Requirements

Elicitation

At the beginning of the requirement elicitation process, we decided that creating a professional interview to successfully elicit as many user requirements as possible was our top priority, so we tackled the task as a full team. To develop these questions, we had a meeting as a team to brainstorm ideas as to what kind of questions we should ask the customer. We then proceeded to create categories that we deemed important for eliciting sufficient requirements to develop the game that the customer desires. We developed the following categories: “game concept”, “story and characters”, “levels and progression”, “gameplay mechanics”, “audio and sound”. We decided to keep most of the questions open ended to allow the stakeholder to provide as much information per question as possible. We then decided that only a couple of team members were required to complete the interview. They did so by fleshing out each question, and sorting them into the appropriate categories that we created. They then formatted them in a way that made the interview flow well from question to question to help avoid causing any confusion with the stakeholder. After we completed the interview document, we assigned an interviewer to conduct the interview and two scribes to document the customer’s responses.

We then carried out the interview with our customer, using a voice recording and scribes to document the responses that we received.

Based upon the responses received, we compiled a single statement of need and a set of requirements organised into three different tables. There is a “User Requirements” table and a “Software requirements” table that is further divided into a “Functional Requirements” and "Nonfunctional requirements" table.
Our “User Requirements” table contains tasks which can be performed by the user. Whereas the “Software Requirements” table contains all of the functions of the system required to fulfil a given user requirement. “Functional requirements” describe a system’s actions whereas “Non functional requirements” describe a system’s qualities. This is done in order to give a brief overview of the user’s interaction with the system for non-technical stakeholders. Which is then broken down into a more specific and detailed set of technical requirements which thus makes the implementation process easier for the developers and designers.

Each requirement in the “User Requirements” table has three properties:

- ID: meaningful and can easily be referenced to,
- Description: brief overview of the requirement.
- Priority: (Shall, Should, May) the necessity of implementation

Each functional (and non-functional) requirement, while also having an ID and description does not have a priority. Instead, it has a “User Requirements ID” field which identifies the user requirement it is implementing. Furthermore, each non-functional requirement is assigned a fit criteria to quantify the outcome required to deem the requirement fulfilled.

We also based some requirements off of the product briefing provided at the start of the project.

SSON

“What I want to achieve is that we have a fun, small game that is an engaging way for people to interact with university life or university themed things”

User Requirements

ID	Description	Priority (Shall, Should, May)
UR_FAMILY	Users of all ages should be able to enjoy the game	Shall
UR_TUTORIAL	Users should be taught how to play the game at the beginning	Shall
UR_INTERACT	The user should be able to interact with university themed items and characters	Shall
UR_SETTING	The game should place the user in a university setting	May
UR_SPLAYER	The game should be a singleplayer experience	Shall
UR_MAP	The user should be able to explore a single large maze in the map	Shall
UR_PASSAGE	The user should be able to skip parts of the maze by accessing hidden passages	May
UR_THEME	The game should contain university themed objects and characters	Shall
UR_INPUT	The user should control the character using keyboard and mouse	Shall
UR_SPEED	The user should be able to find a power-up to increase their speed	May
UR_TIME	The user should have 5 minutes to escape the maze	Shall
UR_SCORE	The user should be given a score at the end of the game	Shall
UR_INTERFACE	The user should be able to track their event progress	Shall
UR_GAMEOVER	The user should be reset to the beginning of the maze whenever they fail	Shall
UR_SOUNDS	The game should have sound effects when the user interacts with the	May

	player and environment	
UR_DEVICE	The game should provide a smooth experience on most desktops and laptops	Shall
UR_STORAGE	The game should not take up too much storage	Shall

Software Requirements

Functional Requirements

ID	Description	User Requirements ID
FR_MENU	The game should open to menu allowing the user to select whether to enter the game or the tutorial	UR_TUTORIAL
FR_SHOWTUTORIAL	The game should display the controls to play the game to the user	UR_TUTORIAL
FR_FINISHTUTORIAL	The game will display a popup menu allowing the user to select whether to enter the game or go to the menu	UR_TUTORIAL
FR_DIALOGUE	The game should allow users to press a button to begin a digalog with an NPC	UR_INTERACT
FR_PROMPT	The game should be able to display boxes at the bottom of the screen displaying text when an interaction begins	UR_INTERACT
FR_INCRSCORE	The score should increase when positive events are triggered	UR_SCORE
FR_DECRSCORE	The score should decrease when negative events are triggered	UR_SCORE
FR_RESET	The game displays a game over screen with options to try again or quit	UR_GAMEOVER
FR_MAZE	The maze should be the same for every playthrough	UR_MAP
FR_PROGRESS	The game should keep count of how many of each type of event have been triggered	UR_INTERFACE
FR_BUS	The user should have catch a bus to unlock the next point of the game	UR_THEME

FR_EXAM	The user should have to overcome an exam to unlock the next point of the game	UR_THEME
FR_GUARD	The user should have to avoid campus security guards	UR_THEME
FR_MUTE	The user should be able to mute the sound effects in the game	UR_SOUNDS
FR_BGMUSIC	The game should have background music	UR_SOUNDS
FR_KEYPRESS	The game should be able to run logic upon key presses	UR_INPUT

Non-functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_LAUNCH	The game will launch into the main menu quickly	UR_DEVICE	Launch in <5 seconds on a lab PC
NFR_FRAMERATE	The game will run at a fixed high framerate	UR_DEVICE	Run at a constant 60fps on a lab PC
NFR_LOAD	The game will load without stuttering for too long	UR_DEVICE	The game will load from the menu into the maze in <3 seconds
NFR_CRASH	The game should not crash during a play session	UR_DEVICE	The game will be playable for the full 5 minutes without crashing
NFR_ASSETS	The game should consist of reasonably sized assets	UR_STORAGE	The assets folder will be <1GB
NFR_DIFFICULTY	The game should be beatable by casual players	UR_FAMILY	>90% of players can finish the game in 5 minutes