

# DRAGON BOAT RACE



Jorge Frías Tello - jorgegemanon@gmail.com

Cristina Montañés Peña - cristinamonpe2004@uma.es

Alma García Ramírez - almagarciaramirez@uma.es

Carmen M. Fernández Ferrer - carmenmariafernandez@uma.es

Antonio Manceras Gámez - amanceragamez@gmail.com

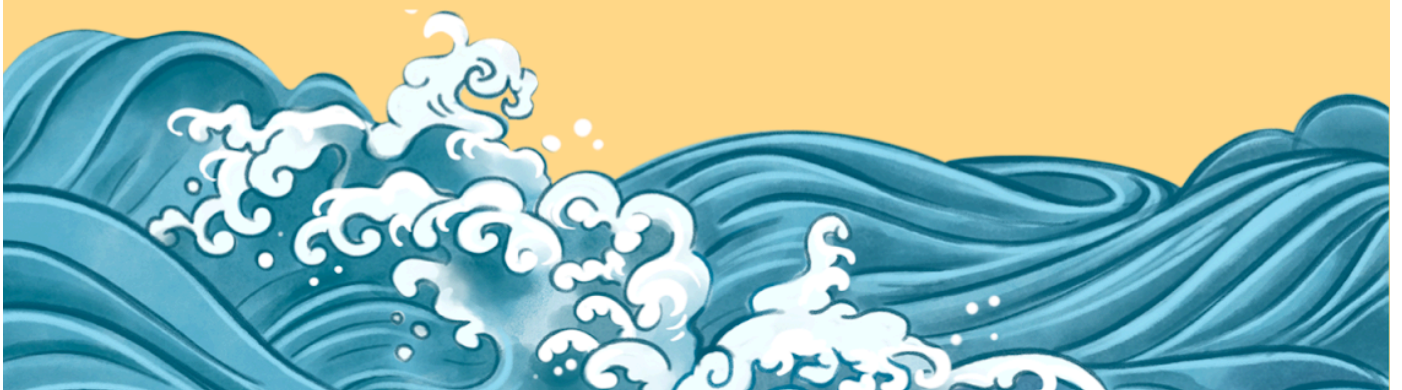
Gabriel del Corral Velasco - gabidcor@uma.es

Bianca Valentina Dinu - biancadinu10@gmail.com

Jean-François Senécal - jeanjeansenecal@gmail.com

Nicolás López - n.lopez9418715@gmail.com

**<https://github.com/jorgefriast/UMA-ISE24-E5>**



1. INTRODUCTION.....	3
2. ROLES.....	3
3. RISK MANAGEMENT.....	4
4. PLANNING.....	5
5. SOFTWARE TOOLS.....	6
6. REQUIREMENTS.....	8
7. REQUIREMENTS DIAGRAM.....	11
8. USE CASES.....	14
9. DOMAIN MODEL.....	16
10. SEQUENCE DIAGRAM.....	17

## 1. INTRODUCTION

Our project, "Dragon Boat Race," aims to simulate traditional dragon boat racing in a single-player gaming experience. Inspired by the historical and cultural significance of this ancient sport, our team is working to develop a software system that provides an engaging and realistic simulation.

Through careful planning and design, we aim to create a game that accurately portrays the challenges and dynamics of dragon boat racing. From managing boat specifications to navigating obstacles and fatigue, our game will offer players an immersive experience as they compete for victory on the virtual river.

## 2. ROLES

### UI/UX Designer & Graphics Artist:

This role involves crafting the visual experience of the game. This includes designing the user interface (UI) to ensure it's intuitive and visually appealing, as well as creating graphical assets such as character designs, backgrounds, and animations to bring the game to life.

Carmen Maria Fernandez Ferrer and Cristina Montañés Peña will be in charge of carrying out this task.

### QA Tester:

The members with this role will be responsible for ensuring the quality and integrity of the game. As a QA tester, they'll thoroughly test the game, identifying and reporting any bugs or issues that may arise.

Jorge Frías Tello and Bianca Dinu will be responsible for this task.

### Coders:

Their primary responsibility will be developing the game using Java programming language. This includes implementing game mechanics, integrating various components, and ensuring smooth functionality across different platforms.

Gabriel Del Corral Velasco, Antonio Mancera Gamez, Jean-François Senécal and Nicolás López will be responsible for this task.

**Documentation Specialist:**

They'll be tasked with writing all necessary documentation for the project. This role is crucial for maintaining clear communication and ensuring the project's progress is well-documented and accessible to all stakeholders.

Alma García Ramírez will be responsible for this task.

