1 - Requirements Justification

1.1 - Elicitation and Negotiation

The primary stage of this project is to elicit our requirements. These will play a crucial role in the life of the project, as they form an adaptable abstract which can be referred to throughout the creation of the end product. With this in mind, we took several steps to understand what the client was hoping we would accomplish with the project and consolidated them into this document.

Group Discussion

We began by reading the project brief individually and then additionally as a team, discussing our initial ideas whilst simultaneously forming a mutual understanding of the key points, followed by the creation of a list of questions that required further clarification with the client.

Stakeholder Meeting

Following this, we had a meeting with our customer to clear up any ambiguity left in the brief, which was useful as it helped refine our initial interpretation of the requirements. Additionally, we negotiated any supplementary features that we came up with during our team discussions and garnered the client's opinion on them.

User Survey

In order to gauge the average feelings of our end user, we created a survey which asked some key questions concerning their personal preferences on games of this type and what should be included. In particular, we were interested in their thoughts on how they felt the mini-game aspect should be incorporated within the main game.

Negotiation of Requirements

Throughout this process we organised multiple group meetings (daily Scrums) to decide our requirements, where discussions involved both what the customer had requested and what we thought could be added to improve the player experience in innovative ways. We further investigated other games of the genre like "Sid Meier's Pirates!" for inspiration, in addition to making use of different techniques like use case and paper prototyping to make sure our requirements were precise and well designed. Once every member of the team came to a general agreement of a requirement it would be added to the specification, which was then later shown to our client, aside our paper prototype, for validation. This also keeps true to the core values of the Agile approach to development (which we will further describe in the Method Selection and Planning document): Promoting Individuals, Interactions and Customer Collaboration.

1.2 - Presentation

To present the requirements we used a tabular format for easy navigation, which complied with the IEEE Software Requirements Specification (SRS)[1]. The full version of this document proved to be too much for the small scope of our project, so we only made use of what we felt necessary. We also used a reference system allowing us to consistently refer back throughout the project's lifecycle. Following this standard, we have divided our requirements into **Functional(F)**, **Non-Functional(NF)** and **Constraint(C)** categories. We also consulted ISO/IEC 25010[2] and further categorised our non-functional requirements into **Usability (U)**, **Reliability (R)**, **Maintainability (M)** and **Freedom from Risk (FR)** for easier management and risk assessment.

2 - Requirements Specification

2.1 - Functional Requirements

Reference	Requirement	Fit Criterion	Risk/Alternative
F1	The game must feature the University of York as represented by islands on a huge lake, with ships as the only mean of transportation and the player as a privateer.	**F1.1** The in-game map must be recognisable as related to the actual University of York, such as containing two main clusters (Campus East and West)	
F2	The game must end when the game's objective is achieved, otherwise the player must respawn with half their money and all their points docked	**F2.1** Game victory screen when final boss defeated, otherwise respawn with points and money docked **F2.2** Points will be shown on post-game screen	
F3	The game must have a sailing mode.	**F3.1** Player will be able to move freely on the water area of the map by keyboard or mouse input. **F3.2** Chance to encounter enemies in the form of ships, monsters. **F3.3** Chance to encounter obstacles and bad weather.	
F4	The game must have a combat mode.	**F4.1** Combat mode begins upon encountering an enemy. **F4.2** Combat mode will be easily distinguishable from sailing mode. **F4.3** Combat mode will take the form of turn-based. **F4.4** Enemies could be other ships or land-bound objects. **F4.5** Both player and enemies will have visible attributes. **F4.6** Combat mode ends after one side's health drops to zero or player flees successfully.	Combat mode can be real time.
F5	The game must have at least five colleges and three departments.	**F5.1** Five colleges and three departments based on real locations in the University of York will be featured. **F5.2** In-game locations and proximity between them will try to imitate as much as possible from real life.	If player has little to no knowledge about the University of York they might miss out on the implications.
F6	Points must be accumulated through combat and passage of time.	**F6.1** Points will be assigned based on the strength of the opponent defeated. **F6.2** For each fixed amount of time passed, the player will be rewarded a certain number of points. **F6.3** Points will be awarded upon survival from bad weather. **F6.4** There will be a point multiplier that increase along with game length.	Player might do nothing to earn point just from passage of time.
F7	Defeating another ship will	**F7.1** Gold will be awarded upon defeating	Drop items instead of just

