Requirements

Group 8: GeNext

Aditya Mydeo
Alex Miles-Thom
Christopher Omoraka
Faisal Nabi
Jesly Prosper
Stefan Maxwell
Zak McGuire

To develop our game, we ensured that our requirements were elicited and recorded in a structured way using the IEEE 29148-2018 standard. We used a collaborative approach amongst our team to devise questions we could ask our key customer- Tommy Yuan. These questions focused on understanding the users' needs, objectives of the game and the desired gameplay mechanics. Additionally, we made sure to seek clarification on the product brief by asking any questions about the areas of confusion to fully understand our objectives.

We decided to use a semi-structured interview format and audio record the session (ensuring a consent form was filled out) so that we could transcribe the information we gathered on our user expectations, accessibility, functionality, and any other stakeholder requirements. This approach encouraged us to follow a structured conversation while also allowing us to ask any follow-up questions to develop the depth of our understanding and gather rich qualitative data.

We prioritised feedback from the stakeholder when negotiating the requirements which involved a thorough discussion between us and them, ensuring that our game would align with both the user desires and any practical constraints we had. This highlighted the importance of clear communication and flexibility allowing adjustment based on any stakeholder input and technical feasibility.
To present these requirements we used the IEEE 29148-2018 standard for requirements specification. Following this standard allows us to use well-structured and defined tables that categorise our requirements into 3 types: User Requirements, Functional Requirements, and Non-Functional Requirements. Organising our requirements this way ensures clarity and traceability throughout the software development cycle so that the architecture is clear, and we can identify which requirements satisfy specific user needs. Each requirement is numbered and appropriately named, providing a systematic framework that presents the requirements. Additionally, a fit criterion is provided for the non-functional requirements to provide a measurement to how this requirement can be achieved. The presentation of requirements like this facilitates ongoing conversations with key stakeholders regarding any changes and provides an easy way to update our requirements for any subsequent assessment deliverables.

Single Statement of Needs:
" The user should be able to play a simulation game of university life, receiving a satisfaction score at the end of 5 minutes"

User Requirements:

ID	Description	Priority
UR_CAMPUS_MAP	The player can access provided map that shows explicit boundaries	Shall
UR_BUILDING_TYPE	The system shall allow placement of various buildings of different sizes. functionalities	shall
UR_PAUSE_GAME	User can pause game at any point and be able to view map but not interact with campus	Shall
UR_PLAY_GAME	User can un-pause/resume game and interact with map	Shall
UR_GAME_TIMER	A timer of 5 minutes should be displayed showing remaining time left	Shall
UR_BUILDING_COUNTER	User can see a how many of each building types have been placed	Shall
UR_SATISFACTION_SCORE	System should display satisfaction score to user	Should

UR_REACT_TO_EVENTS	Users can react and respond to events changing satisfaction score	Should
UR_RESTART	Users can restart/ create a new campus simulation	May
UR_INSTRUCTIONS	Game should provide instructions before starting and can be access at any point	Should
UR_COLOUR_CHANGE	User can change the display colour to address colour blindness	May
UR_USABILITY	The game should be playable for all players despite previous gaming knowledge	Shall
UR_SCREEN_ADAPTABILITY	The game should be able to play on various screen sizes	Should
UR_SOUND	Users can hear sounds relating to the game to enhance user experience	May
UR_BUDGET	Users can see budget for their campus and see how much they have spent	Should

Functional Requirements:

