Requirements

Group 27 - WaddleWare Studios

James Smith

Amanda Ling

Fran Medland

Hannah Vas

James Kellett

Malik Tremain

Mischa Zaynchkovsky

Introduction
From our customer meeting and looking at the product brief, we categorised the requirements into 3 categories- User requirements (UR), Functional requirements (FR) and Non-Functional requirements (NF). Each of these serves a different purpose in specifying what the game should achieve, how it should function and the quality attributes it should have. Research into these requirements has informed the importance of each requirement, ensuring a comprehensive coverage of the games featured and characteristics. We referred to Ian Sommerville's 'Software engineering' book when writing our requirements.

User requirements- This outlined the fundamental expectations from the perspective of the user. We looked at what our target audience of users would expect from the game emphasising on playability, enjoyment, appearance, accessibility, interactions and many more. These requirements came from looking at the product brief and researching our target audience. They serve as guide principles for the overall design and development process. This ensures the game would meet their expectations to give them a good game experience making sure it is enjoyable.

Functional requirements- We looked into what our game must do such as the specific behaviours/functions our game should have. There were also a number of game mechanics such as energy bars, character interactions and map displays. For each requirement, we assigned it to a user requirement which made it easier to group together to understand what was needed in our game. By referring to the product brief, we ensured the criteria was met when creating our requirements. These requirements were elicited from user feedback and customer meetings.

Non-functional requirements- We looked at the quality attributes that our game should have. These requirements ensure the game is not only functional but also enjoyable to our target audience. We wanted to make sure the game was user-friendly, looked nice and was enjoyable to play. When we added a non-functional requirement, we added a functional requirement to go alongside it, then assigned it a user requirement to be able to group together. From there, we added a fit criteria for each NF requirement. This ensures that we could measure the intention of the requirement.

The elicitation and negotiation of these requirements involved discussions/meetings with potential customers which helped us prioritise the features based on importance and looking at the project's goals and constraints. Additionally, this facilitated effective communication and collaboration among the development team, which allows them to make informed decisions throughout the agile lifecycle The presentation of these requirements reflects a comprehensive understanding to make the game engaging, enjoyable, functional and accessible to be able to meet the target audience for the game. By presenting it this way, this gave us a clear understanding and vision for the developers within this project and showed us expectations we need to meet.

User Requirements

ID	Description	Importance		
UR_Access	The game needs to be accessible to play			Must
UR_Activity	The game has 3 activity types - Eating, Recreation and Studying			Should
UR_Apperance	The game will look appealing to play			Could
UR_Easy_Play	The game must be easily playable			
UR_Enjoy	The game must be enjoyable			
UR_Interract	The user must be able to interact with buildings			
UR_Map	The game must have a map			
UR_Movement	The user must be able to move around			
UR_Resting	The user must rest at the end of the day to move to the next day			
UR_Scoring	The game should have a scoring system			
UR_Skill	The game is skill based			
UR_Time	The game will last 5-10 mins			
UR_User	The game is a single player game			
UR_Win	The game must be winable			

Functional

ID	Description	User req
FR_Difficulty	The system must have a reasonable level of difficulty	UR_Access
FR_Simplicity	The game must be simple and easy to understand	UR_Access
FR_Background_Displa	The games writing should contrast with the background	UR_Access
FR_Interaction	The system shall have simple/basic user interraction system	UR_Access
FR_Controls	The game will display the controls to the user visually	UR_Access
FR_Energy_Amount	The amount of energy depletion will depend on the activity type	UR_Activity
FR_Building_Activities	Each building can only have one activity type associated with it	UR_Activity
FR_Buildings	There must be at least one building for each activity type	UR_Activity
FR_Study	The character must study at least once a day	UR_Activity
FR_Energy_Bar	Energy will be depleted by each activity	UR_Activity
FR_Scalability	The games display needs to be scaleable	UR_Apperance
FR_Energy_Display	The game should display the energy bar	UR_Apperance
FR_In_Game_Time	The time (in game) must be displayed	UR_Apperance
FR_Tutorial	The game will have a tutorial	UR_Easy_Play
FR_Menu_Screen	The game may have a menu screen (eg. tutorial/settings and play)	UR_Easy_Play
FR_Music	The game must have music	UR_Enjoy
FR_Interact	The user will use a key for interraction with buildings	UR_Interract
FR_Icon	The user will see an icon when it can interract with a building	UR_Interract
FR_Map	The game map must be of campus east	UR_Map
FR_Map_Locations	The map must contain places for study, food, activities	UR_Map
FR_Whole_Map	The player should be able to see the whole map the entire time	UR_Map
FR_Time_Display	The game should display the current day	UR_Map
FR_Movement	The user will use wasd keys for movement	UR_Movement
FR_Sprite	The user has a fixed icon that it controls	UR_Movement
FR_Time_Sleep	Sleep transitions the user to the next day	UR_Resting
FR_Sleep	The character must sleep at the end of each day	UR_Resting
FR_Score_Calc	The games score is calculated from the users interractions	UR_Scoring
FR_Score_Dec	Score will decrease if the character studies too much	UR_Scoring
FR_Study_Score	Score is increased by studying	UR_Scoring
FR_Recreation_Score	Score is increased by doing a recreational activity	UR_Scoring
FR_Overstudy	Score is decreased by overstudying	UR_Scoring
FR_Forget_to_Eat	Score is decreased by forgetting to eat	UR_Scoring
FR_No_Recreation	Score is decreased by not doing any recreational activity	UR_Scoring
FR_Timinig	The game will have a timing system that will ensure it finishes in around 5-10 mi	UR_Time
FR_Smooth_Time	Time passes smoothly	UR_Time
FR_Time_Drain	Time passively drains	UR_Time
FR_Time_Cut	Time drains in cuts when a user interracts with a building	UR_Time
FR_End_Message	A message is displayed at the end to tell the user if they have won	UR_Win
FR_Score_Display	The score is displayed at end	UR_Win

Non Functional

ID	Description	User req	Fit criteria
NF_Availability	The system should always be available to be played	UR_Access	Once the game is downloaded it is stored locally and doesnt require any internet to run
NF_Colour	The system may have different coulour schemes	UR_Access	Not implemented yet
NF_Graphics	The graphics should be clear/distinguishable	UR_Apperance	The font used is clear and all writing is of a readable size, should be easy to read
NF_Reliability	The system should reliably run	UR_Easy_Play	Testability made on different computers and shouldn't crash
NF_Music	The game could have background music	UR_Enjoy	Not implemented yet
NF_SFX	The game may have SFX	UR_Enjoy	Not implemented yet
NF_Display	The display will be top-down	UR_Map	The whole map is visable on the screen
NF_Map	The map should remain static at all times	UR_Map	The map of the campus doesnt move
NF_Difficulty	The game will be difficult to score highly	UR_Skill	The highest possible score requires a perfect runthrough
NF_Lose	The game must be possible to lose	UR_Skill	If not all the requirements are met, the user fails
NF_Game_Time	The game should last 7 days (in game time)	UR_Time	The day count is shown and lasts 7 in game days
NF_Win	The game must be easily winable	UR_Win	It is easy to fulfil the basic requirments on