Requirements

Group 4 - Cohort 2

Theo Coleman

Matias Duplock

Mohamed Eljak

Katie Schilling

Juliet Urquhart

Nora Wu

As a group, we read through the product brief and extracted the key features we would need to include in the creation of our simulation. In a shared document, we identified and bullet-pointed the key features outlined in the brief and together we began developing ideas around these points. We drafted various plans of how we would implement each of these features and how they could appropriately fulfil the criterion in the brief.

In order to elicit specific requirements, we performed a client interview early on in our development process. While reading through the product brief, we drew questions from any points we wanted to clarify and specify requirements for. We also researched the different types of requirements we would need to define: constraints requirements, functional requirements and non-functional requirements.

We divided these categories between us, and researched what was meant by each type of requirement. This research provided us with guidelines on how to structure our client interview questions and ensure that we covered all relevant areas of the system to effectively engineer our requirements. It also provided us with information relevant to the game development process of our project, as we learnt how each type of requirement should be visible and presented within the system.

As a group, we ensured our interview plan included multiple questions that would address each type of requirement. We then carried out an interview with the client and recorded the answers to our questions. We further developed our plan from these answers, adding more detail and additional features to the existing notes we had. The results of the client interview allowed us to prioritise our requirements, as the answers indicated to us which functions were most desired or actually essential.

We took an agile, developmental approach to negotiating some of our requirements; we found it particularly effective to implement them into the game as originally planned, and then revise how they functioned and performed. Then, if necessary, we would adjust and alter the functions. As we had prioritised our requirements, it was safe for us to remove or combine certain features without reducing the functionality of our system.

As the product brief was generally flexible about requirements (and the results of the client interview maintained this approach), we were allowed some creative freedom when developing our game and this allowed us to continually negotiate and develop the specificity of our requirements and their functionality.

User Requirements

ID	Description	Priority
UR_NAVIGATION	The user should be able to easily navigate the map in a way that provides a satisfying user experience.	Shall
UR_INTUITIVE	The system should be easy for the user to use and understand, with functions and actions being intuitively identifiable.	Shall
UR_ADD_BUILDING	The user should be able to select and add buildings from the menu	Should
UR_INTERACT	The user should be able to interact with the map and the changes they have made to the environment.	Should
UR_BOUNDARIES	The system should provide clear boundaries on what the user can interact with and where they can take action (such as areas of the map and existing buildings).	Shall
UR_VARIABLES	The user should be able to view their user stats and see how their actions in the game affect these values and how these values affect their performance in the game.	Should
UR_TIMER	The user should be able to see a countdown of how long is remaining of the 5 minute timespan of the game.	Should
UR_PAUSE	The user should have the ability to pause the game at any time.	Should
UR_AESTHETICS	The system should have aesthetic visuals and accompanying sounds to make it a more pleasant experience for the user.	Should

Functional Requirements

ID	Description	User Requirements
FR_PLAYER_MOVEMENT	The player shall be able to move around the map to visit different locations easily.	UR_NAVIGATION
FR_INSTRUCTIONS	There shall be clear instructions for every action the user will be able to do.	UR_INTUITIVE
FR_BUILDINGS	The user should be able to select from many different types of buildings to build on the map.	UR_ADD_BUILDINGS

FR_BUILDING_PLACEMENT	The user should be able to place the buildings in almost any location.	UR_INTERACT
FR_ACTIVITIES	The user should be able to perform activities at the placed buildings.	UR_INTERACT
FR_BUILDING_RESTRICTIONS	The user shouldn't be able to place buildings in restricted locations.	UR_BOUNDARIES
FR_MAP_SIZE	The map should be a set size and the user should not be able to traverse or place buildings outside of that boundary.	UR_BOUNDARIES
FR_STATS	The user has stats that are affected by their actions in the game and these stats are displayed on screen.	UR_VARIABLES
FR_TIMER	The game should only last a set amount of time and be visible to the user throughout the whole game.	UR_TIMER
FR_PAUSE_MENU	The user should be able to pause at any time and have options on this pause menu.	UR_PAUSE

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_MAP_TRAVERSAL	The system should quickly respond to the users inputs about traversing the map with little delay.	UR_NAVIGATION	Character movement has little delay between user inputs. (<1 second delay)
NFR_INTUATIVE_CONTROLS	The control schemes should be common ones the user can easily recognise and pick up quickly.	UR_INTUITIVE	Common keyboard inputs used e.g. enter to select, WASD to move.
NFR_INSTRUCTIONS	While the controls should be intuitive, there should also be clear instructions so the user can easily	UR_INTUITIVE	Text prompts on screen.

	pick up how the system works.		
NFR_BUILDING_LIST	The building options should all be visible from the menu.	UR_ADD_BUILDING	The list of all options should be clearly navigable.
NFR_BUILDING_PLACEMENT_TIMING	After the user selects a location to place a building it should immediately become a part of the map.	UR_ADD_BUILDING	Little delay between placing the building and it becoming usable (<1 second).
NFR_MAP_RESTRICTIONS	The system won't let the user traverse outside of the designated map size.	UR_BOUNDARIES	A non-traversable border around the map.
NFR_BUILDING_RESTRICTIONS	The system won't allow buildings to be placed in areas where they are not allowed to be placed.	UR_BOUNDARIES	A grid will show where buildings can be placed and a prompt will show up if the user tries to place a building in a restricted area.
NFR_UPDATE_STATS	The system will update the players stats as the game is played to keep the user informed on their progress.	UR_VARIABLES	Once the user performs an action that changes their stats, it will change the stats displayed on screen.
NFR_GAME_OUTCOME	System saves the players final score	UR_VARIABLES	Saves the final score to a file to compare to other games.
NFR_VISUALS	The system will have appealing visuals.	UR_AESTHETICS	All assets will have a graphical visual representation
NFR_MUSIC	The game will have accompanying music.	UR_AESTHETICS	The game will have background music the user can change the volume of.