Durham University MeditateVR

Test Plan Report

*Group 1*

Adam Seidel (nnng27)

Ben Hughes (krcc82)

Enego Comley (wdsz78)

Pad Pang (gjzt58)

Nathaniel Mosey (wjld56)

Tom Owen (zjmq47)

Contents

[Section 1 – Introduction 2](#_Toc157439420)

[1.1 – Project Overview 2](#_Toc157439421)

[1.2 - Testing Overview 2](#_Toc157439422)

[Section 2 – Test Cases 2](#_Toc157439423)

[2.1 – Unit Testing (UNT) 3](#_Toc157439424)

[2.2 – Integration Testing (INT) 5](#_Toc157439425)

[2.3 – System Testing (SYS) 6](#_Toc157439426)

[2.4 - User Acceptance Testing (UAT) 8](#_Toc157439427)

[Section 3 - Testing context 8](#_Toc157439428)

[3.1 - Testing Environment 8](#_Toc157439429)

[3.2 - Testing Procedure 9](#_Toc157439430)

[3.3 - Testing Success 9](#_Toc157439431)

# Section 1 – Introduction

In this document we will outline a testing plan for our groups project. These tests will cover both functional and non-functional requirements. We will outline multiple tests for each testing type and in the second section of this document will further develop on a selection of these tests. The final section will include testing context and the severity of if any of these tests failed.

## – Project Overview

We are working on developing a VR meditation environment that will enhance the adaptivity and immersion of users during meditation sessions. The project is being developed based on the requirements of our client Prof Alexandra Cristea. This project is based around the belief that modern VR technology can increase the effectiveness of meditation sessions. The objective of the project is to create an immersive and abstract meditation environment for our client to use for research that utilizes visualization-based meditation techniques.

## 1.2 - Testing Overview

Tests will be split into 4 types for the purposes of this document: Unit (UNT), Integration (INT), System (SYS) and User Acceptance (UAT) testing. These tests will also be categorized based on if they are for functional or non-functional requirements.

# Section 2 – Test Cases

In this section below are detailed the planned test cases in each of the 4 testing categories. These are presented in their relevant sections and will be shown in the format of:

|  |  |
| --- | --- |
| Test Case ID |  |
| Description of Test |  |
| Related Requirement/Design Spec Details |  |
| Pre-requisites (If Relevant) |  |
| Test Procedure |  |
| Test Material Used |  |
| Expected Result (Test Oracle) |  |
| Comments |  |
| Created By |  |
| Test Environment(s) |  |

## 2.1 – Unit Testing (UNT)

Unit Testing selects individual components of the project and tests the functionality of these individual units.

Below are a selection of unit tests, the highlighted tests will be expanded upon in section 2. Selected tests are to provide an overview of areas that require unit testing.

|  |  |
| --- | --- |
| Test Case ID | unt\_test-01 |
| Description of test | The biometrics system outputs a list of floats from 0 to 100 that show the user’s stress level over time. |
| Related requirement spec/design spec details | FR 1.1, FR 3.4 |
| Pre-requisites for test |  |
| Test procedure | 1. Start all programs, those used for biometrics and the game. 2. Wear the biometric device. 3. Think about something stressful. 4. Meditate to relax. |
| Test material used | Biometrics device. PC |
| Expected result (test oracle) | The variable userStress should be a list of floating-point numbers ranging from 0 to 100. When the user was thinking about something stressful, these should be lower. |
| Comments | None |
| Created by | EC |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | unt\_test-02 |
| Description of test | User telemetry is stored to in a file. |
| Related requirement spec/design spec details | FR 3.5 |
| Pre-requisites for test | All software that generates user telemetry (The VR game and biometrics system) need to be creating this data and storing it in variables. |
| Test procedure | 1. Start all software. 2. Complete a meditation session. 3. Open and inspect the log file |
| Test material used | Biometrics device, VR headset, PC |
| Expected result (test oracle) | There should be a new file in the logs folder called MeditationDATETIME where DATETIME will be replaced with the time and date of the meditation. The contents of this file should be in the JSON format. It should contain a dictionary whose keys are forms of telemetry, such a heart rate or the direction they are facing. Each of these will then be paired with an array of each telemetry item over time. |
| Comments | None |
| Created by | EC |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | unt\_test-03 |
| Description of test | The system must load a text file containing meditation configuration information. This data must then be parsed into an instance of the config class in the unity game. |
| Related requirement spec/design spec details | FR 3.2 |
| Pre-requisites for test | None |
| Test procedure | 1. Edit the configuration in the default config file. 2. Start the unity game. 3. Inspect the public variables in the newly created config class to see that the changes in 1 were recorded. |
| Test material used | PC |
| Expected result (test oracle) | The values of the variables in the config class should match those specified in the .txt file. |
| Comments | None |
| Created by | EC |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | unt\_test-04 |
| Description of test | During the meditation game, music and atmospheric sound should be playing. |
| Related requirement spec/design spec details | FR 2.4 |
| Pre-requisites for test | The config system must be in place so that the game knows to play music and atmospheric sound. |
| Test procedure | 1. Make sure the pc system’s sound is set to come out of the speakers. 2. Ensure that music and atmospheric sound are enabled in the config. 3. Start the unity game. 4. Listen for music and atmospheric sound. |
| Test material used | The unity game. |
| Expected result (test oracle) | There should be both music and atmospheric sound such as birds chirping. |
| Comments | None |
| Created by | EC |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | unt\_test-050 |
| Description of test | The startup script should start all software. |
| Related requirement spec/design spec details | FR 1.2 |
| Pre-requisites for test | All the individual programs should be able to start up and run without crashing. |
| Test procedure | 1. Run the startup script. 2. Check that all programs are running. |
| Test material used | Unity game, biometrics system |
| Expected result (test oracle) | Both the programs for the unity game and the biometrics system should have started. |
| Comments | None. |
| Created by | EC |
| Test environment(s) | Windows 10 |

