

Intro to Audio Programming

Making a simple synthesiser

Skylar Gill

.NET Developer @ Sorted

Background

- Familiar with synths other people have made
- Haven't touched C++ beyond "hello world!" until a few months ago

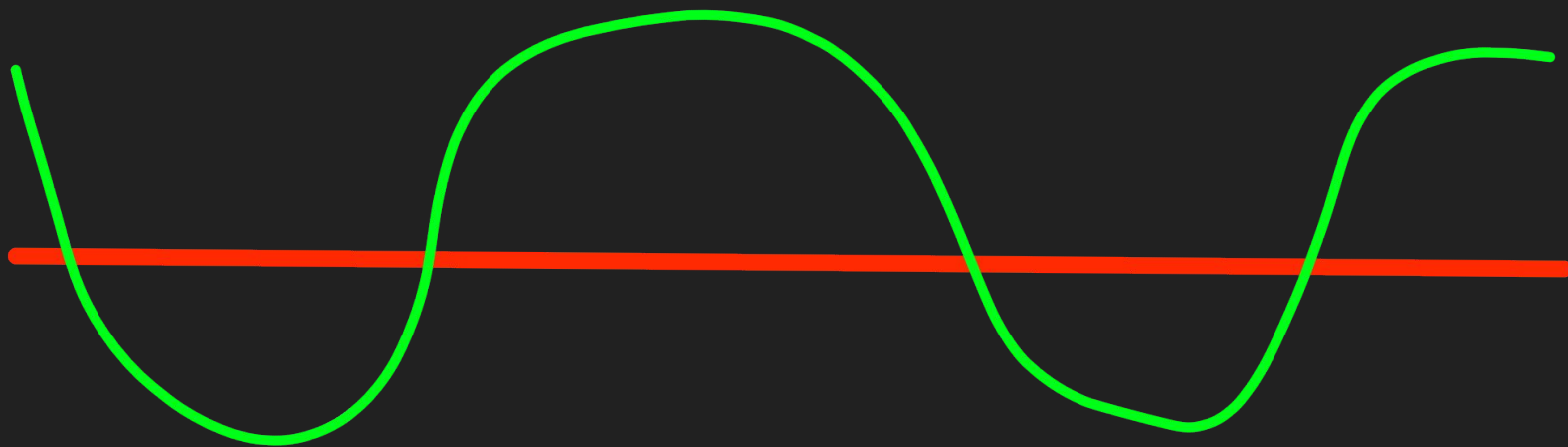
What is a synthesiser?

“machine that electronically generates and modifies sounds”

