

PAT 204/504 (Fall 2026)

Creative Coding

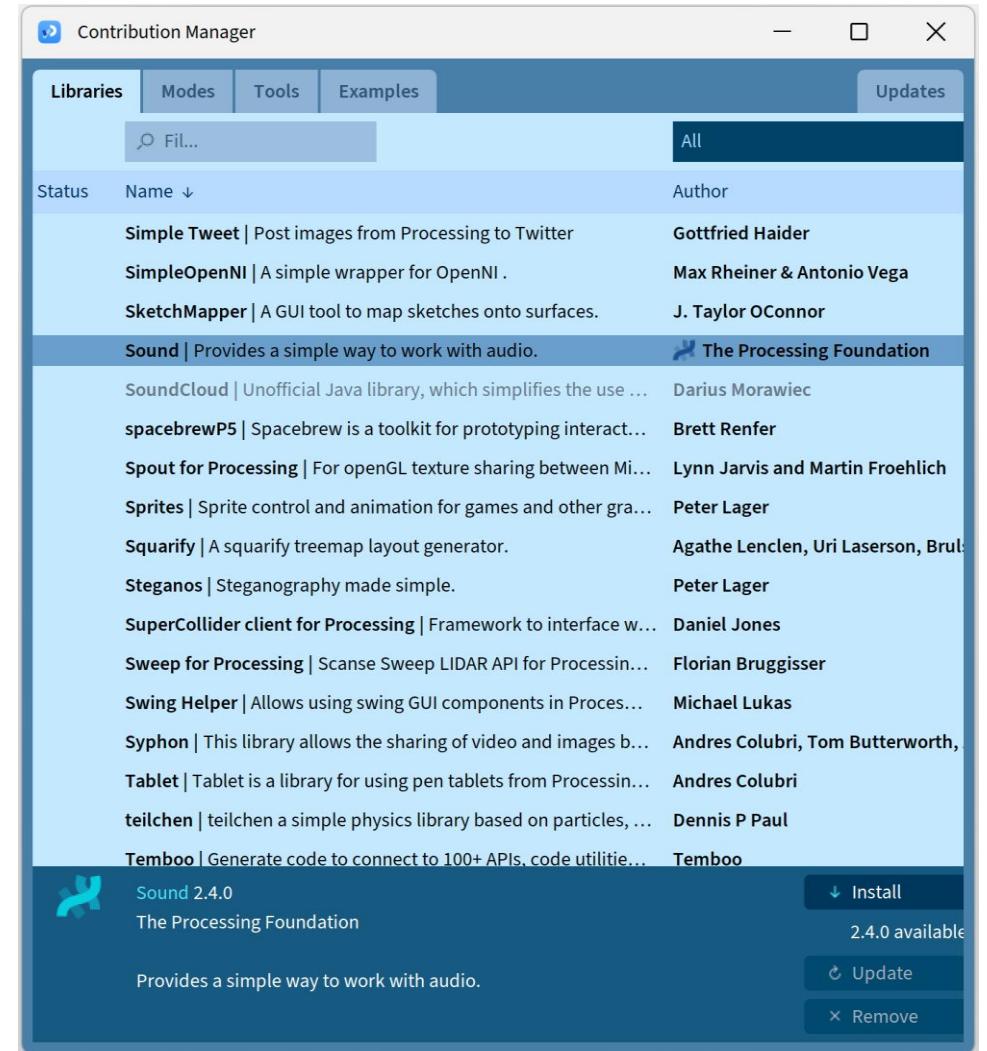
Lecture 9: Sounds

Instructor: Hao-Wen (Herman) Dong

Sounds

Library Manager

- Official Libraries maintained by the **Processing Foundation**
 - **Sound**
 - Video
 - Hardware I/O
 - JavaFX
- Many other third-part libraries
 - Networking
 - GUI
 - Animation
 - DeepVision



Sound Library

- A *library* is a collection of classes and functions
 - **I/O** SoundFile, AudioSample, AudioIn, MultiChannel
 - **Analysis** Amplitude, BeatDetector, PitchDetector, Waveform, FFT
 - **Effects** Delay, Reverb, AllPass, BandPass, HighPass, LowPass
 - **Noises** WhiteNoise, PinkNoise, BrownNoise
 - **Oscillators** Pulse, SinOsc, SawOsc, SqrOsc, TriOsc
 - **Misc** Env, Sound
- ... more at processing.org/reference/libraries/sound

Amplitude Class

```
import processing.sound.*;
```

Initialize an Amplitude object

```
Amplitude amp = new Amplitude(this);
```

```
AudioIn in = new AudioIn(this, 0);
```

```
float a;
```

Initialize an AudioIn object

```
void setup() {  
  size(400, 400);
```

```
  in.start();
```

Start taking audio input

```
  amp.input(in);
```

Route the audio input to the amplitude meter

```
}
```

```
void draw() {  
  background(0);
```

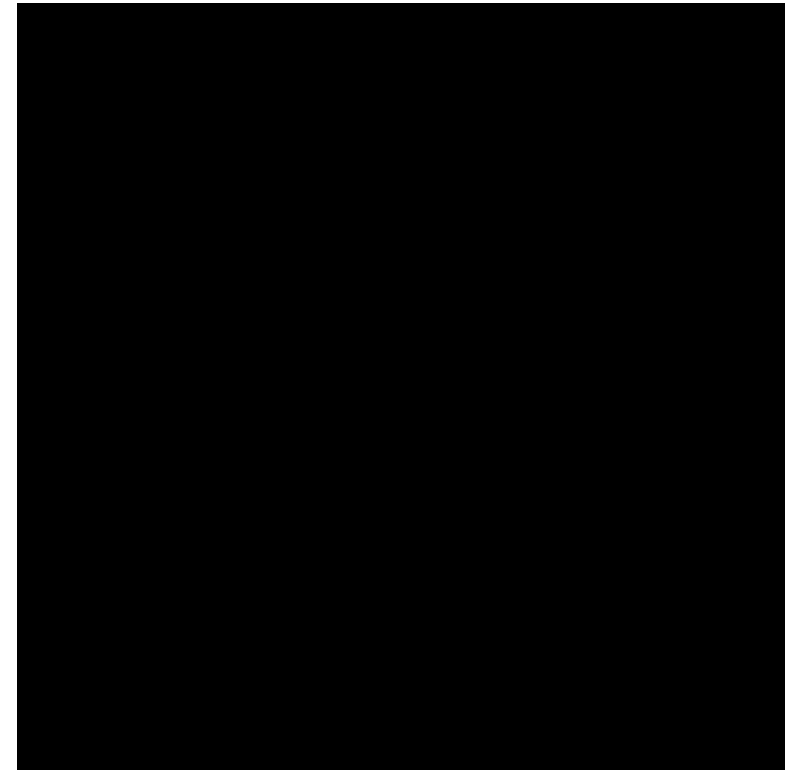
```
  a = amp.analyze();
```

Measure the amplitude

```
  circle(200, 200, a * 400);
```

```
}
```

Normalized to [0, 1]