**3. RISK MANAGEMENT**

RISK	TYPE	DESCRIPTION	PROBA-BILITY	EFFECTS	MITIGATION STRATEGY
Technical Risks	Technical	Potential challenges with implementing complex game mechanics, integrating different software components, or ensuring compatibility across platforms.	Moderate	Serious	Regular code reviews, continuous integration, and thorough testing to identify and address technical issues early. Additionally, maintaining open communication among team members to quickly address any technical challenges that arise.
Schedule Risks	Project Management	Delays in development, unexpected setbacks, or changes in requirements that impact the project timeline.	High	Serious	Regular monitoring of project progress, frequent updates to stakeholders, and agile development methodologies to adapt to changes efficiently. Establishing contingency plans and allocating buffer time for unforeseen delays.
Commu-nication Risks	Team Dynamics	Miscommunication, conflicts among team members, or breakdowns in collaboration that hinder project progress.	Moderate	Serious	Establishing clear channels of communication, setting expectations for team interactions, and promoting open dialogue among team members. Implementing regular team meetings, progress updates, and conflict resolution mechanisms to address any communication issues promptly.
Quality Risks	Product Quality	Potential issues with product quality, such as bugs, glitches, or user experience shortcomings, that affect the game's usability and appeal.	High	Serious	Implementing rigorous testing procedures, including unit testing, integration testing, and user acceptance testing, to identify and address quality issues early. Soliciting feedback from stakeholders and end-users throughout the development process to ensure that the product meets quality standards and user expectations.

## **PLANNING**

### **Planning and Preparation Phase (Week 1-2):**

Define project scope and objectives based on the requirements provided by the instructor.

Assign roles and responsibilities to team members, considering their strengths and interests.

Conduct initial meetings to discuss project requirements and establish a shared understanding of the goals.

Create a backlog of tasks and features to be implemented, outlining the prioritization based on project deliverables and deadlines in Trello.

Set up communication channels such as WhatsApp for informal communication and coordination.

Hold regular check-ins or meetings to ensure everyone is on track and address any questions or concerns.

### **Documentation and Design Phase (Week 2-7):**

Develop use cases, class diagrams, sequence diagrams, and other necessary documentation as outlined by the project requirements.

Conduct regular meetings to review and refine the documentation, incorporating feedback from the instructor or peers if necessary.

Utilize tools like Google Docs for collaborative document creation and sharing.

Begin designing the game interface and graphics, considering the visual appeal and user experience.

### **Development and Testing Phase (Week 7-11/12):**

Transition into the development phase, focusing on implementing the game features and functionalities according to the documented requirements.

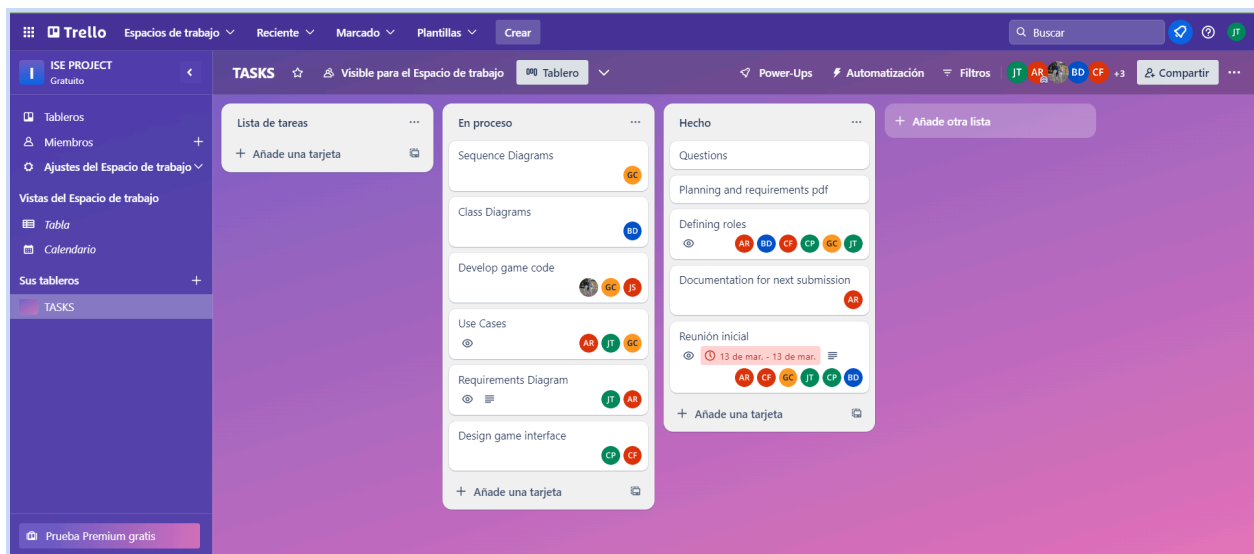
Break down development tasks into manageable chunks, considering the skills and experience level of team members.

Conduct testing and debugging of the game code, identifying and addressing any issues or bugs.

Allocate time for final testing and quality assurance before submitting the project for evaluation.

Prepare a demonstration or presentation of the completed game for the instructor and peers, showcasing its functionality and features.

