Test Plan

Prepared for: Ceros Ski Manual Testing

Prepared by:
Bendon Murgor
Version 1.0 – 05-05-2021

Table of Contents

1. Introduction

This test approach document describes the appropriate strategies, process, workflows and methodologies used to plan, organize, execute and manage testing of the game Ceros Ski.

2. Scope

In Scope

The ceros ski *Test Plan* defines the unit, integration, system, regression, and Client Acceptance testing approach. The test scope includes the following:

- Testing of all functional, application performance, security and use cases requirements.·
- Quality requirements and fit metrics Ceros Ski
- End-to-end testing and testing of interfaces of all systems that interact with the Ceros Ski

Out of Scope

The following are considered out of scope for Ceros Ski Test Plan and testing scope:

- Functional requirements testing for game outside
- Testing of Business SOPs, disaster recovery and Business Continuity Plan

3. Quality Objective

Primary Objective

- Ensure the Ceros Ski conforms to functional and non-functional requirements
- Ensure the AUT meets the quality specifications defined by the client
- Bugs/issues are identified and fixed before go live

Secondary Objective

Identify and expose all issues and associated risks, communicate all known issues to the project team, and ensure that all issues are addressed in an appropriate matter before release. As an objective, this requires careful and methodical testing of the game to first ensure all areas of the Ceros Ski are scrutinized and, consequently, all issues (bugs) found are dealt with appropriately.

At the end of the software testing life cycle, the tester should find that the game has met or exceeded all of their expectations as detailed in the requirements.

4. Features to be Tested

For the Ceros Ski game, the following features will be tested:

- The game options game start, game pause, skiing through the rocks and trees is seamless
- The event/object triggers, and the scoring.
- New Feature: Jump, with the below acceptance criteria;

Acceptance Criteria;

- Have the skier jump by pressing a key
- Have the skier jump whenever he hits a ramp.
- The skier should be able to jump over some obstacles while in the air.
    - Rocks can be jumped over
    - Trees can NOT be jumped over

5. Assumptions for Test Execution

For User Acceptance testing, the Developer team has completed unit, system and integration testing and met all the Requirements (including quality requirements) based on the Requirement Traceability Matrix of the game.

- User Acceptance testing will be conducted by End-users
- Test results will be reported on a daily basis. Failed scripts and defect list with evidence will be sent to the Developer directly.
- Use cases have been developed by Adopters for User Acceptance testing. Use cases are approved by test lead.
- Test scripts are developed and approved.
- Test Team will support and provide appropriate guidance to Adopters and Developers to conduct testing
- Major dependencies should be reported immediately after the testing kickoff meeting.

6. Risks

The following risks have been identified and the appropriate action identified to mitigate their impact on the project. The impact (or severity) of the risk is based on how the project would be affected if the risk was triggered. The trigger is what milestone or event would cause the risk to become an issue to be dealt with.

#	Risk	Impact	Trigger	Mitigation Plan
1	Scope Creep – as testers become more familiar with the tool, they will want more functionality	High	Delays in implementation date	Each iteration, functionality will be closely monitored. Priorities will be set and discussed by stakeholders. Since the driver is functionality and not time, it may be necessary to push the date out.
2	Changes to the functionality may negate the tests already written and we may lose test cases already written	High – to schedule and quality	Loss of all test cases	Export data prior to any upgrade, massage as necessary and re-import after upgrade.

7. Test Methodology

Overview: Agile test methodology because:

Developers and clients are emphasized rather than processes and tools. The agile methodology focuses on responding to change rather than extensive planning and this would ensure that any bugs in the game "Ceros Ski" are fixed before new releases.

The agile methodology uses incremental testing hence every release of a new feature in the game "Ceros Ski" the jump feature will have undergone thorough testing. This minimizes risks and it is possible to make changes in the project at any time to comply with the requirements.

8. Test Levels

N/B **If any Level 1 test case fails, the build is returned to developers un-tested.**

Level 1 - Smoke Tests

These test cases verify the major functionality at a high level.

