



## GROUP 1

ÁNGEL NICOLÁS ESCAÑO LÓPEZ

RICARDO JUAN ALLITT LÓPEZ

MANUEL BARRIOS MORENO

MUHAMMAD ABDULLAH SULTAN

PABLO HORMIGO JIMÉNEZ

DIEGO SICRE CORTIZO

ÁNGEL BAYÓN PAZOS

FRANCISCO JAVIER JORDÁ GARAY

JUAN TORRES GÓMEZ

ANGELESC04@UMA.ES

RICARDOALLITT@UMA.ES

MANUELBARRIOS@UMA.ES

MASU@UMA.ES

PABLOHORJIM@UMA.ES

0611146788@UMA.ES

ABAYPAZ@UMA.ES

FRANCISCO.JORDA@UMA.ES

JTORRESGOMEZ26UMA@UMA.ES

GitHub Repository  
<https://github.com/ProyectoXMA/UMA-ISE24-E1>



## Table of Contents

Table of Contents.....	1
Section 1: Introduction .....	2
Section 2: Roles.....	3
Section 3: Risk Management .....	4
Section 4: Planning .....	6
Section 5: Software tools .....	9
5.1 Communication.....	9
5.2 Project planning & productivity.....	9
5.3 Development related .....	10
5.4 Design .....	10
5.5 Documentation .....	11
Section 6: Requirements.....	12
Section 7: Requirements diagram .....	15



## Section 1: Introduction

This section serves as a small introduction to our project, a videogame called "Dragon Boat Racing" as specified by our client Mr. Javier Cámara. Dragon Boat Racing is a 2D, single-player game developed for the PC platform using the Java programming language.

For the gameplay aspects, Dragon Boat Racing is an arcade racing game where the player will experience the rush of competition, the excitement of power-ups, the satisfaction of overcoming obstacles and most importantly the glory of winning a race. It consists of various levels, each one with increased difficulty and unique scenery. Players will have the ability to customize their boat's appearance and stats to match their playstyle.

Every race is an adrenaline-fueled test of skill and strategy as it takes great skill for a player to control their boat in the lane and defeat their opponents. The game's mechanics are basic yet extremely exciting. Leaving the lane penalizes you making it harder for you to win the race and colliding with obstacles eventually results in the sinking of your boat. It is here that the game will give less skilled players a chance to play a "Simon-says" mini-game to repair and revive their boat. The different power-ups make the experience more exciting, enjoyable, and dynamic.



## Section 2: Roles

To achieve our goal of creating an excellent, enjoyable, and exciting game, it is important that we distribute the work among our team. This is also essential for satisfying the needs of our client, Mr. Cámara, and delivering a great final project.

Keeping in mind the project's needs we have decided to have five roles. They are as following:

- **Programmers** will be responsible for writing all the code and carrying out all the tasks related to development.
- **Project Managers** are there to prepare our Trello, keeping an eye out on the project to make sure that we are advancing and most importantly they are the coordinators of the project, this means that they will make sure that we meet all our deadlines.
- **Spokespersons** will be responsible for coordinating and establishing a healthy relationship with our client. Their main job is to act as a communication channel between our team and Mr. Cámara.
- **Testers'** job is essential for any programming project. They will make sure that the code functions as intended and will give their feedback to the programmers making sure that the final product has no unexpected bugs.
- **Graphic Designers** will design all the graphical elements for our project. Be it mockups, different elements of the game like boats, levels and loading screens. Their job is essential to our project as it will ensure that the final project has stunning visuals.

Next up you will find a table listing all our team members with their assigned roles. Each team member has been assigned two roles to ensure each role has sufficient assignees.

Team Member	Role 1	Role 2
Allitt López Ricardo Juan	Spokesperson	Tester
Barrios Moreno Manuel	Project Manager	Graphic Designer
Bayon Pazos Ángel	Tester	Graphic Designer
Escaño López Ángel Nicolás	Spokesperson	Programmer
Hormigo Jiménez Pablo	Project Manager	Programmer
Jorda Garay Francisco Javier	Spokesperson	Programmer
Sicre Cortizo Diego	Project Manager	Tester
Sultan Sultan Muhammad Abdullah	Programmer	Graphic Designer
Torres Gómez Juan	Tester	Graphic Designer



## Section 3: Risk Management

In this section we introduce the main risks we are facing with regards to the project. We have created a table in which we specify the risk, it's type, a little description in which we present it and we have added a metric of the likelihood of it happening and the severity it presents, all of them are followed by the consequent contingency plan we would apply in case the risk materializes.