PitchDetector Class

```
import processing.sound.*;
```

Initialize a PitchDetector object

```
PitchDetector pd = new PitchDetector(this);
```

```
AudioIn in = new AudioIn(this, 0);
```

```
float pitch;
```

Initialize an AudioIn object

```
void setup() {  
  size(400, 400);
```

```
  in.start();
```

Start taking audio input

```
  pd.input(in);
```

Route the audio input to the pitch detector

```
  stroke(#ff0000);  
  strokeWeight(5);
```

```
}
```

```
void draw() {  
  background(0);
```

Set the sensitivity

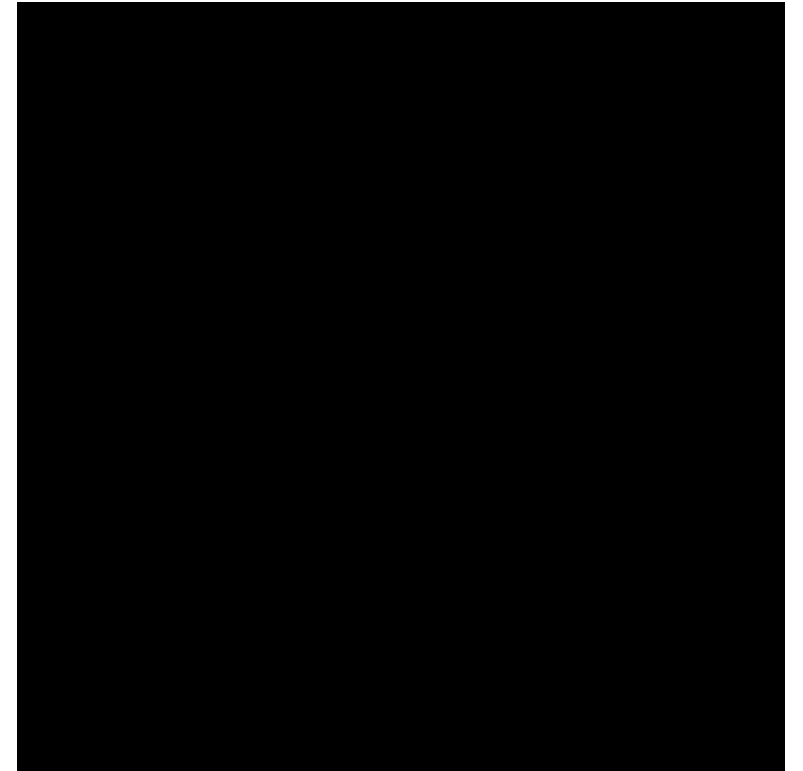
```
  pitch = pd.analyze(0.5);
```

Detect the pitch

```
  if (pitch > 0) {  
    line(0, height - pitch, 400, height - pitch);
```

```
  }
```

```
}
```



FFT Class

```
import processing.sound.*;
```

→ Import the Sound library

```
int bands = 512;
```

```
FFT fft = new FFT(this, bands);
```

→ Initialize an FFT object

```
AudioIn in = new AudioIn(this, 0);
```

→ Initialize an AudioIn object

```
float[] spectrum = new float[bands];
```

```
void setup() {  
  size(512, 360);
```

→ Initialize an array to store the spectrum

```
  in.start();
```

→ Start taking audio input

```
  fft.input(in);
```

→ Route the audio input to the FFT analyzer

```
}
```

```
void draw() {  
  background(255);
```

Specify the array to
store the outputs

```
  fft.analyze(spectrum);
```

→ Run Fast Fourier Transform

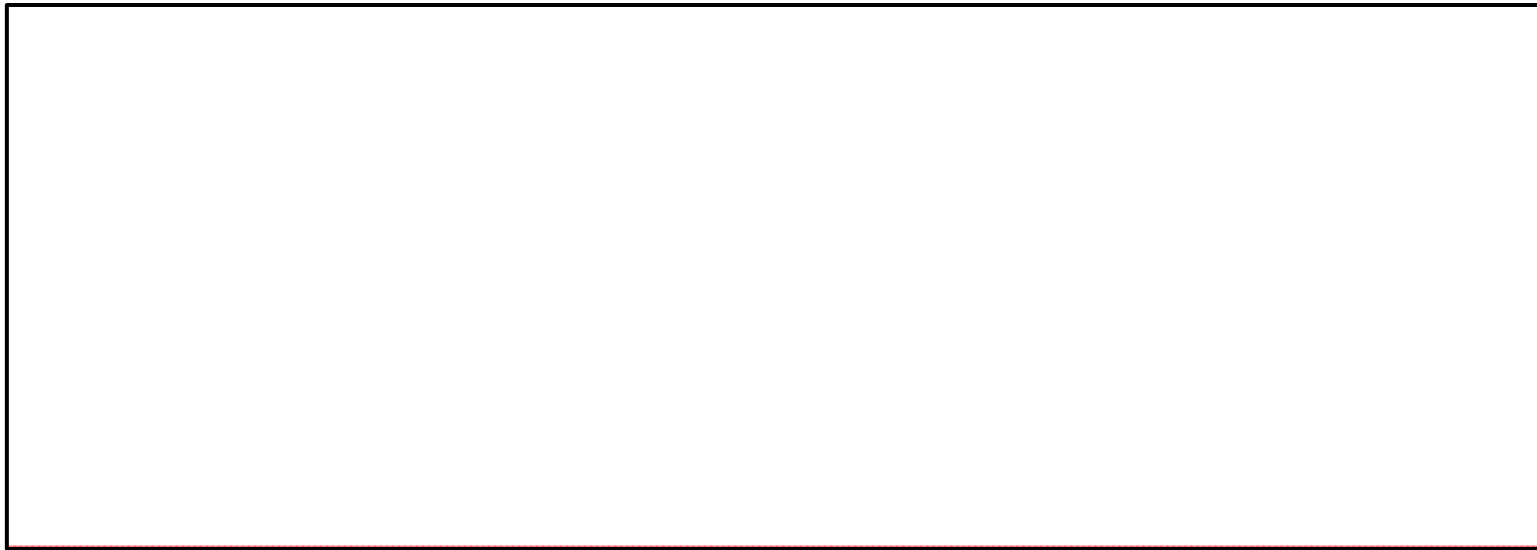
```
  for(int i = 0; i < bands; i++){  
    line(i, height, i, height - spectrum[i] * height * 5);
```

