Requirements

Assessment 2 - Updated from Team 21

Hannah Thompson
Kyla Kirilov
Ben Hayter-Dalgliesh
Matthew Graham
Callum MacDonald
Chak Chiu Tsang

Stakeholders:
Direct stakeholder(s): Tommy Yuan, Daniel Bethell
Indirect stakeholder(s): Future players, university students

SSON: The game will let you move around the university and simulate preparing for your avatars exams while making sure you don't run out of energy and scoring points to pass your exams.

Introduction:

As a group, we came up with our first few user requirements in the first two sessions before our customer discussion, using the Product Brief to come up with the more obvious requirements such as UR_STUDY and UR_MAP. Then in the next week we had our customer discussion with Tommy Yuan, where we had most of our questions answered, which we finished putting into our requirements traceability tables in the coming sessions, before having to trim them down to fit into the 3 page requirement.

Requirements referencing system used [1, p.102-115]:
FR = functional requirements
UR = user requirements
NFR = non functional user requirements

This ensures that anyone involved in the project can quickly reference a requirement making it easier and clearer in the project documentation. The elicitation requirements process led to us creating a set of requirements that fit the stakeholders requirements and considerations of the product brief/game design. The requirements presented are the product of research and communication with the team to deliver a suitable and entertaining game.

Added for Assessment 2:

SSON: The game will let the player move an avatar around a 2D university map to simulate a student preparing for exam week - the player will complete activities while making sure they don't run out of energy and score enough points to pass their exams.

The requirements for this project are presented using a natural language specification [1, p. 96-97] because this can be universally understood by the team, stakeholders and clients with any level of technical knowledge. To ensure a standard format for requirements presentation, the information is arranged into a tabular format, consisting of a unique identifier and description for each requirement. Other columns in each table are:

- The User Requirements table (1.1) contains a column specifying the degree of necessity [2] to help us rank the requirements when it comes to implementing them.
- The Functional Requirements table (1.2) contains a column linking each functional requirement to the user requirement which encompasses it.
- The Non-Functional Requirements table (1.3) contains a column quantifying the performance constraints of each requirement using measurable, verifiable terms [2].

1.1 User Requirements Table:

ID	Description	Priority
UR_RUNTIME	To have an average run time of around 5-10 minutes.	Shall
UR_MOVEMENT	**To be able to move the avatar around the screen.**	Shall
UR_INTERACT	To interact with objects and locations on the world map.	Shall
UR_INTERFACE	**The game has a simple interface that displays all the key information clearly.**	Shall
UR_ENERGY	An energy bar is displayed on the screen	Shall
UR_REST	To be able to go to at least one location in the game to rest to restore energy.	Shall
UR_MAP	To have a map of the digital campus to move around in.	Shall
UR_RECREATION	To have at least 3 places to perform the recreational activity task at.	Shall
UR_TUTORIAL	The user is taught the main gameplay loop quickly and efficiently.	Shall
UR_STUDY	To be able to go to at least one location that allows the user to study and score points at least once per day.	Shall
UR_EAT	To be able to go to at least one location on the map that allows the user to eat.	Shall
UR_DAYS	The game is split into 7 days that each end when the user has rested. The game will finish after the 7th day.	Shall
UR_GAME_END	**Once the game is finished the user is shown a screen that displays the score, leaderboard and any achievements.**	Shall
UR_ACHIEVEMENTS	**Certain 'hidden' achievements are displayed that grant the player additional points. These achievements are based on their activities, and will display on the game end screen.**	**Shall**
UR_SCORE_LEADERBOARD	**Show a leaderboard on the 'end' screen that displays the top 10 scores and associated player names.**	**Shall**
UR_SOUND_EFFECTS	To have appropriate sound effects to add to the immersion of the game.	Should
UR_GRAPHICS	Have graphics that show core functionalities of the game and look nice.	should
UR_SCORE	A score counter that will decide if the user has passed their exams.	Shall
UR_INTRO	Users should be given a short intro for the context of the game to help immersion, **including controls and a description of the aims.**	Should
UR_CLOCK	Have a visible clock that shows the in-game time and a countdown of the days.	May
UR_DAY_NIGHT_CY	Have a visible day and night cycle throughout all days in the game.	Shall

CLE		
UR_ANIMATIONS	To have animations attached to each interactable object in the game.	Should
UR_READABILITY	To make the game readable for people, taking into account factors like colorblindness.	Should
UR_DIFFICULTY	To not have the game be too difficult to complete eg. be able to win at least twice.	Shall
UR_DAILY_DIFFICULTY	To have the difficulty scale **increase across** the 7 days of gameplay.	Should
UR_MINIGAMES	**Have interactive minigames that represent eating, recreation and studying.**	**Should**