It is fundamental for the project to succeed to have an appropriate anticipation of the possible things that could go wrong and the way in which we plan to fix them. This will allow us to anticipate the problems before they occur and will allow us to save valuable resources.

Risk	Type	Risk Description	Severity / Probability	Contingency Plan
Underestimating complexity of a task	Organization	If a given task is underestimated, it may lead to delays in the project and a need for reorganization of tasks.	Medium-High Medium-Low	Redistribute the work force to rapidly get back on track and minimize the effects.
Lack of risk management	Organization and requirements	Failure to identify and proactively manage risks can lead to surprises during project development, resulting in delays, additional costs, and loss of client confidence	Very High Low	Develop a good study plan that encompasses all the risks that may arise during work, from the simplest and least probable to the most complex, to avoid surprises
Finished task is not what was expected	Project and Personal	Failure in communication between the project team and the client leading to an incorrect product.	Medium Low	Replan task with team making sure everyone understands the objectives needed and recommunicating with the client
Other assignments demand our time above the project.	Personal and Organization	A bad planning of personal time results in overlapping of other projects and deadlines that require our attention before this project.	Medium Medium	In a group meeting try to delegate such tasks to other members; this project must be the priority.
A group member leaves the project, is not disponible or does not dedicate enough amount of time	Project	The project might lack in a certain area due to the mediocre work of a team member	Very High Very Low	The project is a team effort, where we must all work equally to move forward, actions will be taken against this team member
The final product could fail and not perform as expected	Product	An unchecked command could make the difference between a correct and a completely disastrous delivery, leading to errors unknown for the group members	Catastrophic Low	Giving enough time to test the final delivery before the submission, considering all cases and their respective consequence (In case x happens, then each of the results y must be contemplated)



Requirements change mid project production	Project and Requirements	The customer could always ask for a substantial change mid project leading to massive changes	Medium-High Medium-Low	Try to adapt our Gantt Diagram to meet the latest changes with minimum change during the project.
Personnel without sufficient information on the tools to use	Personal and Organization	The lack of experience with the java programming language and its corresponding graphic interfaces could mean spending excessive time learning it and solving errors, thus subtracting time dedicated to other tasks	High-Very High Medium-High	When a doubt arises, asking someone from other departments of the project or another more knowledgeable programmer
Develop the project with incorrect focus	Project and Organization	We follow methods or focus a lot of resources in making something that in the end was not necessary	High Medium-Low	During each session revise the work done, work that will be done, and debate whether it helps us reach the desired goal and not the opposite.
Not understanding what the client wants	Project and Requirements	Our understanding of the client's vision and ours aren't in sink	Catastrophic Low	Maintaining active communication with our client and showing results, if possible, an effective way to keep track of the development of the project.
Sudden change in development environment	Technology	The libraries being used for the project could change leading to the need to adapt changes or completely restart the project.	Medium-High Very Low	Before redoing the project, we will consider other technologies that adapt to what we have developed and evaluate if it is better to migrate the project or adapt to changes.
Group members disagree between them	Personal	At the moment of implementing a feature or discussing an idea, teammates may differ in opinions, even resulting in conflict.	Medium Medium	Be open-minded and learn to accept different points of view. Take the opportunity to learn from each other.
Client no longer needs our help	Project	The client may cancel the project mid production for any reason.	Catastrophic Very Low	There is nothing we can do at this point



## Section 4: Planning

In this section, we adopt the SCRUM methodology and agile practices to foster iterative development, adaptability, and continuous improvement throughout the project lifecycle. SCRUM emphasizes teamwork, collaboration, and delivering incremental value to stakeholders through short, time-boxed iterations called sprints. Our approach to project planning involves defining and prioritizing tasks, setting achievable goals, and allocating resources effectively to maximize productivity and project success. This robust planning will help us achieve our goals and make sure that we satisfy the needs of our client, Mr. Cámara.

