

Justin Binder

Group Members:  
Victoria Shen  
Brian Han

ES 95  
May 13, 2010

## Fanny Pick



### Introduction

As a lifelong guitar player, I often find myself pretending that there is a guitar in my hand. Walking down the street, sitting in class, or doing other everyday tasks, I can be found moving my fingers together and humming guitar melodies. When brainstorming for this project, I realized that it would be amazing if I could harness the finger-twiddling gestures I'd been doing and create some real music.

*Fanny Pick* is an electronic instrument based on the input method and logic system of a guitar. The instrument consists of a *Digital* machine, a glove fitted with a variety of sensors, a pedal board fitted with buttons and other sensors, and a webcam. These hardware components work together to allow for dynamic control of more than two octaves of notes.

## Hardware

**DIGITAR:** The *Digitalar* is an Italian made guitar string emulator. It consists of six metal wires housed in a plastic box that can be worn around ones waste. Originally, designed to allow for rhythmic control of keyboard-chosen notes, the *Digitalar* sends default midi data if plugged straight into the computer. That default data consists of the note numbers of a guitar in standard tuning (ex. The sixth string sends note # 40 corresponding to a low E). The *Digitalar* also sends velocity information, and thus is a great emulator of a real guitar (the force of a string plug corresponds to note volume on a guitar).

**GLOVE:** We used an old mechanics glove for “fret” selection in the left hand. We glued small round FSR sensors onto the tips of each finger, and placed a bend sensor on the back of the thumb. Each finger corresponds to a fret on a guitar, and thus by pressing a finger against one’s thumb (or any other surface) and then plucking a given string, a particular note can be chosen. Four frets times six strings gives 24 note options.

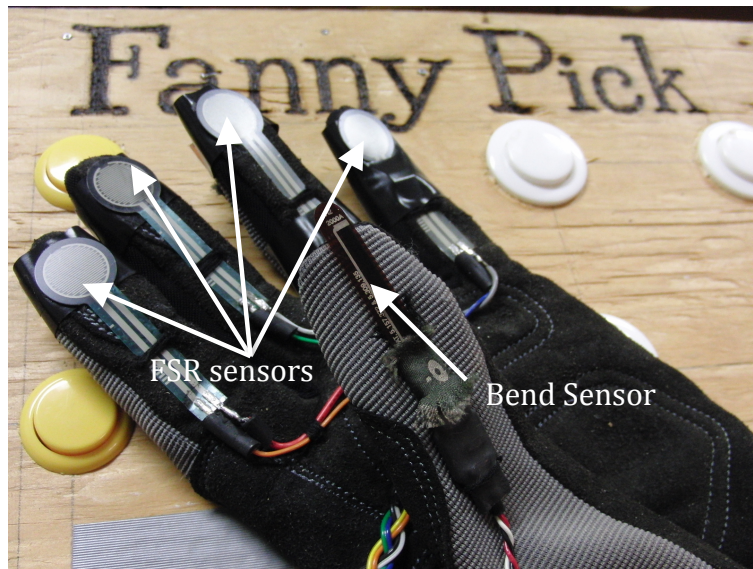


Figure 1 – Left hand “fretting” glove

**PEDAL BORED:** We created a pedal board to allow for easy manipulation of the Fanny Pick’s parameters. Firstly, the yellow buttons allow for the key center of the instrument to be incremented up or down. When the instrument is first turned on (the MAX patch is first loaded) a default key center of G is set. This means that a note choice of the first finger, sixth string corresponds to note 55, or a range from the third to sixth fret of a guitar. By pressing on the yellow buttons, that key center can be shifted up or down. The white buttons correspond to four different preset voices. These voices are described below under the Reason section. Finally, we mounted

four large FSRs to be used as parameter controllers. For example, for one of our subtractor patches, we used an FSR to control filter frequency.

WEBCAM: We used the webcam built in the iMac to control delay by waving the left hand in front of it. Our use of the webcam is described in greater detail below.

## MAX Programming

NOTE SELECTION: We set up the note selection code to work such that a note is created when both a finger is pressed and a string is plucked. The note rings until the contact is removed from the finger. When a finger is pressed, a series of MAX switches opens that essentially allows for a clear path through which a string bang is free to bang a makenote object of a given pitch. We made a subpatch for each string that has four of these “paths”. The pitches are determined relative to a base pitch such that the tuning matches that of a standard guitar.

