COS 436 HCI

Final Project

Presented at the 1st Group Fire Convention

2008 January 18 Princeton, NJ, USA

Acceleguitarometer

John "Johnf" Fontein— in charge of all things sound + Chuck related

Samuel "Samja" Albert— in charge of all things Wii-mote + Processing related

Ota-"no-show"-Benga Amaize— debugger extraordinaire

Correspondence should be addressed to John Fontein (john.fontein.john.fontein.99@gmail.com)

Abstract

The *Acceleguitarometer* is a virtual "air" rock band. "Air" playing refers to the act of flailing ones arms in the air, pretending to play a specific instrument, while not actually holding an instrument in hand or producing any sound. Using nothing but a Wii Remote Controller, a Wii Nunchuck Controller, and provided software, a user is able to "air" play essentially all instruments of an entire rock band. Users are able to switch between 2 modes of play, Loop Mode, where users can browse through and play pre-selected loops while altering their sonic qualities, and Instrument Mode, where the user plays an acoustic instrument live, "strumming" a guitar or "hitting" a snare drum. All instrument motions are measured using the Wiimote's built in accelerometers. The applications and future of such virtual instruments are explored.

1. Introduction

However much you love music and however much you want to play the guitar, if you were never trained to play the instrument or cannot even afford to buy the instrument, you will never have the chance to live out your rock star dreams. As a result, many people tend to stray away from the electric guitar, instead resorting to playing the "air guitar," an instrument which requires no skill, and no training whatsoever to master. However, the downfall of the air guitar isthat it does not produce any sounds, solely relying on the user's imagination or humming to recreate, for example, the classic sounds of a Fender Stratocaster. But what if you could produce real guitar sounds while air guitaring, without ever having to hum a single note? What if you could immediately sound like Jimi Hendrix or Wes Montgomery without having any previous guitar experience? What if you could produce the same rich sounds as the legends, have the same experience of playing a real guitar, and even have a backup band jamming behind your grooves, with nothing but a game controller in your hand? The *Acceleguitarometer* promises to provide all that, plus a little more.

The motivation to build the *Acceleguitarometer* was so that all people, musicians and complete music newbs alike, people who have never had the time to pick up an instrument and learn it, people who couldn't afford to buy an instrument, or even people who have never had the courage to "make friends" and join a band, would have the chance to ROCK N ROLL!!!

Further, the *Acceleguitarometer* hardware is fairly inexpensive, once again allowing almost anybody, regardless of wealth, to purchase the instrument.

Acceleguitarometer equipment consists of the following hardward and software devices:

- Wii-Remote
- Wii-Nunchuck
- Processing
- Chuck
- DarwiinRemote
- Redbull

Essentially, the Wii-Remote and Wii-Nunchuck connect via Bluetooth to a host computer, where Processing and DarwiinRemote capture motion and joystick button inputs and send them to Chuck which controls audio events. Sound too good to be true? Let us now take a closer look at the controller which is at the heart of the instrument, the Wii-remote, and explain how it's all done.

2. Nintendo Wii-Remote

Often referred to simply as a "Wiimote," this device revolutionized game controllers through its ability to sense motion using its built in accelerometers.

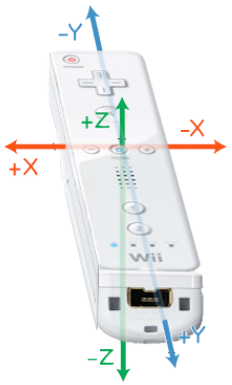

(1)

Although the device was originally manufactured for the sole purpose of manipulating the Nintendo Wii console, since its arrival in 2005, its functionality has since been extended to many other applications, including being used as a light saber[2], a finger tracking device[3], and even being used in a Low-Cost Multi-point Interactive Whiteboard system.[4] Naturally, Group FIRE believed the next step in the advancement of the Wiimote was a virtual air rock band. Although previous research such as Pakl et al.'s Wii Strum[5] has looked into the possibilities of using the Wii-Remote as an air guitar, the *Acceleguitarometer* is the first of its kind to control multiple instruments and multiple audio loops. Further, it is one of the few Wii-mote applications that combines both Chuck and Processing with a smooth graphical interface.

