Requirements

Assessment 2

Team 10: Hard G For GIFs

Dragos Stoican

Rhys Milling

Samuel Plane

Quentin Rothman

Bowen Lyu

Change Report

Eliciting requirements

To define requirements we decided upon using IEEE’s 12207.1 [2] standards as a base and also followed their terminology. As a supplement, we also used guru99’s article [1] to help us clarify the system requirements. to assist us in defining the system requirements in particular.

A brief statement was asked for that describes the idea/issue the customer was facing. “I would like to have a game developed and this would be a single player game and should capture all the excitement of these events: the Dragon Boat Race that happens in York every year". This, combined with the listed requirements in the product brief, gave us our initial requirements to start the process of eliciting requirements.

Upon reading requirements listed in the brief, it was decided that some of the features were vague and could be easily misinterpreted in the development. To make sure our requirements were crystal clear and solid, we created questions to ask the customer for meetings with them, discussing directly with them to be able to understand their vision. These meetings also allowed us to see that the customer put emphasis on certain features which weren’t specifically underlined in the assessment document, such as general appearance and UI of the game.

Once all our questions were answered, we would meet together and sort out our requirements. We decided to separate them into 3 tables:

1. User requirements are high-level abstractions, designed to be presentable to our customers and non-technical people in order to help us develop our functional requirements.
2. Functional requirements, a subset of system requirements we split up the user requirements into a more concrete implementation. This is the “business logic” of the game. Defining the behaviour (inputs and outputs) and how everything interacts together.
3. Non-functional requirements, also a part of system requirements, focusing more on the quality of the game. Not completing these requirements would result in a game that would be unsuitable for the customer.

All the tables are then layed out in similar fashion. Looking at the UR table, we give an ID related to what the requirement is, a description of it, and a priority so that we know how important it is compared to other requirements. The priorities are sorted as follows: “L” for Low, “M” for Medium, and “H” for High. For the FR table, the layout is similar, except we replace priority with the UR reference, which would then translate the priority of the FR. The NFR Table is similar to the FR table, with one added column. We have the “Fit Criteria” column to denote when an NFR is fulfilled .

User Requirements Table

ID	Description	Priority
UR_PLAYABILITY	The game must be playable with a keyboard and mouse	M
UR_BOAT_SPECS	Every boat must have a unique spec in terms of speed, acceleration, maneuverability and robustness	H
UR_TIRED_OVER_TIME	Over time, padders in the team get tired, so speed, acceleration and maneuverability decrease progressively during every leg	H
UR_LANE_PENALTY	Every boat must remain in its lane for the duration of the race. Leaving the lane should result in a penalty at the discretion of the chief race officer	M
UR_OBSTACLES	Teams should find obstacles in the river during the race, like clueless ducks and geese, or tree branches floating down the river	M
UR_COLLISIONS	Colliding against obstacles will progressively reduce the robustness of the boat, until it breaks down, result in the end of the game	M
UR_LEG_DIFFICULTY	Every subsequent leg will increase in difficulty level	M
UR_LEGS	The competition must consist of 3 legs and a final with the fastest from the 3 previous leg in the final	H
UR_BOATS_NO	The number of teams should be consistent with number of legs so that the races have an appropriate number of boats and the race should not be cluttered	L
UR_INFO_DISPLAY	During races, the user should be able to see information such as position in race, distance to go, speed, acceleration, stamina and durability	L
UR_DFFICULTY	The user can choose the difficulty of the race before starting the game	H
UR_PAUSE_MENU	During any point in the race, the user should be able to pause the game. In the pause menu the player should be able to save&quit or resume the game.	H
UR_SAVE_GAME	The user can save the game through the pause menu, and create a save file	H
UR_LOAD_GAME	The user should be able to load a saved game through the start menu at any point. If there is no saved game, the load game button should be inactive	H
UR_PICKUP_BOOST	There should be item boxes that appear throughout the course, like obstacles. Hitting them will give the player a random buff that is displayed on the UI.	H
UR_SETTINGS	The user should have access to settings like volume, resolution and fullscreen option	L
UR_ENEMY_BOATS	The player should go against AI boats that can react to the race environment	M
UR_PERFORMANCE	The game should perform well on any hardware	M
UR_RESULTS	At the end of each leg, results should be displayed in order of best to worst time. At the end of the competition, display each boat's best time, factoring in all the legs they competed in and the penalties they received to determine who the winner was	M

