Requirements

Cohort 2 Group 4 - THADJAM

Abdul Fofanah
Matthew Holleran
Toby Watchorn
Arwen Minton
Daneena Roydean
Jessica McKerill
Henry Bambrough

Requirements Elicitation

Requirements for this project were elicited through a structured, research-informed interview with the customer. Guided by the pre-established engineering requirements, the team developed a set of open-ended, targeted interview questions to identify the customer's objectives, desired system features and constraints. The interview was recorded to later systematically analyse responses.

During customer response analysis, the team extracted explicit requirements directly mentioned by the customer and determined implicit requirements where design or implementation decisions were left to the team's discretion. To organise these findings clearly and ensure traceability, the team documented all requirements in structured tables. Each requirement was assigned a unique identifier (ID), allowing for clear referencing between related requirements and helping simplify later validation and testing processes.

To ensure accuracy and completeness, the team validated the derived requirements by cross-checking them against the customer's goals and the engineering specification. Each user requirement was then prioritised based on its importance to core functionality and user experience, forming the foundation for subsequent design and development decisions.

User Requirements Table

ID	Description	Priority
UR_TUTORIAL	The system will provide a tutorial to teach the player the basics of the game.	Shall
UR_START_GAME	The system shall load onto a start screen.	Shall
UR_UX	The system shall offer a pleasant, family-friendly experience.	Shall
UR_SETTINGS	The system shall include a pause game/settings menu with options to adjust volume, restart or exit.	Shall
UR_OFFLINE	The system should be fully operational without a network connection available.	Shall
UR_USER_TIME	The user should be able to see how long they have been playing for.	Should
UR_GAME_COMPLETION	The game should take the user about 5 minutes to complete.	Shall
UR_END_SCORE	The user shall find out their final score at the end of the game.	Should
UR_ACCESSIBILITY	The system shall provide accessibility options in the settings menu with features such as subtitles and a colourblind mode.	Should
UR_PROTAGONIST	The system shall make the main character a student at the University Of York.	May
UR_MOVEMENT	The system shall let the user move their character around the maze.	Should
UR_EVENTS	The system shall have multiple events of each type with a minimum of 3	Should

	positive events, 5 negative events, and 3 hidden events, alongside a counter showing progress in X/Y format (e.g., '⅔ positive events found').	
UR_ANTAGONIST	The system shall have an antagonist that follows the player around the maze.	Shall
UR_RESTART	The system shall allow the user to restart the game at any point.	Shall

System Requirements

Functional Requirements Table

ID	Description	User Requirements
FR_USER_TIMER	The system shall display a timer to the user which displays how long they have been playing for.	1. UR_USER_TIME
FR_USER_TIME_FINAL	The system shall display the user's final time once they have completed the game.	1. UR_USER_TIME
FR_EXIT_GAME	The system shall allow the user to exit the game at any time of their choosing.	1. UR_SETTINGS 2. UR_RESTART
FR_END_SCORE_TOTAL	The system shall display the user's total final score on an ending screen.	1. UR_END_SCORE
FR_END_SCORE_BREAKDOWN	The system shall display a breakdown of the user's points (e.g. quick completion bonus +2 points).	1. UR_END_SCORE
FR_PLAYABILITY_INCLUSIVITY	The system shall accommodate diverse users by having subtitles and features distinguishable by not just colours.	1. UR_ACCESSIBILITY
FR_DEFAULT_CHARACTER	The system shall display a default character as a sprite which the user has control over.	1. UR_PROTAGONIST
FR_MOVEMENT	The system shall allow keyboard inputs (WASD) to allow the user to have control over the sprite and interact with the maze.	1. UR_MOVEMENT
FR_TUTORIAL_PAUSING	The system shall allow the player to pause the tutorial at any given time from Settings.	1. UR_TUTORIAL 2. UR_SETTINGS
FR_UX_MUSIC	The system shall use appropriate music and sound effects that contain no lyrics.	1. UR_UX
FR_UX_DESIGN	The maze map shall contain family-friendly themes only.	1. UR_UX
FR_UX_PLAYABILITY	The game's difficulty will be kept to a lower level to maintain a pleasant experience and welcome users with limited gaming experience.	1. UR_UX 2. UR_ACCESSIBILITY
FR_OFFLINE	The system and all its features shall not require any connection to a network.	1. UR_OFFLINE