2.1 Wiimote Hardware

So what exactly is the Wiimote made of? The Wiimote consists of the following:

- The ADXL330 Accelerometer senses acceleration on 3 axes, as well as the device's pitch and roll position. As we will see later, changes in the motion on each axis are mapped to a similar musical movement, and thus control some audio event.



- The built in BCM2042 Bluetooth adapter allows us to seamlessly connect our Wiimote to our bluetooth enabled laptop.
- Although the Wiimote itself contains an H7824HE Speaker driver, in theory allowing all audio to be played back on the Wiimote itself, at this stage of our research, we are yet to figure out how to tap into this functionality.

2.2 Using Wiimote as Acceleguitarometer

Now let us look at exactly how we use the Wiimote as an *Acceleguitarometer,* and learn exactly what is involved in playing this instrument. We will first explore the different controls the user has on the Wiimote when *Acceleguitarometering* (can also be used as verb!), then we will examine exactly how our Wiimote speaks to our software. As mentioned previously, the user has 2 different modes to choose from: Instrument Mode and Loop Mode.

3. Instrument Mode

In Instrument Mode, the user can play different instruments live. Although the main instrument is a guitar, the user has the ability to switch instruments at the click of a button. Other instruments include a drum set (user can "hit" different drums using Wii-Remote), a shaker set (user can "shake" shaker instrument by using the Wii-Remote), and other acoustic instruments such as a piano are available to be freely played at this point. All code is open-source and is also included with the *Acceleguitarometer* package, so if the user has experience programming in Chuck or Processing, he or she can easily add different instruments to play. In such a way, our instrument is fully expandable.

3.1 Basic Instrument Mode Controls

The image below shows the function of each key on the Wiimote while under Instrument Mode.

- **Power** — Turns Wii-Remote on/off
- **Instrument Selection** — Select different instruments to play (Guitar, Drums, Piano)
- **Toggle—** Toggle between Instrument Mode and Loop Mode
- **Octave—** Switch between octaves
- **Favorite Instrument 1 & 2—** Save favorite instrument settings
- **String Selection Lights—**Highlights strings being played (Guitar only has 4 strings)
- **Record —** Enter Record Mode (very buggy at this point, known to crash program).
- **Chord Toggle—** Choose which chords to play
- **Minor Chord—** Play minor chords (plays major chords by default)
- **Dominant Chord—** Play dominant chords

Many of the functions are pretty straightforward here, once again allowing any novice to play the device. Highlights in Instrument Mode include being able to change chord quality at the press of a button between a minor, major, and dominant chord. When the selected instrument is a drum set, the user can change between kick, hi-hat and snare drum sounds.

The user is also able to easily switch between 8 different chords using the chord toggle button. Within each chord, individual notes can be played based on the pitch position of the device, while full chords are played by detecting the acceleration of the device. We discuss this further in motion specifics.

Finally, the user is able to store and recall 2 favorite instrument presets. This allows the user to quickly pick up the device and choose, for example, a guitar in favorite instrument preset #1, rather than having to

scroll through all instrument options to find the guitar

3.2 Instrument Specifics

The user is able to change not only the main instrument being played but also the sub-instrument being played. That is, once the *AcceleguitarMaster* (one who plays the *Acceleguitarmeter*), has selected an instrument family (such as guitar, piano, bass, synthesizer, etc.), he or she can then select a style of instrument (such as acoustic guitar, electric guitar, etc).

Each instrument in Instrument Mode has several different types of the instrument, each differing in acoustic qualities that the user can choose from. The list is as follows[1]:

- **Guitar, Bass, Piano**: Electric, Acoustic, Distorted, Jazz, Dance, Funk
- **Drums**: Rock, Jazz, Pop, Funk, Hip-Hop
- **Synth**: various synthesizer sounds

3.3 Playing Specifics

In Instrument mode, we have designed and tested different motion gestures for a realistic modeling of playing different instruments. Ideally, each motion should closely replicate those required for playing the actual instrument selected, within some reasonable bounds. For example, to create a strumming sound on the guitar, the user makes a strumming motion with the Wiimote, exactly as if he or she is playing a real guitar. If the user is on the drum preset and wants to hit the snare drums, once again, the user would snap her wrists and create a hitting motion with the Wiimote, exactly as if there were a real snare drum in front of the user.

