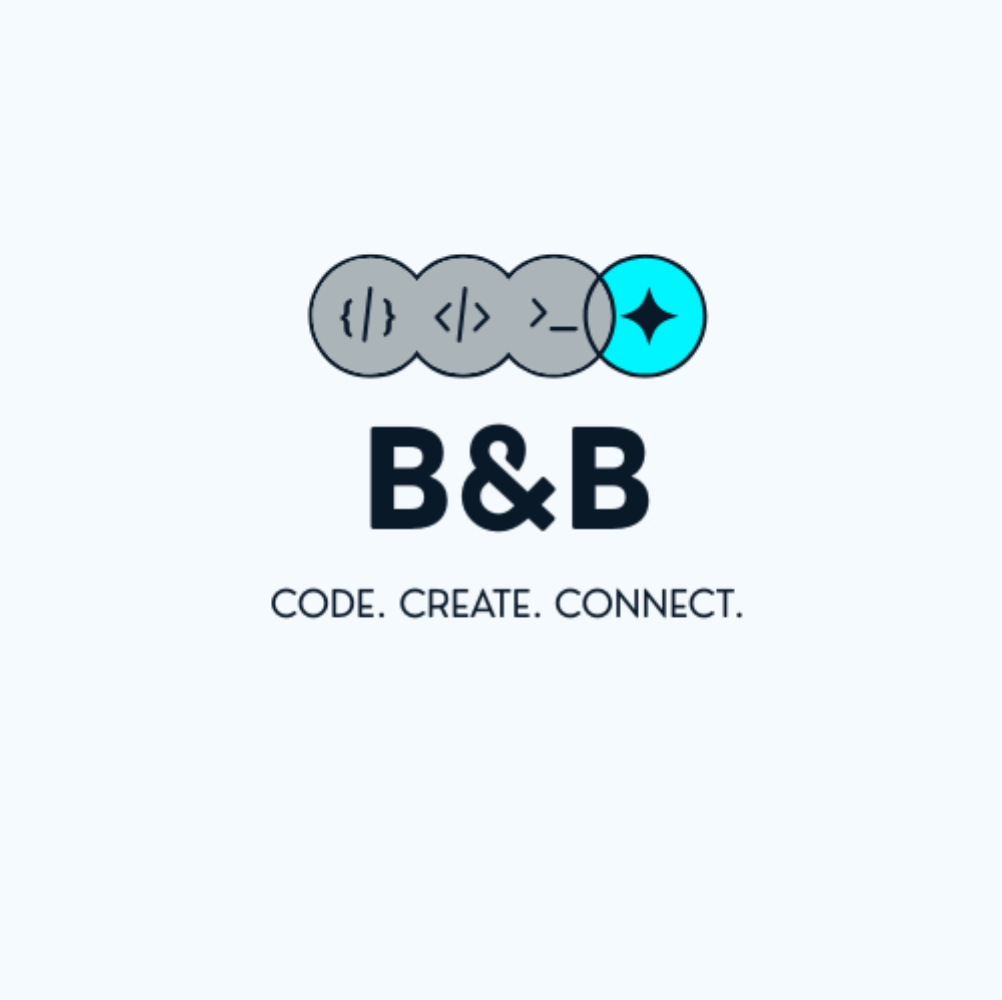
# Requirements

Team 8: Ben and Bensons




Team Members: Alyssa Skipper, Ben Senior, Benson Chow, Chloe Ward, Florian Mengkris, Hannah Thomas, James Ingram, Olivia Spencer

## Introduction

These requirements are informed by the assessment brief provided by the customer, in addition to the customer's responses to our questions during a meeting on 07/10/2025. In advance of this meeting, the team analysed the brief and noted any essential missing information, mainly regarding implementation details (such as operating systems) and design decisions (such as inclusion of sounds and the structure of the map), to ask the customer about. We also discussed any of the unstated wishes of the customer, such as a leaderboard or multiplayer, and clarified on some vague descriptions in the brief.