	lead to the acquisition of plunder.	enemies. **F7.2** The stronger the enemy, the higher the amount of gold acquirable.	gold.
F8	It must be possible to capture another college via combat. A successful capture gives additional point and plunder.	**F8.1** A college can only be attacked after all guarding ships are defeated. **F8.2** Once all the guarding ships are defeated, the player will face a warship with boss-like strength. **F8.3** The college will be captured upon defeat in battle. **F8.4** Large amount of point will be awarded upon successful capture of a college. **F8.5** Captured colleges will give perks: e.g: Generate gold over time, give vision on map...	
F9	Departments cannot be captured	**F9.1** Departments will act as shopping points selling special boosts to the player's ship.	Bosses that need to be defeated before the final boss
F10	Each gameplay should have an objective and it should not be immediately achievable.	**F10.1** There will be a final boss that need to be defeated to complete the game. **F10.2** The boss will be strong enough that the player needs to reach a certain level of upgrade in order to defeat.	Boss too strong player feel discouraged.
F11	There should be a way to spend the plunder acquired	**F11.1** Player will be able to upgrade ships or acquire special boosts with gold. **F11.2** Plunder can also be spent on healing.	
F12	There must be a mini game completely different from the main game	**F12.1** There will be a mini game that involves betting with the player's current gold.	- Need to ensure it is completely random or else the player might abuse. - Another type of mini game

2.2 - Non-functional Requirements

Reference	Requirement	Fit criterion	Risk/Alternative
NF1(U)	The game should be easy to pick up and playing should feel pleasing and satisfying.	**NF1.1** There will be a short tutorial at the beginning. **NF1.2** Difficulty will increase over time. **NF1.3** There will be mechanics that protect users against making error.	- Risk 16.
NF2(R)	The game must be able to operate on a university computer with good performance.	**NF2.1** Crashing rate should be at most 5%. **NF2.2** Frame rate should not drop below 30. **NF2.3** There will be a save/load feature in the case of software or hardware failure.	- Risk 10.
NF3(M)	The game will be built to easily cater for future expansions and bug fixings.	**NF3.1** OOP characteristics will be implemented extensively for easier reusability for expansion. **NF3.2** Code will strictly follow UML models. **NF3.3** Code will be well formatted and	

		commented. Certain conventions will be implemented on the code.	
NF4(FR)	The game will adhere to PEGI 12 requirements[3].	**NF4.1** Violence towards humans mustn't look real unless it's showing trivial injury. **NF4.2** Any included bad language must be mild.	- Risk 16.

2.3 - Constraint Requirements

Reference	Requirement	Fit criterion	Risk/Alternative
C1	The game must be completed and ready for customer presentation by 1st May 2019		- Risk 5. - Definitive deadline might discourage significant changes near the end of the project.
C2	The game must be coded in Java and in-game language must be English	**C2.1** English will be used for all visual and audio effects.	
C3	The game should be commercially marketable and salable.	**C3.1** All aspects of the game should focus on providing a satisfying experience for the player. **C3.2** The game should stand out from others of the same genre.	- Risk 12.
C4	The game should be appropriate for use in the University of York's promotional activities.	**C4.1** Certain aspects of the game should promote the University of York. **C4.2** Implications related to the University could be used to add some humour.	

References

[1] Systems and software engineering — Life cycle processes — Requirements engineering, IEEE Standard 29148, 2011.

[2] ISO/IE 25010 https://edisciplinas.usp.br/pluginfile.php/294901/mod_resource/content/1/ISO%2025010%20-%20Quality%20Model.pdf

[3] PG 12 https://parentinfo.org/article/pegi-games-ratings-explained