Group 2: Bump Boat Game Group

GitHub link
https://github.com/ernmri/UMAsoftwareEngineeringGroupProject

Members

Gonzalo García Rojas; gonzalogrojas@uma.es

Youssef Merroun; youssefmerroun333@gmail.com

Daniil Gumeniuk; gumuma@uma.es

Alejandro Puerto Criado; alejandropc2004@uma.es

Eron Imeri; eron.imeri@uma.es | eronimeri@hotmail.com

Gianluca Nanni; gianlu.nanni@uma.es | gianlu.nanni@gmail.com

Pablo García Moreno pablogarciamoreno11@gmail.com

Table of Contents

Introduction

Dragon Boat Race is a single-player game that leverages advanced game engine technology to deliver a dynamic and immersive experience. The game features multiple choices of boats with different specs, fatigue simulation and obstacles to avoid.

The goal of the game is to win boat races throughout your ability, that will increase while playing, as well as the characteristics of the boat, that you will be able to upgrade temporarily between one race and another, thanks to the featured minigame.

Roles

Developers:

- Gonzalo García Rojas

- Eron Imeri
- Daniil Gumeniuk

Project managers:

- Youssef Merroun
- Alejandro Puerto Criado

Graphical and sound designers:

- Alejandro Puerto Criado
- Gonzalo García Rojas
- Pablo Garcia Moreno

Testers:

- Gianluca Nanni
- Youssef Merroun
- Pablo García Moreno

Scrum masters:

- Eron Imeri
- Gianluca Nanni
- Daniil Gumeniuk

Possible risks

Fair description of risks, but I would suggest using a table form to describe them. It is cumbersome to go through them with the current format. Also, I suggest adding risks related to the technologies to be employed.

Underestimation of the c

- Estimation risk
- Moderate probability
- Serious effects
- To mitigate this risk, our teams must plan each of the steps of the project before starting them and have good communication between all of the members in order to know if we are advancing according to our deadlines.

Quality risk at the end of the project:

- Requirements risk
- High probability
- Catastrophic effects
- To mitigate this risk, we should check the quality of everything we implement in the software, just after implementing it. In order to do it, some of us will have the role of ‘tester’, so that is their duty to test the code and the other parts of the project.

Insufficient qualifications to complete the project:

- Personnel risk

- There is a risk that our team members are not qualified or partially qualified to complete the tasks - especially in the more advanced sections such as: programming, sound and graphic designing. Albeit, this is not too serious since this is a project that has been done before and we have a lot of resources to research on.
- Moderate probability
- Tolerable effect
- To mitigate this risk we will apply research online and we will apply a well-thought of understanding of the tools we will need so that the research takes less time; we will also work and be dependent on each other as a team so that we can combine the skills each of us has.

Dependency between team members:

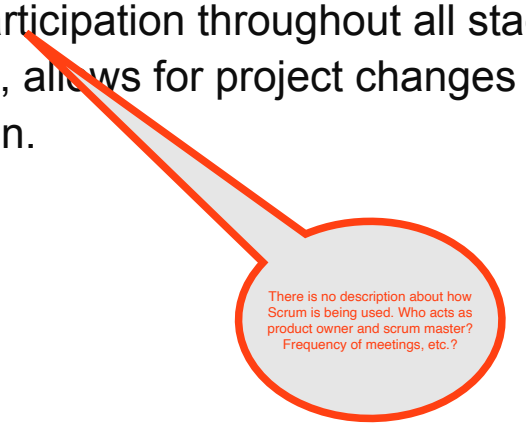
- Personnel risk
- Low probability
- Serious effects
- To mitigate this risk, we should have a contingency plan to divide the work of a team member if they are not available for major reasons, for example, disease.

Lack of time:

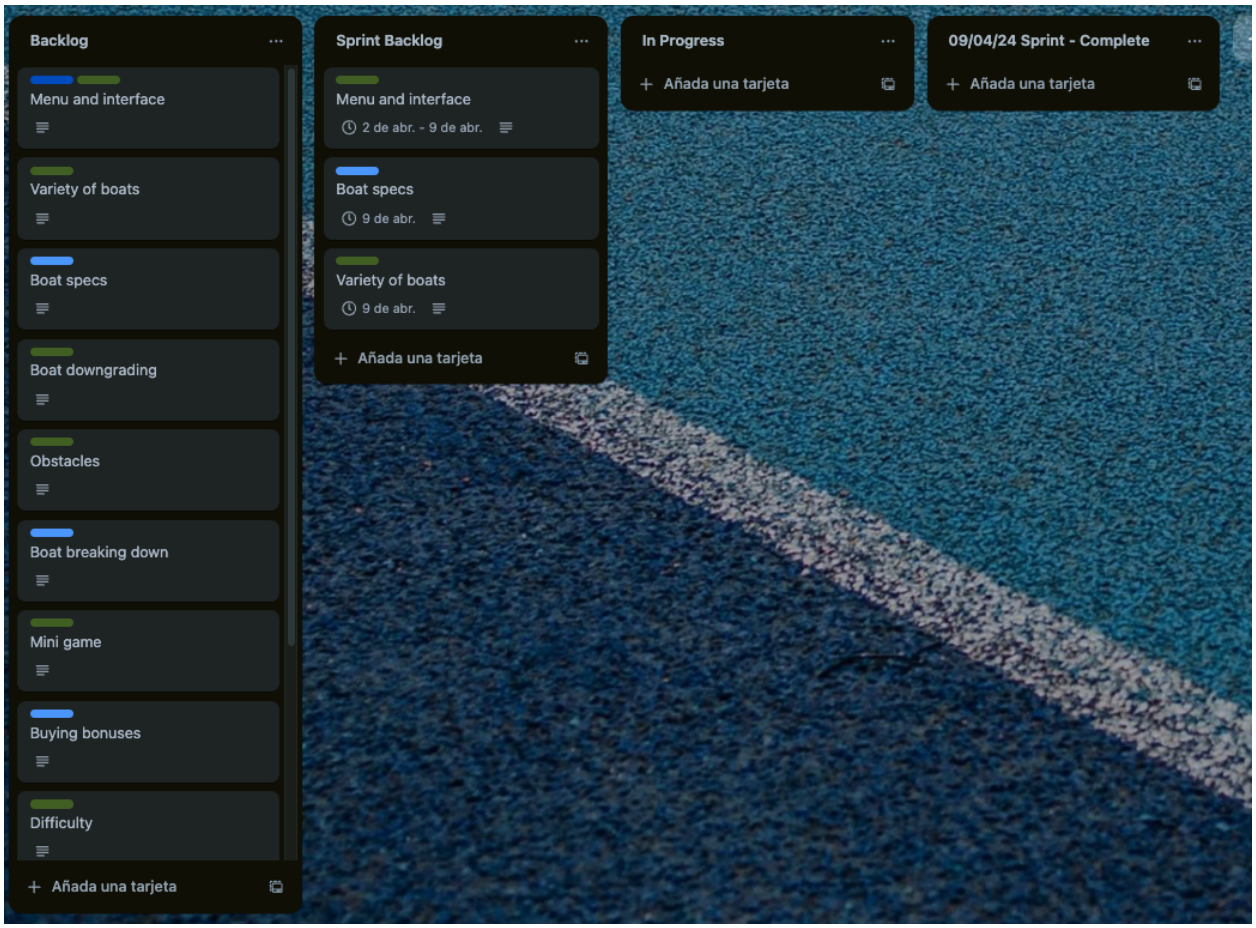
- Estimation risk
- High Probability
- Catastrophic effect
- To mitigate this risk, we should have a good planification and communication between all the members.

Planning

We have selected Scrum as our software process model due to its facilitation of effective communication among team members, encourages everyone's participation throughout all stages, aligns us towards common goals, allows for project changes and its efficacy in time optimization.



Trello boards



Burndown Chart for first sprint

Software tools used during project's realization

- Collaborative work:
  - Github
  - Trello
  - Visual Studio Code
  - Visual Paradigm
- Communication:
  - Discord
  - Whatsapp
- Document Elaboration:
  - Google Documents
  - PDF online converter

Requirements

Requirement ids?
Only 12 requirements? Should be a minimum of 20-30

IREMENTS	TYPE
t be a start menu and a graphic interface to indicate different aspects of the game	FR
Players can choose between boats with different specs to compete in the leg	FR
These boats are differentiated by four specs: speed, acceleration, maneuverability and robustness.	NFR
Speed, acceleration and maneuverability decrease progressively during a leg.	FR
There must be obstacles during the leg that can reduce the robustness of the boat.	FR

If the robustness of the boat goes to zero, it will break down, thus resulting in the end of the game.	NFR
There must be a minigame between each of the legs	FR
Throughout the minigame, you can obtain coins to buy temporal bonuses for your boat.	NFR
Difficulty must increase in each level.	FR
This increment must affect the ability of the boats controlled by the machine.	NFR
If the boat does not respect the borders, it loses the leg	FR
The game must implemen some kind of music and so effects	FR

Phrasing should be more precise

Where does this come from?
Also, no specification about whether reqs are mandatory or optional.

Requirements diagram: