2. Requirements

A. Introduction

Acquisition

- We utilised the ability to meet in person with the stakeholder through the use of an interview, in which we posed questions and noted down answers and useful information that we used to form a project structure.
- The primary focus of the initial elicitation of the requirements from stakeholders was to get a broad outline of the project; this allowed us to form and develop our initial team working strategies, with the option of later developing on this through further research and questioning if needed.
- The group created questions we had wanted to ask the stakeholder, and we planned to develop and add onto these as answers from the stakeholder were given - this meant we were following a semi-structured interview pattern.
- To aid this imperative, the first and last questions asked in the interview were forwarded as open ended. This had the intent of allowing the stakeholder to fill in gaps, and provide us material that we can develop and adjust the questions from, on the go, during the interview.
- After the initial elicitation there is intent to follow up during the course of the interview with shorter form questions and queries. This will allow us to refine our knowledge base and clarify vague or previously unconsidered information.

Formalisation

- The notes acquired from the interview were then compiled and formulated together into the requirements in this document. Consideration was taken to ensure that the key aspects that the stakeholder implied through their answer was retained between the changing of formats. For each requirement this often involved splitting each question and each answer up into multiple sub-requirements, and shortening/refining the description and purpose of each.
- As well as descriptive attributes, each requirement also included information for both **Relevant Risks** and **Severity.** These provided a framework in which we as a team can work around, while also providing goals we can work towards.
- The links to the risk assessment allows cross referencing and ensures that this documentation is actively considered during development, while also meaning other team members can easily see active and current issues. The severity inclusion allows us to quickly identify key points to work on in the implementation and gives focus to the team.

Research

- Our research for the presentation of the requirements section consisted of Sommerville's ‘Software Engineering’[1]. He states that the user requirements should ‘describe the functional and nonfunctional requirements so that they are understandable by system users who don’t have detailed technical knowledge.’
- This should be free from technical terminology, but diagrams and tables can be used to assist complex requirements.
- We believe the best method to format our requirements would be in the form of multiple tables, each representing the non-functional and functional requirements as well as their environmental assumptions. We decided on this approach as it’s easier to understand than a series of paragraphs.

B. Requirements

Non-Functional					
ID	**Name**	**Description**	**Requirement Type**	**Priority**	**Env. ID**
NF001	Audience	Target the demographic of the open day.	User	Medium	A001
NF002	Timing	Game should be 5-10m long	System	Medium	A002
NF003	Difficulty	Difficulty shouldn't be too challenging	User	Medium	A002
NF004	Amount of players	Single player	User	High	N/A
NF005	Map	Map is user friendly / easy to navigate	User	Medium	A002
NF006	Accessibility 1	Accommodate for colour-blindness	User	High	A003
NF007	Accessibility 2	Allow for remapping of keys	User	Med	A003
NF008	Accessibility 3	Allow for controller support	User	Low	A003
NF009	Accessibility 4	Ensure seizure-friendliness	User	Med	A003
NF010	Cheating	No specific hardware or input a user can use to unfairly gain an advantage.	User	Med	A002
NF011	Network Protection	Cannot access / affect the game through malicious networking intent.	User	Med	A002 A004
NF012	Network Isolation	Game is not connected to and cannot access any network.	System	High	A004
NF013	Scalability	Graphics need to support between a 13" and 27" monitor whilst looking good proportionally to resolution	User	High	A003
NF014	Reliability	Game needs to run reliably, accounting for problems such as memory leak.	User	High	A003
NF015	Fluidity	Should not run under a reasonable threshold of FPS such that it is difficult for the user to control the ship.	System	Med	A004
NF016	Ability to win/lose	Game needs to be able to finish and restart with no transition screen	System	High	A002
NF017	Tutorial/ tooltips	User needs to learn controls in such a way that doesn't open a separate menu	User	High	A001 A002
NF018	Stats	Can see stats of the ship on-screen	User	Low	A002

Functional				
ID	**Name**	**Description**	**Requirement Type**	**Priority**
F001	Restart Capability	Restart the game to the start on “tab”	User	High
F002	Restart Consistency	Reset state and ensure no memory leak on restart or close	System	High
F003	Control	“Wasd” moves the ship in the 4 cardinal directions	User	High
F004	Control	Combining “wasd” moves in diagonal directions	User	High
F005	Destroy college gold	When a college is destroyed, the user should be rewarded gold / XP	System	Med
F006	XP Gain	As you play the game, you gain XP over time	System	High
F007	Collision	The ship cannot move into walls	User	High
F008	College Attack	Colleges attack the ship when nearby	System	High
F009	Ship die	Player ship dies when their health hits 0	System	High
F010	College die	Colleges die when their health hits 0	System	High
F011	End screen	When the player ship dies, a game over overlay appears	System	Low
F012	Ship attack	With “lmb” or “rmb” the ship attacks in the mouses direction	User	High
F013	Random Spawn	Colleges spawned randomly on the map	System	Low
F014	Particles	Various particles generated while playing, for: behind boat, projectile break	System	Low
F015	Objective Complete	When the objective is complete the game will end to a victory overlay	System	High
F016	Restart Button	When on the victory screen, game can be restarted using “Space”	User	High
F016	Objective Track	As the player plays the objective will update and keep track of the players progress	User	High
F017	Can Close	Game window closes on escape	System	Med
F018	Passive Regen	Players health regenerates over time while outside of combat	System	High

F019	Repair	Players health regenerates quickly while near home	System	High

Environmental Assumptions			
ID	**Assumption**	**Constraints**	**Risks**
A001	Game must be built for target demographic	- Must be suitable for children. - Must not show specific content.	- Violent content shown to children.
A002	Presented at an open day.	- Game must be designed with the atmosphere of an open day in mind.	- Risks of over running / not engaging
A003	Some players might have disabilities	- Features of the game mustn't discriminate those with disabilities	- Potentially causing harm to users - Potentially putting some users at a disadvantage
A004	Game will be played on the University's computers.	- Game must be able to be played as intended on the university's computer. - Must not allow for any security threats to the University's network	- Libraries or software might be not supported - Might not be compatible with required OS

References

[1] I.Sommerville, “Requirements Engineering” in Software Engineering, M. Hirsch, London, Pearson, 2016, pp 120.