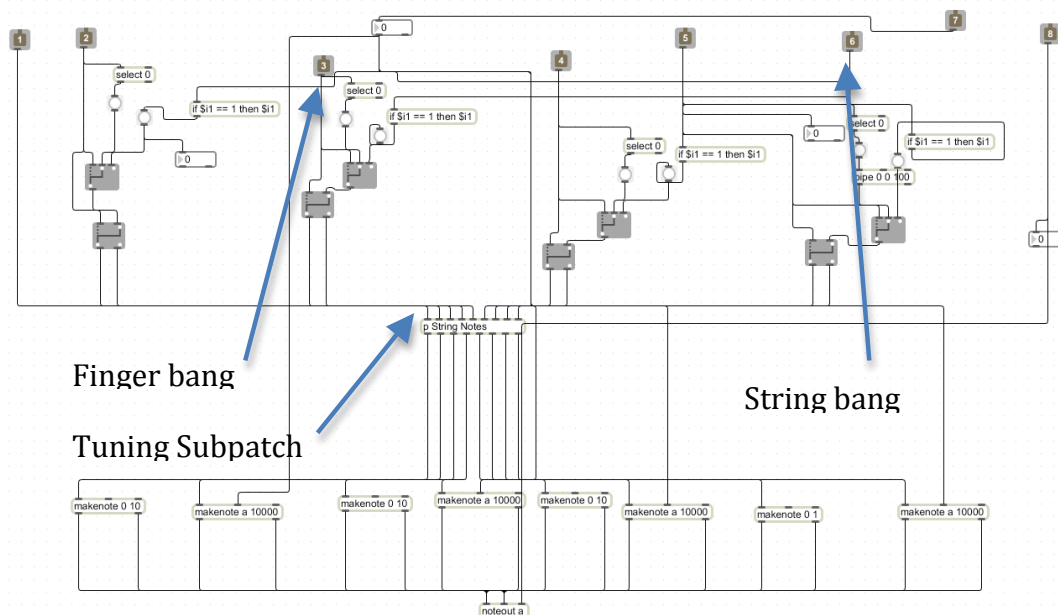


Figure 2 – Note selection subpatch

PITCH BEND: The thumb of the glove was fitted with a bend sensor. We mapped the amount of bend in the thumb to pitch bend for all of our voices.

WEBCAM: We analyzed the video from a webcam and mapped it to the steps parameter on a DDL-1 delay module in Reason. As a video image comes into MAX, it is constantly being compared to an image that happened soon before it. Any differences between the old and new image are noted and a box is placed around the new part. Thus as a hand is waved across the screen, the box moves across the screen, and gets larger or smaller depending on how close to the webcam the musician is.

