Julay Leatherman-Brooks | julay@pdx.edu | **rust-tones**

TOPIC AREA

I will be building a synthesizer in Rust. I am starting out knowing almost nothing of what this entails. What are the elements common to hardware and software synths? What elements are specific to software synths? What are the pieces of the problem?

I found a discussion on the DSP stack exchange...

"the various synthesizer building blocks (e.g. oscillators, filters, noise generators, envelope sharers, etc) ..." -Paul R, https://dsp.stackexchange.com/questions/192/what-does-one-need-to-know-in-order-to-build-a-software-synthesizer. Thanks, Paul R! Now I know what one person thinks the basic components are of the thing I'm going to build.

Paul R, again, from the same discussion: "...and then implement a way of 'wiring' these together and controlling their parameters."

The crates suggested by Bart were, it seems, mostly geared toward performing I/O (connecting with keyboard and speakers) and using the MIDI protocol. This is the vaguest part for me. I sort of get sound waves -- it's just going to be mathematical representations of the basic wave types (sine, square, triangle, sawtooth) and mathematical transformations of various aspects of these to control tone, timbre, duration, pitch, etc, once initial pitch and duration information have been logged from a key press. But I definitely don't understand yet how to glue it all together and make a thing that works. I only have the fuzziest idea of what MIDI even is (I just know what people's MIDI music sounds like).

It might be interesting to read Xenakis's book and try and incorporate some stochasticity into the sound shaping. In the first couple of pages he is already very dismissive of the Fourier transform.

Issues of concern: I am often intimidated at the start of a project, and looking at the crates and the code Bart has for his synthesizer project, I am feeling that. I need to really do the thing where you break a problem down into its tiniest, most actionable component parts. Which is hard when you don't feel like you've sufficiently mapped the problem space yet and defined your little corner of it. This is something I'll want help with at the very start: defining exactly what I will do and what I will leave out, and breaking down what I will do into a series of subtasks.

Project vision: Knowing the little I know, I would like to have a program that links with a keyboard and speakers, and has user settings to control various aspects of the sound. These should be able to be saved in memory as .wav files, so that you could throw them into audio software and mess with them further, but what I'm mainly interested in doing here is creating a rudimentary digital musical instrument.