

## An Exploration of Algorithmic Composition: Technical Details

To develop this piece I utilized a few different technologies, some in more typical contexts than others: I composed MIDI with Logic Pro, developed JavaScript scripts that run within a Max/MSP patch to generate melodies, and routed MIDI through these programs and into MainStage to allow flexibility for real-time performance. I will attempt to focus this discussion on the less typical uses of technology as well as provide an overview of my development environment in hope that someone who would like to do something similar could benefit from this information.

The idea for this composition began with a basic version of what now is the “central melody” that I developed on my bass guitar and then in Logic’s “Piano Roll” interface. I have since developed this melody, as well as algorithmically generated accompaniments with the intention of eventually performing this piece. Because the real-time generated accompaniments can be based on the notes I play in real-time, my bass must be the input to the Max/MSP patch that handles all of this computation. In order to develop the central melody that I will be playing on bass, I routed MIDI from Logic into Max/MSP<sup>1</sup>. This allowed me to compose the melody in Logic and have it analyzed by Max just as if I was actively performing it (and if my MIDI pickup tracked perfectly of course).

When developing this central melody, I set up a small development environment in Max/MSP that allowed me to quickly hear the melody as it was being algorithmically generated. In this environment it was easy for me to get immediate feedback on the results of my algorithms because I was able to listen to all of the renderings when experimenting with the manipulation of a parameter like the pitch set<sup>2</sup>, for example. This rapid “development” environment helped me to focus on the theory behind the composition and less on the details of the implementation, i.e. writing the actual notes. Once the central melody was composed, I was able to simply play it from Logic (as mentioned above), and continue to develop the accompaniments from Max.

I used Max/MSP to generate all MIDI which is fed to MainStage which hosts the synthesis and sampling. This MIDI is generated using typical Max data processing/routing, as well as scripts that run within “js” objects in Max. Currently, these algorithms all take as input the midi of the “drone” voice and/or the central melody. The drone could have easily been generated programmatically but it was one of the first voices that I implemented and I

---

<sup>1</sup> MIDI routing was accomplished with Max/MSP’s built in MIDI ports, or MIDIPipe <<http://web.mac.com/nicowald/SubtleSoft/MidiPipe.html>>

<sup>2</sup> Experimentation with pitch sets was a significant portion of my work for this iteration of the piece. See my paper “An Exploration of Algorithmic Composition via the Fibonacci Sequence” <<http://colin-sullivan.com/main/archives/944>> for more detail.

simply did not get around to programming it. In my Max/MSP patch, the “global Fibonacci number”<sup>3</sup> is determined based on an analysis of the amount of notes played on the drone. This number is then used throughout the patch in different ways to generate MIDI<sup>4</sup>.

I found that when developing algorithmic rhythms, using the signal routing paradigm found within Max/MSP worked extremely well, but when attempting to incorporate melodies, node-based programming made it very difficult to manage all of the aspects of the algorithm. It is for this reason that I began to write JavaScript scripts and run them from within Max/MSP. The imperative control flow was much more reasonable when attempting to algorithmically determine both a pitch and a rhythm, and is how I generated all of the melodic material in the piece.

As I mentioned above, I fed the MIDI from Max/MSP into MainStage so that the environment remained applicable in real-time and could be performed with easily. Therefore, all of the sounds were sampled, synthesized and effected from within MainStage. For the drone timbre, I used a fairly complex synthesizer that allowed for a sample to be used as an oscillator component. This is the distorted “click” sound that is increasingly audible as the piece progresses. The remainder of the drone timbre is sine and saw oscillators that are modulated and filtered in various ways to produce a “buzzy” tone, and a sharp repeating amplitude envelope that produces the actual variation in volume for the buzz effect. Various other parameters are manipulated as the piece progresses to make the drone slowly morph into a much more chaotic timbre, then back again.

Since I had performance in mind throughout the development of this piece, I did not want to hard code the variation in these parameters. Therefore, I mapped most parameters to the same two “pedals” in MainStage (which are actually just controlled by Logic at the moment). This allows me to instantly map all of the same parameters to an actual foot pedal when I perform the piece. The MIDI value of either pedal increases at a slightly different rate as the piece progresses, which is something I could easily manipulate with two physical pedals as I perform.

Another voice that is worth discussing is the timbre of the central melody (and the corresponding echoing timbres). These voices were all developed using Apple’s “Sculpture” synth, which allows for some extremely granular control over the parameters of a physical modeling synthesizer. Originally, I had only decided to use this voice because I did not have a recording of my actual electric bass timbre, but I have since developed this timbre and am beginning to enjoy the synthesis. I intend to incorporate this synth into the performance of the piece in the future.

---

<sup>3</sup> The global Fibonacci number is a significant construct in the piece, and is used to determine many other aspects. For a more precise definition, see the larger paper<sup>1</sup>.

<sup>4</sup> For more specific information on how the MIDI is generated, see my composition paper<sup>1</sup>.

I would like to continue the development of the technical side of this composition. I have experimented with digital signal processing in the past<sup>5</sup>, and I would very much like to develop a digital audio effect where I can utilize the Fibonacci sequence on the sample scale and incorporate this effect into the piece. There are some technical limitations I encountered when attempting to manipulate Max's transport tempo which I hope to overcome in the future, as I feel that the structure of this piece could really benefit from a variable tempo. Overall, utilizing these technologies for the development of my piece was a fairly painless process, and I hope to continue my exploration of these technologies in the future.

---

<sup>5</sup> <http://colin-sullivan.com/main/archives/343>