

Creative Coding

Professor Danne Woo

creativecode.dannewoo.com

ARTS 249

Spring 2020

Thursday 2:00 PM – 5:50 PM

I-Building 213

Motion and Interaction

Week 08: Kinetic Forms

Week 09: Drawing and Interaction (Mouse)

Week 10: Interaction and Image Import (Keyboard and Events)

Week 11: Work from Home

Week 12: 3D

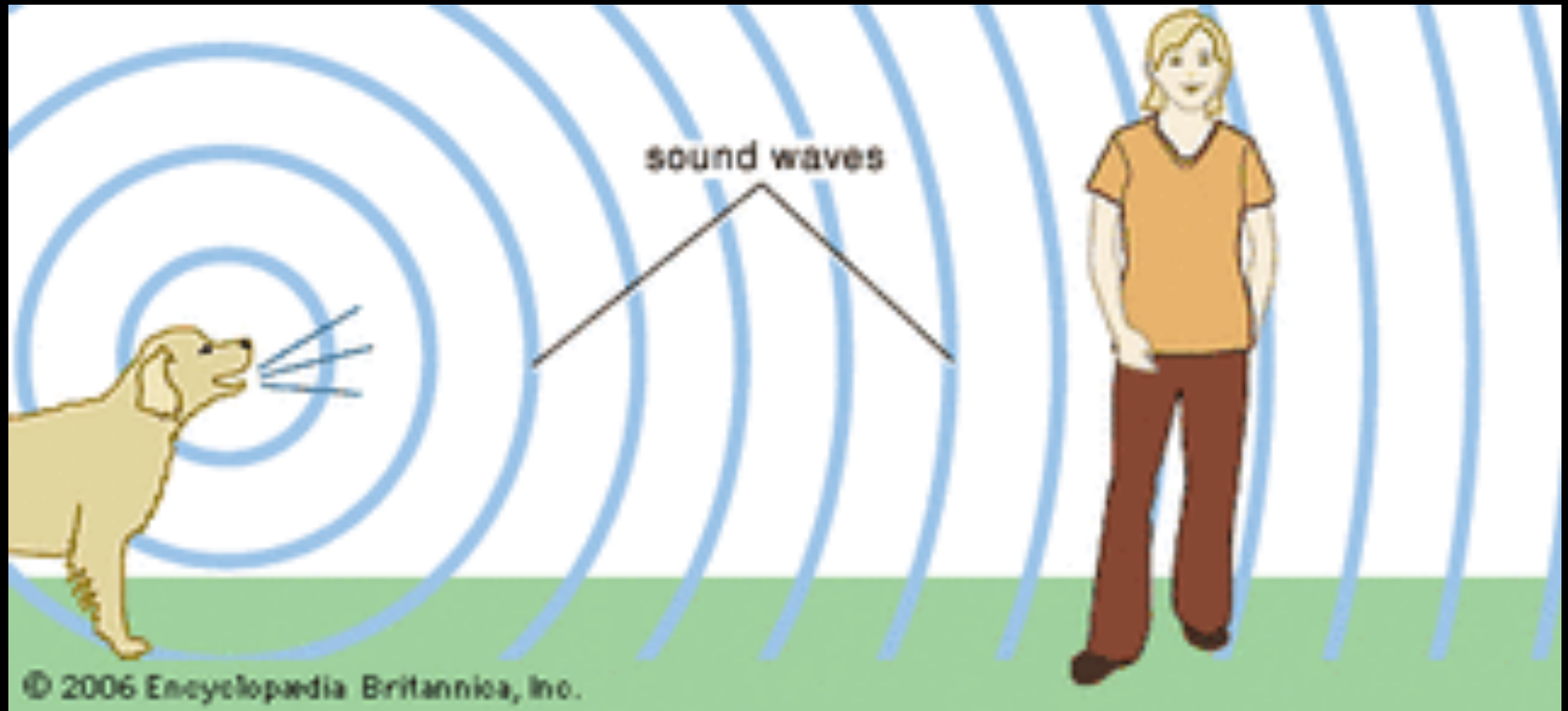
Week 13: Audio and p5.Sound

Week 14: Computer Vision and In Class Work

Week 15: Final Presentation

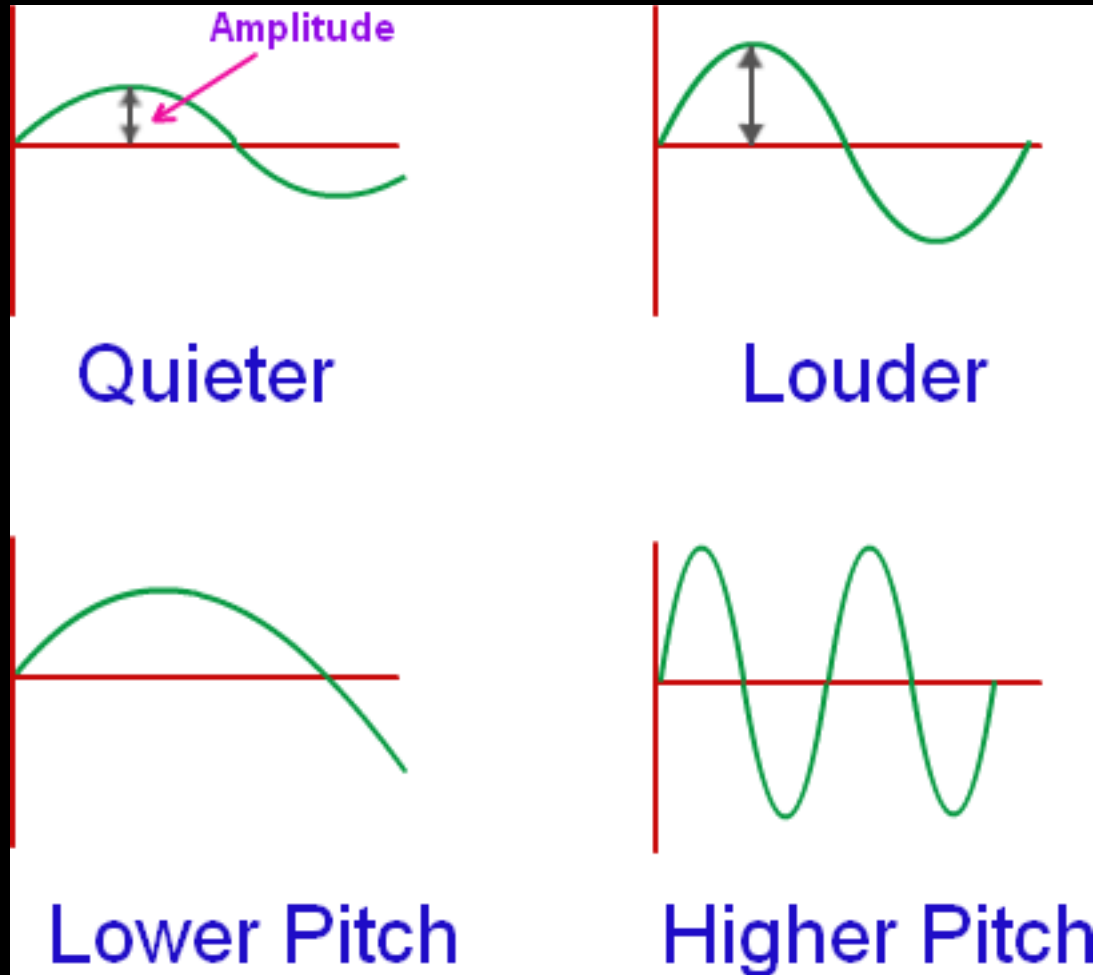
Audio

How Does Sound Work?



