

## Requirements

### User Requirements

Requirement Number	Description	Priority
U1	User should be able to select a boat and be presented with the stats for each boat	Shall
U2	User should be able to race against other CPU controlled boats	Shall
U3	User should be able to have a practice round followed by 3 proper rounds	Shall
U4	The participants (User and CPU) with the highest scores from the first 2 rounds (not including practice) will go into the final	Shall
U5	The user should be provided with instructions on how to play the game	Shall
U6	User should have to dodge obstacles	Shall
U7	The user will be penalised if they leave their lane	Shall
U8	The User will be able to interact with the game using their keyboard	Shall
U9	The user will be able to interact with the game using their mouse	May
U10	Each leg will be harder than the last	Shall
U11	There should be a variety of boats available to the user, each with different stats	Shall
U12	Despite their different attributes, the boats should be the same length in order to ensure balance	Shall
U13	The health of the user's boat will decrease every time they hit an obstacle	Shall
U14	At the start of each race the boat's healths are restored	Shall
U15	The difficulty will increase with each race	Shall
U16	The boat's will slow each race as the paddlers get tired	Shall
U17	The art style of the game should be consistent	Shall
U18	The user experience should be smooth and enjoyable	Shall

### Functional Requirements

Requirement Number	Description	User Requirement
F1	Obstacles will be created in the path of the user	U6
F2	There should be "AI" of some level to control CPU boats	U2
F3	The player's time should be penalised whenever they leave their lane	U7

F4	The game will have a main menu screen, which allows the selection of a boat type	U1
F5	The game should support keyboard functionality	U8
F6	The game could support mouse functionality	U9
F7	The game should support 4 “legs”	U3, U4
F8	The game will have an instruction screen which appears after the main menu	U5
F9	Hitting too many obstacles will cause the player’s boat to be destroyed, making the player lose	U13
F10	Each leg will result in decreased maneuverability, acceleration and speed	U16
F11	More obstacles will be created in later legs	U10
F12	Players should be presented with graphical sprites for the obstacles and boats	U17
F13	The background and menu screen should be presented to players	U17

### Non-Functional Requirements

Requirement Number	Description	User requirement
NFR1	Not be able to access memories outside of the application (memory leak)	
NFR2	Game should never crash under any circumstances	U18
NFR3	Transition screens should be a fade-in/fade-out action.	U18
NFR4	Should not hog too much resources.	
NFR5	Should be accessible over low-cost hardware without any lags or UX detriment.	U18
NFR6	Errors should be as verbose as possible for debugging.	
NFR7	Developers should be familiar with the codebase and which part of the code is responsible for the mechanics / asset management / game style.	
NFR8	Documentation (comments) for all code should be well-written for other devs on the team to reference upon.	
NFR9	Ideally, the application should be developed in a modular style such that code can be reused, and other devs will only need to know how to use the modules through interfaces.	
NFR10	Full-screen-able	U18
NFR11	Efficient, descriptive comments, logical reasoning behind code. (No “this is a for loop” comments)	
NFR12	Fat jar.	

NFR13	Development Environments and build environments should be as consistent as possible across different machines and every devs.	
NFR14	Devs should make use of the version control system as much as possible.	