

REQUIREMENTS

Cohort 1 group 11

Freddie Aberdeen
Mutaz Ghandour
James Given
Jasper Owen
Jungwan Park
Joe Reece
Ivan Shestakov

Introduction

The requirements for this project were developed through a structured requirements engineering process, consisting of requirements elicitation and specification, and verification and validation. The process was scaled appropriately for the scope and nature of this project, with a focus on clearly identifying what the game should do, how it should perform, and any constraints that would shape its development.

Requirements Elicitation and Negotiation:

The primary method of requirements elicitation was a semi-structured interview with the customer. This interview allowed the team to gather information about the key requirements for the game that needed to be clarified before development could proceed, for example, information about system requirements and core gameplay mechanics.

From this session, the customer provided a clear Single Statement of Need (SSON): “A short, single-player game in a university setting, for young adults to have an immersive, escaping-uni experience”. This initial interview served as the main source of requirements. Beyond that, the elicitation process continued informally, with only occasional follow-up questions asked for clarification. These follow-ups typically occurred during milestone reviews, or when the development team needed confirmation on design decisions.

Requirements Presentation:

Following elicitation, the team documented the requirements into a structured format. This process began by outlining user requirements, which described what the player should experience and be able to do in the game. User requirements were given a priority either shall, should or may. With shall being requirements that are necessary and will be implemented, should being requirements we would prefer to implement but are not necessary and, may being requirements we are considering implementing. User requirements were then translated into more detailed system requirements, which described what the game system must implement. The system requirements were categorised into functional requirements and non-functional requirements. Due to the game’s relatively limited scope, the specification process was largely completed early in the project, with only minor revisions being made later.

Verification and Validation:

Once the requirements were specified, the team carried out verification and validation to ensure they were complete, consistent, and aligned with the original vision. As part of the verification process, we compared the specified requirements against the ISO/IEC 25010 [1] software quality model to ensure they align with recognised quality characteristics for software systems. This comparison focused on assessing whether the requirements addressed the nine quality characteristics outlined in the product quality model. For validation, the team conducted internal reviews of the brief and interview notes to confirm that the documented requirements accurately represented the customer’s intended user experience.

User Requirements

ID	Description	Priority
UR_AUDIENCE	The game will be aimed at young adults.	Shall
UR_UX	The user shall be able to understand how to navigate the menu and play the game within 2 minutes of starting the game	Shall
UR_TIME	The game should last approximately 5 minutes.	Shall
UR_RESULTS	The game should show a score at the end of each run.	Shall
UR_TUTORIAL	The user may have the option to complete a tutorial at the start of the game that tells them how to play the game and achieve a high score.	May
UR_CHARACTER_SELECTION	The game may have an option to customise the playable character.	May
UR_UNIVERSITY	The game shall have a university setting, and provide an escape-from-uni experience.	Shall
UR_THEME	All art shall be consistent with a main theme and art style.	Shall
UR_DIFFICULTY	The user should be able to change a difficulty setting to have a new game experience.	Should
UR_MUSIC	There may be music that plays while the user plays the game.	May
UR_ACCESSIBILITY	The game shall be as widely accessible as possible for a small project.	Shall
UR_HARDWARE	The game shall be playable on any regular PC and fit on any monitor	Shall
UR_RELIABILITY	The game shall not crash when run on an average computer	Shall
UR_COUNTER	The user shall be able to see how many of each event they have interacted with	Shall
UR_EVENTS	The user shall be able to encounter at least 3 events that help them, at least 5 events that hinder them, and at least 3 hidden events.	Shall
UR_PAUSE	The game should allow the player to pause and resume the game at any time.	Shall
UR_END	The game will end when the player reaches the exit or the 5-minute timer expires, and displays a result screen showing score.	Shall
UR_LEADERBOARD	The game shall include a leaderboard with the top 5 scores achieved on the computer and the usernames of the people who achieved the scores	Shall
UR_ACHIEVEMENTS	The game shall include achievements that the user can work towards	Shall
UR_ACHIEVEMENT_SCORES	Accomplishing an achievement may affect the user's score	May
UR_ENJOYABILITY	The User shall find the game enjoyable to play	Shall

