The Musical Instrument Project

Software Requirements Specification

v1.4

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. Introduction ................................................................................................................................. 4
   2.1. Purpose of this document ..................................................................................................... 4
   2.2. Overview .............................................................................................................................. 4
3. Functional Requirements ............................................................................................................ 6
   3.1. Main Interface ...................................................................................................................... 6
       3.1.1. General .......................................................................................................................... 6
       3.1.2. Toolbar .......................................................................................................................... 6
       3.1.3. Tabs ............................................................................................................................... 7
   3.2. Menu ..................................................................................................................................... 8
       3.2.1. General .......................................................................................................................... 8
       3.2.2. New Tab ........................................................................................................................ 8
       3.2.3. New Session .................................................................................................................. 9
       3.2.4. Open Session ............................................................................................................... 9
       3.2.5. Save Session ................................................................................................................. 9
       3.2.6. Save Session As ............................................................................................................ 9
       3.2.7. Most Recently Used ...................................................................................................... 9
       3.2.8. Exit ............................................................................................................................... 9
   3.3. Modes ................................................................................................................................... 9
       3.3.1. General: ......................................................................................................................... 9
       3.3.2. Play Mode ..................................................................................................................... 10
       3.3.3. Edit Mode .................................................................................................................... 10
   3.4. Instrufaces .......................................................................................................................... 11
       3.4.1. General ........................................................................................................................ 11
       3.4.2. Property Pages ............................................................................................................. 11
       3.4.3. PolyTheremin .............................................................................................................. 13
       3.4.4. Piano ............................................................................................................................ 13
       3.4.5. Gridtar .......................................................................................................................... 14
       3.4.6. Hexatonnetz ............................................................................................................... 15
       3.4.7. Sample Pad .................................................................................................................. 16
       3.4.8. Volume ........................................................................................................................ 16
4. Non Functional Requirements ................................................................................................... 17
   4.1. User input hardware ........................................................................................................... 17
   4.2. Performance ....................................................................................................................... 17
   4.3. Evolution ............................................................................................................................ 17
5. Appendices ................................................................................................................................ 18
   5.1. Glossary .............................................................................................................................. 18
   5.2. References .......................................................................................................................... 18
1. Document History

1.1. Document Information

<table>
<thead>
<tr>
<th>Title:</th>
<th>The Musical Instruface Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s):</td>
<td>Vijay Balchandani, Boris Block, Dan Hennessey, Zenko Klapko, William Morgan</td>
</tr>
<tr>
<td>Reviewer(s):</td>
<td>Dan Hennessey, Zenko Klapko, David Millar, William Morgan</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>11/2/2006</td>
<td>WM, VB, BB</td>
<td>Included Multi-touch API.</td>
</tr>
<tr>
<td>v1.1</td>
<td>12/5/2006</td>
<td>WM, VB, BB</td>
<td>Redone from scratch. Scope limited to Musical Instruface project.</td>
</tr>
<tr>
<td>v1.2</td>
<td>1/20/2007</td>
<td>DH</td>
<td>Addition of Table of Contents as well as additional formatting.</td>
</tr>
<tr>
<td>v1.3</td>
<td>2/6/2007</td>
<td>DH</td>
<td>Correction of some grammatical errors as well as other minor edits</td>
</tr>
</tbody>
</table>
2. Introduction

2.1. Purpose of this document

This document is intended to specify in exact detail the Musical Instruface Project. It specifies the visible and audible behaviors of the software, and will not discuss the specifics of implementation. It does this formally by establishing Functional and Nonfunctional requirements of the software. Supplementary information, including a video mock-up demonstration of the interface, is available in the appendices at the end.

This document is considered live and will be updated throughout the project.

2.2. Overview

The Musical Instruface Project is a musical instrument interface in software. It has dynamic visual feedback that traditional instruments are incapable of, and it’s design is focused on the musician and not the physical method of sound production.