Normalized to [0, 1]

```
  }  
}
```

🔥 HW 4: Spectrum Visualizer

- Instructions will be sent by **emails** and released on the **course website**
- Take **microphone inputs** and apply fast Fourier transform (**FFT**)
- Show the **spectrum** and a **trace**



Midterm Assignment

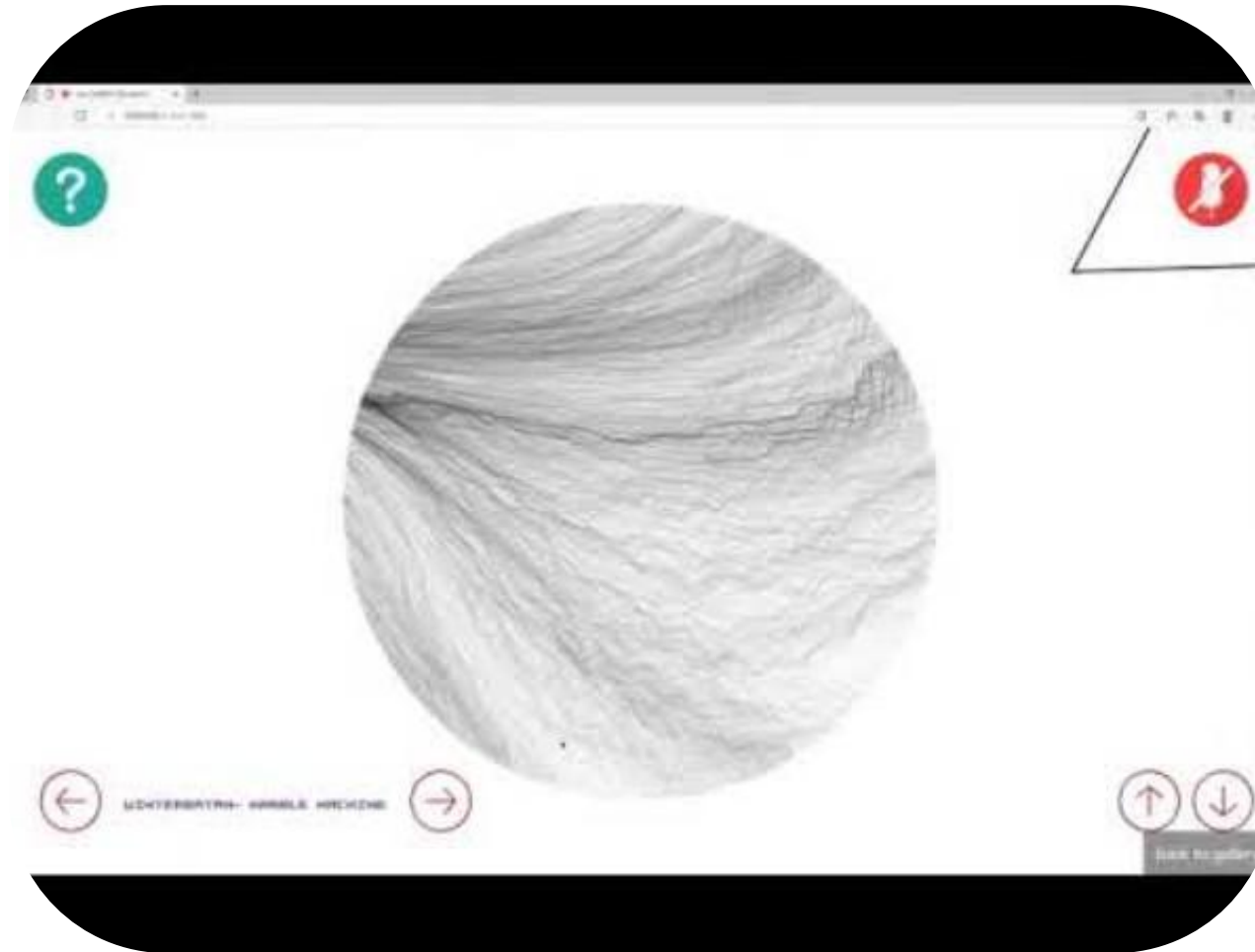
🔥 Midterm Assignment: My Music Visualizer

- Instructions will be sent by **emails** and released on the **course website**
- **Open-ended** assignment
- Use what you've learned from the class (and beyond!)

🔥 Midterm Assignment: Rubrics

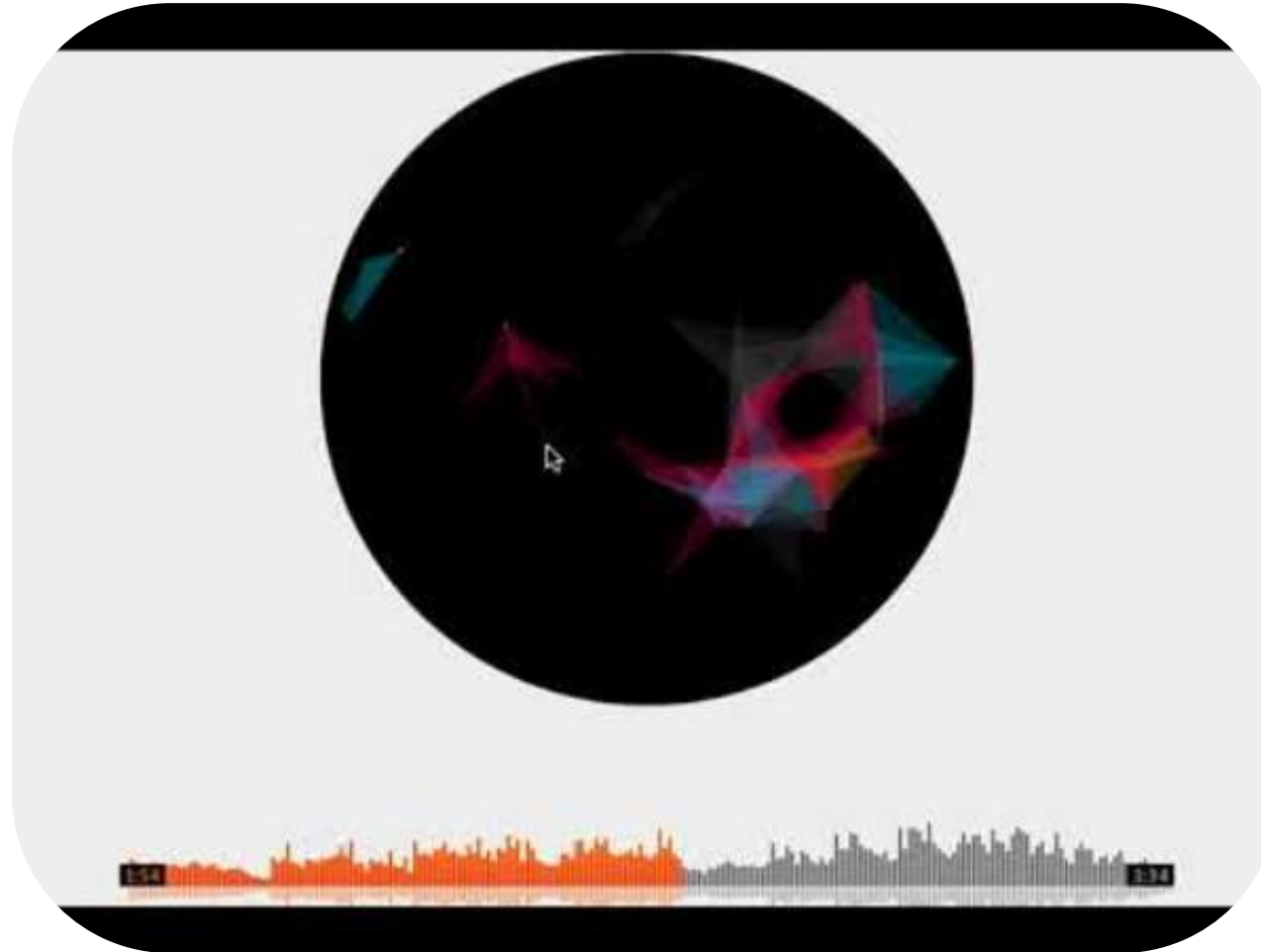
- Use **two of the following concepts (10pt)**
 - **Loops or recursion**
 - **Data structures** (e.g., arrays, lists, dictionaries, etc.)
 - **Classes and objects**
- **Clear documentation** in code (5pt)
- **Live demo** in class (5pt)

