REQUIREMENTS

Aaron Price, Alec Coats, Charlie Curedale-Rayner, Eleanor Griffin-Smith

The main user requirements for the game were first introduced in the Cohort 2 Product Brief. From this we formed a series of follow up questions to ask our stakeholder to ensure we fully understood the requirements. We asked these questions via a zoom call due to the pandemic. We also used these questions to clarify specific features such as the type of penalty incurred by a boat leaving its lane. This allowed us to create the user requirements and from that we could create the functional and non-functional requirements related to each user requirement.

The user, functional and non-functional requirements are all recorded in tabular format with each requirement having a unique ID and a short description so they can be uniquely identified during planning and group discussions. Each functional and non-functional requirement is related to a use requirement.

<u>SSON</u> – The Single-Player game will enable players to simulate the annual dragon boat race in a digital format.

<u>User requirements table</u>

ID	**Description**	**Priority**
UR_BOAT_CONTROLS	A player's boat can be controlled by the user	Shall
UR_COMPETITION	The user's boat will be simultaneously be competing with computer generated boats and abide by the rules	Shall
UR_EXPERIENCE	The game will be enjoyable and engaging for the user to play	Should
UR_SKILL	The game progresses in difficulty toward the final race	Should
UR_BOAT_SELECTION	User can choose a unique boat to race in	Should

<u>Functional requirements table</u>

ID	**Description**	**User requirements**
FR_RULES	The game must show the user a list of instructions to play the game	UR_EXPERIENCE
FR_USER_BOAT_CONTROL	User input will cause the players boat to move around the map	UR_BOAT_CONTROLS
FR_BOAT_STAT_SPEED	A user's boat cannot exceed its speed statistic	UR _BOAT_SELECTION
FR_BOAT_STAT_ACCELERATION	If a user's boat is not and full speed and moving it will accelerate at the pace of the acceleration statistic	UR _BOAT_SELECTION
FR_BOAT_STAT_MANOEUVERABILITY	The manoeuvrability statistic controls the time	UR _BOAT_SELECTION

	it takes for the user's boat to respond to user input	
FR_ROBUSTNESS	A user's boat will sink if the robustness of the boat is reduced to zero	UR _BOAT_SELECTION
FR_UNIQUE_BOAT	Every boat in the race will have unique statistics	UR_BOAT_SELECTION
FR_STAMINA	Speed, acceleration, and manoeuvrability will decrease as each leg progresses	UR_BOAT_CONTROLS
FR_STAMINA_RECOVERY	The user's boat will recover stamina at the start of each leg and if there is no control input	UR_BOAT_CONTROLS
FR_QUALIFICATION	The user will only progress onto the next round if they complete the race in the top x times	UR_COMPETITION
FR_RULES	The system will give a penalty to a boat who strays out of it assigned lane	UR_COMPETITION
FR_MAPS	The system will show a map with increasing difficulty at each leg	UR_SKILL
FR_ OBSTACLES	The leg maps contain obstacles which, when hit with a boat, will reduce the boat robustness	UR_EXPERIENCE
FR_REALISTIC_COMPETITORS	The system will include computer generated competitor boats which will actively try and win	UR_COMPETITION
FR_WIN	The user will win or lose the game and start again	UR_EXPERIENCE
FR_GRAPHICS	The game should generate a consistent, family friendly art style	UR_EXPERIENCE

Non-functional requirements

ID	**Description**	**User requirements**	**Fit criteria**
NFR_AVAILABILITY	The game should not crash or lag when running	UR_EXPERIENCE	Game should run at 60 FPS
NFR_INPUTS	The game should accept user input as fast as possible	UR_BOAT_CONTROLS	All inputs will be recorded by the system in <2 seconds

NFR_OUTPUT_SCREEN	The game must output outcomes of events as they happen	UR_EXPERIENCE	Output given to user in <2 seconds
NFR_TIME_ELAPSED	Race times must be recorded to the same precision	UR_COMPETITION	Time recorded in seconds to 2 decimal places
NFR_MAINTAINABILITY	Software engineers must have the ability to release new versions of the game if errors are reported	UR_EXPERIENCE	Software engineers can release new versions of the game
NFR_RESILIENCE	An error in the game will not impact the device it is being played on	UR_EXPERIENCE	Computer will not crash if the game crashes
NFR_SCALABILITY	The game will be able to be run on any standard university desktop	UR_EXPERIENCE	Fits to [uni spec]
NFR_OPERABILITY	The game can be played by a user when they have read the rules	UR_SKILL	95% of users should understand the rule after reading instructions
NFR_USABILITY	The game instruction will be simple and have no jargon. As well as intuitive controls	UR_EXPERIENCE	95% of users should understand the rule after reading instructions
NFR_INTEGRABILITY	The game will access the predetermined statistic for each boat from within the program	UR_BOAT_SELECTION	The user will be able to select different boats with different statistics

<u>Constraint requirements</u>

Project constraint – No budget. Team of 5 student software engineers.

Process constraints – Must use Java.

Design constraints – Must run on university specification computers