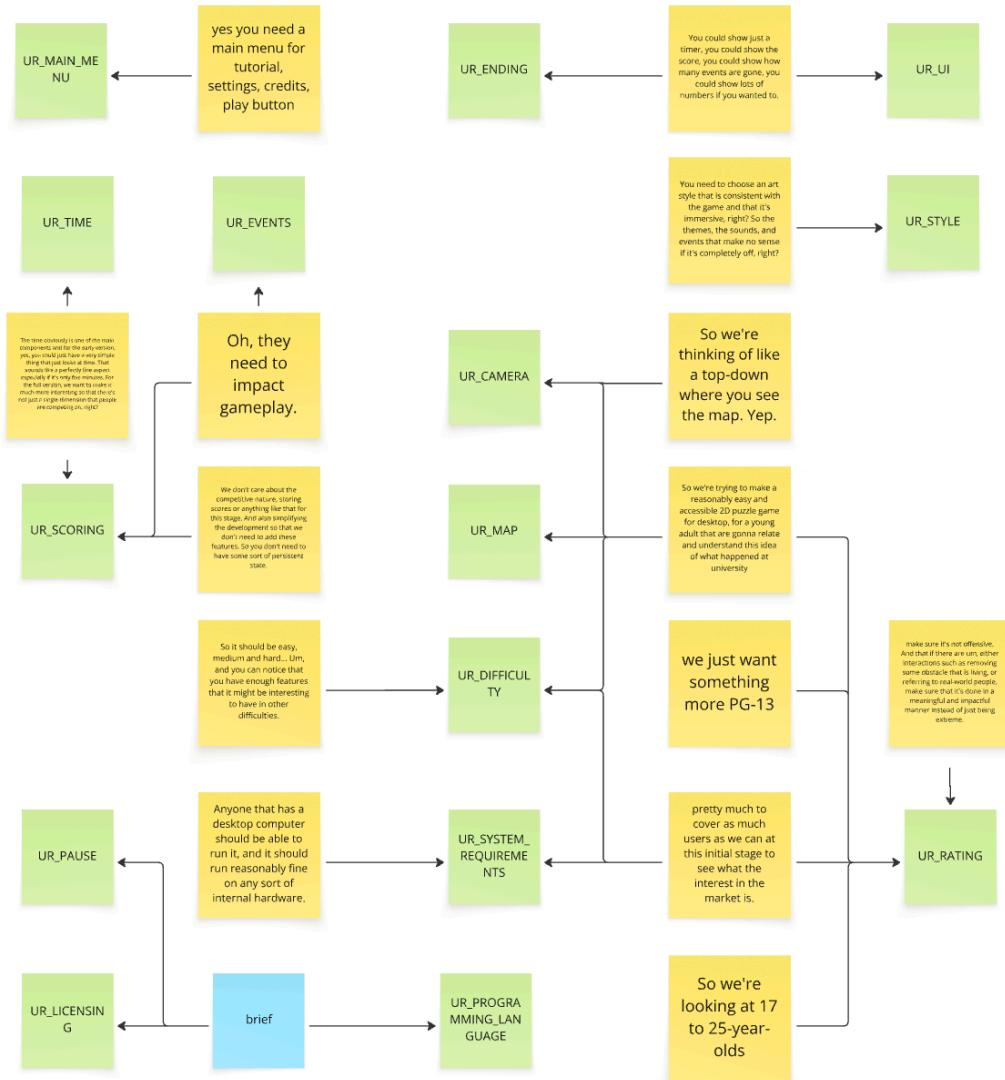Requirements

Cohort 3 Team 7 (Yetti)

Bak, Bartek
Burberry, Katelyn
Collins, Lucy
Ganley, Joe
Keegan, Josh
Kirkham, Katharine
Mayall, Daniel
Sawdon, Theo

Introduction:
The following requirements are presented in three sections: user requirements, functional requirements and non-functional requirements. Each requirement is labeled with a unique id so it can be referenced in other deliverables. The format for the ids is as follows: UR_XXXX is a user requirement, FR_XXXX is a functional requirement and NFR_XXXX is a non_functional requirement, where XXXX is a descriptive identifier of said requirement. It is formatted in this way to help with referencing in other deliverables, as the identifiers clearly state what the requirement is and are unique so as to avoid confusion when being used.

Requirements were gathered from the design brief initially presented to us and then expanded upon through customer meetings and their transcriptions which we analysed to identify what the user wants from the game and to what degree each requirement is necessary. For many requested features only the broad framework and theming was specified, leaving us with a lot of freedom but also meaning some requirements cannot be made more specific at this stage. To begin we converted answers we got from the meeting to user requirements which we then break down into several functional and non-functional requirements designed to be more targeted and give a clearer description of what needs to be developed for the game.