Music Visualizer: Example



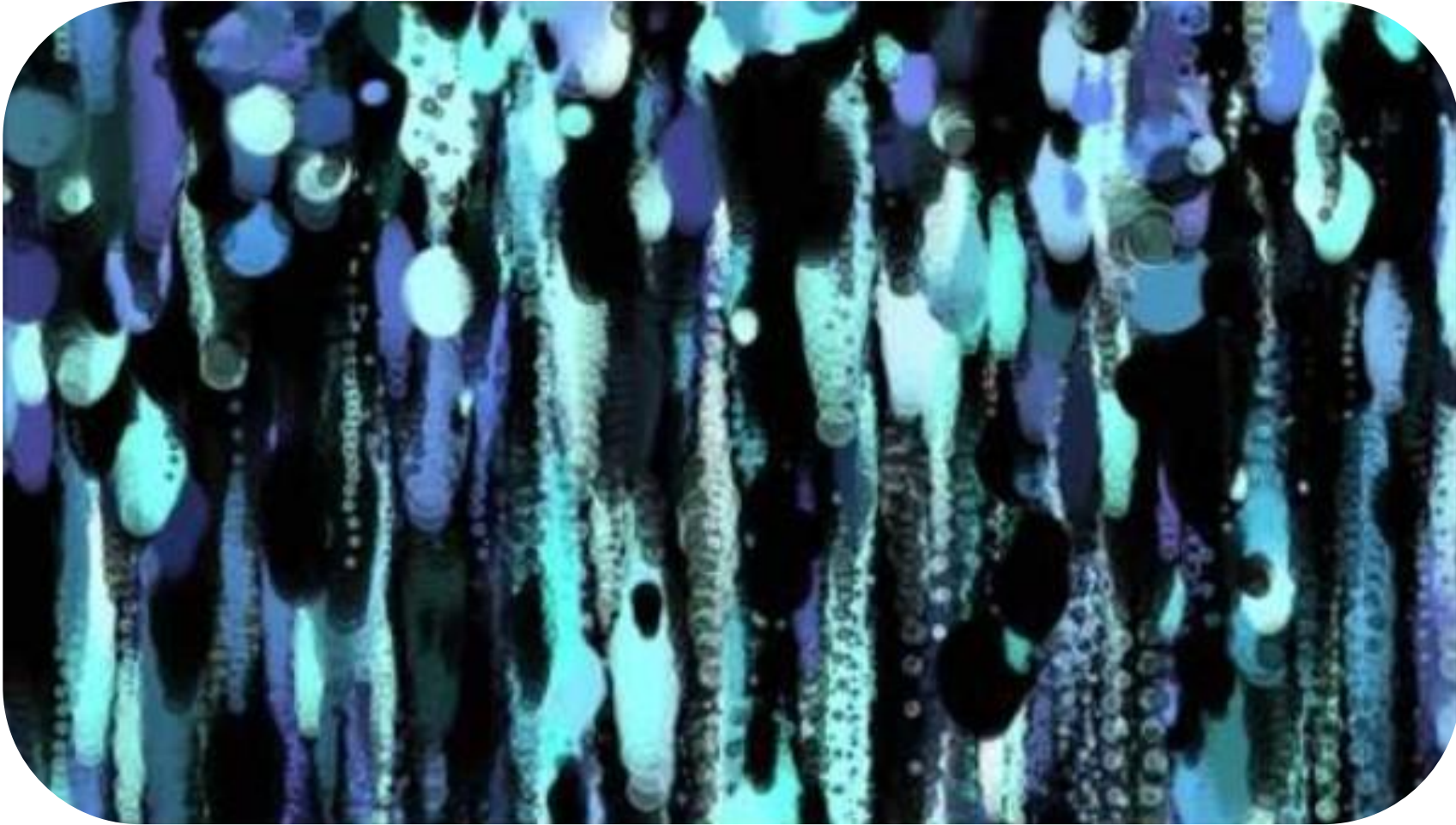
youtu.be/MoM6AxN7TK0

Music Visualizer: Example



youtu.be/00kQX4m28IU

Music Visualizer: Example



youtu.be/LMBx6AYaRXc

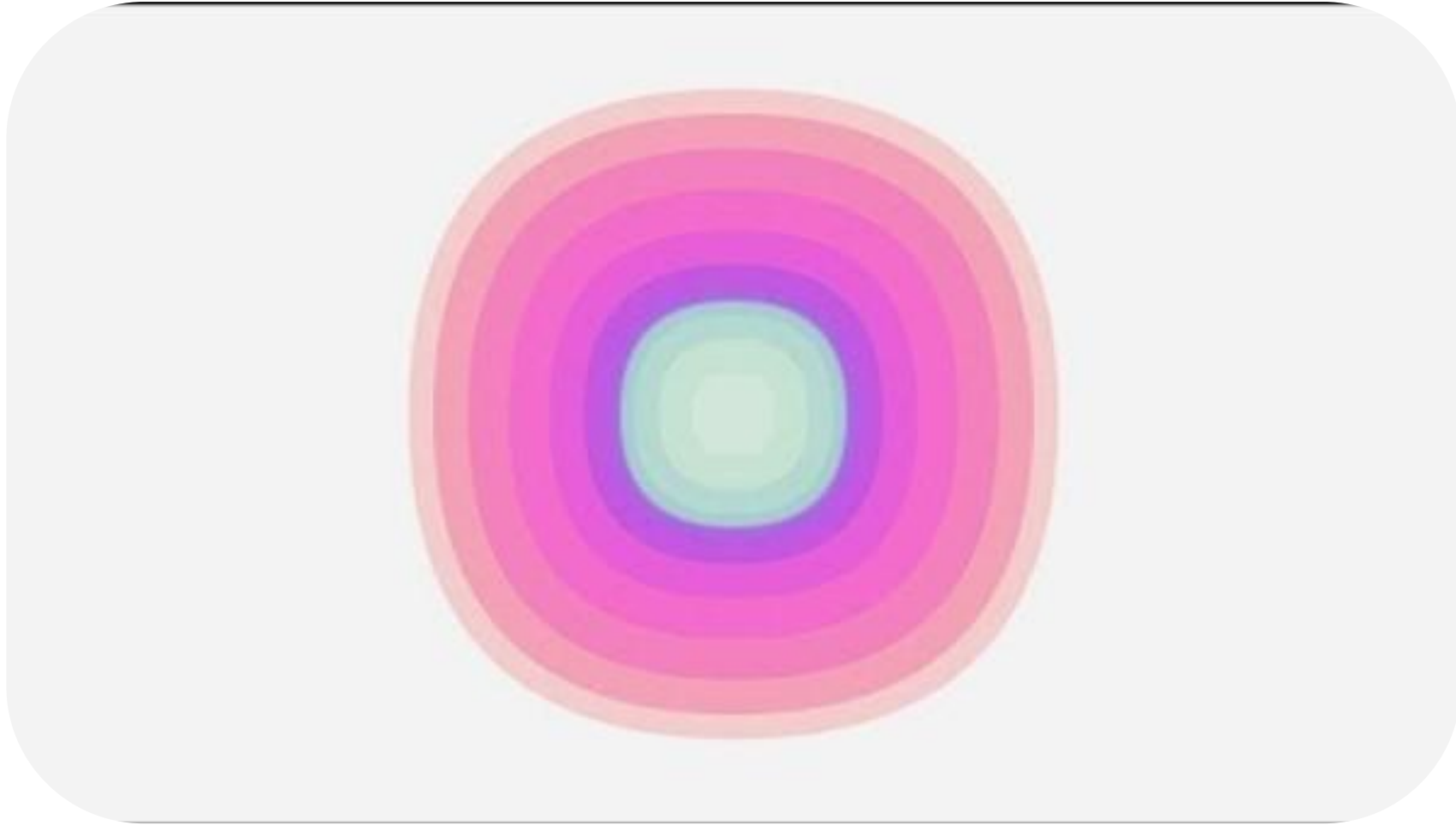
Music Visualizer: Example



youtu.be/UvoABmr4Fdo

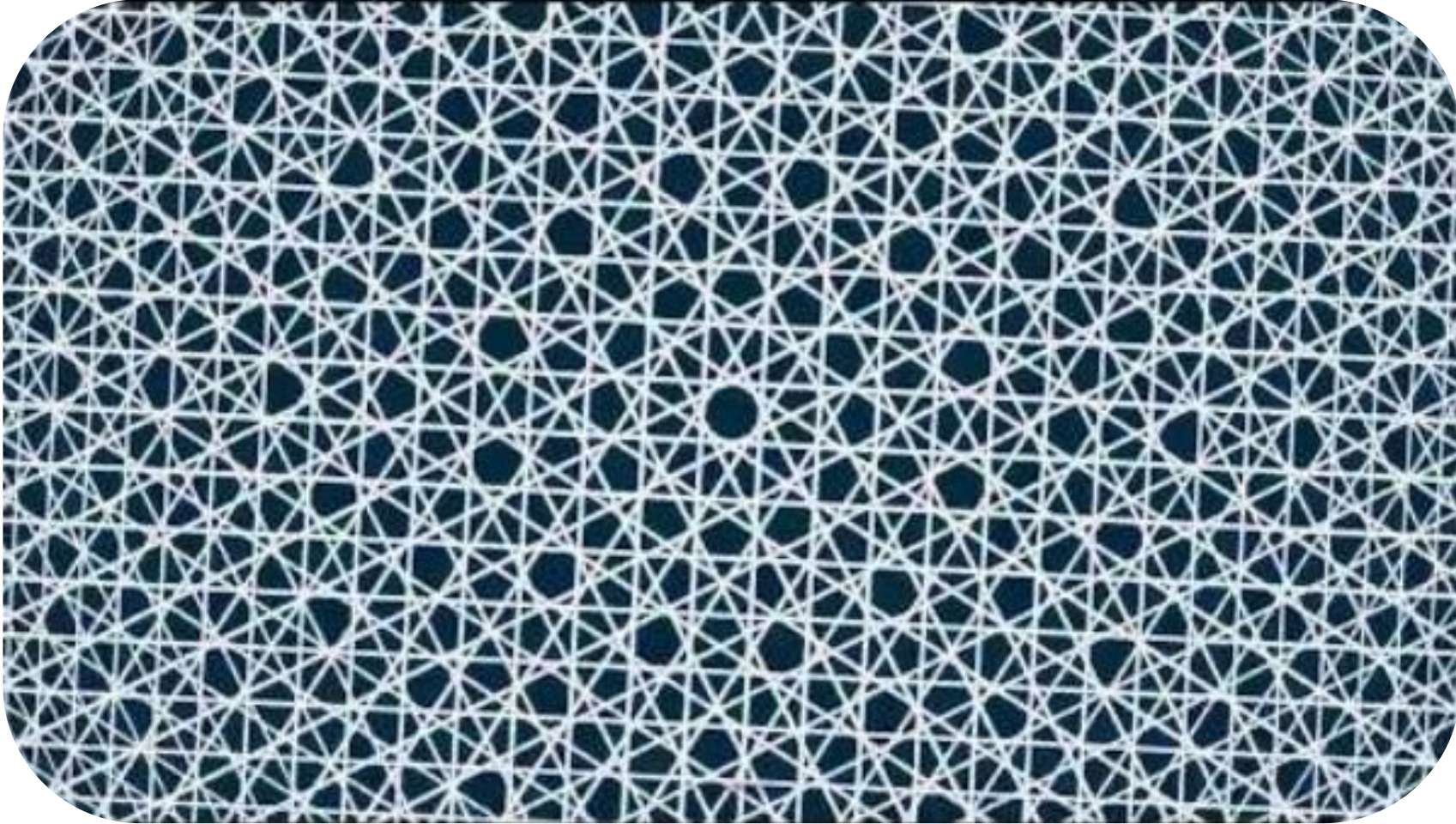
Generative Arts

| Generative Arts: Example



