

15-104 Introduction to Computing for Creative Practice

Fall 2020

24 More Sound

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

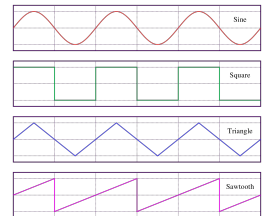
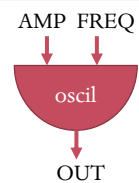
1



p5.Oscillator


- Creates a signal that oscillates between -1.0 and 1.0.
 - By default, the oscillation takes the form of a sinusoidal shape ('sine').
 - The frequency defaults to 440 oscillations per second (440Hz).
- Some methods:

• <code>start()</code>	Start an oscillator.
• <code>stop()</code>	Stop an oscillator.
• <code>amp()</code>	Set the amplitude between 0 and 1.0.
• <code>freq()</code>	Set frequency of an oscillator to a value.
• <code>setType()</code>	Set type to 'sine', 'square', 'triangle', or 'sawtooth'.
• <code>disconnect()</code>	Do not send output of this oscillator to the speakers.



2

Example (review)



```
function draw() {
  background(200);
  fill(0);
  ellipse(mouseX, mouseY, 20, 20);
  myTone.amp(constrain(mouseX / width, 0, 1));
  myTone.freq(constrain(200 + 1000*(mouseY / height), 200, 1200));
  if (mouseX > 2*width) {
    myTone.stop();
    noLoop();
  }
}
```


Volume ↔

Pitch ↑↓

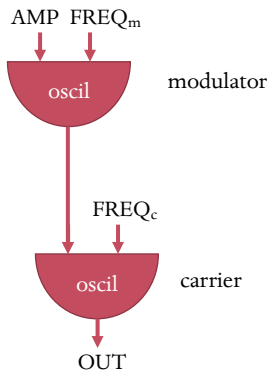
A fraction between 0 and 1 when the mouse is in the canvas bounds.

3


Amplitude Modulation



- When the amplitude of an oscillator is controlled by another oscillator, you have amplitude modulation.
- The sound producing oscillator is called the carrier and the oscillator controlling the amplitude is called the modulator.
- If the frequency of modulation is small, we get the musical effect we know as tremolo.
- If the frequency of modulation is large, we get a spectrally-rich sound instead.



4



Tremolo

```


var myTone;
var tremolo;
function setup() {  SAME AS BEFORE  }

function soundSetup() { // setup for audio generation
  myTone = new p5.Oscillator();
  myTone.setType('sine');
  myTone.start();
  tremolo = new p5.Oscillator();
  tremolo.setType('sine');
  tremolo.disconnect();    // don't send this oscil to speakers
  tremolo.start();
}

```



5



Tremolo (cont'd)

```

function draw() {
  background(200);
  fill(0);
  ellipse(mouseX, mouseY, 20, 20);
  tremolo.amp(0.75);
  tremolo.freq(constrain(2 + 6 * (mouseX / width), 2, 8));
  myTone.amp(tremolo);
  myTone.freq(constrain(200 + 1000 * (mouseY / height), 200, 1200));
  if (mouseX > 2*width) {
    myTone.stop(); noLoop();
  }
}

```

The amplitude will increase to 0.75 from 2 to 8 times per second depending on the horizontal position of the mouse.

The `amp()` function can also have an oscillator as its argument.

6

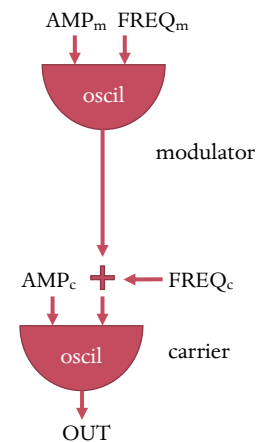
Amplitude Modulation

```
function draw() {
  background(200);
  fill(0);
  ellipse(mouseX, mouseY, 20, 20);
  tremolo.amp(0.75);
  tremolo.freq(constrain(100 + 100 * (mouseX / width), 100, 200));
  myTone.amp(tremolo);
  myTone.freq(constrain(200 + 1000 * (mouseY / height), 200, 1200));
  if (mouseX > 2*width) {
    myTone.stop(); noLoop();
  }
}
```

7

Frequency Modulation

- When the frequency of an oscillator is controlled by another oscillator, you have **frequency modulation**.
- The sound producing oscillator is called the carrier and the oscillator controlling the amplitude is called the modulator.
- If the frequency of modulation is small, we get the musical effect we know as vibrato.
- If the frequency of modulation is large, we get a spectrally-rich sound instead.



