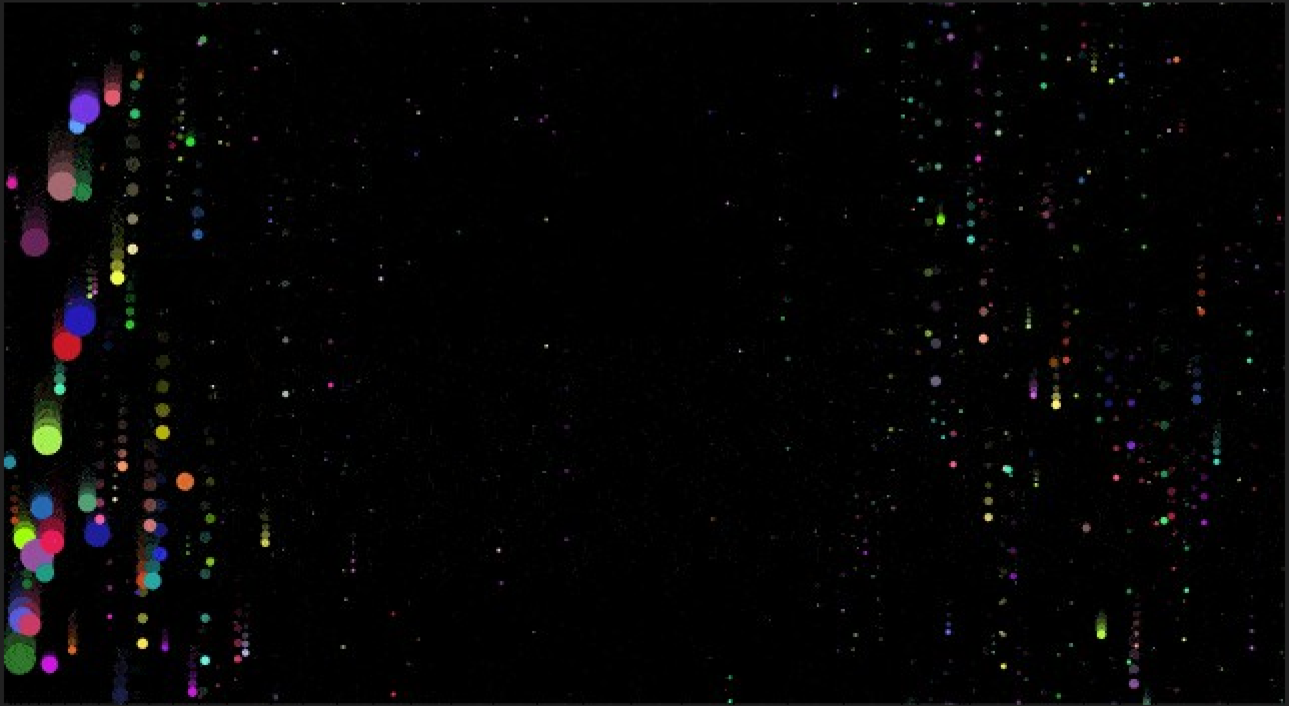
Sound Galaxy

Hack Club

https://replit.com/@TomasVargas-Ber/Galaxy-for-slide-show-purposes

Sound Visualization

Today’s Goal:



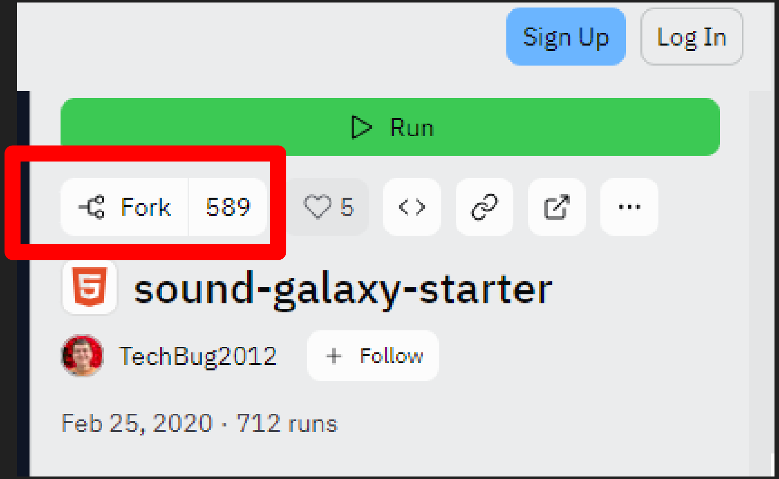
Sound Visualization

Getting Started:

