low priority. Documents called journal pages will be hidden in the areas of the game. These journal pages will contain helpful facts about cybersecurity that will equip the player with the knowledge they need to answer the trivia questions in the game. Players will be encouraged to find all of the journal pages in a given area to do well in the question trials. Pages that are collected will be saved in the player’s journal. The number of journal pages the player collects will be tracked.

4.1.2 Stimulus/Response Sequences

When the player locates a journal page, they will press the interact button. A prompt will appear on the screen with the designated ID number of the journal page and the cybersecurity fact it contains. The page will then be saved to the player’s journal. The player can view their journal by selecting the journal option in the menu. They can see and read the contents of the journal pages they have collected. The player can also see the total number of journal pages they have collected compared to the total number of journal pages that are in the game.

4.1.3 Functional Requirements

REQ-1: Create the journal pages table in the database

REQ-2: Add the journal pages to the levels

REQ-3: Create a system to track which journal pages are collected

REQ-4: Create the journal GUI

## Menu

4.1.1 Description and Priority

High priority. An in-game menu that will allow the user to access certain features of the game. One of the options will be a map that will allow the player to travel to any location they have previously visited. Another option is a journal where the player can view any of the hidden notes they have collected. There will also be an option to save progress and to exit to the title screen.

4.1.2 Stimulus/Response Sequences

The user will press a designated button on their keyboard to open the menu. They will then be able to choose either the map, journal, save, or exit to title. When the exit button is selected, the user will be asked if they are sure they want to exit and that any unsaved progress will be lost. If the user selects yes, the game will load the title screen. If the user selects no, the game will return to the previous menu.

4.1.3 Functional Requirements

REQ-1: Linking the menu buttons to the implemented map, journal, and save features

REQ-2: Creating a button that exits to the title screen

REQ-3: Designing the GUI of the menu

## Question Trials

4.1.1 Description and Priority

High priority. There will be points within the game where users will have to answer a series of questions in order to proceed. These questions are meant to serve as gateways within the game in order to determine user skills and proficiency in how they were learning so far. These will typically be placed within areas that are blocked until they are able to answer the series of questions properly.

If the user gets a question right, they will get a prompt to click to load the next question. When the user gets a certain number of answers correct, they receive a “passing grade” and can proceed to the next area. If the user gets a question wrong, a prompt will appear that will explain the right answer. The user can then click a button to go to the next question. When the user gets a certain number of answers incorrect, they will receive a “failing grade” and will be forced to exit the area they tried to enter.

4.1.2 Stimulus/Response Sequences

The user comes across one of these question trials, be it in an area where they enter via movement, or by interacting with certain areas within the world.

The user can then interact with the questions given and give an appropriate answer by clicking one of the options they can navigate with either the arrow keys or the click of the mouse.

4.1.3 Functional Requirements

REQ-1: The creation of questions pertaining to Cybersecurity

REQ-2: A location to store both the questions and answers to the following questions.

REQ-3: The separation of these questions into groups/pools of varying difficulty levels

REQ-4: A prompt or trigger that begins the question trials

REQ-5: Text boxes, that display the question

REQ-6: An interactive text box that features the potential answers to the questions

REQ-7: The ability for the player to select an answer.

REQ-8: A system in place to check if the answer is correct or incorrect.

REQ-9: A system to check the amount of (in)correct answers.

REQ-10: A system that selects a random sample of questions within the question pool for the player to be quizzed on.

# **Other Nonfunctional Requirements**

## **Performance Requirements**

We expect that the game should function in a responsive manner with no lag or input delay.

## **Safety Requirements**

None

## **Security Requirements**

None

## **Software Quality Attributes**

We want the software to be light on system resources so that it can run on any modern computer. We also want to make sure that all of the cybersecurity information is accurate and will be using reliable sources to collect our information.

## **Business Rules**

The game is designed to be played in a linear format. The user should only be able to view and travel to areas they have already discovered when using the map. The user should only be able to see the journal pages they have collected and nothing else. The player should only be able to access areas in a chronological order. The difficulty of the questions will match how far along the player is in the game.

# **Other Requirements**

We intend to have a database for this program within MongoDB.

**Appendix A: Glossary**

2D: 2-dimensional

Player/User: the person interacting with the software

Player character: the object the user controls to interact with the software

Dungeon: an area that the user can explore featuring puzzles and question trials

Tutorial: a section of a game that is designed to teach the user how to play

Save data: a file that contains the user’s progress in the game

Title screen - the first menu the user of the game sees

**Appendix B: Analysis Models**

None

**Appendix C: To Be Determined List**

None