The *Acceleguitarometer* currently registers the Wiimote's pitch and acceleration for analysis. Motions along the horizontal axis and changes in pitch act as different musical techniques and give users the option of playing a chord in either solid or broken form. A strong shaking motion is registered as a full chord in solid form, whereas rotations across specified pitch positions

[1] List is subject to frequent changes.

activate individual notes within the selected chord.

In both playing modes as described above, users also have the option of selecting whether motions along the "forward" and "backward" (or "down" and "up") direction will be registered. When playing the guitar for example, it may be desired to only strike the strings on the down-stroke. A single button triggers either option while playing: motion in both directions or only motion in the "down" direction will be registered.

All motions are interpreted by Processing, and are sent appropriately to Chuck for analysis and sound production. You will be pleasantly surprised with a slightly more detailed description to come.

4. Loop Mode

In Loop Mode, the user can browse through different audio loops using the Wiimote. The user has full control over each loop, and is really able to "play" each audio file, controlling the tempo, pitch, and sonic qualities of each loop. After the user finds a desired audio loop, the user can then select this audio loop and place it in a mixer channel. Once the desired loop is placed in the mixer channel, the user can then select another open mixer channel and, if desired, find another loop, alter it, and place it in this mixer channel. There are 4 total channels on the mixer, 1 being reserved, however, for the Live instrument. Thus the user is able to select up to 3 different audio loops, and place them each into the mixer channel where the user can then mix them as desired.

Once the loop is placed in the mixer channel, it will continue to play whether or not the user is in Loop Mode or Instrument Mode. In this fashion, the user can select a loop, alter it exactly to the user's likings, drop it in a mixer channel and keep it playing, then switch to Instrument Mode and play, for example, a guitar over a drum loop.

Think about browsing through the given loops, finding a groovy piano lick, setting the desired tempo and pitch for this loop, dropping it in a mixer channel, then going back and finding a slamming bass line and kicking drum groove to complement this piano lick, mixing it as desired, then jamming over your groove with your guitar. The user can truly feel as if he/she is performing in a real rock band.

The controls in Loop Mode are fairly similar to those in Instrument Mode, lowering the amount of confusion for the user in switching between the two modes.

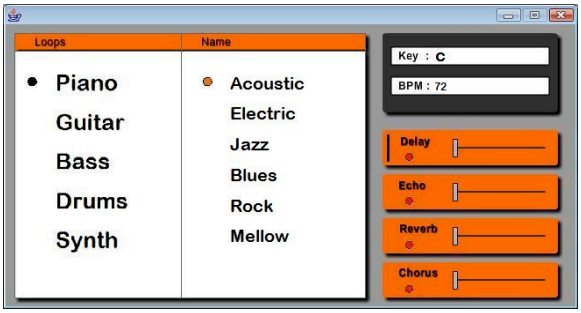


Figure: Loop Mode

4.1 Basic Loop Mode Controls

- **Power** — Turns Wii-Remote on/off
- **Instrument Selection** — Select different instrument loops (Guitar, Drums, Piano)
- **Toggle—** Toggle between Instrument Mode and Loop Mode
- **Tempo and Pitch—** Switch desired tempo and pitch
- **Loop View & Mixer View—** Switch screen between Loop View and Mixer View
- **Mixer Window Lights—**Highlights mixer track selected (Only capable of recording 4 tracks)
- **Record —** Enter Record Mode
- **FX Toggle—** Choose which FX to add
- **FX Send 1—** Choose which FX to send to FX channel1
- **FX Send 2—** Choose which FX to send to FX channel2

Once again, many of the functions here are pretty straightforward. Highlights in Loop Mode include the ability to change tempo and pitch of the loop at the press of a button, the ability to enter the Mixer View (which was previously unavailable in Instrument Mode). As we saw above, the Mixer View is a very powerful aspect of our device. Finally, the user is able to select different effects and add this to different mixer channels.

There also exists a record mode, but in tests, we discovered that this was extremely buggy. When users tried to enter into record mode, the entire program would either crash or recording would simply not be enabled. We would like to look into enhancing this function in future studies.