System Requirements - Functional Requirements

ID	Description	User Requirements
FR_DESKTOP_PLATFORM	The system shall run smoothly on standard desktop systems without requiring high-end hardware.	UR_HARDWARE
FR_MAP_STRUCTURE	The system shall provide a maze-like map representing a university environment with clear boundaries and defined escape paths.	UR_UNIVERSITY
FR_PLAYER_MOVEMENT	The system shall allow the player to move within the maze using keyboard or mouse controls.	UR_UX
FR_PAUSE_RESUME	The system shall suspend the game timer when paused, and resume it when unpaused.	UR_PAUSE
FR_VISIBLE_HINDRANCES	The system should include at least five visible hindrance events that obstruct the player's progress.	UR_EVENTS
FR_VISIBLE_BENEFITS	The system should include at least three visible beneficial events that enhance the player's performance.	UR_EVENTS
FR_HIDDEN_EVENTS	The system shall include at least three hidden events that are invisible until triggered, producing interactive or humorous effects.	UR_EVENTS
FR_EVENT_INTERACTION	The system shall allow the player to interact with events, resolving obstacles or collecting items as needed.	UR_EVENTS
FR_SCORING_SYSTEM	The system shall calculate and display a final score that is greater when the user takes less time to complete the game. The score can also be affected by the events.	UR_RESULTS
FR_TIMER_LIMIT	The system shall run each game session for a maximum of 5 minutes.	UR_TIME
FR_EXIT_CONDITION	The system shall end the game when the player reaches the exit or when the timer expires, displaying a results screen.	UR_END
FR_TUTORIAL_MENU	The system shall include a tutorial or instruction section telling the player the objective of the game, how to use the controls and how to achieve a high score.	UR_TUTORIAL UR_ACCESSIBILITY
FR_MAIN_MENU	The system should include a main menu with options such as Start Game, Tutorial, and Settings. Selecting an option should make it appear on screen.	UR_TUTORIAL, UR_UX
FR_USER_INTERFACE	The system shall display key information during gameplay, including timer, score, and status messages.	UR_RESULTS

FR_COUNTER	The game shall keep track of how many of each event (positive, negative, hidden) have been interacted with	UR_COUNTER
FR_LEADERBOARD	When the player reaches a score higher than another user's score on the leaderboard, they shall take the other user's place on the leaderboard and the other user shall move down a place on the leaderboard	UR_LEADERBOARD
FR_ACHIEVEMENTS	When a user meets the requirements of the achievement, it shall be listed as accomplished, with any relevant adjustments benign made to the user's score	UR_ACHEIVMENT, UR_ACHIEVMENT_SCORE
FR_HITBOXES	The sprites collision/pickup hitbox should cover the whole sprite making items intuitive and easy/simple to pick up there should be no point where you expect you should be picking up an item and yet you are not	UR_ENJOYABILITY
FR_NOTIFY	The game should notify you when an event happens that could seem unclear to the user as to why it happened	

System Requirements - Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_PORTABILITY	The game shall have cross-platform desktop support and not require any high-performance hardware	UR_HARDWARE	Game passes a standard playtest on Windows, macOS, and Linux with no manual configuration changes
NFR_ACCESSIBILITY	Most players shall be able to understand and use the game's UI without assistance.	UR_UX	At least 90% of Players shall understand how to use the game's UI without assistance.
NFR_USABILITY	The game will provide a tutorial such that someone inexperienced with games will be able to play.	UR_TUTORIAL UR_ACCESSIBILITY	90% of first-time players will be able to complete the tutorial without assistance.
NFR_Maintainability	Codebase will be modular and well documented in order to allow for future extensions and development	N/A	Game logic and systems are separated into independent modules, with each major module including docstring explaining purpose and usage.
NFR_RELIABILITY	The game will not crash upon being given reasonable input, as well as in edge-cases	UR_RELIABILITY	Playtest with 10 sessions shows no crashes or freezes under normal or extreme inputs.
NFR_CONTENT_CONSTRAINTS	The game will be family-friendly, and not feature any events or themes that may be considered harmful or inappropriate.	UR_AUDIENCE	Internal review upon every addition confirms narrative and visual cohesion.
NFR_THEME_CONSISTENCY	A consistent narrative and thematic experience shall be maintained across all elements of the game	UR_THEME	All game assets and events must support the aesthetic direction, with no conflicting

	(visuals, sounds, events, etc.)		elements. Internal review upon every addition to confirm narrative cohesion.
NFR_LEGALITY	The game shall only include third-party assets with appropriate licenses.	N/A	All third-party assets must be documented and reviewed before implementation to ensure that they have the appropriate licences.
NFR_PERFORMANCE	The game shall maintain at least 30 FPS under normal gameplay conditions	UR_HARDWARE	Performance test confirm stable framerate on standard desktops
NFR_SCALABILITY	The system should allow for future expansion (e.g. adding new maps or difficulty)	N/A	Modular code structure allows addition of at least 5 new event types without affecting existing functionality
NFR_ERROR_HANDLING	The game shall handle unexpected input gracefully without crashing	UR_RELIABILITY	Test edge cases (such as invalid user input) and verify the game does not terminate.
NFR_COUNTER	The game's event counter shall update promptly	UR_COUNTER	The event counter will be incremented less than 1 second after the event has occurred.
NFR_LEADERBOARD	The leaderboard shall be regularly checked to see if the user has earned a place on it.	UR_LEADERBOARD	Users are added to the leaderboard less than 1 second after getting a place on it.
NFR_ACHEIVEMENTS	It shall be regularly checked if the user has completed an achievement or not.	UR_ACHIEVEMENTS	Users are registered as having completed the achievement less than one second after they achieved it.
NFR_IMMERSION	The game shall make the player feel like they are in a university environment	UR_THEME, UR_UNIVERSITY	At least 90% of players shall feel like they are in a university environment while playing the game
NFR_ENJOYABILITY	The game shall be fun to play	UR_ENJOYABILITY	At least 90% of players shall enjoy playing the game.

References

[1] ISO25000, “ISO/IEC 25010,” iso25000.com, 2022.

<https://iso25000.com/index.php/en/iso-25000-standards/iso-25010>

(Last accessed 30th November 2025)