Requirements
Team 5
Lizzard Entertainment

Gregory Binu
Lydia Eaton
DJ Fox
Daniel Maffei
Michael Papafilippou
Nathalie Rizzo

Using the requirements structure detailed in the requirements engineering video [1] the team compiled the necessary user requirements (tasks the user should be able to carry out on the system) and system requirements (how the system will deliver on the needs of the users) for our project. We first gathered the requirements by interviewing our stakeholder before finalising and structuring them in tables.

The team used the project brief given to us as a starting point to make questions for our stakeholder meeting. We carefully went through and planned the interview, breaking it down into the main sections; systems, events, game aesthetic/ story, scoring/ game ending, difficulty, audience, general.

Two of our team members then conducted the meeting with the stakeholder and discussed requirements they had for the game, using a semi-structured interview style. Throughout the meeting the team took notes and the meeting was also audio recorded for further analysis.

We went through the notes gathered in the stakeholder meeting and the audio recording of the meeting to cover any points the notes missed. From here we negotiated our user requirements, checking they covered everything our customer wanted in the game.

Once we had finalised our user requirements, we went through each one individually and conceived all the system requirements needed for the user requirement to be successfully carried out.

The team decided on a table structure to display requirements, with each requirement having a unique id prefixed with UR, FR or NFR (user, functional, non-functional). Presenting our requirements in this manner made it very clear what the purpose of each requirement was, and very easy to refer to specific requirements, not just for system requirements but also when requirements need to be referred to later in Architecture.

Each record in the user requirements table had its id, a description of the requirement, and a priority (shall/ should/ may). "Shall" refers to requirements that must be met whereas "should" means we could be without them. "May" is for requirements that would be nice to have but should be viewed more as extra. This clearly communicated what the requirement was there for, and how important it was to be implemented.

In the systems requirements table, requirements were broken down into functional and non-functional. Functional requirements refer to required responses a system must have, properties that must be upheld and forbidden responses that shouldn't happen. These requirements were recorded with an id, description and the id of the user requirement they linked to.

Non-functional requirements articulate how a system should perform, for example its reliability or security. These were recorded with an id, description, id of the user requirement it linked to and the fit criterion. The fit criterion is a condition that has to be met for the requirement to be successfully realised. An example would be "the game start screen will load in < 2 seconds".

Overall, this process and structuring of the requirements worked well for a project of this scale as all requirements the user had for the game had been collected and recorded in a timely manner, ready for the team to quickly move onto Architecture and Implementation.

Single Statement of Need: A 5-minute long pausable single-player game involving escaping a university-like maze with multiple obstacles, to achieve a high score.

User requirements:

ID	Description	Priority
UR_CONTROL_CHARACTER	Users should be able to control their character.	Shall
UR_PAUSE_GAME	Users should be able to pause/unpause the game.	Shall
UR_SEE_TIMER	Users should be able to see how much in-game time has elapsed (excluding paused time).	Shall
UR_FINISH_GAME_SESSION	Users should complete the game within 5 minutes of in-game time or the game will end.	Shall
UR_UNDERSTAND_OBJECTIVE	Users should be able to understand the objective of the game, as well as how to score maximum points.	Shall
UR_POSITIVE_EVENT	Users should be able to partake in an event that will benefit them.	Shall
UR_NEGATIVE_EVENT	Users should be able to partake in an event that will hinder them from progressing.	Shall
UR_HIDDEN_EVENT	Users should be able to activate a hidden event.	Shall
UR_MUSIC	Users should hear atmospheric background music while in game.	May
UR_VOLUME	Users should be able to adjust the volume of the background music, or mute it.	May

System requirements:
Functional requirements:

ID	Description	User Requirement
FR_CHARACTER_MOVEMENT	The system should move the character sprite when a user input is made on the keyboard.	UR_CONTROL_CHARACTER
FR_PAUSE	The system should pause the game and timer when	UR_PAUSE_GAME

	control is pressed.	
FR_UNPAUSE	The game should continue the game and timer when the user unpauses the game	UR_PAUSE_GAME
FR_WIN_GAME	The system should present the user with their score, calculated as a combination of time taken as well as other factors, after completing the maze within 5 minutes.	UR_FINISH_GAME_SESSION
FR_LOSE_GAME	The system should end the game after 5 minutes of in-game time (not paused)	UR_FINISH_GAME_SESSION
FR_DISPLAY_OBJECTIVES	The system should use concise and easy-to-understand language to inform the user of the game objective as well as how points are rewarded	UR_UNDERSTAND_OBJECTIVE
FR_OFFER_EVENT	The system should run an event (positive/negative) when the user interacts with it on screen.	UR_POSITIVE_EVENT UR_NEGATIVE_EVENT
FR_HIDDEN EVENT	The system should run a hidden event when the user triggers it.	UR_HIDDEN_EVENT
FR_REPEAT_EVENT	The system should rerun an event multiple times if a user chooses to interact with it multiple times.	UR_POSITIVE_EVENT UR_NEGATIVE_EVENT
FR_MUSIC	The system should begin playing the music track when the player loads into the game.	UR_MUSIC
FR_VOLUME	The system should lower or turn up the volume when the user interacts with volume controls.	UR_VOLUME

Non-functional requirements:

ID	Description	User Requirement	Fit Criterion
NFR_INTUITIVE_INPUTS	User inputs to control character should be intuitive and user-friendly.	UR_CONTROL_CHARACTER	Users should be able to figure out the controls by themselves
NFR_LEGIBLE_TEXT	In order to make understanding the goals of the game as easy as possible, the text should be easy to read through appropriate fonts and no clashing with the background.	UR_UNDERSTAND_OBJECTIVE	Users should be able to read the text easily across many display sizes, and should not need to reread it or change their brightness/colour settings
NFR_OBJECT_DESIGN	All interactive parts of the game shall be differentiated by shape not colour, for visuals to be clear for users who are colourblind.	UR_UNDERSTAND_OBJECTIVE	Users should be able to easily differentiate between game objects.

References:

[1] Kolovos, D (2025), *Requirements Engineering*, Department of Computer Science, University of York.