Requirements

To elicit the requirements, we had an initial customer meeting to ask basic requirements of the game and base functionality so we could understand the system and have context to what the development process would require. Once we had gone away and discussed the implementation of such requirements, and what the system requirements would be, we had further customer meetings to clarify on some functions and to ask about other requirements that may be difficult, or suggest functionalities that would be rather easy to implement.

The requirements will first be presented by a single statement of need, followed by tables of user requirements and finally requirements, split into functional and non-functional, addressing any assumptions and risks associated with them.

Single Statement Of Need

The initial brief of the project stated that we needed a single-player dragon boat racing game that involves the user driving one of the boats in a dragon boat competition.

User Requirements

ID	Description	Priority	Assumptions	Risks
Boat_Select	The user must be able to pick a boat from a selection at the start of the competition	Must	The user won't want to change boats mid race	None
Boat_ Control	The user must be able to control the boat during the race	Must	The user won't need absolute full control	The user can drive the boat offscreen

Functional Requirements

ID	Description	Priority	Assumptions	Risks
Unique_ Stats	Each boat must have unique stats, including speed, durability, acceleration, and maneuverability.	Must	None	1 or 2 boats may turn out objectively better than the others
River_ Obstacles	The river that the boat drives in must contain obstacles for the user to avoid.	Must	Only the player needs obstacles	Making obstacles that some boats cannot get around
Time_ Penalty	The user must get a time penalty for hitting obstacles or going into other lanes	Must	The players time must be tracked	None
Obstacle_ Collision	Hitting an obstacle will reduce the boat's robustness	Must	The boat will be disqualified when it's health hits 0	None
Rower_ Fatigue	Over the course of the race the boats will lose stats to simulate fatigue	Must	The loss will be multiplicative	The end legs may feel too slow
2D	The game must be a 2D graphical game	Must	Top down is acceptable	None
Java	The game must be written in Java	Must	None	None
End_ Podium	The end screen must have a podium showcasing the top 3 boats	Must	None	None
Music	The game can contain music	Can	None	May add too much unnecessary code

Non-Functional Requirements

ID	Description	Priority	Assumptions	Risks
Time_Frame	The game must last around 5 minutes total, with each leg being around 1 minute	Shall	None	None
Difficulty_ Increase	Each leg will increase in difficulty	Must	There will be a difficulty cap	None