**VR Dashboard**

Testing Plan

Version *2.0*

*11/16/2015*

**VERSION HISTORY**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Version #** | **Implemented By** | **Revision Date** | **Approved By** | **Approval Date** | **Reason** |
| 1.0 | Justis Gill  Trevor Vasher  Joshua Marshall | 11/10/2015 |  |  | Initial Draft |
| 1.5 | Trevor Vasher  Joshua Marshall | 11/11/2015 |  |  | Reworked Many tests |
| 2.0 | Justis Gill | 11/16/2015 |  |  | Reworked Integration tests |

# Table of Contents

# **1. Introduction**

## 1.1 Purpose of the Test Plan Document

This test plan outlines the process of testing for the VR Data Visualization project. Its intended audience is the VR Data Visualization project team. Some portions of this document may on occasion be shared with the client/user and other stakeholder whose input/approval into the testing process is needed.

## 1.2 References

The test cases will be included in the VR-Test-Cases document which is attached.

# **2.Compatibility Testing**

## 2.1 Test Risks / Issues

We may be unable to access the necessary hardware to complete compatibility testing.

## 2.2 Items To Be Tested

## /

|  |  |  |
| --- | --- | --- |
| **ID** | **Item to Test** | **Test Description** |
| C2\_1 | Cross operating system compatibility | Test that the program will run on Windows 7, 8 and 10 |
| C2\_2 | Cross graphics card compatibility | Test that the program will run on both nVidia and AMD graphics cards that meet minimum requirements as outlined in SRS document. |

## 2.3 Test Approach

## The Data Visualization program will be downloaded onto the system for which compatibility will be tested.

## Both the Coaster and Map visualizations should run through a full visualization at least one time. At least one of the functions requiring Dual Shock input should be tested in each visualization. Headtracking should also be tested. There should not be a noticeable difference in response to input between systems

## We will test on any hardware that is available to use that meets the requirements in our SRS document.

## 2.4 Test Entry / Exit Criteria

## Verify that Oculus Rift and Dual Shock controller are properly configured.

## Verify that minimum system specifications are met by the machine.

## 

## 

## 

## 2.5 Test Deliverables

## Upon completion of testing a log of completed tests will be provided. A detailed test plan is already available and is attached.

## 2.6 Test Pass / Fail Criteria

A test will pass if there are no issues encountered during the test that either noticeably hinder the user experience or prevent a function of the program from performing. If this criteria is not met a test will be considered failed and the issue will need to be resolved before resumption of testing.

## 2.7 Test Suspension / Resumption Criteria

Testing will suspend upon any test failures. We will diagnose the failure, adjust, and then retest.

# **3. Functional Testing**

## 3.1 Test Risks / Issues

There is no significant risks involved with our Functional Testing.

## 3.2 Items To Be Tested

|  |  |  |
| --- | --- | --- |
| ID | Item to Test | Test Description |
| FT\_1 | Forward Traversal of Graph in Coaster Visualization | Start Coaster Visualization and play through to end. Make sure that the graph can be traversed smoothly and with no issues. |
| FT\_2 | Backwards Traversal of Graph in Coaster Visualization | With Coaster Visualization underway Press L2 on Dual Shock controller and ensure backwards traversal of graph takes place with no issues. |
| FT\_3 | Pause/Play in Coaster Visualization | With Coaster Visualization underway press pause button on Dual Shock controller. Ensure that the visualization plays and pauses with no issues. |
| FT\_4 | Switch Lines / Reset Coaster Visualization | With Coaster Visualization underway press the left and right directional buttons on the Dual Shock controller. Ensure that the visualization switches between lines and restarts with no issues. |
| FT\_5 | Pause/Play in Map Visualization | With Map Visualization underway press pause button on Dual Shock controller. Ensure that the visualization plays and pauses with no issues. |
| FT\_6 | UI Toggle in Map Visualization | With Map Visualization underway press the Triangle button on the Dual Shock controller. Ensure the UI panels toggle on and off with no issues. |
| FT\_7 | Coaster Headtracking | Ensure Oculus Rift headtracking responds to user input in Coaster Visualization. |
| FT\_8 | Coaster UI Graph Billboard | Ensure 2D line graph of data is correctly represented on billboard UI display in Coaster Visualization. |
| FT\_9 | Coaster UI Controls Overlay Toggle | Ensure Start button on Dual Shock controller can toggle the UI panel showing the control scheme on and off. |
| FT\_10 | Coaster UI Ring around point | Make sure that UI rings are appearing correctly relaying poll date and current polling percentage |
| FT\_11 | Select Coaster Candidate | Make sure that selection of candidate from the main menu for the coaster visualization performs as expected |
| FT\_12 | Map Headtracking | Ensure Oculus Rift headtracking responds to user input in Map Visualization. |
| FT\_13 | Map Rewind | Ensure that Map Visualization properly rewinds when the L 2button is held down on the Dual Shock controller. |
| FT\_14 | Map Forward | Ensure that Map Visualization properly rewinds when R2 is held down on the Dual Shock controller. |
| FT\_15 | Map Objects | Ensure that map objects are moving as expected and in sync with data is Map Visualization |
| FT\_16 | Map Investigation Mode | Ensure that when pressing X map investigation mode loads properly, and circle returns to normal function. |
| FT\_17 | Main Menu UI | Ensure that UI elements on the main menu work. |
| FT\_18 | Load CSV | Ensure that users can load a new CSV file from the main menu. |
|  |  |  |
|  |  |  |
|  |  |  |