4.2 Included Loops

The *Acceleguitarometer* comes with over 40 top quality loops from which the user can choose. The loops all range in different sonic qualities and artistic styles. The full list of loops follows. Each loop has 1 different variation to it as well, thus resulting in 40 total loops:

- Guitar: Acoustic, Electric, Jazz, Rock, Pop
- Drums: Rock, Hip-Hop, Jazz, Dance, Funk
- Bass: Acoustic, Electric, Jazz, Rock, Pop
- Piano: Acoustic, Electric, Jazz, Rock, Synth

4.3 Mixer View

As stated above, the Mixer View is what we really believe sets the *Acceleguitarometer* apart from previous virtual instrument related Wiimote research. As far as we know, no virtual instrument that uses the Wiimote has this functionality yet. The mixer is what really emphasizes the virtual

rock-*band* element of the device, as the user is able to surround him/herself with all instruments found in a traditional rock band.

Each channel in the Mixer View shows exactly what loop is being played on that specific channel. And when in Loop Selection Mode, the lights on the Wiimote are highlighted depending on which channel you have selected on the mixer.

Figure: Mixer View

Further, the user has the ability to add up to 2 effects to each mixer channel. This can offer extreme control over the sounds. The effects the user can choose from are reverb, delay, chorus, and echo. Think, for example, of adding just the right amount of reverb: the user could then feel as if he/she is performing at a rock club! The quality and amount of each effect is fully customizable, as these are all created by different UGens in Chuck.

5. Software

One great thing about the *Acceleguitarometer* software is that it is entirely open source. All code will be bundled with the product, allowing the user to fully customize the *Acceleguitarometer* depending on what types of sounds he/she would like to play. All code is done in Processing and Chuck.

5.1 Processing

As described earlier, all motion on the Wiimote and activity on its buttons are captured as OSC messages using Processing.

It took group Fire a substantial amount of time to figure out how to talk with the Wii equipment via Bluetooth; many frustrating hours (days!) were spent developing applications in java using JSR-82 J2SE implementations, such as BlueCove (5), to provide a JAVA API for Bluetooth. All experiments were unsuccessful, and many tense hours (days?) of frantic cursing, Googling, and crying ensued.

OSCP5, the OSC implementation for Processing, and more specifically, the darwiinOSC, package came to the rescue! The software package for OSX includes the Processing "WiiController" class, which provides real-time communications with a WiiRemote and Nunchuck.

WiiController Object

Firstly, the WiiController class was modified slightly for our application needs. The Nunchuck joystick, which previously mapped "x" and "y" positions, is currently mapped as eight distinct buttons.

The WiiController object is sampled at 30 times a second, or the rate equivalent to the interface refresh rate. That is, 30 times every second, the client application verifies if any events have occurred on the Wiimote or Nunchuck (accelerations, tilt, button

presses). This implementation allows for the easy verification of events, such as if a button is *currently* depressed or not. However, this implementation also renders interface navigation particularly tricky. For example, if a user were to hold the "next instrument button" for a full 1 second, that particular event would be invoked at every sample, or the selected instrument would change 30 consecutive times in that second. The WiiController class was therefore also enhanced as to recognize button *pushes*, in which the event is invoked only once when the specific button is depressed.

Track Class

The Track class manages all defined track properties within the application. For both Loop and Instrument modes, users select and specify from the following options:

- instrument family, instrument, music key, beats per minute, level of volume, reverb, echo, chorus, and delay

There exist instance methods for appropriate control over these variables.

Strum Class

The Strum class manages the Wiimote's acceleration and pitch values and detects when each event occurs.

Distinct "up" and "down" movements will trigger a strum pattern threshold when performed with a certain force. An array of acceleration values is stored and updated at every sampling of the WiiController object, in which the ensuing strum pattern is analyzed. The maximum and minimum peaks are detected, and an appropriate volume is set based on the absolute distance between the two values. In this way, the volume for the Live instrument track will be automatically set by the intensity in which the user is strumming. In order to achieve a smoother volume distribution, a rolling average is tabulated for new volume updates; the current volume is set by an average of the current and 5 preceding magnitudes.

Distinct pitch rotations will also trigger individual note thresholds when activated. Group Fire currently has enabled a 4 and 6 string (or note within a chord of 4 or 6 notes) implementation, in which each note is defined at a certain pitch of the Wiimote controller. When rotating the Wiimote, if the pitch threshold is surpassed, the corresponding note is triggered.

