Requirements

Mario

GROUP 28

Joseph Frankland

Anna Singleton

Saj Hoque

Leif Kemp

Shi (Lucy) Li

Hugo Kwok

2(a) Introduction

These requirements were gathered from a combination of the original product brief we received and an interview we conducted with the customer to gain clarification on some points within the original brief. Before consulting our stakeholders we communicated the requirements to each other so that each member adequately understood the context of the system we would be building and the development process for such a system. We ensured to seek agreement from all stakeholders as well as each team member in order to avoid misinterpretations of the product brief whilst also aligning all of our expectations of the product. One example of a specific requirement we elicited was how to distinguish between colleges, which we discussed with the customer and came to the conclusion of one suitable method. We plan to continue questioning the customer when we approach any difficulties with our planning / implementation. Once we had a good idea of what the customer required we set out creating the user and system requirements. This allowed us to further break down the system requirements in which we made a distinction between functional and non-functional requirements.

We organised the requirements into three tables: User, functional, and non-functional. We provided columns for the Unique ID and description for each table as it is essential they are all uniquely identifiable as this allows us to refer to the requirements in a different context as well as make connections between the different types of requirements. The user requirements table should also have a priority column to indicate how urgent and important each requirement should be fulfilled. This will give us an indication of which feature needs more attention when it comes to the implementation stage. The two system requirements tables will each have a column to link back to the corresponding user requirement as each system requirement is invoked from a specific user requirement. Finally, the fit criterion column for non-functional requirements is used for measuring the success of each requirement in order to aid us when it comes to the developing stage.

2(b) User Requirements

User Requirement ID	Description	Priority
UR_WIN_TIME	Game is winnable in 5-10 mins with no experience - Defeating the central college - Final objective not immediately achievable	Shall
UR_DEFEAT	Player can be defeated - Time runs out - Combat - Sufficient collisions with other ships	Shall
UR_CHILD_FRIENDLY	Game is child-friendly (No blood / violent content)	Shall
UR_ACCESSIBILITY	Accessibility standards, Objects are distinguishable by more than just colour	Shall
UR_COMBAT	Gameplay includes combat with colleges	Shall
UR_NPC_BOATS	Gameplay includes other boats sailing around	Should
UR_SAIL	Player can sail around the lake	Shall
UR_SPEND_XP	Player can use XP for upgrades (upgrading stats) (done anywhere at any time)	Shall
UR_EARN_XP	Player can earn XP and Gold over time or defeating colleges	Shall
UR_PLATFORMS	Game is able to run on different platforms	Shall
UR_SCALABILITY	Scalability of the game screen for different resolutions	Shall
UR_MAKE_ALLIES	Player can make other colleges allies after they've been defeated	May
UR_COLLEGES_AMOUNT	There are at least three colleges (including central college and the player's college)	Shall

2(b) Functional Requirements

Functional Requirements ID	Description	User Requirements ID
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	UR_WIN_TIME

	objective (central college battle) being immediately available to the player	
FR_ENEMY_COLLEGE	The system shall include at least 3 enemy colleges in the lake as well as allow the player to engage in combat with them	UR_COMBAT, 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 but not allow them to engage in combat with the player	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
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

2(b) Non-Functional Requirements

NF Requirement ID	Description	User Requirement	Fit Criterion
NFR_VICTORY	Once the player has completed the main objective they get taken to a victory screen	UR_VICTORY	Must happen every time the player wins the game
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_ACCESSIBILITY	The game will have some options to help with accessibility	UR_ACCESSIBILITY	Available in the menu and with possibility to be turned on and off
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 differences between boats
NFR_INPUT_LAG	There will be a minimal amount of input lag	UR_SAIL	Input lag will never be over 30ms
NFR_XP_GOLD_UPDATES	When XP/Gold score is changed it will be automatically updated on the player's screen	UR_EARN_XP	XP change will never take more than 50ms
NFR_PLATFORMS	The game should be able to run on multiple platforms	UR_PLATFORMS	The game should run on most OS
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

2(b) Constraints

We have a few constraints to work within, our main constraint is the time we have to complete the project, the deadline is February 2nd so we have a very limited amount of time to get this done. This will be managed through frequent communications and through organising deadlines for each aspect of this project. The project must also run on multiple platforms, however, this constraint shouldn't give us too much trouble since we are creating the game in Java which is platform independent. Licensing of code bases we use could also constrain us, we have to make sure we are allowed to use the code for what we want and follow rules like making it open-source if it is required.

2(b) 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