Requirements 1

Group 2 - Vikingz

Damian Boruch

Tommy Burholt

Elliott Bryce

James Butterfield

Ren Herring

Zhenggao Zhang

Sharlotte Koren

Requirements

Eliciting Requirements

We started by looking at the assessment brief and identifying all the key tasks and objectives that our game must fulfil. Our goal was to make sure that the game met every criterion that was outlined in the brief. From this, we wrote down a list of questions to ask our client which both clarified the brief and allowed us to gain feedback on our own ideas. For example, we were unsure on how the student satisfaction was going to be measured and what devices our game would be running on.

Next, we conducted research into what the best practices are for eliciting and documenting requirements. This led to an exploration of writing a requirements specification - the industry standard for presenting requirements. We decided to split it into three categories - user, functional, and non-functional requirements. Which allows us to easily cover each aspect of the system ensuring a balanced product.

We put ourselves in the shoes of the customer by considering their perspective on what the game should achieve and how it would fulfil the requirements. We went through each point of the brief and brainstormed specific features the game would need to include. For example, we considered things that could potentially be limitations of placing buildings into the map, like placing them too close together or too close to a dangerous place like a lake. This detailed brainstorming allowed us to filter each question into one of the three categories we created, whilst still covering the whole brief.

Now that we had a list of questions, we developed a scenario walking through the game from the user's perspective, visualising how the player would interact with the interface. This helped us outline the game's flow, features and key interactions. During the interview with our prospective client, we presented this walkthrough, to ensure that our requirements aligned. As we described the game's features, he provided us feedback, pointing out areas that needed adjustment. We then refined the requirements based on his input, ensuring that the game aligned with their expectations. For example, we were thinking of including a finance system as a way of adding difficulty to the game. After discussing this with our client, we refined the system so that it didn't include any complexities like working out the number of students in a building to generate specific profits. This was changed to a set rate of income which a building generated.

During the interview we also received a few requirements from our customer such as; making sure that changing the resolution of the game doesn't cause errors, including a basic tutorial and including more accessibility settings such as colour blind support.

The next step was determining how to detail these requirements. After conducting further research, we found that the best approach for our project is to use a single statement of need and then build our user requirements from this broader goal whilst keeping them concise, clear and focused on a specific need. This would make the requirements easy to understand and trace.

Requirement Referencing System

Single Statement of Need
“The system shall enable customers to intuitively play a simulation game of building a university campus, allowing them to place a variety of buildings and receive a satisfaction score”

User Requirements

Requirement ID	Description	Priority
UR_EXPERIENCE	The system shall have a pleasant user experience	Shall
UR_BUILDINGS	The user shall be able to place a variety of different buildings	Shall
UR_ACCESSIBILITY	The user shall be able to use the game without needing any training, and the game shall be playable for users with colour blindness.	Shall
UR_SOUND	The system shall have enjoyable sound but must be able to be muted	Shall
UR_LEADERBOARD	The user will maybe be able to save the game score and see it on a leaderboard.	May
UR_SYSTEM	The user shall be able to play the game in university software labs with no additional hardware	Shall
UR_TIME	The game shall last only 5 minutes	Shall
UR_MAP	The system shall have a fixed size, usable map for all required buildings	Shall
UR_COUNTER	The user shall be able to see a count of the number of buildings they have placed.	Shall

Functional Requirements

Requirement ID	Description	User Requirement
FR_BUILDING_TYPE	Every building should have a single purpose such as teaching, food, etc.	UR_BUILDINGS
FR_CONTROLS	The system shall be operated by using a standard mouse and keyboard	UR_SYSTEM
FR_SYSTEM_REQUIREMENTS	The system shall be usable on a big screen with a software lab pc (integrated intel graphics, w11 and 16gb of ram)	UR_SYSTEM
FR_ACCESSIBILITY	Buttons must have clear descriptions and easy to understand functions	UR_ACCESSIBILITY
FR_SOUND	The system should have a sound system integrated with ability to change the sound files	UR_SOUND
FR_MUTEABLE	There should be a mute button so that the game can be played without any sound playing.	UR_SOUND
FR_SCALING	The system shall be able to scale to small and large screen with low hardware requirements	UR_SYSTEM
FR_TIMER	The system shall be able to keep track of the time of the game and end when the timer has finished	UR_TIME
FR_MAP	The system must support a map that can fit all required buildings for the player to fulfil objectives	UR_MAP
FR_ACCOMODATION_BUILDING	There shall be at least 1 placeable accommodation building	UR_BUILDINGS
FR_LEARNING_BUILDING	There shall be at least 1 placeable building for students to learn in.	UR_BUILDINGS
FR_EATING_BUILDING	There shall be at least 1 placeable building for students to eat in	UR_BUILDINGS
FR_RECREATIONAL_BUILDING	There shall be at least 1 placeable building for recreational activities	UR_BUILDINGS
FR_NO_OVERLAP	The system should not allow buildings to be placed on top of one another, or on squares that are off limits (e.g. in a lake)	UR_BUILDINGS
FR_LEADERBOARD	The system shall maybe have a leaderboard to see previous top scores	UR_LEADERBOARD

Non-functional Requirements

Requirement ID	Description	Appropriate fulfilment of criteria	User Requirement
NFR_EASE_OF_USE	The system shall be operable by A-level students with no training	This criteria is fulfilled if users who have never seen can <70% of the time play it without asking questions.	UR_ACCESSIBILITY
NFR_SAVES	The system shall remember top scores of player playing the game	The criteria is fulfilled if users can access and see the leaderboard of top scores	UR_SAVES
NFR_COLOR	The system shall use colours scheme that wouldn't be confusing to colour blind people	This criteria is met when colour blind users can successfully play the game without needing help	UR_ACCESSIBILITY
NFR_FAST_PLACEMENT	Placed buildings will appear instantly on the map	In <2 seconds after placing	UR_BUILDINGS