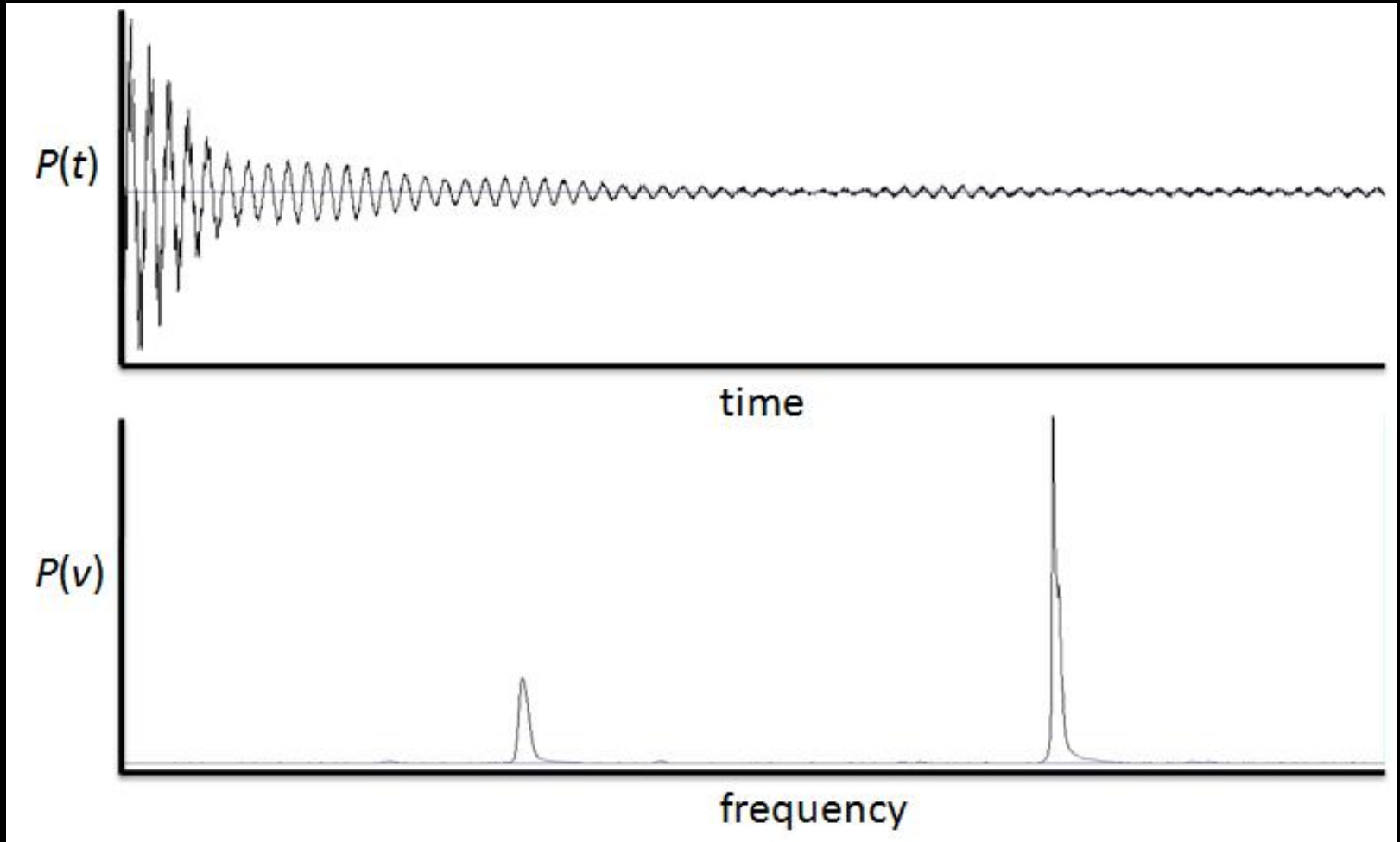
Audio

How Does Sound Work?



