Section 2A - Requirements Introduction

Gathering and recording requirements is very important for designing a project hence why it was the first thing we started working on. In our first in person meeting, we brainstormed and discussed a lot of different requirements, then decided which ones were system requirements and which ones were user requirements, we wrote down all the requirements that we all agreed upon. We then divided the written requirements into three different categories, as in functional requirements, non-functional requirements and constraint requirements. We made a document stating all of these requirements and wrote beside each one its level of priority based on their significance in the project.

Later on, we had our first client meeting where we were able to clarify any questions we had about the requirements, through this process we got a clearer picture about what the client wanted out of the game. As a result, we were able to filter out any requirements that weren't needed and added some new ones mentioned by the client. Finally we had another meeting to do a final review of all the requirements and made a systematic and appropriately-formatted statement of requirements, later tabulating these requirements in part 2b.

In our requirements table, we use a simple statement to indicate the priority of a requirement and any associated risks as well as if there are any alternatives to the requirements. For most requirements they are strict requirements that have little to no alternatives, however, for some they are more open to changes. However, for functional requirements, we have used "fit criteria", a brief statement indicating the condition that must be met to have satisfied the non-functional requirement.

Section 2B - Requirements Table

The following requirements table splits our main requirements into three categories, user (UR), functional (FR) and non-functional requirements (NFR).

ID	**Requirements**	
UR	**User Requirements**	**Priority**
UR_1	The game shall cater to colourblind users.	Strict requirement
UR_2	The game must be non-violent and feature no flashing images to cater to all users.	Strict requirement
UR_3	The user will have full control over their ship.	Strict requirement
UR_4	Users will be able to engage in combat with colleges.	Strict requirement, no alternative.
UR_5	The game will be interesting to prospective students.	Strict requirement
FR	**Functional Requirements**	**Priority, Risks and Alternatives**
FR_1	The game should restart upon completion.	The game must restart to allow another user to play the game shortly after.
FR_2	The ship’s movement will be controlled using ‘WASD’.	High priority, no alternative.
FR_3	The ship should accelerate and decelerate at uniform rate until it reaches maximum speed or stops.	Optional, could instantly reach top speed or stop.
FR_4	The ship should lose speed gradually when movement keys are not pressed.	Optional, could stop instantly if needed.
FR_5	The ship’s weapons should aim towards the mouse pointer	High priority, no alternative
FR_6	The ship’s weapons should fire when the mouse is clicked.	High priority, but the ship’s weapons could be fired using any button e.g., the spacebar.
FR_7	The ship’s weapons will be a projectile that fires in the direction of the mouse pointer and has a travel time.	High priority, no alternative.

FR_8	The projectile will stop if it hits an object, dealing damage to that object if it is a college or ship.	High priority, no alternative.
FR_9	The projectile should have a lifespan, disappearing after some time if it does not collide with an object.	High priority, could also disappear once the projectile is no longer on the screen.
FR_10	The player will be able to earn XP points while playing the game, small amounts from sailing around the map and a large amount for defeating colleges.	High priority, no alternative
FR_11	The player should also be able to earn plunder from defeating colleges or ships.	High priority, no alternative.
FR_12	The game should have a fixed map with colleges spread out evenly.	High priority, could use a randomised map but that may be more difficult and time consuming.
FR_13	The game will contain colleges as the main enemies.	High priority, no alternative
FR_14	The colleges will be a collection of buildings that will shoot at unaffiliated ships.	High priority, the colleges could be a singular building.
FR_15	The colleges should take damage when shot by the player, and the player should take damage when shot by a college.	High priority, no alternative.
FR_16	When a college is defeated, it should become allied with the player and stop attacking.	High priority, colleges could simply be destroyed and removed from the map.
FR_17	Colleges should be surrounded by allied ships that can be destroyed by the player	High priority, no alternative
FR_18	The game should consist of a quest that needs to be completed in order to beat the game.	High priority, no alternative
FR_19	The quest should have at least one smaller objective that needs to be completed before the main objective.	High priority, no alternative
FR_20	The final objective should involve defeating a certain college.	High priority could be changed to defeating all colleges or something similar.

FR_21	Once the player has taken too much damage, their ship should sink, and the game should end.	High priority, no alternative
FR_22	The game should feature a tooltip at the start, telling the user how to play the game.	High priority, no alternative.
FR_23	The game should feature bright, large, eye-catching graphics while still being colour-blind friendly	High priority, no alternative
FR_24	A college should be surrounded by allied ships.	Optional - ships could be included mainly to add more to the map.
FR_25	The game should feature a pause menu.	Optional - the game will only last around five minutes and may not need to be paused.
NFR	**Non-Functional Requirements**	**Fit Criteria**
NFR_1	The game should easily be playable by new users.	99% of players should be able to play the game without extra instructions.
NFR_2	The game should run on all major operating systems.	The game should function properly on windows, linux and MacOS.
NFR_3	The game should scale to any screen size.	The game should fill 100% of the screen regardless of how large or small.
NFR_4	All user-facing messages shall be in plain English and not use technical jargon.	100% of text displayed to the player should be understandable by someone with little to no technical knowledge.
NFR_5	The game should be playable by colour-blind users.	Colour-blind users should be able to play the game as easily as non-visually impaired users.
NFR_6	All code for the implementation of the game should be well documented.	Anyone who takes over this project should be able to easily make sense of all parts of the code.