Throughout the project, we integrated Scrum practices to manage development effectively. This included conducting Sprint Planning meetings to prioritize tasks from the backlog using tools like Trello. We broke down tasks into manageable chunks and held stand-up meetings to discuss progress, address obstacles, and plan tasks. At the end of each sprint, we conducted Sprint Review meetings to demonstrate completed work and gather feedback. Embracing an iterative approach, we made adjustments to plans and strategies as needed, prioritizing flexibility and continuous improvement to optimize workflow and productivity. Overall, implementing Scrum facilitated effective collaboration, priority management, and the delivery of a high-quality product meeting stakeholder needs.



## 4. SOFTWARE TOOLS

Throughout the development of "Dragon Boat Race," our team utilized a variety of software tools to facilitate communication, collaboration, and project management. Below is a list of the key tools we employed:

**Google Docs:** It served as our primary platform for collaborative document creation and editing because it allowed team members to work together in real-time on project documentation, such as meeting agendas, minutes, and progress reports.

**Microsoft Word:** It was used for creating formal project documents due to its robust formatting features and compatibility with other Microsoft Office applications. This ensured the production of professional-quality documents such as the project proposal, requirements specification, and risk management plan, meeting the standards expected for academic and professional presentations.

**WhatsApp:** WhatsApp served as our main communication tool for quick and informal exchanges among team members. We utilized group chats to share updates, discuss project-related matters, and coordinate tasks in real-time, enhancing team collaboration and responsiveness.

**Gmail:** Gmail was our primary email platform for formal communications with stakeholders, including the project instructor and university representatives. It facilitated seamless communication, file sharing, and scheduling of meetings, ensuring effective and efficient correspondence throughout the project lifecycle.

**Trello:** Trello was used as our project management tool for organizing tasks, tracking progress, and managing project workflows. Its visual interface provided an overview of project status, allowing us to prioritize tasks, assign responsibilities, and monitor deadlines effectively. Trello's flexibility and customization options adapted well to our agile development approach, facilitating task organization and team coordination.

**Visual Paradigm:** It was used for creating the requirements diagram, aiding in the visualization and documentation of the relationships between project requirements. Its features enabled us to communicate project scope and objectives effectively, facilitating understanding and alignment among team members and stakeholders.

**JUnit:** It was required for testing the behavior of classes and methods. Its use ensured the reliability and correctness of the codebase by automating unit tests, identifying defects early in the development process, and supporting code refactoring and maintenance efforts.

**Git:** Git was used as a version control tool to organize project files in a common repository. Its features allowed for the management of code changes across different stages of the project, enabling collaboration among team members, tracking of project history, and ensuring code integrity and stability.

**IntelliJ:** IntelliJ (or any preferred Integrated Development Environment) served as an example of an IDE used by team members. While specific IDE choices varied among team members based on individual preferences, the use of IntelliJ highlights the importance of providing developers with tools they are comfortable and productive with, enhancing coding efficiency and quality.

## 5. REQUIREMENTS

### a. FUNCTIONAL REQUIREMENTS (FRs)

#### **FR01 Boat control**

As a player, I want to be able to control the movement of my dragon boat using keyboard inputs, so that I can navigate through the race course effectively

#### **FR02 Boat selection**

As a player, I want to be able to select my preferred dragon boat from a list of available options, so that I can personalize my gaming experience.

#### **FR03 Race progress display**

As a player, I want the game to display my current position and progress relative to other competing teams, so that I can gauge my performance during the race.

#### **FR04 Unique Boat Specifications**

As a player, I want each boat to possess distinct characteristics such as speed, acceleration, robustness, glide and maneuverability, in order to add customization to my gaming experience.

#### **FR05 Level of fatigue display**

As a player, I want to see the level of fatigue my team has all along the race in order to manage my race better and so that I have a more realistic gaming experience.

#### **FR06 Encountering obstacles**

As a player, I expect to encounter during the race static and dynamic obstacles like tree branches, rocks or even animals which will slow down my boat or reduce its robustness, in order to add challenge to my gaming experience.



#### **FR07 Mini game integration**

As a player, I want to play mini games that would help me prepare for the main races and help me gain coins, in order to have additional entertainment and that would help me progress in the game.

#### **FR08 In-game money**

As a player, I want to be able to upgrade or buy new boats with the coins I won in the mini game, in order to add to my experience a management of my ressources and a motive to continue playing.

#### **FR09 Power-Up activation**

As a player, I want the ability to activate collected power-ups during the race.

#### **FR10 Obstacle variety**

As a player, I want to encounter a variety of obstacles during the race.

#### **FR11 Obstacle effects**

As a player, I anticipate different effects from encountering obstacles.

#### **FR12 Boat upgrade system**

As a player, I want to upgrade my boat's specifications using in-game currency.

### **FR13 Lane maintenance**

As a player, I want expect my boat to remain in its lane throughout the race.

### **FR14 Progressive difficult levels**

As a player, I expect subsequent race legs to increase in difficulty level.