FR_SETTINGS_OPTION	The system shall include a settings option for the user where they can pause the game and adjust sound controls. The setting shall also present the user with the options to exit the game, alongside restarting the game. The user can also select the tutorial.	1. UR_SETTINGS
FR_END_SCORE	The system shall display the user's end score when they complete the game or fail the game.	1. UR_END_SCORE 2. UR_SETTINGS
FR_POSITIVE_EVENT_LOCKER	The system shall present a positive event in which the player finds a locker, opens it and gains a speed boost. The movement speed increases from 1f to 2f while active. Implementation: Locker.java	1. UR_EVENTS
FR_POSITIVE_EVENT_FREEZE_DEAN	The system shall allow the player to activate lab materials that freeze all deans (chase dean and patrol deans) for 30 seconds by pressing the E key. Implementation: Freeze_Dean.java	1. UR_EVENTS
FR_POSITIVE_EVENT_EXTRA_TIME	The system shall provide an NPC (non-player character) that adds 30 seconds to the game timer when the player collides with it. Auto-collects on contact and disappears after pickup. Implementation: Extra_Time.java	1. UR_EVENTS
FR_NEGATIVE_EVENT_DEAN_CHASE	The system shall present a negative event of the Dean chasing the player around the maze. When caught, the player is reset to the starting position with a -5 points penalty. Implementation: Dean.java	1. UR_EVENTS
FR_NEGATIVE_EVENT_PATROL_DEAN	The system shall include 3 deans that patrol vertically in the bottom area at 0.7f speed. Collision with any patrol dean resets the player to the starting position with a -5 points penalty. Fourth patrol dean spawns if the questionnaire is failed. Implementation: Patrol_Dean.java	1. UR_EVENTS
FR_NEGATIVE_EVENT_SLOW_DOWN	The system shall include a bush obstacle that reduces player movement speed by 50% for 20 seconds when collided with. Implementation: Slow_Down.java	1. UR_EVENTS
FR_NEGATIVE_EVENT_DROWN	The system shall include water tiles that trigger player respawn at the starting position with a -10 points penalty when collided with. Implementation: Drown.java	1. UR_EVENTS
FR_NEGATIVE_EVENT_DECREASE_TIME	The system shall include a tree obstacle that reduces remaining game time by 30 seconds when collided with. Implementation: Decrease_Time.java	1. UR_EVENTS
FR_HIDDEN_EVENT_BUS_PASS	The system shall contain a hidden event of the player's bus pass being hidden in a bush that, until triggered, the bus pass will remain undiscovered to the player. The player must interact with the bus to trigger a win condition. Implementation: BusTicket.java	1. UR_EVENTS
FR_HIDDEN_EVENT_T	The system shall include a hidden science lab entrance	1. UR_EVENTS

ELEPORT	that teleports the player to a shortcut location when triggered. Implementation: Teleport.java	
FR_HIDDEN_EVENT_QUESTIONNAIRE	The system shall include a hidden questionnaire that awards bonus points for correct answers. Failing the questionnaire spawns a fourth patrol dean. Implementation: Questionnaire.java	1. UR_EVENTS
FR_ACHIEVEMENT_SYSTEM	The system shall track player achievements based on gameplay performance. Each achievement has a name, description, and bonusScore (positive or negative integer). Implementation: Achievement.java	1. UR_END_SCORE
FR_LEADERBOARD_SAVE	The system shall save the top 5 player scores persistently in a local JSON file (leaderboard.json). Implementation: Save_Leaderboard.java	1. UR_END_SCORE
FR_LEADERBOARD_DISPLAY	The system shall display a leaderboard screen showing the top 5 scores with the format '[rank]- [fullName] - [score]'. Implementation: LeaderboardScreen.java	1. UR_END_SCORE
FR_NAME_ENTRY	The system shall provide a name entry screen. Implementation: NameScreen.java	1. UR_END_SCORE
FR_ANTAGONIST	The system shall include an antagonist, the dean, who follows the player around the map, and if they catch the player, the player is reset to the starting position with a -5 point penalty. Implementation: Dean.java	1. UR_ANTAGONIST