8

Vibrato




```


var myTone;
var vibrato;
function setup() {  SAME AS BEFORE  }

function soundSetup() { // setup for audio generation
  myTone = new p5.Oscillator();      // default freq: 440Hz
  myTone.setType('sine');
  myTone.start();
  vibrato = new p5.Oscillator();
  vibrato.setType('sine');
  vibrato.disconnect();    // don't send this oscil to speakers
  vibrato.start();
}

```

9

Vibrato (cont'd)



```


function draw() {
  background(200);
  fill(0);
  ellipse(mouseX, mouseY, 20, 20);
  vibrato.amp(constrain(100 * (mouseY / height), 0, 100));
  vibrato.freq(constrain(1 + 3 * (mouseX / width), 1, 4));
  myTone.amp(0.5);
  myTone.freq(vibrato);
  if (mouseX > 2*width) {
    myTone.stop(); noLoop();
  }
}

```

The pitch will vary above and below 440 Hz between 1-4 times a second by 0 to 100 Hz three times, depending on the vertical position of the mouse.

The `freq()` function can also have an oscillator as its argument.


10



Frequency Modulation

```
function draw() {
  background(200);
  fill(0);
  ellipse(mouseX, mouseY, 20, 20);
  vibrato.amp(constrain(440 * (mouseY / height), 0, 440));
  vibrato.freq(constrain(440 * (mouseX / width), 0, 440));
  myTone.amp(0.5);
  myTone.freq(vibrato);
  if (mouseX > 2*width) {
    myTone.stop(); noLoop();
  }
}
```