In traditional instruments, the method of sound production and the instrument user interface are closely tied. With a guitar, the user must actually pluck the strings that produce vibrations in the air. With a trumpet, the user is part of the mechanism that produces sound, forcing air through the tubes to produce pleasing vibrations. Even with a traditional piano or harpsichord the interface is based around the physical mechanism of sound production. In these systems, the user even supplies the actual energy to operate the device. Why require the user to operate strings, valves, hammers, bags, bellows, etc? Why not design the interface around the musician?

When thinking of design, the modern computer scientist makes a distinction between front-end user interface and back-end implementation. Such a distinction was likely not influential in the design of traditional musical instruments: The actual sound produced was surely most important. Because of modern advancements like the MIDI standard, VST instrument plugins, ReWire, etc. we can design a musical instrument interface alone without having to worry much about the actual method of sound production. This will allow us to focus on the user, rather then forcing the user to focus on the physical eccentricities of the sound production mechanism.

Beyond User Centered Design, there are also new capabilities that software musical interfaces present to musicians. Software musical interfaces can be dynamic: changing depending on the context of the song, scale, key, last note played, tempo and more. They can also provide visual feedback to the musician, including the location of octave notes, chord intervals, and much more.

Musical interfaces have been explored somewhat using typing keyboards and computer mice, though most are not intended for real-time, serious play. Rather, they are used mostly as a makeshift replacement for a MIDI controller in composition software where
real-time interaction is not an issue. Adapting this type of interface for real-time music play is clearly feasible.

But we are more ambitious then the clear-cut case. There are more motivating interaction technologies that are also quite feasible. It would be very easy to extend musical interfaces to single point touch screens, pen-tablets, or a multipoint touch screen such as the one presented by Jeff Han.
3. Functional Requirements

![Image of software interface]

Figure1: The software in its initial state.

3.1. Main Interface

3.1.1. General

3.1.1.1. The main interface is the first screen displayed when the application is started.
3.1.1.2. A blank session is created as defined in section 2.2.1.
3.1.1.3. The main interface runs in full screen mode at all times.
3.1.1.4. The main interface consists of the following two sections:

3.1.2. Toolbar

3.1.2.1. General

3.1.2.1.1. The toolbar takes up 5% of the main interface height and is positioned at the very top of the main interface.
3.1.2.1.2. The toolbar takes up the full width of the main interface.
3.1.2.1.3. Each element in the toolbar is surrounded by a uniformly sized gap that is 10% of the width of the smallest toolbar element.

3.1.2.1.4. The toolbar contains the following three elements.

3.1.2.2. Menu Button
3.1.2.2.1. The width of the menu button is 5% of the width of the toolbar.
3.1.2.2.2. The menu button is the first element from the left side of the toolbar.
3.1.2.2.3. When clicked, the menu defined in section 2.2 is displayed with its top left corner in the same position as the bottom left corner of the menu button.

3.1.2.3. Tab Header Space
3.1.2.3.1. The tab header space takes up all of the available space in the toolbar not taken up by any other elements or gaps in the toolbar.
3.1.2.3.2. Tab headers as defined in section 2.1.2.1 are contained within this space and separated by gaps smaller than those separating toolbar elements.
3.1.2.3.3. All tab headers as defined in section 2.1.2.1 have the same width as each other at all times.
3.1.2.3.4. If there is not enough space in the header space for a new tab header as defined in section 2.1.2.1, the width of all existing tab headers will shrink uniformly so that the new tab can fit in the tab header space.
3.1.2.3.5. Once the tab headers in the tab header space have reached their minimum width as defined in section 2.1.2.1, no new tabs can be added and the new tab menu defined in section 2.2.1 will be disabled.
3.1.2.3.6. Clicking on a tab header in the tab header space displays that tab in the tab section defined in 2.1.2.

3.1.2.4. Mode button
3.1.2.4.1. The width of the mode button is 5% of the width of the toolbar.
3.1.2.4.2. The mode button is the first element from the right side of the toolbar.
3.1.2.4.3. When clicked, the button toggles between the modes defined in section 2.3.
3.1.2.4.4. The mode button’s graphic updates according to the current mode as defined in section 2.3. Note that the graphic shows which mode the user will transition to upon tapping it, not the current mode.

3.1.3. Tabs
3.1.3.1. General
3.1.3.1.1. Each tab contains one layout in its tab body.
3.1.3.1.2. Each tab consists of the following elements.

3.1.3.2. Tab Header
3.1.3.2.1. The tab header is located in the tab header space as defined in 2.1.1.2.
3.1.3.2.2. The tab header has title text of no more than 32 characters.
3.1.3.2.3. The tab header width is fixed initially to 1/8th the width of the tab header space defined in section 2.1.1.2.
3.1.3.2.4. The tab header’s font size is fixed to 12 point Arial.
3.1.3.2.5. The tab header’s minimum width is reached when only one character of the title is seen.
3.1.3.2.6. As the tab header shrinks, if there is not enough room for the full title, as much of the title as possible is displayed along with an ellipsis indicating that the title continues beyond the border of the tab header.

3.1.3.3. Tab Body
3.1.3.3.1. The tab body is where all of the instrufaces defined in section 2.4 are placed.
3.1.3.3.2. Any area of the tab body not occupied by an instruface is considered empty space.

3.2. Menu

3.2.1. General
3.2.1.1. The menu will be hidden once an option is chosen or the user clicks outside of the menu.
3.2.1.2. The menu contains the following elements

3.2.2. New Tab
3.2.2.1. Creates a new tab as defined in section 2.1.2.
3.2.2.2. The new tab body has no interfaces and consists entirely of empty space.
3.2.2.3. The new tab header is placed in the rightmost position on the tab header space as defined in section 2.1.1.2.
3.2.2.4. The tab header text has the focus and is editable using the keyboard as input.

3.2.3. **New Session**
3.2.3.1. If any changes have been made to the session, asks the user if they would like to save using the procedures in section 2.2.4
3.2.3.2. Creates a new session.
3.2.3.3. The session has one tab that is created with the same requirements as defined in section 2.2.1.

3.2.4. **Open Session**
3.2.4.1. If any changes have been made to the session, asks the user if they would like to save using the procedures in section 2.2.4
3.2.4.2. Uses the O/S’s standard open file dialogue box.
3.2.4.3. Opens the session and displays the same tab that was being used at the point that the session was saved as defined in section 2.2.4.

3.2.5. **Save Session**
3.2.5.1. If the session has never been saved before, follows procedures defined in section 2.2.5.
3.2.5.2. Saves all session information including the currently active tab.

3.2.6. **Save Session As**
3.2.6.1.Brings up the O/S’s standard save file dialogue box.
3.2.6.2. Saves all session information including the currently active tab.

3.2.7. **Most Recently Used**
3.2.7.1. Contains a list of the most recently used sessions ordered such that the most recently used session is at the top of the list.
3.2.7.2. Show the last four used sessions.

3.2.8. **Exit**
3.2.8.1. If any changes have been made to the session, asks the user if they would like to save using the procedures in section 2.2.5.
3.2.8.2. Closes the application.

3.3. **Modes**

3.3.1. **General:**
3.3.1.1. The user’s interaction with the music application depends upon the current mode.
3.3.1.2. There are two modes: play and edit.
3.3.2. **Play Mode**

3.3.2.1. The play mode allows the user to produce or alter sound through instrufaces as defined in section 2.4.

3.3.2.2. Other instrufaces not on the current tab will continue to operate and produce sound but are not visible.

3.3.2.3. Play mode disables the menu button as defined in section 2.1.2.2.

3.3.3. **Edit Mode**

3.3.3.1. **General**

3.3.3.1.1. Play mode enables the menu button as defined in section 2.1.2.2.

3.3.3.1.2. All instrufaces become silent when a switch to edit mode is made.

3.3.3.1.3. Tapping once on the selected tab header <link> highlights the current text and provides the ability to change the text description.

3.3.3.1.4. Tapping once on empty space in the tab body creates a new instruface and opens its configuration dialog box to the Type property page described in section 2.4.

3.3.3.1.5. Tapping once on an existing instruface selects it.

3.3.3.1.6. Placing two fingers on a selected instruface and dragging will translate, uniformly scale, and rotate it. See the video by Han, et al. for demonstration of this gesture.

3.3.3.1.7. Tapping on a selected instruface brings up its configuration dialog box as described in the instruface section. This is a modal dialog box.

3.3.3.1.8. An instruface is deleted by dragging more then 50% of it off screen, at which point it becomes highlighted in red. This permanently removes the instruface and its configuration from the music application. It does not delete any pre-existing outside resources such as associated wav files. Deleting an instruface will remove all links to other instrufaces.
3.4. Instrufaces

3.4.1. General

3.4.1.1. Many instrufaces have keys. A key is a geometric region with an associated 12-tone equal temperament pitch. When a key is tapped during play mode, the sound specified on the Sound property page is played at the associated pitch. There are many exceptions to this rule: please consider the interface’s volume, transpose link, and mimic link, all of which can cause a key not to directly output a sound when hit.

3.4.1.2. Instrufaces are configured in edit mode through an instruface configuration dialog box. This dialog box is organized into property pages described below:

3.4.2. Property Pages

3.4.2.1. Type

3.4.2.1.1. All instrufaces have a type. These types are specified in the sections following the General Instruface section 2.4.1.

3.4.2.2. Sound
3.4.2.2.1. Pitches can be synthesized as sine, square, or saw tooth waveforms.
3.4.2.2.2. Pitches can be also synthesized using a MIDI device with a specific program and channel number.
3.4.2.2.3. Pitches can also be synthesized as user loaded waveforms from files. Such waveforms are simply resampled according to pitch. The middle key, as defined on the instruface Layout property page, corresponds to the waveform’s original sample rate.

3.4.2.3. View
3.4.2.3.1. Checking the Brightness box assigns the brightness of keys (using the HSB or HSV standard color model) by absolute pitch number. The lowest pitch will be black; the highest pitch will be white.
3.4.2.3.2. Checking the Hue box assigns the hue of keys according to their pitch class. Because we offer only 12 tone equal temperament, this means that an increment of one semitone corresponds to an increment of thirty degrees in hue, according to the HSV/HSB color model.
3.4.2.3.3. The Piano checkbox assigns a brightness value of pure black or white to keys, as on a traditional piano keyboard. Checking this box unchecks the Brightness and Hue checkboxes.
3.4.2.3.4. The Enable Labels checkbox enables the display of individual textual labels on each key of a keyed instruface.
3.4.2.3.5. The Numbers checkbox, under the label section, makes key labels display a number, for the pitch class, not a letter. A pitch class of C is assigned 0, C# is assigned 1, and so on.
3.4.2.3.6. The Absolute checkbox, under the label section, makes textual labels absolute instead of only by pitch class. If Numbers is checked, then the lowest semitone is labeled 0, and the numbers increase from there. Otherwise, the label consists of the pitch class label followed by a space, and then a number indicating the octave of that note (e.g. C 0, C 1, C 2, C# 0, C# 1, etc…)

3.4.2.4. Link
3.4.2.4.1. An instruface may be linked to a set of other instrufaces from any tabs including its own.
3.4.2.4.2. A valid linking is an acyclic directed graph, where the vertices are instrufaces and the edges are links.
3.4.2.4.3. The Select Links button enters selection mode as described in the edit mode section.
3.4.2.4.4. Selection of the mimic option forces the instruface to treat key depressions on linked instrufaces as key depressions on its own; Note that the corresponding pitches are maintained.
3.4.2.4.5. Selection of the transpose option has the following effect: when this instruface A is linked to some other instruface B on which a key has just been depressed, all of the corresponding pitches of keys in A are transposed by the difference between B’s central key and the depressed key.

3.4.2.5. Layout
3.4.2.5.1. The middle key text box specifies the corresponding pitch to the middle key of an instrument interface. For the PolyTheremin, this means the pitch corresponding to the exact middle position on its axis.

3.4.2.5.2. The Rows text box specifies the number of rows of keys in a keyed interface.

3.4.2.5.3. The columns text box specifies the number of columns of keys in a keyed interface. It doesn’t appear for the Piano Interface.

