Section B
Unit #4
Visual

Content

Unit #4 Visualisation

Introduction

This section explores sound and music. For this you will need a microphone, most machines have them built in. You will be creating and graphing sound using the microphone and downloading free music, and from this you can create visualisation of the sounds and/or music.

This brings in two very useful topics that have many applications. If nothing else it is fun and interesting.

For where we just using sounds you will need a built in microphone (most computers have a built in one, or if you are attaching a separate USB webcam they often have the microphone built in. If you can I would recommend getting a good quality microphone that you can add yourself (usually though a USB port). It just gives a better sound input but is not essential.

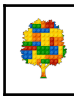
Here also you can create your own sounds and I guess you can create your own music if that is your thing. I have seen some examples where you can create a simple keyboard once you know the frequency of the different notes.

Visualisation

Unit #4 Visualisation

Introduction

This unit looks at using a microphone and visualising music

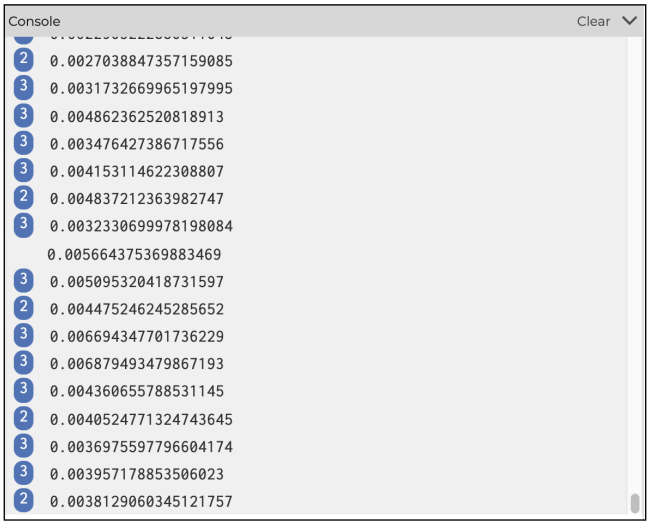


Sketch B4.1 microphone (part 1)

You may need to give p5.js permission to use the microphone on your computer. This sketch will give you the volume of your microphone. Delete all the previous code and type the following in

```
let mic
let vol

function setup()
{
  createCanvas(400, 400)
  mic = new p5.AudioIn()
  mic.start()
}

function draw()
{
  background(220)
  vol = mic.getLevel()
  console.log(vol)
}
```

You will get a value for the microphone in the console



Notes

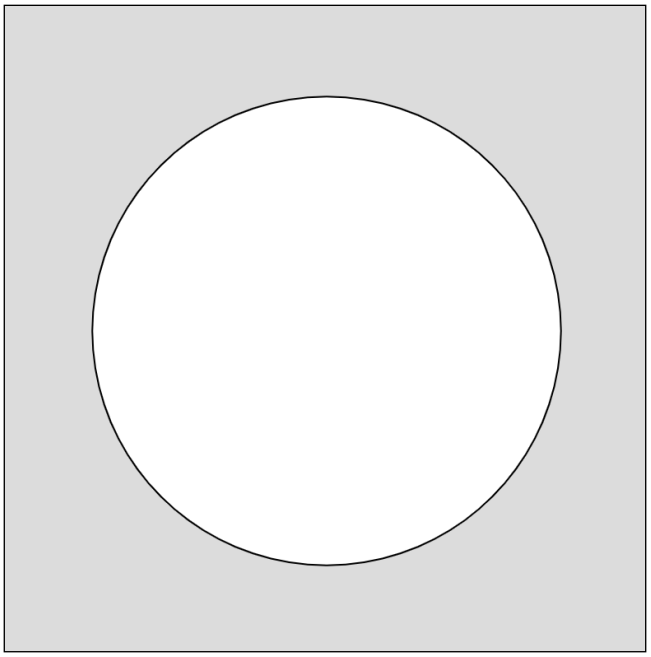
You will get a value < 1 in the console depending on how sensitive your microphone is. An external microphone will probably give you better results.

microphone (part 2)