ID	Description	User Requirements
FR_GAME_BOUNDARIES	The system provides feedback when a building is being placed in an out of bounds area	UR_CAMPUS_MAP
FR_DRAG_AND_DROP	System must include drag and drop functionality for placing all types of buildings	UR_BUILDING_TYPE
FR_BUILDING	System provides at least 4 building types: learning, sleeping, eating and recreational	UR_BUILDING_TYPE
FR_BUILDING_NAMES	Users can distinguish between the 4 different building types by seeing the name of the building on it	UR_BUILDING_TYPE
FR_PAUSE_FUNCTION_BUTTON	Players can click on pause button at any point which stops the timer and any interactions	UR_PAUSE_GAME
FR_MAP_VIEW_ON_PAUSE	System allows user to navigate across map whilst on pause	UR_PAUSE_GAME
FR_RESUME_GAME	System lets user to unpause game, restoring full interactions with campus and timer	UR_PLAY_GAME
FR_DISPLAY_TIMER	system displays a countdown from 5 minutes in minutes and seconds	UR_GAME_TIMER
FR_TIMER_WARNING	The system provides a visual warning when only 1-minute remains	UR_GAME_TIMER

FR_BUILDING_COUNTER_DISPLAY	Systems displays counter indicating number of each building type placed on campus in real time	UR_BUILDING_COUNTER
FR_DISPLAY_SATISFACTION	System displays satisfaction score using a bar, updating based on user actions to events	UR_SATISFACTION_SCORE
FR_EVENT_REACTION	System allows users to respond to events by relocating, moving or adding buildings or other items	UR_REACT_TO_EVENTS
FR_EVENT_ANNOUNCEMENT	All events announced to user so that they can plan and react accordingly	UR_REACT_TO_EVENTS
FR_EVENTS	During the 5 minutes there is at least 3 events that are: positive, negative and neutral	UR_REACT_TO_EVENTS
FR_RESTART_GAME	System shall provide an option to restart (as a button) when game is over and sets the game to the start	UR_RESTART
FR_GAME_INSTRUCTIONS	Systems visually displays instructions before game begins in a couple sentences	UR_INSTRUCTIONS
FR_HELP_DURING_GAME	System allows user to access instructions during gameplay via a button	UR_INSTRUCTIONS
FR_COLOUR_ADJUSTMENT	System allows users to select black and white mode via a button	UR_COLOUR_CHANGE
FR_NON-TECHNICAL_JARGON	System shall contain no technical jargon in instructions	UR_USABILITY
FR_INTUATIVE_CONTROLS	System will be very intuitive, and no previous gaming knowledge is needed	UR_USABILITY
FR_SCREEN_RESOLUTION	System will provide similar resolution on all displays sizes, so user experience is the same	UR_SCREEN_ADAPTABILITY
FR_SOUND_EFFECTS	System includes sound effects relating to game actions and indicating events	UR_SOUND
FR_VOLUME_CONTROL	System allows users to adjust volume or mute game via buttons	UR_SOUND
FR_BUDGET_DISPLAY	System will display total budget at top of screen and track expenditures in real time	UR_BUDGET
FR_BUDGET_WARNING	System will alert user when budget is too low and prevent further purchasing of buildings	UR_BUDGET

Non-functional requirements:

ID	Description	User Requirements	Fit Criteria
NFR_AVAILIBILTY	The system shall be highly available on supported devices	UR_SCREEN_ADAPTABILITY	Uptime: 99% on compatible devices
NFR_RESPONSIVENESS	Game actions shall have minimal delays when clicking buttons or dragging items	UR_BUILDING_TYPE	Actions processed within 1 second
NFR_INSTRUCTIONS_CLARITY	System has clear,easy to understand instructions for all user levels	UR_INSTRUCTIONS	Instructions read 50 cm away and contain no technical jargoon
NFR_SOUND_QUALTIY	System provides high quality sound effect to enhance user immersion	UR_SOUND	Audio sounds are at least 120 kbps bitrate
NFR_USABILITY	System shall be intuitive for all players, even with no gaming experience	UR_USABILTIY	90% of users fully complete the game
NFR_SCALABILITY	System shall support increased functionality as the game progressive	UR_BUILDING_COUNTER	Support for up to 15 buildings without performance lags and counter updates within 1 second
NFR_PAUSE_FUNCTIONALITY	Pause functions works seemingless and stops timer	UR_PAUSE_GAME	Responds within 2 second