Unity Project

Test Plan Document

Members: Anthony Jamora, Jonny Nabors, Thomas Burke

Draft 1

*CSC 4420*

*Unity Group*

*2015*

[1 Introduction 4](#_Toc414383519)

[1.1 Purpose 4](#_Toc414383520)

[1.2 Problem Description 4](#_Toc414383521)

[1.3 Project Scope 4](#_Toc414383522)

[1.4 Glossary 4](#_Toc414383523)

[2 Compatibility Testing 4](#_Toc414383524)

[2.1 Test Approach 4](#_Toc414383525)

[2.2 Items to be Tested 4](#_Toc414383526)

[2.3 Test Risks/Issues 4](#_Toc414383527)

[2.4 Test Pass/Fail Criteria 4](#_Toc414383528)

[2.5 Test Entry/Exit Criteria 4](#_Toc414383529)

[2.6 Test Deliverables 4](#_Toc414383530)

[2.7 Test Suspension/Resumption Criteria 4](#_Toc414383531)

[2.8 Test Environmental Needs 4](#_Toc414383532)

[3 Functional Testing 5](#_Toc414383533)

[3.1 Test Approach 5](#_Toc414383534)

[3.2 Items to be Tested 5](#_Toc414383535)

[3.3 Test Risks/Issues 5](#_Toc414383536)

[3.4 Test Pass/Fail Criteria 5](#_Toc414383537)

[3.5 Test Entry/Exit Criteria 5](#_Toc414383538)

[3.6 Test Deliverables 5](#_Toc414383539)

[3.7 Test Suspension/Resumption Criteria 5](#_Toc414383540)

[4 Integration Testing 5](#_Toc414383541)

[4.1 Test Approach 5](#_Toc414383542)

[4.2 Items to be Tested 5](#_Toc414383543)

[4.3 Test Risks/Issues 5](#_Toc414383544)

[4.4 Test Pass/Fail Criteria 5](#_Toc414383545)

[4.5 Test Entry/Exit Criteria 5](#_Toc414383546)

[4.6 Test Deliverables 5](#_Toc414383547)

[4.7 Test Suspension/Resumption Criteria 5](#_Toc414383548)

[5 Performance Testing 5](#_Toc414383549)

[5.1 Test Approach 5](#_Toc414383550)

[5.2 Items to be Tested 5](#_Toc414383551)

[5.3 Test Risks/Issues 5](#_Toc414383552)

[5.4 Test Pass/Fail Criteria 5](#_Toc414383553)

[5.5 Test Entry/Exit Criteria 5](#_Toc414383554)

[5.6 Test Deliverables 6](#_Toc414383555)

[5.7 Test Suspension/Resumption Criteria 6](#_Toc414383556)

[6 User Acceptance Testing 6](#_Toc414383557)

[6.1 Test Approach 6](#_Toc414383558)

[6.2 Items to be Tested 6](#_Toc414383559)

[6.3 Test Risks/Issues 6](#_Toc414383560)

[6.4 Test Pass/Fail Criteria 6](#_Toc414383561)

[6.5 Test Entry/Exit Criteria 6](#_Toc414383562)

[6.6 Test Deliverables 6](#_Toc414383563)

[6.7 Test Suspension/Resumption Criteria 6](#_Toc414383564)

[7 References 6](#_Toc414383565)

[8 Document Revision History 6](#_Toc414383566)

[9 Appendix 7](#_Toc414383567)

1. Introduction

The purpose of this document is to give an overview of both how the Unity3D testing suite allows automated and manual test cases to be performed as well as how they specifically apply to the Unity Capstone game.

* 1. Testing Overview

The Unity Game will utilize the Integration and Assertion Test Components for test script creation and automation. Integration tests are entire GameObjects that exist in a hierarchy where each TestComponent that is a part of the GameObject is tested sequentially. Assertion Components are a code-free way of creation conditions that are always expected to be true and testing for failure, throwing an exception when a failure occurs. These two testing methods will be integrated into the various test cases and scenarios outlined in this document.

