**IEEE 829 Test Plan Template**

# 

# 

# 

# 

# 

# 

# 

**Table of Contents**

[**Test Plan Identifier**](#_isen70isgcsf) **3**

[**Introduction**](#_5ey6oisgpksj) **4**

[**Test Items**](#_gm9l58commkz) **4**

[**Features To Be Tested**](#_lzrlonwys5aj) **4**

[**Features to be tested: (updated)**](#_ud5ol8gwck47) **5**

[**Features Not To Be Tested**](#_sx2gv65ocrxs) **5**

[**Approach**](#_2zhldmti7vcm) **6**

[**Item Pass/Fail Criteria**](#_r2xd23c8dd7s) **6**

[**Suspension Criteria And Resumption Requirements**](#_y4u8extf3qpg) **6**

[**Test Deliverables**](#_r4srcjnf43nz) **7**

[**Testing Tasks**](#_mjsp6wm8pihg) **7**

[**The following activities must be completed**](#_lxdyalbihjbp) **7**

[**10. Environments**](#_b9dtw3flrjh8) **7**

[10. 1 Software needs](#_i60jnen8d0ee) 7

[10. 2 Hardware needs](#_8rr3c4lfglgi) 8

[**11. Responsibilities**](#_44lmlgydkiva) **8**

[**12. Staffing And Training Needs**](#_ur48uf8469l1) **10**

[**13. Schedule**](#_szldiwogze4q) **11**

[**14. Risks And Contingencies**](#_na16ppo2fxz3) **13**

[**Approvals**](#_lpghdeoqasd8) **14**

# 

# 

# Test Plan Identifier

|  |  |
| --- | --- |
| Test Plan Name | SportZ Game Testing TP1.0 |
| Test Plan Version | 1.0 |
| Year of Creation | 2022 |
| QA Name | Chee Zi Hoe |
| Reference to other documents | Project Plan, SRS |

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description | Date | Author |
| 1.0 | Initial Draft | 01/04/22 | Hermes Lim |
|  |  |  |  |
|  |  |  |  |

# 

# 

# 

# 

# Introduction

This document specifies the strategy being used for the testing of the deliverables in project SportZ. This is done through outlining the features to be tested, approaches, specific pass/fail requirements as well as schedule.

This Test Plan encompasses the following levels of testing: Unit, Integration and System testing. The bulk of testing will be done for the unit and integration levels and the objective is to ensure the deliverable game SportZ is playable and fulfils requirements as per System Requirements Specifications.

# 

# 

# Test Items

The primary test items include:

* SportZ game application
  + Features of the game to be tested as highlighted in the section below
* Systems Requirement Specification to ensure traceability.

# Features To Be Tested

The following features are either unit level or integration level testing features which can be further separated into individual test cases.

|  |  |
| --- | --- |
| Feature to be tested | Priority(High, Medium, Low) |
| Open game | High |
| Start game | High |
| Pause game | High |
| Player movement(jump and attack) | High |
| Non playable characters and props generation | High |
| In-game ability enhancement(invincibility, coin magnet,etc) | High |
| In-game objects collision | High |
| Game logic(death, health depletion, score counting, etc) | High |
| Game theme transition | Medium |
| In game Shop | Medium |
| Adjust game settings(music volume, etc) | Medium |
| Credits | Low |

# Features Not To Be Tested

|  |  |
| --- | --- |
| Feature not to be tested | Reason |
| User interface design (colour, fonts, etc) | Subjected to user experience |
| Visual Effects in the game | Subjected to user experience |
| Time performance of the game | Subjected to user experience |
| Entertainment experience of the game | Subjected to user experience |
| Usability, user friendliness of the game | Subjected to user experience |

# Approach

The testing of the application is made up of Unit and Integration Tests. The test will be conducted using the **Black-Box Testing** technique for both levels of test. The tests will be done in Unity’s debug mode to observe in-game values and data in real time. The testing tool does not require special training as it is the same tool used for the development. The outcome of each test case will be recorded. As SportZ is a video game software, there will be more emphasis on the outcome of the various actions players can perform in the game.