## 2.2 – Integration Testing (INT)

After performing each Unit test we will then move onto Integration testing to ensure that all the atomized parts of the system work together and they all communicate correctly.

|  |  |
| --- | --- |
| Test Case ID | int\_test-01 |
| Description of test | The Biometrics tracking system successfully integrates with the plant growth system |
| Related requirement spec/design spec details |  |
| Pre-requisites for test | The Biometrics system outputs data which can be routed into the main unity program and the plant growth system is able to take parameters to adjust its growth rate |
| Test procedure | 1. Run biometrics tracking system, with a live feed of the data it produces 2. Run the Unity game system and load into the game. 3. Have the user do things that would cause the flower to be small i.e. relax mentally or lower heart rate and see if the plant stays small 4. Have the user do stressful or physically intensive things to see if the plant growth increases |
| Test material used | VR Headset. Biometrics System. PC |
| Expected result (test oracle) | The plant growth rate is reflective of the data coming from the biometrics system. |
| Comments | None |
| Created by | Ben Hughes |
| Test environment(s) | Windows 10, Unity, Ocuclus VR |

|  |  |
| --- | --- |
| Test Case ID | int\_test-02 |
| Description of test | The Researcher config data goes throughout the system to the different parts that it affects |
| Related requirement spec/design spec details |  |
| Pre-requisites for test | The File loading and variable parsing system works and all configurable aspects take variable parameters. |
| Test procedure | 1. Configure variables to have certain values and then run the program. 2. Open the Unity Variables view 3. Ensure the configurable parts of the system have changed in line with the configuration. 4. Change the variables to other values and repeat step 2 |
| Test material used | PC |
| Expected result (test oracle) | The parts of the system that can be configured are shown |
| Comments | None |
| Created by | Ben Hughes |
| Test environment(s) | Windows 10, Unity, Ocuclus VR |

## 2.3 – System Testing (SYS)

The system testing plan has been devised to ensure that our system conforms to the requirements specification that we have previously outlined.

|  |  |
| --- | --- |
| Test Case ID | sys\_test-01 |
| Description of test | The system is started by a single script |
| Related requirement spec/design spec details | FR 1.2 |
| Pre-requisites for test | System has not been started. |
| Test procedure | 1. Launch the system using the start script |
| Test material used | Start script |
| Expected result (test oracle) | System is operational and the user can begin a meditation session |
| Comments | None |
| Created by | Adam Seidel |
| Test environment(s) | Windows 10, Unity, Oculus VR |

