Requirements

Team 21
Zain Alshaikh
Corin Bertrand
Damien Heaton
Mandy Li
Brandon Oliver
Tom Tafijs

Elicitation

The requirements are defined according to the IEEE standards. We also referred to other sources such as Uptech's article for research purposes. Firstly, we went through the explicit requirements listed in the brief and noted any features that were vaguely described or we felt unclear on. With this in mind, we created questions and sought further clarification during our client meetings. These discussions also shed light on the customer's preferences for certain elements which weren't explicitly defined in requirements brief, such as the UI and game style.

As before, we split the requirements in three sections:

User requirements

1. User requirements are designed for non-technical people and relates directly to the actions that the user will take and its associated environmental assumptions.

System requirements

2. Functional requirements encompass the necessary behaviour of the system, and is represented by inputs, operations and outputs.
3. Non-functional requirements are quality constraints, which allow us to judge the performance and usability of the system.

We aimed to keep the table layouts as similar as possible for easier comprehension. In the UR table, each requirement is assigned an ID and has a corresponding description and priority level. In the FR table, each requirement is linked to one of the user requirement IDs as appropriate, and this replaces the priority column. The NFR table is the same as FR, with one additional "fit criteria" column which defines how we can measure the success of a requirement.

User Requirements

ID	Description	Priority
UR_PERFORMANCE	The game should perform smoothly on all hardware	M
UR_FUN	The player should leave the game with a positive mindset	H
UR_WIN_TIME	Game is winnable in 5-10 mins with no experience - Defeating the central college - Final objective not immediately achievable	H
UR_DEFEAT	Player can be defeated: - Time runs out - Combat - Sufficient collisions with other ships	H
UR_CHILD_FRIENDLY	Game is child-friendly (No blood / violent content)	H

UR_ACCESSIBILITY	Accessibility standards, objects are distinguishable by more than just colour	H
UR_COMBAT	Gameplay includes combat with colleges	H
UR_NPC_BOATS	Gameplay includes other boats sailing around	M
UR_SAIL	Player can sail around the lake	H
UR_SPEND_XP	Player can use XP for upgrades (upgrading stats) (done anywhere at any time)	H
UR_EARN_XP	Player can earn XP and Gold over time or defeating colleges	M
UR_PLATFORMS	Game is able to run on different platforms	M
UR_SCALABILITY	Scalability of the game screen for different resolutions	L
UR_MAKE_ALLIES	Player can make other colleges allies after they've been defeated	L
UR_COLLEGES_AMOUNT	There are at least three colleges (including central college and the player's college)	H
UR_LEVELS	The player can select the difficulty when starting a new game: Easy, Normal, Difficult	M
UR_SAVE_GAME	The player can save the game state at any point by quitting the game, which creates a local save file	M
UR_LOAD_GAME	The player can load a previously saved game state if a local save file exists	M
UR_OBSTACLES	The player should come across obstacles such as floating ducks, debris and shipwrecks	H
UR_WEATHER	The player may experience bad weather which slows it down and degrades the boat's health	M
UR_SHOP	The player can access the shop menu via a keypress button	H
UR_SHOP_POWERUPS	The player can use gold to purchase powerups from the shop	H
UR_MAGICAL_POWERUPS	The player can obtain magical powerups found within the game world	M
UR_SHIP_COMBAT	The player can engage in combat with other enemy ships	H

System Requirements: Functional Requirements

ID	Description	User Requirements
FR_BOSS_DEFEAT	The system shall allow for the final central college to be defeated	UR_WIN_TIME
FR_BOSS_REQUIREMENTS	The system shall provide a barrier that is not easily penetrable that weakens as tasks are completed (colleges are defeated) in order to prevent the objective (central college battle) being immediately available to the player	UR_WIN_TIME
FR_ENEMY_COLLEGE	The system shall include at least 3 enemy colleges in the lake	UR_COLLEGES_AMOUNT
FR_PLAYER_COMBAT	The system shall allow the player to engage in combat with the enemy colleges	UR_COMBAT
FR_ENEMY_COLLEGE_DEFEAT	The system shall allow enemy colleges to be defeated	UR_COMBAT
FR_NPC_BOAT_MOVEMENT	The system shall have NPC boats sail around the lake	UR_NPC_BOATS
FR_NPC_BOAT_COMBAT	The system shall have NPC boats shoot projectiles at the player ship	UR_NPC_BOATS
FR_NPC_BOAT_COLLISION	The system shall ensure NPC boats avoid collision with islands and other NPC boats	UR_NPC_BOATS
FR_PLAYER_MOVEMENT	The system shall allow the player to sail around the lake	UR_SAIL
FR_PLAYER_COLLISION	The system shall detect when player is involved in a collision and affects the player accordingly	UR_SAIL, UR_OBSTACLES
FR_XP_EARN	The system shall allow for XP to be earned over time or through combat	UR_EARN_XP
FR_XP_SPEND	The system shall allow for XP to be spent for boat upgrades at any time	UR_SPEND_XP
FR_UPGRADE_IMPLEMENTATION	The system shall ensure that boat upgrades are implemented immediately	UR_SPEND_XP
FR_UPGRADE_CHOICE	The system shall allow the player to choose from an array of possible upgrades	UR_SPEND_XP

