Requirements

Original Assessment 1 Team

Team 15

    Joe Wrieden

    Benji Garment

    Marcin Mleczko

    Kingsley Edore

    Abir Rizwanullah

    Sal Ahmed

Assessment 2 Team

Team 12

    Richard Liiv

    Umer Fakher

    William Walton

    James Frost

    Olly Wortley

    Joe Cambridge

Requirements Specification v0.5

Preface

This document is intended to be read by the System Engineers as well as the Stakeholder. The document is added to as new requirements arise (refer to version history for this document's various iterations).

- v0.1 Creation of the document, outlining the Introduction.
- v0.2 User, functional and non-functional requirements have been added but require revision.
- v0.3 System requirements have been revised. System Requirements added to as we checked back and found childless User Requirements. The additions could be revised.
- v0.4 Notes section in the User Requirements added to. This needs to be revised and linked to the Risk Assessment and Mitigation.
- v0.5 Added to and revised. Final version.

Introduction

Detailed below are the user requirements we gathered from the product brief and refined into system requirements in team-customer meetings between the development team and the client. The software is described in its entire scope; each requirement describes the game in enough detail to meet all of the client's needs.

The system itself is to be a single-player video game imitating the York Dragon Races, in which the player controls a boat in a top down view as it traverses a river in a race against other boats, controlled by the computer. The system will be written in Java and shall be available as a zip file, consisting of the code and an executable JAR file containing any external dependencies, and thus would be OS independent.

The following requirements are presented through natural language, without excessive technicalities so that it can be easily understood by the customer. They are written in a concise format so that confusion caused by ambiguity will be avoided, but have enough detail so software developers have enough guidance. Each record in the tables represents an individual requirement, referred to by its ID.

User Requirements

ID	Descriptions	Priority	Notes
UR_BOAT_UNIQUENESS	Every boat involved in the game must be unique in terms of speed, acceleration, maneuverability and robustness.	Shall	We should have every boat have its own strengths and weaknesses.
UR_DIFFICULTY_LEVEL	For each subsequent leg in the game, the difficulty of the game will increase. (This could be the rate at which obstacles appear during the race.)	Shall	The difficulty shouldn't be to the extent that the player loses hope in the game, nor so low that they become disinterested.
UR_PADDLERS_STAMINA_DECREASE	Paddlers in the teams get tired over time, so their boat's speed, acceleration and maneuverability decrease progressively during every leg.	Shall	The movement of the paddlers should still be of a visually engaging speed.
UR_PLAYER_PENALTY	Every boat must remain in its lane for the duration of the race. Leaving the lane may result in a penalty at the discretion of the chief race official.	Shall	The boundary detection shouldn't be consistent and not seem buggy to the player.
UR_OBSTACLES	Obstacles will be present in the game and teams may find these obstacles in the river during the race.	Shall	There may be so many obstacles that the player cannot get past without hitting them and being damaged.
UR_OBSTACLE_COLLISION	When a boat collides against obstacles, this will progressively reduce the robustness of the boat, until it breaks down (resulting in the end of the game).	Shall	The system must spawn obstacles appropriately and not cause the player's avatar's health to be reduced to a great amount too easily.
UR_UX	The game should be playable and enjoyable. It should offer a pleasant user experience.	Shall	This is very subjective and not always easy to measure. Associated risks are R6 and R7.
UR_MOVEMENT	The user must be able to move left, forward and right.	Shall	The user must be informed beforehand how to move their avatar.
UR_POWERUPS	The user may be able to pick up ~~powerups.~~ **five power-up packs, which can be found floating down the river and be picked up by boats to improve characteristics: health, agility, speed, stamina or a 5th one which improves all at once.**	~~May~~ Assessment 2 **SHALL**	~~We may not have enough time to implement this, due to time constraints. Associated with R8.~~
UR_RACE_TOTAL	The user must play three races before it is decided whether they are eligible to compete in the final.	Shall	It must be possible for the player to compete in the final.