Now we can draw a circle that responds to the microphone and have visual feedback, remove the **console.log()**

```
let mic
let vol

function setup()
{
  createCanvas(400, 400)
  mic = new p5.AudioIn()
  mic.start()
}

function draw()
{
  background(220)
  vol = mic.getLevel()
  circle(width/2, height/2, vol * width)
}
```

The circle should respond to you level of your voice



Notes

You can play around with the value of the diameter (vol) of the circle if there appears nothing or too much happening

Challenge

Could you connect the vol to the colour of a circle e.g. white to black, or other colours

Sketch B4.2 visualisation (part 1)

Creating visualisation for a piece of music. So first add an mp3 file as you have done previously. In my example I have called my folder **sounds** and the mp3 file is **music.mp3** from earlier. This is so that when you play this piece of music the microphone will pick up the music and the circle just react to it.

```
let mic
let song
let vol

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  mic = new p5.AudioIn()
  mic.start()
}

function draw()
{
  background(220)
  fill(200, 0, 0)
  vol = mic.getLevel()
  circle(width/2, height/2, vol * width)
}
```

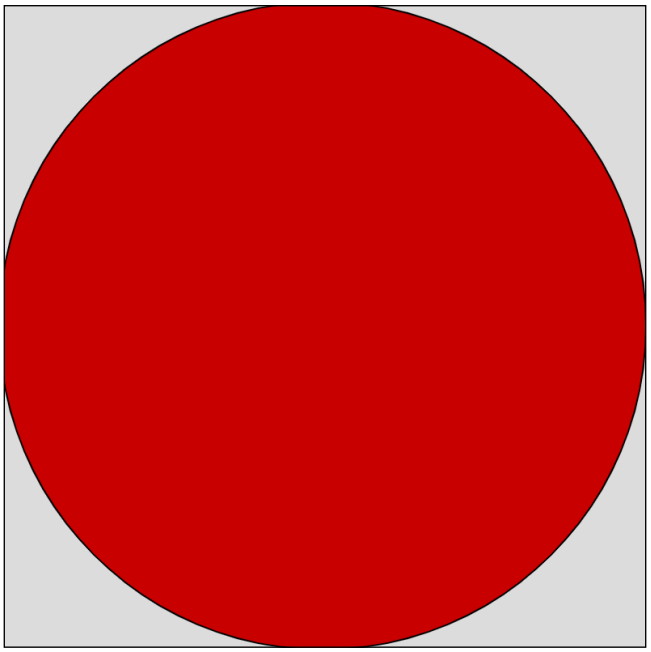
visualisation (part 2)

That is all well and good but we want it to come directly into the sketch not out through the microphone. To do so we create a button to stop and start, having removed any reference to the **mic**

```
let song
let vol
let button

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
}

function toggle()
{
  if (song.isPlaying())
  {
    song.stop()
  }
  else
  {
    song.play()
  }
```

```
}

function draw()
{
  background(220)
  fill(200, 0, 0)
  vol = 1
  circle(width/2, height/2, vol * width)
}
```

This is still not what we want, we want it to respond to the volume of the music



Notes

You have set the volume to 1 which is obviously just stays at that value, the next part will address this and show how to get the volume from the sound file itself

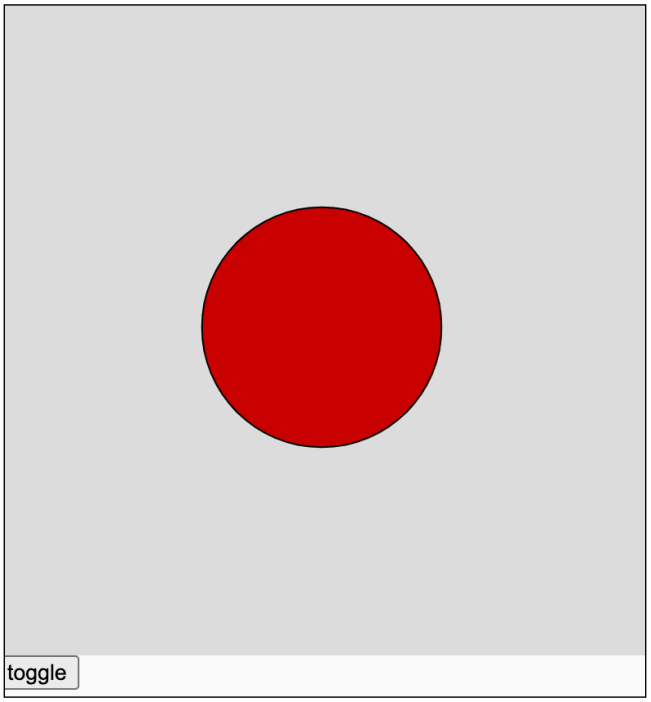
visualisation (part 3)