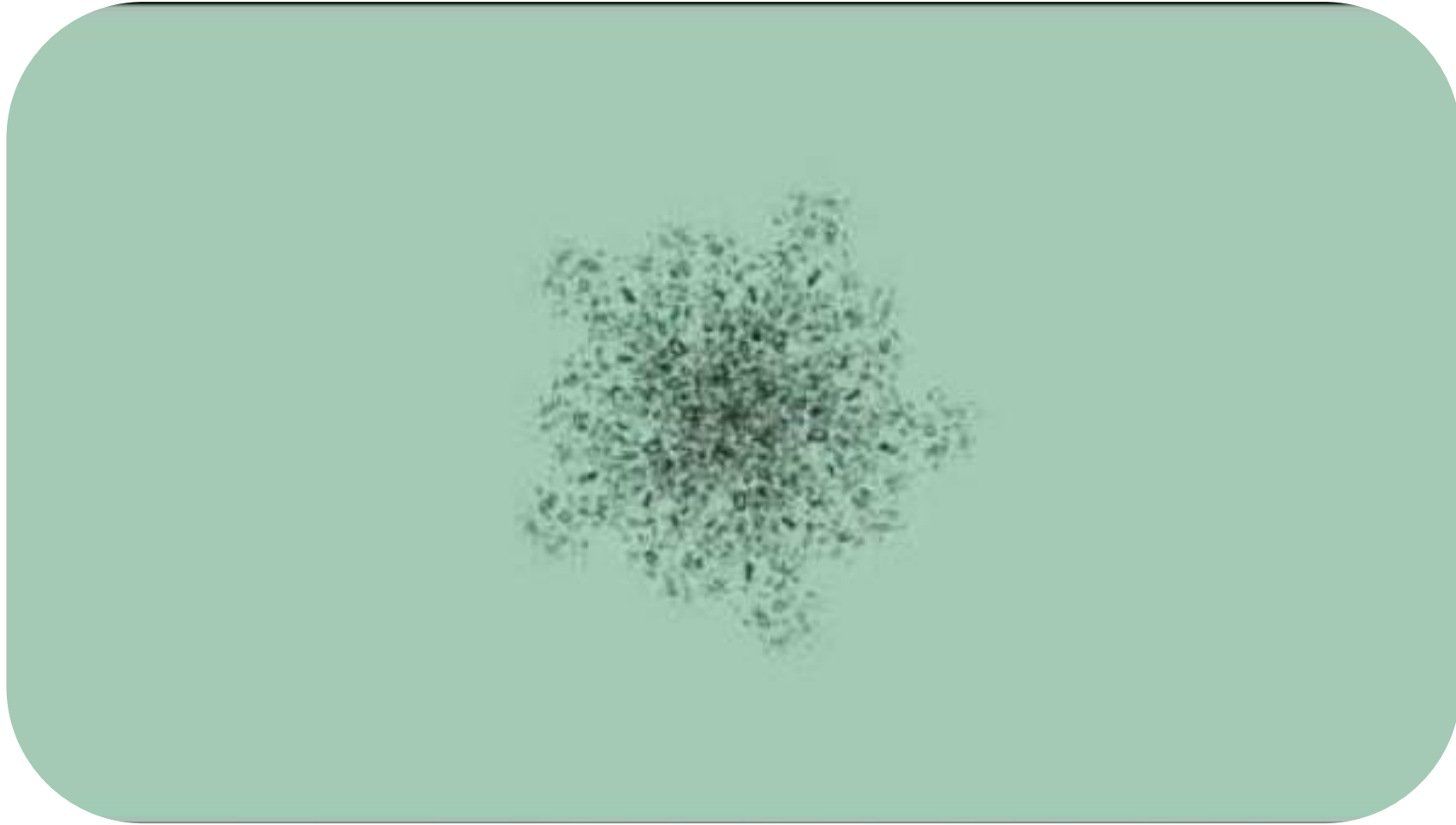
youtu.be/5-ttqEsf518

Generative Arts: Example



youtu.be/1ZXOnR9qNDI

| Generative Arts: Example



youtu.be/Keol4CnBWUU

Generative Arts: Example



youtu.be/zP_paWcXkaE

Generative Arts: Example



youtu.be/zT7We17o4zU

Generative Arts: Example



youtu.be/ZzNO3FvkTJM5

| Generative Arts: Example