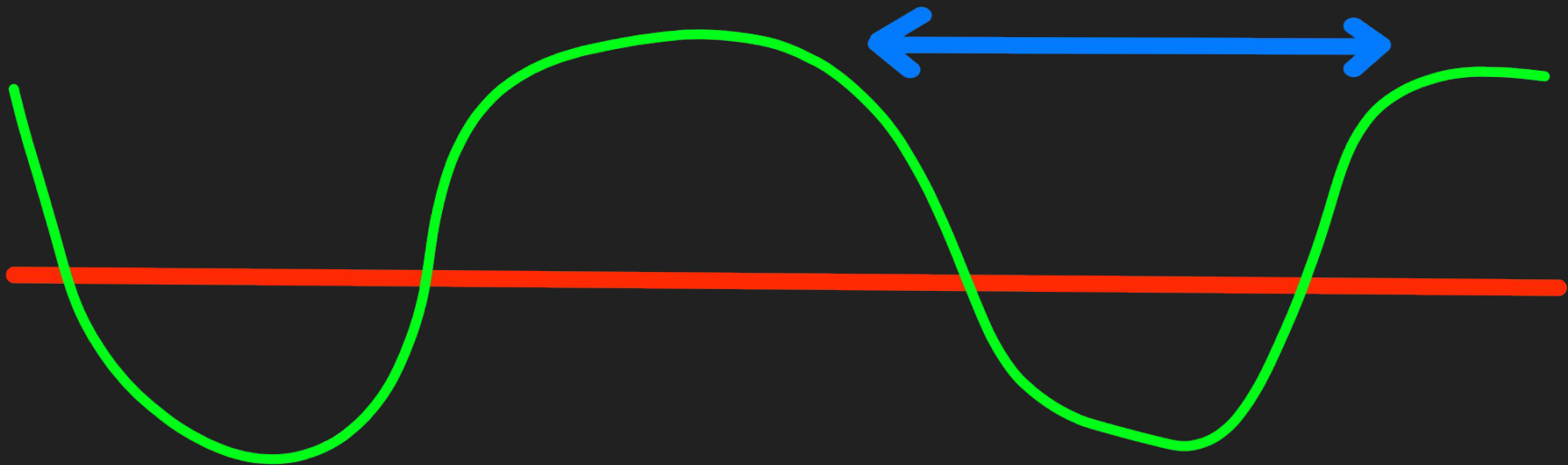
Libraries Used

- JUCE (UI and audio hardware communication)
- DaisySP (audio processing)

