

Requirements

Cohort 1, group 9:

Chris Sewell, Fedor Kurochkin, Matt Durham, Max Peterson, Oladapo Olaniran, Wojtek
Tomaszewski, Yuqi Fu.

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 feature 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

From the client interview the team determined that accessibility and target audience were of particular importance since they are relevant to all stages of development.

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).

During the interview the team requested further clarity on the game's different event types and the creative autonomy we had.

We determined that providing each event functioned generally as the client intended the implementation was left to our discretion:

Negative events must be manageable and avoid excessive punishment or confusion.

Positive events provide a fun boon of some kind to the player without making the experience too easy or boring.

A hidden event of some form invisible to the player until triggered.

The next set of requirements identified during the interview were regarding gameplay mechanics and narrative freedom. Again, creative autonomy and implementation were given to the team regarding how the player interacts with the maze and escapes so long as the structure of the maze always remains constant to maintain familiarity. This still left the team with potential to alter objects, NPC's or events to keep players engaged after one attempt at the maze. However the client reminded us that although the teams proposed RPG-like elements, a scoreboard and the inclusion of a narrative were acceptable, the primary goal is to escape the maze in the allotted time thus the player shouldn't be engaged in side quests or other distractions and the score should be a secondary goal.

Finally 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 and without any focus on competitive gaming.

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

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.

Since the win condition of the game is for the player to escape the maze within 5 minutes, a timer will be implemented as part of the player's UI.

Should the player fail to cross the map's exit boundary before the timer expires, an appropriately sized failure message will be triggered along with the options to restart the maze or exit to the main menu.

A simple player score shall also be displayed based on the time remaining upon escaping the maze and how the player handled the various events. The client emphasized the focus to be placed on accessibility and less on competitiveness and difficulty, thus a simple timer and score tracker would suffice.

The client made clear that a pause feature must be added to the prototype but potentially removed for future increased difficulty modes.

We decided a pause command would prompt a small menu to appear with resume, restart and exit to the main menu as the options available to the player.

For navigating the maze, a player's avatar will be controlled by a set of simple input keys mapped in a configuration that is accessible to new and average players alike.

After analysing the requirements referencing system the team agreed upon certain invariants to take top priority: the game must always have at least one means to escape the maze, the timer shall consistently display the current elapsed time and freeze/resume seamlessly during pause and the player avatar should always respond instantly to input commands.

Non-Functional System Requirements

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

The primary goal for the project is to produce an accessible, easy to use and family friendly 2D game aimed at as large of a market as possible and to have a working prototype that meets the described requirements by 10/11/25.

Each stage and iteration of the project lifecycle shall be systematically recorded and preserved via a version controlled repository (Git/ Github).

Any problems or additional queries specific to an iteration of the game can then be clearly and effectively deliberated over by the team and with the client at any stage in the project lifecycle.

Due to the time-frame, the mass-market audience and emphasis on accessibility, we decided on a minimalist approach to game design and difficulty. The game will be as simple and intuitive as possible whilst maintaining a fun, family-friendly and engaging theme.

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.

Further aiding in player accessibility, the control scheme will be kept to a minimal mapping of simple directional and interaction keys both for the avatar and menu. This control scheme will be as intuitive as possible but also be displayed in an instructions window within the menu along with win/fail conditions and basic introduction to the game, with an appropriate font size and contrasting colours for clarity.

Other non-functional priorities include the game's resilience and maintainability. Despite being a first iteration, the game should be fully functional and bug free at launch with no ability for player cheating or game manipulation. The API will only be accessible to developers and the areas/ boundaries and objects available to the player shall be made clear and intuitive.

The game performance should be consistent at a target of 30-60fps, with no custom screen scaling and kept at standard 1080x1920 resolution.

The game will be programmed using a Java based engine compatible with the brief (LibGDX) and executed as a standard PC application available on windows with no current or future support for other devices or platforms.

The client expressed the necessity for any 3rd party assets and libraries to be fully compliant with intellectual property laws.

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, paired with use case diagrams.

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.