## 3.3 Test Approach

## The overall approach to our functional testing is to test the functionality of the program as it would be used by an end user. By doing this we hope to identify and bugs or issues that may come up when using the program in a way that tests the functional requirements.

## 3.4 Test Pass / Fail Criteria

A test will pass if there are no issues encountered during the test that either noticeably hinder the user experience or prevent a function of the program from performing. If this criteria is not met a test will be considered failed and the issue will need to be resolved.

## 3.5 Test Entry / Exit Criteria

The criteria needed for test entry will be to:

* Ensure a system that meets the minimum requirements specified in the SRS document is available.
* Ensure Unity is installed on the machine
* Ensure Oculus Rift is hooked up to system and properly configured.
* Endure Dual Shock controller is hooked up to system and properly configured.

## 3.6 Test Deliverables

After testing is completed the following will be made available to our client:

* Test Cases – Already available.
* Log of all completed tests

## 3.7 Test Suspension / Resumption Criteria

Testing will suspend upon any test failures. We will diagnose the failure, adjust, and then retest.

# 4. Integration Testing

## 4.1 Test Risks / Issues

The different components of the program must work together for functionality. The UI, the visualization generators, and the data processing component are only useful if they all work together.

## 4.2 Items To Be Tested

|  |  |  |
| --- | --- | --- |
| **ID** | **Item to Test** | **Test Description** |
| IT\_1 | Map Data to Map Generator | The map generation requests data from the data storage and creates pins. Verify that pin generation is accurate. |
| IT\_2 | Rollercoaster Data to Rollercoaster Generator | The rollercoaster requests data from the data storage and creates the lines. Verify that line generation is accurate. |
| IT\_3 | Rollercoaster Data to Rollercoaster UI | The rollercoaster UI will display information from the rollercoaster data. Verify that the displayed data is accurate. |

## 4.3 Test Approaches

After individual tests are done on each piece of the program, we will test integration of these pieces. We will start by ensuring data is moved properly from the data to the visualization generators, then moving on to ensure data is displayed properly on the UI as is service information.

1. Data to Visualization generators
2. Data to UI

## 4.4 Test Pass / Fail Criteria

The integration testing will pass when each of the test cases succeeds. If any failures are encountered, testing will stop to correct the errors, restarting from the beginning.

## 4.5 Test Entry / Exit Criteria

The criteria needed for test entry will be to:

* All preceding tests have passed
* Verify if the testing environment is available
* Verify if testable code is available
* Verify if test data is available

If the test cases pass, then the testing process will move on to the next phase of testing.

## 4.6 Test Deliverables

The deliverables we will be using include:

* Test Cases - *See Test Case Document*
* Test Logs from running the test cases

## 4.7 Test Suspension / Resumption Criteria

If any failures are encountered, testing will stop to correct the errors, restarting from the beginning.

## 4.8 Test Environmental Needs

Integration testing can be performed in any supported environment.

# 5. Performance / Load Testing

## 5.1 Test Risks / Issues

We will be testing performance using our development machine. The specs of this machines will not be equal to the machines of our users.

## 5.2 Items To Be Tested

|  |  |  |
| --- | --- | --- |
| **ID** | **Item to Test** | **Test Description** |
| PT\_1 | 100000 CSV Row Load | We will load a CSV with 100000 rows of data looking for a visualization start time of 4 seconds or less |
| PT\_2 | Coaster FPS Test | Start Coaster visualization and ensure frames per second does not fall below 40 while providing random user input |
| PT\_3 | Map FPS Test | Start Map visualization and ensure frames per second does not fall below 40 while providing random user input |

## 5.3 Test Approaches

A generator of some sort will be used generate a 100000 row CSV file.