Based on studies done by T. Maki-Patola and P. Hamalainen, we realized that controlling a musical device in free-space poses many issues, including how much latency can be allowed for control paradigms. In order to provide some feedback to users, we have utilized the vibration-feedback of the Wiimote. When a solid chord or individual note is triggered, a short corresponding burst of vibration is activated. We discuss the implications of our implementation within the testing and results section.

Interface

The *Acceleguitarometer's* graphical user interface appropriately maps the rocking-out options with different buttons of the Wiimote and Nunchuck. Each feature within the Mixer view and Loop view (and the ability to switch between the two views) are mapped to buttons on the Wiimote and Nunchuck. Further analysis of this setup is also provided in the testing and results sections.

5.2 Chuck

The information is then sent to Chuck from Processing using an OSC send and receive object. Based on Dan Trueman's use of the OSC object in his Cyclotron,[6] communication between the two programs is made very simple and smooth this way. Once chuck receives an event, it then sporks the necessary shred. For example, if the + button on the Wiimote is pressed, Processing sends a message to Chuck indicating that the + button is down, and as a result, the tempo should be faster. Tempo changes in loop mode are controlled by chucking different amounts of time to *buf.rate*. Further, we are using the PitShift unit generator in Chuck to change the desired pitch. As a result, our sound starts to get distorted and fairly ugly as the user starts making changes to both tempo and pitch. In our tests, we started to realize a noticeable difference in sound at the interval of about a 4^{th} and at about a 20% difference in tempo. In future studies, we would like to improve this function to allow for smoother tempo and pitch changes.

In Instrument Mode, we make use of many of the STK instrument ugens which we familiarized ourselves with from the Chuck program manual.[7] In Loop Mode we make heavy use of wav samples, which are then chucked to the *buf*.

6. Testing

In order to evaluate the usability and functionality of the *Acceleguitarometer*, Group Fire structured a thorough and rigorous testing program. More specifically, we first measured the navigability of our user interface design and corresponding button mappings. Secondly, we measured the control in which users exerted while playing the live instrument track.

Sample Population

We sampled a total of 20 people for testing, broken down by the demographics shown in figure table 6.1. We tried to recruit a nice variety of people with different life experiences.

Test 1

Our first test measured how accurate and speedy our subjects could implement defined settings within our program, given two different navigation options. The first navigation option (mode "A") uses a minimal amount of buttons; that is, the user must navigate to each setting using only "left" and "right" and then using "up" and "down" to change the setting. The second navigation option (mode "B") used many buttons mapped to different settings; that is, "left" and "right" changed the instrument,

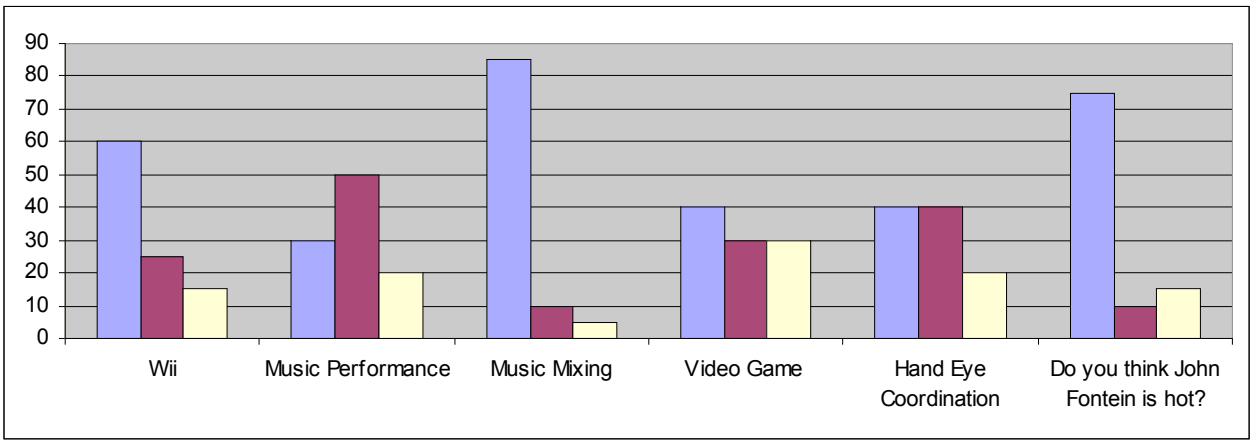


Figure 6.1 ***Percentage of users who responded: "very little" (blue), "a bit" (red) and "tons" (white).***

"plus" and "minus" changed beats per minute, etc.

