The Musical Instruface Project

Integration Test Plan

v1.0

Vijay Balchandani
Boris Block
Dan Hennessey
Zenko Klapko
David Millar
William Morgan
Table of Contents

1. Document History ........................................................................................................... 3
   1.1. Document Information ............................................................................................. 3
   1.2. Revision History ......................................................................................................... 3
2. Overview ......................................................................................................................... 4
   2.1. Scope of this document ............................................................................................ 4
3. Roles and Responsibilities .............................................................................................. 5
   3.1. Introduction .............................................................................................................. 5
   3.2. Team Roles ............................................................................................................... 5
   3.3. Reporting Procedures ............................................................................................... 5
4. Testing Approach ............................................................................................................ 6
   4.1. Introduction .............................................................................................................. 6
   4.2. Testing Objectives .................................................................................................... 6
   4.3. Testing Structure ...................................................................................................... 6
5. Entry and Exit Criteria .................................................................................................... 7
   5.1. Entry Conditions ...................................................................................................... 7
   5.2. Exit Conditions ......................................................................................................... 7
   5.3. Exception Conditions ............................................................................................... 7
6. Test Plan .......................................................................................................................... 8
   6.1. Features to be Tested ............................................................................................... 8
   6.2. Test Deliverables ...................................................................................................... 8
7. Test Cases ........................................................................................................................ 9
1. Document History

1.1. Document Information

<table>
<thead>
<tr>
<th>Title:</th>
<th>The Musical Instruface Project: Integration Test Plan</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s):</td>
<td>Dan Hennessey</td>
</tr>
<tr>
<td>Reviewer(s):</td>
<td>Dan Hennessey, David Millar</td>
</tr>
</tbody>
</table>

1.2. Revision History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Editor</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>v1.0</td>
<td>2/27/2007</td>
<td>DH</td>
<td>Initial draft</td>
</tr>
<tr>
<td>v1.1</td>
<td>2/28/2007</td>
<td>DM</td>
<td>Minor additions and rephrasing of confusing sections</td>
</tr>
</tbody>
</table>
2. Overview

The Musical Instruface project seeks to create a musical instrument interface for use with a simultaneous multi-touch display. This interface will allow the user to arrange musical instrument parts called “instrufaces” to make their own musical performance space.

Integration Testing (IT) will be used to test the objects defined for this project and how they communicate to form a working version of our Musical Instruface Application. In order to accomplish this goal we have defined the ways in which various objects will need to communicate and designed test scenarios to ensure this data connectivity is present and functioning as designed.

2.1. Scope of this document

This document was designed to cover the following activities:
   a) Testing that objects are accepting necessary input values.
   b) Testing that the objects are also outputting correct and complete data as necessary and expected.
   c) Validating that inter-object and interprocess communication is occurring as specified and designed for.
   d) Developing the test scripts and scenarios to properly test these goals.

This document was NOT designed to cover any of the following areas or activities:
   a) Testing of functionality of the application as a whole in relation to the Software Requirements Specification.
   b) The requirements for any other form of testing beyond that of the Integration Testing. This includes user acceptance testing, unit testing, stress testing, regression testing, and system testing.
3. Roles and Responsibilities

3.1. Introduction
This section defines the various roles that each team member will play in the acceptance testing portion of the project. Also outlined will be the proper procedure for reporting the results of a testing session.

3.2. Team Roles

<table>
<thead>
<tr>
<th>Testing Team Leader:</th>
<th>Dan Hennessey</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sub System Leads:</td>
<td></td>
</tr>
<tr>
<td>Sound</td>
<td>William Morgan, Vijay Balchandani</td>
</tr>
<tr>
<td>Graphics</td>
<td>Dan Hennessey</td>
</tr>
<tr>
<td>MTD Interface</td>
<td>Boris Block, Zenko Klapko</td>
</tr>
<tr>
<td>Data Save/Load</td>
<td>David Millar</td>
</tr>
</tbody>
</table>

3.3. Reporting Procedures
All testing sessions should be fully documented. The testing participant running the session should make note of all test scripts/scenarios being run during that session. As the tests are run, the testing participant should make note whether each test passed or failed. If a test failed take detailed notes on which test failed, what parameters it failed on, and if any errors were thrown by the program, which errors they were.

These testing reports should be submitted to the Team Test Leader after each testing session for review. Any problems noted will be tested again by the Testing Team Leader to verify it is reproducible. Once the problem has been verified the problem will be reported to Project Leader for discussion in the regular status meetings.
4. Testing Approach

4.1. Introduction
This section will detail how we plan to approach the Integration Testing. This includes the objectives we hope to achieve through our integration testing as well as the structure that our testing shall adhere to.

4.2. Testing Objectives
For the Integration Testing of this application we primarily want to verify that all classes are properly accepting and producing data as necessary, maintaining proper state and behaving as intended. We are also aiming to ensure that there is proper connectivity between classes and processes that require it. This includes creation and deletion of new class objects and that connected data is properly handled, inputting and outputting data between objects, proper communication and state between threads, and manipulating data between objects.

4.3. Testing Structure
Our Integration Testing shall be carried out in an incremental depth-first style with our primary method to develop and test critical sections first. This critical section approach will be combined with a bottom-up approach to ensure that we get critical application sections up and running in full and then tied together in the necessary manner. We first want to build and test our most basic systems, such as sound generation and object interaction and expand from there.

With the critical sections defined and working we can more easily test and develop other sub-systems such as the graphical user interface and the session saving and loading mechanism. Also, by focusing on critical sections we can ensure that we can at least produce the most basic requirements of our application in the designated time frame of the project.
5. Entry and Exit Criteria

In order to both begin and end Integration Testing there are certain criteria that must be met. This section will outline those criteria necessary for both entry and exit from the integration testing phase. Also outlined will be any exceptions that may interrupt the Integration Testing activities.

5.1. Entry Conditions

The Integration Testing phase can be entered when the following entry criteria are met:

- All relevant code modules have been properly designed.
- At least one code module has started development and reached some level of functionality.
- The module to be tested has undergone Unit Testing.
- Any necessary “drivers” have been created and put into place to facilitate the necessary testing.
- Both Project Lead and Testing Team Lead have agreed to proceed with Integration Testing.

5.2. Exit Conditions

The Integration Testing phase can be entered when the following exit criteria are met:

- All code modules have finished development.
- All code modules have undergone Unit Testing.
- All specified test scripts and scenarios have been executed.
- There are reported errors or data inconsistencies appearing in Integration Testing reports.

5.3. Exception Conditions

The Integration Testing activities may be interrupted if any of the following conditions arise:

- The design of any individual code modules have been found inadequate or incorrect.
- If code modules fail Unit Testing after development updates.
6. Test Plan

6.1. Features to be Tested
The diagram displayed below details the approach to developing and testing our code modules for the Musical Instruface Application. The arrows represent the order of integration. Modules that have no dependencies are the primary critical sections. The arrows also indicate the level of priority assigned to various modules.

6.2. Test Deliverables
In order for an integration test to take place the following must be delivered:
- The unit testing test reports for modules to be tested, which should be free of major issues and errors.
- The two code modules to be tested.
- Any necessary drivers to properly perform the integration testing.
- Any necessary input data for the tests to be performed.

Upon completing a particular integration test the following items should be delivered to either the Test Team Leader or the Project Leader.
- The Integration test report.
- A read out of any output data provided from the integration test.
- Recommendations for further testing or error corrections.
7. Test Cases

This section remains undefined as of now. The reason for this is that we did not have a fully flushed out design document on which to base the creation of these test cases. This section will be updated with appropriate test cases once a final version of the design document has been agreed upon.