Requirements:

Team 23

Kyle Mace (kjm560)

Josh Quinn (jtq501)

Louis Hatton (lwh506)

Faris Alblooki (fma527)

Lewis Power (lp1263)

Part A: Explanation of How Requirements Were Elicited

Introduction:

Before even implementing any code, our team thought it would be logical to delineate exactly what our requirements were first. We knew this was good practice, as per the KISS acronym that we came across during our initial software engineering research **[1]**.

Based on further research, our team understood that we should first produce a Single Statement Of Needs (SSON), then consider user requirements, and then consider system requirements. System requirements further ramify into functional and non-functional requirements **[2]**.

How requirements were elicited:

Our SSON, user requirements, and system requirements were established through:

- Thorough reading of the ENG1 product brief.
- Discussion and negotiation amongst team members.
- Virtual meeting with a product customer.

Through the product brief, we were able to establish seven user requirements and six system requirements. Despite this, we felt that there were some ambiguities within the product brief preventing us from establishing a few more requirements. So, as a team, we compiled a list of questions to be posed towards one of our product customers in a meeting on 30th November 2021. This proved useful, because seven further requirements (three user, four system) were derived as a result.

Also, after the meeting, an SSON was now able to be created to help us keep our project streamlined.

“The system should allow the user to compete in a single player game, moving as a ship with the option to attack other AI enemy ships or colleges; the game ends once either the users’ health bar is depleted by an enemy, or the user completes the overall objective of the game.”

Why requirements are presented as they are:

Our team found the ‘IEEE Guide For Developing System Requirements’ to provide insight into the level of detail required and the appropriate formatting. Namely in how to define a requirement and what the pitfalls of doing so usually are. **[3]**

To allow ease of distinguishing between the different types of requirements, the different tables have been coloured.

Part B: Systematic statement of requirements

User Requirements

Requirement ID	Description	Risks/Environmental assumptions/Alternatives	Priority
UR_TRANSPORT	The user can take only a ship as their mode of transport	Users may unknowingly pause camera tracking and not know where they are going	**Shall**
UR_MIN_COLLEGES	There must be at least three other colleges in the game	Too high a minimum, could conflict with UR_GAMETIME if too much combat is required.	**Shall**
UR_COMBAT	User can engage in combat with another college	Colleges may decide to attack the user at the same time. Conflicting with UR_DIFFICULTY.	**Shall**
UR_PLAYABLE	The system shall offer a pleasant user experience; HCI principles should be adhered to and music should be mutable	The game is intractably difficult to program whilst adhering to certain HCI principles	**Shall**
UR_DIFFICULTY	The game shouldn't be too easy or too difficult for the user to complete	The game is too easy or too difficult for the user to complete, making them likely to quit	**Shall**
UR_TERMINATE_DEFEAT	The system shall end the game if the users' health level drops to 0	The game does not end if the users' health level does not drop to 0, erroneously allowing them to continue to move around/attack	**Shall**
UR_TERMINATE_COMPLETION	The system shall end the game if the user completes the games' objective	The game does not end if the user completes the games' objective, erroneously allowing them to continue to move around/attack	**Shall**
UR_POINT_ACCUMULATION	The user must accumulate points through the passage of time	Points are not accumulated through the passage of time, meaning the user is likely to quit	**Shall**
UR_GOLD_ACCUMULATION	The user must earn gold after the capture of either colleges or other ships	Gold is not earned through the capture of either colleges or ships, meaning the user is likely to quit	**Shall**
UR_OTHER_SHIPS	The user may encounter other ships	The other ships do not move, therefore making them quite unrealistic enemies	**Shall**
UR_GAMETIME	Combat against enemies shouldn't last too long	The game could last too long, making the user likely to quit	**Should**

System Requirements - Functional

Requirement ID	Description	Risks/Environmental assumptions/Alternatives	User Requirements
FR_HEALTH	The health of the user and other colleges must be stored, and shown to the user	Health Display may get in the way of objects in the game	UR_TERMINATE_DEFEAT, UR_COMBAT, UR_PLAYABLE
FR_ENEMY	The system will initialise enemy ships that belong to other colleges	Alternatively boats could be deployed dynamically	UR_DIFFICULTY, UR_OTHER_SHIPS

