2. Requirements

Group 11 - 11 Musketeers

Osama Azaz

Adam Dawtry

Tom Jackson

Brendan Liew

Holly Reed

Harry Ryan

2.a

How requirements were elicited and negotiated:

- The requirement elicitation, specification, and validation, were carried out as Ian Sommerville suggests [1] in order to complete requirements engineering.
- Requirements were elicited by interacting with a stakeholder via an interview in the first few weeks of the project, and then were analysed soon after.
- We decided to employ a mixture of open and closed interview methods.
- Prior to this customer meeting, our team collected ideas and concerns about a range of issues already stated by the design brief.
- We structured this by grouping the questions together under a specific issue heading.
- We developed a set of predefined questions we could enter the meeting with, whilst also leaving room for us to spontaneously develop further questions based on the customer's responses.
- The way these questions were grouped together allowed two scribes to quickly coordinate and organise the customer's requirements into a structure we were already familiar with, allowing for the quick development of further questions for clarification and higher detail.
- First, in collaboration with the customer, we began by determining who exactly would be using the software.
- From this, we were able to transition into clarifying and developing user requirements based on the design brief the customer had already delivered to us.
- The functional requirements of the system were then elicited.
- Finally, the customer was asked about any non-functional requirements or constraints they had envisioned for the system.
- This included questioning the customer about any hardware requirements, reliability, documentation, security, user interfaces and accessibility.

Why the requirements are presented as they are:

- We have decided to present our list of requirements by splitting them into user requirements and system requirements (consisting of functional and non-functional requirements).
- A table has been created for each of the types of requirements listed above.
- For each user requirement: a specific ID has been assigned to it, there is a brief description of the requirement, and a priority of implementation (shall, should or may) has been assigned.
- For each functional requirement: a specific ID has been assigned to it, there is a brief description of the requirement, and any relevant user requirements it satisfies are listed.
- For each non-functional requirement: a specific ID has been assigned to it, there is a brief description of the requirement, any relevant user requirements it satisfies are listed, and a fit criteria has been developed.
- This was done to more easily detail all of the requirements in enough detail, as suggested by Charles Lane and Nico Krüger [2].

2.b

User Requirements Table

ID	Description	Priority
UR_INTUITIVE	User should find the game intuitive to pick up and easy identify objectives	Should
UR_FIRST_ATTEMPT	User should find it somewhat easy to, and should be able to, complete the game on their first attempt	Should
UR_POINTS	The game has a point system allowing users to power up their ship	Shall
UR_GOLD	The game has a gold system which the user can use to purchase upgrades to their ship and weapons	Shall
UR_SIDE_OBJECTIVES	The main objective of the game should not be immediately achievable; there should be side objectives	Should
UR_COMBAT	College buildings should be able to damage the user as the user enters combat with them	Shall
UR_ONE_SHIP	User commands and moves only one ship	Shall
UR_TOGGLEABLE_SOUNDS	Music and sound effects, if implemented, must be toggleable	May
UR_LOOSEABLE	User should be able to lose the game	Should
UR_RESTART	A restart button should be available to the user at any point during the game	Should
UR_RANDOM	The game should have a randomised element	Should
UR_STABLE	User shall be able to run the game with no framerate or stability issues	Shall
UR_ACCESSIBLE	User shall find the software to be suitable to their age and any accessibility needs	Shall

- Requirements UR_INTUITIVE and UR_FIRST_ATTEMPT are based on the assumption that the game will be played by one-time players who are visiting the University of York. The game should be simple and easy to pick up, yet fun to play.
- Requirement UR_TOGGLEABLE_SOUNDS is based on the assumption that the game will have sounds implemented, however this has been given a priority of ‘May’ as sounds may not be implemented at all.
- Requirement UR_ACCESSIBLE is based on the assumption that the game will be played by a range of different visitors to the University of York: this could include younger family members or those with colour blindness.
- The risk R5, that the game window may close after the game has ended, should be mitigated by the requirement UR_RESTART, providing the user with a restart button at the end of the game instead of closing the window altogether.