### **FR15 Real-time fatigue display**

As a player, I want to monitor my team's fatigue level in real-time.

### **FR16 Tutorial mode**

As a player, I want to a tutorial mode to learn game mechanics as a new player.

### **FR17 Customizable controls**

As a player, I want to customize keyboard inputs for boat control.

### **FR18 Easy of use**

As a player, I expect simple and easy-to-understand controls for steering, accelerating the dragon boat, enabling me to focus on the race without getting overwhelmed by complex control schemes.

**FR19 Randomized Obstacle Placement**

As a player, I expect the game to randomly position obstacles within each race segment, ensuring that the arrangement of obstacles varies each time I play the game.

**FR20 Obstacle interaction variety**

As a player, I expect each obstacle to present different consequences upon encountering them.

b. NON-FUNCTIONAL REQUIREMENTS (NFRs)

**NFR01 Intuitive navigation**

As a player, I want the game interface to be intuitive and easy to navigate.

**NFR02 User interface responsiveness**

As a player, I want the software to provide responsive user interface interactions and smooth transitions, to enhance user satisfaction and usability.

**NFR03 Minimal load times**

As a player, I want minimal load times between race segments.

**NFR04 Save game state**

As a player, I want to be able to return to the game later and have the state of my game (coins, acquired boats) saved.

#### **NFR05 Engaging sound effects**

As a player, I want engaging sound effects to enhance immersion.

#### **NFR06 Scability**

As a player, I want the game architecture to allow for scalability to accommodate future updates.

#### **NFR07 Secure data storage**

As a player, I want player data to be securely stored to prevent loss or unauthorized access.

#### **NFR08 Accessibility features**

As a player, I want the game to include accessibility features for players with disabilities.

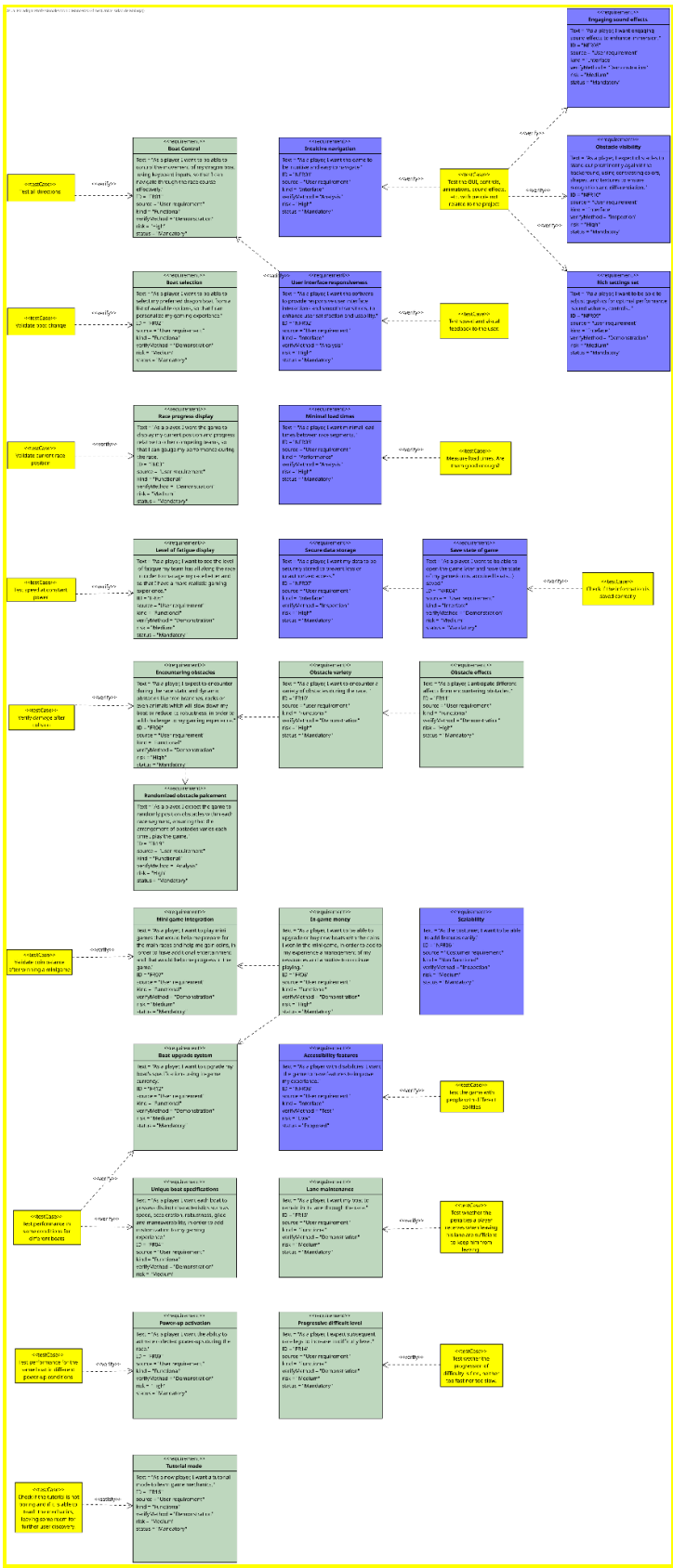
#### **NFR09 Graphics performance optimization**

As a player, I want to adjust graphics settings for optimal performance on various devices.

