UNIVERSITY OF YORK

DEPARTMENT OF COMPUTER SCIENCE

Requirements

Cohort 2 - Group 17 (Rich Tea-m 17)

Group Members:

George Jopson
Ben Slater
Meg Tierney
William Potts
Jamie Burgess
Seyi Towolawi
Alex Staicu

Eliciting Requirements

Introduction Requirements were elicited and negotiated through the product brief and an initial client meeting. A product brief document was provided at the start of the project. This set out the story of the game and the interactions that the user must complete. It specified the control scheme of the game and the timescale for the play. Primary objectives for the users were laid out which were used to create user requirements. Items the game must include were set out which created a basis for requirements. How to successfully win the game was also defined which gave an initial idea of player goals which was important as players of the game will be stakeholders. Other stakeholders identified in the document were the customer (the main stakeholder) and the remainder of the cohort (who will be deciding whether to continue the project). This was used as a starting point to prepare a list of questions to take to the client and ask in order to get a better understanding of their aims and preferences for the project.

The list of questions asked in the meeting can be found on the project website. Questions were split into topics to allow for in-depth discussion and follow-up questions were asked as they were thought of in the meeting. The client meeting crucially gave an insight into who the project was targeted towards and what the aim of the project was. It also allowed for features to be assigned priorities and made clear exactly what was and wasn't wanted within the project. The final question asked for any additional requirements that hadn't been discussed already to ensure that nothing had been missed. This allowed a single statement of need to be formed: "The system shall enable users to play a game based on the life of a university student in which they have interactions that influence their score".

User requirements and functional and non-functional requirements are discussed later in this document. They are presented through three tables - user requirements, functional system requirements and non-functional system requirements. Requirements were specified and presented by adapting the guidance given in IEEE 29148-2018 [1]. First, the stakeholder needs and goals as established in the customer meeting were refined to create user requirements. Following this, functional and non-functional requirements were established. To ensure they were well-formed requirements each functional requirement was a requirement that shall be met or possessed by the system to solve the problem and each non-functional requirement is possible to qualify by specific measurable conditions. Wording used followed the conventions specified. A referencing system was used to maintain traceability of all types of requirements. Requirements did not include design decisions or implementation ideas or suggestions. In addition to these requirements, there were also a few constraint requirements. One was the project constraint of the timeline with the deadline for the first part of this project being non-negotiable and due by 21st March.

In the second part of the project, we decided to rename a few requirements as we felt they were not clear enough on what they were referring to at a glance, and we wanted to make the requirements as easy to read as possible for stakeholders and users who may be unfamiliar with technical jargon. Therefore, requirements that were numbered (e.g. FR-GAME-PLAY1, FR-GAME-PLAY2, etc.) were changed to more reflect what that requirement was about. The changes to the requirements have been marked as follows:

> * Renamed requirements, given alongside the original ID for the requirement, in brackets, in order for the team to keep these requirements consistent during the changeover.
> ** New requirements
> *** Updated requirement descriptions

The second part of the project also included new features requested by the stakeholders, being the implementation of Streaks (with corresponding achievements) and Leaderboards. We decided not to conduct another interview to gather more information about these features due to time constraints.

User Requirements

ID	Description	Priority
UR-DEVICE	The game shall be playable on a desktop/laptop.	Shall
UR-MENU	The game shall provide the user a main menu which they can use to navigate to starting a new game, accessing settings, pausing games, etc.***	Shall
UR-CUSTOMISE	The user should be able to personalise their in game character.	Should
UR-WORLD	The user shall move their character around a 2D map, appropriately representative of Heslington - The user shall recognise that the map represents Heslington.	Shall
UR-INTERACT	The user's character shall interact with objects/buildings within the world to complete tasks.	Shall
UR-TIMED	The game shall be timed so that the user plays through a sped up version of a week's worth of university life of the in-game student character. The game shall inform the user of the current in-game time.	Shall
UR-ENERGY* (UR-INFO)	The user shall be informed of their character's energy levels.	Shall
UR-SOUND	The game may have music and sound effects.	May
UR-SETTINGS	The user shall be able to access music and sfx settings in the menu.***	Shall
UR-SLEEP	The user shall replenish their character's energy levels by sleeping.	Shall
UR-ACCESSIBLE	The game shall cater to users that may have accessibility issues (e.g. hearing difficulties, colour-blindness) by ensuring aspects are not reliant on colour, sound, etc.***	Shall
UR-DESIGN	The game shall have a happy/positive aesthetic and vibe.	Shall
UR-LEADERBOARD **	The user shall be able to save/see their own and other user's scores on a leaderboard.	Shall
UR-STREAKS **	The user shall be able to gain secret achievements throughout the game by completing actions repeatedly.	Shall