UR_DIFFICULTY_BEFOR E_GAME	The user may be able to choose different difficulty settings before the game **(e.g. easy, normal, hard, ultra)**	~~May~~ Assess ment 2 **SHALL**	There will have to be a menu screen from which the player can select before the game actually starts.
UR_GAME_LENGTH	One whole game should last between 3 to 5 minutes.	Should	Each race should end in 1 approximately one minute.
UR_GAME_END	The game must end if the boat is broken, the user finishes 3 races and does not get into the final or if the user finishes the final.	Shall	We need to make sure it is possible for all three outcomes to occur.
UR_SAVE_RESUME_GA ME	**The players should be allowed to save the state of the game and resume a saved game later**	Assess ment 2 **SHALL**	**When the player saves by pausing the game it will save progress made up to the start of the last leg.**

Functional Requirements

ID	Description	User Requirements
FR_CHOOSING_UNIQUE_BO AT	The system must provide an option to choose a unique boat before the game starts.	UR_BOAT_UNIQUENE SS
FR_DIFFICULTY_SELECTIO N	The system allows the user to select a (initial) game level of difficulty from easy, medium ~~or hard.~~**, hard and Ultra.**	UR_DIFFICULTY_BEF ORE_GAME
FR_OBSTACLE_RATE	The system must spawn obstacles appropriately, depending on raceNo and/or difficulty.	UR_DIFFICULTY_LEV EL
FR_BOUNDARY_DETECTIO N	The system must be able to detect whether the user has left their individual lane.	UR_PLAYER_PENALT Y
FR_AWARD_PENALTY	The system must be able to award an appropriate penalty to a user that has left his lane.	UR_PLAYER_PENALT Y
FR_OBSTACLE_SPAWN	The system must not have obstacles spawn in a way that is impossible for the user to avoid colliding into them.	UR_OBSTACLES
FR_COLLISION_DETECTION	The system must be able to detect whether the user's boat has collided with an obstacle.	UR_OBSTACLE_COLL ISION
FR_HIT_DECREASED_BOAT _CONDITION	The system decreases the robustness of the boat if it has been hit by an obstacle.	UR_OBSTACLE_COLL ISION

FR_INPUT_DETECTION	The system must be able to detect key presses and user selections/clicks.	UR_MOVEMENT
FR_MOVEMENT	The system must register the input and act accordingly.	UR_MOVEMENT
FR_POWERUP_RATE	The system must decide on an appropriate amount of power ups to spawn during a race.	UR_POWERUPS
FR_BOAT_BREAKAGE	The system must be able to detect whether the boat is broken.	UR_GAME_END
FR_QUALIFIER_RACES	The system allows the user to play at least three boat races.	UR_RACE_TOTAL
FR_FINAL_RACE	The system allows the user to play in the championship finals race (on top of the other three), if they have performed well enough previously.	UR_RACE_TOTAL
FR_TIMED_DECREASED_BO AT_CONDITION	The system should have the boat's speed, acceleration and maneuverability decrease progressively over time throughout the race.	UR_PADDLERS_STA MINA_DECREASE

Non-Functional Requirements

ID	Description	User Requirements	Fit Criteria
NFR_MOVEMENT_E XPLANATION	The system must alert the user to the arrow keys to control the boat	UR_UX	Before the game starts via the help menu.
NFR_RULES_EXPLA NATION	The system must inform the user what the objective of the game is and generally how the game works.	UR_UX	Before the game starts.
NFR_POSITIVE_UX	The system should create anticipation and maintain the player's engagement.	UR_UX	Throughout the game.
NFR_END_SCREEN	The system must display an appropriate screen after the user has finished (either won or lost).	UR_GAME_EN D	After the final race or when the boat breaks.
NFR_ATTRIBUTES	The system must explain what the different attributes are and how they affect your boat. A **menu screen for boat selection should display the different attributes of each boat type.**	UR_BOAT_UNI QNESS	Before the game starts.
NFR_GAME_DURATI ON	The game is a short game that runs for a reasonably short length of time and has a short time to complete.	UR_GAME_LE NGTH	**The game leg can be completed by the user in typically around 30 seconds and the whole game may be completed in around 3/5 minutes.**

Bibliography

- "Software Engineering", Ian Sommerville, Chapter 4
- Omar Elgabry's Blog - "Requirements Engineering - Requirements Specification (Part 3)"
    - https://medium.com/omarelgabrys-blog/requirements-engineering-elicitation-analysis-part-5-2dd9cffafae8