#### **NFR10 Obstacle visibility**

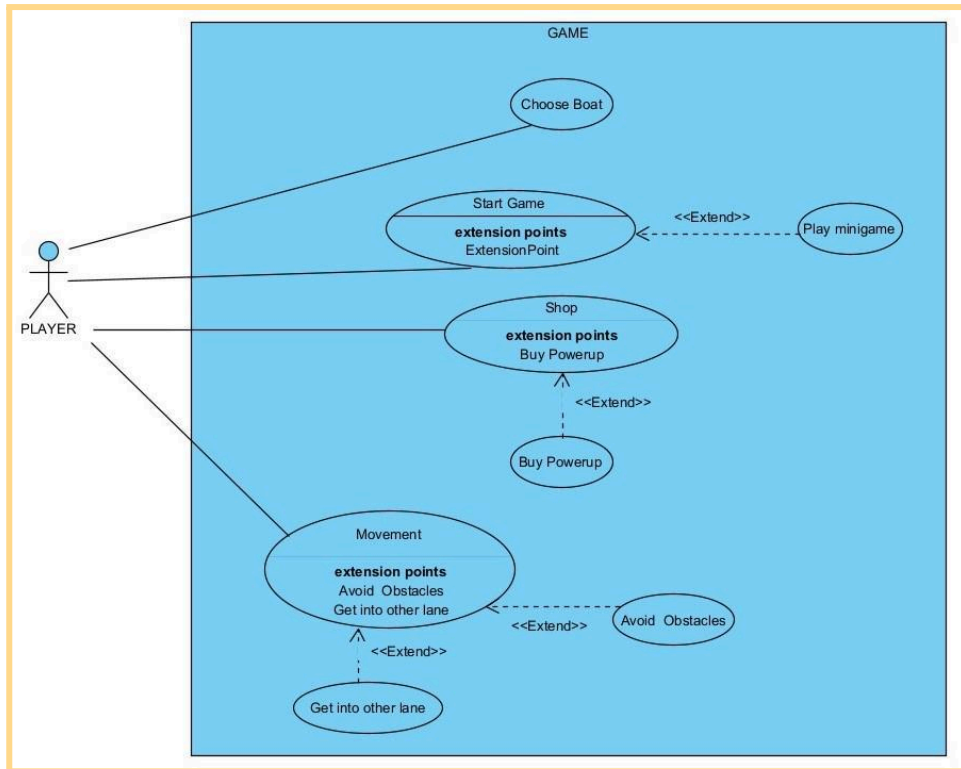
As a player, I expect obstacles to stand out prominently against the background, using contrasting colors, shapes, and textures to ensure recognition and differentiation.

6. REQUIREMENTS DIAGRAM



For a more detailed view, see the [link](#) to the .vpp on Github.

