Purpose

UAT (user acceptance testing) is important for this mobile game as application testing with users ensures that the mechanics and features that have been implemented are functional and meets the needs of the users.

Scope

* Player movement
* Platform spawning
* Platform jumping
* Score system

Objective

Player movement should function as the user expects it to. The player game object should move fluidly with the movements of the user at their command

## Test

---

**Test Owner** -

**Environment**

|  |  |  |  |
| --- | --- | --- | --- |
| User | Screen size | Operating System | Phone Model |
| Example | 16:9 Portrait | Android | Samsung Galaxy S10 |
|  |  |  |  |

Player Movement

**Test Case** - Verify that the user can interact with the player game object and that it functions as they expect it to.

**Expected Outcome** - The user should touch and hold the player game object with one finger on the touchscreen and drag it left and right. The game object should follow the movements of the user along the X axis exclusively.

**Result** –

**Pass/Fail** -

Platform Spawning

**Test Case** - Verify that the platforms spawn continuously as the user is playing.

**Expected Outcome** - Every time the user jumps on a platform, a new platform should spawn outside of the screen view to create a seamless illusion that there are endless platforms.

**Result** –

**Pass/Fail** -

Platform Jumping

**Test Case** – Verify that the user can jump to and from every platform.

**Expected Outcome** – The player object should be continuously bouncing so that the user can jump and reach every platform in the scene.

**Result** –

**Pass/Fail** -

Score System

**Test Case** – Verify that the user’s high score (height) gets updated in game and is saved in player preferences to then be displayed.

**Expected Outcome** – Every time the user enters the main scene, the height starts at “0m” and increments by one with each platform they land on. If the user exits the game or returns to the main menu, their high score is displayed on the title screen.

**Result** –

**Pass/Fail** –

**Test Owner** -

**Environment**

|  |  |  |  |
| --- | --- | --- | --- |
| User | Screen size | Operating System | Phone Model |
| Example | 16:9 Portrait | Android | Samsung Galaxy S10 |
|  |  |  |  |

Player Movement

**Test Case** - Verify that the user can interact with the player game object and that it functions as they expect it to.

**Expected Outcome** - The user should touch and hold the player game object with one finger on the touchscreen and drag it left and right. The game object should follow the movements of the user along the X axis exclusively.

**Result** –

**Pass/Fail** -

Platform Spawning

**Test Case** - Verify that the platforms spawn continuously as the user is playing.

**Expected Outcome** - Every time the user jumps on a platform, a new platform should spawn outside of the screen view to create a seamless illusion that there are endless platforms.

**Result** –

**Pass/Fail** -

Platform Jumping

**Test Case** – Verify that the user can jump to and from every platform.

**Expected Outcome** – The player object should be continuously bouncing so that the user can jump and reach every platform in the scene.

**Result** –

**Pass/Fail** -

Score System

**Test Case** – Verify that the user’s high score (height) gets updated in game and is saved in player preferences to then be displayed.

**Expected Outcome** – Every time the user enters the main scene, the height starts at “0m” and increments by one with each platform they land on. If the user exits the game or returns to the main menu, their high score is displayed on the title screen.

**Result** –

**Pass/Fail** –

**Test Owner** -

**Environment**

|  |  |  |  |
| --- | --- | --- | --- |
| User | Screen size | Operating System | Phone Model |
| Example | 16:9 Portrait | Android | Samsung Galaxy S10 |
|  |  |  |  |

Player Movement

**Test Case** - Verify that the user can interact with the player game object and that it functions as they expect it to.

**Expected Outcome** - The user should touch and hold the player game object with one finger on the touchscreen and drag it left and right. The game object should follow the movements of the user along the X axis exclusively.

**Result** –

**Pass/Fail** -

Platform Spawning

**Test Case** - Verify that the platforms spawn continuously as the user is playing.

**Expected Outcome** - Every time the user jumps on a platform, a new platform should spawn outside of the screen view to create a seamless illusion that there are endless platforms.

**Result** –

**Pass/Fail** -

Platform Jumping

**Test Case** – Verify that the user can jump to and from every platform.

**Expected Outcome** – The player object should be continuously bouncing so that the user can jump and reach every platform in the scene.

**Result** –

**Pass/Fail** -

Score System

**Test Case** – Verify that the user’s high score (height) gets updated in game and is saved in player preferences to then be displayed.

**Expected Outcome** – Every time the user enters the main scene, the height starts at “0m” and increments by one with each platform they land on. If the user exits the game or returns to the main menu, their high score is displayed on the title screen.

**Result** –

**Pass/Fail** –

**Test Owner** -

**Environment**

|  |  |  |  |
| --- | --- | --- | --- |
| User | Screen size | Operating System | Phone Model |
| Example | 16:9 Portrait | Android | Samsung Galaxy S10 |
|  |  |  |  |

Player Movement

**Test Case** - Verify that the user can interact with the player game object and that it functions as they expect it to.

**Expected Outcome** - The user should touch and hold the player game object with one finger on the touchscreen and drag it left and right. The game object should follow the movements of the user along the X axis exclusively.

**Result** –

**Pass/Fail** -

Platform Spawning

**Test Case** - Verify that the platforms spawn continuously as the user is playing.

**Expected Outcome** - Every time the user jumps on a platform, a new platform should spawn outside of the screen view to create a seamless illusion that there are endless platforms.

**Result** –

**Pass/Fail** -

Platform Jumping

**Test Case** – Verify that the user can jump to and from every platform.

**Expected Outcome** – The player object should be continuously bouncing so that the user can jump and reach every platform in the scene.

**Result** –

**Pass/Fail** -

Score System

**Test Case** – Verify that the user’s high score (height) gets updated in game and is saved in player preferences to then be displayed.

**Expected Outcome** – Every time the user enters the main scene, the height starts at “0m” and increments by one with each platform they land on. If the user exits the game or returns to the main menu, their high score is displayed on the title screen.

**Result** –

**Pass/Fail** –

## Evaluation

|  |  |  |  |
| --- | --- | --- | --- |
| User | Functionality | Usability | Overall Experience |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## Reporting

Summary (testing process, coverage, overall findings)

Open issues and bugs

Recommendations for improvements based on test results and user feedback