MelodyMorph: A Reconfigurable Musical Instrument

Eric Rosenbaum
Lifelong Kindergarten Group
MIT Media Lab
Cambridge, MA
ericr@media.mit.edu

ABSTRACT
I present MelodyMorph, a reconfigurable musical instrument designed with a focus on melodic improvisation. It is designed for a touch-screen interface, and allows the user to create "bells" which can be tapped to play a note, and dragged around on a pannable and zoomable canvas. Colors, textures and shapes of the bells represent pitch and timbre properties. "Recorder bells" can store and play back performances. Users can construct instruments that are modifiable as they play, and build up complex melodies hierarchically from simple parts.

Keywords
Melody, improvisation, representation, multi-touch, iPad

1. INTRODUCTION
For the improviser, an instrument's interface creates a landscape of possibilities: from a particular gesture, some gestures are "nearby" and others are far; there is a set of familiar pathways that are easy to traverse; and there is a tradition of idioms to fall back on. This landscape can change over time, but slowly, because the physical configuration of instruments is typically fixed: the order of the piano keys or the tuning and fretting of a guitar do not change as you play.

What would it be like to radically reconfigure the interface to your instrument as you play it? A reconfigurable instrument could have the potential to create new possibilities for improvisers, and open up a new creative space. By "reconfigurable," I am referring to the ability to change the spatial mapping between gestures and sounds in real-time as you play the instrument. Imagine an exploded piano, with individual keys that you can position anywhere you want them.

Some affordances of traditional instruments will be lost with such an instrument, of course. The familiar pathways, if any, will be only temporary ones. And there is no tradition of idioms for a reconfigurable instrument (at least, not yet).

But other affordances unique to the new genre of reconfigurable instruments could be gained. For example, because the notes can be positioned anywhere, the proximity of gestures to each other is completely redefinable. For example, the leaps in an angular melody might make it difficult to play on a piano, but could be made easy on a reconfigurable instrument. Another new affordance is customizability. A reconfigurable instrument could be rearranged to suit the needs of a particular player, a particular composition, or even a particular moment in time.

I present MelodyMorph, a reconfigurable instrument designed with a focus on melodic improvisation. It is designed for a multi-touch screen interface. A palette allows the user to create "bells," graphical representations of individual notes that can be played by tapping and moved by dragging. Special "recorder bells" can store and play back sequences of bells. A pannable and zoomable canvas enables the creation of large melodic landscapes. Canvases can be saved and reloaded for later use. My initial investigations suggest that this interface is usable, fun, and ripe with new possibilities.

2. RELATED WORK
The inspiration for MelodyMorph comes in part from a set of educational manipulatives called Montessori Bells. They are small metal bells on wooden stands, identical in appearance but varying in pitch. The bells are used with young children to pose puzzles about pitch (which two are the same?) or melody (can you construct this tune?). Jeanne Bamberger has investigated children's thought processes as they worked with the bells and invented their own notations that act as instructions for other children to play them [1]. Drawing on this work, one of the initial concepts motivating the MelodyMorph project is the idea that people could use it to create something that acts as both a "notation," visually representing the structure of a melody, and simultaneously as an instrument that lets you expressively play that melody.

There is relatively little existing work in the area of reconfigurable instruments. Tangible controllers such as Audiopad [2] and reacTable [3] allow the user to control sound synthesis and sequencing by positioning and manipulating pucks on a table. Unlike MelodyMorph, these controllers focus on controlling parameters, rather than constructing melodies. The reacTable-based scoreTable [4] focuses on melody construction, but it uses a sequencer metaphor. Sequencers in general have fixed mappings, so they are not reconfigurable by my definition. Pin & Play & Perform [5] enables the ad-hoc construction of instruments, by pinning dials, sliders and buttons onto a conductive substrate. The focus of that project is more on controlling musical parameters than playing melodies. Similarly, the Spinner project [6] enables users to freely map physical dials onto GUI controllers for musical parameters.

3. MELODYMORPH DESIGN
3.1 Palette
The palette is shown at the top of the screen (see figure 1), and enables the user to quickly create any number of bells. It shows one octave at a time of a chromatic scale. Buttons at the sides slide the palette up or down an octave, giving a total of three octaves. An instrument switcher at the top of the palette makes three different instrument timbres available.

3.2 Bells
Each bell can be tapped to play its corresponding note. An animation showing it briefly increasing in size and then shrinking provides feedback that it is playing. A bell can be dragged by pressing it near the center (causing it to play), and then dragging beyond its edge. A bell can be deleted by dragging it back to the palette.