11




Common Pitches

A0	Z7.5	A0#	Z91.135	F2#	92.489	C4#	277.18	F6#	739.99	F6	1480.0
B0	30.668	C1#	34.648	G2#	103.83	D4#	311.13	G5#	830.61	G6	1661.2
C1	32.703	D1#	36.951	A2#	116.54	E4#	329.63	A5#	832.33	A6	1664.7
D1	36.708	F1#	46.249	C3#	130.81	F4#	349.23	B5#	897.77	C7#	2217.5
E1	41.203	G1#	48.999	D3#	146.83	G4#	392.00	C6#	1046.5	D7#	2349.3
F1	43.654	A1#	50.000	E3#	164.81	A4#	416.30	D6#	1174.7	E7#	2489.0
G1	48.999	B1#	61.735	F3#	174.61	B4#	439.88	E6#	1316.5	F7#	2560.0
A1	50.000	C2#	69.296	G3#	186.00	C5#	523.25	F6#	1568.0	G7#	3136.0
B1	61.735	D2#	77.782	A3#	200.00	D5#	587.33	G6#	1760.0	A7#	3522.4
C2	66.406	E2#	82.407	B3#	216.94	E5#	659.25	A6#	1979.5	B7#	3951.1
D2	73.416	F2#	87.307	C4#	235.66	F5#	739.46	B6#	2183.0	C8	4186.0
E2	82.407	G2#	97.999	D4#	253.66	G5#	793.99	C7#	2349.3		
F2	87.307	A2#	116.54	E4#	329.63	A5#	832.33	D7#	2489.0		
G2	97.999	B2#	123.47	F4#	349.23	B5#	897.77	E7#	2637.0		
A2	110.00	C3#	130.81	G4#	392.00	C6#	1046.5	F7#	2793.8		
B2	123.47	D3#	146.83	A4#	416.30	D6#	1174.7	G7#	3136.0		
C3	130.81	E3#	164.81	B4#	439.88	E6#	1316.5	A7#	3522.4		
D3	146.83	F3#	174.61	C5#	523.25	F6#	1568.0	B7#	3951.1		
E3	164.81	G3#	186.00	D5#	587.33	G6#	1760.0	C8	4186.0		
F3	174.61	A3#	200.00	E5#	659.25	A6#	1979.5				
G3	186.00	B3#	216.94	F5#	739.46	B6#	2183.0				
A3	200.00	C4#	235.66	G5#	793.99	C7#	2349.3				
B3	216.94	D4#	253.66	A5#	832.33	D7#	2489.0				
C4	235.66	E4#	329.63	B5#	897.77	E7#	2637.0				
D4	253.66	F4#	349.23	C6#	1046.5	F7#	2793.8				
E4	329.63	G4#	392.00	D6#	1174.7	G7#	3136.0				
F4	349.23	A4#	416.30	E6#	1316.5	A7#	3522.4				
G4	392.00	B4#	439.88	F6#	1568.0	B7#	3951.1				
A4	416.30	C5#	523.25	G6#	1760.0	C8	4186.0				
B4	439.88	D5#	587.33	A6#	1979.5						
C5	523.25	E5#	659.25	B6#	2183.0						
D5	587.33	F5#	739.46	C7#	2349.3						
E5	659.25	G5#	793.99	D7#	2489.0						
F5	739.46	A5#	832.33	E7#	2637.0						
G5	793.99	B5#	897.77	F7#	2793.8						
A5	832.33	C6#	1046.5	G7#	3136.0						
B5	897.77	D6#	1174.7	A7#	3522.4						
C6	1046.5	E6#	1316.5	B7#	3951.1						
D6	1174.7	F6#	1568.0	C8	4186.0						
E6	1316.5	G6#	1760.0								
F6	1568.0	A6#	1979.5								
G6	1760.0	B6#	2183.0								
A6	1979.5	C7#	2349.3								
B6	2183.0	D7#	2489.0								
C7	2349.3	E7#	2637.0								
D7	2489.0	F7#	2793.8								
E7	2637.0	G7#	3136.0								
F7	2793.8	A7#	3522.4								
G7	3136.0	B7#	3951.1								
A7	3522.4	C8	4186.0								
B7	3951.1										
C8	4186.0										


Middle C




12



MIDI (Musical Instrument Digital Interface)



13



Example: Play major scale

```


var note = 60; // starting note in MIDI notation
var steps = [2, 2, 1, 2, 2, 2, 1]; // 2=whole step, 1=half step
var stepIndex = 0;
var osc;
function setup() {
  createCanvas(200, 100);
  frameRate(4);
  useSound();
}
function soundSetup() {
  osc = new p5.Oscillator();
  osc.amp(0.25); osc.freq(midiToFreq(note)); osc.start(); }

```

midiToFreq returns the frequency that corresponds to the MIDI value given as its argument.

14

Example: Play major scale (cont'd)




```
function draw() {
  print(frameCount);
  background(200);
  if (frameCount % 4 == 0) {
    if (stepIndex == steps.length) {
      osc.stop(); noLoop();
    }
    note += steps[stepIndex];
    osc.freq(midiToFreq(note));
    stepIndex++;
  }
}
```

When you listen to this program's sound, why is the first note shorter in duration than the others? How would you fix this?

15

switch statement



```
if (x == 0) {
  freq += 100;
} else if (x == 1) {
  freq -= 100;
} else if (x == 2) {
  freq *= 2;
} else if (x == 3) {
  freq /= 2;
} else {
  freq = 440;
}
```

```
switch (x) {
  case 0: freq += 100; break;
  case 1: freq -= 100; break;
  case 2: freq *= 2; break;
  case 3: freq /= 2; break;
  default: freq = 440;
}
```

The break statements are important here!
They break out of the switch statement.
Without them, case 0 would execute all 5 cases!
(case only indicates where to start)

16