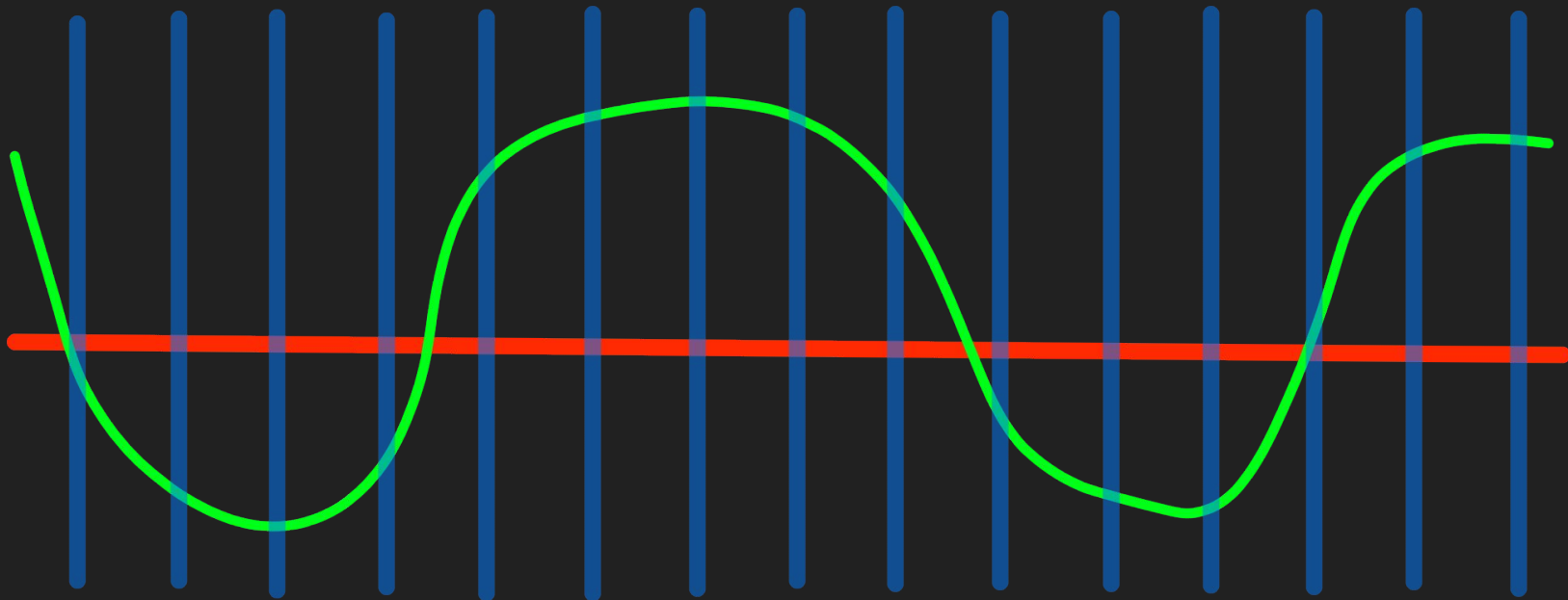
Audio Signals



Wavelength/Frequency



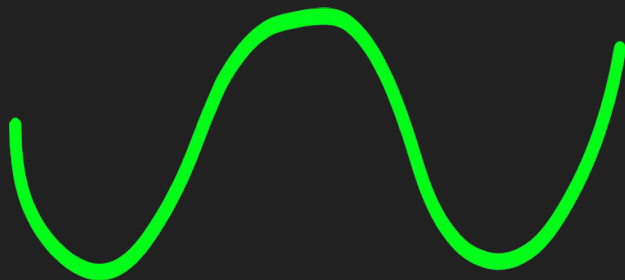
Samples



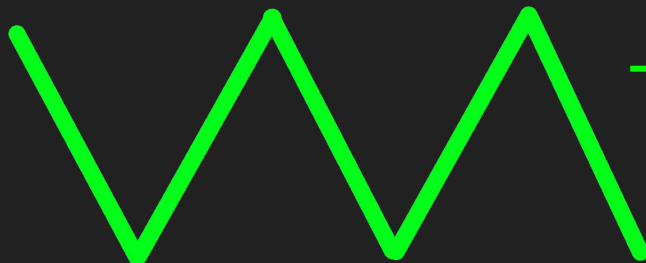
Oscillators

Waveshapes

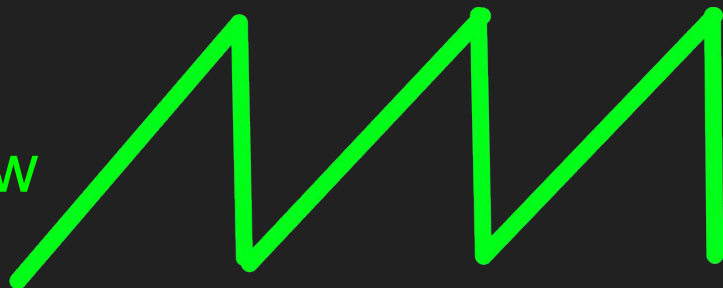
Sin



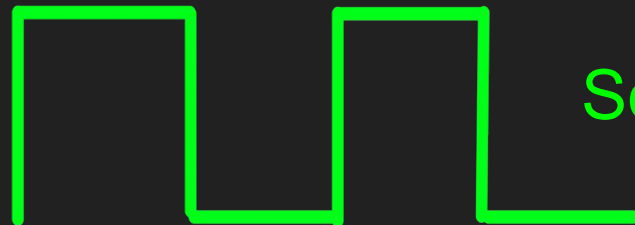
Triangle



Saw



Square



Code to generate a wave

We use an oscillator to produce a wave as seen here

```
//Get sample for each sound source and scale them to their current level
```

```
float oscillator1Value = oscillator1.Process() * osc1Level;
```

Code to generate a wave

We also set the waveform of the wave like this

```
case Parameter::Osc1Waveform:{
    if ((Waveform)value == Waveform::Square){
        oscillator1.SetWaveform( wf: Oscillator::WAVE_SQUARE);
    }
    if ((Waveform)value == Waveform::Saw){
        oscillator1.SetWaveform( wf: Oscillator::WAVE_SAW);
    }
    if ((Waveform)value == Waveform::Sin){
        oscillator1.SetWaveform( wf: Oscillator::WAVE_SIN);
    }
    if ((Waveform)value == Waveform::Triangle){
        oscillator1.SetWaveform( wf: Oscillator::WAVE_TRI);
    }

    break;
}
```

Code to generate a wave

Also set the frequency/pitch when we change notes

```
case Parameter::NoteValue:{  
    float osc1Frequency = mtof(value);  
    oscillator1.SetFreq(osc1Frequency);
```

Mixing multiple waves

Mixing multiple signals is simple as we're only working with one sample at a time

We can just add the floats together and compensate the levels

```
//Get sample for each sound source and scale them to their current level
```

```
float oscillator1Value = oscillator1.Process() * osc1Level;
```

```
float oscillator2Value = oscillator2.Process() * osc2Level;
```

```
float noiseValue = noise.Process() * noiseLevel;
```

```
//Sum the sound source samples to get a mixed signal. Dividing by the number of sources then compensates the level
```

```
float out = (oscillator1Value + oscillator2Value + noiseValue) / 3;
```

