

Test Plan

Black Box Testing

We have used Black Box Testing to test the requirements of the program. This has mostly been focused on the user requirements, however in instances where these tests did not cover requirements from the other categories, then new tests have been created. For example, NFR_USABILITY_TEST was created to test the usability requirements from the non-functional-requirements section that weren't covered by the user requirements.

This testing was performed both during and after the creation of the program. Tests on initial requirements could be performed before the additional requirements from assessment 2 were implemented.

Black box testing was used for the requirements because it was not necessary to see the code in order to check that the requirements have been met. For example, simply attempting to play the game will reveal whether the boat moves as intended, if the obstacles spawn evenly, and if the game progresses from stage to stage correctly.

Therefore, the testing served several processes. By playing the game it could be seen whether there are any obvious initial bugs that could then be fixed. It would also help highlight if there were any features promised in the requirements that were not implemented.

Due to the nature and simplicity of the game, there was not a huge variety of different inputs to test. The main menus used the mouse click and scroll functionality and the main game used WASD to control movement (although "S" wasn't used as backwards movement wasn't a feature). The only other inputs were ESC to return to the main menu, and SPACE to move between stages.

A traceability matrix was created for the Black Box testing. This was useful as it highlighted that some requirements in the NFR section were not tested by the existing tests. Furthermore, it shows that some requirements are tested by other tests, which limits the number of tests that must be created. This was the case with the functional requirements, all of which were covered by the tests that were derived from the user requirements. In turn this helps to reduce redundant tests.

White Box Testing

We will not be using white box testing. This is because of the nature of designing a game that interacts with the player; there is a constant player interaction and feedback loop that makes it impossible to account for all circumstances.

e.g. updating the boat's stats through a powerup doesn't account for all the minor possibilities before hitting the powerup.

Another limitation for white-box testing in LibGDX is that the headless client in LibGDX has some missing features, which crashes the tests every time we try to instantiate an Entity.

Test Case	Input	Expected output	Actual Output	Success?
UR_BOAT_UNIQUENESS_TEST	N/A	Each boat will look and act different	Boats are unique and have noticeably different stats	✓
UR_DIFFICULTY_LEVEL_TEST	N/A	The difficulty increases in each leg	Each leg feels harder than the last	✓
UR_PADDLERS_STAMINA_DECREASE_TEST	W	Stamina decreases throughout the leg	The stamina noticeably decreases, and refills when letting go of the forward button	✓
UR_PLAYER_PENALTY_TEST	N/A	Leaving the required lane results in a time penalty	Upon finishing the leg, penalties are added for leaving the lane	✓
UR_OBSTACLES_TEST	N/A	Obstacles spawn along the river	Obstacles spawn evenly throughout the leg	✓
UR_OBSTACLE_COLLISION_TEST	W A D	The robustness of the boat decreases upon hitting an obstacle, until the boat breaks	Boat takes damage from obstacles until it breaks	✓
UR_MOVEMENT_TEST	W A D	User can move forward, left and right	Successfully implemented movement	✓
UR_POWERUPS_TEST	W A D	The user can pick up power ups	Power-ups vanish when collided with. Health/Stamina visibly increase.	✓
UR_RACE_TOTAL_TEST	SPACE (transition between races)	The user plays three races before the final	Three races are played if the user's boat doesn't break	✓
UR_DIFFICULTY_BEFORE_GAME_TEST	MOUSE CLICK	The user can select a difficulty before starting	There is a difficulty select screen after choosing a boat	✓
UR_GAME_LENGTH_TEST	N/A	The game lasts between 3 and 5 minutes	Successful, the game lasts around 4 mins	✓
UR_GAME_END_TEST	ESC (return to main menu)	The game ends when: User boat is broken, User doesn't reach the final; User completes	Successful. The game ends as intended for each instance with an appropriate end screen	✓

		the final		
NFR_USABILITY_TEST	MOUSE CLICK MOUSE SCROLL	There are features in place to help usability. These include a help screen and an explanation of the game.	Help screen is implemented however it cannot be scrolled without a physical mouse. Scrolling with arrow keys or touch pad does not work.	☐
UR_SAVE_TEST	1 2 3	At any point during the race, the game can be saved	Pressing escape allows you to save the game using the number keys	✓
UR_LOAD_TEST	1 2 3	There is an option to load up a saved game from the main menu and continue playing	The load game button takes you to a screen where the user can choose from the three slots to load from. Game has same health, stamina and positions as before.	✓
UR_ROUNDS_TEST	Escape	Playing the game more than once resets rounds to 1.	After round 3, game goes to round 6, then round 7 with only 4 boats	☐

