

P5 Mini Synth

This final project was a lot like letting the mold harden. It is not as comprehensive as I'd like, but I've solidified what it is I am capable of doing with this set of skills, and what I can still be much better at.

I designed a rudimentary synthesizer. As of now it is a single oscillator, monophonic keyboard with adjustable parameters for the sounds' envelope and a filter for cutting frequencies. I am happy with the success I had with this project because it functions as a musical tool. It is monophonic, meaning only one note can be played at a time, providing users the ability to find basic melodies on its keyboard, just by clicking on the keys. Sounds offered include long built up sounds, as well as short stabbing ones, thanks to the envelope.

Along the way I had to relearn some concepts I didn't focus on a lot in class, as well as work through a seemingly never-ending sea of bugs. As of now my faders are still buggy. Their y-axis updates beautifully if you start changing values from the bottom of the fader. However, if the slider is in any median, or located on the extreme top of its range it jumps around as you click and drag it.

The knobs presented a challenge in their functionality as well. In the beginning they were rotating with no rhyme or reason. Once I gathered the circular math to get them to limit their rotation they worked as butter. But then their values seemed inverted. If I clicked and drug down on them their value would increase, when I wanted it to go up and vice versa. My solution here was a simple reversal of the knob class's `mouseClicked()` value's mapping. Now the knobs function on a 0-1, top to bottom, scale rather than 1-0.

The class for my keyboard was one of the simpler tasks, including assigning frequency values. The rectangles are all indexed in an array; one for the white keys, one for the black. Arrays of equal length to the key arrays hold all the coinciding frequency values for each key.

I could not find a way to eliminate the two unnecessary black keys to display a traditional piano keyboard. I was running a `visibility if()` function, but could not target the right variable and object values.

Each key on the keyboard is running 3 different functions as users interact with them. The first is to be drawn from the aforementioned array. The second is a function triggered when the user clicks on a key. This action triggers one command for the oscillator telling it to start having amplitude. The third function is executed when the mouse is released by the user. When this happens the amplitude of the oscillator is set back to zero.

I've built a function in which a knob changes the waveform. I have all my waveform shapes indexed in an array. However, when I attempt to pull a value from the knob and assign it as an index the oscillator does not read the index being referenced. I've decided to exclude this feature for now and workshop it in my own time.

Overall, this project is not complete. I am very satisfied with my final product, and just the fact that I got it working, and featured adjustable sound parameters, however elementary they may be. I'm excited to continue working on this to learn more about coding logic, and constructing synthesizers.