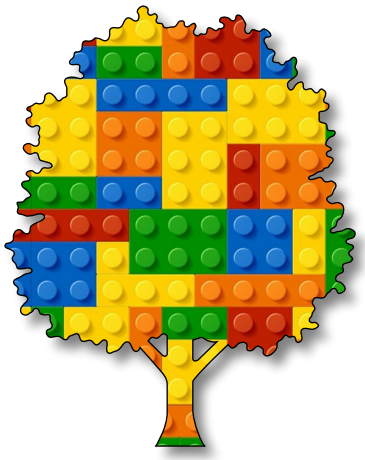
Algorithmic Art

Module F

Unit #5

the sound of music



Content

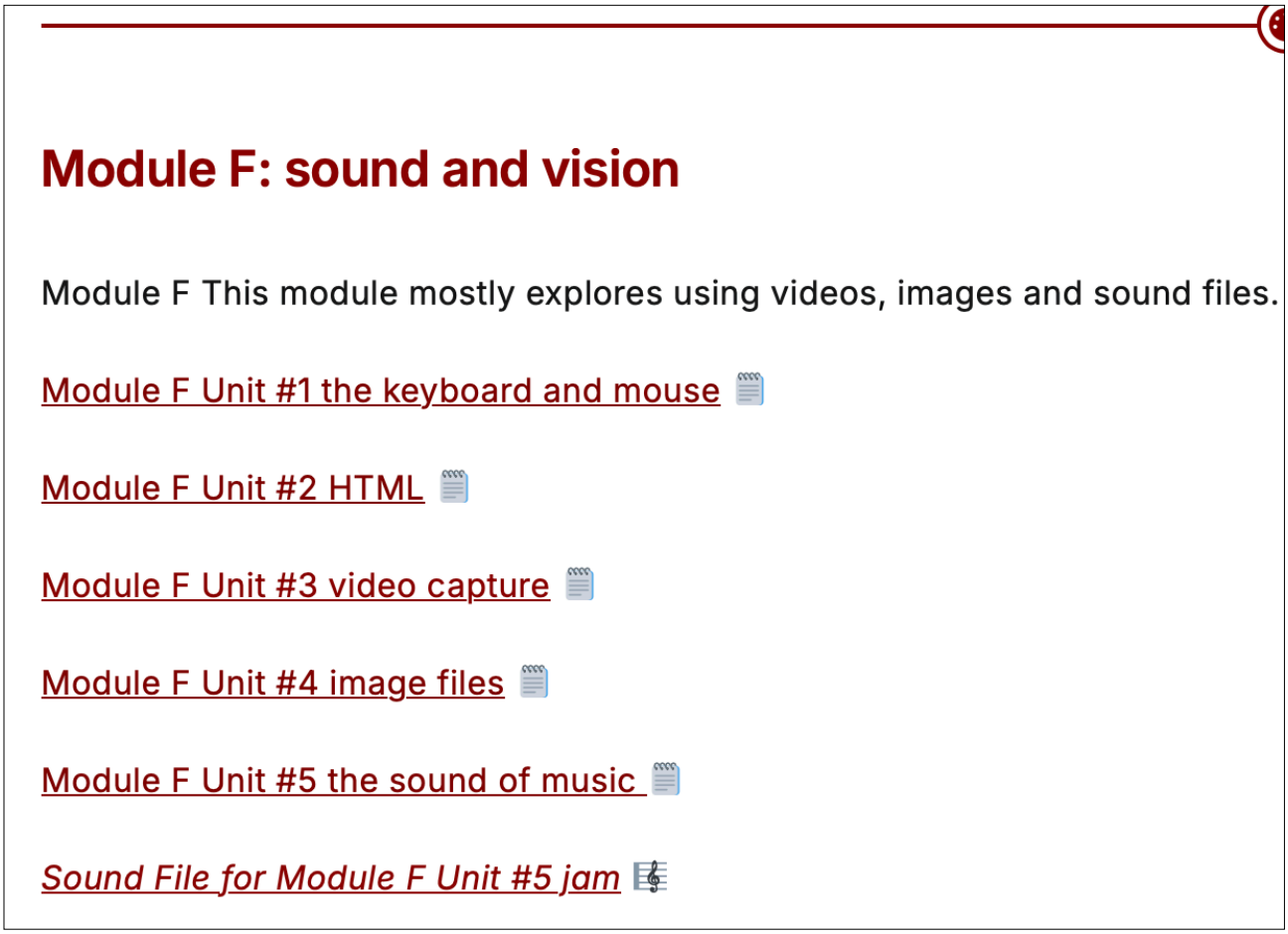
Module F Unit #5 the sound of music

The music I have provided is called Tangerine Jam by Symplocarpush.

It was downloaded for free from: www.freemusicarchive.org
for non-commercial use. You can choose from a range of styles, but be aware that the size of the file matters; if it is too big, it will not upload because there is a limit on the size of the file.

Download the sound file from my website; just click on the link.

Figure 1: download the jam sound file (website)



Upload the MP3 file as you have done with other types of files. Rename it `jam` or anything you want.

