

Requirements

Cohort 1 Group 2

Yusuf Almahmeed
Jane Boag
Charmaine Chamapiwa
Charlie Coyne
Anupam Cunden
Bertie Kerry
Leo Xu-Latti

This page does not count toward the total page number

Introduction

Initially, we were provided with a product brief by our customer. This detailed basic requirements with respect to the genre of the game, and certain gameplay elements. From this, the team brainstormed additional questions to clarify information that was in the brief, and gain additional information through the answers to our questions. These questions were generated through analysis of similar games and through group discussions on potential ideas and features for the game. They were then shared to our customer allowing us to elicit further information to fully understand the requirements.

The negotiation process for the requirements was carried out as a team during a group meeting. Although there were no major conflicts, some ideas still clashed. This stemmed from differences in interpretations of the brief, for example, regarding the setting of the game, whether that needed to be indoors or outdoors or how similarly it needed to represent university life. This was resolved by group discussions where members shared their thought processes and came to a final decision that aligned with the product brief but also considered accessibility and time constraints. This made sure all requirements were achievable within the project scope and consistent with the escape from university theme.

The requirements are structured in a tabular format. Initially we listed these requirements as a simple, indented list. However, we decided that the table format made it easier and clearer to read and had more space for detailed explanations for each requirement. The table format also allowed each requirement to be separated distinctly, which easily supported a requirement referencing system to be called upon in other documents.

The requirements were split into user and system requirements with a separate table for each to separate the different aspects of the game, what the player should experience and what the system must do to deliver these aspects. Each requirement was systematically numbered, e.g. U1 for user requirements and S1 for system requirements.

The format of the requirements was informed through research into standards for software specification and presentation. These highlighted a clear numbering system, structured tables and consistent statements which we have incorporated through the requirement reference header in our table format and use of 'should' and 'must' statements throughout. All sites used to research proper formatting are cited at the bottom of the document ("Functional and Non Functional Requirements") (*Guide for Developing System Requirements Specifications*).

User Requirements

Requirement Reference	Requirement	Explanation
U1	There should be a single, pre designed map for the user to play on.	The game will feature one map which the user can play, navigate and attempt to escape from.
U2	The game should have events that affect and interact with the player.	There needs to be events to keep the user engaged during gameplay.
U2.1	There must be 5 visible events that hinder the player.	Hindering events gives the user a light challenge, increasing the fun factor and the depth of the gameplay.
U2.2	There must be 3 visible events that benefit the player.	Positive events provide rewards and progression, giving users the opportunity to recover from negative events and maintain the motivation to complete the game.
U2.3	There must be 3 hidden events.	There should be events unknown to the user, to encourage them to fully explore the map, thoroughly engage with the game beyond obvious elements and increase enjoyability.
U3	The user must be able to pause and resume the game at any time.	This allows the user to take breaks during gameplay, if necessary, without losing progress.
U3.1	When paused, the user should see the game freeze and the timer stop.	This ensures that players understand when the game is paused and prevents time and score loss during this time.
U4	The user should earn a score based upon their performance.	Points are based upon time, events, and performance showing users how well they have played.
U4.1	Escaping faster results in a higher score.	The time which the user escapes directly impacts the score they receive when they beat the game.
U4.2	User interactions with events will increase/decrease the score, depending on the type of event.	Negative events that slow the user down may increase their points, while others that speed you up may reduce points
U5	The game should take place	Helps to resonate with the target

	in a university setting and represent university life.	demographic.
U5.1	The game environment should take place mainly indoors.	Escaping from a building gives the user a clear objective.
U6	The game should feature clear and intuitive user interface elements on screen.	A simple UI helps the user and more other casual players to better understand the game progress and options.
U6.1	The game should have a main menu that allows the player to start the game from.	This provides a clear starting point for gameplay.
U6.2	There should be a timer visible on-screen during gameplay.	A visible timer allows the user to know how much time they have left to complete the game.
U6.3	A pause menu should appear on screen when the game is paused.	This allows the user to pick whether to resume or quit the game while paused.
U6.4	A 'game over' screen should appear when the player wins or loses.	This screen will display the user's score and allow them to return to the main menu.
U7	The game should use a standard, keyboard control input.	Ensures familiarity and accessibility for users, making the game more straightforward.
U8	The game should be able to be completed by a casual player, of any level of experience, within 5 minutes.	This keeps the difficulty level accessible for all users and ensures that they can complete the game easily, making it hard to fail at escaping.
U9	The game should have background music.	This enhances the user's experience throughout the game by creating a more engaging feel to gameplay and increasing enjoyability.

System Requirements

Requirement Reference	Requirement	Explanation
S1	The system must have a fixed map layout with no randomisation.	Preventing randomisation helps make the game more straightforward for a casual user to understand and complete.
S2	The system must handle all game events, both visible and invisible.	This ensures the game correctly handles interactions between the user and various events.
S3	The system should be designed for easy extendability.	Other developers can extend and modify the game without changing the core logic and functionalities.
S3.1	The system should allow for more levels to be added in the future.	Other developers can introduce new mazes without changing the existing game code to make it more challenging and increasing gameplay time.
S3.2	The system should allow for more events to be added in the future.	This allows for the game to be easily expanded by future developers to make gameplay longer and more engaging.
S4	The system should support a score system that reflects player performance.	The score must change based on completion time, user interactions with events and powerups.
S4.1	The system must maintain a frame rate of 60 frames per second on lab computers.	A good benchmark to ensure smooth gameplay to meet expected quality standards.
S4.2	The system must prevent assets stretching or distorting across different display sizes.	This helps to maintain visual quality regardless of the display the user is playing on.
S4.3	The system should support full screen game display.	This ensures that the game will fill the entire display area, regardless of the OS or screen resolution.
S4.4	The system must register and respond to user input within a single frame.	Immediate responsiveness makes the game feel more fluid and smooth, enhancing the user's experience.

Works Cited

"Functional and Non Functional Requirements." *GeeksforGeeks*, 18 October 2025, <https://www.geeksforgeeks.org/software-engineering/functional-vs-non-functional-requirements/>. Accessed 23 October 2025.

Guide for Developing System Requirements Specifications. 2021. *IEEE explore*, IEEE explore, <https://ieeexplore.ieee.org/document/741940>.