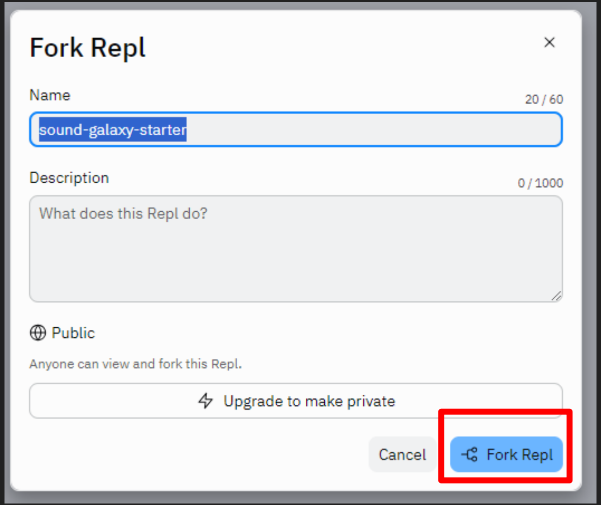
1. Open the starter project:

https://replit.com/@TechBug2012/sound-galaxy-starter

2a. Fork the Repl



2b.


Fork Repl

Name 20 / 60
sound-galaxy-starter

Description 0 / 1000
What does this Repl do?

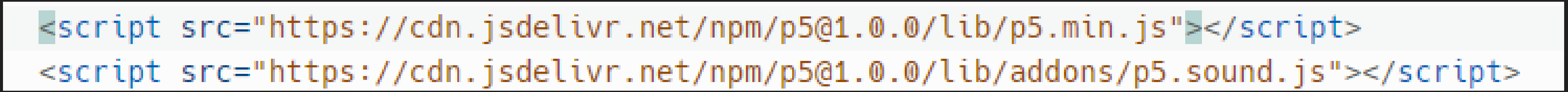
Public
Anyone can view and fork this Repl.

Sound Visualization

Importing Libraries:

Import p5.js and p5.sound.js:

(Add to Index.html under just before the end of your <head> tag)

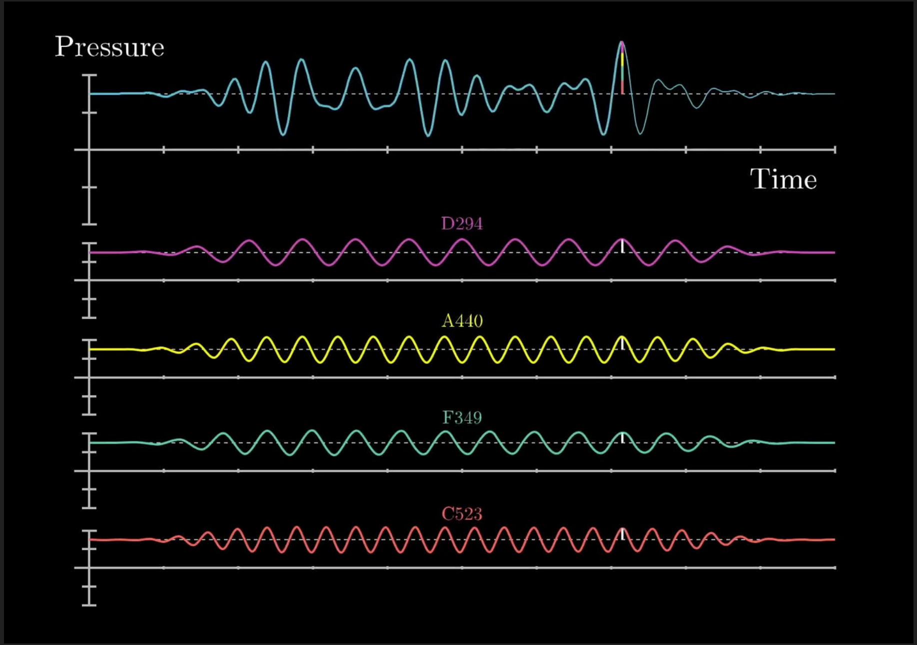


```
<script src="https://cdn.jsdelivr.net/npm/p5@1.0.0/lib/p5.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/p5@1.0.0/lib/addons/p5.sound.js"></script>
```

