ENG 1 - Requirements

Sameer Minhas

David Lun

Harry Muir

Phyo Lin

Alex McBride

Introduction to Requirements Elicitation and Negotiation

Requirements Elicitation and Negotiation are critical phases in the development of any software project, creating the foundations necessary for a system that meets all of the needs of the customer. Requirements elicitation was conducted using techniques such as conducting customer interviews and analysing existing games similar to the one we are building. This process was designed to identify the important features that the game has to have, interactions the player will have with the game and system limitations.

After gathering the initial requirements, negotiation was used to create a balance between different stakeholder priorities. For example making gameplay complex enough to be fun and engaging for the end user but not too complicated that it would take the average player a long time to understand. Negotiation helped with setting clear boundaries for what the game needed to include and what should or may be included (“shall”, “should” and “may” labels for requirement priority).

Presentation of Requirements

The requirements for the game are presented in a structured table format to improve their clarity and traceability. They are split into three categories:

User requirements (UR) Table

These are statements that outline the functions that a user should be able to perform using the proposed system. An example of this would be placing down or moving a building.

The user requirement table has 3 columns to it: **ID**, **Description**, **Priority**. The **ID** is a unique identifier which is prefixed with UR to denote user requirement. The **Description** is a statement regarding tasks that users should be able to carry out using the system. The **Priority** has 3 levels to it: Shall, Should and May. Shall requirements are requirements we definitely need, Should requirements are requirements that we could do without and May requirements are requirements that would be nice to have

Functional requirements (FR) Table

These specify the specific functions or features the system must perform to fulfil user requirements. Each functional requirement relates to a user requirement.

The functional requirements table has 3 columns to it: **ID, Description, User Requirement**. The **ID** is a unique identifier which is prefixed with FR to denote functional requirements. The **Description** represents statements of actions that the system has to take to deliver on the needs of the users. The **User Requirements** is the unique identifier of the User Requirements table that the functional requirement can trace back to.

Non functional requirements (NFR) Table

Non functional requirements are physical qualities that the system needs to have in order to meet the user requirements. These each have a fit criteria which measurable benchmarks that the system must meet in order for the requirement to be satisfied, such as performance,

usability, or reliability. Similarly to the functional requirements, each non-functional requirement references a user requirement.

The non-functional requirements table has 4 columns to it: **ID, Description, User Requirement, Fit Criteria**. The **ID** is a unique identifier which is prefixed with NFR to denote non-functional requirements. The **Description** is statements that represent qualities a system must have. The **User Requirements** is the unique identifier of the User Requirements table that the non-functional requirement can trace back to. The **Fit Criteria** is a statement that quantifies/measures the non-functional requirement.

All of the requirements are formatted with unique IDs to allow for easy referencing.

Over the course of the development of the project, some requirements needed to be altered and some needed to be added. This was due to new stakeholder insights and evolving project needs. An example of this was sound effects. In the initial gathering of requirements, sound effects weren't initially considered. However after feedback user tests were analysed, it became clear that sound effects would significantly improve user immersion. This led us to adding the new requirement.

SSON:

The system will be a single-player game that allows players to design and build their own university campus from scratch, strategically placing buildings to create a fun and intuitive environment that maximises student satisfaction. The game should provide a cohesive and immersive experience to the user.

User requirements:

ID	Description	Priority
UR_CAMPUS_CONSTRUCTION	The player shall be able to construct a university campus from scratch by placing various buildings and activities.	Shall
UR_BUILDING_VARIETY	The player shall be able to place at least one building for each category such as: educational, residential, recreational and dining	Shall
UR_BUILDING_COUNT	The player should be able to see the total number of buildings that have been placed down.	Shall
UR_PAUSE_FUNCTIONALITY	The player should be able to pause the game at any time for convenience.	Shall
UR_LEADERBOARD	The player shall be given a leaderboard with the names and score of the top 5 scores at the end and start of the game	Shall
UR_ACHIEVEMENTS	The player shall be able to obtain achievements during their playthrough of the game	Shall
UR_MOVE_BUILDINGS	The player should be able to move buildings after placing them to improve campus layout or respond to events.	Should
UR_TIME_ELAPSED	The player should be able to see how much time has passed within the game session.	Shall
UR_MUTE	The player should be able to mute the volume of the game at any time.	May

UR_SOUND_EFFECTS	Sounds shall play that will give the player auditory feedback on their actions.	May
UR_BUILDING_RESTRICTIONS	The user will not be able to place buildings outside the bounds of the map, and will have limits to the number they can place by making each building have a “cost”	Shall
UR_REMOVE_BUILDINGS	The player will be able to remove buildings that they place, should they wish to restructure their university campus.	Should