- **(March 8 - March 13):** Dedicated to discussions regarding the video-game implementation, creating a list of questions to present to the client on March 13. These questions cover aspects such as boat movements, obstacle generation, boat skins, and levels. Additionally, communication channels through Discord were established, along with task management using Trello and GitHub integration.
- **(March 13 - March 22):** With a more refined concept, we implemented the requirement diagram, risk management, role assignment, and future planning. Several meetings were conducted via Discord to oversee the progress of these tasks and their respective reviews.
- **(March 23 - April 1):** Break for Holy week.
- **(April 2 - April 7):** Devoted to meetings between programmers and testers, consulting the proposed Java library and necessary documentation, as well as defining and developing initial programs and functions. Graphic designers convened to sketch and specify the number of boats, obstacles, and level skins. A general meeting was also held to consolidate ideas and review progress across the different sectors.
- **(April 7 - April 21):** Beta versions of the game were archived, featuring boat movement, obstacles, and basic skins for boats, obstacles, and levels.
- **(April 22 - May 6):** Improved versions of the game incorporating physics collisions with obstacles and boats, diverse attributes for boats such as handling or speed, even the increase of difficulty across levels, random object appearances, and mini-games. Graphic developers expanded the repertoire of boat and level skins.
- **(May 6 - May 13):** Final versions from developers with the latest implementations and improvements, potentially including the integration of power-ups during gameplay or enhanced developer-friendly controls. Graphic designers introduced a degraded version of boats to depict damage accumulation, upgrading previous iterations, as necessary.





- **(May 13 - May 31):** Dedicated to testing the final game, with testers assuming responsibility for this phase. Programmers focused on bug fixes in the code, without introducing additional mechanics. Delivering the project to the client and asking for their feedback.

