Synthesizer Keyboard

Does not actually synthesize keyboards

This program was a spinoff of the PCM Encoder program. I realized that I had a wave generator with frequency/amplitude controls so the next logical step was to turn that into audible feedback. Plus I had never worked with audio before so it sounded fun!

So I learned about generating audio in Java, mostly by reading parts of this tutorial. <http://www.developer.com/java/other/article.php/2226701/Java-Sound-Creating-Playing-and-Saving-Synthetic-Sounds.htm>

It took some experimenting to figure out how to break my sound wave samples into a byte array and feed them into an audio dataline, but it was so satisfying to hear the first multi-note chord come out of my tinny laptop speakers. It worked!

A friend of mine told me about an interesting musical property he’d recently learned about, regarding equal temperament vs Pythagorean tuning. Basically, equal temperament is how most instruments like pianos are tuned, based on a logarithmic scale, and are defined independently of each other. In Pythagorean tuning, all the notes are based on harmonic ratios relative to a base note. A small example of the difference can be heard here:

<https://www.youtube.com/watch?v=APtJsaPxNgo>

This synth I’ve put together has an interesting property that if you play a chord up from the bottom note it uses exact harmonic ratios to determine the note pitches, whereas if you play a chord down from the top note it will use equal temperament. I used Just tuning (http://www.phy.mtu.edu/~suits/scales.html) rather than true Pythagorean tuning (https://en.wikipedia.org/wiki/Pythagorean\_tuning) because it fit my keyboard range better. This was fun to develop and really interesting to hear the difference between harmonic ratios and equal temperament.

There was one issue I encountered building this that I never fully resolved. There is some latency between pressing the key and hearing the sound. I suspect it has to do with either the buffers I fill, or calls AudioSystem.getline, or maybe I’m creating too many threads. At the time I didn’t understand how to use threads effectively, or the overhead involved. However, it works well enough for what it’s intended to demonstrate. Try it out!