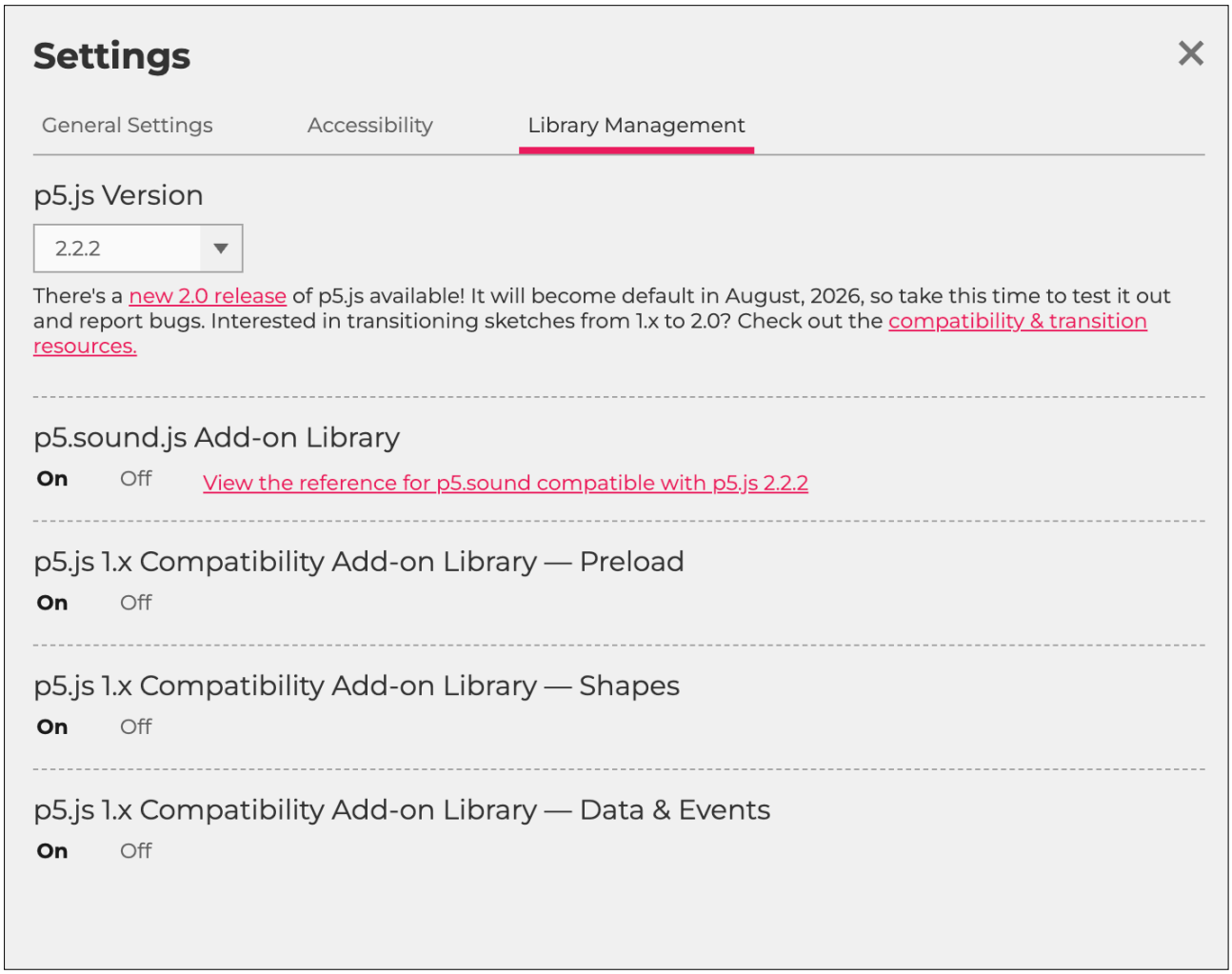
This unit covers aspects such as visualising music and the microphone sound. Although we will be working with `v2` of `p5.js`, we do need to add a library that will allow us some compatibility with p5.js version 1. This is easy to do; remember to do it, even especially if you start a new sketch.

Click on the button that indicates the version you are using; yours might be `1.11.13` or something similar. See Fig. 2.

Figure 2: the version button (top left)

Now select the p5.sound.js Add-on Library in `Settings`, as shown in Fig 3.

Figure 3: settings



Sketch F5.1 uploading

The piece of music I have provided is approximately 24 seconds long. I have called it jam for simplicity. You upload it just the same as a video and an image.

```
let music


async function setup()
{
  createCanvas(400, 400)
  music = await createAudio('jam.mp3')
}


function draw()
{
  background('darkred')
}
```

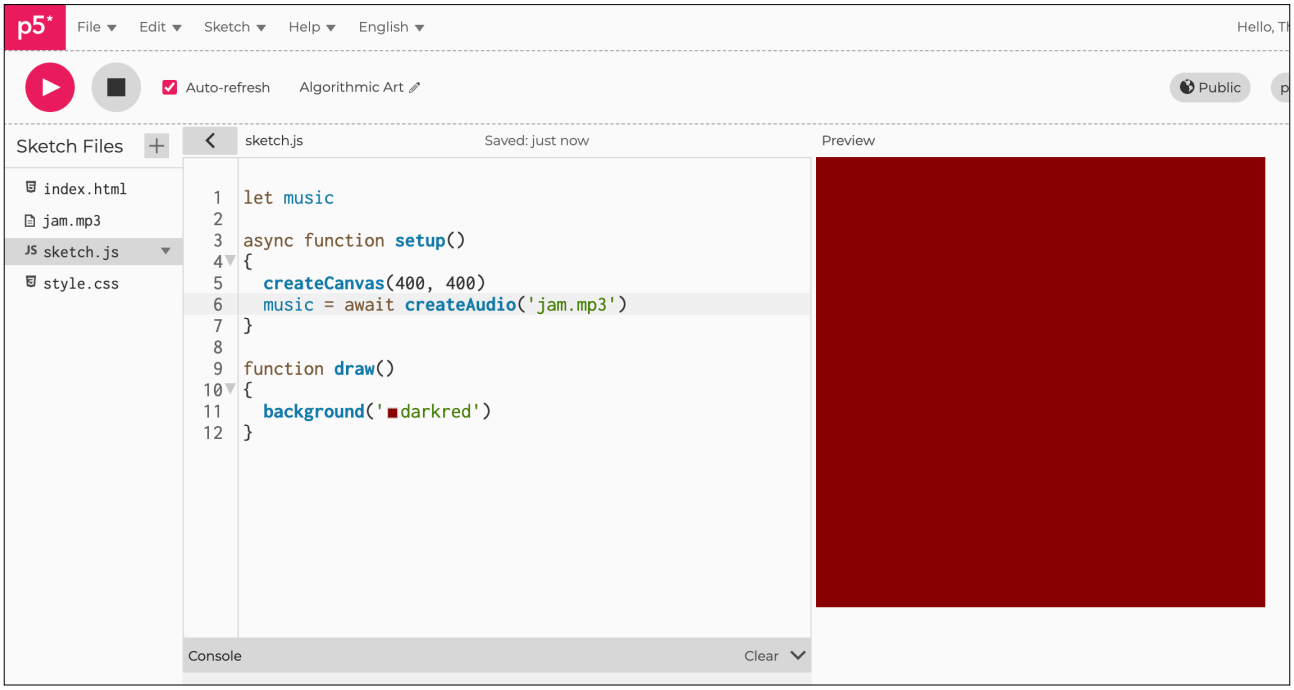
🗒️ Notes

You can't hear anything yet.

🌻 Challenge

You can use your own piece of music if you wish.

Figure F5.1


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
Public
Sketch Files
index.html
jam.mp3
sketch.js
style.css
sketch.js
Saved: just now
Preview
let music
async function setup()
{
createCanvas(400, 400)
music = await createAudio('jam.mp3')
}
function draw()
{
background('darkred')
}
Console
Clear

Sketch F5.2 play automatically

As soon as you run the code, the music will start.

```
let music


async function setup()
{
  createCanvas(400, 400)
  music = await createAudio('jam.mp3')
  music.play()
}


function draw()
{
  background('darkred')
}
```

🗒️ Notes

The `play()` function needs to be in the same function as the music download, or you will get an error message.