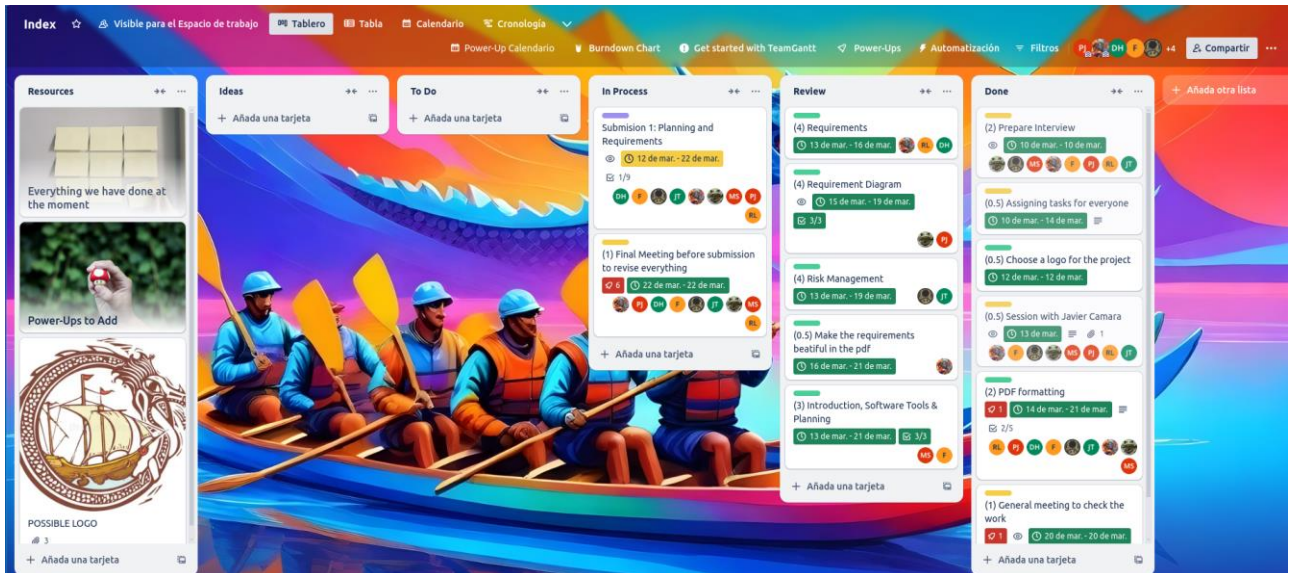


Figure 1: Trello Board

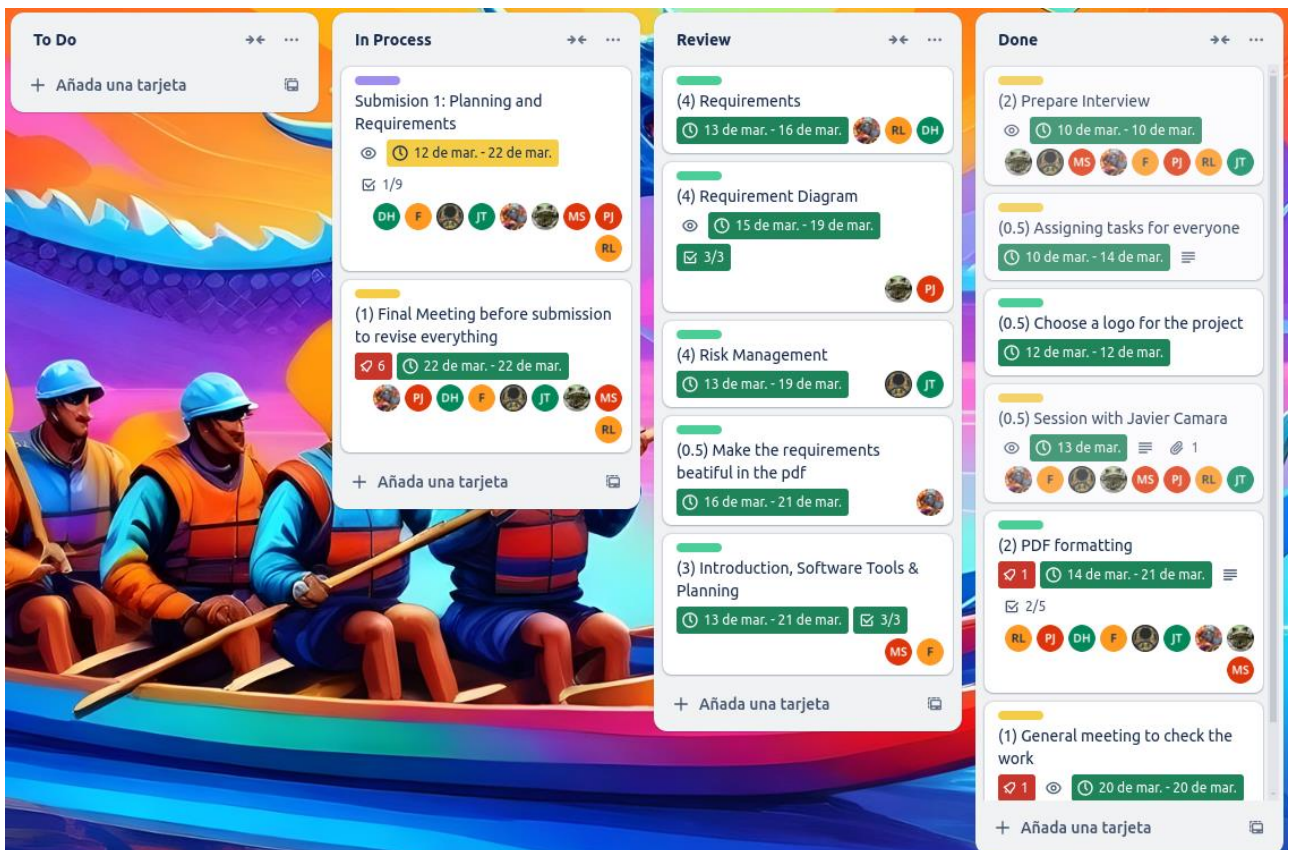


Figure 2: Trello charts for the organization



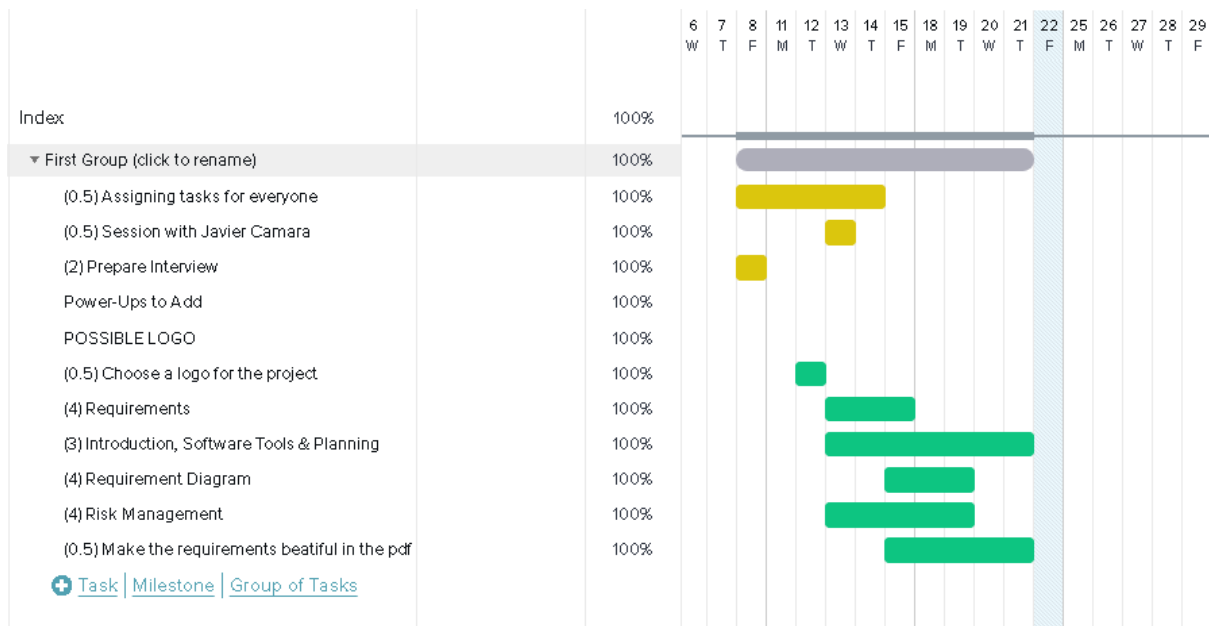


Figure 3: Gantt Chart of the first part of the project.



## Section 5: Software tools

Every project and team need different tools. Some tools are more suitable for the job than others. Here we will list the tools that we have chosen for our Dragon Boat Racing project after carefully looking at all the options, consulting with different team members and of course making sure that they are the best tools for the job.

### 5.1 Communication

This section outlines the tools we utilize for seamless communication within the team. Effective communication software is crucial for real-time collaboration, swift issue resolution, and maintaining a cohesive team environment. These tools facilitate instant messaging, and videoconferencing ensuring that team members stay connected regardless of geographical locations.

- **Discord** will serve as our main communication platform. The app is great for videoconferencing between our team members, has a modern interface and mobile applications to make sure our team is always connected.
- **WhatsApp** will serve as another communication platform for its convenience, and its instant messaging capabilities as anything critical can be discussed via WhatsApp between our team members for instant response.

### 5.2 Project planning & productivity

This section enumerates the tools we employ for project planning, task management, and enhancing overall productivity. These tools are essential for organizing workflows, setting project milestones, allocating resources, tracking progress, and managing timelines efficiently. By utilizing such tools, we ensure transparency, accountability, and alignment with project objectives, optimizing team performance and project outcomes.

- **Trello** will be used exclusively as it is very well known in the industry and of course preferred by our project managers. It will make it easy for us to assign tasks and distribute work between our members thanks to its excellent, simple yet powerful web interface.
- **ChatGPT** is an excellent modern AI tool which will help boost the productivity of every one of our members. Be it helping our programmers write tests and similarly helping us document our code due to its great language capabilities.



### 5.3 Development related

This section delineates the software tools integral to the development process, including integrated development environments (IDEs), and version control systems. These tools facilitate code creation, debugging, version tracking, and testing. By leveraging these tools, we enhance code quality, accelerate development cycles, and foster collaboration among our team's programmers.

