Sound Design Mid-Term: Polyphonic Synthesizer w/ Multiple Oscillators

Lysha Smith

Sound Design: Jim Mahoney
Submitted: March 26th, 2017

A brief paper explaining my midterm project.

For my midterm I set out to employ all that I have learned so far this semester about synthesis generation within pure-data to create a simple additive based synthesizer. I decided to model the synthesizer after one that I own, so as to best be able to determine how successful (or not) my emulation ended up being. I settled on the Arturia Mini-Brute, which is a monophonic additive synthesizer. As with most such synthesizers, there is also a level of subtractive synthesis that is present, one the combined waveforms hit the filter. This point is something I have come to better understand through this effort. The additive bit is essentially just the convolution of various simple waveforms to create an intricate waveform, quite often this is done with just various phased sine waves. In the case of the Mini-Brute, and in turn my own "Fake-Brute," these multiple waveforms are not just sine waves. Besides the basics of the synthesis functionality, I also wanted to make the synth very playable and thus spent a good bit of time and energy on MIDI implementation, mapping the final synth to a MIDI keyboard.

Where the Mini-Brute has a sub-oscillator (with selectable waveforms), a square wave, triangle wave, sawtooth wave and white noise oscillators (as well as an audio input mix), the Fake-Brute would have a sine wave, triangle wave, square wave, pulse wave, sawtooth wave and noise oscillators. It is of course ultra simple to create a few of these oscillators, namely the sine, sawtooth, and noise waveforms as they are the building blocks of the others and I have used them regularly thus far with PD and have a good grasp on what they do, how to use and manipulate them and shape them. The new wave forms: triangle, square wave and the pulse wave modulation bit took a good bit more doing and research.

Let's start with the square wave. Conceptually, the square wave is actually a pretty simple thing consisting of two sawtooth waves phased so that there is no time between the peaks a troughs of the wave form. With a bit of filtering square-waves are good for creating bass tones and with some pulse width modulation can create some really interesting tones. My version of the square wave oscillator is ultra simple, and in the end I had some difficulty getting the pulse width modulator to actually function as it should, or at least as it does with the Mini-Brute. I

think with a bit more time and effort I could have achieved this, but I decided to move on and focus on other bits. The triangle wave was also pretty straight forward, at least to create the simplified version that I came up with. The interesting thing I learned is that both the square wave and triangle wave are similar/related in that they both contain only odd harmonics. To my ear it the relationship is audible and I had always wondered why this was.

Now that I had these oscillators setup, it was time to get them playable with the keyboard before moving on with other features. As I mentioned the Mini-Brute is monophonic, which is great for lead or bass sounds, and some arpeggiations as well as squeaky blips and bleeps, but not as good for creating lush pad or string sounds, which tend to be the types of sounds I am most attracted to. Thus, it became important for me to create a polyphonic synthesizer. In the main patch I was able to receive MIDI notes with the "notein" object. From there, I used the "poly" object and set it to have 4 voices, so that full chords could be sounded. This then needed to be packed and each voice routed to a sub-patch called "note," which essentially routes the key played on the MIDI keyboard to each oscillator in the correct pitch. When unpacked in the "note" sub-patch the MIDI signal was routed to a "mtof" object which transposes a MIDI value into a frequency in Hertz, and is such that the value is in the usual 12-tone, equal tempered, Western scale as with the piano. Interestingly enough, this can be use to create a microtonal scale/instrument, which is something I may look into in the future, as I really enjoy instruments that have that capability. Once passed through the various oscillators, the signals were sent through back to the main patch.

At this point I had just a mess of sound, and so I needed to be able to mix the various waveforms as I saw fit so as to create a playable complex waveform. The MIDI keyboard I used is the Novation Launchkey, which excellently has various knobs and sliders. These all have their own distinct control change (CC) value and, with the "ctlin" object in PD I was easily able to map the volume of the various oscillators to their own slider. Thus, the patch is able to additively create a complex waveform.

From here I wanted to create an envelope, so that I could have more control with how the waveform would sound. I created a basic amplitude ADSR (attack, decay, sustain, release) envelope. The Mini-Brute has an envelope for both the amplitude as well as the filter, but I opted to skip the filter bit, in truth on most synths I have used in the past I can't really get much more than very subtle results with the filter ADSR. For this, I really had to rely heavily on a tutorial I found online (listed in bibliography below), as it involved "throwing" signals with the "s" and "r" objects/functions in PD, as well as the various ways in which to ramp and delay functions worked with triggers and bangs to produce the envelope(s). While I was able to get things working properly, I think I will be revisiting this a good bit as things unfold as my understanding is surely short of mastery.

At this point I was able to produce some nice sounds, but wanted more control and polish. I eventually added a low pass filter right before the ~dac, but I wanted to learn about some other types of filtering (aside from lp and high pass filters that is). I added a VCF (voltage controlled filter) to the synth, mostly because I knew very little, nor have played much with one before in the past. The advantage of the VCF is that it has four different parameters (start, end, time, and Q) which I then mapped to other sliders on the MIDI keyboard with great success. I should note at this point that (as you know) I needed a bit of a nudge to grasp how to best setup the parameters for the various sliders so that they functioned smoothly. The thing that caught we was the logarithmic vs linear values of the sliders, as well as setting up the appropriate min/max for each slider. Once I had these in line the sliders all worked well.

I added a couple more features, one is a reverb that I created in a previous patch I created that seems to function somewhat capably, and is centered around the "freeverb" object, which was something that was in PD extended but I was able to add. It has a nice freeze function (which I mapped to a button on the MIDI keyboard) that is quite useful, but I want to go further with understanding how the "freeverb" object itself was created and how it functions so as to eventually make one myself that has parameters that I find more useful, particularly a more

useful room size and decay time modes. I also added a graphic display of the wave. I did so before the reverb and low pass filter (but after the VCF, which all the waves go through in the "note" subpatch). I wanted to get a sense of what the basic complex wave was before these bits of processing.

As with much of the work this semester, I relied mostly on tutorials that I found online, as well as Farnell's text as the resources for this project. I feel pleased with the final result to an extent, though there is plenty more I could do to make it even better; however it was primarily an exercise in trying to emulate a synth that already exists, so I hope to try and come up with new ideas for synthesis so as to have in my arsenal new sounds and techniques. As for the rest of the semester, I think I am most interested in focusing on learning the basics of visualization capability in PD, as well as crafting patches that work with realtime audio, so as to be used in performative environments. I do hope to go further with understanding the basics of synthesis, but these avenues are of most interest to me.

Bibliography

Farnell, Andy. *Designing Sound | The MIT Press*. MIT Press. Accessed March 27, 2017.

"/Chapter: Additive-Synthesis / PURE DATA." *FLOSS Manuals*. Accessed March 6, 2017. http://write.flossmanuals.net/pure-data/additive-synthesis/

"/Chapter: Square-Waves / PURE DATA." *FLOSS Manuals*. Accessed March 27, 2017. http://write.flossmanuals.net/pure-data/square-waves/

"4.3 HIDs." Accessed March 6, 2017. http://pd-tutorial.com/english/ch04s03.html

"Creating a Simple Synthesizer in Pure Data – Part II | Libre Music Production." Accessed March 27, 2017. http://libremusicproduction.com/tutorials/creating-simple-synthesizer-pure-data-%E2%80%93-part-ii

"What MIDI CC Messages Do the Controls on the Launchkey Send?" *Novation*. Accessed March 6, 2017. http://support.novationmusic.com/hc/en-gb/articles/207561095-What-MIDI-CC-Messages-do-the-controls-on-the-Launchkey-Send-