🌻 Challenge

Try it in `draw()`.

Sketch F5.3 music loop on mouse

❗ Remove `play()` function
Introducing `loop()` and `stop()`.

```
let music

async function setup()
{
  createCanvas(400, 400)
  music = await createAudio('jam.mp3')
}

function draw()
{
  background('darkred')
}

function mousePressed()
{
  music.loop()
}

function mouseReleased()
{
  music.stop()
}
```

🗒️ Notes

Click the canvas (and hold the button down), release to stop.

🌻 Challenge

Replace `music.loop()` with `music.play()`.

Sketch F5.4 slider for rate

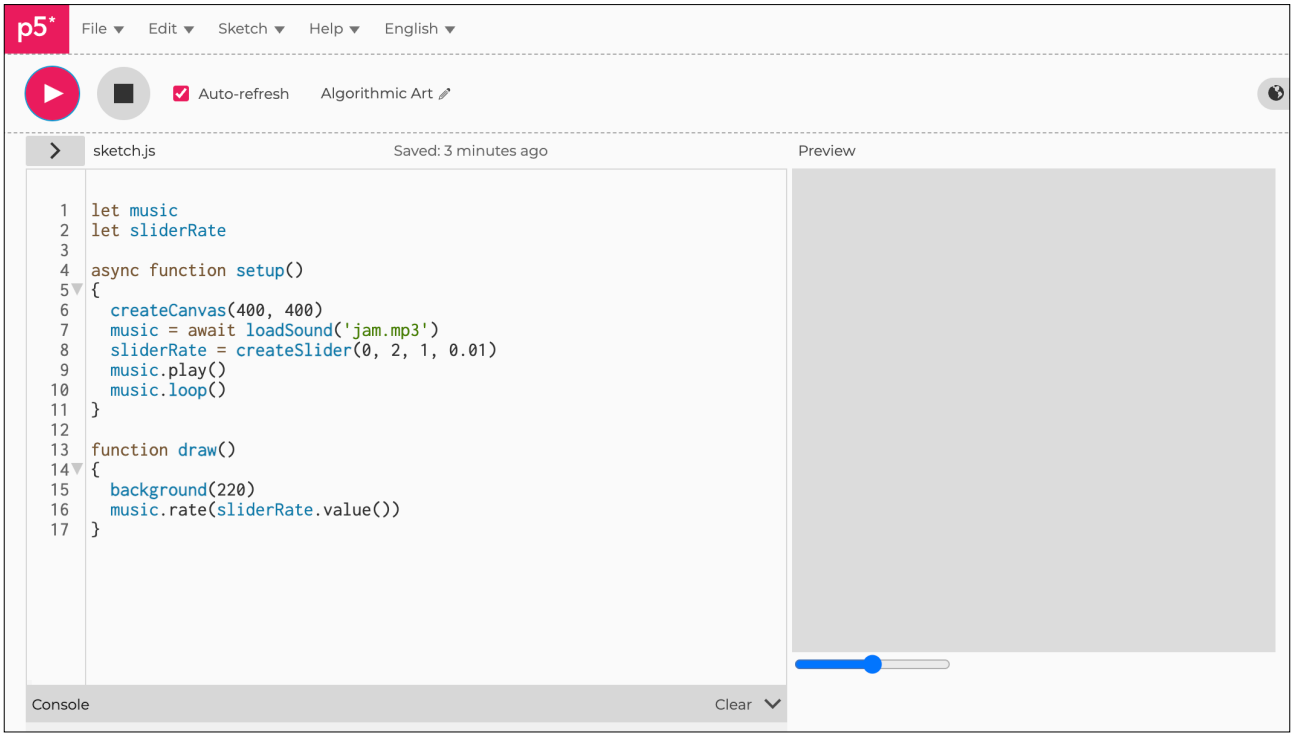
The volume slider is removed and a `rate` slider is added. The `rate` is the speed of playback. The default is `1`, half speed is `0.5`, double is `2`, etc.

```
let music
let sliderRate

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  sliderRate = createSlider(0, 2, 1, 0.01)
  music.play()
  music.loop()
}

function draw()
{
  background(220)
  music.rate(sliderRate.value())
}
```

🗒️ Notes

As you move the slider, you should hear the music speed up or slow down.

Figure F5.4


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 3 minutes ago
Preview
let music
let sliderRate
async function setup()
{
createCanvas(400, 400)
music = await loadSound('jam.mp3')
sliderRate = createSlider(0, 2, 1, 0.01)
music.play()
music.loop()
}
function draw()
{
background(220)
music.rate(sliderRate.value())
}
Console
Clear

Sketch F5.5 toggle play/pause

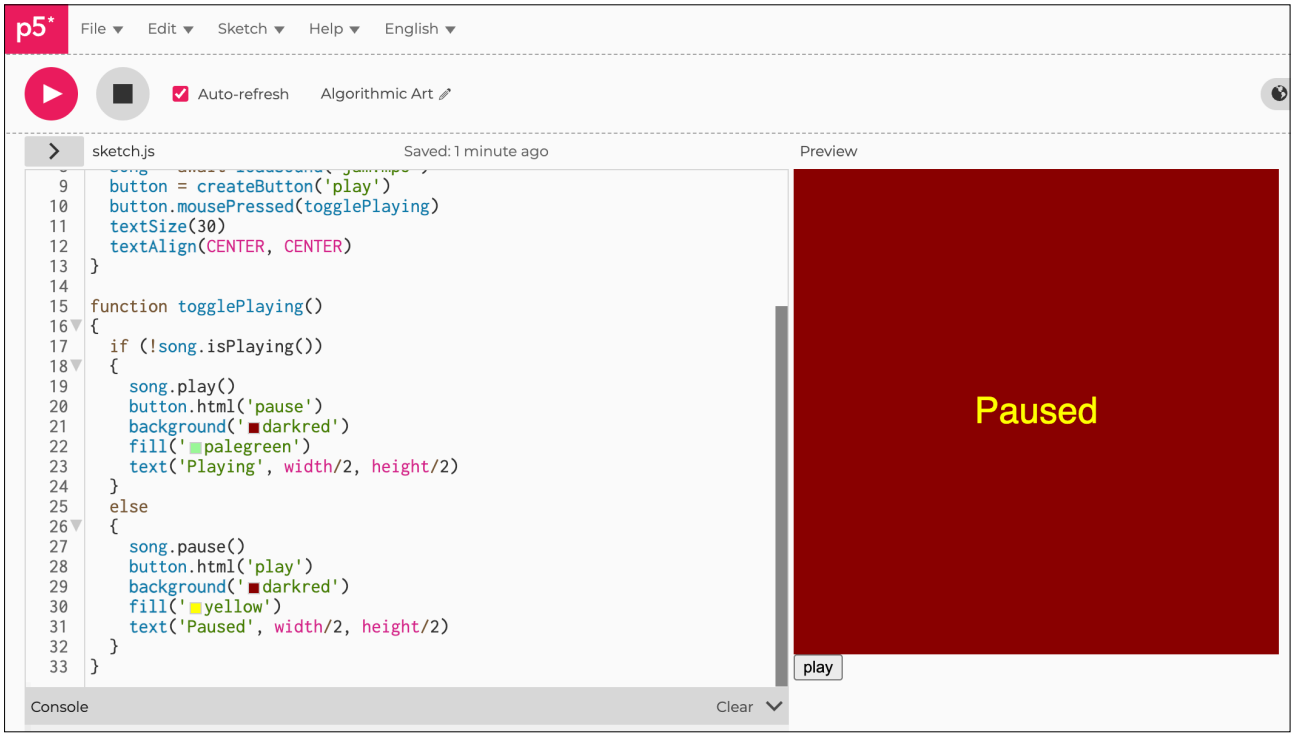
Just a simple toggle button.

```
let song
let button

async function setup()
{
  createCanvas(400, 400)
  background('darkred')
  song = await loadSound('jam.mp3')
  button = createButton('play')
  button.mousePressed(togglePlaying)
  textSize(30)
  textAlign(CENTER, CENTER)
}

function togglePlaying()
{
  if (!song.isPlaying())
  {
    song.play()
    button.html('pause')
    background('darkred')
    fill('palegreen')
    text('Playing', width/2, height/2)
  }
  else
  {
    song.pause()
    button.html('play')
    background('darkred')
    fill('yellow')
    text('Paused', width/2, height/2)
  }
}
```