- **IntelliJ IDEA** is an IDE very well known in the java development space. It is modern, simple, powerful and above all our team of developers is comfortable using it. It has support for different build tools like **gradle** which we will use for our Dragon Boat Racing game.
- **Git** is an excellent version control system and is an industry leader. It is fully free and open source and will give us access to powerful version controlling. As for a git server, we will use **GitHub** simply because it does not require us self-hosting our git server, is easy to use and provides an excellent desktop client for us to use.

### 5.4 Design

This section highlights the tools utilized for graphical design, prototyping, and user interface (UI) development. These tools enable our graphic designers to conceptualize, iterate, and refine design elements, ensuring intuitive user experiences and visually appealing interfaces. From wireframing and mockup creation to asset management and prototyping, these tools play a pivotal role in translating design concepts into tangible product features, aligning with our client Mr. Cámara's expectations.

- **Visual Paradigm** is versatile tool for UML diagramming and requirements definition. It offers a user-friendly interface and a wide range of features for creating diagrams like use case, class, and sequence diagrams. Facilitating collaborative design and analysis ensures efficient communication and alignment of design concepts between us and our client, Mr. Cámara. It is simply put ideal in our case of agile development.
- **Canva** is an excellent modern graphic design tool which serves to create simple mock-ups as well as different marketing and mock-up material when working in our team.
- **Adobe Creative Cloud** is a comprehensive suite of creative tools for graphic design, photo editing, video production, and web development. Adobe Creative Cloud includes industry-standard software such as **Photoshop** and **Illustrator** which will be used by our professional



graphic designers to design different graphical elements of the game be it the simple game menu or complex boat designs.

- **Freepik Pikaso** is an AI-powered art generator that revolutionizes the creative process. Leveraging advanced algorithms, Pikaso transforms concepts into stunning visual artworks effortlessly. With its intuitive interface and vast library of styles and elements, Pikaso will empower our graphic designers to reach their maximum potential be it generating photos of different game elements or simply using it to get inspired when they are out of ideas.

### 5.5 Documentation

This section identifies the tools employed for documenting project requirements, specifications, codebase, and user documentation. Effective documentation tools facilitate knowledge sharing, maintain project transparency, and serve as invaluable references for our team and are important for our client Mr. Cámara.

- **Google Docs** is a cloud-based document collaboration platform offering real-time editing and sharing capabilities. Google Docs enables our team members to work concurrently on documents, facilitating seamless collaboration regardless of geographical locations.
- **Microsoft Office** is a suite of productivity tools including Word, Excel, and PowerPoint, providing robust capabilities for document creation, data analysis, and presentation. Microsoft Office offers a familiar interface to our team members and is an industry standard.

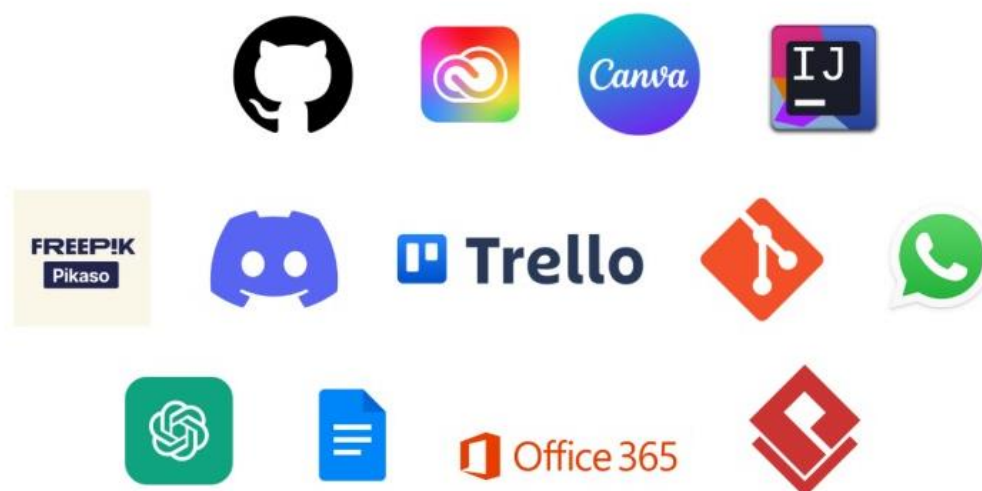


Figure 4: Best software tools for job



## Section 6: Requirements

In this section, there will be explained all the requirements that this project must hold for its correct functionality, those whose color is green, are meant to be the functional requirements, those who explain what the project must do. Furthermore, those whose color is yellow will be denoted as the Non-Functional Requirements, that are explained as the qualities that the project holds itself.