* 1. Test Flow

A test starts once the Test Object is enabled. The test may finish its run in multiple ways1:

* Function Testing.Pass() is called. This will successfully finish the test
* Function Testing.Fail() is called. This will fail the test
* Execution times out. This can happen when none of the above functions is called within a specified period of time (you can set the timeout value per test).
* An unhandled exception is thrown.
* An expected exception is thrown (Expect exception must be checked)
* Every Assertion Component on objects under tests is checked at least once ( the "Succeed after all assertions are executed" option needs to be selected)
  1. Glossary

**Demo** - Playable demonstration of core gameplay mechanics

**Invariant** – Condition always expected to be true

**NPC –** Non-player character

**UI** – User interface

**HUD** – Heads-Up-Display

# Compatibility Testing

* 1. Test Approach
  2. Items to be Tested
  3. Test Risks/Issues
  4. Test Pass/Fail Criteria
  5. Test Entry/Exit Criteria
  6. Test Deliverables
  7. Test Suspension/Resumption Criteria
  8. Test Environmental Needs

1. Functional Testing
   1. Test Approach
   2. Items to be Tested
   3. Test Risks/Issues
   4. Test Pass/Fail Criteria
   5. Test Entry/Exit Criteria
   6. Test Deliverables
   7. Test Suspension/Resumption Criteria
2. Integration Testing
   1. Test Approach

Integration tests are designed be run in a separate scene than the main game. In the integration tests, a Test Object is a GameObject in the scene that has a TestComponent attached to it. Everything under the Test Object in the hierarchy is considered to belong to this test. Any object not under a Test object will be common for every test on the scene such as the ocean or the floor. Only one test will be active at a time1.

When the test is run, the following steps are executed:

1. Play mode is enabled
2. The first test becomes active
3. Wait until the test has finished or timeout has occurred
4. The current active test gets disabled
5. If there are more tests, enable the next test and continue from step 3
6. Report results and finish test run
   1. Test Runner Functionality

The Integration Tests will utilize Unity’s Integration Test Runner. This Test Runner will automate the execution of the testing process. The flow is as follows1:

1. Run all tests in the scene (excluding ignored tests)
2. Run selected test(s).
3. Create a new test - creates new test object on the scene
4. Options - options for working with Integration Tests
   1. Add GameObjects under selected test - when selected, when you add a new object to the scene it will be automatically placed under the test GameObject instead of the hierarchy root
   2. Block UI when running - when selected, a dialog will appear during test execution
5. Test Filter - will filter out tests where name does not contain the string
   1. Show succeeded - show tests that succeeded
   2. Show failed - show tests that failed
   3. Show ignored - show tests that are ignored
   4. Show not ran - show tests that haven’t been run
6. Test list - list of all tests available in the scene
7. Test log and exception messages
8. Test name - name of the test
9. Included platform - on what platform the test should included
10. Timeout - number of second after the test will timeout
11. Ignored - ignore the test when running all tests
12. Succeed after all assertions are executed - select if the test should finish after all assertions from Game Object in the test got checked at least once.
13. Expect exception - the test will not fail if an exception if thrown.
14. Expected exception list - a list of exception that will not fail the test when thrown. Separate the exceptions with comma (","). Derived types from types on the list will also be considered as expected. If the list is empty, any exception type will be accepted.
15. Succeed when exception is thrown - the test will succeed when one of the excepted exceptions is thrown.
    1. Items to be Tested
    2. Test Risks/Issues
    3. Test Pass/Fail Criteria
    4. Test Entry/Exit Criteria
    5. Test Deliverables
    6. Test Suspension/Resumption Criteria
16. Performance Testing
    1. Test Approach
    2. Items to be Tested
    3. Test Risks/Issues
    4. Test Pass/Fail Criteria
    5. Test Entry/Exit Criteria
    6. Test Deliverables
    7. Test Suspension/Resumption Criteria
17. User Acceptance Testing
    1. Test Approach
    2. Items to be Tested
    3. Test Risks/Issues
    4. Test Pass/Fail Criteria
    5. Test Entry/Exit Criteria
    6. Test Deliverables
    7. Test Suspension/Resumption Criteria
18. References

|  |  |  |  |
| --- | --- | --- | --- |
| Doc Number |  | Doc Version | Doc Name & Location |
| 1 |  | 1 | [Integration Tests in Unity](https://bitbucket.org/Unity-Technologies/unitytesttools/wiki/IntegrationTestsRunner) |
| 2 |  | 1 | [Assertion Component](https://bitbucket.org/Unity-Technologies/unitytesttools/wiki/AssertionComponent) |
|  |  |  |  |

1. Document Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Date | Author | Changes |
| 1.1 | 3/17/2015 | Jonathan Nabors | Initial Draft |
| 1.2 | 2/22/2015 | Jonathan Nabors | Release Plan, Input Module, Script Flowchart |

1. Appendix

Material including referenced documentation the web or elsewhere, as well as alternative designs or items/ideas for future improvements.