Using the Mouse-Keyboard System as an Advanced MIDI Controller: An Innovative Approach.

Parth Dalal

Abstract — **External MIDI controllers have long been used by producers to aid in musical composition on a computer. Because advanced MIDI controllers are often expensive investments for the hobbyist musician, it would be useful for one to use a simple keyboard and mouse to send much of the same messages an advanced MIDI controller is capable of sending such as velocity, pitch bends and modulation. We present an innovative approach in which a commonplace keyboard and mouse can be used together to emulate an external MIDI controller through software that is programmable by the user to match his or her compositional preferences.**

I. Introduction

MIDI, which stands for Musical Instrument Digital Interface, is a communication protocol that electronic devices use in order to transmit musical information to compatible equipment [1]. Devices capable of sending MIDI messages, called MIDI controllers, commonly come in the form of a piano keyboard with a pitch-bend wheel, mod wheel, and additional controls. A pitch-bend wheel allows one to change the pitch of a note as it is played. This allows for more articulate musical compositions. A mod wheel is used to change any given parameter provided the user assigns that parameter to the mod wheel. Musicians that would like to record MIDI messages to their computers may use external MIDI controllers to play notes. Compatible software can then record the time a note was held down, the velocity with which it was struck, the after touch, the pitch bend, and many other characteristics that the MIDI communication standard supports, provided the controller is capable of outputting that kind of information.

There are also many unorthodox MIDI controllers, those that allow different forms of expression from traditional devices [2]. Mostly, each of these devices is unique hardware that does not come with a personal computer. To the best of our knowledge, there has been very little work done towards using a simple mouse and keyboard system as an advanced MIDI controller.

The purpose of this paper is to explore an innovative concept in which a typical computer keyboard and mouse could be used together as a MIDI controller to send complex MIDI messages. We first discuss the limitations of using a keyboard alone, and then we explore methods for the user to be able to send advanced information such as velocity and pitch bends through the keyboard and mouse together. Velocity is an important parameter of a MIDI message that dictates the speed at which a note was struck. Based on the speed, the resulting sound may be louder or softer, sharper or duller, or as programmed in the playback software. We then explore the development of a prototype MIDI controller plug-in or independent application that processes keyboard and mouse input as programmed by the user to output advanced MIDI messages. In essence, we are talking about a customizable MIDI instrument made out of a simple mouse and keyboard system.

Parth Dalal is with the Computer Science Department, Oklahoma State University, Stillwater 74078. Email: parth@okstate.edu

II. Sending Velocity and Pitch-Bend

Computer keyboards are already used to send MIDI note information through music software [3]. The catch is that these messages can only correspond to a note held down or a note released, because a computer keyboard does not detect velocity or pressure; a particular key can only be in the down or up position. Velocity information is not sent because there does not seem to be an intuitive method for a user to be able to express such information using the keyboard alone. Pitch bends cannot be sent either; generally, to bend the pitch of a note, a sort of knob or wheel is most intuitive. For a software controller, a virtual knob or wheel exists that can be moved with the mouse by clicking and dragging – for a hardware controller, a physical knob or wheel is usually present. An everyday keyboard only has keys on it.

To be able to send advanced messages, we must use the mouse in conjunction with the keyboard. It would be very natural to use the scroll-wheel, present in every modern mouse, as a pitch-bend or modulation wheel. The question then arises of velocity. Since velocity is about the speed at which one strikes a key, the most intuitive method to measure velocity is by using the motion of one’s hand. The movement speed of the mouse is a reasonable way to measure velocity. This may seem counter-intuitive, because one hand is controlling the velocity of notes played by the other hand. But there are many examples of instruments that require musicians to do the same. Guitarists strum with one hand while holding down notes with the other. Perhaps the best example of velocity being controlled by a separate hand is that of the harmonium. One hand plays the keys and the other pumps a bellow. Compare this to the concept of our MIDI controller, where one hand plays the keys and the other moves the mouse. This concept can be seen in Image 1 below.

Image 1: Using the mouse to control velocity

To play one's keyboard and mouse as a MIDI controller, one would play a note on the keyboard and simultaneously move the mouse in any direction at the desired speed. The easiest way to play would be to move the mouse back and forth so that the overall position of the mouse does not change over time.

III. CONCEPT TESTING

We developed a simple program to test the feeling of using the mouse to control the velocity of a note. The program works as follows: The 'spacebar' key is pressed to play a harp pluck WAV sample at a volume corresponding to the speed of the mouse at the time the key was pressed, as shown in Image 2.

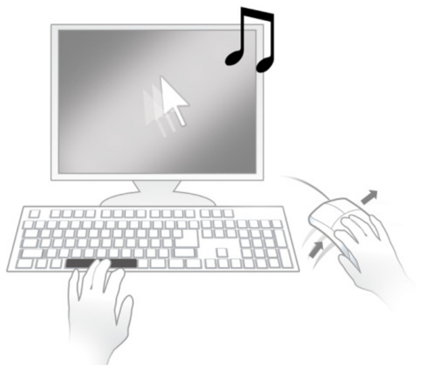

Image 2: Testing the concept with a simple program

In this program, the mouse speed is measured based on the position of the Windows cursor. The algorithm consists of recording the distance of the cursor from the center of the program window, then re-positioning the cursor to the center. A drawback of this algorithm is that the speed depends on the speed of the cursor which is not necessarily proportional to the physical speed of the mouse hardware device. However, this algorithm is merely used to test the feel of the method used to measure velocity. As we build on this program, a more advanced algorithm will be used.