Audio

FFT Analysis (Fast Fourier transform)



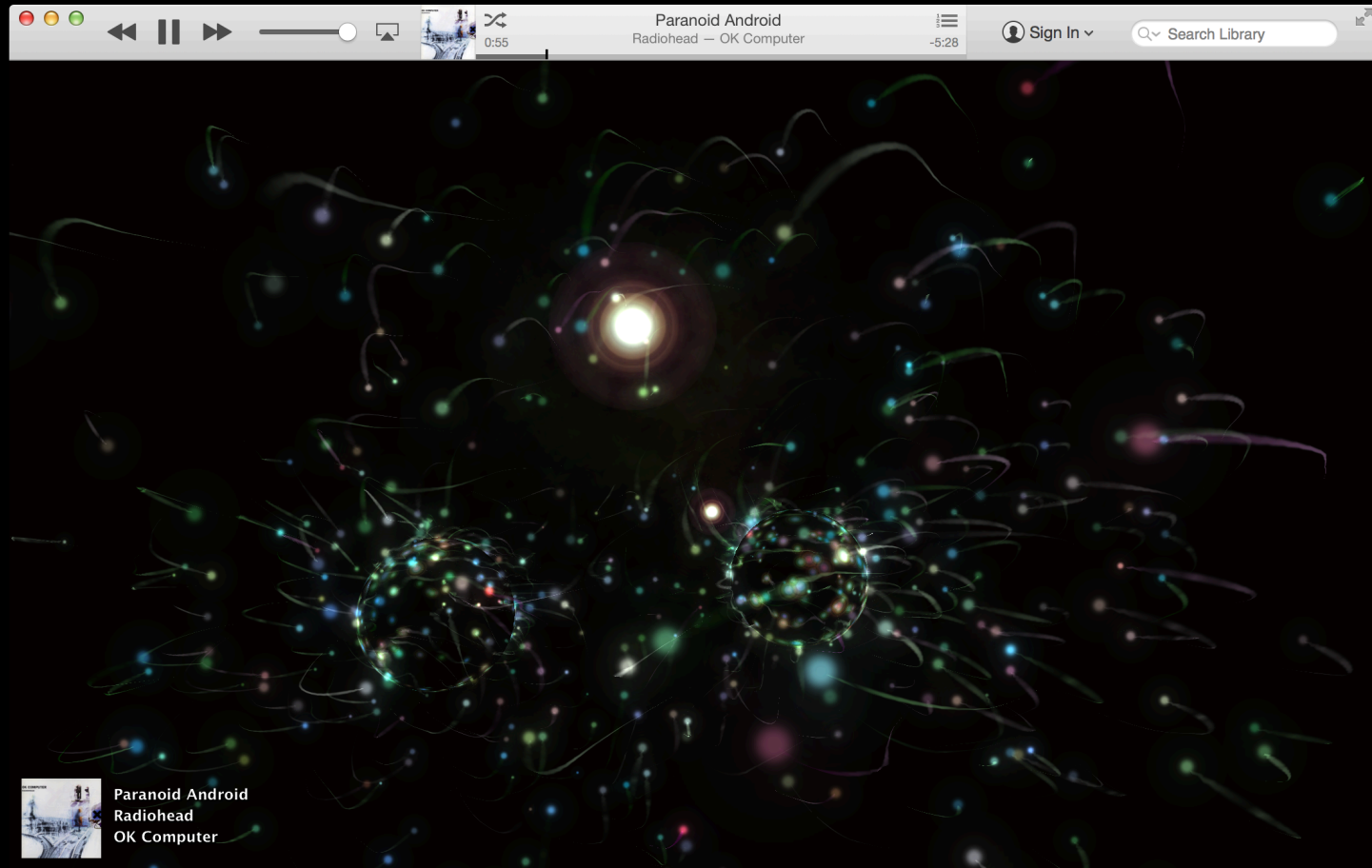
Audio

Audio Visualizers



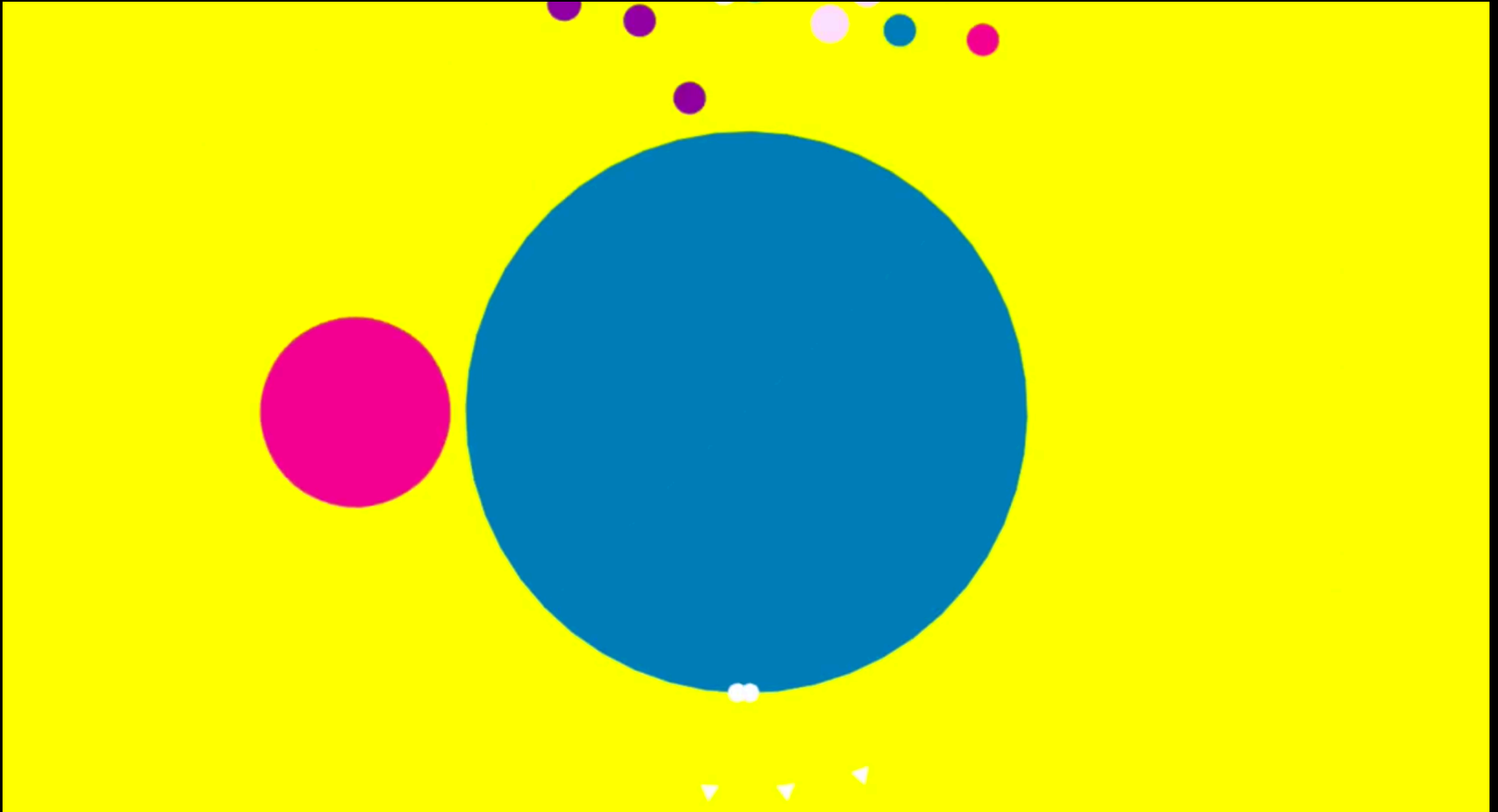
Audio

Audio Visualizers



Audio

Instruments



p5.Sound

Classes and Functions

Link Library – `<script src="addons/p5.sound.min.js"></script>`

AudioPlayer – `loadSound()`, `play()`, `pause()`, `loop()`, `setVolume()`

AudioInput – `p5.AudioIn()`, `connect()`, `disconnect()`, `setInput()`

Oscillator – `p5.Oscillator()`, `freq()`, `amp()`

FFT – `p5.FFT()`, `analyze()`, `linAverages()`, `logAverages()`, `waveform()`

p5.sound

Load Audio File

```
var mySound;

function preload() {
  mySound = loadSound('audio/marcus_kellis_theme.mp3');
}

function setup() {
  createCanvas(600, 400);
  background(0);
  // Volume is a value between 0 and 1
  mySound.setVolume(1);
  // Play the audio file
  mySound.play();
}
```