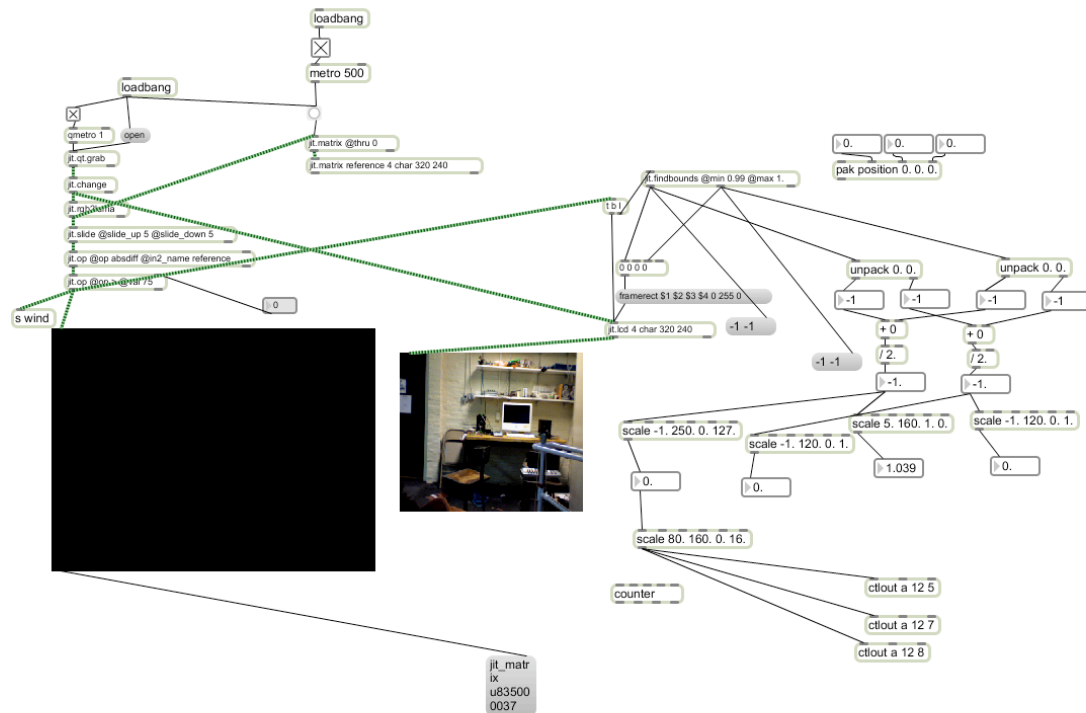


Figure 3 – Webcam analysis subpatch

PEDAL BOARD FSR: We used large square FSRs (wired with resistors for decreased sensitivity) on the pedal board to control Reason parameters such as filter and distortion amount. One of our goals was to be able to use the FSRs for both dynamic input and to set a certain value and have it stay there. We accomplished this goal by programming the FSRs such that if they are stepped on and the pressure is suddenly removed, the previously maximum value is maintained. If the pressure is gradually reduced then the value moves down towards zero.

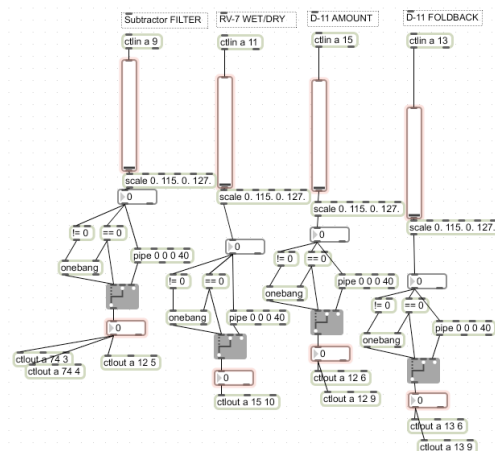


Figure 4 – Pedal FSR programming

VISUAL INTERFACE: Since there is no physical guitar in the players hands, we wanted to design some sort of feedback system so that the musician could feel confident about what notes they were playing and what their various settings were. Our presentation mode consisted of a variety of LEDs and sliders to show when fingers were pressed, strings were plucked, what the pedal board parameters were, and where on the virtual guitar neck they were playing.