Traceability Matrix

Requirement Identifiers	Reqs tested	UR 1	UR 2	UR 3	UR 4	UR 5	UR 6	UR 7	UR 8	UR 9	UR1 0	UR1 1	UR1 2	UR1 3	UR1 4	UR1 5
Test cases	57	1	1	1	1	2	1	6	1	1	1	1	1	2		
UR_BOAT_UNIQUENESS_TEST	5	x						x								
UR_DIFFICULTY_LEVEL_TEST	4		x					x								
UR_PADDLERS_STAMINA_DECREASE_TEST	2			x												
UR_PLAYER_PENALTY_TEST	3				x											
UR_OBSTACLES_TEST	4					x										
UR_OBSTACLE_COLLISION_TEST	7					x	x							x		
UR_MOVEMENT_TEST	5							x	x							
UR_POWERUPS_TEST	4							x		x						
UR_RACE_TOTAL_TEST	3										x					
UR_DIFFICULTY_BEFORE_GAME_TEST	4							x				x				
UR_GAME_LENGTH_TEST	4							x					x			
UR_GAME_END_TEST	4													x		
NFR_USABILITY_TEST	3															
UR_SAVE_TEST	2														x	
UR_LOAD_TEST	3															x

Requirement Identifiers	Reqs tested	FR 1	FR 2	FR 3	FR 4	FR 5	FR 6	FR 7	FR 8	FR 9	FR1 0	FR1 1	FR1 2	FR1 3	FR1 4	FR1 5	FR1 6	FR1 7	FR1 8	FR1 9
Test cases	57	1	1	2	1	1	1	1	1	1	1	1	2	1	1	1	1			
UR_BOAT_UNIQUENESS_TEST	5	x																		
UR_DIFFICULTY_LEVEL_TEST	4			x																
UR_PADDLERS_STAMINA_DECREASE_TEST	2															x				
UR_PLAYER_PENALTY_TEST	3				x	x														
UR_OBSTACLES_TEST	4			x			x													
UR_OBSTACLE_COLLISION_TEST	7							x	x				x							
UR_MOVEMENT_TEST	5									x	x									
UR_POWERUPS_TEST	4											x								
UR_RACE_TOTAL_TEST	3														x	x				
UR_DIFFICULTY_BEFORE_GAME_TEST	4		x																	
UR_GAME_LENGTH_TEST	4														x					
UR_GAME_END_TEST	4												x							
NFR_USABILITY_TEST	3																			
UR_SAVE_TEST	2																	x		
UR_LOAD_TEST	3																		x	x

Requirement Identifiers	Reqs Tested	NFR1	NFR2	NFR3	NFR4	NFR5
Test cases	57	1	1	10	1	1
UR_BOAT_UNIQUENESS_TEST	5			x		x
UR_DIFFICULTY_LEVEL_TEST	4			x		
UR_PADDLERS_STAMINA_DECREASE_TEST	2					
UR_PLAYER_PENALTY_TEST	3					
UR_OBSTACLES_TEST	4			x		
UR_OBSTACLE_COLLISION_TEST	7			x		
UR_MOVEMENT_TEST	5			x		
UR_POWERUPS_TEST	4			x		
UR_RACE_TOTAL_TEST	3					
UR_DIFFICULTY_BEFORE_GAME_TEST	4			x		
UR_GAME_LENGTH_TEST	4			x		
UR_GAME_END_TEST	4			x	x	
NFR_USABILITY_TEST	3	x	x	x		
UR_SAVE_TEST	2					
UR_LOAD_TEST	3					

Evidence of Testing Links:

<https://github.com/UoY2021Eng1Group22/UoY2021Eng1Group22.github.io/blob/master/Assessment2Deliverables/EvidenceOfTesting.pdf>

<https://github.com/UoY2021Eng1Group22/UoY2021Eng1Group22.github.io/blob/master/Assessment2Deliverables/Powerups%20%2B%20Difficulty.mp4>

<https://github.com/UoY2021Eng1Group22/UoY2021Eng1Group22.github.io/blob/master/Assessment2Deliverables/SavingLoading.mp4>