Sound Visualization

Fourier Transform

- mathematical operation
- takes a frequency decomposes it into the individual wavelengths that make it up
- Built into p5

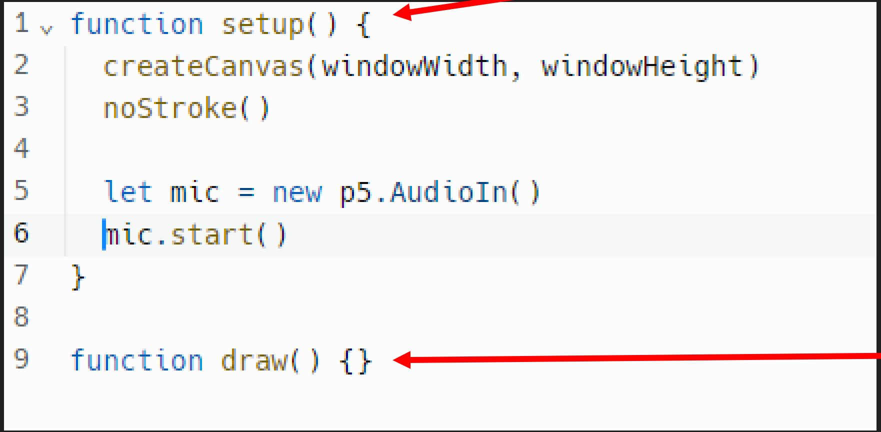


Sound Visualization

Main Functions

Setup the 2 main functions of this project: setup() and draw()

Add the following to script.js:



```
function setup() {
  createCanvas(windowWidth, windowHeight)
  noStroke()

  let mic = new p5.AudioIn()
  mic.start()
}

function draw() {}
```

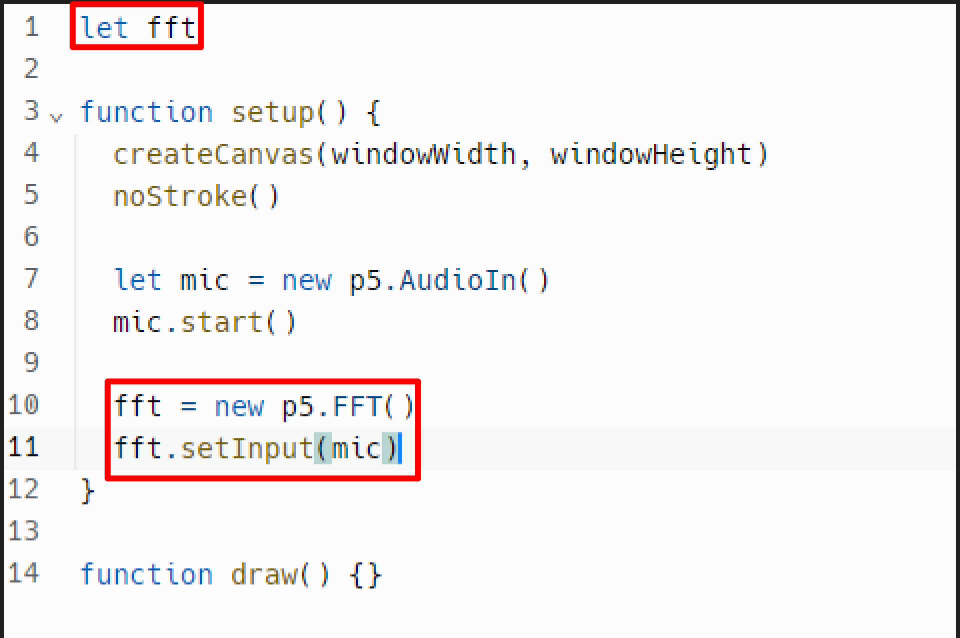
Runs once at start

Runs continuously after setup()