The bells have different shapes to represent the different instrument timbres: circles for bass, squares for piano, and triangles for vibraphone. Their colors indicate different pitches. The range of twelve pitch classes is mapped onto the cycle of hues, resulting in a rainbow of notes. The lower octave is shown with darker colors, and the upper octave is shown with lower saturation, for paler colors. Additionally, visual textures are used to differentiate the "function" of each note within the key of the root note shown in the palette. Chord tones (1, 5, and 8, in the chromatic scale) are shown with a solid color. Other tones in the major scale are shown with a horizontal band (3, 6, 10 and 12). Tones outside the major scale have vertical stripes (2, 4, 7, 9, and 11).

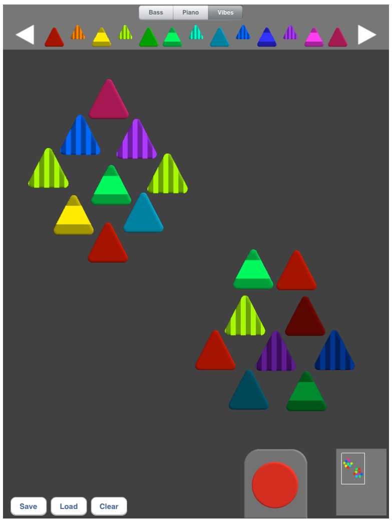

Figure 1: The MelodyMorph Interface

3.3 Expression

When each bell is tapped, the force of the tap is estimated using data from an accelerometer in the axis perpendicular to the screen. This value determines the loudness of the resulting note. Additionally, if a bell is held down while it is playing, an accelerometer in the plane of the screen is used to determine pitch bend. This enables control of pitch, by wiggling or tilting the whole device, over a range from subtle pitch vibrato to whammy effects.

3.4 Canvas

The bells inhabit a canvas much larger than the screen. The canvas can by panned by dragging anywhere there is not a bell. It can be zoomed in and out with a pinch gesture. Zooming way out gives a view of the entire composition. Zooming way in makes the bells larger and easier to tap. A small "mini-map" in the corner of the screen shows the entire canvas, with colored dots representing the bells and a box showing the current screen view.

3.5 Recorder bells

A recorder widget at the bottom of the screen can be tapped to toggle on recording. Any bells that are played are recorded into a special "recorder bell," which appears as soon as recording is toggled off again. This recorder bell behaves in many ways like a regular bell. Tapping it triggers the playback of its recorded sequence. A tap during playback stops the sequence. The recorded sequence is notated on the surface of the bell as a sequence of colored dots, with pitch on the vertical axis, and normalized time on the horizontal axis (see the white and gray objects in figure 2). During playback, the notation is animated, with each note appearing as it plays. The recording process includes recording the playback of other recorder bells, enabling complex melodies to be built up in layers.

3.6 Saving

A canvas with its configuration of bells and recorder bells can be saved as a file, along with a thumbnail image of the screen, for later reuse.

3.7 Implementation

MelodyMorph has been implemented on an iPad, under iOS 3.2. The functionality is built primarily using OpenFrameworks, with some Objective-C for user interface elements.

4. SCENARIOS

Here I will provide two examples of ways people might use the MelodyMorph system.

4.1 Kalimba Making

A simple use case (and one the author enjoyed early in the development process) involves constructing a small spatial arrangement of notes that are consonant with each other, and playing patterns on them with the fingers or thumbs (see e.g. figure 1). The result is a bit like a Kalimba, or thumb piano, except that it is completely customizable, and can even be modified during a performance.

A refinement of this technique involves constructing two or more kalimba patterns based on related chords, and switching between them to create a more complex improvised structure.

4.2 Hierarchical Melody Construction

A more elaborate use case involves using the recorder bells to build up a complex structure out of simpler elements.