Notes

Click to play or pause.

Figure F5.5



Sketch F5.6 jump

Jumps to a specific time in seconds. In this case, it jumps to the 20th second of the 23-second-long track.

```
let music
let button
let jumpButton

async function setup()
{
  createCanvas(400, 400)
  background(220)
  music = await loadSound('jam.mp3')
  button = createButton('play')
  button.mousePressed(togglePlaying)
  jumpButton= createButton('jump')
  jumpButton.mousePressed(jumpMusic)
}

function jumpMusic()
{
  music.jump(20)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
}
```

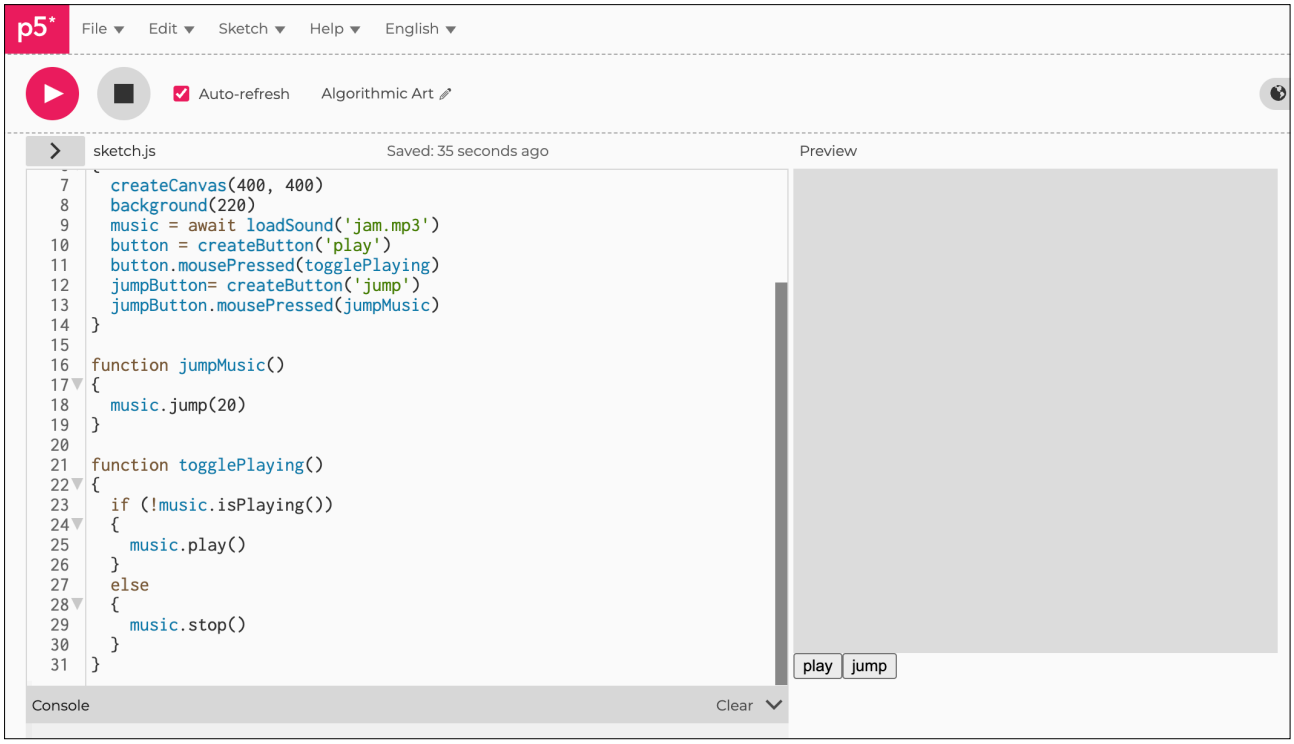
🗒️ Notes

It might be hard to hear the jump.

🌻 Challenge

Try several jump buttons.