1.2 Functional Requirements Table:

ID	Description	User Requirements
FR_SCALABLE	**The game scales to different screen resolutions and has a full-screen option.**	**UR_INTERFACE**
FR_START_GAME	**When the game is first opened, the player is presented with a screen that shows the objective, current leaderboard, any instructions, as well as controls. There is a START GAME button that closes this menu and opens the main game.**	**UR_INTERFACE** **UR_TUTORIAL** **UR_INTRO**
FR_LEADERBOARD	**The 10 highest scores and player names are saved in an external file that is loaded each time the game is opened.**	**UR_LEADERBOARD**
FR_ENTER_PLAYER_NAME	**The player is able to input their name within a fixed set of characters to save to the leaderboard.**	**UR_LEADERBOARD**
FR_AVATAR_MOVEMENT	There is an avatar through which the user can interact with the game. **The avatar is moved around the screen using the WASD keys.**	UR_AVATAR **UR_INTERACT** **UR_MOVEMENT**
FR_WORLD_MAP	World map with icons of places and things the user can interact with **to complete tasks.**	UR_MAP **UR_INTERACT**
FR_ENERGY_BAR	An energy bar that changes depending on the energy level of the avatar. **This has a maximum of 100% and a minimum of 0%, at which point the user can't perform any more activities.**	UR_ENERGY **UR_INTERACT**
FR_INTERACTION	To be able to interact with the locations around the map **through collisions. These locations will allow the player to complete activities.**	UR_INTERACT **UR_MAP** **UR_EAT** **UR_SLEEP** **UR_RECREATION** **UR_STUDY**
FR_HIGHSCORE_TABLE	There is a table that saves the **top 10 scores** recorded and displays them to the user **in the game end screen.**	UR_SCORE_LEADERBOARD **UR_GAME_END**
FR_EATING	To be able to do the eating activity when interacting with the appropriate location.	UR_EAT **UR_MAP** **UR_INTERACT**
FR_RECREATION	To be able to do any recreation activity when interacting with the appropriate location.	UR_RECREATION **UR_MAP** **UR_INTERACT**
FR_RESTING	To be able to do the rest activity when interacting with the appropriate location.	UR_REST **UR_MAP** **UR_INTERACT**

FR_STUDYING	To be able to do the study activity when interacting with the appropriate location.	UR_STUDY **UR_MAP** **UR_INTERACT**
FR_SOUND_EFFECTS	To play different sound effects when different actions are taken by the user.	UR_SOUND_EFFECTS
FR_SCORE_COUNTER	Score counter is changed based on the action completed by the user and is updated during playthrough. **In assessment 2, this is used to calculate the final score and doesn't need to be shown on the screen.**	UR_SCORE
FR_ACHIEVEMENT_TRACKER	**There are a set of defined achievements that are not known to the player which are tracked during gameplay. These achievements are completed when the player completes certain tasks. In the game end screen, the achievements are displayed and a set number of points is added to the score for each achievement made.**	**UR_GAME_END**
FR_GAME_ENDING	To have an ending to the game which results in showing your score, the leaderboard and any hidden achievements.	UR_GAME_END

1.3 Non-Functional Requirements Table:

ID	Description	User Requirement	Fit Criteria
NFR_JAVA_VERSION	**The game is programmed using Java version 11.**		**The system is fully implemented in Java version 11.**
NFR_TIME	Game shouldn't be too long	UR_RUNTIME	Game should last between 5-10 minutes
NFR_UPTIME	The game should available to play with minimal crashes	UR_INTERFACE	Uptime: The game should not crash more than once during 3 playthroughs.
NFR_COMPATIBILITY	**The system runs on Linux, Windows and MacOS.**		**The system has full functionality on both Linux and Windows operating systems. The system has a minimum of 75% functionality on MacOS systems.**
NFR_PERFORMANCE	**The game appears to respond immediately to user input.**	UR_INTERFACE	**Keypresses are registered and acted upon within 10 milliseconds of input.**
NFR_COMPLIANCE	**The system conforms to the current industry regulated standards relevant to its domain.**		**Regulatory information is documented.**
NFR_EFFICIENCY	**The system minimises CPU and resource use while running to optimise performance.**	UR_INTERFACE	**The average response time for all user interactions is less than 10 milliseconds under typical operating conditions.**
NFR_SIZE	**The size of the end system is kept as small as possible such that it can run on the majority of devices.**		**The program experiences no memory-related crashes.**
NFR_BUILD_TIME	**The build time of the project is minimised during development.**	UR_INTERFACE	**The project builds in a maximum of 20 seconds.**
NFR_SIMPLICITY	The game should be able to pick up the gameplay loop quickly	UR_TUTORIAL	Once the tutorial is done the user should understand the gameplay
NFR_FUN	The game should be enjoyable for the player		Case by case with interacting with testers
NFR_DIFFERENT_STUDY	Give the player more score for studying in multiple places over the various days.	UR_STUDY UR_SCORE_COUNTER	Give the player 10% more score per different location studied at.

NFR_ENERGY_BAR	A representation of the amount of energy the player	UR_ENERGY_BAR	A bar that shows a % of the energy that the student has left.

References:

[1] I. Sommerville, “Requirements engineering”, in *Software engineering*, 10th ed. Boston, Mass. Amsterdam Cape Town Pearson Education Limited, 2016,

[2] "IEEE Recommended Practice for Software Requirements Specifications," in IEEE Std 830-1998 , vol., no., pp.1-40, 20 Oct. 1998, doi: 10.1109/IEEESTD.1998.88286. pp. 7