Sound Visualization

Add the Fast Fourier Transform

Add the following to script.js:

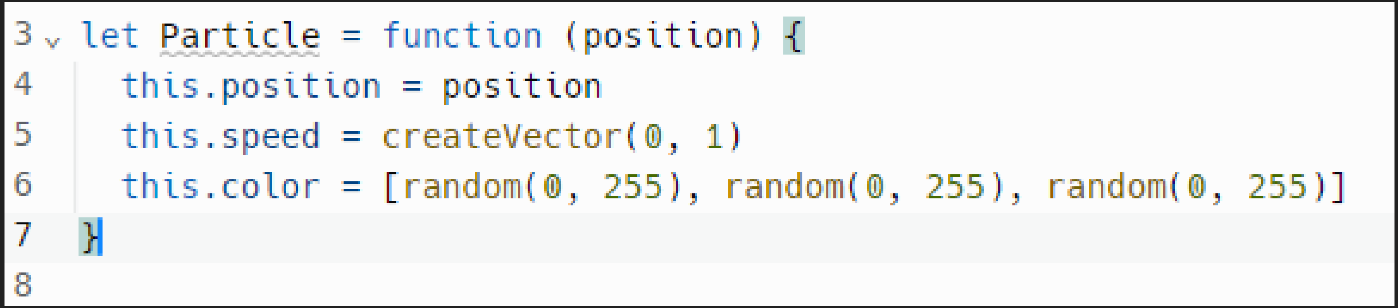


```
let fft

function setup() {
  createCanvas(windowWidth, windowHeight)
  noStroke()

  let mic = new p5.AudioIn()
  mic.start()

  fft = new p5.FFT()
  fft.setInput(mic)
}

function draw() {}
```

Sound Visualization

Creating a Particle

Add the following above the setup()



```
let Particle = function (position) {
    this.position = position
    this.speed = createVector(0, 1)
    this.color = [random(0, 255), random(0, 255), random(0, 255)]
}
```

Here, we’re setting:

Position: to a position given to the Particle when it’s created

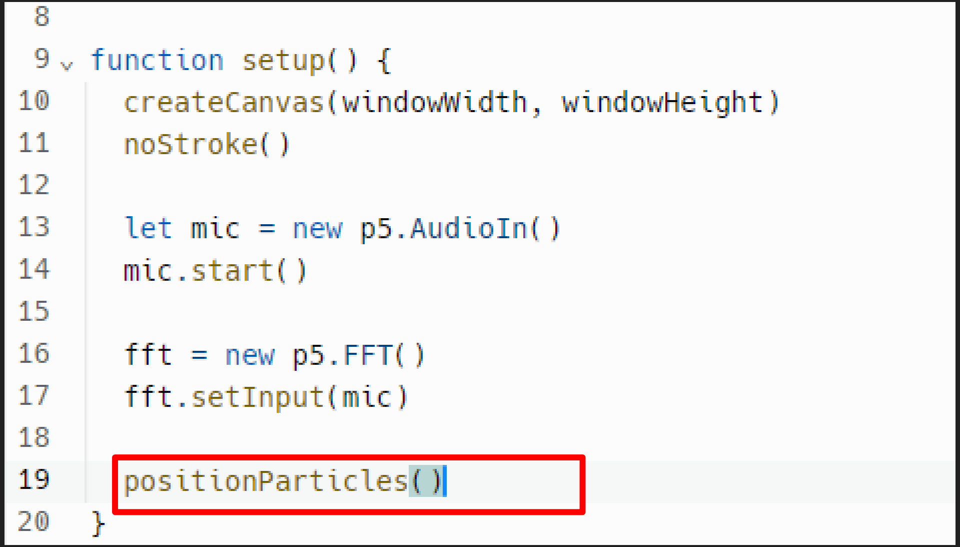
Speed: 0 in the x direction and 1 in the y direction

Color:a random RGB color.