Functional Requirements

ID	Description	User

		Requirement(s)
FR-VIEW	The system shall always use a top-down view point in the third-person	UR-WORLD
FR-START	Starting the game shall allow the user to choose from a range of avatars.	UR-CUSTOMISE
FR-STAY-OUTSIDE * (FR-INTERACT1)	When a player interacts with a building, they shall stay outside the building.	UR-INTERACT
FR-CHOICE-POPU P* (FR-INTERACT2)	When a player starts to interact with a building, there shall be a pop-up with text and choices.	UR-INTERACT
FR-OPTIONS* (FR-MENU1, FR-MENU4)	Going to the main menu shall give the user the choice of credits, start game, exit game and audio settings.	UR-MENU
FR-SAVED-DATA* (FR-MENU2)	The player shall save their score to the leaderboard once the game ends. ***	UR-MENU, UR-LEADERBOARD
FR-PAUSE-MENU * (FR-MENU3)	Pressing 'esc' shall pause the game and navigate to a pop-up menu with options to resume, navigate to settings or exit.	UR-MENU
FR-NAVIGATE	The user using the arrow keys shall cause the player to navigate around the map.	UR-WORLD
FR-DAY-END* (FR-SLEEP1)	Reaching the end of the day (16 hours) shall lock all other activities other than sleeping.	UR-INTERACT, UR-TIMED
FR-ENERGY-LOSS * (FR-ENERGY1)	A player completing an interaction shall deplete their energy by a set amount.	UR-INTERACT, UR-ENERGY
FR-NO-ENERGY* (FR-ENERGY2, FR-SLEEP2)	The game shall never allow players to continue with activities other than sleeping once they have no energy left.	UR-INTERACT
FR-WEEK	The game shall end after a week.	UR-TIMED
FR-TIME	A player completing an interaction shall jump the time along by a set amount.	UR-TIMED
FR-DEVICE	The game shall only run on a Windows OS.***	UR-DEVICE
FR-SLEEP-LOCATI ON*	The player shall interact with one sleep location.	UR-INTERACT

(FR-GAME-PLAY1)		
FR-STUDY-LOCATION* FR-GAME-PLAY2)	The player can interact with two study locations and make choices at this location.	UR-INTERACT
FR-EATING-LOCATION* (FR-GAME-PLAY3)	The player shall interact with two eating locations and make choices at these locations.	UR-INTERACT
FR-LEISURE-LOCATION* (FR-GAME-PLAY4)	The player shall interact with three leisure locations and make choices at this location.***	UR-INTERACT
FR-COUNTER	The system shall count the number of times the player completes an activity to display at the end of a game.***	UR-ENERGY
FR-LEADERBOARD-DISPLAY**	Top 10 scores from previous playthroughs will be displayed.	UR-LEADERBOARD
FR-HIGH-SCORE**	The system shall update the leaderboard if a new high score is achieved.	UR-LEADERBOARD
FR-STREAK**	The system shall allow the player to earn streaks through completing actions repeatedly.	UR_MENU
FR-STREAK-FLOWERS**	The player will earn a streak by smelling the flowers on campus 5 days in a row.	UR-STREAK, UR-INTERACT
FR-STREAK-STUDY**	The player will earn a streak by studying for their exams 5 days in a row.	UR-STREAK, UR-INTERACT
FR-STREAK-EATING**	The player will earn a streak by eating 3 meals a day (at the correct times) 3 days in a row	UR-STREAK, UR-INTERACT
FR-STREAK-TOWN**	The player will earn a streak by going to town 5 days in a row.	UR-STREAK, UR-INTERACT
FR-STREAK-SHOP**	The player will earn a streak by going to the shop (and purchasing something) 5 days in a row.	UR-STREAK. UR-INTERACT
FR-STREAK-LIBRARY**	The player will earn a streak by going to the library 5 days in a row.	UR-STREAK. UR-INTERACT
FR-STREAK-NIGHT-OWL**	The player will earn a streak by being awake between the hours of 20:00 and 00:00 5 days in a row.	UR-STREAK. UR-TIMED