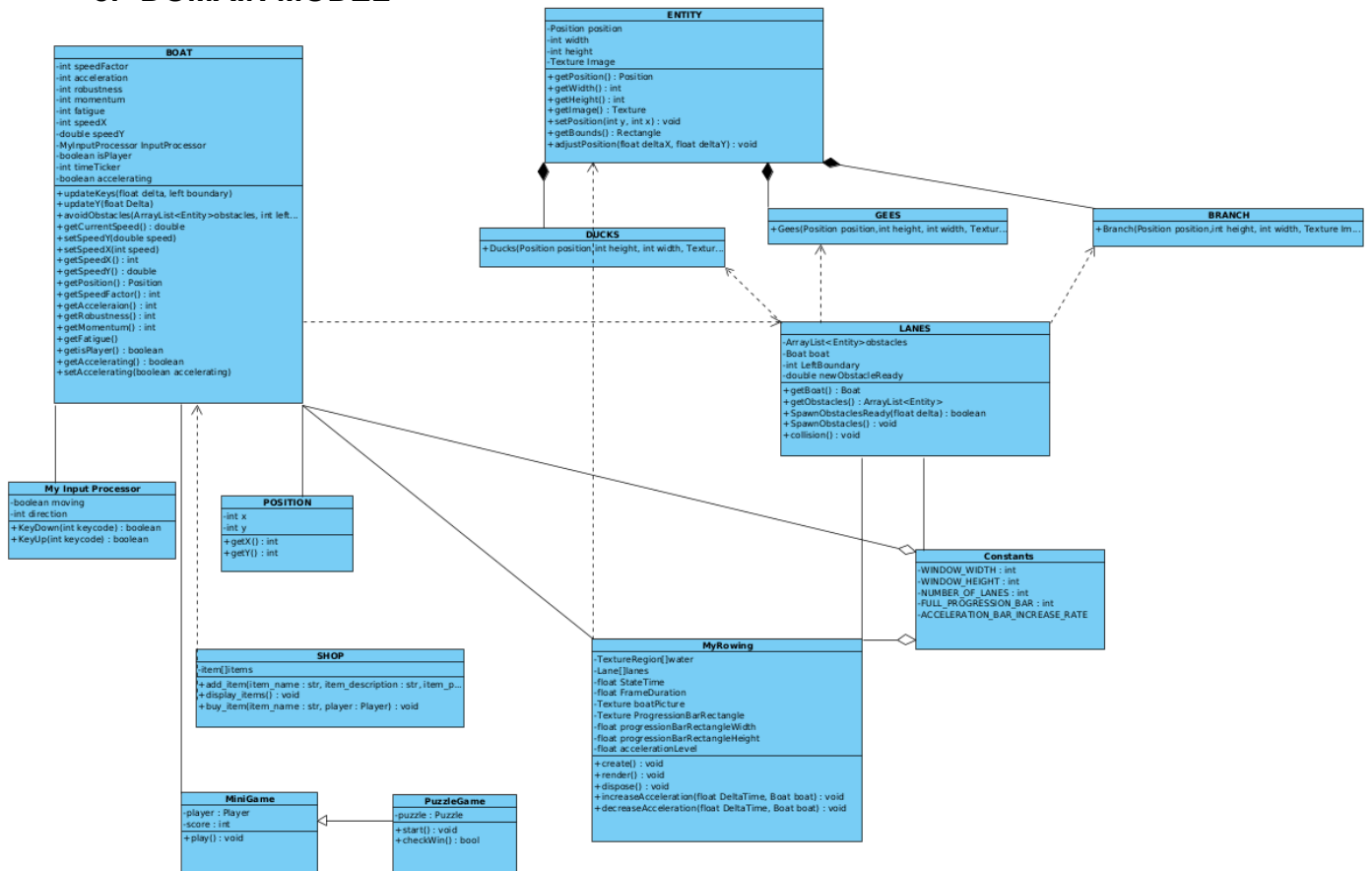
## 7. USE CASES



Unique Identifier	Context of Use	Preconditions and Activation	Success Guarantees	Main Scenario	Alternative Scenarios
UC1. Choose Boat	A player can select a dragon boat to compete in the race	The game is in the boat selection menu and the player is on the boat selection screen.	The player has selected a dragon boat to compete in the race.	The player navigates through the available boats list. The player selects a dragon boat. The player confirms their selection.	If the player cancels the boat selection, they return to the main menu.
UC2. Start Game	The player will be able to start the race.	The player has selected a dragon boat and the game is on the race start screen.	The player has initiated the race with the selected boat.	The player presses the button to start the race. The race begins with the selected boat.	If the player decides not to start the race, they return to the boat selection screen.
UC3. Shop	The player wants to purchase upgrades for their boat.	The player is in the shop screen and the player has coins or points to spend.	The player has purchased the selected upgrades for their boat.	The player selects a specific powerup. The player confirms the purchase of the powerup.	If the player cancels the purchase, they remain in the shop screen without buying the powerup.

<i>UC 4. Buy Powerup</i>	The player will buy a specific powerup for their boat.	The player is in the shop. The player has coins or points to spend.	The player has purchased the selected powerup for their boat that they can use in the race.	The player selects a specific powerup. The player confirms the purchase of the powerup.	If the player cancels the purchase, they remain in the shop screen without buying the powerup.
<i>UC 5. Play Minigame</i>	The player has finished a leg and will play a minigame.	The player has finished a leg.	The player completes the minigame and gains points or advantages for the main race.	The player plays the minigame and wins points/coins.	If the player fails to complete the minigame they continue to the next leg.
<i>UC 6. Movement</i>	The player needs to control the movement of the boat during the race.	The game is in progress. The player is playing a leg..	The player successfully navigates the boat.	The player controls the direction boat using controls to stay within its own lane and avoid obstacles.	If the player steers the boat out of the designated lane or crashes with an obstacle, penalties are applied.
<i>UC 7. Avoid Obstacles</i>	The player is competing in the race and needs to dodge to avoid obstacles.	The game is in progress. The player is controlling the boat.	The player has successfully avoided an obstacle.	The player moves the boat to avoid obstacles and navigates successfully without colliding.	If the player collides with an obstacle, the boat's robustness decreases, and the player could fail the race.
<i>UC 8. Get Into Other Lane</i>	The player decides to move the boat into another lane during the race.	The game is in progress. The player is actively racing in the river. The player intends to switch lanes.	The dragon boat remains in its designated lane without moving into another lane to avoid an obstacle	The player changes lanes trying to avoid an obstacle. It's robustness decreases	The player avoids the obstacle successfully without having to change lanes

