# Testing Plan

**Project Name:**

*UMA-ISE24-E1*

Authors:

*Ricardo Juan Allit López, Manuel Barrios Moreno, Ángel Bayón Pazos, Ángel Nicolás Escaño López, Pablo Hormigo Jiménez, Francisco Javier Jordá Garay, Diego Sicre Cortizo, Muhammad Abdullah Sultán Sultán, Juan Torres Gómez.*

## Test Case Overview

**Test Case ID:**

*TC\_Navigation.*

**Purpose:**

*Verify that user can navigate through the menus using the selected controls in expected response time.*

*This test cases involves FR018(Controls), FR005(Main Menu), FR016(Boat Selection Menu) and NFR003(Low Latency Responses).*

**Test Case Description:**

*This test case evaluates the functionality of controls specifically for menu navigation within the game interface. It aims to ensure that players can traverse through main menu and boat selection screens and select options without encountering any issues or delays.*

*Related JUnit tests will check methods relative to the control’s listeners and event handlers of the keyboard.*

### Pre-Conditions

**Prerequisites:**

*Game is installed and running successfully.*

*User is in main menu.*

*Controls settings are set to default (Arrows).*

**Test Data:**

*Not applicable.*

### Test Steps

**Step Description:**

*1. Press up and down arrow keys and verify that the different menu options are highlighted.*

*2. Press enter key on "Boat selection" and verify that it is open.*

*3. Press esc key and verify that the main menu is open.*

*4. Press enter key on "Settings" and verify that the settings menu is opened.*

*5. Press esc key and verify that the main menu is open.*

*6. Press enter key on "Tutorial" and verify that the tutorial is opened.*

*7. Press esc key and verify that the main menu is open.*

*8. Press enter key on "Credits" and verify that the credits are opened.*

*9. Press esc key and verify that the main menu is open.*

*10. Repeat navigation process multiple times to check for consistency and reliability.*

*11. Press enter key on "Exit" and verify that the game is closed.*

### Post-Conditions

**Expected Outcome:**

*Response time of each operation is less than 30ms.*

**Cleanup:**

*Not applicable.*

### Notes

*This specific test can be performed with different control settings after the control’s customization test is completed.*

## Test Case Overview

**Test Case ID:**

*TC\_BoatSelection*

**Purpose:**

*Verify that user can select a boat from the boat selection menu.*

*This test cases involves FR018(Controls), FR016(Boat Selection Menu) and NFR003(Low Latency Responses).*

**Test Case Description:**

*This test case ensures that the user can select a boat from the boat selection menu. It verifies that the variable that stores the selected boat is updated correctly and that the game can load the selected boat.*

*JUnit tests will check methods related to the controls listeners and event handlers related to the keyboard and the change of playerBoat variable.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*User is in boat selection menu.*

*Controls settings are set to default (Arrows).*

**Test Data:**

*playerBoat*

### Test Steps

**Step Description:**

*1. Press left and right arrow keys and verify that the different boats are highlighted.*

*2. Press enter key on a boat and verify that it is selected.*

*3. Press enter key on other boats and verify that the selected boat is updated correctly.*

### Post-Conditions

**Expected Outcome:**

*playerBoat variable is updated correctly.*

*Response time of each operation is less than 30ms.*

**Cleanup:**

*Set playerBoat to default value.*

*Close the game.*

### Notes

*This test can be performed with different control settings after the control customization test is completed.*

## Test Case Overview

**Test Case ID:**

*TC\_ControlCustomization.*

**Purpose:**

*Verify that the user can customize the controls.*

*This test cases involves FR018(Controls).*

**Test Case Description:**

*This test case checks whether the control related tests can be performed with different control settings.*

*JUnit tests related to this test case will check the methods that update the variables related to keyboard listeners.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*User is in settings.*

*Controls settings are set to default (Arrows).*

**Test Data:**

*Key binding variables.*

### Test Steps

**Step Description:**

*1. Press "Customize Controls" on the settings screen.*

*2. Desired new key binding for UP, DOWN, LEFT, RIGHT, ENTER and ESC keys are selected whenever it is informed on screen.*

*3. Press "Save" and verify that the new key bindings are saved.*

*4. Perform the above tests with the new key bindings.*

### Post-Conditions

**Expected Outcome:**

*Key binding variables are updated correctly.*

*Tests related to navigation and boat selection are performed successfully with the new key bindings.*