### FR1: INITIALIZE THE GAME

As a player I want the game to start whenever I want to play it

### FR2: BOAT CONTROL

As a player I want to be able to configure my own key binds and I want the controls to be responsive with low latency, in order to have a more personalized and pleasant experience.

### FR3: POWER UPS

As a player I want to have powerups that modify the way the game is played, to have as different an experience as possible in each run.

### FR4: LEVELS

As a player I want to have various levels in order to get a sense of progress while I'm playing and to raise the stakes whilst I advance in the game.

### FR5: BOAT SPRITE

As a player I want to be able to modify the way my boat looks to avoid a repetitive and dull gameplay.

### FR6: DIFFICULTY

As a player not having an incrementally more difficult gameplay would make the gaming experience bland and boring. This goes together with the levels functional requirement

### FR7: OBSTACLES

As a player I want to have different obstacles positioned in various places for the game to be challenging and engaging.

### FR8: SCENERY

As a player I want to experience a visual change once I advance through the levels, to have a notion of progress and risk.

**FR9: AFTERLIFE**

As a player I would like to have an opportunity to come back if I die as it would make it more engaging and fun to play.

**FR10: BOAT LANES**

As a player I want to have a boat lane in which I must stay during the race to reduce chaos if there are multiple boats on the screen.

**FR11: RIVALS**

As a player I want to be competing against other alleged players, for the game to have a little of competition.

**FR12: PENALTY**

As a player I want to be punished and rewarded accordingly, for the game to have a bit of edge.

**FR13: EXITTING THE GAME**

As a player I want to be able to move fast between screens and to exit the game seamlessly to avoid annoyance.

**FR14: BOAT STATS**

As a player I want to be able to choose boats with distinct characteristics that add depth to the way the game is played.

**FR15: SIMON SAYS MINIGAME**

As a player I want to have a minigame inside the main game to have a more complete experience.

**FR16: LOADING SCREENS**

As a player I want to have an enjoyable experience while I wait for the game to load.

**FR17: STUN TIME**

As a player I would like to have a fair crash system that 'stuns' me the right amount of time to have a challenging but possible game.

**NFR1: IMAGE REFRESH RATE**

As a game developer, I want the game to execute at a flawless 30 fps for the experience to be more pleasant.



**NFR2: LOW LATENCY AND KEYBINDS**

As a game developer I want the controls to be responsive and accurate for the player to have a smooth experience.

**NFR3: AVOIDING SYSTEM ERRORS**

As a developer I want the code to be robust and free of errors to provide the player with a satisfactory experience.

**NFR4: DIFFICULTY AND RANDOMNESS**

As a developer I want the experience to be easy and simple at first but harder as the player advances through the levels to have an entertaining game for all publics but a challenging one for the adventurous.

**NFR5: JAVA LANGUAGE**

As a developer I want to use the language in which all my team is experienced at.

**NFR6: LIBRARIES**

As a developer I want to use the same libraries that have been used for similar projects that have been successful

**NFR7: COLLISIONS**

As a developer I want the game to have fair hit boxes for the game not to be frustrating.

**NFR8: SYSTEM FAILURES**

As a developer, I do not want to deliver a definitive version of the project that could contain fatal errors making the game not playable.

**NFR9: AI**

As a developer I want to implement an ai to make competition between different players and make the game more interesting.



## Section 7: Requirements diagram

For every project which has requirements to classify, there must be a diagram that shows all the relations between Functional Requirements and Non-Functional ones, this part consists of the definitive version of the diagram with all the requirements of this projects and how everyone affects others.