Functional Requirements Table

ID	Description	User Requirements
FR_PROGRESSION	The game will have a progression structure where users can gain points, and gold from defeating colleges and ships or from discovering plunder on the map	UR_POINTS, UR_GOLD
FR_GAMEPLAY_INSTRUCTIONS	All in-game instructions and information will be displayed on screen; there will be no manual and instructions must be simple	UR_INTUITIVE, UR_FIRST_ATTEMPT
FR_RANDOMISED	Randomised placement of plunder/gold on the map	UR_RANDOM, UR_SIDE_OBJECTIVES
FR_ENEMY_BOATS	Enemy boats will be roaming around near their respective enemy college buildings, they should not engage the player in combat at this stage	UR_INTUITIVE, UR_SIDE_OBJECTIVES
FR_PLAYER_HEALTH	Player will have a health bar health which, once depleted, the game will end; damage should be taken from enemy ships and enemy colleges only	UR_LOOSABLE
FR_ENEMY_COLLEGE	Enemy colleges will have a leader/main ship, they should not engage the player in combat at this stage	UR_INTUITIVE, UR_SIDE_OBJECTIVES
FR_STATE_MEMORY	System will clean up state memory at the end of a game	UR_RESTART, UR_STABLE
FR_GAME_DURATION	Game duration should last around 5 to 10 minutes	UR_INTUITIVE, UR_FIRST_ATTEMPT
FR_REAL_TIME	Gameplay will be in real time	UR_INTUITIVE
FR_MAP	There will be one continuous map that has a fixed layout	UR_INTUITIVE

- Requirements FR_GAMEPLAY_INSTRUCTIONS, FR_GAME_DURATION, and FR_MAP are based on the assumption that each player will play the game only once, therefore the game should have simple instructions on screen, should last 5 to 10 minutes, and the map doesn't need to change.
- An alternative to requirement FR_RANDOMISED was discussed with the customer: the team raised the idea of each map being unique and procedurally generative, however this was dismissed as the game would only be played once by each person therefore good replayability is not needed.

Non-Functional Requirements Table

ID	Description	User Requirements	Fit Criteria
NFR_SECURITY	System should avoid using an internet connection or calling networks	UR_STABLE	Network calls = 0, internet connection not used
NFR_INPUT_METHODS	Game can be played via mouse and/or keyboard	UR_INTUITIVE, UR_ACCESSIBLE	Mouse only controls, keyboard only controls, and a combination of mouse and keyboard controls
NFR_ACCESSIBILITY	Must accommodate colour blind users	UR_INTUITIVE, UR_ACCESSIBLE	All objects identifiable by shape/size
NFR_OS	System should run on multiple platforms including Windows, Linux, and macOS.	UR_STABLE	Game is able to be completed on all platforms
NFR_HARDWARE	Minimum hardware requirements are a laptop with 4GB RAM and a 13 inch screen	UR_STABLE	Constant minimum frame rate of 30fps
NFR_DISPLAY	Display must be scalable and able to enter full screen mode	UR_STABLE, UR_ACCESSIBLE	Display >= 1920x1080 pixels
NFR_GRAPHICS	Top down art that is bright and large so it is visible from a distance	UR_INTUITIVE, UR_ACCESSIBLE	Each tile = 32x32 pixels, bright colour palette, and scalable
NFR_AGE_RATING	PG rated - no blood or gore	UR_ACCESSIBLE	Child <8 years old should be able to play and not become upset

- Requirement NFR_SECURITY is based on the assumption that the game will be demonstrated to, and played by, a range of people in a range of places throughout the campus where perhaps an internet connection is not available.
- Requirements NFR_INPUT_METHODS, NFR_ACCESSIBILITY, and NFR_AGE_RATING are based on the assumption that has already been listed in regards to UR_ACCESSIBLE.
- There is a risk that requirement NFR_GRAPHICS may not be fully completed due to the risk R17, that the member of the team responsible for developing each tile may fall ill.

References

[1] I. Somerville, *Software Engineering*, Global Edition. Pearson Education, 2016. [E-book] Available: https://ebookcentral.proquest.com [Accessed: 13 January 2022].

[2] C. Lane and N. Krüger (2021, Dec. 16). *How to Write a Software Requirements Specification (SRS Document)*. Perforce. [Online]. Available at: https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document [Accessed: Jan. 15, 2022].