p5.sound

Keypressed Instrument

```
var s01, s02, ...;
```

```
function preload() {  
  s01 = loadSound('wipe.mp3');  
  s02 = loadSound('zig-zag.mp3');  
  ...  
}
```

```
function setup() {  
  createCanvas(400, 400);  
  background(0);  
  s01.setVolume(1);  
  s02.setVolume(1);  
  ...  
}
```

```
function draw() {  
  background(0);  
}
```

```
function keyTyped() {  
  if (key == 'a') {  
    s01.play();  
  }  
  if (key == 's') {  
    s02.play();  
  }  
  ...  
}
```

p5.sound

FFT Analysis

```
var mySound, fft;

function preload() {
  mySound = loadSound('audio/marcus_kellis_theme.mp3');
}

function setup() {
  createCanvas(600, 400);
  background(0);
  // Create a new FFT analysis
  fft = new p5.FFT();
  mySound.setVolume(1);
  mySound.play();
}
```

p5.sound

FFT Analysis Analyze

```
function draw(){  
  background(0);  
  // Use the analyze function to find the values for each frequency  
  var spectrum = fft.analyze();  
  noStroke();  
  fill(0,255,0);  
  for (var i = 0; i< spectrum.length; i++){  
    var x = map(i, 0, spectrum.length, 0, width);  
    var h = map(spectrum[i], 0, 255, 0, height);  
    rect(x, height - h, width / spectrum.length, h )  
  }  
}
```

p5.sound

FFT Analysis Waveform

```
function draw(){  
  background(0);  
  // Use the waveform function to find the values for audio wave  
  var waveform = fft.waveform();  
  noFill();  
  beginShape();  
  stroke(255,0,0);  
  strokeWeight(1);  
  for (var i = 0; i< waveform.length; i++){  
    var x = map(i, 0, waveform.length, 0, width);  
    var y = map( waveform[i], -1, 1, 0, height);  
    vertex(x,y);  
  }  
  endShape();  
}
```

p5.sound

FFT Analysis Linear Averages

```
var mySound, fft;

function preload() {
  mySound = loadSound('audio/marcus_kellis_theme.mp3');
}

function setup() {
  createCanvas(600, 400);
  background(0);
  fft = new p5.FFT();
  mySound.setVolume(1);
  mySound.play();
  noStroke();
  fill(255);
}
```

p5.sound

FFT Analysis Linear Averages

```
function draw(){  
  background(0);  
  
  fft.analyze();  
  var fftLin = fft.linAverages(5);  
  for (var i = 0; i < fftLin.length; i++){  
    var x = map(i, 0, fftLin.length, 0, width);  
    var h = map(fftLin[i], 0, 255, 0, height);  
    rect(x, height - h, width / fftLin.length, h )  
  }  
}
```

p5.sound

Oscillator

```
var osc;
```

```
function setup(){  
  createCanvas(600,400);  
  osc = new p5.Oscillator();  
  osc.start();  
}
```

```
function draw(){  
  background(0);  
  // Change the audio frequency based on the mouseX location  
  // Change the amplitude (volume) based on the mouseY location  
  var modFreq = map(mouseX, 0, width, 20, 15000);  
  var modAmp = map(mouseY, 0, height, 0, 1);  
  osc.freq(modFreq);  
  osc.amp(modAmp);  
}
```

p5.sound

Audio In

```
var mic;  
  
function setup() {  
  createCanvas(512,400);  
  noStroke();  
  fill(0,255,255);  
  mic = new p5.AudioIn();  
  mic.start();  
}
```

p5.sound

Audio In

```
var mic;

function setup() {
  createCanvas(512,400);
  noStroke();
  fill(0,255,255);
  mic = new p5.AudioIn();
  mic.start();
}

function draw() {
  // Plays the audio in sound on the speakers
  mic.connect();
}
```

Creative Coding

Professor Danne Woo

dwoo@qc.cuny.edu

creativecode.dannewoo.com