

MMI 593 Project Proposal

University of Miami

Mission:

I want to synthesize audio using an intuitive accelerometer-based controller, and an iPhone.

The Problem:

Digital, or computer music has been a rising form of artistic expression for several years now. It seems, however, that there has yet to have been developed a simple, intuitive controller that a user can interact with to directly synthesize sounds in a logical and intuitive fashion. With a bit of programming and some hacking of hardware, I will create this interface. The finished product will be a Nunchuck-controlled audio synthesis program that runs on an iPhone, and is completely controlled with the Wii Nunchuck.

How I plan to complete the project:

Taking advantage of the electronic components and sleek form-factor of the Wii Nunchuck, an interface will be constructed which takes accelerometer data in from the Nunchuck and uses it to synthesize audio through an iPhone by swinging the controller around and pressing various buttons. In total, the Wii Nunchuck features two buttons, a 2-axis joystick, and a 3-axis accelerometer. These sensors will provide ample resources for a user to enable and disable sounds, change pitches, and possibly control many other aspects of the sound synthesis.

The Nunchuck's sensory information will be sent to an Arduino micro-controller, which will most-likely serve as a bus for receiving input signals from the controller, and send the data right back out to be harnessed to synthesize audio. The signals from the Arduino will be sent to an iPhone, on which an application that expects the input signals sent from the Nunchuck-controlled Arduino will respond to the various input streams from the controller. This application will be the heart and soul of the audio synthesis, and will ideally be playable through either the iPhone speakers or the 3mm jack.

Furthermore, secondary controls could be incorporated into the Nunchuck, allowing for tone control/manipulation of output spectrums other than pitch. Other controllable attributes are limited only to the signal-processing flexibility of the synthesis environment (in this case, the iPhone SDK and other audio processing frameworks). Possible tone control options might include time-based effects such as a Wah-wah or reson-filter.

Components:

Item	Price	Link
iPhone	Too much	AT&T
Wii Nunchuck	\$18.99	Amazon
Wii Nunchuck Breakout	\$2.95	SparkFun

Arduino Duemilanove	\$29.95	SparkFun
30-Pin Connector	\$4.95	SparkFun
Housing for Arduino	\$9.95	SparkFun
Total (without iPhone):	\$66.79	

Personal Computing/Embedded Systems:

My device fits the frame of a personal computing device by its very definition: the Wii Nunchuck iPhone controller connects a user to the digital world of audio synthesis through the analog means of flailing your arms around like a fool. The nature of using the controller bypasses all technical tasks and methods, and allows the user to interact with the device without understanding or even knowing that there is work being done behind the scenes. This fact reinforces how this device will function as an embedded system as well. Ideally, all of the hardware used in translating the signal from the Nunchuck to the iPhone will fit in a sleek and compact enclosure, suited well for an average sized pocket.

Existing Projects:

There are a number of projects that take advantage of the Wii remotes and the ensuing capabilities of controlling accessory devices. Many of them are linked [here](#). I intend to utilize one of several scripts that I've found which aid in linking the Arduino's serial output to Xcode. A script that manages communication between the micro controller and my computer will aid greatly in mastering the controlled usage of the data stream coming in from the Wii Nunchuck.

I also intend to step on the shoulders of existing projects on the iPhone that synthesize audio onboard. There aren't a ton of examples with source code available, but I have managed to scrape up a few examples of basic tone generation from web searches and friends in the biz. With these resources, tackling the rest of the project will be significantly easier than starting from scratch.

The Saddest Panda:

I foresee the most difficult part of this project in transmitting intelligible data packets to the iPhone from the Arduino. Getting data from the Wii Nunchuck into the Arduino is a simple matter, but bridging the gap between the open-source Java Platform of Arduino and the closed world of Apple will take some serious hacking and foresight. A transmission line between the micro controller and the iPhone will need to be established, and this will manifest itself in one of several ways. If there exists an API for communication through the 30-pin connector of the iPhone, we're in business. Alternatively, a custom protocol could be written to send data packets through the Line-in channel via the 3mm jack on the iPhone. This would require significantly more work in testing and design, and will remain as a backup plan if an API does not pan out.

Sketches:

Refer to the pages attached for mock-ups of the idealized product that this project will yield.