Additionally, a FPS monitor object will be added to our application for the sake of testing frames per second.

## 5.4 Test Pass / Fail Criteria

The test passes if all performance tests pass, and fails otherwise.

## 5.5 Test Entry / Exit Criteria

The criteria needed for test entry will be to:

* All previous test cases have been run and passed

Testing ends when the test case has passed.

## 5.6 Test Deliverables

The deliverables we will be using include:

* Test Cases - *See Test Case Document*
* Test Logs from running the test cases

## 5.7 Test Suspension / Resumption Criteria

Testing will suspend upon any test failures. We will diagnose the failure, adjust, and then retest.

## 5.8 Test Environmental Needs

Performance testing needs a machine that meets the minimum specs for the Oculus Rift and enough disk space to hold a 100000 row csv file.

# 6.System Testing

## 6.1 Test Risks / Issues

None

## 6.2 Items To Be Tested

|  |  |  |
| --- | --- | --- |
| **ID** | **Item to Test** | **Test Description** |
| ST\_1 | All Coaster Functionality | User can load a CSV, shoose the coaster visualization, select a candidate, look around, pause, play, rewind, and fast-forward visualization, switch lines, and toggle Controls overlay. |
| ST\_2 | All Map Functionality | User can load a CSV, choose the map visualization, look around, pause, play, rewind, and fast-forward visualization, select states for investigation, and toggle Controls overlay. |

## 6.3 Test Approaches

System testing begins after Functional and Integration testing. Since those have already been tested, we will simply test them in a full sequence to ensure one function does not break another.

## 6.4 Test Pass / Fail Criteria

During system testing all Test Cases must pass.

## 6.5 Test Entry / Exit Criteria

The criteria needed for test entry will be to:

* All System test cases have been run and passed.

If the Test case of this section passes, then the testing process can move on.

## 6.6 Test Deliverables

Test Cases - *See Test Case Document*

## 6.7 Test Suspension / Resumption Criteria

Testing will be halted if any system test case fails. A fix will be implemented and then all system tests must be restarted.

## 6.8 Test Environmental Needs

System testing requires a machine that meets the minimum specs for the Oculus Rift.

# 7. User Acceptance Testing

## 7.1 Test Risks / Issues

User Acceptance Testing is best performed with Oculus Rift and a high performance system.

## 7.2 Items To Be Tested

|  |  |  |
| --- | --- | --- |
| **ID** | **Item to Test** | **Test Description** |
| UAT\_1 | Coaster UAT | User is able to use rollercoaster visualization |
| UAT\_2 | Map UAT | User is able to use map visualization |

## 7.3 Test Approaches

Our testing approach will be based upon the requirements and design set forth in our Requirements Document and our Design Document.

## 7.4 Test Pass / Fail Criteria

The user acceptance testing will pass when each of the test cases succeeds. If any failures are encountered, testing will stop to correct the errors, restarting from the beginning.

## 7.5 Test Entry / Exit Criteria

The criteria needed for test entry will be to:

* All preceding tests have passed
* Verify if the testing environment is available
* Verify if testable code is available
* Verify if test data is available

If the test cases pass, then the testing process will be complete.

## 7.6 Test Deliverables

Test Cases - *See Test Case Document*

## 7.7 Test Suspension / Resumption Criteria

If any failures are encountered, testing will stop to correct the errors, restarting from the beginning.

## 7.8 Test Environmental Needs

User acceptance testing can be performed in any supported environment.

|  |
| --- |
| 8. Test Schedule and Responsibilities |

## 8.1 Purpose

Testing will follow the flow outlined by the test cases in their entry/exit criteria. Each step of testing will be performed according to its suspension/resumption criteria, and judgement will be based upon their pass/fail criteria. For test preparation, functional testing, and results preparation all members of the team will be responsible. The rest of the test responsibilities are broken down in the schedule below.

## 8.2 Schedule

|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Test** | **Date** | **Responsible Tester** |
| 1. | Prepare Test Environments | 12/7/2015 | All |
| 2. | Functional Testing | 12/8/2015 | All |
| 3. | Integration Testing | 12/9/2015 | Justis |
| 4. | Performance Testing | 12/9/2015 | Josh |
| 5. | Compatibility Testing | 12/9/2015 | Trevor |
| 6. | System Testing | 12/10/2015 | Josh |
| 7. | User Acceptance Testing | 12/10/2015 | Justis |
| 8. | Prepare Test Results | 12/11/2015 | All |