The subjects were then given a defined set of parameters, in which a typical *AcceleguitaroMaster* may wish to jam with, for each track: "Guitar, acoustic, Key of C, 120 beats per minute, Reverb: 20, Chorus 10, Echo 0, Delay 3". Each set comprised of definitions for all 3 Loop mode tracks.

We timed the performance of each subject given the following conditions:

1. The subject is given a brief 30 second introduction to the product. He or she has never previously touched the *Acceleguitarometer* or navigated the interface. We randomized the sample so that one group navigated the interface using the navigation mode "A" first, followed by "B", and the other group navigated the interface using mode "B" then "A".

2. The subject is given 2 minutes to become accustomed to the device and explore the interface in each navigation mode. The test is repeated for each navigation mode.

The results are shown in figure 6.2.

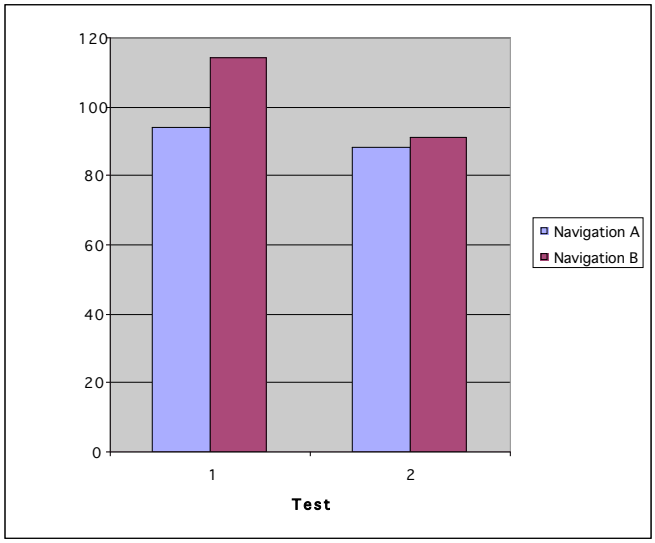


Figure 6.2 ***Time (in seconds) to complete the task for each navigation mode***

Test 2

Our second test measured how much control the users perceive to have when playing the live instrument.

We asked the user to select his or her favorite instrument, and then gave a set of directions, from the following list:

1. Play loudly;

2. Play softly;

3. Play only the top string a bunch of times

4. Play only the 3, 4 or 5 string a bunch

5. Play any chord, alternating between down stroke only and up and down stroke.

6. Play chord C! Play chord B! Play chord D! Play chord A! Play chord C! G! C! D! G!

These commands were given when both the "vibration" mode (vibration feedback is provided when a string or chord is struck) and non-vibration mode. Comments by the subjects were then recorded to evaluate their perceived performance.

7. Results

Based on our tests, we believe that the *Acceleguitarometer* is a success!

As we see above, our first observation is that while navigation in mode B (in which many buttons are used to navigate the interface) may initially be trickier to use, they achieved much better performance times when given time to familiarize themselves with the interface. This is a very promising result.

We would even predict that given a longer training period, performance in navigation mode B would be better than performance in navigation mode A.

Even after a few short minutes of learning the interface, our results show that *AcceleguitaroMasters* can easily and quickly navigate through different track settings. A goal of the project was to ensure users would have the chance to rock-out and jam in a real rock-band environment; the quicker they can change the feel of the music according to their taste and adjust different settings, the more realistic that environment will be.

Secondly, a lot of people had a lot of fun! We learned a lot by observing out subjects as they played the device on their own, and when following our orders. While playing solid form chords appears very straightforward and very successful, playing individual notes is difficult. The vibration feedback does help to identify where notes are at very slow speeds; however, when tilting through all notes of a chord, the feedback becomes slow and quickly falls out of sync. An interesting note from a subject gave us the idea of implementing a visual aid for better registering notes using tilt. By notifying the *AcceleguitaroMaster* where the current pitch is in relation to where the notes are within the range, it may be easier to select an individual note. We may implement this in future versions of our product.