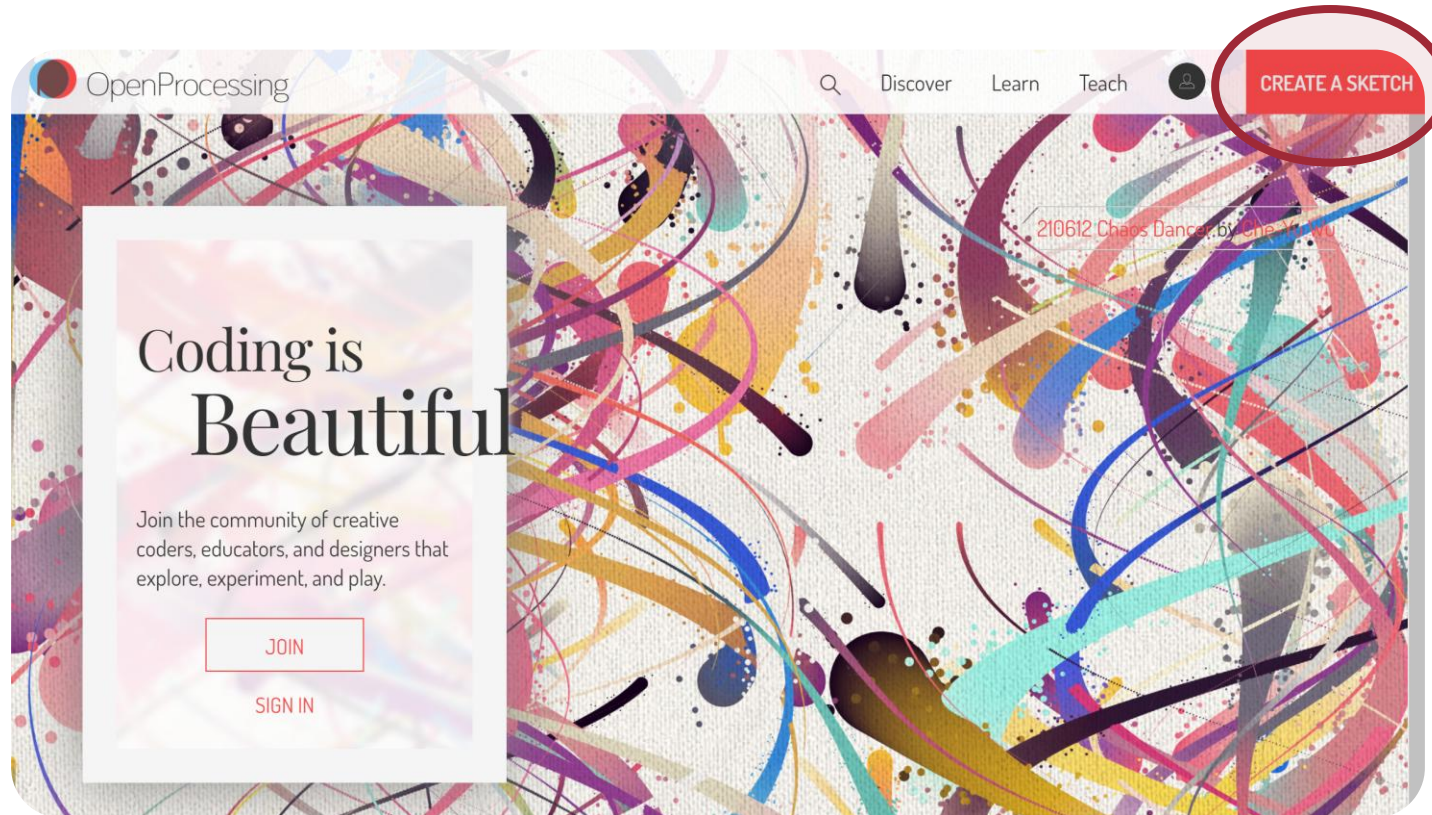


youtu.be/ITB2uq4MeBQ

OpenProcessing

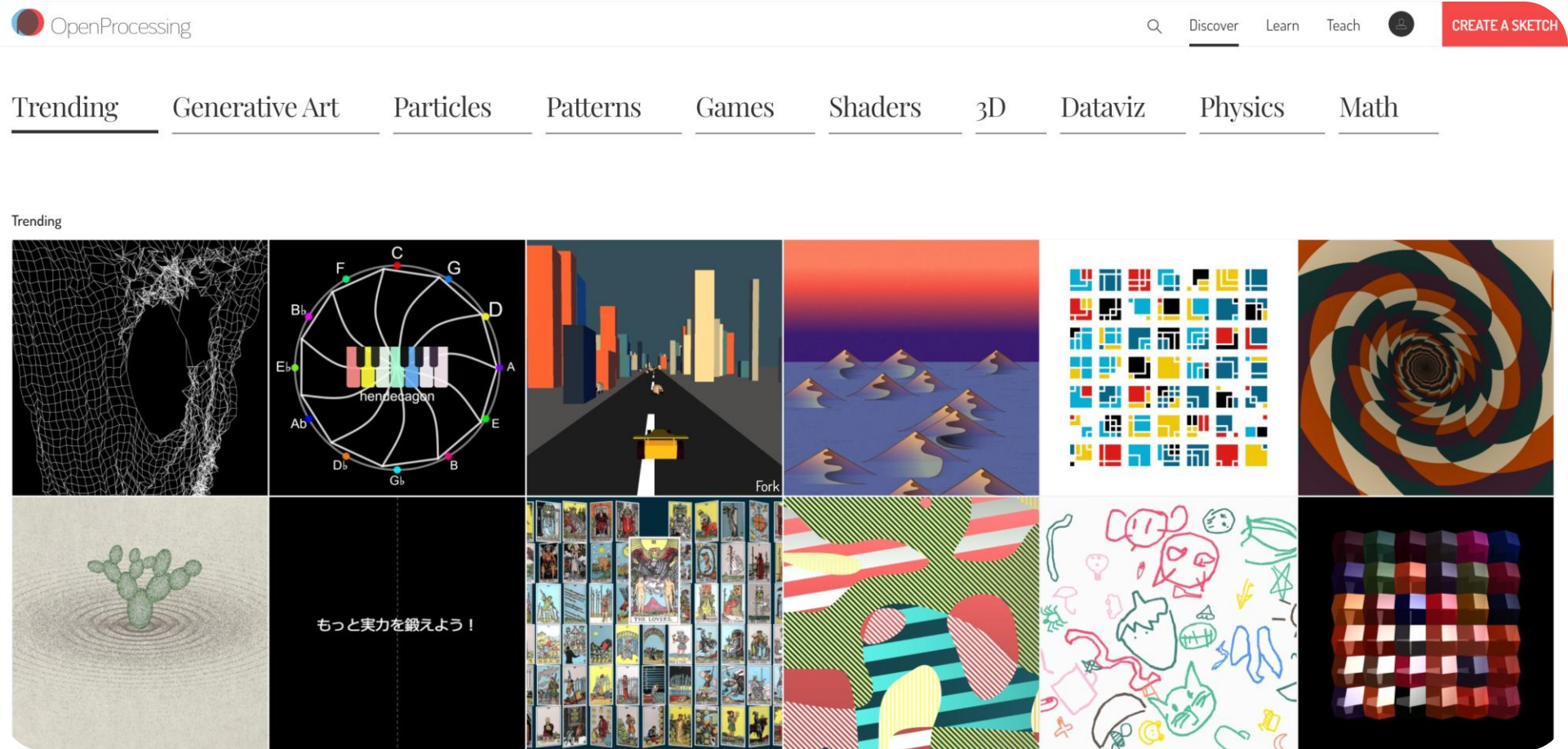
OpenProcessing

- Large community of creative coders, educators and designers!



openprocessing.org

OpenProcessing Gallery



openprocessing.org/discover

Processing on Different Platforms

- | | | |
|----------------|---------------|---|
| • JavaScript | p5.js | p5js.org |
| • Python | processing.py | py.processing.org |
| • Android | | android.processing.org |
| • Raspberry Pi | | pi.processing.org |

| p5.js: Processing for JavaScript

- Ideal for **web programming**
- Can be **embedded on websites**
- Slightly different syntax but same design philosophy
- Online editor: editor.p5js.org