Test Case	Description	Expected results
The game options game start, game pause, skiing through the rocks and trees is seamless	This is to test the major functionality of the game "Ceros Ski"	Up, down, left and right arrows allow player to have a smooth user experience Skiing through the rocks and trees

Bug Regression

Bug Regression will be a central tenant throughout all testing phases.

All bugs that are resolved as "Fixed, Needs Re-Testing" will be regressed when the Testing team is notified of the new drop containing the fixes. When a bug passes regression, it will be considered "Closed, Fixed".

If a bug fails regression, adopters testing team will notify development team by entering notes into Bugzilla. When a Severity 1 bug fails regression, adopters Testing team should also put out an immediate email to development. The Test Lead will be responsible for tracking and reporting to development and product management the status of regression testing.

Bug Triage

The goal of the triage is to:

- To define the type of resolution for each bug
- To prioritize bugs and determine a schedule for all "To Be Fixed Bugs'.

9. REVISIONS HISTORY

Revision No.	Date of Issue	Author	Description
1.	06-04-2021	Bendon Murgor	Functional testing
2.	06-04-2021	Bendon Murgor	Updated pre steps

10. Test Completeness

Testing will be considered complete when the following conditions have been met:

Standard Conditions:

- When Adopters and Developers, agree that testing is complete, the game is stable, and agree that the game meets functional requirements.
- Script execution of all test cases in all areas have passed.
- All priority 1 and 2 bugs have been resolved and closed
- Each test area has been signed off as completed by the Test Lead.
- 50% of all resolved severity 1 and 2 bugs have been successfully re-regressed as final validation.
- Ad hoc testing in all areas has been completed.

Test Deliverables and schedule

Test Schedule

Task Name	**Effort**
Test Planning	1 day
Review Requirements documents	0.5 days
Create initial test estimates	0.5 days
Functional testing – Iteration 1	0.5 days
Functional testing – Iteration 2	1 day
Regression testing	0.5 days
UAT	1 day
Performance testing	0.5 days

Deliverables

Deliverable	**For**	**Date / Milestone**
Test Plan	Project Manager; QA Director; Test Team	09/04/2021
Traceability Matrix	Project Manager; QA Director	09/04/2021
Test Results	Project Manager	12/04/2021
Test Status report	QA Manager, QA Director	13/04/2021
Metrics	All team members	15/04/2021

11. Resource & Environment Needs

Testing Tools

Requirements Tracking Tool:

- Use case diagram

  The use case diagram helps depict the interaction between the game and its users. Each user role is called an “actor” and different processes or functions of the game are represented in the diagram.

- Bug Tracking Tool

For Ceros Ski, we would use Bugzilla because it has:

- Email notification for change in code
- Reports and Charts
- Patch Viewers
- List of bugs can be generated in different formats

- Schedule daily, monthly and weekly reports
- This bug tracking tool detect duplicate bug automatically
- Setting bug priorities by involving customers
- Predict the time a bug may get fixed

12. Test environment

Since Ceros Ski is web based, tests will be done on different browsers to determine behavior:

Browser	Google Chrome	Mozilla Firefox	Microsoft Edge	Internet Explorer 11

13. Functional testing

Checking colors and backgrounds, screen orientation and screen resolution, font size, alignment errors, usability, system navigation such as loading time, timeout and display, sorting, sequences, animations User Interactions, User Interfaces, Transaction's testing, Calibration and accuracy Screen resolutions, Mobile responsive design testing.

14. Compatibility testing

This will be used to check if the game is compatible across different devices, and on different configurations of hardware and software.

15. Performance Testing

- Response time on client and servers, Transaction completion time(s), Peak load performance, Longevity, network coverage, Memory leakage, low memory, low battery, Time taken to download applications, simultaneous (Multiple users) access to application's server, speed, throughput, reliability, scalability, etc.
- **Processor and memory constraints**: Performance counters are used to measure the CPU and memory consumption of the game application.
- **Network connectivity**: Measures the response time of the game on different network types (Wi-Fi, 2G, 3G, 4G), It gives an overall insight into how well the game will perform on unreliable networks. The whole Peak Times, Jittery Connections, Duplication of Data, Packet loss, Fragmentation of Data are monitored.