FR_GOLD_EARN_COMBAT	The system shall allow for gold to be accumulated from defeating colleges	UR_EARN_XP
FR_HEALTH	The system shall show a health count of the player’s/college’s current health	UR_COMBAT
FR_VICTORY	Once the player has completed the main objective they get taken to a victory screen	UR_VICTORY
FR_SHOP	The shop will offer three powerups that can be purchased for gold: repairing damage, speed boost, extra cannons	UR_SHOP, UR_SHOP_POWERUPS
FR_MAGICAL_POWERUPS	The system will allow the player to obtain two types of special powerups (temporary immunity & damage upgrade) within the game world by completing a certain task	UR_MAGICAL_POWERUPS
FR_OBSTACLES	The game should have obstacles in the water, which causes damage to the boat upon collision	UR_OBSTACLES
FR_WEATHER	Ships experiencing bad weather will slow down and take progressive damage over time (unless blessed with temporary immunity)	UR_WEATHER
FR_INCREASE_DIFFICULTY	As difficulty increases, the player ship should deal less damage via projectiles	UR_LEVELS
FR_SAVE	The system will automatically create a local save file when the game is quit that saves the current game state	UR_SAVE_GAME
FR_LOAD	The system will automaticallly load a saved game file upon start if it exists	UR_LOAD_GAME
FR_NO_SAVE_FILE	If there is no save file, the game starts anew	UR_LOAD_GAME

System Requirements: Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_AGE_APPROPRIATE	The game won’t include any content that could be harmful to children	UR_CHILD_FRIENDLY	Must be kept throughout the entire game

NFR_DISTINCT_BOATS	Enemy boats will be distinct from the player boat and one another	UR_NPC_BOATS	The player should always be able to tell the difference between boats
NFR_INPUT_LAG	There will be a minimal amount of input lag	UR_PERFORMANCE	95% of the time, input lag < 30ms
NFR_XP_GOLD_UPDATES	When XP/Gold score is changed it will be automatically updated on the player’s screen	UR_EARN_XP	95% of the time, XP update time < 50ms
NFR_PLATFORMS	The game should be able to run on multiple platforms	UR_PLATFORMS	The game should run on Windows, Linux, Mac (except M1)
NFR_SCALING	The game should be able to scale resolution without making the game look worse	UR_SCALABILITY	When the resolution changes, the game should remain the same quality
NFR_BOAT_ALLEGIANCE	When the college a boat is aligned with is defeated they become allies	UR_MAKE_ALLIES	Must happen no longer than 0.5s after a college is defeated
NFR_PERFORMANCE	The game should run smoothly	UR_PERFORMANCE	Framerate should be locked at 60 FPS
NFR_USABILITY	The game should be simple to understand wwith intuitive controls	UR_CONTROLS	95% of the time, the player should be able to familiarise themselves with the concept/controls in < 30s
NFR_EXPERIENCE	The game should be aesthetic and enjoyable to play	UR_FUN	90% of users are satisfied with the game and design

Environmental Assumptions, associated risk and alternatives

- NFR_PLATFORMS is based on the assumption that the game will be used by users whose device can support OpenGL, specifically the LWJGL3 library.
- There is a substantial risk that any of the functional requirements may not be met due to R10, updates to LibGDX, causing our code to become obsolete and unusable.
- An alternative perspective/projection style for the game, isometric, was discussed with the customer but was dismissed because a top down approach was preferred.
- There is a possibility that FR_PLAYER_COLLISION may not be met due to lack of previous experience of coding polygons in Java

References:

https://ieeexplore.ieee.org/document/278253
https://www.uptech.team/blog/functional-vs-non-functional-requirements
I.Sommerville, Software Engineering, Pearson Education 2008