Figure F5.6



Sketch F5.7 music length

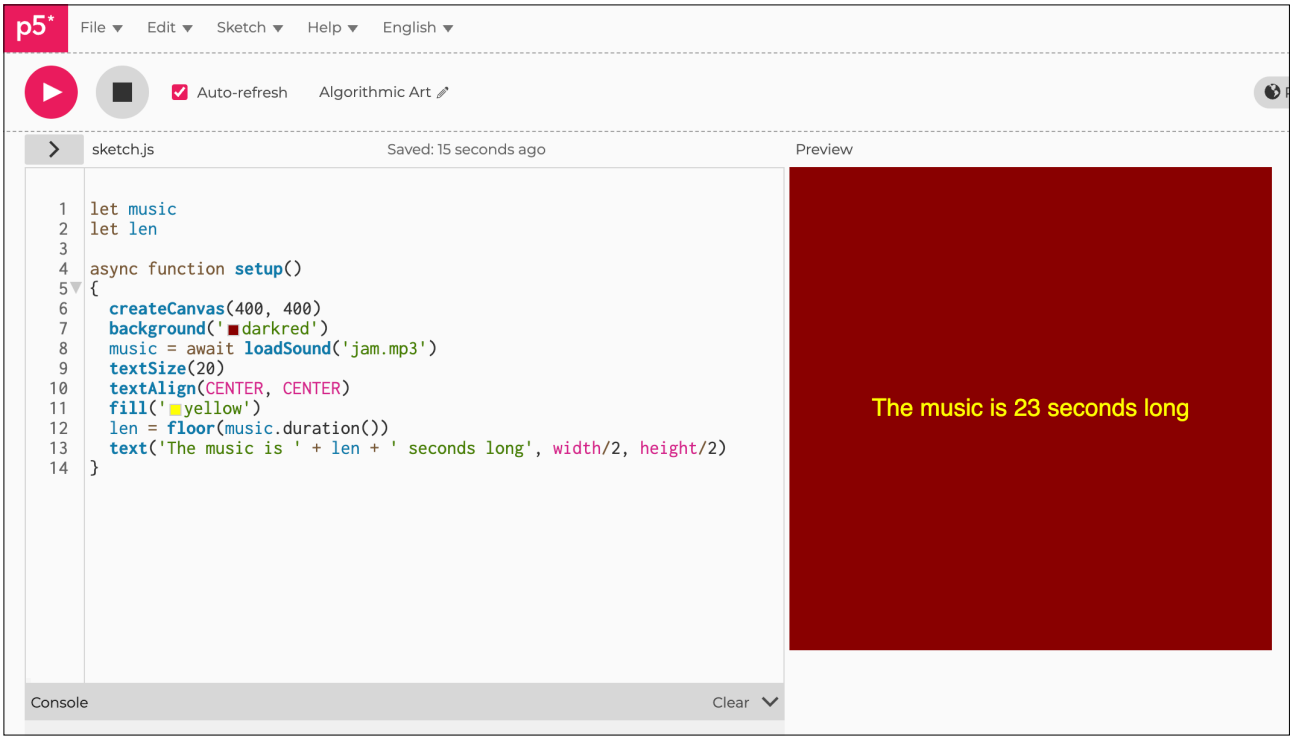
Using this function, we can measure the length of the piece of music or sound file.

```
let music
let len

async function setup()
{
  createCanvas(400, 400)
  background('darkred')
  music = await loadSound('jam.mp3')
  textSize(20)
  textAlign(CENTER, CENTER)
  fill('yellow')
  len = floor(music.duration())
  text('The music is ' + len + ' seconds long', width/2, height/2)
}
```

Notes

You should get the correct value of 23 seconds.

Figure F5.7



Get the volume

To get the volume, we need to use the p5 `Amplitude()` function. We need to connect the music to the `Amplitude()` function.

Sketch F5.8 getting the volume

We create two variables, `amp` and `vol`. We connect the amplitude to the music (sound file). From the amplitude, we get the volume (level). This is then fed into the diameter of the circle. The volume (`vol`) values can be quite small (between `0` and `1`), so we multiply by `500` just to make it a bit more dramatic.

```
let music
let amp
let vol

async function setup()
{
  createCanvas(400, 400)

  music = await loadSound('jam.mp3')
  amp = new p5.Amplitude()
  music.connect(amp)
  music.play()
}

function draw()
{
  background('darkred')
  fill('yellow')
  vol = amp.getLevel()
  circle(200, 200, vol * 500)
}
```

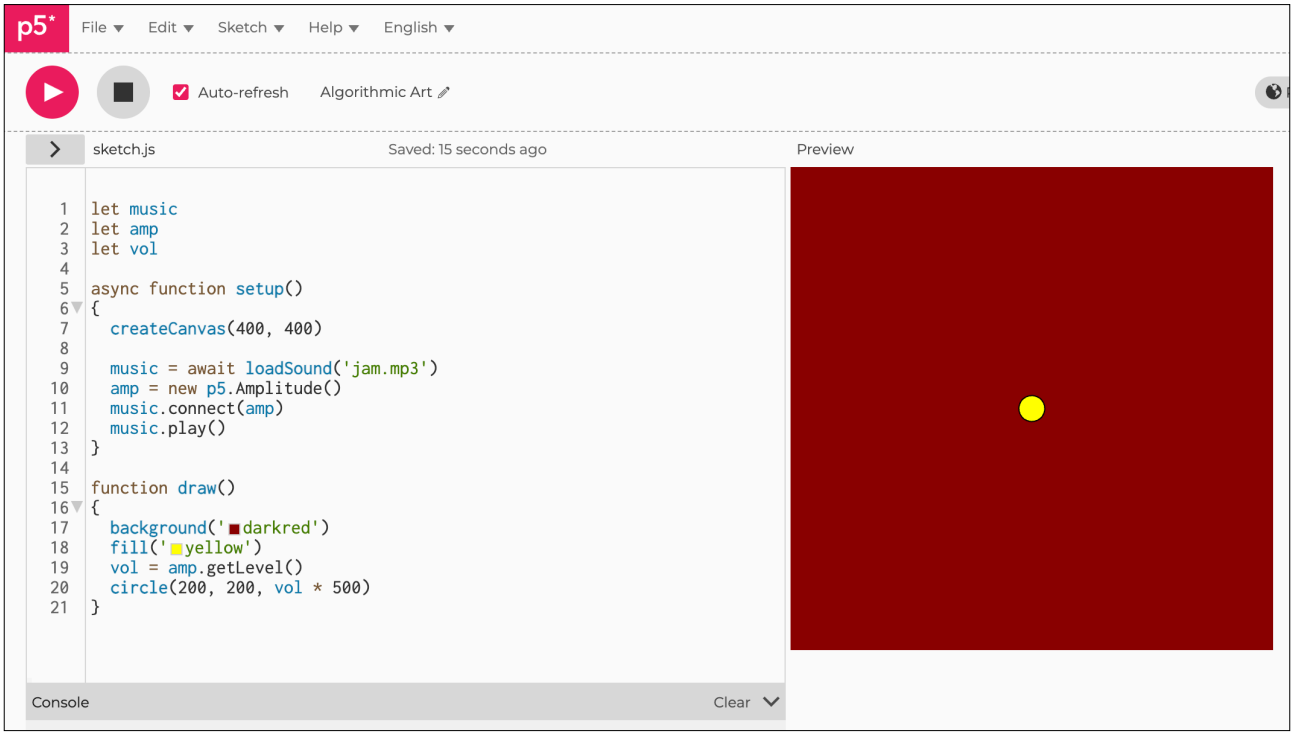
🗒️ Notes

The circle should rapidly enlarge and constrict as the music is played.

🌻 Challenge

Times it by the width instead.

