

15-104 Introduction to Computing for Creative Practice

Fall 2020

25 Sound Processing

Instructor: Tom Cortina, tcortina@cs.cmu.edu, GHC 4117, 412-268-3514

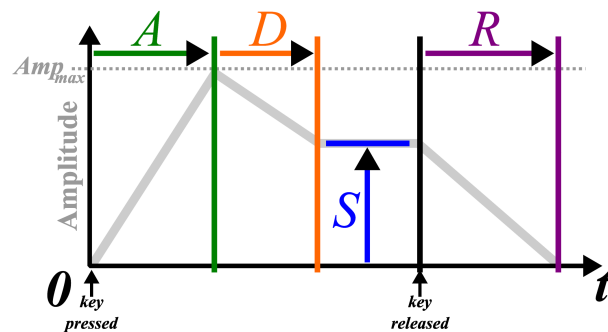
1



Envelopes

Envelopes are used to control the output volume of an object, a series of fades referred to as Attack, Decay, Sustain and Release (ADSR).

```
var attackTime = 0.01;
var decayTime = 0.2;
var sustainPrct = 0.2;
var releaseTime = 0.5;
```



2



Example (Scale with Envelopes)

```

var count = 0;
...
function setup() { /* SAME */ }
function soundSetup() {
  osc = new p5.Oscillator();
  osc.setType('sawtooth');
  env = new p5.Envelope();
  env.setADSR(attackTime, decayTime, sustainPrct, releaseTime);
  osc.amp(env);
  osc.freq(midiToFreq(note));
  osc.start();
}

```

3



Example (cont'd)

```

function draw() {
  background(200);
  if (count == 0) { // start first note
    osc.freq(midiToFreq(note));
    env.play();
  }
  else if (count % 4 == 0) {
    if (stepIndex == steps.length) { osc.stop(); noLoop(); }
    note += steps[stepIndex];
    osc.freq(midiToFreq(note));
    env.play();
    stepIndex++;
  }
  count++;
}

```

4

Chords

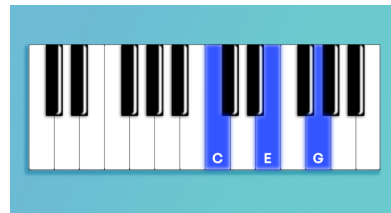
We can play a major chord by creating three oscillators for the note components of the chord.

- We use the first oscillator as the root.
- We disconnect the other two oscillators from direct audio output and add them to the root instead.

If the root is (MIDI) note n ,
then the other two notes are:

$n+4$ major third

$n+7$ perfect fifth



5

Example (Chords)

```
var osc, osc2, osc3;
...
function soundSetup() {
  osc = new p5.Oscillator(); osc.setType('sawtooth');
  osc2 = new p5.Oscillator(); osc2.setType('sawtooth');
  osc3 = new p5.Oscillator(); osc3.setType('sawtooth');
  env = new p5.Envelope();
  env.setADSR(attackTime, decayTime, sustainPrct, releaseTime);
  osc.amp(env); osc.start();
  osc2.amp(env); osc2.start();
  osc3.amp(env); osc3.start();
}
```

6

Example (Chords) – cont'd

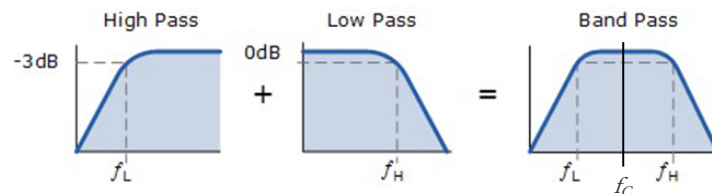
```
function draw() {
  if (count == 0) { // start first note
    osc.freq(midiToFreq(note)); osc2.freq(midiToFreq(note+4));
    osc3.freq(midiToFreq(note+7)); env.play();
  } else if (count % 4 == 0) {
    if (stepIndex == steps.length) {
      osc.stop(); osc2.stop(); osc3.stop(); noLoop();
    }
    note += steps[stepIndex];
    osc.freq(midiToFreq(note)); osc2.freq(midiToFreq(note+4));
    osc3.freq(midiToFreq(note+7)); env.play();
    stepIndex++;
  }
  count++;
}
```