Non-functional Requirements Table

ID	Description	User Requirements	Fit Criteria
NFR_GAME_COMPLETION	The game should not last too long.	1. UR_GAME_COMPLETION	90% of users complete the game within 5 minutes.
NFR_RESTART_GAME	The system shall allow the user to restart the game at any time by taking them back to the tutorial page to start over.	1. UR_SETTINGS	The game should restart in < 6 seconds.
NFR_PLAYABILITY_DIFFICULTY	The system shall not require any prior gaming experience to interact with.	1. UR_SETTINGS	70% of users achieve a high total score.
NFR_STABILITY	The system shall work reliably for those on supported setups / OSes.	1. UR_UX 2. UR_OFFLINE	95% of users should not experience any crashes on supported hardware when playing the game for at least half an hour.
NFR_ACCURATE_INPUT	The system should have inputs that work as intended, with minimal lag.	1. UR_MOVEMENT 2. UR_SETTINGS 3. UR_UX	The player should respond to any valid input on the keyboard within 3 frames.
NFR_TUTORIAL_LOADING	Upon clicking start game, the system should present the user with a tutorial sequence to demonstrate how to play the game.	1. UR_TUTORIAL	The system should load the tutorial sequence in < 6 seconds.
NFR_TUTORIAL_PROGRESSION	The system shall give the user an appropriate amount of time to read each slide of the tutorial before progressing to the next.	1. UR_TUTORIAL	The slides will be displayed for 10-15 seconds, depending on their length.
NFR_PAUSING	The system shall allow the user to pause at any time using the P key.	1. UR_SETTINGS	The system should respond to a user pausing within 3 seconds.
NFR_MUSIC_CONTROL	The system shall respond to the user by muting the music as well as adjusting the volume of the music.	1. UR_SETTINGS	The system shall respond in < 4 seconds.
NFR_SOUND_EFFECTS_CONTROL	The system shall respond to the user muting as well	1. UR_SETTINGS	The system shall respond in < 4 seconds.

	as adjusting the volume of the sound effects.		
NFR_EXIT_GAME	The system shall allow the user to exit the game at any time.	1. UR_SETTINGS	The system shall take < 6 seconds to close.
NFR_EXIT_GAME_SCREEN	The system shall display an exit game screen to the user, which displays their current statistics and a farewell message.	1. UR_SETTINGS	The system shall display the exit game screen for 15-20 seconds before the game closes.
NFR_RESTART_GAME	If the user selects to restart the game in setting the system shall allow the user to start the game over again from scratch.	1. UR_SETTINGS	The system shall take < 6 seconds to restart.
NFR_RESTART_GAME_SCREEN	The system shall display an exit game screen to the user, which displays their current statistics.	1. UR_SETTINGS	The system shall display the restart game screen for 12-20 seconds before restarting the game in < 6 seconds.
NFR_START_GAME	The system shall open up to a start screen.	1. UR_START_GAME	The system should load the screen in < 6 seconds.
NFR_MAP_SIZE	Map dimensions expanded approximately 50% (640x640 pixels) to accommodate event locations while maintaining a 5-minute completion target.	1. UR_EVENTS 2. UR_GAME_COMPLETION	90% of users complete the game within 5 minutes with an expanded map.

Constraints Requirements Table

Project Constraints	Development Process Constraints	Design & Technical Constraints
Game Scope Constraint: - The game should be designed as a 'one-shot' experience without implementing account systems or local data saving.	**Asset & Legal Constraint:** - All third-party assets utilised in the game's development must be appropriately licensed or self-developed to respect IP rights.	**Technology Constraint:** - The game shall be built to run on a standard computer/desktop.
Financial and Resource Constraint: - The project should be developed with a £0 budget, by utilising already licensed tools, software engines and assets.	**Gameplay Constraint:** - The game shall not exceed a single-player game-play format.	**Gameplay Constraint:** - The game shall implement more than one positive, negative and hidden event, respectively.
Game Scope Constraint: - The game shall be limited to a single, finite maze. No infinite runners or selectable maze levels should be implemented.	**Gameplay Constraint:** - The game's progression must be static; player decisions should not deviate from the game's narrative nor alter the core maze layout during gameplay.	**Gameplay Constraint:** - The user's game session must have a maximum real world duration of five minutes tracked by an in-game timer.
	Development Life Cycle Methodology: - Git shall be solely used to manage all source code and design documents to document changes and enable collaboration.	**Gameplay Constraint:** - The game should be designed with a single, standard difficulty level to make the game accessible to a wider audience of players.
	Development Life Cycle Methodology: - The game shall be built using an incremental development process (scrum/agile-based sprints) to facilitate regular testing and refinement of code.	**Event Constraint:** - The game shall implement multiple events with a minimum of 3 positive events, 5 negative events, and 3 hidden events. - The game must include a counter to track how many of each event type the user has interacted with in X/Y format.
	Coding Standards: - The code must be written with clear naming conventions and sufficient comments to ensure it is fit for cross-collaboration. - Core game design decisions	**Accessibility Constraint:** - The game must provide an option to mute/disable all music and sound effects. **User-interface Constraint:** - The game must provide a

	(event list, scoring mechanisms) must be collated into the documentation prior to implementation.	pause/exit option allowing the user to quit or restart their session.