Figure F5.8


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 15 seconds ago
Preview
let music
let amp
let vol
async function setup()
{
createCanvas(400, 400)
music = await loadSound('jam.mp3')
amp = new p5.Amplitude()
music.connect(amp)
music.play()
}
function draw()
{
background('darkred')
fill('yellow')
vol = amp.getLevel()
circle(200, 200, vol * 500)
}
Console
Clear

Sketch F5.9 toggle button

We add a button to toggle play or not play. We have a conditional if statement to check if it is not playing already. We move the `play()` function into the new function `togglePlaying()`.

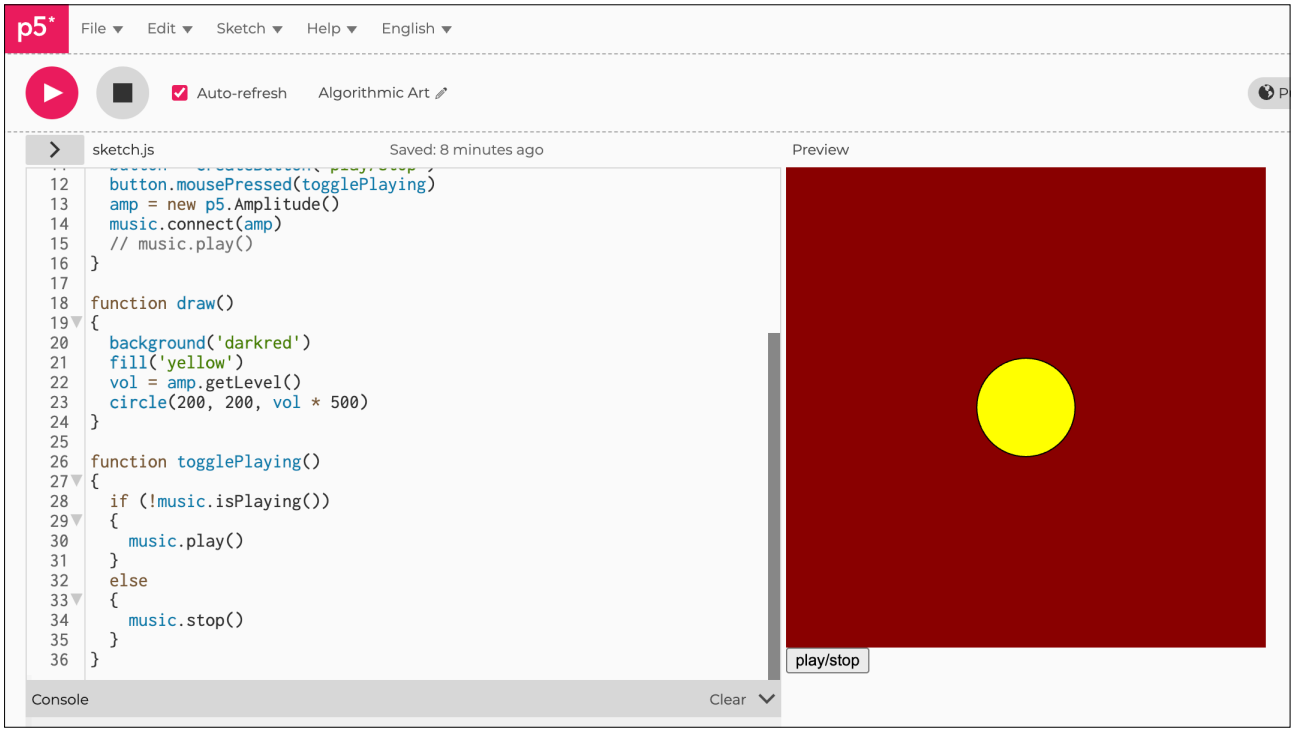
```
let music
let amp
let vol
let button

async function setup()
{
  createCanvas(400, 400)

  music = await loadSound('jam.mp3')
  button = createButton('play/stop')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
  // music.play()
}

function draw()
{
  background('darkred')
  fill('yellow')
  vol = amp.getLevel()
  circle(200, 200, vol * 500)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
```

```
  {
    music.stop()
  }
}
```

Notes

This is nothing new.

Figure F5.9



Sketch F5.10 a bit of style

Adding a bit of style.

```
let music
let amp
let vol
let button
let rDot
let x = 200

async function setup()
{
  createCanvas(400, 400)
  background('darkred')
  music = await loadSound('jam.mp3')
  button = createButton('play/stop')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
  noFill()
}

function draw()
{
  noStroke()
  fill(255, 50)
  vol = amp.getLevel()
  rDot = random(-5, 5)
  x += rDot
  circle(x, height + 20 - vol*2*height, 20)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
```

```
  }
  else
  {
    music.stop()
    x = 200
  }
}
```

🗒️ Notes

Should have a nice effect.

🌻 Challenge

Create something with lots of colour, maybe linked to the volume.

Figure F5.10

Sketch F5.11 newish sketch

! Remove unnecessary code and put background into the `draw()` function.

```
let music
let amp
let vol
let button

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
}

function draw()
{
  background('darkred')
  fill('yellow')
  vol = amp.getLevel()
  circle(width/2, height/2, vol * width)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
```

```
}
```

Notes

The circle will now respond to the music.

Figure F5.11

Sketch F5.12 an array of data

We want to create an analysis graph of the sound file. To do this, we need to create an array to store all this data. We `console.log()` the length to make sure it is actually collecting the data.