Functional Requirement

ID	Description	User Requirements
FR_DISPLAYED_BUILDINGS	The game will display the different building options the player can choose from. It should provide a description for each building type and what they do.	UR_CAMPUS_CONSTRUCTION
FR_BUILDING_VARIETY	The system will allow the user to place down buildings. These buildings will affect student satisfaction either positively or negatively from its distance to other buildings.	UR_BUILDING_VARIETY
FR_BUILDING_SELECTION	The game will allow the player to select which building type they want to place down.	UR_CAMPUS_CONSTRUCTION
FR_BUILDING_PLACE	The game will allow the player to place a building they have selected onto the map.	UR_CAMPUS_CONSTRUCTION
FR_BUILDING_COUNT	The game will display a counter that increments every time a building is placed down. This counter will be decremented if a building is moved.	UR_BUILDING_COUNT
FR_PAUSE	The game will allow the player to pause the game at their convenience, the player will still be able to interact with their campus but time will not pass.	UR_PAUSE_FUNCTIONALITY
FR_MOVE_BUILDINGS	The game will allow the player to move buildings from their original location to another, appropriate space to respond to events.	UR_MOVE_BUILDINGS
FR_BUILDING_RESTRICTIONS	The game will not allow the player to place buildings in invalid locations, e.g. on other buildings, on map features, outside the bounds of the map.	UR_BUILDING_RESTRICTIONS
FR_MUTE_FUNCTIONALITY	The system will allow the player to mute the game at any time, eliminating all sound from the game.	UR_MUTE
FR_DISPLAY_TIME	The system shall display the amount of time that is left before the game ends.	UR_TIME_ELAPSED
FR_SOUND_EFFECTS	The system may play sound effects to provide auditory feedback to the player when they perform actions.	UR_SOUND_EFFECTS
FR_REMOVE_BUILDINGS	The system shall allow the player to remove or demolish buildings they have previously placed.	UR_REMOVE_BUILDINGS

FR_LEADERBOARD	The leaderboard will be a table that has three columns; username, score and achievements. The system will update the leaderboard if at the end of the game the user has a bigger score than the top 5. The system will ask for a username to save score. The system will make it apparent which score is the user's	UR_LEADERBOARD
FR_ACHIEVEMENTS	The system shall award achievements while the game is being played that can change the final score after the game ends either positively or negatively. The achievements acquired will be shown at the endscreen	UR_ACHIEVEMENTS

Non Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_USABLE_UI	The game shall have an intuitive user interface that allows players to easily navigate and perform actions.	UR_CAMPUS_CONSTRUCTION UR_MOVE_BUILDINGS	The interface shall allow >90% of new players to locate and use primary functions, within <5 minutes of starting the game.
NFR_PAUSE_RELIABILITY	The pause functionality shall be responsive and not cause the game to freeze or crash.	UR_PAUSE_FUNCTIONALITY	The game shall pause or resume <0.1 seconds of the player's command, without causing errors, in 100% of test cases.
NFR_IMMEDIATE_RESPONSIVENESS	The game shall provide immediate feedback to player actions, such as placing, moving, removing buildings or mute/unmute the game	UR_CAMPUS_CONSTRUCTION UR_MOVE_BUILDINGS UR_REMOVE_BUILDINGS	The system shall respond to player inputs <0.1 seconds, ensuring seamless interaction.
NFR_PERFORMANCE_SCALABILITY	The game shall maintain performance standards even as the number of buildings increases.	UR_CAMPUS_CONSTRUCTION UR_BUILDING_COUNT	The game shall maintain >30 FPS when up to 100 buildings are placed, on systems that meet minimum specifications.
NFR_STABILITY_RELIABILITY	The game shall operate reliably without crashes during building placement, movement, or removal.	UR_CAMPUS_CONSTRUCTION UR_MOVE_BUILDINGS UR_REMOVE_BUILDINGS	The game shall not crash during >95% of 5-minute game sessions in testing over 5 hours of gameplay.
NFR_DISPLAY_READABILITY	The time elapsed and building count displays shall be easily readable and will update in real time.	UR_TIME_ELAPSED UR_BUILDING_COUNT	Displays shall update within <0.1 seconds of changes, with a readable text size on a standard 1920x1080 resolution.
NFR_PLATFORM_COMPATIBILITY	The game shall be compatible with commonly used operating systems	UR_CAMPUS_CONSTRUCTION	Full functionality on Windows 10 and above, macOS 10.13 and above, and Ubuntu 18.04 LTS and above.
NFR_TRANSITION_LOAD_TIMES	The game shall have minimal load times when starting up or transitioning between screens.	UR_CAMPUS_CONSTRUCTION	Loading screens shall not exceed 5 seconds on systems that meet minimum hardware requirements.