Sound Visualization

Creating a Particle

Add the following to setup()



```
function setup() {
  createCanvas(windowWidth, windowHeight)
  noStroke()

  let mic = new p5.AudioIn()
  mic.start()

  fft = new p5.FFT()
  fft.setInput(mic)

  positionParticles()
}
```

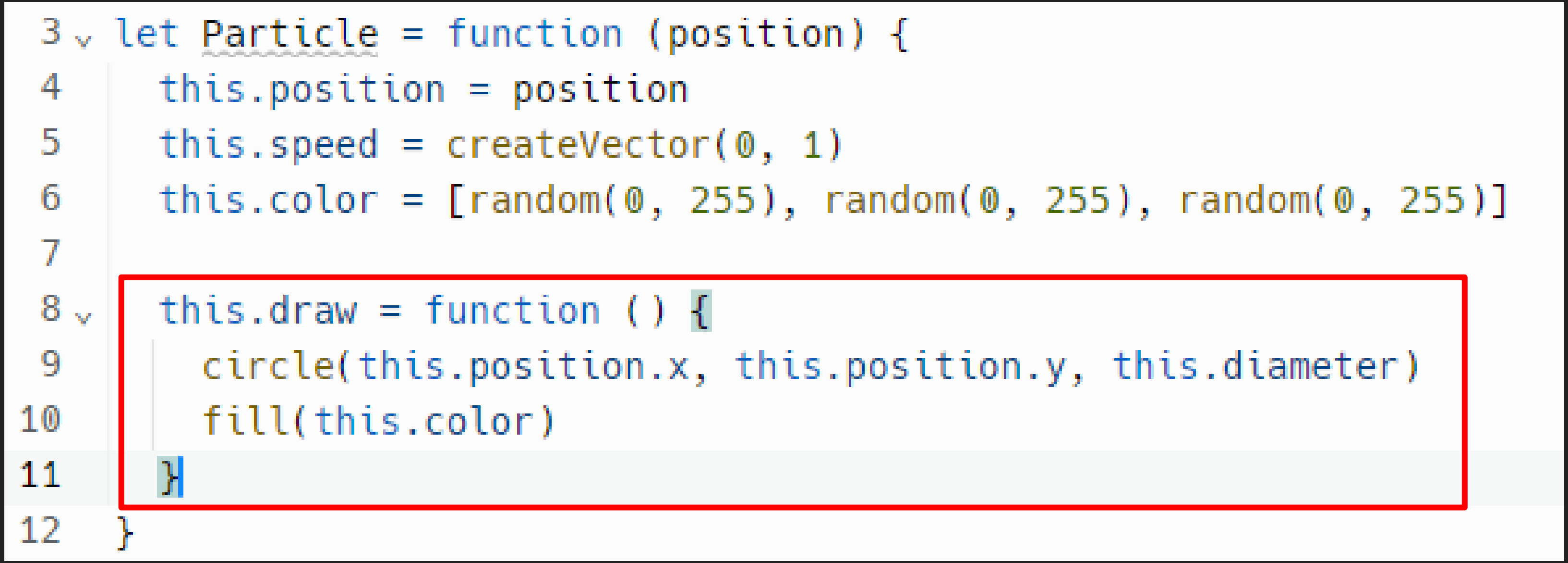
Calling positionParticles() from galaxyManager.js

(creates an array of 1024 particles at random places on your screen)

Sound Visualization

Creating a Particle

Add the following to setup()

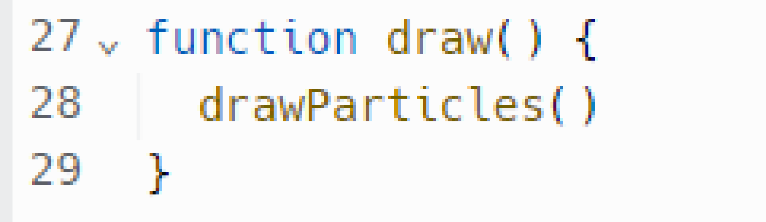


```
let Particle = function (position) {
  this.position = position
  this.speed = createVector(0, 1)
  this.color = [random(0, 255), random(0, 255), random(0, 255)]

  this.draw = function () {
    circle(this.position.x, this.position.y, this.diameter)
    fill(this.color)
  }
}
```