Functional Requirements Table

ID	Description	User requirement ref
FR_CONTROL	The game must take user’s key-presses as input for controlling the boat in-game	UR_PLAYABILITY
FR_STATS	The game has different properties for boat specification in terms of speed, acceleration, stamina, maneuverability and robustness	UR_BOAT_SPECS
FR_BOAT_ASPECT	The aspect of a boat should differ according to the type of boat	UR_BOAT_SPECS
FR_STAM_USAGE	Stamina decreases during the race as it is used. Having lower stamina affects your stats, such as maneuverability and acceleration.	UR_TIRED_OVER_TIME
FR_STAM_REGEN	Stamina replenishes if movements are conservativve and stamina is fully restored between the legs. There is a timer before stamina starts to regenerate once the boat stops accelerating.	UR_TIRED_OVER_TIME
FR_STAM_DEPLETED	Running out of stamina causes the boat to have to wait until the paddlers have replenished at least half of the maximum stamina	UR_TIRED_OVER_TIME
FR_PENALTIES	Boats must remain in their lane for the duration of the race. Leaving the lane must result in a penalty at a discretion of the chief race official	UR_LANE_PENALTY
FR_HIT_OBSTACLE	Colliding against obstacles will reduce the speed and robustness of the boat, until it breaks down (resulting in the end of the leg).	UR_COLLISIONS
FR_MOVING_OBSTACLES	Some obstacles should start stationary, but then in later legs start becoming more dynamic, and moving left to right. Some obstacles are permanently stationary	UR_OBSTACLES
FR_OBSTACLE_CLUTTER	Game should not display too many obstacles in order to not clutter the screen	UR_OBSTACLES
FR_OBSTACLES	Game should display obstacles in the river during the race, like clueless ducks and geese, or tree branches floating down the river	UR_OBSTACLES
FR_INCREASE_OBSTACLES	The game should increasing the difficulty level with every subsequent leg by changing the number, speed, and type of obstacle (dynamic or static)	UR_LEG_DIFFICULTY
FR_DIFFICULTY_DISPLAY	During gameplay the game should overlay/display the difficulty level clearly and this should update to to show any changes to the difficulty level based on the leg of the race	UR_DIFFICULTY
FR_TUTORIAL	At the start of the race there should be a tutorial and/or overlay of contents that are clearly visible and understandable to a new player	UR_ACCESSIBILITY
FR_LEGS	The competition must consist of 3 legs and a final, with the fastest from the 3 previous one	UR_LEGS
FR_BOATS_NO	Each race should have 3-6 boats	UR_BOATS_NO
FR_POV	The game view should be fixed in a central position for the user’s boat and shows boats around the user’s boat (but not all of them)	UR_BOATS_NO
FR_UI	During race, screen should display user’s position in rance, distance remaining, stamina, speed, acceleration and damage	UR_INFO_DISPLAY
FR_PEN_NOTIF	When a penalty is incurred a notification should be displayed on screen	UR_LANE_PENALTY
FR_DIFFICULTY_SELECTION	The user can select the difficulty from the boat selection screen	UR_DIFFICULTY

FR_DIFFICULTY_EFFECT	The chosen difficulty should affect the player's stamina regeneration. The harder the difficulty, the more the player has to wait for stamina to regenerate	UR_DIFFICULTY
FR_SAVE	Allows the player to save the state of the game at any point	UR_SAVE_GAME
FR_LOAD	Allows the player to resume the state of the game at any point. The loaded state should be identical to the saved state.	UR_LOAD_GAME
FR_NO_SAVE_FILE	If there is no save file, the load button is inactive	UR_LOAD_GAME
FR_RANDOM_BOOST	The item boxes offer the boat a one of five random effects: speed boost, repair boost, maneuverability boost, stamina boost and invincibility boost	UR_PICKUP_BOOST
FR_BOOST_DURATION	The duration of the boosts is 5 seconds	UR_PICKUP_BOOST
FR_MULTIPLE_EFFECTS	The boat can be affected by multiple boosts at the same time	UR_PICKUP_BOOST
FR_AI	The AI in the game can dodge obstacles and stay its lane until the end of the race	UR_ENEMY_BOATS
FR_AI_VARIETY	The AI should have different boat types	UR_ENEMY_BOATS
FR_PAUSE	When paused, the timer stops increasing and all boats stop moving	UR_PAUSE_MENU
FR_CHANGE_VOLUME	The user can change the volume in the settings screen	UR_SETTINGS
FR_FULLSCREEN	The user can change the game to fullscreen mode with a any resolution	UR_SETTINGS

Non-Functional Requirements

ID	Description	Fit Criterion	User Requirements Ref
NFR_PERFORMANCE	The game must run smoothly enough for the game to be enjoyable	90% of the time the user does not experience dropped frames or stuttering	UR_PERFORMANCE
NFR_CLUNKINESS	No clunkiness in the movement of the object/sprites	90% of the time the user feels the controls are responsive and visuals reflect this with low latency	UR_PERFORMANCE
NFR_RESILIENCE	The game should not crash at any point during use or impact performance of the computer	90% the game does not crash	UR_PERFORMANCE
NFR_RELIABILITY	The game should be available to use all of the time the user's computer is on	99% of the time the game should be available to use	UR_PERFORMANCE
NFR_USABILITY	The game's messages and information should be in plain English and easy to comprehend	95% of users find the information in the game is easy to understand	UR_INFO_DISPLAY
NFR_RESPONSE_TIME	The game should respond quickly to any input from the user	90% of the time the game responds within 16 ms	UR_PERFORMANCE
NFR_MEMORABILITY	The game should be memorable so that users know how to use it when they go back to it	95% of users remember how to use the game a second time round	UR_PLAYABILITY
NFR_SATISFACTION	The game's design should be satisfying to look at and use	90% of users are satisfied with the design	UR_PLAYABILITY

Bibliography

[1] https://www.guru99.com/functional-vs-non-functional-requirements.html
[2] https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=669648
[3] I. Sommerville, Software Engineering, Pearson Education, 2008, pp. 101-122.