FR-STREAK-EARLY-BIRD**	The player will earn a streak by being awake between the hours of 00:00 and 08:00 5 days in a row.	UR-STREAK. UR-TIMED
FR-STREAK-ACTIVE**	The system will only count actions towards a streak if the action is performed on consecutive days rather than all in one day.	UR-STREAK, UR-INTERACT
FR-ACHIEVEMENT-UNLOCK**	The system shall award the player achievements if they successfully complete a streak.	UR-STREAK
FR-ACHIEVEMENT-DISPLAY**	The system shall display all achievements gained by the player at the end of a playthrough.	UR-STREAK
FR-SCORE**	The system will keep a score for the player, this will increase when the player completes an activity.	UR-MENU
FR-END-SCREEN**	The system shall display the score of a player and how many times they completed an activity at the end of the game.	UR-MENU

Non-Functional Requirements

ID	Description	User Requirement(s)	Fit Criteria
NFR-ARCHITECTURE* (NFR-DOCUMENTATION1)	The game shall be accompanied by detailed architecture documentation.		6 pages of architecture documentation containing diagrammatic representations and justifications shall be produced
NFR-DOCUMENTATION* (NFR-DOCUMENTATION2)	The game code shall be commented and documented.		>95% of code should either be self-explanatory or well-documented.
NFR-MAP-RESILIENCE* (NFR-RESILIENCE 1)	A problem with one map location shall not impact other map locations.	UR-INTERACT	>95% of game plays that experience an issue with one location, all others will not be affected.
NFR-SCALABILITY	The game shall support a single player at a time.	UR-INTERACT	No more than 1 person will play in 1 game.
NFR-BEGINNERS* (NFR-OPERABILITY1)	The game shall be playable by users with no prior experience of it.	UR-INTERACT	>95% of users will find the game easy to understand even if they previously played for 0 hours.

NFR-SETUP* (NFR-OPERABILITY2)	Users shall set up the game without needing training.	UR-INTERACT	>95% of users will find the set up easy despite having 0 hours of training.
NFR-ITEM-VARIANCE* (NFR-ACCESSIBILITY1)	All game items shall always be distinguishable by shape as well as colour.	UR-ACCESSIBLE	>95% of colour-blind users will be able to access the game.
NFR-ERRORS* (NFR-USABILITY1)	Any technical error messages shall be hidden from the user and a user-friendly, plain English message shall be presented instead.	UR-DESIGN	<1% of users will see a technical error message when playing the game.
NFR-STARTUP* (NFR-RESILIENCE2)	The game shall be reliable and start as expected without being unavailable.	UR-WORLD	>98% of game starts will be successful.
NFR-INSTRUCTIONS* (NFR-USABILITY2)	All game instructions shall be provided in plain English and avoid technical and university jargon.	UR-DESIGN	100% of game instructions will be in plain English with no jargon.
UR-SOUND-ACCESS* (NFR-ACCESSIBILITY2)	No elements or instructions of the game shall only be indicated by sound.	UR-ACCESSIBLE	100% of sounds and music will be supplemental and not necessary.
NFR-EXPERIENCE* (NFR-OPERABILITY3)	The game shall be playable by users who have had no experience of the game features or locations in real life.	UR-WORLD	>95% of players shall report that they found it easy to play the game even with 0 hours of university experience.
NFR-DAY-END* (NFR-TIMING1)	After 16 game hours, the player shall be unable to do anything other than sleep.	UR-SLEEP	After 16 hours of game time, players must be forced to sleep in 100% of cases.
NFR-PLAY-TIME* (NFR-TIMING2)	The game shall last between 5-10 minutes for an average player.	UR-TIMED	>90% of players will play for a minimum of 5 minutes and a maximum of 10 minutes.
NFR-UNDERSTAND-CODE* (NFR-MAINTAINA	Team members not involved in implementation shall understand what is happening in the code.		All team members will be able to understand the code within 1 hour.

BILITY1)			
NFR-NEW-TEAM* (NFR-MAINTAINA BILITY2)	A new team shall be able to understand and change the code.		>90% of comments and code documentation must be understandable to new teams immediately***
NFR-VISUALS* (NFR-USABILITY3)	The game shall be appealing and present the university in a happy and positive way.	UR-DESIGN	>90% of users should report that the design of the game was appealing.

References

[1]   IEEE Systems and software engineering - Life cycle processes - Requirements engineering, IEEE Standard 29148 Second edition, 2018