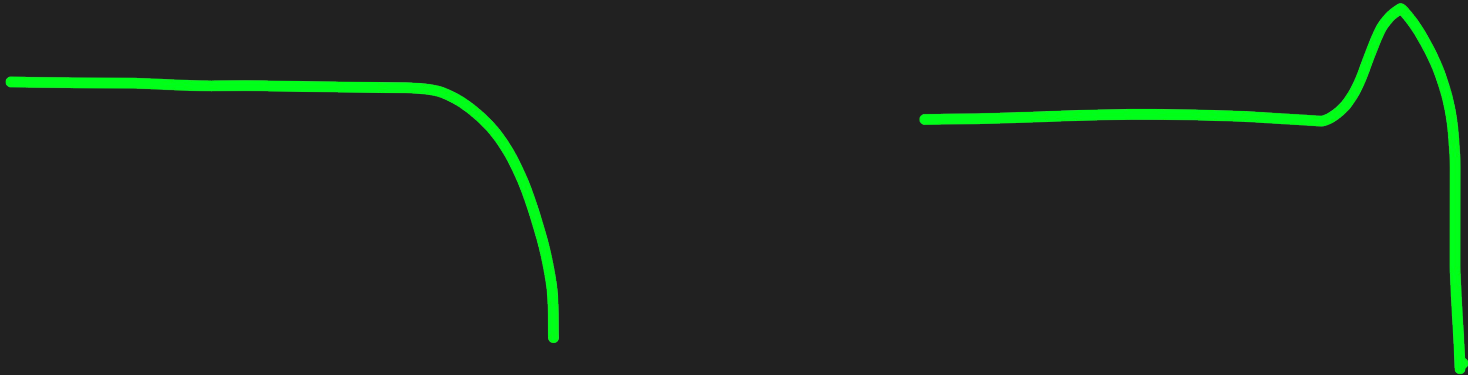
Filters

Filters

Filters can shape the sound we've produced

Low-pass filters remove high frequencies

Resonance can emphasise frequencies the cutoff point



Code for Filter

DaisySP makes this really simple

We just need to give it the sample we're affecting as a parameter

```
out = filter.Process(out);
```