|  |  |
| --- | --- |
| Test Case ID | sys\_test-02 |
| Description of test | The system should have a high enough framerate to not cause nausea. |
| Related requirement spec/design spec details | NFR 1.1 |
| Pre-requisites for test | The system is operational with the Oculus VR headset connected |
| Test procedure | 1. Start the system 2. Enter a meditation session 3. Complete meditation session |
| Test material used | Oculus VR |
| Expected result (test oracle) | The framerate of the system does not fall below 60 for more than 5% of the meditation session. |
| Comments | None |
| Created by | Adam Seidel |
| Test environment(s) | Windows 10, Unity, Oculus VR |

|  |  |
| --- | --- |
| Test Case ID | sys\_test-03 |
| Description of test | Data stored by the system must not be personally identifiable |
| Related requirement spec/design spec details | NFR 4.1 |
| Pre-requisites for test | System is operational. |
| Test procedure | 1. Carry out and complete a meditation session. 2. Assess the JSON log file for the previous meditation session. No information that could identify the user can be stored in the file. |
| Test material used | JSON session file |
| Expected result (test oracle) | JSON session file does not contain any personal information about the user |
| Comments | None |
| Created by | Adam Seidel |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | sys\_test-04 |
| Description of test | Certain parameters can be set by the researcher via a config file. |
| Related requirement spec/design spec details | FR 3.2 |
| Pre-requisites for test | System is not operational. |
| Test procedure | 1. Adjust parameters in the config file. 2. Launch the system. 3. Begin a meditation session. 4. Verify that the parameters changed are being reflected in the meditation session. |
| Test material used | Config file |
| Expected result (test oracle) | The change of session parameters are reflected in the meditation session |
| Comments | None |
| Created by | Adam Seidel |
| Test environment(s) | Windows 10 |

|  |  |
| --- | --- |
| Test Case ID | sys\_test-05 |
| Description of test | The plant growth rate must be affected by the user’s meditation quality. |
| Related requirement spec/design spec details | FR 1.3 |
| Pre-requisites for test | The system is operational, and the user is participating in a mediation session. |
| Test procedure | 1. Follow the on-screen guidance to carry out a period of high-quality meditation. 2. Ignore the on-screen guidance and purposely carry out a period of low-quality meditation. |
| Test material used | Oculus VR |
| Expected result (test oracle) | The growth rate of the plant should be greater during the first meaningful period of meditation than during the second low quality period of meditation. |
| Comments | None |
| Created by | Adam Seidel |
| Test environment(s) | Windows 10 |

## 2.4 - User Acceptance Testing (UAT)

AUT overview…

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# Section 3 - Testing context

This section includes details of the testing process and what is required for them to be carried out. It will further expand upon testing prerequisites and testing procedure.

## - Testing Environment

All tests require the project running through Unity we will be utilizing Unity version 2022.3.14f1 running on a Windows 10 operating system. The device will be connected to an Oculus Quest 2 as well as an Emotiv BCI EPOC X 14 channel wireless EEG headset. This will be setup prior to any testing is carried out.

## - Testing Procedure

Unit, System and Integration tests will be carried out by group members with some observing whilst another utilizes the equipment to provide results. User Acceptance Testing will be carried out with willing volunteers taking and documenting their feedback on the relative test focuses.

All tests will be undertaken manually with the startup script being ran to initialize the BCI application and Unity game. Those tests which require editing of the configuration file (namely: unt-test-03; unt-test-04; and int-test-02) will need further user input before commencing the test.

## - Testing Success

Success of Unit, System and Integration tests will be dictated by the group members observing if they sufficiently satisfy the expected outcomes.

User Acceptance Tests will be considered a success if the majority of those volunteering give positive responses to the criteria specified.

Unit testing

Does biometrics system output biometrics data as floating point numbers

Does the system save data to a file

The system loads the researcher config and parses the data to get parameters

The plant growth system takes data from the biometrics system.

Does the system load the music and play music and sounds

Does the script startup all the software.

System Testing

NFR 1.1 System Does not Cause Nausea

NFR 2.1 Quick to set up

NFR 4.1 No Personal Information

FR 2.2 Low poly meditation environment

FR 2.6 Detailed User movement

FR 2.1 Visual Meditation Guidance

UAT

Plant growth system is relaxing

NFR 3.1 A Cohesive Visual Style

NFR 3.3 Calming Audio

NFR 2.2 Researcher config file should be easy to use

The user feels like the plant reflects how relaxed they are

NFR 3.2 Audio fits the setting

Integration and Intro – Ben

Unit Testing – Enego

System Testing – Adam

UAT – Tom

2 People for Testing context – Pad and Nathanial