## 8. DOMAIN MODEL

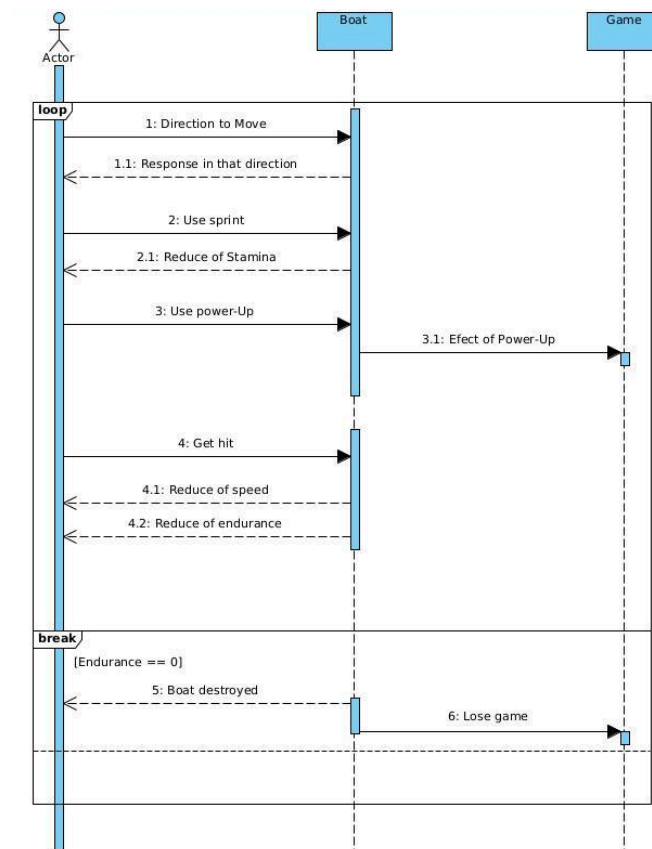


( To see in a better way, the image and the .vpp file are uploaded to the team's Github)