The presentation and contents of this software requirements specification (SRS) have been designed primarily under the guidance of IEEE 830-1998[1]. Whilst this standard has been superseded by ISO/IEC/IEEE 29148:2018[2], most of the modern refinements are unlikely to be of concern to a project of this scale. The ambiguity of unrestricted natural language is discussed without specific guidance in IEEE[1, p. 5], so for further guidance we have also taken into consideration more modern discussions of which languages or notations to use for writing an unambiguous SRS[3, p. 12].

Since we intend to use a single requirements document for all stakeholders, we decided that formal notations such as Z-Notation[4] and Clear[5, p. 1050] would not be appropriate nor intuitive for those unfamiliar with them [1, p. 5]. Modern SRSs are predominantly written in *restricted* natural language[6, p. 378], so we've decided to use a particular restriction of English known as ACE[7] as rough guidance. ACE is easy to understand for all stakeholders and restricts grammar to a collection of concise constructs in order to ensure concreteness [7, p. 5]. We are also keeping in mind further cautions outlined in ISO/IEC/IEEE 29148:2011[3, p. 12], such as avoiding statements that are not verifiable.

Within the requirements we will also use the terms *shall*, *should* and *may* to express extents of priority. Throughout this document, these words will be highlighted in a "priority" column in the requirements table and exclusively used with the following meanings:

- *Shall:* If the requirement is not met then the software will be unacceptable.
- *Should:* The requirement is not essential but it would enhance the product.
- *May:* The requirement is acceptable but it is not guaranteed to enhance the product.

These definitions were constructed to minimise ambiguity and subjectivity, mirroring the corresponding definitions of the *degrees of necessity* specified in IEEE[1, p. 7].

IEEE[1, pp. 12–17] discusses separating the user and system requirements into many different sections, such as interfaces, logical database requirements and performance requirements: we believe this to be excessive for a project of this scale, and so we've kept our statement of requirements to three key sections: user requirements (URs), functional requirements (FRs) and non-functional requirements (NFRs). We've had discussions of separating requirements by the "mode" the program runs in, however we concluded that multiple deployed modes would be unnecessary due to the briefness and simplicity of the program. We also decided to use an iterative[8, p. 54] approach to constructing the requirements, making it easier to refine the document if any problems or changes occur. To keep track of changes made, we decided to use an internal requirement repository that tracks which requirements supersede others, which we can then document here whenever significant changes occur.

## Simple Statement of Requirements

*SSON: Tommy needs an exciting, 2D PC game demo designed for prospective students of the UoY where the goal is to escape the university within a five-minute time frame.*

### User Requirements

| ID | Description | Priority |
|---|---|---|
| UR_OS_REQ | The game shall be run on desktop as an OS-agnostic file. | Shall |
| UR_SCALE | The game's graphics should scale with the size of the physical screen. | Should |
| UR_FAMILY | The game's content shall be appropriate for all ages. | Shall |
| UR_SINGLE | The game shall be playable by a single person. | Shall |
| UR_PERS | The game's graphics shall be two-dimensional and top-down. | Shall |
| UR_PAUSE | The game shall be able to be paused at any time and shall start in a paused state. | Shall |
| UR_TIMER | The screen shall have a timer which counts down from 5 minutes whilst the game is unpaused. | Shall |
| UR_WIN | The game shall be won when the player escapes the map. | Shall |
| UR_LOSE | When the timer reaches 0 the game shall be lost. | Shall |
| UR_AUDIO | The game should have audio features and music which both are able to be muted. | Should |
| UR_VOLUME | The volume of the game's audio may be adjustable via the pause menu. | May |
| UR_KEYBIND | The game shall use intuitive and well-understood controls. | Shall |
| UR_SAVE | The game may offer both saving and loading features. | May |
| UR_ACCESS | The visuals shall be accessible for colourblind users. | Shall |
| UR_MAP | The game's map shall be a maze with locations similar to those at the university. | Shall |
| UR_PATHS | The player's movement shall be constrained to specific paths on the map. | Shall |
| UR_ALT_PATHS | The map may contain multiple different correct paths. | May |
| UR_CHAR | The player may be able to select from a number of character appearances. | May |
| UR_HIDDEN | The game shall have at least 3 hidden events which are triggered by interactions | Shall |
| UR_HINDRANCE | The game shall have at least 5 visible events which hinder the player. | Shall |
| UR_BENEFIT | The game shall have at least 3 visible events which benefit the player. | Shall |
| UR_EASE | The game's difficulty should not be frustrating for inexperienced players. | Should |
| UR_INSTRUCTIONS | The game shall provide clear instructions accessible from the main menu. | Shall |
| UR_MENU | The game shall have a main menu with at least three options: Play, Instructions and Quit. | Shall |
| UR_END | The game shall display a win or loss screen upon game completion. | Shall |
| UR_GRAPHICS | The game’s interface and visuals should be easy to follow and consistent. | Should |
| UR_DOC | The game shall be clearly documented to facilitate future development and maintenance. | Shall |
| UR_EFFICIENT | The game shall run efficiently without any frustrating performance issues. | Shall |

