

Requirements

Elicitation and Specification of Requirements

In order to elicit requirements, we first researched how best to go about doing and presenting these as well as the point of requirements engineering in a project. Requirements engineering is the process of discerning what the stakeholders want from the product, then presenting it in a way which is readable and makes the set of requirements processable [1]. According to Sommerville [2], open interviews are often not an ideal way to conduct research, so we elected to use a mostly closed-format interview in order to gather our initial set of user requirements. Our initial interview was with our main stakeholder, Dr. Javier Cámara, and was used to gather basic information about the system we were to implement.

Our other stakeholder, the Communications Office at the University of York, was also represented by our main stakeholder in this specific instance as they are in contact with them. We had to do very little negotiation between stakeholders to decide on final requirements because of a clean overlap in stakeholder interests. Both stakeholders want a product that will be playable and enjoyable, with the main stakeholder wanting it marketable additionally and the other stakeholder wanting it to be usable in promotional activities. The majority of negotiation was between our team and the stakeholder to reach a compromise between what is possible given the allotted resources and what the desired product is.

Presentation format

After completing the interview, we had to find a way to present them that would be understandable and concise. There are four main ways to present requirements - using natural language sentences, structured natural language, graphically using UML or diagrams, or mathematically [3]. User requirements should be written in a way that describes what the functional and non-functional requirements mean in an understandable way, so we wrote our requirements in a structured natural language format. This format involves writing the requirements using a standardised template so that each requirement is unambiguous and clear, one of the main advantages of this format over natural language sentences. Whilst it is less unambiguous than pure mathematical notation, it is often far easier for a customer to understand and means they are more likely to accept the set of requirements.

Citations

1. Dick, J., Hull, E. and Jackson, K., 2017. *Requirements Engineering*. 4th ed. Cham: Springer International Publishing, pp.93-65.
2. Sommerville, I., 2016. *Software Engineering, Global Edition*. 10th ed. NOIDA: Pearson Education Limited, p.116.
3. Sommerville, I., 2016. *Software Engineering, Global Edition*. 10th ed. NOIDA: Pearson Education Limited, p.121.

User Requirements

Our client has asked us to produce a game about Dragon Boat racing. The player will control a boat over three heats and a final race, using the keyboard and mouse to dodge obstacles which may slow them down or damage their craft. Between these races, the player can upgrade their boat if they have the required currency to do so, which will give them a competitive advantage. The player will win if they have the fastest time, place if they are second or third fastest, or lose otherwise.

- UR_1 CUSTOMISATION: Boats must have unique specs in terms of speed, acceleration, maneuverability, and robustness. Stated in the initial brief
 - Risk: The game becomes unnecessarily complex
 - Alternative: The stats of the rowers and upgradeable stats could be changed
- UR_2 STAMINA: Over time, the stats of the boat will reduce. Stated in the initial brief.
 - Risk: The game could feel very restrictive.
 - Alternative: Either remove this feature or make it less restrictive.
- UR_3 STAY IN LANE: The boat must stay in its lane or a penalty may be issued. Stated in the initial brief.
 - Risk: Lanes are too narrow or filled with too many obstacles
 - Alternative: Boats are penalised for collisions with other boats
- UR_4 OBSTACLES: There are obstacles that the boat must avoid. Stated in the initial brief.
 - Risk: The game will become too complex
 - Alternative: Collisions could reduce the rowers stamina
- UR_5 DAMAGE: Colliding with obstacles will cause damage to the boat until it breaks at 0 durability. Stated in the initial brief.
 - Risk: The game can become too difficult
 - Alternative: Damage instead impacts the speed or maneuverability of the boat.
- UR_6 SCALING DIFFICULTY: The difficulty of the game must increase with each subsequent leg. Stated in the initial brief.
 - Risk: The game could become too difficult toward the end of the race
 - Alternative: The time required to complete each leg could decrease
- UR_7 INPUT DEVICE: The game should be playable using a keyboard & mouse. Assuming a UK QWERTY Keyboard is used
 - Risk: Alienating certain players who prefer a different device
 - Alternative: Include mappings for other input devices as well
- UR_8 MINIMUM REQUIREMENTS: The game should be playable on a low spec computer. Assumed a dual-core laptop with 4GB of RAM less than 6 years old.
 - Risk: We can't leverage more powerful hardware limiting the potential of the game
 - Alternative: The game could be developed for a higher minimum specification
- UR_9 CROSS PLATFORM: The game may be playable on multiple platforms. Assumed users are on Windows, Linux or Mac OS X.
 - Risk: An increase in development time to have it work on all platforms
 - Alternative: We could target just one platform
- UR_10 TUTORIAL: The game may have a tutorial to teach the user how to play. Assuming the user has no knowledge of the game
 - Risk: The tutorial is too long winded and simple
 - Alternative: Show the player the controls and let them figure out the rules.