**Cleanup:**

*Set key bindings to default values.*

*Close the game.*

### Notes

*This test depends on the navigation and boat selection tests.*

## Test Case Overview

**Test Case ID:**

*TC\_Player*

**Purpose:**

*Verify that boats have the correct properties and if properties are updated appropriately when adding user inputs.*

*This test cases involves FR016(Boat selection menu), FR018(Controls), FR002(Boats) and FR001(Player).*

**Test Case Description:**

*This test case checks whether the selected boat by the player exists and is accurate with the one selected by the boat selection menu and that the player can move perfectly.*

*JUnit tests related to this test case will check the numbers related to the player are equal to each property of the boat selected and with the test data, in case selection menu is incorrect. After this, the test will ensure the player only controls a unique player boat and that it follows all restrictions provided for the player, for example altering its x and y values appropriately with its speed property.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*Game is in the selection menu.*

**Test Data:**

*Structure with n number of boats with their properties.*

### Test Steps

**Step Description:**

*1. Selecting the desired boat from the selection menu.*

*2. Movement of the player is tested.*

*3. Quitting the race.*

*4. Selecting the next desired boat.*

*5. Steps are repeated until all boats are tested.*

### Post-Conditions

**Expected Outcome:**

*JUnit test verifies and returns if all boat properties were the same or not inside player, selection menu and test data.*

*JUnit test verifies player can alter with input uniquely the properties it is allowed to change.*

**Cleanup:**

*Close the game.*

### Notes

*This test depends on the navigation, player and boat selection tests.*

## Test Case Overview

**Test Case ID:**

*TC\_Lanes*

**Purpose:**

*Verify that every player has its own space inside the race to clarify where the user must stay during it.*

*This test cases involves FR020(Lanes).*

**Test Case Description:**

*This test evaluates the player's ability against the division of the entire map into lanes to check all illegal moves that the user may cause.*

*This ensures that players can stand their ground and that if they do not comply with this rule, we can penalize them for the illegal action they have committed.*

### Pre-Conditions

**Prerequisites:**

*Game is installed and running.*

*User has already selected a boat and is inside a race.*

**Test Data:**

*Not applicable*

### Test Steps

**Step Description:**

*1. Enter a race with the user selected boat.*

*2. Check if the collision between the boat and the division lane is well implemented.*

*3. Check that the lane collision penalty is correctly applied to the player.*

### Post-Conditions

**Expected Outcome:**

*In case the player collides with a division line, then the player must die, and a Game Lost screen must pop up to tell the user that the race has ended because of an illegal movement.*

**Cleanup:**

*Close the game.*

### Notes

*This test can be performed with any controls and it's just to verify that the lanes between competitors are not crossable.*

## Test Case Overview

**Test Case ID:**

*TC\_Tutorial*

**Purpose:**

*Verify that all users can take a training run to familiarize themselves with the game methodology and learn the basic controls.*

*This test cases involves FR008(Tutorial).*

**Test Case Description:**

*With this test we will check the full functionality of the test run as well as the readiness it gives users to get involved in a real race.*

### Pre-Conditions

**Prerequisites:**

*Game is installed and running.*

*User selects the tutorial section in the main menu.*

*Control settings have been already modified (default as well if no modification is needed).*

**Test Data:**

*Not applicable.*

### Test Steps

**Step Description:**

*1. Enter the tutorial and check if it starts correctly.*

*2. If the tutorial gives the user the instructions to move, check whether the controls are well specified or not (if user has changed its control inputs).*

*3. verify that all tutorial purposes can be completed without any system failures.*

*4. Finish the tutorial and check if the user has successfully retreated to the main menu.*

*5. Pressing the "Exit" button in the middle of the tutorial to check if the user can end the tutorial whenever he wants.*

### Post-Conditions

**Expected Outcome:**

*After the tutorial is completed, the user must see a screen telling him that the tutorial has been successfully completed and can return to the main menu.*

*If the user decides to close the tutorial, he must go back to the main menu screen.*

**Cleanup:**

*Close the game.*

### Notes

*This test may have multiple outcomes because the user has the privilege of exiting the tutorial whenever he wants.*

## Test Case Overview

**Test Case ID:**

*TC\_Credits*

**Purpose:**

*Verify that the user can see a final with all the people involved in the creation of the project in the main menu section.*

*This test cases involves FR022(Credits).*

**Test Case Description:**

*This test ensures that the user can see the final credits screen from the main menu.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running.*