# Item Pass/Fail Criteria

|  |  |  |
| --- | --- | --- |
| Feature to be tested | Pass Criteria | Fail Criteria |
| Open game | Able to launch the  Game application without any crashes | Failed to launch the game application due to crashes |
| Start game | Able to start  a game session after pressing the start button | Unable to start a game session after pressing the start button |
| Pause game | Enters the pause state  in a game session after clicking on the pause button | Unable to enter the pause state in the game session after pressing the pause button |
| Player movement(jump and attack) | Able to control the character with the movement keys available and performs the correct action according to the respective keys. Eg. “Space” to Jump | Unable to control the characters with the movement keys available or performs the wrong action after an action key pressed. Eg. “Space” performs attack instead of jump |
| Non playable characters (NPC) and props generation | Able to spawn different NPC and props on generated platform | Unable to spawn different NPC and props on the generated platform for the past 30 seconds |
| In-game ability enhancement(invincibility, coin magnet,etc) | Able to use the respective abilities after collecting the power-up | Unable to use the respective abilities of the power-up after collecting it |
| In-game objects collision | Able to collide or interact with different objects in the game. (Eg. player is able to collide with the platform) | Unable to collide or interact with the different objects in the game. (Eg. player falls through the platform) |
| Game logic(death, health depletion, score counting, etc) | Able to trigger various game logic in the game:  - Player dies upon falling off the platform  - Player taking damage after touching the obstacles and enemies  - Player score continue to increase the longer the player stays alive | Unable to trigger various game logic in the game:  - Player continues to stay alive after falling off the platform  - Player not taking damage after touching the obstacles and enemies  - Player score does not increase |
| Game theme transition | Able to trigger the game theme transition after staying alive for 30 seconds | Unable to see any game theme transition after staying alive for 30 seconds |
| In game Shop | Able to access the in-game shop and purchase or upgrade enhancement with sufficient coin | Unable to access the in-game shop or unable to purchase enhancement with sufficient coin |
| Adjust game settings(music volume, etc) | Able to adjust the music volume using the settings adjustment page | Unable to adjust the music volume using the settings adjustment page |
| Credits | Able to view the credits using the credit page | Unable to view the credits using the credit page |

Unit Level Testing:

* All test case completed
* Specified percentage (90%) of the test case completed with minor defects. If the test cases pass 90% of the time, it can be considered as test complete.

Integration Level Testing:

* All test case completed
* Specified percentage (90%) of the test case completed with minor defects. If the test cases pass 90% of the time, it can be considered as test complete.
* Integrated units able to communicate and interact flawlessly

System Level Testing:

* All test case completed
* Specified percentage (90%) of the test case completed with minor defects. If the test cases pass 90% of the time, it can be considered as test complete.
* The whole application runs smoothly and properly

Test exit criteria

* The test shall be completed when satisfactory code coverage is reached
* Sufficient amount of time spent for that specific test. If a specific test case has gone through 1-2 hour of testing and achieved a 90% passing rate, the test is considered complete. Otherwise, depending on the importance of the test case, it will be revised or bypassed.

# Function Testing:

* Ensure the inputs from users are processed correctly
* Ensure game can be started and played
* Ensure game score and earned coin are calculated accurately

User Interface Testing:

* Ensure font size is appropriate regardless display devices
* Ensure no dead-end path exists and the user should have an option to terminate any pop out window/menu.
* Ensure that user-interface is intuitive

Performance Testing:

* Ensure the loading time of starting game application is within 5 second
* Ensure the loading time to actual game sense from main menu is within 5 second
* Ensure minimum FSP is 60 during game play
* Verify the response time of user input during game play is unnoticeable or within 10 millisecond

Installation Testing:

* Ensure game application executable file can be successful run on devices with Window OS and Mac OS that are not from development teams

# Suspension Criteria And Resumption Requirements

**Criteria for Suspension:**

* If the application has a particular flaw that will hinder the testing process of other test cases due to dependency, the application will be sent back to the developers for revision as further testing will not yield any effective results.

**Criteria for Resumption:**

* The application will resume testing as soon as the development team has finished reviewing and rectifying the errors. Meanwhile, non-dependent test cases can still be tested.

# Test Deliverables

* Test plan document.
* Test cases and requirements test coverage report
* Problem reports and corrective actions.

# Testing Tasks

# The following activities must be completed

* Test plan prepared
* Identify items to be tested
* Identify method of testing
* Design test cases
* Functional specifications written and delivered to the testing team
* Perform the tests
* Address issues that arise during testing
* Repeat testing and bug fixing (if necessary)
* Prepare a test summary report

# 

# 10. Environments

## 10. 1 Software needs

Since this game is created using Unity Game engine and written for Windows OS, each tester should have an operating system running Windows. The test data in black box testing is limited to what is defined in the test case.

## 10. 2 Hardware needs

To conduct the testing, the minimum requirements needed for the hardware is as follows:

|  |  |
| --- | --- |
| Minimum Requirements | Windows |
| CPU | X64 architecture with SSE2 instruction set support |
| Graphics API | DX10, DX11 and DX12-capable GPUs |

Each tester will require a working desktop or laptop.