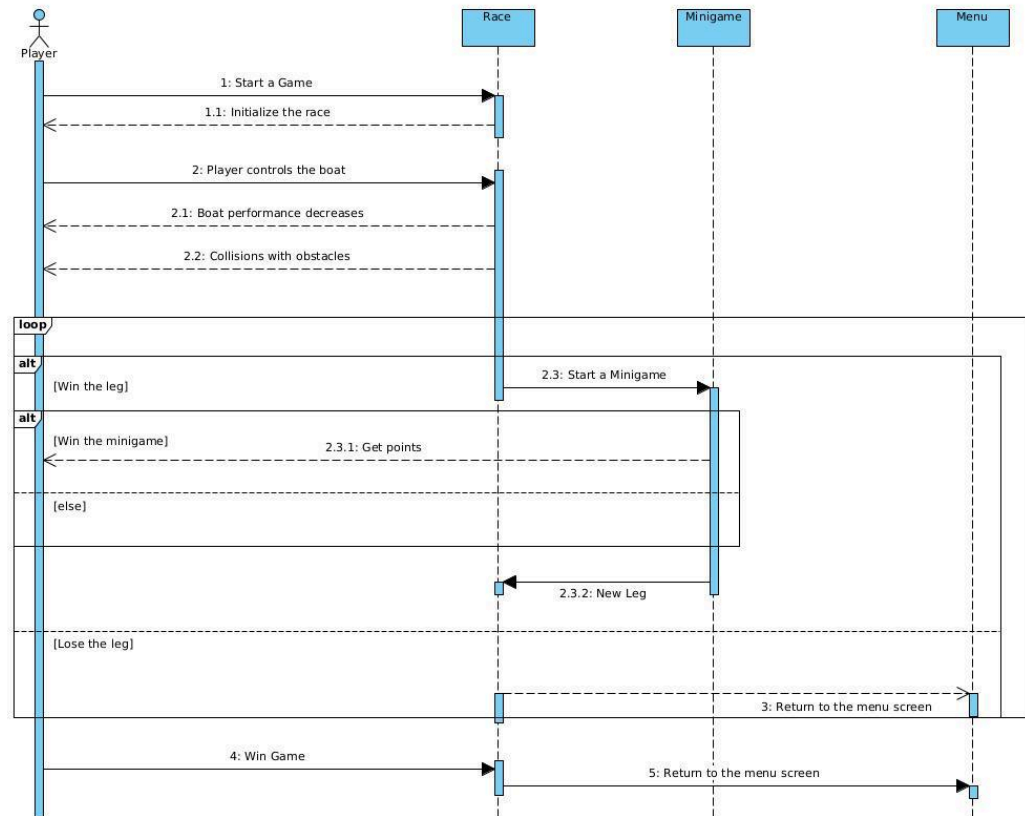
## 9. SEQUENCE DIAGRAMS

### Boat Control:



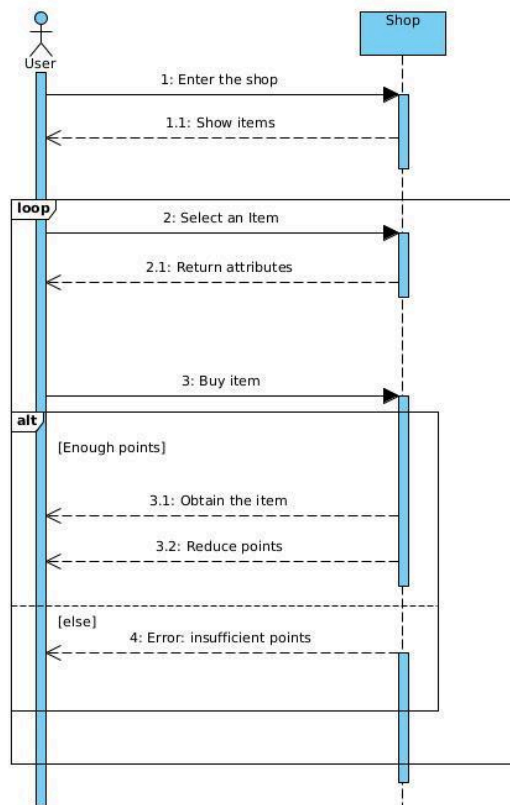
A loop is created where the player chooses the direction in which they want to move the boat, and the boat responds by moving in that direction. When the player uses a sprint, the boat's stamina decreases. If the player uses a power-up, the game responds with the effect of that power-up. If the player gets hit, there is a reduction in the boat's speed and endurance. Finally, if the endurance reaches 0, the loop exits, and the boat is destroyed, resulting in a game over.

## Races:



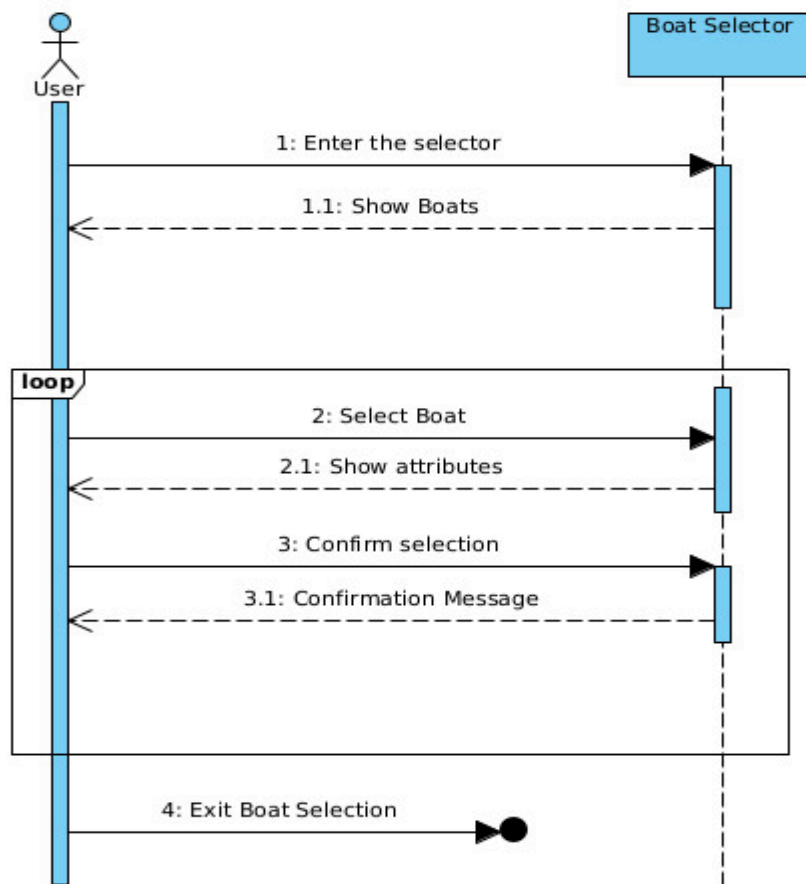
The player begins the game, and the race starts. While racing, the player controls the boat. Boat performance decreases as the game progresses, and collisions with obstacles affect the player. If the player wins the leg, a mini-game starts, where they can earn points. Failing to score points starts a new leg. If the player loses that leg, they return to the home screen.

## Shop:



The player begins the game, and the race starts. While racing, the player controls the boat. Boat performance decreases as the game progresses, and collisions with obstacles affect the player. If the player wins the leg, a mini-game starts, where they can earn points. Failing to score points starts a new leg. If the player loses that leg, they return to the home screen.

## Boat Selection



The player has the option to enter the selector of boats. Here, the selector would show the possible boats and the user will be able to select one of the boats, see its attributes and decide if to select that boat as the one he/she will use during the next games.