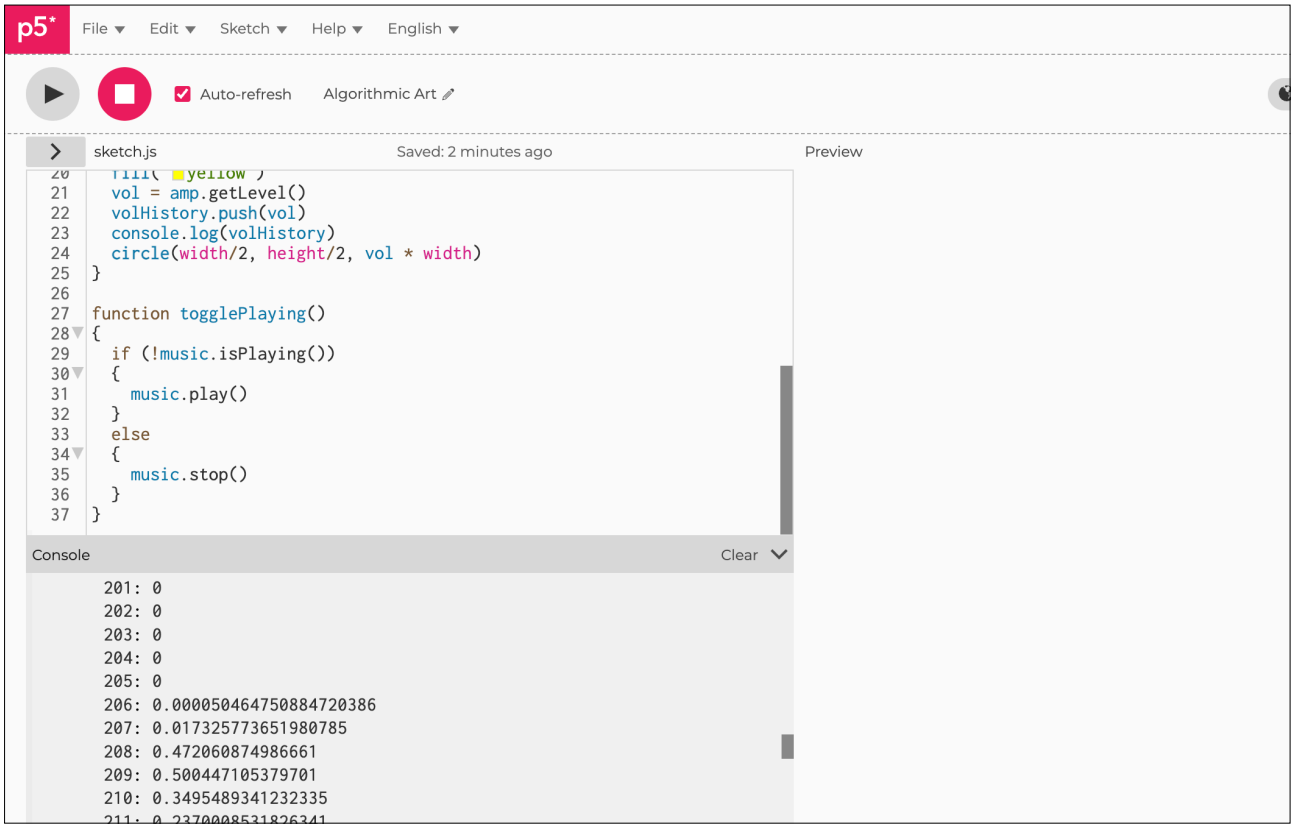
```
let music
let amp
let vol
let button
let volHistory = []

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
}

function draw()
{
  background('darkred')
  fill('yellow')
  vol = amp.getLevel()
  volHistory.push(vol)
  console.log(volHistory)
  circle(width/2, height/2, vol * width)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
```

```
  else
  {
    music.stop()
  }
}
```

🗒️ Notes

We get a result in the console showing that we are increasing the number of data points; this continues even though the music stops because it will still receive 0.

Figure F5.12



Sketch F5.13 replacing the circle

! Remove the line of code for the circle and the `console.log()`.
Instead, we are going to draw points for the values of the volume.

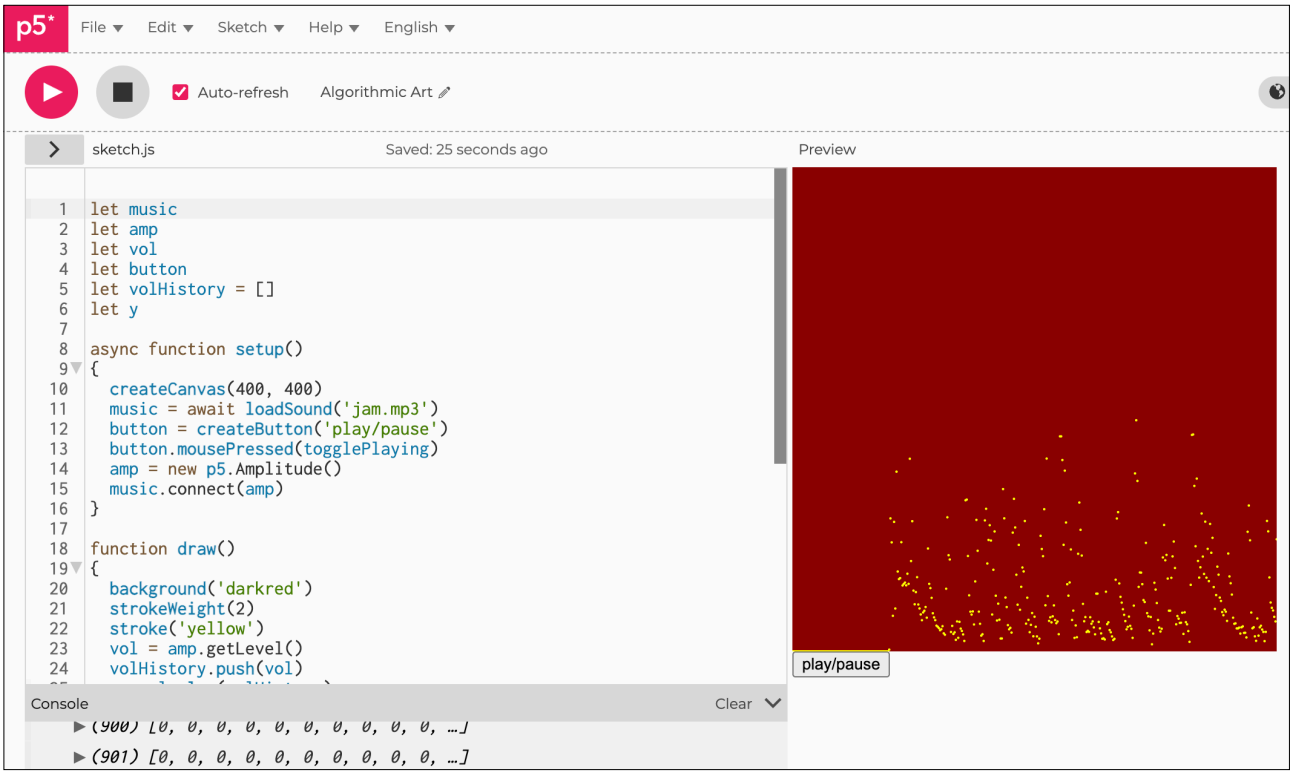
```
let music
let amp
let vol
let button
let volHistory = []
let y

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
}

function draw()
{
  background('darkred')
  strokeWeight(2)
  stroke('yellow')
  vol = amp.getLevel()
  volHistory.push(vol)
  console.log(volHistory)
  for (let i = 0; i < volHistory.length; i++)
  {
    y = map(volHistory[i], 0, 1, height, 0)
    point(i, y)
  }
}

function togglePlaying()
```

```
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
}
```

📝 Notes

We now get a series of dots for the relevant values of the volume at any one moment in time. The dots run off the edge as the music continues to play; we will fix that in due course.