We conducted a pilot study in which a test subject who knows how to play a musical instrument (guitar) used the program and commented on the feel. For this subject, using the mouse to control the sample volume felt natural. Based on this input, we believe that a more sophisticated program, described in the next section, also feels natural to musicians.

IV. ADVANCED CAPABILITIES

The mouse's left, right, and middle click buttons will be used to output more advanced MIDI messages. Since there are so many possibilities, a software program that allows the users to customize the actions for those buttons is handy. Take the example of a hammer-on, which is a guitar technique where a finger hits a note close to the currently played note to instantly change the pitch of the current note. The hammer-on on a guitar sounds similar to instantly snapping the pitch-wheel to a certain position near the current note. A user emulating guitar-like sounds could program one of the mouse buttons to instantly move the pitch wheel to a certain position on the button's DOWN event. The mouse button's UP event could be programmed to reset the pitch wheel to the neutral position. Now, the user could hit keys with different velocities, adding hammer-ons every so often for very expressive playing and detailed MIDI recording.

Another example of this customizability is that the user could program the plug-in to change the mode by holding down a mouse button. For example, normally the speed of the mouse could control note velocities, but when the right mouse button is held down, the up or down movement of the mouse could control the modulation wheel. When the right mouse button is not held down, the mode could change back to normal. There are many such possibilities and allowing users to program their own instrument can result in new musical creativity. The instrument one is using to compose definitely affects the resulting nature of the melody.

A customizable software essentially allows the user to program in an object-oriented environment. As with all programming, there are conditions that trigger certain actions. In our software, conditions trigger actions that affect MIDI parameters by formulas that use Characteristics. Below are various Conditions, MIDI Parameters, Characteristics, and sample pseudo-code.

Conditions:
Mouse moved horizontally
Mouse moved vertically
General Mouse Movement
Right-Mouse Down/Up
Left-Mouse Down/Up
Middle-Mouse Down/Up
General Click
Mouse-Wheel Movement

MIDI Parameters:
Sustain Pedal
Pitch-Wheel
Mod-Wheel

Characteristics:
MIDI Pitch-Wheel State

MIDI Mod-Wheel State
MIDI Sustain Pedal State
Mouse X Speed
Mouse Y Speed

Sample Pseudo-Code:

```
If there is General Mouse Movement then
Set the Pitch-Wheel to: MIDI Pitch-Wheel State + Mouse Y Speed
```

Of course, the user is not required to do as much typing as one would do in a traditional programming environment. The interface is very visual. For example, the interface has a mouse graphic on the screen. When the left-mouse button of the mouse graphic is clicked, it opens up a menu giving several options regarding the action to perform for the condition: Left-Mouse Down. These actions include setting parameters, sending a MIDI message, and many other actions. When changing or setting a value, an equation editor allows the user to use Characteristics as variables in complex algebraic expressions.

V. Development Process

A simple software of this type will be created in as little as 2 months by following the steps below:

1. Create a basic template program based on the program in the 'Concept Testing' section.
2. Use system parameters to get the 'physical' speed of the mouse as opposed to the cursor's speed on screen. This ensures that the feel of playing the mouse-keyboard MIDI controller does not vary from computer to computer.
3. Elaborate on this template by playing the sound sample at a different pitch based on the key pressed on the keyboard.
4. In addition to playing the sound sample, send a MIDI message. For example, if a key on the keyboard is hit so that a middle F note is played at volume 100, send a MIDI message for a middle F note ON with velocity 100.
5. Add functionality to detect changes in the mouse wheel and send corresponding changes to the MIDI pitch bend so that the mouse wheel acts as a pitch bend wheel.
6. Begin to add customizing options by creating functions that perform certain actions such as changing the pitch bend wheel by a certain value or toggling the sustain pedal. Each function is assigned a value, and these values will be stored in an array. Each Condition will be assigned a value, and these values will be stored in another array. The user programmed settings are then numbers. The numbers are looked up in the array to see what functions to run. Start small and only allow a little customization.
7. Create an intuitive graphical user interface.
8. Save an executable of the program.
9. Attempt to create a Virtual Studio Technology (VST) plug-in out of the program.

VI. Conclusion

At this stage, software development is in its beginning phase. In the next phase, following steps from the 'Development Process' section, the work will be refined to a prototype application that simulates a plug-in of this type, one that takes keyboard and mouse input, processes it according to what the user wants, and plays some notes and/or outputs MIDI data. Such an application meets a clear need of a simple mouse-keyboard system that serves as an advanced MIDI controller at virtually no cost.

References

[1] J. Rothstein, MIDI: a comprehensive introduction. Madison, WI: A-R Editions Inc., 1993

[2] "Non-Traditional MIDI controllers," [Online]. Available: http://www.midi.org/aboutmidi/alternate.php [Accessed: Sep. 29, 2009].

[3] "Sweet Little Piano for Windows – freeware" [Online]. Available: http://www.ronimusic.com/sweet_pi.htm [Accessed: Oct. 1, 2009].