16. Conformance /Compliance Testing

- The game targets a particular content rating. If there is an objectionable content that is inappropriate for the desired rating, then they are identified and reported. Even a single violation in submission for license approval may have the game rejected, incurring additional costs in further testing and resubmission.

17. Localization testing

- **If Cero Ski** is targeted for the global markets, this becomes essential.
- Game titles, content, and texts need to be translated and tested with devices in multiple languages.

18. Soak testing

- This involves leaving the game running for a prolonged period in various modes of operation.
- For example, Game has begun, and the character is made to stand idle for 24 hours.
- This technique is used to detect crashes brought on by memory leaks and other faults in the game engine.

19. Recovery testing

This checks how well the application can be recovered from crashes, hardware failures, and other similar failures. The application is forced to fail, and later should be observed how it recovers from the failure conditions and the environment.

For instance, while Ceros Ski is running, a player could restart the game console and check the validate data integrity in this case, if the scores have been interfered with, if the character has moved, if events have changed etc.

20. Security testing

This is to check how safe the game is from external threats.

21. Test Cases

Functional Test Cases

TC #	Scenario	Steps	Expected results	Test Results	Comments
1.	Start the game	1. Load the game on the chrome browser using the address https://ceros-ski-cem.herokuapp.com/ 2. Long Press the Down arrow for the sprite to start moving	Sprite should be able to ski downhill successfully	Game starts successfully	Pressing the Up arrow moves the sprite upwards
2.	Pause game	1. Start the game by clicking the Down arrow button 2. Click the Spacebar key	Game should pause and the scores should not disappear	Game pauses and the scores are not lost	
3.	Unpause game	1. Pause the game by clicking the spacebar and unpause by clicking the same button as well	User should be able to unpause the game successfully	Game is unpaused and the points are counted from the points that were scored before the game was paused	
4.	Collision	1. Move the sprite towards a tree 2. Move the sprite towards a rock	Collision should be triggered when a sprite comes into contact with these objects and the user should also be notified that a collision occurs. Game should stop immediately	Collision is triggered when the sprite comes into contact with a tree or a rock	Collison however does not occur when the sprite is moved horizontally.

Acceptance Tests

TC #	Scenario	Steps	Expected results	Test Results	Comments
1.	Jumping over a rock	1. Start the game 2. Click any key when the sprite is almost colliding with a rock	Sprite should be able to jump over rocks	Collision happens regardless of the number of keys user tries to click	
2.	Jumping over trees	1. Start the game 2. Click any key when the sprite is almost colliding with a tree	Sprite should not be able to jump over rocks	Sprite crashes when it collides with a tree	
3.	Jump over some obstacles while in the air.	1. Start the game 2. Click any key when the sprite is approaching any obstacle	Sprite should be able to jump over obstacles while in the air	Sprite cannot jump over obstacles while in air	
4.	Jump when sprite hits ramp	1. Start the game 2. Click any key when the sprite is approaching a ramp	Sprite should be able to jump when it hits the ramp	No ramp has been encountered therefore the sprite cannot jump when it hits the ramp	

Usability Test Cases

TC no.	Scenario	Steps	Expected results	Test Results	Comments
1	Restart the game after sprite has crashed	1. Start the game 2. Crash the sprite into an object 3. Click the Spacebar	Game restarts	Game restarts	This is not the case if the user has not crashed into an object but still wants to restart the game
2.	Controls	The main controls of the game are the UP,Down,Right and Left	These keys move the sprite in the	User can be able to control sprite with the keys	Introduce a specific jump key. User wastes time trying a

		arrow keys and the space bar	direction the user wants		combination of keys to just jump. Include the controls on the screen or add a menu with this information
3.	Notification	1. Crash sprite into a tree or a rock 2. Click the Spacebar	Sprite should crash first, then the notification to play again should appear after	Notification appears first, then after user restarts the game, the sprite crashes	