A few comments we noted during test 2:

1. "Damn you string, play!"

2. "Woooo!"

3. "Sonafabitch that's hard"

4. "No John, I will not go out with you"

5. "LookitmeimaRocksTAAAR!!"

6. "Bum ba baaah, bum bah bah bah!

7. "Wait, who teaches this class? That's awesome; I'm so taking this next year"

8. Future Areas of Focus

Although the *Acceleguitarometer* clearly promises to be a very exciting new device, as it is still in its first stages, there are many improvements that need to be made in future research of the device.

First, it would be interesting to tap into the speaker function of the Wiimote, thereby eliminating the need of an external amp. This could make it more of an "acoustic" type instrument with its own "resonance," and could surely offer new grounds for musical expression.

Further, the use of 3rd axis of the Wiimote's accelerometer has yet to be mapped to a function. Although during our tests, many users said they thought there were, if anything, too many functions on the Wiimote, we believe that it is important to tap into the full potential of the Wiimote. Future projects include adding more functionality and more musical expression by exploring possible uses of the 3rd axis.

Further research and development may lead to better model the physical instruments available to *AcceleguitaroMasters.* For example, a "hammy" effect could be triggered for the guitar preset when some motion is detected, as if a real hammy bar were present. Ideally, while many of these gestures would be preset for each instrument, a training component would allow for users to develop their own gestures that would trigger specific events while rocking out.

Recording was giving us trouble throughout our experiments, constantly crashing our computer. We are currently trying to record using Processing instead of Chuck. Perhaps future studies could look at the possibility of using Chuck to record instead.

Finally, once the record mode is fully functional, it would beg for an export function, where a user could record his/her own jam session and immediately export it on the spot for future listening.

9. Conclusion

Musicians are constantly searching for new means of expressing their musical ideas. What place does the *Acceleguitarometer* have in the music world? Will musicians one day turn to the *Acceleguitarometer* for musical inspiration? It's hard to say. The device clearly needs improvement, but it surely presents itself as a viable option for musical expression.

As Brandon Boyd once said, "I'll make music, whether or not anyone is listening, for the rest of my life. It's the most natural form of expression for me, the same way I draw and write and sing." [8]

Perhaps the greatest feat of the *Acceleguitarometer* is that, regardless of whether or not it is taken as a serious musical instrument by professional musicians, it still provides even the most musically untalented individuals, individuals who previously never had the joy of experiencing what Boyd refers to as the most natural form of expression, the opportunity to make music for the rest of their lives. This in itself seems to make the *Acceleguitarometer* a great success.

Bibliography

1. http://en.wikipedia.org/wiki/Image:Wii_Remote_Image.jpg

2. http://www.wiili.org/forum/wiinsaber-(current-version-v11)-t436.html

3 http://www.cs.cmu.edu/~johnny/projects/wii/

4. http://www.cs.cmu.edu/~johnny/projects/wi/

5. http://pakl.net/wii/

6. http://music.princeton.edu/%7Edan/Cyclotron/index.html

7. http://chuck.cs.princeton.edu/

8.http://www.brainyquote.com/quotes/quotes/b/brandonboy181086.html

References

DarwiinRemote

(**http://sourceforge.net/projects/darwiin-remote/**)

WiiLi

(http://www.wiili.org/index.php/Main_Page)

LibWiimote (http://sourceforge.net/projects/libwiimote/)

Processing (**http://www.processing.org**)

Chuck (**http://chuck.cs.princeton.edu/**)

T. M¨aki-Patola and P. H¨am¨al¨ainen, "Latency Tolerance for Gesture Controlled Continuous Sound Instrument Without Tactile Feedback," *Proc. Int. Computer Music Conf.*, Miami, USA, Oct. 2004.

T. M¨aki-Patola and P. H¨am¨al¨ainen, "Effect of Latency on Playing Accuracy of Two Continuous Sound Instruments Without Tactile Feedback," *Proc. DigitalAudio Effects Conference* (DAFx'04), Naples, Italy, Oct. 2004.

Acknowledgements

- **Cook, Perry**
- **Fiebrink, Rebecca**
- **Holio, Corn**
- **Plork**
- **Chuck**
- **STK**
- **Jesus**