Sound Visualization

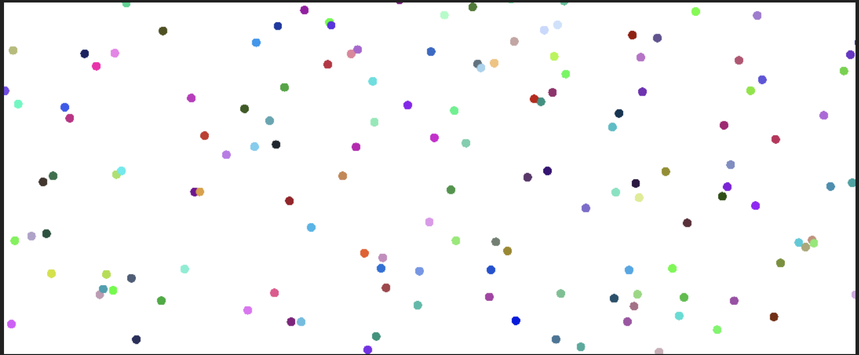
Drawing the Particles

Drawing particles to the screen using draw()



```
function draw() {
  drawParticles()
}
```

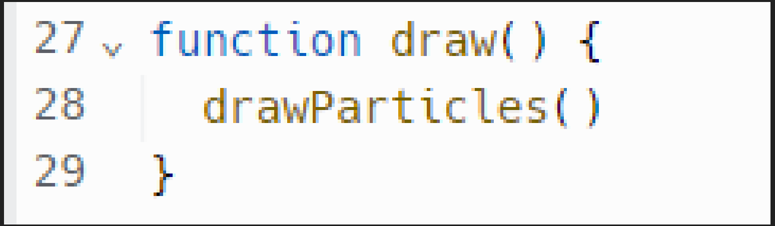
When you ▶ Run the program you should get the following output:



Sound Visualization

Drawing the Particles

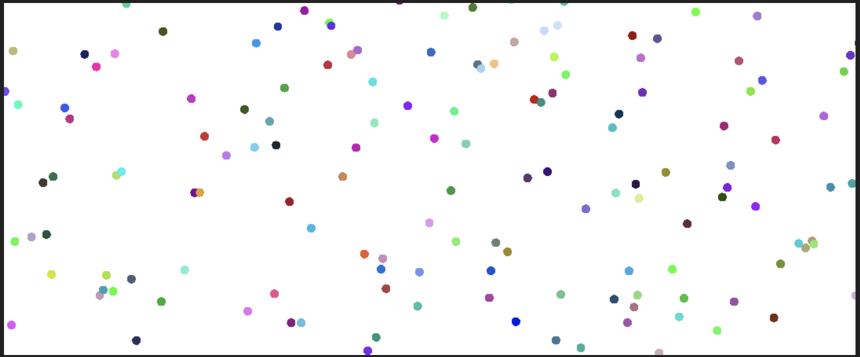
Drawing particles to the screen using draw()



```
function draw() {
    drawParticles()
}
```

(Another f() from galaxyManager.js)

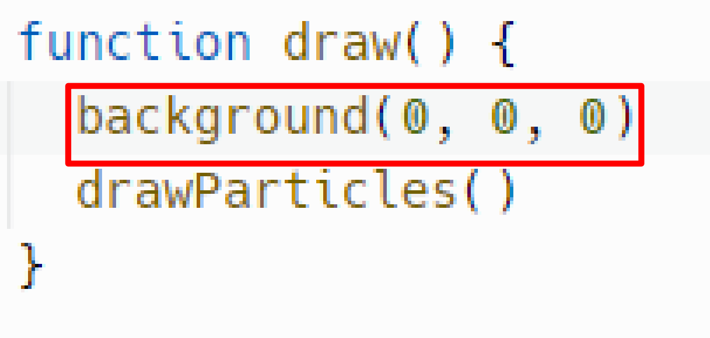
When you ▶ Run the program you should get the following output:



Sound Visualization

Changing the background

You can change the background by adding to following to draw()