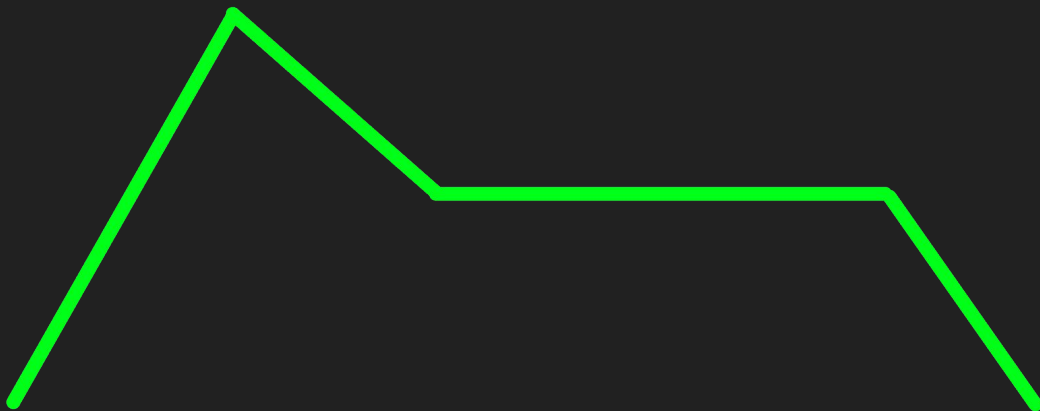
Envelopes

Envelopes

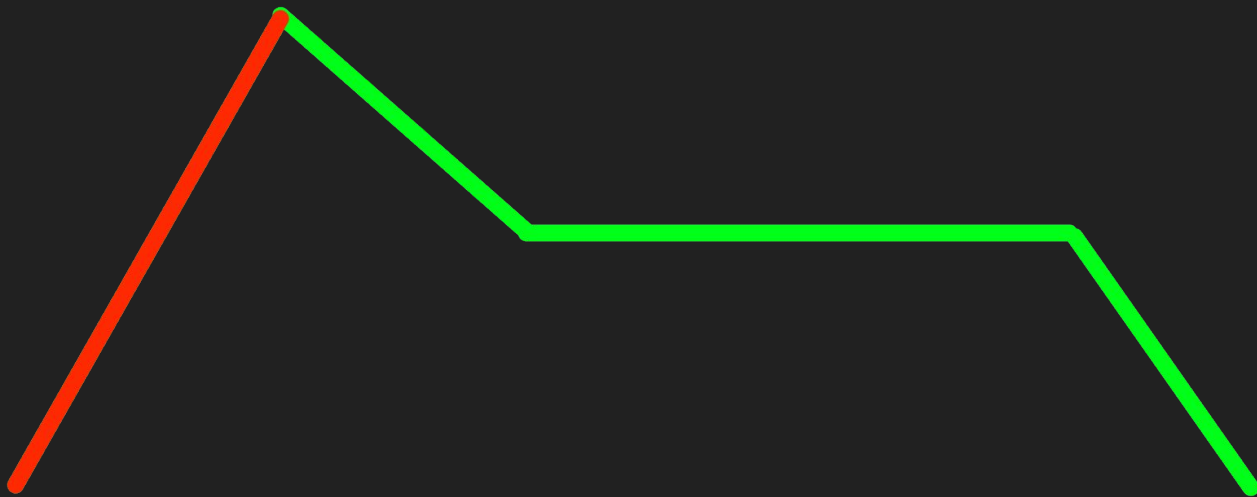
Used to change a note's sound over time

Like an invisible hand moving a control

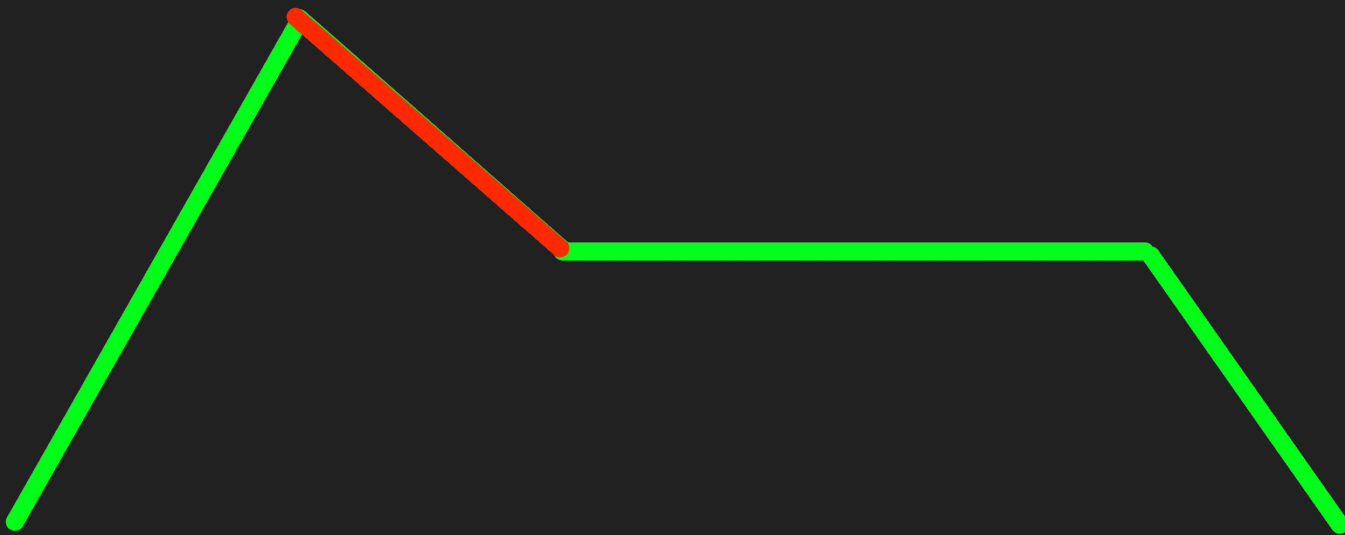
Can change sound parameters like the volume or filter



