Requirements

Cohort 1, Team 3

Ari Kikezos

Arya Enkhnasan

Ben Green

Calum Wright

Lilac Graham

Skylar Garrett

In order to establish a strategy for planning and managing the project, the product brief served as the foundation for the team's requirements analysis.
Certain project requirements described in the brief were specific and allowed for an initial plan to be developed: conceptualising a 2d maze game to be built in a java based engine, with a 5 minute timer win condition and featuring three different types of events. However it became clear that 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).
These questions were divided into the following categories: resource limitations, project schedule and timeframe, risk management, the target audience and participants, modes of operation and constraints requirements. Examples of these updated requirements and their purposes will be presented throughout this document and supporting documentation found on the project website.

During interview preparation the team discovered the need for additional questions to clarify specific design and creativity limitations and requirement constraints. Particular focus was placed on difficulty implementation and complexity, as well as accessibility such as language support, target audience and competitiveness, since these were either loosely defined or missing from the product brief.
Clarification regarding the use of 3rd party assets and artificial intelligence generation/tools was also discussed, concluding that any AI generated content would only be acceptable in non-code or documentation areas such as art, audio, or textures, provided strict legal adherence to intellectual property and correct referencing was met.
These requirements were then presented in a hierarchical structure of user, functional, and non-functional system requirements in the form of a referencing system, according to the IEEE 29148-2018 guidelines 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.

User Requirements

ID	Description	Priority
UR_Accessibility	The game should be accessible and appropriate for the target audience of students.	Shall
UR_User_Interface	Any visual pop ups, cues or text boxes must be clear and understandable.	Should
UR_Language	The game must be in English	Shall
UR_Events	Negative events must be manageable and without excessive punishment. Positive events should offer a boon without trivialising the game.	Shall
UR_Maze	The maze should remain consistent to maintain familiarity, whilst having some variation (NPC movement for example).	Should
UR_Story	Narrative elements used to tell the story of the game.	May
UR_Local	All game files should be local to the machine allowing for offline play.	Shall
UR_Movement	The user will be able to move the player through the maze using the keyboard	Shall
UR_Legality	All assets used will be legally sourced and credited.	Shall
UR_Settings	The user should be able to change sound, difficulty, UI elements and other settings in the game.	May

System requirements

Aside from the user requirements, the project's system requirements must also be defined and describe areas of autonomy the team has over the implementation of them.

The team divided these into functional and non functional requirements in order to provide a clear hierarchy of priority, areas in which these share a relationship with user requirements and outline a more precise fit criterion.

Functional System Requirements

ID	Description	User Requirement
FR_Map	The map will be a hardcoded, university style map, with collision.	UR_Maze, UR_Story, UR_Legality
FR_View	The game will be viewed top down, with the	UR_Maze

	player able to move through different rooms.	
FR_Timer	A 5 minute timer to show the player how long they have left to escape.	UR_User_Interface
FR_Game_Over	If the player fails to escape in time, a game over menu will display and they are given the option to restart.	UR_User_Interface
FR_Score	Upon a successful escape, the player's score should be displayed, showing how the time and each of the events affected the overall score.	UR_Events, UR_User_Interface
FR_Pause	The player should have the ability to pause the game at any time.	UR_Accessability
FR_Input	The avatar will be controlled by a set of simple input keys in a standard configuration that is accessible to players.	UR_Accessability, UR_Movement
FR_Leaderboard	A leaderboard showing the names and scores of the most successful players should be able to be accessed and updated upon winning a run.	UR_User_Interface, UR_Local
FR_Achievements	There should be a range of achievements that can be unlocked over a run or series of runs.	UR_Local, UR_Events, UR_Legality

Non-Functional System Requirements

The non-functional requirements along with their respective user requirements and fit criteria were also confirmed by the client feedback.

ID	Description	User Requirement	Fit Criteria
NFR_Logs	Make sure there is a record of the project.		Use Git/Github to manage the codebase.
NFR_Design	The game will be as simple and intuitive as possible whilst maintaining a fun, family-friendly and engaging theme.	UR_Accessability, UR_User_Interface, UR_Maze, UR_Events, UR_Legality	Cartoon style, fantasy theme for the map and characters, along with elements of magic and light humour.
NFR_Events	There should be events in	UR_Events,	Include a range of

	the game that fit with the theme of the game.	UR_Story, UR_User_Interface, UR_Accessibility	interesting and engaging, thematic events. (eg. magic, secret passages) using appropriate assets.
NFR_Controls	Keep to a minimal mapping of simple directional and interaction keys.	UR_Accessability, UR_Movement, UR_Events	Used the standard WASD control scheme.
NFR_Quality	Despite being a prototype, the game should be fully functional and bug free.	UR_Accessability, UR_User_Interface UR_Maze, UR_Local	The API will only be accessible to developers. Player areas and objects shall be clear and intuitive.
NFR_Display	Make sure the game's visuals are runnable on most modern (Windows) machines.	UR_Accessability, UR_User_Interface	30-60fps, 1080x1920 resolution
NFR_Legality	Make sure all third party assets are legally used.	UR_Legality	Check and reference the IP laws to ensure fair use.

References

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

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