# 11. Responsibilities

|  |  |  |
| --- | --- | --- |
| Name | Role | Responsibilities |
| Lin Zixing | Project Manager/Test Manager | * Provides management oversight. * Plan and control the testing process. * Identify motivators. * Acquire appropriate resources. * Present management reporting. * Advocate the interests of the test. * Evaluate the effectiveness of the test. |
| Fabian Wong | Lead Developer | * Review results of test run to ensure software meets necessary requirements before launch. * Check if testing is going in hand with the software development in all phases. |
| Chia Songcheng | Test Designer | * Defines the technical approach to the implementation of the test effort. * Definite test approach * Define test automation architecture * Verify test techniques * Define testability elements * Structure test implementation. |
| Lim Sheng Zhe | Test Analyst | * Identifies and defines the specific test to be conducted. * Identify test ideas. * Defines test details. * Determine test results. * Document change requests. * Evaluate product quality. |
| Chew Poshi | Tester | * Implement tests and test suites. * Execute the test suites. * Log the results. * Analyse and recover from test failures. * Reports issues encountered while testing. * Definite severity and priority for each defect. |
| Chee Zi Hoe | QA Manager | * Monitor test status of all testing activities. * Selecting features to be tested. * Generate test report. * Review test cases. * Review and approve the issues. * Prepare the status report of testing activities. * Updating the project manager regularly about the progress of testing activities. |
| Hermes Lim HongJun | QA Engineer | * Reviewing quality specifications and technical design documents to provide timely and meaningful feedback. * Creating detailed, comprehensive and well-structured test plans and test cases. * Assigns tasks and tracks the testing process. |

# 

# 

# 12. Staffing And Training Needs

Testers would have to be trained on how to launch SportZ and how to enter the correct inputs according to the various test cases. Team members will be assigned to conduct different points of testing and results will be compiled by the Quality Assurance Manager.

It is preferred that there will be at least one full time tester assigned to the project for the system/integration and acceptance testing phases of the project, SportZ.

In order to provide complete and proper testing the following areas need to be addressed in terms of training.

1. The developers and tester(s) will need to be trained on the basic operation of the game interface.
2. At least one developer needs to be trained on the installation and control of the PC based game. The distributor personnel will also have to be trained on the PC based game and its operational characteristics.

# 

# 

# 

# 

# 13. Schedule

For each phase of development, the testing schedule is as follows:

|  |  |
| --- | --- |
| **Task** | **Duration** |
| Design test cases | 2 days |
| Execute test cases + documentation | 2-3 days |
| Bug fix and correction | 2-3 days |
| Repeat execution and bug fix (additional iterations if necessary) | 3-4 days |
| Final test report and documentation | 1-2days |

# 14. Risks And Contingencies

|  |  |  |
| --- | --- | --- |
| Risk | Impact | Contingencies |
| Lack of training for testers. | Low | * Assign testers with the required skills to take over the testing. At the same time, QA manager and Lead Developer would train the testers for future test activities. |
| Delay in testing due to incomplete build or unexpected bugs. | Medium | * Preparation for testing should be done in concurrent with the software development phases. Modularity in design also allows for separate components to be tested while other parts are still in the midst of development. |
| Prerequisite entry criteria is not met. | Low | * Meeting of outstanding prerequisites and conducting Load Test Failure can aid in mitigating this risk. |
| Test data proves to be inadequate | Medium | * Refinement of test data is required. Review of test plan and modification to testing components. |
| Testing schedule is state. If the start of testing is delayed due to design tasks, the test cannot be extended beyond User Acceptance Testing scheduled start date. | High | * Testing team can control the preparation tasks in advance and the early communication with involved parties. * Some buffers can be added to the schedule or contingencies, although not as much as best practices advice. |
| Defects are found at the late stage of the cycle or at a late cycle. | High | * Consideration of a defect management plan can be put in place to ensure that there is prompt communication and fixing of issues. |
| Delayed testing due to new issues. | Medium | * Defect management and issue management procedures can be put in place to immediately provide a resolution for defects that are encountered. |

# 

# Approvals

After testing is completed, the Quality Assurance team will write a report on the results. All findings and issues should be clearly reported. The Quality Assurance Manager (QAM) is in charge of reviewing the report and sending it for approval to the Project Manager. Once the Project Manager has accepted the documentation, he or she will notify the relevant stakeholders of the test results and seek approval to continue with the system's release.