- UR_11 UPGRADES: The game may have an upgrade system that allows the user to customise their boat as they would like it. Stated in the initial brief.
 - Risk: The game becomes unnecessarily complex
 - Alternative: The game could have power-ups for temporary stat upgrades

Functional Requirements

- UR_1 CUSTOMISATION:
 - FR_1.1 CUSTOM ACCELERATION: Boat acceleration statistic should affect how quickly the boat gains speed
 - FR_1.2 CUSTOM MANEUVERABILITY: Boat maneuverability statistic should affect how quickly the boat can turn or strafe
 - FR_1.3 CUSTOM ROBUSTNESS: Boat robustness statistic should affect how much damage the boat can take
 - FR_1.4 CUSTOM SPEED: Boat speed statistic should affect how fast the boat moves
 All Stated in the initial brief.
 - Risk: Stats are not well balanced making some boats better than others
 - Alternative: The stats only make a small difference so no boats are too hard or too easy to win with
- UR_3 STAY IN LANE:
 - FR_3.1 COLLISION LANE: Boat colliding with another lane leads to a penalty being issued
 Stated in the initial brief
 - Risk: Penalties are too harsh making user play too safely
 - Alternative: Lanes are wider making it harder to collide with another lane
- UR_4 OBSTACLES:
 - FR_4.1 OBSTACLE SPAWNING: Obstacles will spawn at a rate directly proportional to which leg the player is on
 Stated in the initial brief
 - Risk: Obstacles become too hard to avoid on the later legs of the race
 - Alternative: More obstacles spawn at the start of the race when your stats are at their highest making it easier to avoid the obstacles
- UR_5 DAMAGE:
 - FR_5.1 COLLISION OBSTACLES: Boat colliding with objects will lead to damage to the boat
 - FR_5.2 DAMAGE GAME OVER: Taking enough damage to reduce the boat to 0 health will lead to the game ending
 - FR_5.3 DAMAGE STATS: Taking damage will affect the speed, acceleration, and maneuverability of the boat
 All stated in the initial brief.
 - Risk: The effect of damage on the boat will make it too hard to win a race
 - Alternative: The effect of damage to a boat is greatly reduced
- UR_6 SCALING DIFFICULTY:
 - FR_6.1 LEVEL SYSTEM: Each leg of the race should be treated as a different level, which will be loaded when played
 - FR_6.2 OBSTACLE COUNT: The number of obstacles in the lane increases with each leg
 - FR_6.3 STAT DECAY: The stamina and health drain on the boat increases with each leg
 Stated in the initial brief.
 - Risk: The race will become too difficult towards the last legs

- Alternative: Other boats become faster in later legs
- UR_7 INPUT DEVICE:
 - FR_7.1 INPUT SYSTEM: User pressing assigned keys will lead to the boat moving
Stated in the initial brief
 - Risk: The users input device of choice isn't supported by the game
 - Alternative: We add support for more input devices
- UR_10 TUTORIAL:
 - FR_10.1 TUTORIAL LEVEL: The tutorial is a special-case level that will only play once
Assuming the user has no knowledge of the game
 - Risk: The level is too drawn out and the user gets bored
 - Alternative: Show the player the controls and let them figure out the rules
- UR_11 UPGRADES:
 - FR_11.1 UPGRADE SYSTEM: Upgrading a boat stat should directly affect the relevant statistic
 - FR_11.2 CURRENCY: Upgrades must be purchased with a special currency that can be earned during races
All stated in the initial brief
 - Risk: Boats become too powerful from upgrades making the game too easy
 - Alternative: Opponent boats also get upgrades so the game doesn't become too easy

Non-Functional Requirements

- UR_7 INPUT DEVICE:
 - NFR_7.1 LATENCY: The response time must be less than 0.5 seconds and not exceed 2 seconds
 - Risk: We do not reach this goal and the game feels unresponsive
- UR_8 MINIMUM REQUIREMENTS:
 - NFR_8.1 TARGET FRAME RATE: The game must run at 30 FPS on the minimum hardware specifications
 - NFR_8.2 MAINTAINABILITY: The game must be updated within 2 min
 - NFR_8.3 MAX MEMORY USAGE: The game must be able to run on 1GB of RAM
 - NFR_8.4 STORAGE USAGE: The game must use less than 500MB of space
 - NFR_8.5 RELIABILITY: The game must not crash 99% of the time
 - NFR_8.6 USABILITY: 90% of the target audience will understand the game within 1 min of their first try.
 - Risk: We fail to meet these requirements meaning the game doesn't run on the minimum specifications
- UR_9 CROSS PLATFORM:
 - NFR_9.1 PLATFORM: The game shall run on Windows, Linux and Mac OS
 - Risk: The game is not compatible with all systems excluding some players
- UR_10 TUTORIAL:
 - NFR_10.1 ACCESSIBILITY: An average user without any major vision, mental or physical disabilities shall be able to play the game within 1 min after completing the tutorial
 - Risk: The tutorial level is too simple or too complex excluding players