Requirements

Team 06:

Adam Hagan

Tudor Ciobanu

Armintas Zadeika

Sanjel Sadikaj

Liu Zhang

Na Tang

Requirements Introduction:

The following software we are creating is a single-player boat race game where a user is able to race against other boats (which will be CPU controlled). The following set of requirements for this piece of software were elicited through various methods. One method, which was used to extract requirements for the software, was analysing the software brief and searching for clear requirements that create the base of the software. An example of this method in use was the brief document stating “Every boat must have a unique spec in terms of speed, acceleration, maneuverability, and robustness.” As the brief stated specific requirements, the requirements that were stated in the document were recorded and further broken down into sub-requirements.

Furthermore,we contacted and set meetings with the stakeholders of the software to receive any additional information. Meetings were set up via Google Meet and communications took place over Email. The meetings consisted of our team asking detailed questions (which have been planned prior) regarding requirements which were not specified in the product brief. Examples of questions which we asked include; “How should the boat be controlled by the player?”, “is it acceptable for the CPU to try not to avoid obstacles?” and “should boats be repaired between legs”. After receiving the answers we added them to our list of requirements.

We were left with a document which contained numerous requirements for the software we need to implement. The next process we needed to perform was to effectively represent our requirements. This was accomplished by viewing the requirements and categorising each requirement either as an overall user requirement or a system requirement. Once that task was completed, we further broke down the system requirements into two different sections; functional requirements or non-functional requirements. This left us with a viable solution to represent each requirement section as a table (total of 3 tables). The contents of the table’s would include; the ID of the requirements (so we could link and reference the requirements between system and user and, uniquely distinguish a requirement), a brief description of what the requirement is and finally, a fit criteria, user requirement or priority column. The fit criteria column was specific to the non-functional requirements so we could measure the success of the requirement. The user requirement column, in the Functional Requirements Table, was important as it allowed us to visualise the link between each individual user and system requirement. Lastly, the priority column, in relation to the user requirement table, was crucial in understanding the importance of each requirement.

Requirement Specification:

User Requirement Table:

ID	Description	Priority
UR_USER_MOVEMENT	The user should be able to paddle the boat by pressing the movement keys.	**MUST**
UR_RACE_AGAINST_CPU	The user should be able to race against other boats which are controlled by the CPU.	**MUST**
UR_BOAT_DAMAGAE	The user's boat and other boats should lose health if they crash into obstacles or boats - if all health is lost, the boat is disqualified.	SHOULD
UR_BOAT_COLLISION	When the boat collides with another boat or an obstacle, simple collision physics will be simulated.	SHOULD
UR_OBSTACLE_COLLISON	Obstacles should be positioned such that they can never collide with other obstacles.	SHOULD
UR_DEGRADATION_OF_STATS	Every boat's stats will degrade as the race continues, degrading faster if the user paddles more frequently.	SHOULD
UR_PENALTIES	The player should receive penalties for leaving their lane.	SHOULD
UR_RACE_STRUCTURE	The race should be structured into 3 legs, and a final with the best boats from the previous legs.	**MUST**
UR_RESULTS	After each leg, the user should receive their position and time. If they are disqualified, this should also be shown.	SHOULD
UR_PROGRESSIVE_DIFFICULTY	The race difficulty for the user should increase after each subsequent leg.	SHOULD
UR_PICK_BOAT	The player will be able to pick a boat to race in, and each boat must be distinct with varying *acceleration*, *top speed*, *maneuverability* and *robustness*.	**MUST**
UR_QUIT_DIALOGUE	The 'Esc' key should open a menu that allows the user to end the game early and return to the start screen.	*COULD*
UR_INSTRUCTIONS	A brief menu prior to starting the game will mention some brief instructions, including how to control the boat.	*COULD*
UR_ACCESSIBLE	The game should be easily accessible (and enjoyable) for other people in our ENG1 cohort.	SHOULD

Functional Requirements Table:

ID	Description	User Requirements
FR_BOAT_MOVEMENT	When the user presses the designated movement keys, the boat should be accelerated and turned according to the stats of the boat.	UR_USER_MOVEMENT
FR_COLLISION_PHYSICS	The boat should not be able to pass through other boats or obstacles. Simple physics should be calculated instead. **Additional Notes:** Collision physics are a very difficult problem. If a physics library for this can't be found, obstacles may be destroyed on contact instead.	UR_BOAT_COLLISION
FR_ROBUSTNESS_LOSS	If a boat hits an obstacle or boat in the river, it should lose robustness.	UR_BOAT_DAMAGE
FR_ZERO_ROBUSTNESS	If a boat loses all its robustness value then that boat is disqualified and will stop moving. If it is the user's boat, they get a game over.	UR_BOAT_DAMAGE
FR_OBSTACLE_COLLISION	Obstacles should be positioned such that they never collide with each other.	UR_OBSTACLE_COLLISION
FR_PADDLER_STAMINA_ DECREASING	Frequent paddling by the user (i.e. pressing the movement keys every second) will result in the stats of the boat degrading faster.	UR_DEGRADATION_OF_STATS
FR_REPLENISHING_LEGS	At the end each leg the boat's robustness and stamina will be replenished to its initial starting value.	UR_DEGRADATION_OF_STATS UR_BOAT_DAMAGE
FR_LEAVING_LANE	If a boat has left their designated lane then they should be fined with a time penalty, equal to the time spent outside their lane.	UR_PENALTIES
FR_RACE_LEGS	After each leg, the player should be shown their results, and then moved into the next leg.	UR_RACE_STRUCTURE
FR_FINAL_LEG	After the third leg, only the boats with the 4 fastest times will move to the final leg.	UR_RACE_STRUCTURE
FR_NUMBER_OF_BOATS	There should always be a fixed number of 6 boats in the race legs, and 4 boats in the final.	UR_RACE_STRUCTURE

FR_BOAT_TIMES	For each leg, each individual boat should be timed for the duration of finishing the race.	UR_RESULTS
FR_PICK_BOAT	Before the race begins, the player can choose one of at least 6 boats. Once their boat is chosen it can't be changed until the full race is over.	UR_PICK_BOAT
FR_MOVING_OBSTACLES	Some obstacles in the river are able to move, later legs including more moving obstacles. Movement may occur from left to right.	UR_ PROGRESSIVE_ DIFFICULTY
FR_INCREASED_ OBSTACLES	Later legs should include more obstacles.	UR_ PROGRESSIVE_ DIFFICULTY
FR_QUIT_DIALOGUE	If you press the 'esc' key during a race, a small pop-up dialog will ask the player to confirm whether to quit the game, which will be confirmed or cancelled pressing 'y' and 'n', respectively.	UR_QUIT_ DIALOGUE
FR_INSTRUCTIONS	The game should include a screen or dialogue with instructions explaining the structure of the race, and the controls, no longer than 250 words.	UR_ INSTRUCTIONS

Non-Functional Requirements:

ID	Description	User Requirement	Fit Criteria
NFR_CPU_ MOVEMENT	The CPU controlled boats should be able to navigate and race through the course.	UR_RACE_ AGAINST_CPU	90% of the CPU controlled boats should successfully complete the first leg without receiving a disqualification. **Additional Notes:** This may prove difficult - if there are time constraints, AI boats may have increased health or ignore obstacles.
NFR_INSTR UCTIONS	The instructions should be easily accessible.	UR_ACCESSIBLE	It should only take the user either one mouse/key press to reach the instructions page from the starting menu.
NFR_DIFFE RENT_BOAT	All boats have distinct stats and appearances to the other boats.	UR_PICK_BOAT	All boats should have at least two stats different from any other boat, and a unique colour.