Introduction

The process of developing the game begins with a thorough understanding of the user's needs and translating those needs into system requirements. In our project, we initiated this process by conducting brainstorm sessions to elicit user requirements and then proceeded to establish system requirements based on the product brief. This introduction outlines how these requirements were elicited, negotiated, and presented, drawing on research into requirements specification and presentation methodologies.

Initially, we identified four main markers to guide our requirement elicitation process: Purpose (A single statement of need that defines the general purpose the software), Intended Audience (A consideration of the end user of the software), Intended Use (A consideration of how the software will be used by the end user), and Product Scope (The benefits and goals of the software for the end user). These markers provided a framework for understanding the overarching goals of the software and its intended users. However, some requirements remained unclear, necessitating further clarification through discussions with our client via an in-person client meeting. This allowed us to create more accurate and specific requirements that were in line with the clients expectations, as well as generate some new requirements that were not initially clear to us.

After discussion with the group and the client, these markers were decided to be: Purpose - The game shall provide a fun, single-player experience of a university student preparing for their final exams, Intended Audience - A current or prospective university student, Intended Use - To provide a fun and entertaining game that can be shown at university open days, and Product Scope - To provide an enjoyable game experience for the user and demonstrate the basics of university student life.

The new requirements were firstly added to a table of ‘ideas’ before being discussed as a group. We then confirmed the requirements, or in some cases proposed re-wordings or clarifications of the requirements, to ensure we were satisfied with the final requirements. They were then added to their respective requirement tables, based on their category as explained below.

The requirements are presented in 2 categories, User Requirements - Non-technical statements outlining how the user will interact with the game, and System Requirements - More detailed, technical descriptions of how the game will be built to satisfy the User Requirements. The system requirements are then split into two further categories, functional and non-functional. The functional requirements relate to the features the game must provide for the end user i.e. the ability to move around the map, the game to end after 7 days etc. The non-functional requirements are instead reserved for the qualities the game must have, i.e. enjoyable to play, or for the constraints on functional requirements i.e. the player must not be able to move outside of the playable map area.

Finally, we outlined some system-wide constraints for the project. These involve a mix of project constraints- limitations or requirements on the development of the project, and design constraints- constraints on the compatibility and creation of the game. These were tabulated and added after our requirements tables. This approach to requirements elicitation and presentation ensures that the software development process remains focused on meeting user needs while also considering technical constraints and quality attributes essential for a successful product.

Requirements Elicitation

I. User Requirements Table

ID	Description	Priority
UR_MOVEMENT	The user should be able to move their avatar around the games map by keyboard input	Shall
UR_INTERACTION	The user should be able to interact and perform activities at different locations on the map	Shall
UR_RESTRICTED_INTERACTION	The user shall not be able to perform activities if they have low energy/resource metre	Shall
UR_RESTRICTED_MOVEMENT	The user shall not move outside the map	Shall
UR_CUSTOMISATION	The user should be able to customise their avatar or game settings	Could
UR_INTUITIVE_CONTROLS	The user should find the game controls intuitive and easy to use	Should
UR_CLEAR_FEEDBACK	The game shall provide clear feedback to the user for their actions and progress as well as summaries at the end of each day	Should
UR_SAVE_PROGRESS	The user should not be able to save the game progress, requiring completion in one session	Should
UR_IN_GAME_HELP	The game shall provide in-game help or instructions for new players	Should
UR_END_GAME	After 7 days the game must end and the player should be given their score.	Should
UR_PARENT_FRIENDLY	The game content should be suitable for parents	Could
UR_ENTERTAINMENT	The game shall provide entertainment and familiarity with the campus layout	Should
UR_AUDIO	The game may include audio with an option to mute	Could
UR_REPLAY_VALUE	The game shall offer replay value with multiple outcomes and scores based on different interactions	Could
UR_COMPATIBILITY	The game should be playable on windows desktop. Stay away from a browser-based game.	Should

System Requirements

I. Functional Requirements Table

ID	Description	User Requirements
FR_MAP	The game should have a map that the user can move around	UR_MOVEMENT
FR_MULTIPLE_LOCATIONS	The game shall have places for study activities, rest, eating, leisure/recreational activities, and etc	UR_INTERACTION
FR_TRACK_STATISTICS	The game shall track and display statistics such as time, study, energy, personal time, activities, etc	UR_CLEAR_FEEDBACK
FR_STRESS_CALCULATION	The game shall use a stress meter for overall score calculation	UR_CLEAR_FEEDBACK
FR_DAY_NIGHT_CYCLE	The game shall feature a dynamic day-night cycle affecting availability of activities and resources	UR_INTERACTION, UR_RESTRICTED_INTERACTION
FR_NPC_INTERACTION	The game shall allow for interaction with non-player characters (NPCs) for quests, information, or rewards	UR_INTERACTION, UR_RESTRICTED_INTERACTION
FR_OUTCOMES	Interactions may have multiple outcomes for increased randomness	UR_INTERACTION
FR_RANDOM_EVENTS	The game shall include random events or challenges	UR_INTERACTION, UR_ENTERTAINMENT, UR_REPLAY_VALUE
FR_STUDY_CHOICES	The game shall allow users to choose how they want to study, with options and possible quizzes	UR_INTERACTION
FR_AVATAR_CHOICES	The game may offer multiple avatar choices	UR_CUSTOMIZATION

II. Non-functional Requirements Table

ID	Description	User Requirements	Fit Criteria
NFR_PERFORMANCE	The game should run smoothly with minimal lag and	UR_ENTERTAINMENT	The game shall maintain a consistent frame rate of at least 30 frames per second on recommended hardware.
NFR_UX	The game should offer a pleasant user interface	UR_INTUITIVE_CONTROLS	The game should have elements that are easy on the eyes and good readability
NFR_COMPATIBILITY	The game shall be compatible with multiple platforms	UR_COMPATIBILITY	The game shall be playable on Windows.
NFR_PARENT_FRIENDLY	The game content shall be suitable for a family-friendly audience	UR_PARENT_FRIENDLY	The game content shall not contain inappropriate or mature themes.
NFR_GAME_LENGTH	The game session duration shall be between 10 to 20 minutes	UR_SAVE_PROGRESS	The average game session duration shall be within the specified time range.
NFR_PLAYER_GUIDANCE	The game shall provide clear instructions and guidance for gameplay	UR_IN_GAME_HELP	The game shall include tutorials or tooltips to assist players in understanding game mechanics.
NFR_AUDIO_OPTIONS	The game shall provide audio options with a volume control	UR_AUDIO	The game shall include volume controls for adjusting audio levels.
NFR_ENGAGEMENT	The game shall maintain player engagement and interest through varied interactions and outcomes	UR_REPLAY_VALUE	Players shall report high levels of enjoyment and replayability based on diverse interactions and outcomes.

Bibliography

Gerhard Krüger, “How to Write a Software Requirements Specification (SRS Document) | Perforce Software,” *Perforce Software*, 2018. https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=Back%20to%20top-,What%20Is%20a%20Software%20Requirements%20Specification%20(SRS)%20Document%3F,stakeholders%20(business%2C%20users). (accessed Mar. 19, 2024).