3.4.2.5.4. For the Hexatonnetz and Gridtar interface, there is a text box called string difference. This affects the corresponding pitches of keys as specified in their sections.

A PolyTheremin Interface

3.4.3. PolyTheremin

3.4.3.1. Does not have keys, but instead a horizontal axis corresponding to frequency. Values on this axis increase logarithmically to the right. Depressing a finger along the axis produces the corresponding pitch. Multiple fingers will produce multiple pitches. This is a single rectangle.

3.4.3.2. When an interface that outputs to a MIDI device links with transpose enabled to a PolyTheremin interface, MIDI pitch bend messages are sent to the device to approximate the pitch.

3.4.4. Piano

3.4.4.1. Consists of keys as defined at the beginning of the interface section.

3.4.4.2. The layout and corresponding pitches of keys is the same as on a traditional piano keyboard: Corresponding pitches increase by one semi-tone between adjacent keys. The longer white keys are only adjacent if there is no black key between them.
3.4.5. Gridtar

3.4.5.1. Consists of keys as defined at the beginning of the instruface section.
3.4.5.2. The layout and corresponding pitches of keys are a regular tiling of rectangles such that for any given rectangle \( A \), the rectangle to its immediate right has a corresponding pitch exactly one semitone higher and the rectangle above \( A \) has a corresponding pitch exactly \( d \) semitones higher, where \( d \) is the “string difference” variable defined on the layout tab of the instruface’s configuration dialog box.
3.4.6. **Hexatonnetz**

3.4.6.1. Consists of keys as defined at the beginning of the instruface section.

3.4.6.2. The layout and corresponding pitches of keys are a regular tiling of hexagons such that for any given hexagon A, the non-adjacent hexagon directly to its right has a corresponding pitch exactly one semitone higher and the adjacent hexagon above and to the left of A has a corresponding pitch exactly \( d \) semitones higher, where \( d \) is the “string difference” variable on the Layout property page.
3.4.7. Sample Pad

3.4.7.1. Tapping on the Sample Pad instruface plays the WAV file specified on its sound property page.

3.4.7.2. When a finger is released from the sample pad, the sound continues until the file has played to its end (unless the Loop checkbox is checked).

3.4.7.3. The Sample pad has a Loop checkbox on the layout property page. If checked, the sound continues playing in a looping fashion until the sample pad is depressed while sound is being played.

3.4.8. Volume

3.4.8.1. When not linked, the volume instruface controls the total mixed sound level of the application.

3.4.8.2. When linked to a set of instrufaces, a volume instruface controls their volume.

3.4.8.3. If more then one volume instruface has the exact same links and one is actuated in play mode, the others will change in the same way. This is true even if (and especially if) they are on separate tabs.

3.4.8.4. The volume instruface has no layout, view, or sound property pages. On its link property page, there are no Mimic or Transpose checkboxes.
4. Non Functional Requirements

4.1. User input hardware
   4.1.1. This system requires a multi-point touch screen.
   4.1.2. However, the system also requires a standard keyboard and mouse.

4.2. Performance
   4.2.1. Latency, measured as time between the user’s touch screen finger depression and corresponding perception of sound, must not exceed 100ms.
   4.2.2. The system must support the simultaneous operation of at least 32 instrufaces.

4.3. Evolution
   4.3.1. Future versions of the software will support modular instruface plug-ins and VSTi instrument plug-ins for sound generation.
5. Appendices

5.1. Glossary

5.1.1. Key - A geometric shape on screen that has a corresponding pitch. See the beginning of the instruface section.

5.1.2. Semitone – a difference of one discrete pitch in the 12 equal temperament scale. Also known as a “half-step”

5.2. References

5.2.1. Jeffery Han et al. have created a multi-point touch screen. This is the technique we will use: http://cs.nyu.edu/~jhan/ftirtouch/index.html

5.2.2. Please see our video mock-up of the hexatonnetz instruface at The following URL: http://www.pages.drexel.edu/~wbm25/prototype.wmv