## Functional Requirements

| ID | Description | User Requirement | Priority |
|---|---|---|---|
| FR_MUSIC | Music should play from when the game starts to when it ends. | UR_AUDIO | Should |
| FR_MUTE | The pause screen should have a "mute" button which toggles audio. | UR_AUDIO | Should |
| FR_PAUSE_MENU | Pressing ESC shall open a pause menu with options to resume and quit. | UR_PAUSE | Shall |
| FR_VOLUME | The pause screen may allow the volume of audio to be adjusted. | UR_VOLUME | May |
| FR_SAVE_LOAD | The pause screen may allow the game to be saved and loaded. | UR_SAVE | May |
| FR_MAIN_MENU | When the game starts, the main menu shall be displayed, providing options for Play, Instructions and Quit. | UR_MENU | Shall |
| FR_INSTRUCTIONS_SCREEN | When "Instructions" is selected, the controls and objectives shall be displayed. | UR_INSTRUCTIONS | Shall |
| FR_PLAY | When "Play" is selected, the game shall transition from the main menu to the active game screen. | UR_MENU | Shall |
| FR_QUIT | When "Quit" is selected, the game application shall close. | UR_MENU | Shall |
| FR_MOVEMENT | The player shall be able to move using arrow keys or WASD keys. | UR_KEYBIND | Shall |
| FR_COLLISIONS | Player movement shall be constrained by maze boundaries. | UR_PATHS | Shall |
| FR_RESUME | The resume button shall return the player to the active game screen from the pause menu. | UR_PAUSE | Shall |
| FR_TIMER | The game shall display a timer that starts when gameplay begins and counts down only whilst the game is unpaused. | UR_TIMER | Shall |
| FR_LOSE_CONDITION | When the timer reaches zero, the game shall transition to the end screen showing a loss message. | UR_LOSE | Shall |
| FR_WIN_CONDITION | When the player reaches the ‘Central Hall’ area, the game shall transition to the end screen and win. | UR_WIN | Shall |
| FR_END_SCREEN | When the user wins/loses a game, an end screen shall display the results of the game. | UR_END | Shall |
| FR_BENEFICIAL | The game shall contain at least 3 beneficial events triggered during play. | UR_BENEFIT | Shall |
| FR_HINDRANCE | The game shall contain at least 5 hindering events triggered during play. | UR_HINDRANCE | Shall |
| FR_HIDDEN | The game shall contain at least 3 hidden events triggered during play. | UR_HIDDEN | Shall |
| FR_CHAR | When paused, the player may be able to change the appearance of their character sprite | UR_CHAR | May |