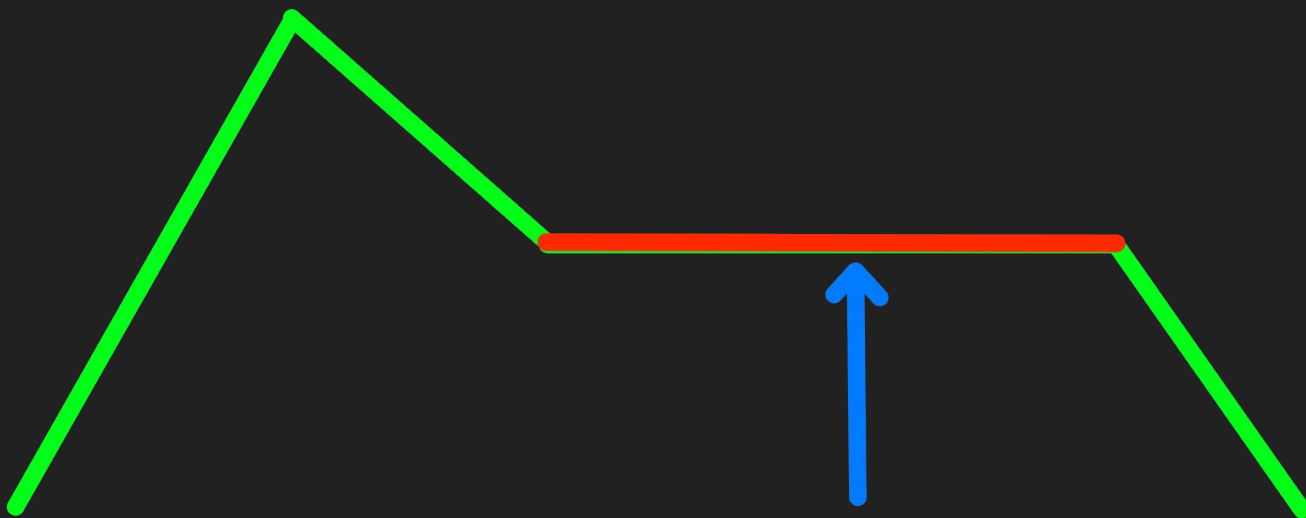
Attack



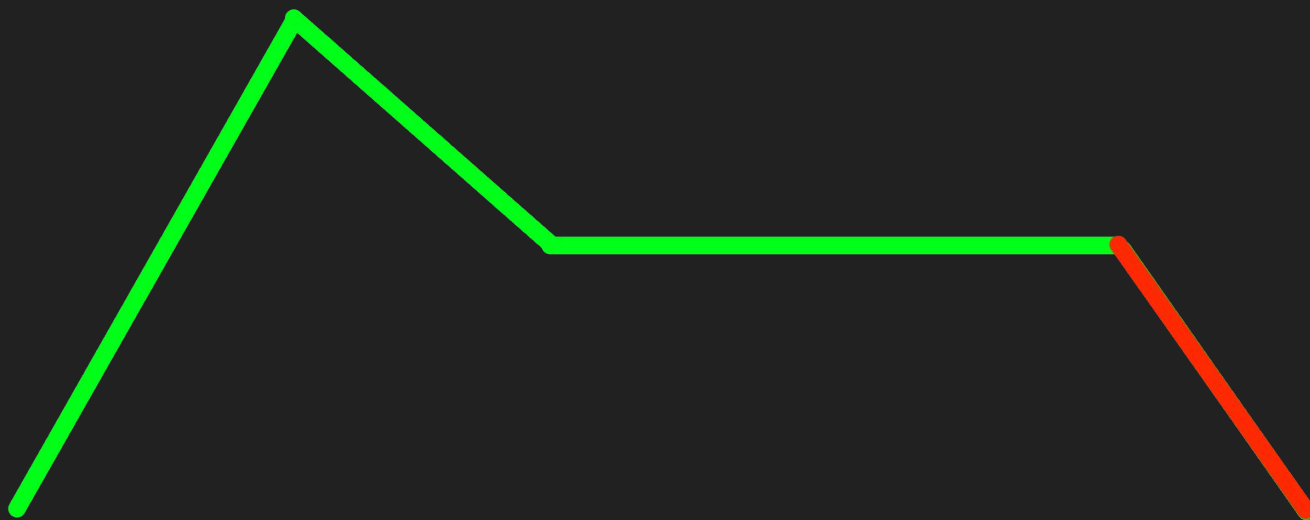
Decay



Sustain



Release



Code for amp (volume) envelope

DaisySP handles the envelope for us. All we need to do is change the signal volume

The VCA allows us to change a signal's volume

```
//Change the amplitude of the sound based on the amp envelope  
float ampEnvValue = ampEnvelope.Process(gateOpen);  
out = vca.operator() (out, ampEnvValue);
```


Code for filter envelope

We need to process the filter envelope before we run the sample through the filter

We set the filter's cutoff frequency and process the sample as before

```
float filterEnvValue = filterEnvelope.Process(gateOpen) * filterEnvelopeAmount;  
filter.SetFreq( freq: filterFreq + filterEnvValue);  
out = filter.Process(out);
```

Mistakes to learn from

Mistakes to learn from and improvements

- You need to add new files to the CMakeLists.txt file
- There is a lot of code duplication and boilerplate code
 - This was accepted to reduce complexity
- **Add a volume control first**

Where's the code?

<https://github.com/SkylarGill/skye-nthesiser>

Fin