*Users are in the main menu section.*

**Test Data:**

*Not applicable.*

### Test Steps

**Step Description:**

*1. Press the "Credits" button inside the main menu.*

*2. After the execution of the credits, press the return to main menu button.*

### Post-Conditions

**Expected Outcome:**

*The user, after pressing the credits button, must see the credits screen and when it finishes, then he must be redirected to the main menu section.*

**Cleanup:**

*Close the game.*

### Notes

*This test depends on the main menu section and the controls of the user in the menus.*

*The user can quit this section whenever he wants by pressing the pause button and selecting the return to main menu option.*

## Test Case Overview

**Test Case ID:**

*TC\_Levels*

**Purpose:**

*Verify that the user experiences a difference in difficulty through the race.*

*This test case involves FR009 (Legs), FR003 (Levels), FR014(Leg Duration).*

**Test Case Description:**

*This test case evaluates the level of difficulty of each leg which entails.*

*the level of the opposing boats, quantity of obstacles and the speed at which the obstacles move.*

*It aims to ensure that players can acknowledge the changes in difficulty between the 3 legs.*

*Related JUnit tests will check methods relative to the legs.*

Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*User is in main menu.*

*User initiates the race by pressing the play button in main menu.*

**Test Data:**

*Not applicable.*

Test Steps

**Step Description:**

*1. Register time of the run produced by ai in leg 1.*

*2. Time speed of objects movements in leg 1.*

*3. Register number of obstacles in leg.*

*3. Register time of the run produced by ai in leg 2.*

*4. Time speed of objects movements in leg 2.*

*5. Register number of obstacles in leg.*

*6. Register time of the run produced by ai in leg 3.*

*7. Time speed of objects movements in leg 3.*

*8. Register number of obstacles in leg.*

*9. Compare the results obtained with each other to ensure a visible change in difficulty.*

Post-Conditions

**Expected Outcome:**

*Object movements speed increase, obstacle amount increases and ai leg times decrease as you advance in the legs.*

**Cleanup:**

*Not applicable.*

Notes

*Must be run with all 3 boats to check that all 3 boats can beat these difficulties.*

*The collision of the ai boats will vary from run to run as they have a level of randomness.*

## Test Case Overview

**Test Case ID:**

*TC\_Legs*

**Purpose:**

*Verify that the user can play all three legs in the race if the previous legs have been completed.*

*This test case involves FR009 (Legs), FR003 (Levels), FR014(Leg Duration).*

**Test Case Description:**

*This test case evaluates the functionality of the 3-leg system inside the level including the change in difficulty and scenery. It aims to ensure the correct transitioning in the race for the player.*

*Related JUnit tests will check methods relative to the clock, the control listeners and the objects.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*User is in main menu.*

*User initiates the race by pressing the button in main menu.*

**Test Data:**

*Not applicable.*

### Test Steps

**Step Description:**

*1. Complete the first leg in the race.*

*2. Verify the boat moves correctly.*

*3. Check the difficulty has been increased by confirming the change in speed*

*4. Check the scenery has changed*

*5. Check the quick menu is still accessible*

*6. Repeat process for all three legs and repeat several times to ensure correct functioning.*

### Post-Conditions

**Expected Outcome:**

*User changes legs without encountering any glitches or errors like not having a change in scenery.*

*Difficulty is modified.*

*The level is beatable.*

**Cleanup:**

*Not applicable.*

### Notes

*The test must be repeated several times and must be run with the three different boats.*

*To ensure all three boats can beat the levels and that none of them cause the race to crash.*

## Test Case Overview

**Test Case ID:**

*TC\_LegDuration*

**Purpose:**

*Verify the length of the leg is approximately 60 seconds.*

*This test case involves FR009 (Legs), FR003 (Levels), FR014(Leg Duration).*

**Test Case Description:**

*This test case evaluates the duration of each leg of the race.*

*It aims to ensure the requirements set by the client are met.*

*Related JUnit tests will check methods relative to clock.*

### Pre-Conditions

**Prerequisites:**

*The game is installed and running successfully.*

*User is in main menu.*

*User initiates the race by pressing the button in main menu.*

**Test Data:**

*Not applicable.*

### Test Steps

**Step Description:**

*1. Start a timer.*

*2. Complete leg.*

*3. Stop timer.*

*4. repeat the process several times to ensure the nonexistence of any anomalies.*

### Post-Conditions

**Expected Outcome:**

*Leg duration is at maximum at the 60 second mark.*

**Cleanup:**

*Not applicable.*

### Notes

*The test must be performed with all three boats to ensure none of them are above the threshold.*