We next need to get the amplitude of the song for the sketch

```
let song
let vol
let button
let amp

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
}

function toggle()
{
  if (song.isPlaying())
  {
    song.stop()
  }
  else
  {
    song.play()
```

```
  }
}

function draw()
{
  background(220)
  fill(200, 0, 0)
  vol = amp.getLevel()
  circle(width/2, height/2, vol * width)
}
```

That is so much better



Notes

Now we are back on track, literally

visualisation (part 4)

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 song
let vol
let button
let amp
let volHistory = []

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
}

function toggle()
{
  if (song.isPlaying())
  {
    song.pause()
  }
  else
```

```
  {
    song.play()
    amp = new p5.Amplitude()
  }
}

function draw()
{
  background(220)
  fill(200, 0, 0)
  vol = amp.getLevel()
  volHistory.push(vol)
  console.log(volHistory.length)
  circle(width/2, height/2, vol * 500)
}
```

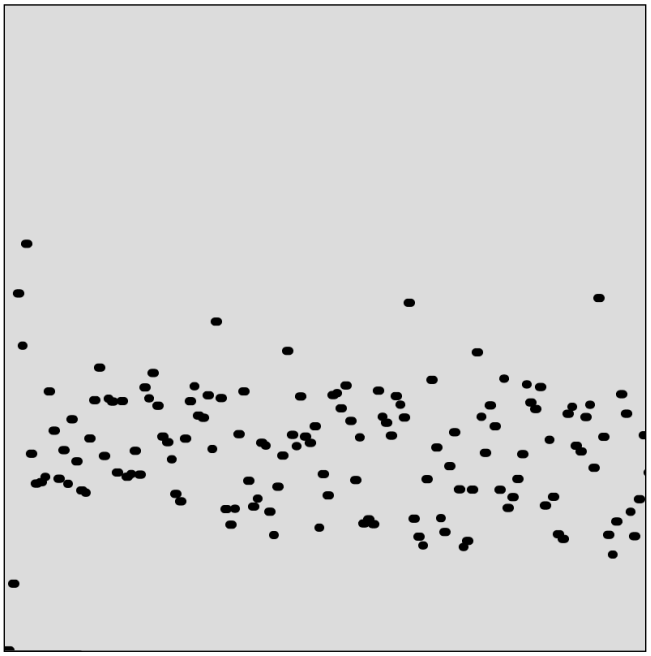
visualisation (part 5)

Instead of a circle we will draw some points (remove reference to the circle in draw())

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

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
}

function toggle()
{
  if (song.isPlaying())
  {
    song.pause()
  }
  else
```

```
  {
    song.play()
    amp = new p5.Amplitude()
  }
}

function draw()
{
  background(220)
  strokeWeight(5)
  vol = amp.getLevel()
  volHistory.push(vol)
  for (let i = 0; i < volHistory.length; i++)
  {
    y = map(volHistory[i], 0, 1, height, 0)
    point(i, y)
  }
}
```