Figure F5.13



Sketch F5.14 a line of dots

! Remove `console.log()` as we don't need it anymore.
Replacing the `point()` with `vertex()`, we can draw a line joining the dots.

```
let music
let amp
let vol
let button
let volHistory = []
let y

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
}

function draw()
{
  background('darkred')
  strokeWeight(2)
  stroke('yellow')
  noFill()
  vol = amp.getLevel()
  volHistory.push(vol)
  beginShape()
  for (let i = 0; i < volHistory.length; i++)
  {
    y = map(volHistory[i], 0, 1, height, 0)
    vertex(i, y)
  }
  endShape()
}
```

```
function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
}
```

📝 Notes

We get a line this time but it's still static and disappears off the edge!

Figure F5.14

Sketch F5.15 a continuous line

Now the graph scrolls.

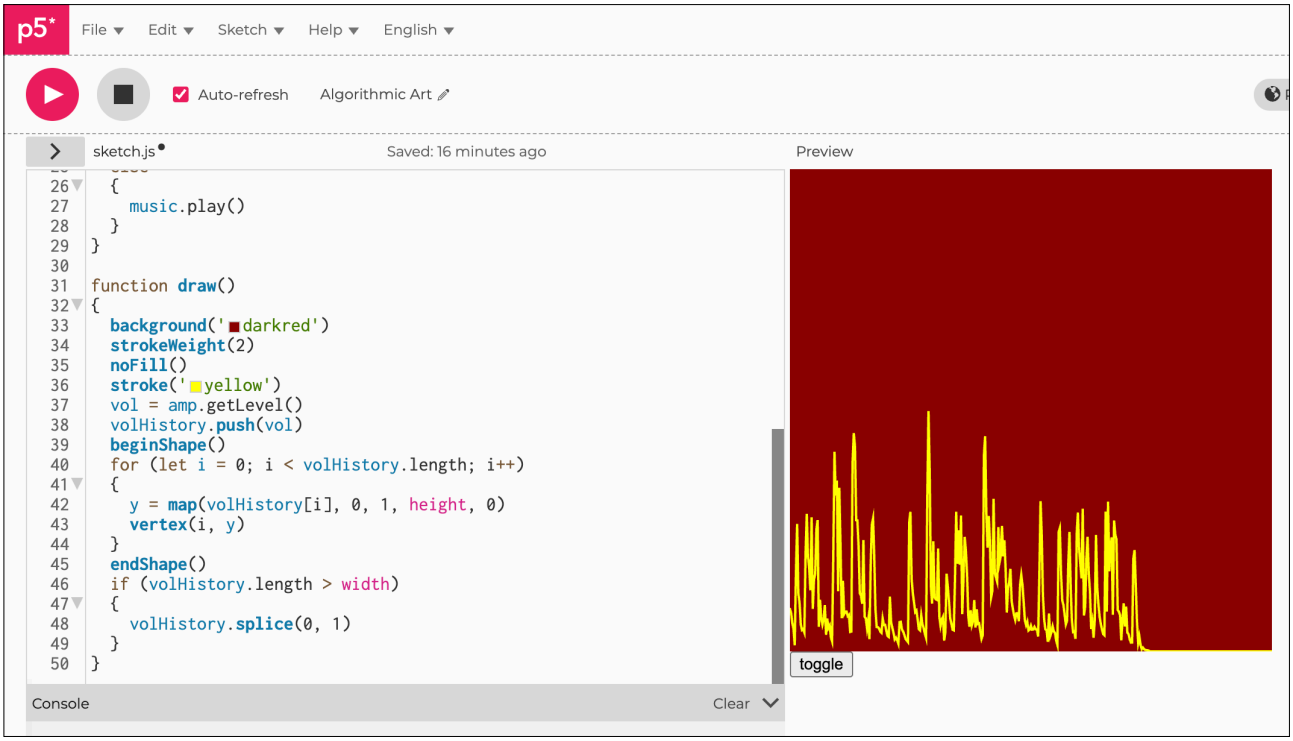
```
let music
let amp
let vol
let button
let volHistory = []
let y

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
}

function draw()
{
  background('darkred')
  strokeWeight(2)
  stroke('yellow')
  noFill()
  vol = amp.getLevel()
  volHistory.push(vol)
  beginShape()
  for (let i = 0; i < volHistory.length; i++)
  {
    y = map(volHistory[i], 0, 1, height, 0)
    vertex(i, y)
  }
  endShape()
  if (volHistory.length > width)
  {
```

```
    volHistory.splice(0, 1)
  }
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
}
```

🗒️ Notes

It continues to move even when the music stops.

Figure F5.15



Sketch F5.16 a rotary version

The line is now rotating.

```
let music
let amp
let vol
let button
let volHistory = []
let y
let r
let x

async function setup()
{
  createCanvas(400, 400)
  music = await loadSound('jam.mp3')
  button = createButton('play/pause')
  button.mousePressed(togglePlaying)
  amp = new p5.Amplitude()
  music.connect(amp)
  angleMode(DEGREES)
}

function draw()
{
  background('darkred')
  strokeWeight(2)
  stroke('yellow')
  noFill()
  vol = amp.getLevel()
  volHistory.push(vol)
  translate(width/2, height/2)
  beginShape()
  for (let i = 0; i < volHistory.length; i++)
  {
    r = map(volHistory[i], 0, 1, 10, width/2)
```

```
    x = r * sin(i)
    y = r * cos(i)
    vertex(x, y)
  }
  endShape()
  if (volHistory.length > 360)
  {
    volHistory.splice(0, 1)
  }
  line(0, 0, 0, width/2)
  fill('darkred')
  circle(0, 0, 20)
}

function togglePlaying()
{
  if (!music.isPlaying())
  {
    music.play()
  }
  else
  {
    music.stop()
  }
}
```

🗒️ Notes

This has a spinning graph which has to start and finish somewhere. It looks a bit odd, so I have drawn a line to hide the jump.

🌻 Challenge

Can you find a better way of representing the sound file?

Figure F5.16


p5*
File
Edit
Sketch
Help
English
Auto-refresh
Algorithmic Art
sketch.js
Saved: 35 seconds ago
Preview
x = r * sin(i)
y = r * cos(i)
vertex(x, y)
}
endShape()
if (volHistory.length > 360)
{
volHistory.splice(0, 1)
}
line(0, 0, 0, width/2)
fill('darkred')
circle(0, 0, 20)
}
function togglePlaying()
{
if (!music.isPlaying())
{
music.play()
}
else
{
music.stop()
}
}
play/pause
Console
Clear