Requirements

Cohort 1, Group 6 - M6

Members:

Mir Baydemir
Adam Fraulo
Azib Hj Abu Akmar
Agata Pittarello
Esther Scanlon
Jazz Stubbins
Sonia Szetela

The client's SSON served as the foundation for the team's requirements analysis.
Certain game properties described in the brief were specific and already allowed for an initial requirement plan to be developed. Some aspects of the brief were ambiguous, such as the physical attributes of the maze, events that hinder/boost the player, the theme and overall art style, game difficulty, the target market and accessibility expectations, etc.
This warranted further communication with the client regarding the brief and any specific user and system design preferences they may have.
Thus the team organised a series of questions to be delivered during an interview with the client, following the process of elicitation, analysis, and negotiation described by Kolovos (2025) [2]. These questions were divided into the following categories: resource limitations, project schedule and timeframe, risk management, the target audience and participants, modes of operation,design and creativity limitations and constraints requirements.
Clarification regarding the use of 3rd party assets and artificial intelligence generation/tools was also discussed.
These requirements were then presented in a hierarchical structure of user, functional, and non-functional system, and constraints requirements in the form of a referencing system, according to the IEEE 29148-2018 guidelines [1] to ensure clarity, traceability, and testability.
This then allowed the team to allocate tasks to address each requirement and establish a more concrete plan for the project moving forward into the design phases.
Some examples of how the requirements were derived are:

- UR_ACCESSIBILITY, UR_THEME

  It was confirmed that the game must be made suitable for a target audience consisting primarily of students and adults with no explicit or violent themes. The game must maintain a family-friendly and humorous tone and be accessible to new and casual players alike.

  The client also emphasised that any visual cues and text boxes must be simple, clear and understandable to minimise player confusion or frustration. However the client did not extend this requirement to any colourblind or hearing impaired accessibility features. Likewise the game should only support local regional languages (English).
- FR_MAP:

  From the brief and client feedback, the team discerned that the university campus itself would function as the physical maze walls and boundaries. Due to the timeframe limitations outlined by the client we decided to hardcode the campus map with no procedurally generated structures. The map and its boundaries therefore will be visible to the player at all times in which the player avatar will navigate from a top-down 2D perspective opposed to an isometric one.
- NFR_AESTHETIC

  Due to the client's flexibility over the theme and overall art style, we decided on a cartoon-style fantasy theme for the campus maze and characters, along with elements of magic and light humor. The fantasy theme will allow for increased freedom of level design such as implementing the three event types, without increasing complexity thanks to elements like magic, teleportation and secret passages, with the art style and texture assets retaining player interest.

Finally, in order to test key requirements and prepare for anticipated user behaviour, a set of textual use cases are listed on the project website

(https://keybordkat.github.io/websiteGroup6/Use%20cases.pdf), paired with use case diagrams.

USER REQUIREMENTS

ID	DESCRIPTION	PRIORITY
UR_THEME	The game's tone should be family friendly with light humour without adult or overly violent content and should evoke university-like scenery, objects and events themes.	should
UR_ACCESSIBILITY	The game should be accessible by individuals with varying levels of gaming experience, by providing simple, clear and understandable visual cues, text boxes and controls.	should
UR_VICTORY_CONDITION	The game victory condition is escaping the maze within 5 minutes and failing otherwise.	shall
UR_SCORE	Once completed the game a score will be shown based on the in-game actions performed and the time elapsed.	shall
UR_VISIBILITY	The player should be able to view the whole map and its boundaries at all times.	shall
UR_EVENTS	The game shall include at least: 5 visible negative events, 3 visible positive events and 3 hidden events.	shall
UR_ACHIEVEMENTS	Through the game the player may achieve in game objectives that might alter the score.	shall
UR_LEADERBOARD	A leaderboard with the name and the top 5 scores will be shown to the player once completed the game.	shall
UR_TIMER	A five minute countdown timer should be always visible to the player.	should
UR_PAUSE	The game can be paused at any given time during play and the timer must halt during the duration of the pause.	shall
UR_TRACKER	The game should track and display the number of events the player interacts with or triggers.	should
UR_INSTRUCTIONS	Clear and simple instructions should be given to the player to learn how to play the game.	should
UR_GAMEPLAY	The game should provide a consistent gameplay handling invalid inputs, having a consistent performance and resetting after the game victory conditions have been reached/not reached.	should
UR_TECHNICAL	The architecture of the game, a standalone java based application on PC, should be structured in a way that it can be readily maintained and expanded in the future.	shall