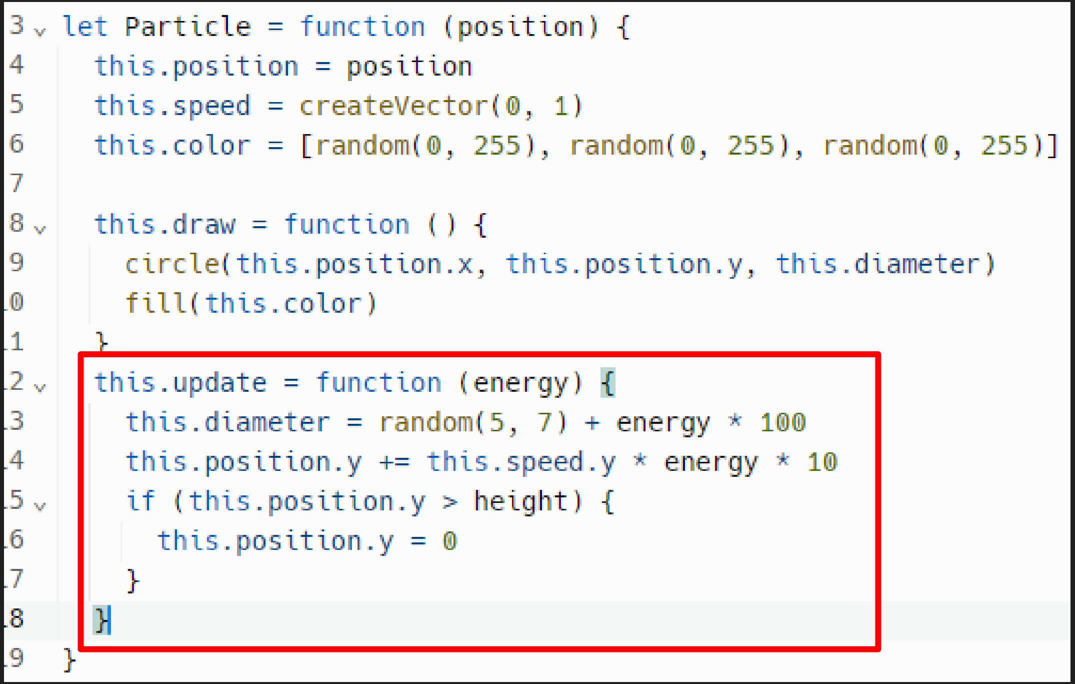


```
function draw() {
  background(0, 0, 0)
  drawParticles()
}
```

Sound Visualization

Updating the particles

Add a new method to Particle



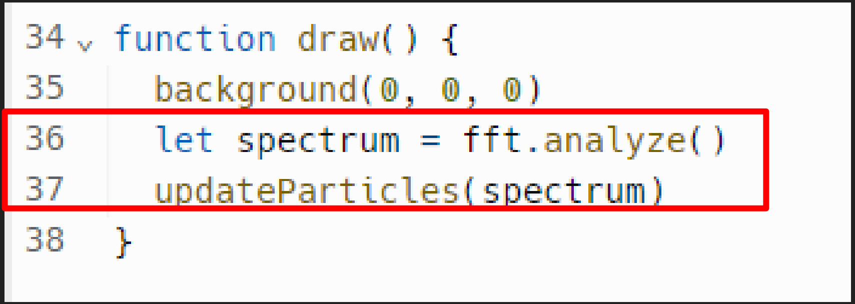
```
let Particle = function (position) {
  this.position = position
  this.speed = createVector(0, 1)
  this.color = [random(0, 255), random(0, 255), random(0, 255)]

  this.draw = function () {
    circle(this.position.x, this.position.y, this.diameter)
    fill(this.color)
  }
  this.update = function (energy) {
    this.diameter = random(5, 7) + energy * 100
    this.position.y += this.speed.y * energy * 10
    if (this.position.y > height) {
      this.position.y = 0
    }
  }
}
```

Updates a particle based on the intensity of the sound

Sound Visualization

Updating the particles in Draw()

Update the draw() and pass in the fast fourier transform information



```
function draw() {
  background(0, 0, 0)
  let spectrum = fft.analyze()
  updateParticles(spectrum)
}
```

Sound Visualization

Result

If you ▶ Run the program you should get the following result!