Requirements

Cohort 1, Team 1:

Connor Blenkinsop
David Luncan
Emily Webb
Muhidin Muhidin(MO)
Sam Laljee
Sooyeon Lee

We first began the discovery of requirements for our product via the product brief (ENG1 Product Brief 2021 - York Pirates!.pdf). We interpreted this as a (comprehensive) single statement of need, and felt as if it generated a good amount of requirements. We continued the elicitation of requirements through the client meeting - this was invaluable as it helped us understand and spot large gaps in our User Requirements (UR), which had had a knock on effect on the quality of System Requirements. This meeting not only answered existing questions and offered prompts to many new requirements, but also pointed us in a better direction regarding future, and more detailed requirements to be added upon a review of the lecture on requirements engineering.

Throughout the client meeting - and beyond that, asynchronously throughout the first weeks of the project - requirements were first just noted down informally on a shared document for easy access and flexibility. This was the early version of our URs that can now be seen in the first table. In the formal writeup, prior to starting development work on the game itself, these URs were given a ID tag, for easier identification and reference, but also an implementation priority rating, ranging from Shall - requirements essential to the functionality of the game, to Should/May - requirements that would improve the product and/or allow it to better fulfill the brief, but that aren't crucial for it's running.

From these URs we were then able to extract more technical descriptions of functionality and system constraints, to aid implementation. These are split over functional and non-functional requirements, with some being grouped in categories (denoted by the second part of the ID tag matching) and all Non-functional requirements containing a specific fit.

All assumptions and risks are referenced in the description and further detailed at the end of the document

Lastly, as this part of the project was shared between multiple people, we decided on including a very basic version log so we can track when changes are made, and by whom.

User Requirements Table (UR)

ID	Description	Priority
UR_USABILITY	The application must be playable on a department pc, even without network access	Shall
UR_USABILITY_CONTROLS	Only a keyboard and/or mouse will be needed to play the game [1.1, 2.3]	Shall
UR_GOAL	The application must have a goal for the user to complete. If the game is lost, the user can return to the start screen. The user must be made aware when they have won. The game goal must involve college combat	Shall
UR_TIMING	The game should take 5 to 10 minutes to complete, for the average user (on a university open day) [1.1, 1.2, 2.1]]	Should
UR_ACCESSABILITY	The application shall offer a good user experience be suitable for a wide audience, and the game should be able to be accessed and closed easily, at any time	Shall
UR_GAME_OUTLINE	There will be (at least) 3 colleges based on the real UoY colleges that the player can sail between/interact with using boats	Shall
UR_STATS	Users should be able to easily check their progress, XP and etc which would be visible on screen throughout the game [2.1]	Should
UR_GAME_MAP	Users shall be able to see where they need to go on a map that can be easily accessed/visible	Shall

System Requirements (FR)

ID	Description	User Requirements
FR_COLOUR	The application must be usable for people with colour blindness	UR_ACCESSABILITY, UR_STATS, UR_USABILITY
FR_INSTRUCTIONS	The application must include instructions within the game - no external instructions	UR_ACCESSABILITY, UR_USABILITY
FR_USABILITY_LANGUAGE	Instructions and all language within should be plain and simple to understand	UR_ACCESSABILITY

	[1.2]	
FR_RELIABILITY	The application must not crash on invalid inputs or upon a win/loss	UR_ACCESSABILITY
FR_USER_DEFEAT	The application must allow user to play again if they are defeated by a college	UR_ACCESSABILITY, UR_GOAL
FR_END_SCREEN	The application will have an end screen with the option to play again	UR_ACCESSABILITY, UR_GOAL
FR_XP	The user will be able to earn XP via time passage and travel	UR_GOAL, UR_STATS
FR_PLUNDER	The user will be able to earn plunder via college combat	UR_GOAL
FR_PROGRESS	The goal should not be immediately achievable	UR_GOAL
FR_SCALING	The application increases difficulty when defeating non objective colleges	UR_ACCESSABILITY, UR_GOAL
FR_RATING	The application should be appropriate for a PG rating	UR_ACCESSABILITY
FR_FLASHING_LIGHTS	The application must not contain flashing lights	UR_ACCESSABILITY
FR_MUTABILITY	If the game does contain any sound these must be able to be set on mute	UR_USABILITY
FR_START_SCREEN	The game will have a start screen that the user can go back to anytime. The game may return to this screen if the user is AFK for too long. [2.4]	UR_ACCESSABILITY
FR_PLOT_COLLEGES_COUNT	A minimum of three colleges, representing existing UoY colleges must be represented	UR_GAME_OUTLINE
FR_PLOT_CAPTURE	Each college must be a goal/defeatable	UR_GAME_OUTLINE
FR_ALWAYS_VISIBLE	The game map and xp/plunder bars will be visible while the game is being played [2.2]	UR_GAME_MAP, UR_STATS

System Requirements (NFR)

ID	Description	User Requirements	Fit Criteria

NFR_COMPATABILITY	The application will only be required to operate on MacOS, Linux, and Windows operating systems.	UR_USABILITY	100% of the game will perform on MacOS, Linux, and Windows OS
NFR_INPUT	The application will be operated using keyboard and mouse [1.1]	UR_USABILITY, UR_USABILITY_CONTROLS	90% of the users will be able to play the games using keyboard and mouse
NFR_NETWORK	The application must not connect to the network	UR_USABILITY	100% of the game will perform without network connection
NFR_SCALING	The application must be usable on multiple screen sizes [2.2]	UR_USABILITY,	100% of the game will perform from 15 inch laptop to 27 inch monitor
NFR_USABILITY	The application must be immediately usable and understandable by the target audience [1.1,1.2]	UR_ACCESSABILITY	90% of the users will be able to understand the game within 5 minutes of game play
NFR_MEMORY	Memory should be cleared at the end of each game	UR_USABILITY	100% of the game will be cleared at the end of each game
NFR_FILE_SIZE	The application will have a limited file size to accommodate the data portal	UR_USABILITY	The entire zipfile can be successfully uploaded to the portal
NFR_TIMING	The application will be completed within 5-10 minutes	UR_TIMING	90% of the users will be able to complete the game within 10 minutes
NFR_RESPONSIVENESS	The game will be responsive to human input [1.3]	UR_ACCESSABILITY, UR_USABILITY	Any response should be within 0.5 of an input

Assumptions made:

1.1 The user is familiar with basic use of a qwerty keyboard
1.2 The user can understand and speak basic english
1.3 The user input is a valid input

Risks:

2.1 Should requirement not being implemented:
2.2 UI becoming too cluttered on smaller screen sizes
2.3 Not following conventions may be confusing (clear explanation needed to mitigate this risk)
2.4 This can become frustrating if timeout is set too short, and a useless implementation if it is set to too long