FUNCTIONAL SYSTEM REQUIREMENTS

ID	DESCRIPTION	USER REQUIREMENT
FR_EVENT	The game will detect and trigger events based on players interaction/collisions accordingly.	**UR_EVENTS**
FR_NEGATIVE_EVENTS	The game shall include negative events that hinder the player's progression and may modify the score.	**UR_EVENTS**
FR_POSITIVE_EVENTS	The game shall include positive events that facilitate the player's progression and may modify the score.	**UR_EVENTS**
FR_HIDDEN_EVENTS	The game shall include hidden events that remain invisible to the player until triggered and that may either facilitate, hinder or have no effect on their progression. They may modify the score.	**UR_EVENTS**
FR_EVENTS_LOCATION	Each event is not necessarily bound to a specific location in the game map. Events can move in the map or be randomly placed at every new iteration of the game.	**UR_EVENTS**
FR_TIMER	The game shall implement a timer to track the players play and escape time throughout the game.	**UR_TIMER**
FR_MAP	The map will be a hardcoded 2D maze and visible at all times with clear boundaries.	**UR_VISIBILITY**
FR_MOVEMENT	The game shall allow the player avatar to navigate the map using standard directional keyboard inputs.	**UR_ACCESSIBILITY**
FR_SCORE_CALC	The system shall calculate the score based on the player's escape time, events and achievements completed. Only if the victory condition is reached the score will be saved and displayed in the ending screen.	**UR_SCORE**
FR_INSTRUCTIONS	The game objective and commands should be presented to the player at the start of the game and in the pause menu.	**UR_INSTRUCTIONS**
FR_PAUSE_MENU	The game should have a pop-up menu when the player pauses the game that contains some basic instructions, volume setup and a resume button.	**UR_PAUSE**
FR_START_SCREEN	There will be an initial screen containing the instructions, game objective and a PLAY button that will start the game once triggered.	**UR_INSTRUCTIONS**
FR_ENDING_SCREEN	Either the victory condition or the timer expiry will trigger a screen informing the player of the outcome and only if won the other relevant information: as score, leaderboard, achievements and events tracker.	**UR_VICTORY_CONDITION, UR_SCORE, UR_LEADERBOARD, UR_ACHIEVEMENTS, UR_TRACKER**

NON-FUNCTIONAL SYSTEM REQUIREMENTS

ID	DESCRIPTION	USER REQUIREMENT	FIT CRITERIA
NFR_USABILITY	The game shall be simple and intuitive with limited instruction.	**UR_ACCESSIBILITY**	Any visual cues and instructions must be simple, clear and understandable to minimise player confusion or frustration.
NFR_STANDALONE	The game has to be a standalone application on PC with all resources and data to be managed locally.	**UR_TECHNICAL**	The game is to be marketed as a standalone application on PC with all resources and data to be managed locally rather than rely on online systems ensuring reliable offline play.
NFR_RELIABILITY	The game shall remain stable during and after play.	**UR_GAMEPLAY**	The system should handle all expected scenarios without crashing or having significant performance issues.
NFR_RESILIENCE	The game should handle invalid inputs or missing objects appropriately.	**UR_GAMEPLAY**	Should a player try to interact with an area/object/npc without meeting the expected criteria (missing item, event not triggered yet etc) the game will not crash.
NFR_UI	The map and user interface should be clear and accessible.	**UR_VISIBILITY, UR_ACCESSIBILITY**	The map and maze design should be clear and intuitive, with text prompts having >= 12pt fonts with bright and contrasting colours.
NFR_AESTHETIC	The game shall have a fantasy university theme with magical elements and assets and be family friendly.	**UR_THEME**	The map, maze and NPC's shall all fit within the fantasy theme with magic effects, colour pallets, tiles and pixel art all cohesive to the theme.
NFR_MAINTAINABILITY	The game’s code should be easy to understand, modify and maintain.	**UR_TECHNICAL**	The game’s code should be structured clearly and modularly making it possible for other developers to understand, modify and maintain the system.
NFR_EXTENDABILITY	The game should be designed to support expansions.	**UR_TECHNICAL**	The game’s architecture should support expanding the map (based on Tiled) or adding events and achievements with only minimal changes to existing code.
NFR_DOCUMENTATION	The game’s code and project should have ample documentation to support them.	**UR_TECHNICAL**	The game’s code shall include appropriate comments and the project’s documentation shall be detailed enough to support future maintenance or expansions.
NFR_DEPLOYABILITY	The game shall be available on all standard PC operating systems.	**UR_TECHNICAL**	Available on Mac, Windows and Linux OS

REQUIREMENTS CONSTRAINTS

ID	DESCRIPTION	FIT CRITERIA
CR_ENGINE	The game must be built in a Java based engine for PC only. No support for other devices.	Runs successfully on windows/linuxPC hardware without additional dependencies using Java runtime.
CR_RESOLUTION	The game shall be a fixed screen resolution of 1920 x 1080 with no custom screen scaling.	Full map visible and clear on a standard 1920 x 1080 resolution for 16:9 average monitor size.
CR_FORBIDDEN	The game shall make clear what the player can and cannot interact with.	The API should be accessible to developers only with no access available to the player. The player should not be able to pass through walls or into areas not intended for the player.
CR_RATING	Must be accessible to by the majority of English speaking players, individuals of all ages and with varying levels of gaming experience and be family friendly	The game difficulty should not be overly complex or competitive and not feature adult themes or violence.
CR_LANGUAGE	Only supports the English language.	The game supports only the English language.

References

[1] IEEE (2018) ISO/IEC/IEEE 29148:2018 – Systems and software engineering – Life cycle processes – Requirements engineering. EEE Standards Association. Available at: https://standards.ieee.org/standard/29148-2018.html (Accessed: October 2025).

[2] Kolovos, D. (2025) Requirements Engineering [video recording].
Engineering 1 module,
University of York Virtual Learning Environment (VLE), accessed October 2025.