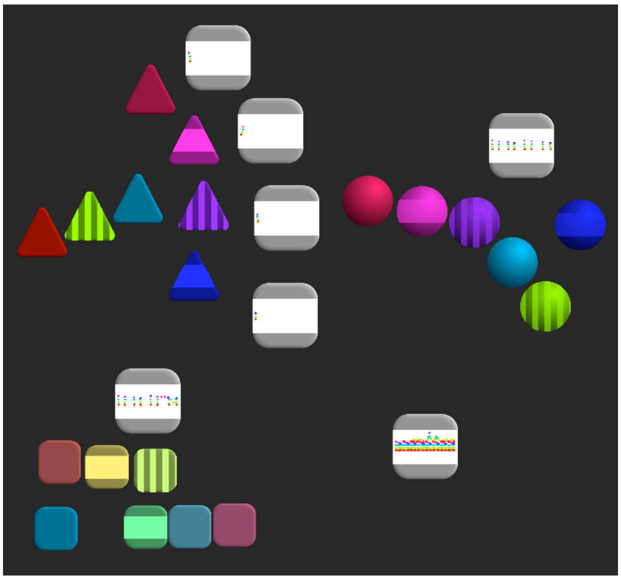

Figure 2: A more complex MelodyMorph construction

Figure 2 shows a melodic structure built up in layers. At the top left, a minor triad is shown along with four different upper structure notes (representing a standard descending "line cliché"). The four recorder bells next to these notes contain four note minor chords constructed with each of them. These four chords were then recorded in a sequence, twice through, resulting in the recorder bell at the top right. Below it, bass notes are shown in a pattern that matches the descending line cliché, along with two additional chord tones. This was used to play a bass line along with the chord sequences, resulting in the recorder bell at bottom left. Below that, a cluster of piano notes was constructed for the purpose of improvising a melody along with the accompaniment created so far. An improvisation over four repetitions of the accompaniment was recorded into the recorder bell shown at the bottom right.

5. EVALUATION AND FUTURE WORK

A priority for continued work on MelodyMorph is to carry out some initial user studies, consisting of careful observations of users with various degrees of musical training as they play and improvise with the interface. These studies will likely reveal bugs and usability issues that can then be resolved.

One problem with MelodyMorph is that the touch screen provides no tactile feedback, making it difficult to tap the bells accurately, especially when zoomed out. It's not clear how best to provide this feedback. A tangible version of MelodyMorph could provide such feedback, but would only be feasible if the individual bells (each with sensing and communication on board) could be made cheaply enough that a large number could be fabricated.

Another problem with MelodyMorph is in synchronization. Because it does not use a sequencer metaphor, there is no fixed time base. It can be difficult to accurately time a melody when "overdubbing" on to a recorder bell. It may become desirable to add a toggle-able metronome to provide a tempo reference.

I am considering several features to add to the system. An annotation system would allow users to draw in freehand on the canvas, so they could do things like label melodic sections, decorate their instruments, and create flowcharts showing how to play larger melodic structures.

An additional palette section for "transformation" elements would contain objects that can be applied to bells and recorder bells, effecting musical transformations such as transposition, harmonization, timbre changes, etc.

The system of colors, textures, and shapes to represent pitch, function and timbre will be evaluated for its intuitiveness and possibly redesigned. Similarly the notation system for recorder bells may need to be redesigned.

MIDI or OSC output would enable MelodyMorph to send control data to other synthesizers, creating much more flexibility in possible timbres.

A networking feature would enable multiple devices to share data, such as the ability to pass groups of bells and recorder bells between devices.

6. REFERENCES

[1] Jeanne Bamberger. 1995. The Mind behind the Musical Ear: How Children Develop Musical Intelligence. Harvard University Press, Cambridge.

[2] James Patten, Ben Recht, and Hiroshi Ishii. 2002. Audiopad: a tag-based interface for musical performance. In Proceedings of the 2002 conference on New interfaces for musical expression (NIME '02), Eoin Brazil (Ed.). National University of Singapore, Singapore, Singapore, 1-6.

[3] Sergi Jorda. 2003. Sonigraphical instruments: from FMOL to the reacTable. In Proceedings of the 2003 conference on New interfaces for musical expression (NIME '03). National University of Singapore, Singapore, Singapore, 70-76.

[4] Sergi Jorda and Marcos Alonso. 2006. Mary had a little scoreTable* or the reacTable* goes melodic. In Proceedings of the 2006 conference on New interfaces for musical expression (NIME '06). IRCAM, Centre Pompidou, Paris, France, France, 208-211.

[5] John Bowers and Nicolas Villar. 2006. Creating ad hoc instruments with Pin&Play&Perform. In Proceedings of the 2006 conference on New interfaces for musical expression (NIME '06). IRCAM, Centre Pompidou, Paris, France, France, 234-239.

[6] Shigeru Kobayashi and Masayuki Akamatsu. 2005. Spinner: a simple approach to reconfigurable user interfaces. In Proceedings of the 2005 conference on New interfaces for musical expression (NIME '05). National University of Singapore, Singapore, Singapore, 208-211.