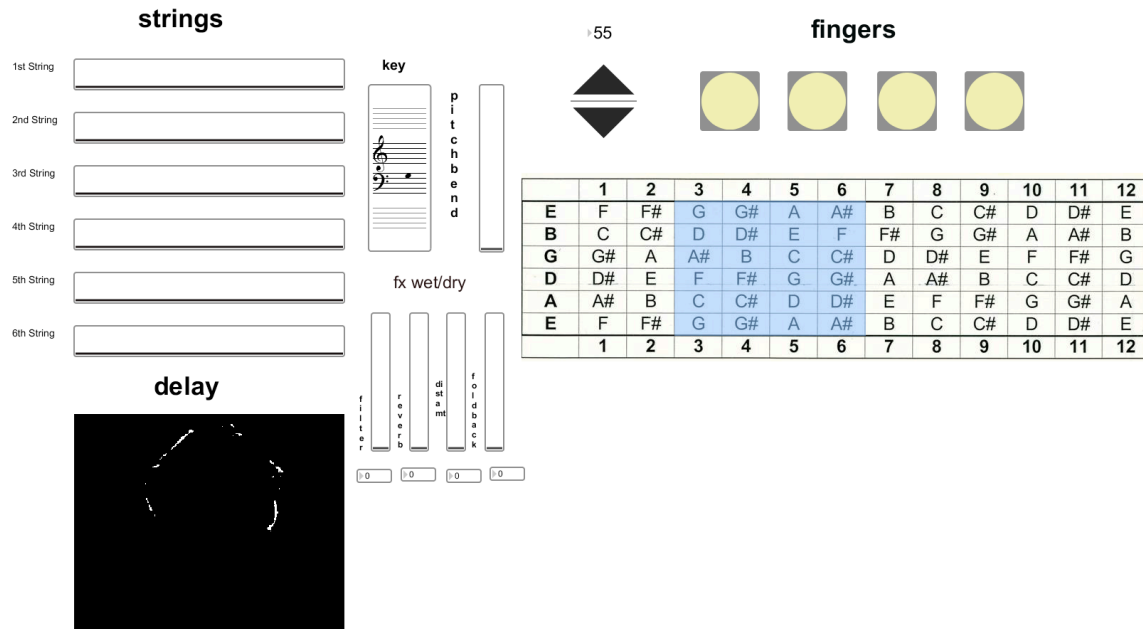


Figure 5 – Fanny Pick presentation mode

We used a system of U-button objects overlaid on a picture of a guitar fret board (essentially just a chart of notes). When a key center is chosen, a blue box highlights which four frets of the neck the musician can access. If a note is played, a small red box lights up the particular note on this virtual fret board. The logic behind the individual notes was particularly tricky to figure out. Although there are only four “frets” of notes able to be played, those frets could be anywhere on the neck. Thus we recreated the code ten times in order to cover the entire neck.

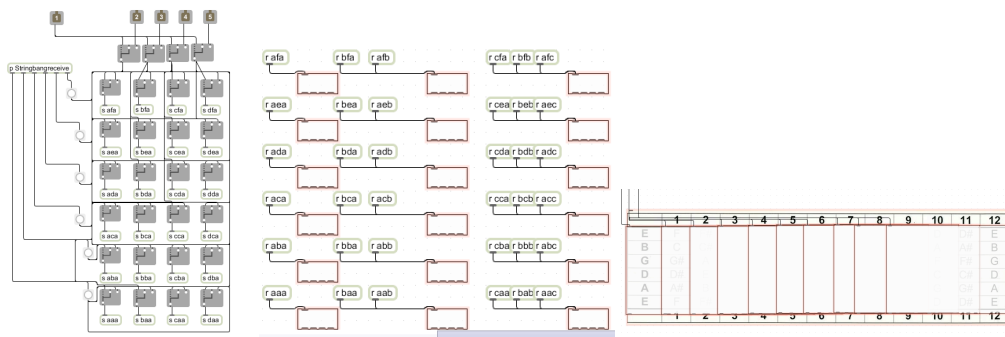


Figure 6 – Various aspects of the fret board visual interface



## Reason

There are four different voices in our reason patch. Each one is accessible by pushing one of the white buttons on the pedal board. The first is an NNXT violin section sample bank playing through both delay and distortion. The second is an NNXT upright II sample bank, also playing through delay. The third and fourth are homemade subtractor patches.



Figure 7 – Reason Patch

## Obstacles

The FSRs on the glove posed a lot of problems throughout the course of this project. As they are attached to the length of each finger they often broke due to the stress of being bent while the instrument was being played. Furthermore, they consist of a relatively small contact surface and thus the musician must be particularly careful to press exactly on the FSR in order to create a successful note. In the future, rather than using FSRs, I would use wires contacts that complete a switch when the thumb and finger make contact. This would allow more flexibility in terms of the area of the contact surface and ideally more durability.

## Conclusion

Overall, I considered this project to be very successful. We were able to complete the initial construction and programming very early on, and thus had ample time to devote to more interesting challenges such as the webcam programming, the visual interface, and the pedal board.

## **Group Contributions**

In general, we all helped each other throughout this project.

I did most of the MAX programming, including patches for note selection and the fret board visual interface. Victoria did a lot of the physical work with the glove. She soldered the FSRs, braided the wires, and sewed them nicely so that they didn't interfere with playing. She also created the webcam patch, and helped with other MAX stuff. Brian created most of the pedal board, and programmed the FSRs so that they could work as both discrete and continuous controllers. I had a great time working with both of them!