FR_ENEMY_ATTACK	When within vicinity, enemy ships will move towards the user ship and fire projectiles when close enough	User being within enemy vicinity may not register due to other processes changing the user location.	UR_COMBAT, UR_OTHER_SHIPS
FR_BULLETS	When a ship shoots. Projectiles will be created at run time	Users' projectiles could cause damage to the users boat accidentally	UR_COMBAT
FR_ARROW_KEYS	Ships will drive using input from the arrow keys	Simultaneous key presses may cause unexpected movement	UR_TRANSPORT
FR_TERMINATION	System halts all game functions upon termination and prompts end game results	User may be able to still increase their points via the passage of time	UR_TERMINATE_DEFEAT, UR_TERMINATE_COMPLETION
FR_INCREASE_GOLD	When a college is captured and all the associated enemies are defeated the users gold will be increased by x amount	User may be able to double their take in gold by spending it as it's simultaneously is earned	UR_GOLD_ACCUMULATION
FR_INCREASE_POINTS	Every t seconds, increase the users points by a set amount	System clock is delayed due to other processes	UR_POINT_ACCUMULATION
FR_GAME_SCREEN	Always display the game screen to the user	if the user minimises the screen, the game may still play on instead of pausing	UR_PLAYABLE
FR_CAMERA	System will always follow the point of view of the user unless prompted otherwise	Camera may show outside of the map which may be displeasing to users.	UR_PLAYABLE, UR_TRANSPORT

<u>System Requirements - Non-Functional</u>

Requirement ID	Description	User Requirements	Fit criteria	Risks/Environmental assumptions/Alternatives
NFR_GAME_LENGTH	The game should be completable in a reasonable length of time	UR_GAMETIME	Each combat takes < 1 minute	Assumes boat movement and combat works properly
NFR_SCALABLE_SIZE	The game must be scalable	UR_PLAYABLE	size adjustments take < 1s	The size could be bigger than the map, showing outside the map. Users may find this displeasing.
NFR_INTEGRABLE_SYSTEM	Must be able to run on different OS	UR_PLAYABLE	Game compiles to Bytecode	The OS system has no Java Virtual Machine to interpret bytecode, rendering the game unplayable on that machine.
NFR_TIMING_	Change in menus should	UR_PLAYABLE,	Screen Changes	Assumes hardware is

OF_MENU_CHANGE	happen in reasonable time	UR_GAMETIME	< 1s	capable of running a game at basic requisite speed
NFR_REALTIME_GAMESCREEN	The game screen is updated in real time	UR_TRANSPORT	Screen Updates at least once every 1ms	Complex processes could take a long time to cause change on the screen
NFR_INPUT_CONSTRAINT	System can handle simultaneous inputs	UR_TRANSPORT	All navigation controls can be used simultaneously	When two contradicting movements are being made simultaneously assume the process of whichever was made first.
NFR_LEVEL_MAINTENECE	Levels will be easy to develop	UR_Difficulty	Platform builder provides an interface to remove technical jargon	The platform builder tool may not be intuitive to all technicians.
NFR_RESILIENT_INPUTS	Game will accept alternative hardware technology (e.g. touch pad/mouse)	UR_PLAYABLE	System accepts 100% of tested cursor technology	Assume hardware has actionable left and right clicks
NFR_OPERABLE	System is intuitive to all users	UR_PLAYABLE	System will be operable by users with no training	Assume that users have seen similar programs.
NFR_USEABLE	All user-facing messages ar in plain english and don't include any technical jargon	UR_PLAYABLE	100% of text is in plain english	Assume the user can interpret english.

System Requirements - Constraint Requirements

Requirement ID	Description	Risks/Environmental assumptions/Alternatives	Constraint Type
CR_PROJECTION	Project must be complete within 3 months	Project is not fully implemented by deadline	Project Constraint, Timing
CR_STAKEHOLDERS	Stakeholders must be convinced about any assumptions that we've made about the game.	Stakeholders could be displeased with aspects of the game which will take time to change	Design Constraint, Approval

Bibliography:

[1] https://www.interaction-design.org/literature/article/kiss-keep-it-simple-stupid-a-design-principle

[2] https://ecomputernotes.com/software-engineering/softwarerequirement

[3] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=741940