system (essentially, you have to watch that the derivative of the functions do not get too great).

4. Frequency Modulation 1 - 5:02 - This is perhaps my favorite track on the CD. This again was created in Perl, with functions of the form

$$f(t) = \sin(a_1 t + b_1 \sin(a_2 t + b_2 \sin(a_3 t + b_3 \sin(a_4 t + b_4 \sin(\ldots)))))$$

which goes twenty deep. The function you hear in this track changes periodically, more and more rapidly until the last 90 seconds or so when it changes at a constant rate. I really like the types of sounds this method generates, though I haven't played with it long enough to find anything that changes nicely over time by itself.

5. Modular Arithmetic, Mostly 1 - 3:20 - This was also created in Perl. The idea was to use no standard oscillator-like components, like sinusoidal functions. This was created entirely with integer operations. Essentially the i-th sample in the track is determined by testing i as an integer against various moduli and conditions which themselves change as the track progresses. I'm really happy with the unexpected nature of this track and some of the timbres are just great, and particularly the change around 31 seconds in.

6. All Together - 8:00 - Created with Csound (QuteCsound actually). I had been really enjoying the sounds I was getting in QuteCound just by letting five or six oscillators interfere with each other, so I set up a bunch of oscillators with linearly changing frequencies interfere with each other, mixing and combining them in various ways and let it go. This was the last track I made this year, and I probably made it a little longer than it needs to be but I like a lot of the timbres, and the blatant "electronic" feel of it.

7. Sines Inside Sines 2 - 5:00 - Another like track 3, but with a much more mellow feel. Same function type, with different coefficients and more of them.

Matthew Conroy

doctormatt@madandmoonly.com

www.matthewconroy.com

RPM 2012

Matthew Conroy

All tracks recorded during February 2012 for the RPM Challenge 2012.

I was really busy this February, with drawing, cycling, and too many cheating students in my classes, so my ideas were pretty scattered. I think this was successful, though, in the fact that I got a few new techniques tried, and an old one or two expanded on.

1. Temporary Rhythms 2 - 5:10 - This is a followup to last year's Temporary Rhythms. The idea is the same: a set of random rhythms (on a 16 beat measure) are generated, and gradually combined in different combinations and with different "instruments". For this one, I also layered half-speed and quarter-speed versions of the rhythms, too, so it's nice and thick, and it gets thicker as the track progresses, but I wish I'd pushed it a bit more. Next year I think I'll try something similar, but with more extreme sounds.

2. Granular 1 - 8:20 - This is a pile of one-wavelength sine wave sound "grains", thrown out on the timeline with varying densities and varying pitch centers. I was hoping for a little more variety in the sound, but I think the rain-like nature and the roaring bass parts are good.

3. Sines Inside Sines 1 - 1:00 - A very old idea which I have recently come back to. I recently found how to write sound files with a Perl script, and this allows me total control of the final result, plus it gives me access to proper trigonometric functions (which I have had problems with in Csound). So, the idea is simple: create a sound file from a straight-up mathematical function. For this track, I define three functions of the form

$$f_i(t) = \sin\left(\frac{\sin(at)\sin(bt) + \sin(ct)\sin(dt)}{r + \sin(et)\sin(ft) + \sin(gt)\sin(ht)}\right)$$

and then the output waveform is essentially $f(t) = f_1(t) f_2(t) f_3(t)$.

This track turned out nicely crunchy. The challenge with this kind of stuff is to get the sounds to be interesting without going beyond the limits of the 44100 samples-per-second