7

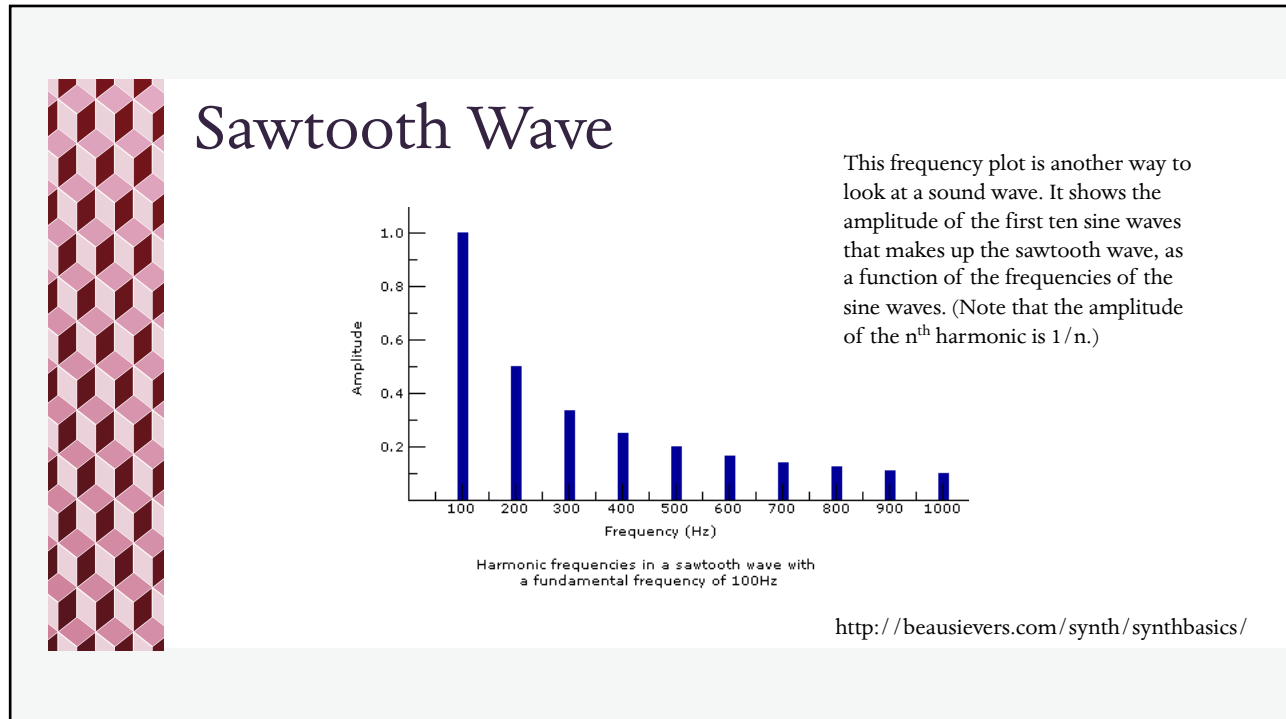
Filters

An oscillator can be passed through a filter to reduce some of the frequencies passing through.

- LowPass – allows frequencies below f_H to pass
- HighPass – allows frequencies above f_L to pass
- BandPass – allows frequencies around f_C to pass




8



9

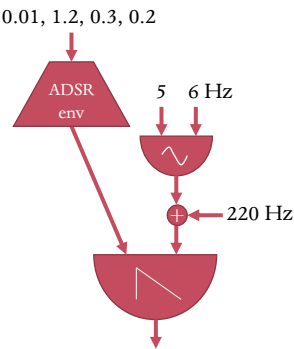
Filters in p5.js



```
function soundSetup() {
  osc = new p5.Oscillator();
  osc.setType('sawtooth');
  osc.freq(220);

  env = new p5.Envelope();
  env.setADSR(0.01, 1.2, 0.3, 0.2);
  osc.amp(env);

  vib = new p5.Oscillator();
  vib.setType('sine');
  vib.freq(6); vib.amp(5);
  vib.start();
  osc.freq(vib);
}
```



10

Filters in p5.js (cont'd)

```

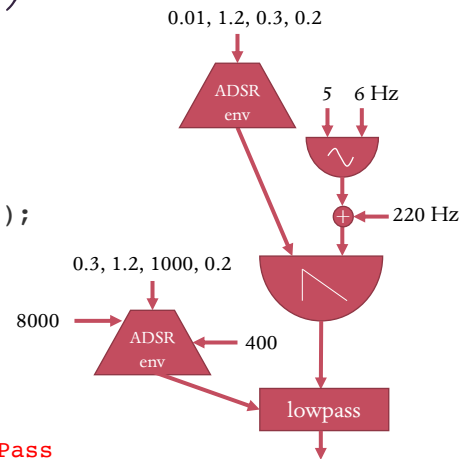
filt = new p5.LowPass();
osc.disconnect();
osc.connect(filt);

fenv = new p5.Envelope();
fenv.setADSR(0.3, 1.2, 1000, 0.2);
fenv.setRange(8000, 400);
filt.freq(fenv);

filt.freq(5000);
osc.start();
}

```

Can also use `p5.HighPass` or `p5.BandPass`



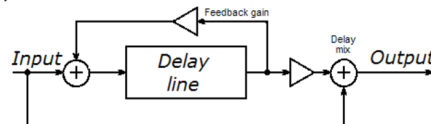
11

Delay & Reverb

```

delay = new p5.Delay();
delay.process(osc, 0.12, .7, 2300);
osc.start();

```



```

reverb = new p5.Reverb();
reverb.process(osc, 3, 2);
osc.start();

```

Parameters:
 sound source,
 delayTime (in seconds),
 feedback (less than 1!),
 lowpass filter frequency

Parameters:
 reverb time (seconds),
 decay rate (percent)

12