The volume is represented by a series of dots



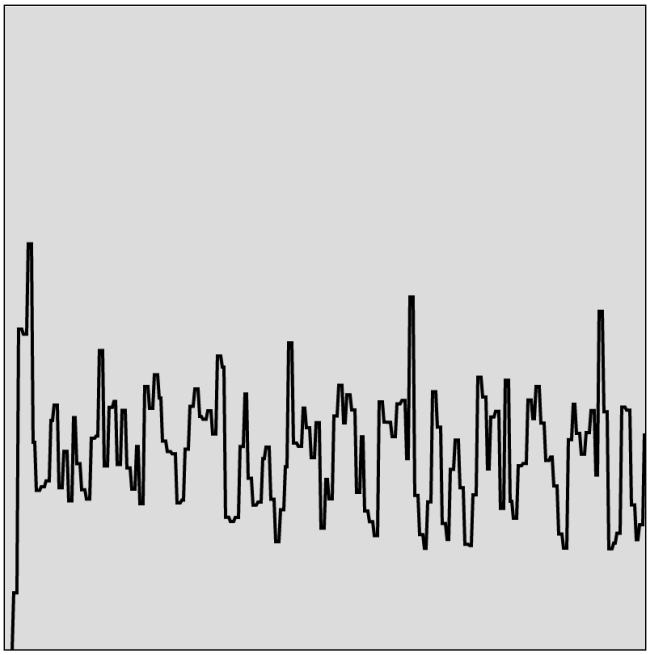
visualisation (part 6)

Then have it graph a line

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

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
}

function toggle()
{
  if (song.isPlaying())
  {
    song.pause()
  }
  else
```

```
  {
    song.play()
    amp = new p5.Amplitude()
  }
}

function draw()
{
  background(220)
  strokeWeight(2)
  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()
}
```

Now we have a line trace



visualisation (part 7)

Now to make it move continuously

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

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
}

function toggle()
{
  if (song.isPlaying())
  {
    song.pause()
  }
  else
```

```
  {
    song.play()
    amp = new p5.Amplitude()
  }
}

function draw()
{
  background(220)
  strokeWeight(1)
  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)
  }
}
```

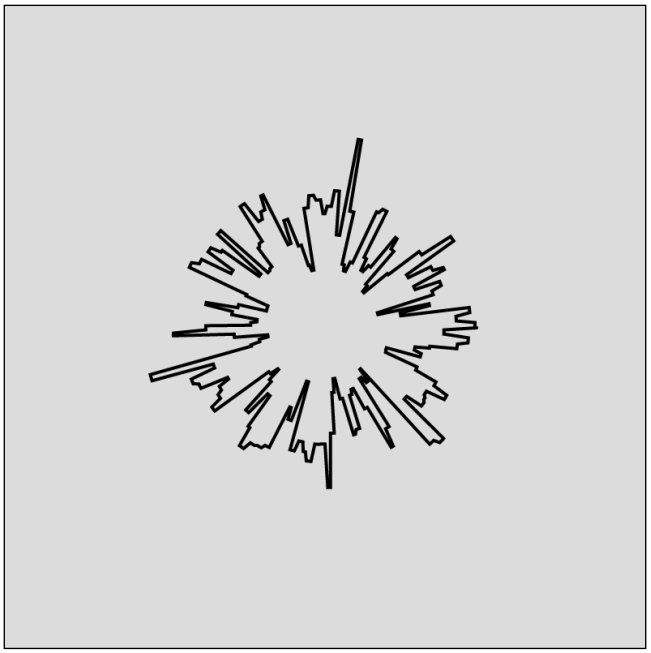
Sketch B4.3 radial graph

Making the data visualisation in the forms of a circle

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

function preload()
{
  song = loadSound('sounds/music.mp3')
}

function setup()
{
  createCanvas(400, 400)
  song.play()
  button = createButton('toggle')
  button.mousePressed(toggle)
  amp = new p5.Amplitude()
  angleMode(DEGREES)
}

function toggle()
{
  if (song.isPlaying())
  {
```

```
    song.pause()
  }
  else
  {
    song.play()
    amp = new p5.Amplitude()
  }
}

function draw()
{
  background(255)
  strokeWeight(2)
  noFill()
  let 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 * cos(i)
    y = r * sin(i)
    vertex(x, y)
  }
  endShape()

  if (volHistory.length > 360)
  {
    volHistory.splice(0, 1)
  }
}
```

A nice effect, this rotates and changes to the music



Challenges

1. Change the values of the mapping
2. Try other effects to do with the strokeWeight or the colours