User Requirements

ID	Description	Priority
UR_RATING	Game should be appropriate for children (age-12), no offensive content	Shall
UR_LICENSING	Game should not result in company being sued	Shall
UR_EVENTS	The game will have 5 visible events that hinder the player, 3 that benefit the player and 3 hidden events	Shall
UR_SCORING	User has a score and can win by achieving the best score	Shall
UR_MAP	There are several maps for the user to choose from and play through	Shall
UR_PAUSE	Game should be able to be paused at least on low difficulty	Shall
UR_STYLE	Art style should be consistent and immersive	Should
UR_TIME	Game should last 5 minutes maximum	Shall
UR_SYSTEM_REQUIREMENTS	The game should be able to run on any operating system and minimal hardware to reach as many users a possible	Shall
UR_DIFFICULTY	The game should have difficulty modes may remove some features or somehow make gameplay harder compared to the base difficulty	May
UR_CAMERA	The game should be top down and the player should be able to see the map.	Should
UR_ENDING	The game should have an end screen which at a minimum displays their score	Shall
UR_UI	The game should have a UI showing the time remaining, pause button	Should
UR_PROGRAMMING_LANGUAGE	The game MUST be coded in Java 17	Shall
UR_MAIN_MENU	The game should open with a main menu with buttons to start a new game, open settings and exit. The game should return here after each maze.	shall

Functional Requirements

ID	Description	User requirements
FR_CREDITS	The game should properly credit licensed products used (like sprites and art) on screen (such as credit screen) when the game is played	UR_LICENSING
FR_POSITIVE_EVENTS	Throughout the duration of the game, the game must contain three beneficial events which the player has to interact with and must benefit them in some way.	UR_EVENTS
FR_NEGATIVE_EVENTS	Throughout the duration of the game, the game must contain two negative events which the player has to interact with and must impede them in some way.	UR_EVENTS
FR_HIDDEN_EVENTS	Three of the events within the game must be hidden, meaning they must not be visible until the player activates them.	UR_EVENTS
FR_SCORING	The game must calculate score based on time taken to escape and any point modifiers from events	UR_SCORING
FR_MAP_CREATION	The program has to load a pre-made, or generate, a map at the start of the game which has all events within it and has an escape	UR_MAP
FR_MAP_STYLE	Map should have a consistent art style and should be immersive	UR_STYLE
FR_EVENT_STYLE	Events must have a consistent art style and be immersive	UR_STYLE
FR_PAUSING	Game must have a pause function, with a pause screen or indicator that the screen is paused	UR_PAUSE
FR_GAME_CAMERA	The gameplay should be from a top down perspective where the player can view their character and see the map	UR_CAMERA
FR_EASY_DIFFICULTY	The game should have a default difficulty which is an easy mode. It should not be difficult for the average person with little gaming experience.	UR_DIFFICULTY
FR_WIN_SCREEN	If the user wins their score will be displayed on the screen at a minimum, and maybe a congratulations message.	UR_ENDING
FR_LOSS_SCREEN	If the user loses the game their score is displayed (if viable) and a message which tells them they have lost	UR_ENDING

ID	Description	User requirements
FR_MAIN_MENU	The main menu should have a 'new game' button that starts a maze, an 'exit' button the closes the game and a 'settings' button that opens the settings screen	UR_MAIN_MENU
FR_SETTINGS	The game should have a settings screen where the user can change volume	UR_MAIN_MENU

Non-Functional Requirements

ID	Description	User Requirements	Fit criteria
NFR_LICENSING	All third party products used should be non-copyrighted and all checks should be certain we are allowed to use them in game	UR_LICENSING	100% of assets have correct licensing
NFR_RUNTIME	The game should not crash or fail to load at any point.	UR_SYSTEMS_REQUIREMENT	Game runs 100% of the time
NFR_SYSTEM_RESTRICTIONS	The game must not be difficult to run and should run on any OS and most machines	UR_SYSTEM_REQUIREMENTS	Game runs on 99% of machines
NFR_AGE_LIMITATIONS	The game should not contain any features that will make it inappropriate to children.	UR_RATE	Game is pegi 12
NFR_GAME_TIME	The game should have enough functionality to run for a minimum of 5 minutes	UR_TIME	Game runs for >=5 minutes
NFR_LANGUAGE	The game should be coded fully in java 17. Libraries can be used but must be java.	UR_PROGRAMMING_LANGUAGE	All code files are .jar files