## Non-Functional Requirements

| ID | Description | Fit Criterion | User Requirement | Priority |
|---|---|---|---|---|
| NFR_OS_REQ | The game shall be run via a platform-agnostic desktop file. | The game successfully runs on Windows and two other OSs. | UR_OS_REQ | Shall |
| NFR_PERS | The game's sprites shall be from a 2D, top-down perspective. | At least three unfamiliar users agree that every sprite in the game files is recognisable under the given description. | UR_PERS | Shall |

| | | | | |
|---|---|---|---|---|
| NFR_HELP | The controls shall be easy to use for the player. | At least three unfamiliar users are able to complete the game without external help. | UR_INSTRUCTIONS | Shall |
| NFR_FILE_PATH | The game's capacity to run shall not depend on the root directory. | The game successfully runs with the root folder placed in two different directories. | UR_OS_REQ | Shall |
| NFR_RESOLUTION | The game's window should scale to fit the resolution of the user's screen. | The game successfully scales up/down between 2 different test screens. | UR_SCALE | Should |
| NFR_FAMILY | The game shall not hold any content that would take it above the PEGI-12 age rating. | The game does not meet any of the criteria to be classed as PEGI-16 or PEGI-18 as opposed to PEGI-12 | UR_FAMILY | Shall |
| NFR_SINGLE | The game shall not have any form of multiplayer. | The game runs offline with a single player in control of the device. | UR_SINGLE | Shall |
| NFR_PAUSE_DELAY | There shall be a negligible delay between the game pausing and the timer pausing in sync. | During testing, mean delay over at least 3 recordings is less than 0.1 seconds. | UR_PAUSE | Shall |
| NFR_ACCESS | The game shall be visually accessible to colourblind users. | An unfamiliar user is able to play the game comfortably in greyscale without any issues. | UR_ACCESS | Shall |
| NFR_EASE | The conditions to win the game should not be frustrating to satisfy for the player. | At least 3 unfamiliar users agree that the game isn't frustrating or too challenging to complete. | UR_EASE | Should |
| NFR_MAP | The game map shall have locations that resemble those of the University of York | A member of the university recognises all locations on the map that are inspired by the university. | UR_MAP | Shall |
| NFR_ALT_PATHS | The game map's route may contain multiple correct paths to the exit. | At least 3 familiar users agree that there are multiple paths to the win condition. | UR_ALT_PATHS | May |
| NFR_GRAPHICS | The UI and visuals should be clear, consistent and readable. | At least 3 unfamiliar users agree that the UI and visuals are clear, consistent and readable. | UR_GRAPHICS | Should |
| NFR_MAZE | The paths the player can take shall resemble a maze. | At least 3 familiar users agree that the layout of the paths available to them resemble a maze. | UR_MAP | Shall |
| NFR_DOC | The game's code and design shall be clearly documented and consistently named. | An unfamiliar tester is able to explain any section of the game's code or design from a first glance. | UR_DOC | Shall |
| NFR_EFFICIENCY | The game's code is optimised to use time and resources efficiently for seamlessness. | The game runs without noticeable delays on a PC owned by the University of York. | UR_EFFICIENT | Shall |

## References


[1] *830-1998 - IEEE Recommended Practice for Software Requirements Specifications*. [Online]. Available: https://ieeexplore.ieee.org/document/720574. [Accessed: 09 Oct. 2025].

[2] *ISO/IEC/IEEE 29148: 2018(E): ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering*. 2018.

[3] 'ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes --Requirements engineering', *ISO/IEC/IEEE 29148:2011(E)*, pp. 1–94, 2011.

[4] M. Spivey, *The Z Notation: A Reference Manual (Second Edition)*. Prentice Hall, 1998.

[5] Rod M. Burstall and Joseph A. Goguen, 'Putting Theories Together to Make Specifications', in *International Joint Conference on Artificial Intelligence*, 1977[Online]. Availablehttps://www.ijcai.org/Proceedings/77-2/Papers/095.pdf[Accessed: 9October2025].

[6] X. Franch et al., 'The state-of-practice in requirements specification: an extended interview study at 12 companies', *Requirements Engineering*, vol. 28, no. 3, pp. 377–409, Apr. 2023.

[7] N. E. Fuchs and R. Schwitter, 'Attempto Controlled English (ACE)', Mar. 13, 1996[Online]. Availablehttp://arxiv.org/abs/cmp-lg/9603003[Accessed: 9October2025].

[8] D. Firesmith, 'Modern Requirements Specification', *J. Object Technol.*, vol. 2, pp. 53–64, 03 2003.