

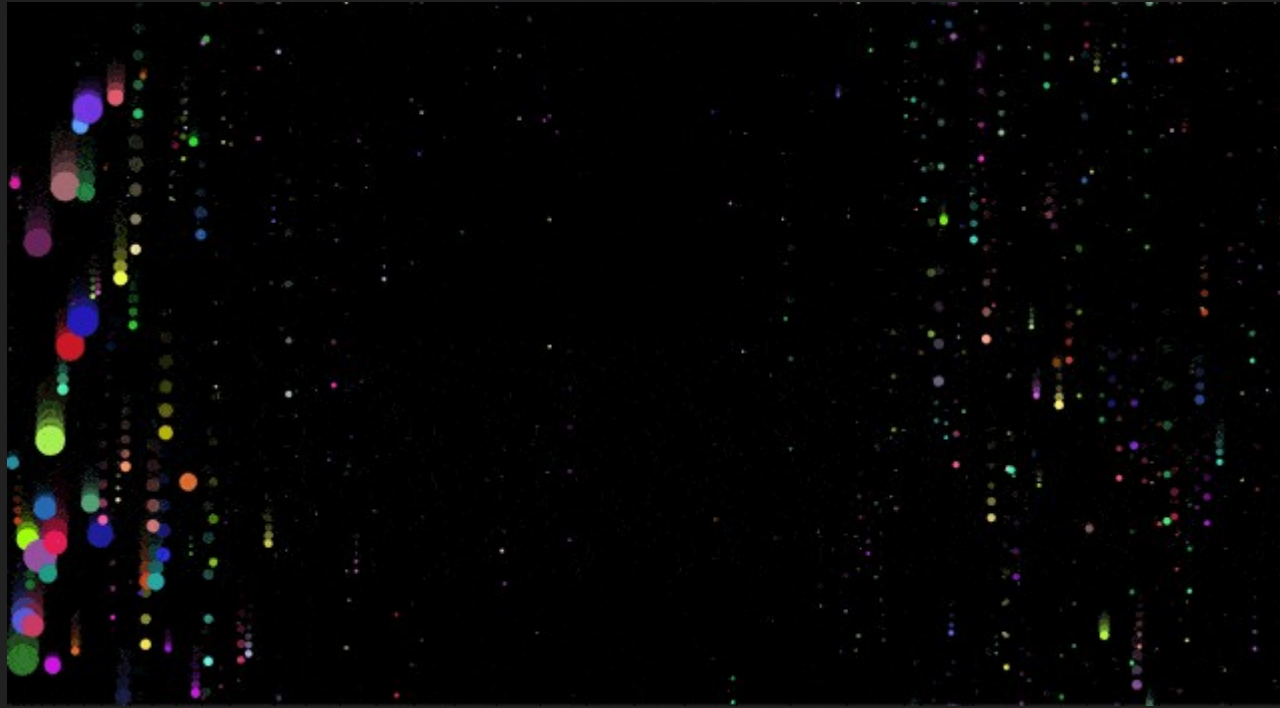
Sound Galaxy

Hack Club

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

Sound Visualization

Today's Goal:



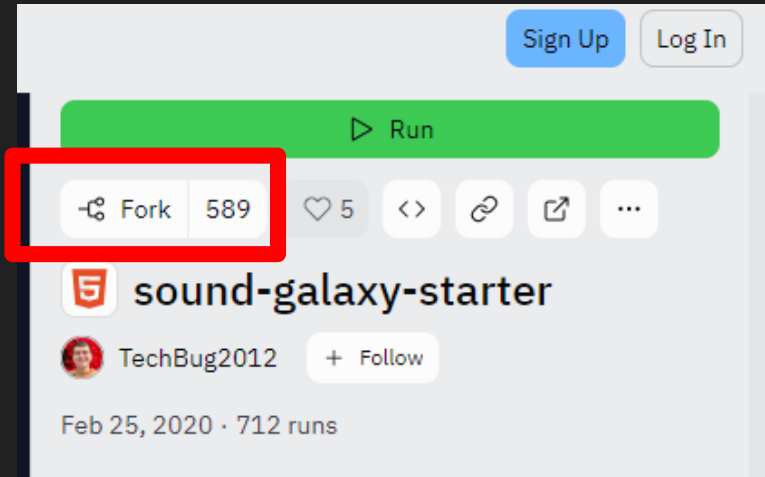
Sound Visualization

Getting Started:

1. Open the starter project:

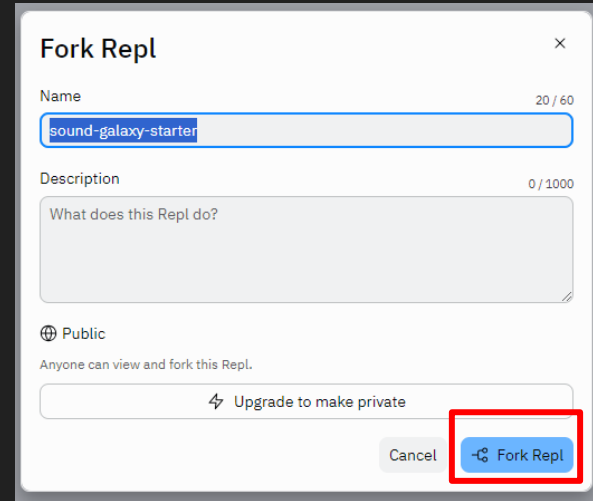
<https://replit.com/@TechBug2012/sound-galaxy-starter>

2a. Fork the Repl



The screenshot shows the Replit interface for the project 'sound-galaxy-starter' by user 'TechBug2012'. At the top right, there are 'Sign Up' and 'Log In' buttons. Below them is a green 'Run' button with a play icon. A red box highlights the 'Fork' button, which shows '589' forks, and the '5' likes button. Below the buttons, the project name 'sound-galaxy-starter' is displayed with the Replit logo. Underneath, the user profile 'TechBug2012' is shown with a '+ Follow' button. At the bottom, it indicates 'Feb 25, 2020 · 712 runs'.

2b.



The screenshot shows the 'Fork Repl' dialog box. The title is 'Fork Repl' with a close button (X) in the top right corner. The 'Name' field contains 'sound-galaxy-starter' and has a character count of '20 / 60'. The 'Description' field is empty and has a character count of '0 / 1000'. Below the description, it is set to 'Public' with a globe icon and the text 'Anyone can view and fork this Repl.'. There is an 'Upgrade to make private' button with a lock icon. At the bottom right, there are 'Cancel' and 'Fork Repl' buttons, with the 'Fork Repl' button highlighted by a red box.

Sound Visualization

Importing Libraries:

Import `p5.js` and `p5.sound.js`:

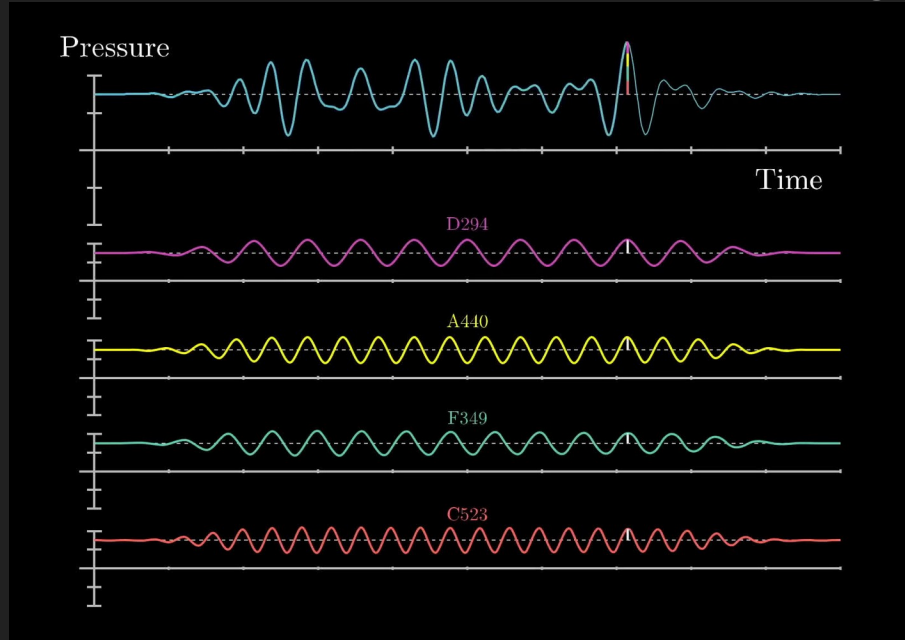
(Add to `Index.html` under just before the end of your `<head>` tag)

```
<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`:

Runs once at start

```
1 function setup() {  
2   createCanvas(windowWidth, windowHeight)  
3   noStroke()  
4  
5   let mic = new p5.AudioIn()  
6   mic.start()  
7 }  
8  
9 function draw() {}
```

Runs continuously after
`setup()`

Sound Visualization

Add the Fast Fourier Transform

Add the following to `script.js`:

```
1 let fft
2
3 function setup() {
4   createCanvas(windowWidth, windowHeight)
5   noStroke()
6
7   let mic = new p5.AudioIn()
8   mic.start()
9
10  fft = new p5.FFT()
11  fft.setInput(mic)
12 }
13
14 function draw() {}
```

Sound Visualization

Creating a Particle

Add the following above the `setup()`

```
3 let Particle = function (position) {  
4   this.position = position  
5   this.speed = createVector(0, 1)  
6   this.color = [random(0, 255), random(0, 255), random(0, 255)]  
7 }  
8
```

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()`

```
8
9  function setup() {
10   createCanvas(windowWidth, windowHeight)
11   noStroke()
12
13   let mic = new p5.AudioIn()
14   mic.start()
15
16   fft = new p5.FFT()
17   fft.setInput(mic)
18
19   positionParticles()
20 }
```

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()`

```
3 let Particle = function (position) {  
4   this.position = position  
5   this.speed = createVector(0, 1)  
6   this.color = [random(0, 255), random(0, 255), random(0, 255)]  
7  
8   this.draw = function () {  
9     circle(this.position.x, this.position.y, this.diameter)  
10    fill(this.color)  
11  }  
12 }
```

Sound Visualization

Drawing the Particles

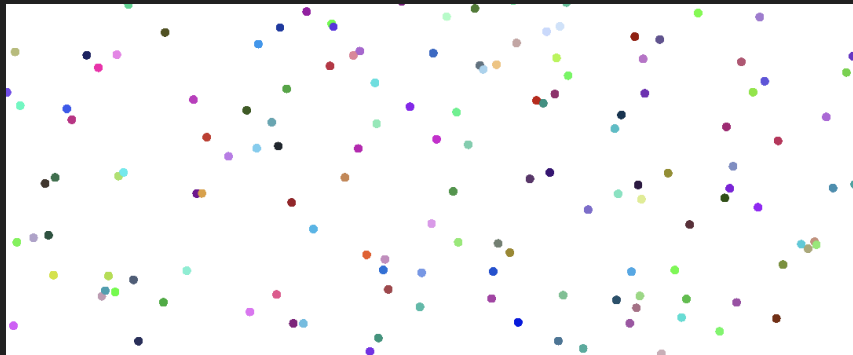
Drawing particles to the screen using `draw()`

```
27 v function draw() {  
28   drawParticles()  
29 }
```

When you

▶ Run

the program you should get the following output:



Sound Visualization

Drawing the Particles

Drawing particles to the screen using `draw()`

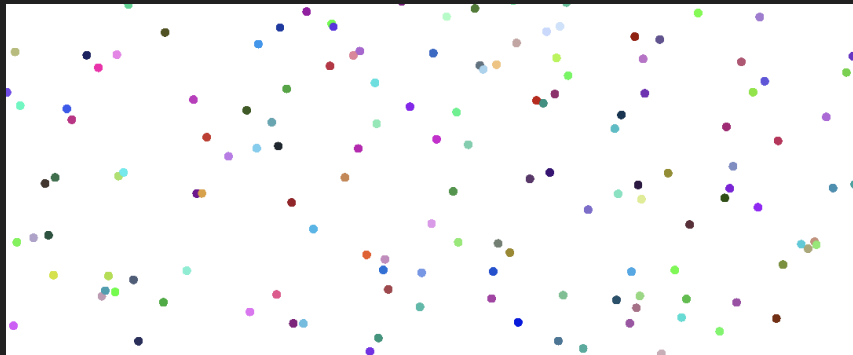
```
27 ✓ function draw() {  
28   drawParticles()  
29 }
```

(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`

```
3 let Particle = function (position) {
4   this.position = position
5   this.speed = createVector(0, 1)
6   this.color = [random(0, 255), random(0, 255), random(0, 255)]
7
8   this.draw = function () {
9     circle(this.position.x, this.position.y, this.diameter)
10    fill(this.color)
11  }
12
13  this.update = function (energy) {
14    this.diameter = random(5, 7) + energy * 100
15    this.position.y += this.speed.y * energy * 10
16    if (this.position.y > height) {
17      this.position.y = 0
18    }
19  }
20 }
```

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

```
34 function draw() {  
35   background(0, 0, 0)  
36   let spectrum = fft.analyze()  
37   updateParticles(spectrum)  
38 }
```

Sound Visualization

Result

If you



the program you should get the following result!