Recap

Sound Library

- A *library* is a collection of classes and functions
 - **I/O** SoundFile, AudioSample, AudioIn, MultiChannel
 - **Analysis** Amplitude, BeatDetector, PitchDetector, Waveform, FFT
 - **Effects** Delay, Reverb, AllPass, BandPass, HighPass, LowPass
 - **Noises** WhiteNoise, PinkNoise, BrownNoise
 - **Oscillators** Pulse, SinOsc, SawOsc, SqrOsc, TriOsc
 - **Misc** Env, Sound
- ... more at processing.org/reference/libraries/sound

Amplitude Class

```
import processing.sound.*;
```

Initialize an Amplitude object

```
Amplitude amp = new Amplitude(this);
```

```
AudioIn in = new AudioIn(this, 0);
```

```
float a;
```

Initialize an AudioIn object

```
void setup() {  
  size(400, 400);
```

```
  in.start();
```

Start taking audio input

```
  amp.input(in);
```

Route the audio input to the amplitude meter

```
}
```

```
void draw() {  
  background(0);
```

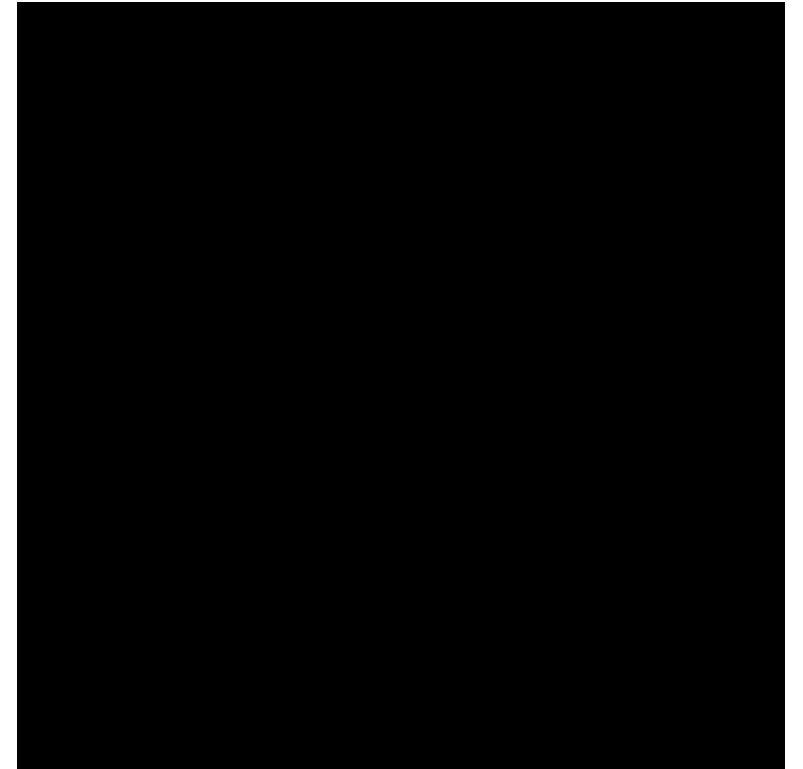
```
  a = amp.analyze();
```

Measure the amplitude

```
  circle(200, 200, a * 400);
```

```
}
```

Normalized to [0, 1]



PitchDetector Class

```
import processing.sound.*;
```

Initialize a PitchDetector object

```
PitchDetector pd = new PitchDetector(this);
```

```
AudioIn in = new AudioIn(this, 0);
```

```
float pitch;
```

Initialize an AudioIn object

```
void setup() {  
  size(400, 400);
```

```
  in.start();
```

Start taking audio input

```
  pd.input(in);
```

Route the audio input to the pitch detector

```
  stroke(#ff0000);  
  strokeWeight(5);
```

```
}
```

```
void draw() {  
  background(0);
```

Set the sensitivity

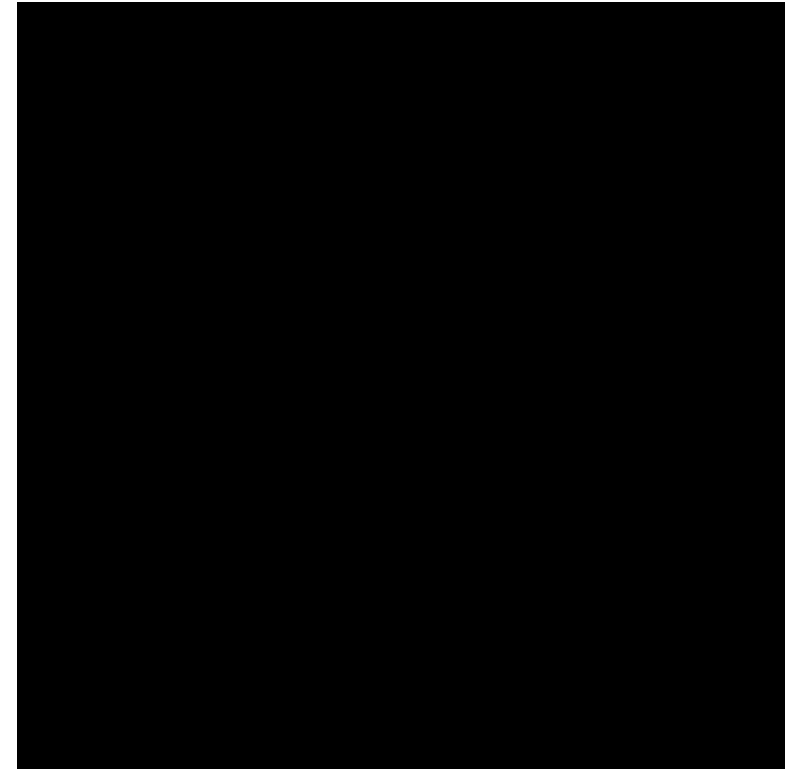
```
  pitch = pd.analyze(0.5);
```

Detect the pitch

```
  if (pitch > 0) {  
    line(0, height - pitch, 400, height - pitch);
```

```
  }
```

```
}
```



FFT Class

```
import processing.sound.*;
```

→ Import the Sound library

```
int bands = 512;
```

```
FFT fft = new FFT(this, bands);
```

→ Initialize an FFT object

```
AudioIn in = new AudioIn(this, 0);
```

→ Initialize an AudioIn object

```
float[] spectrum = new float[bands];
```

```
void setup() {  
  size(512, 360);
```

→ Initialize an array to store the spectrum

```
  in.start();
```

→ Start taking audio input

```
  fft.input(in);
```

→ Route the audio input to the FFT analyzer

```
}
```

```
void draw() {  
  background(255);
```

Specify the array to
store the outputs

```
  fft.analyze(spectrum);
```

→ Run Fast Fourier Transform

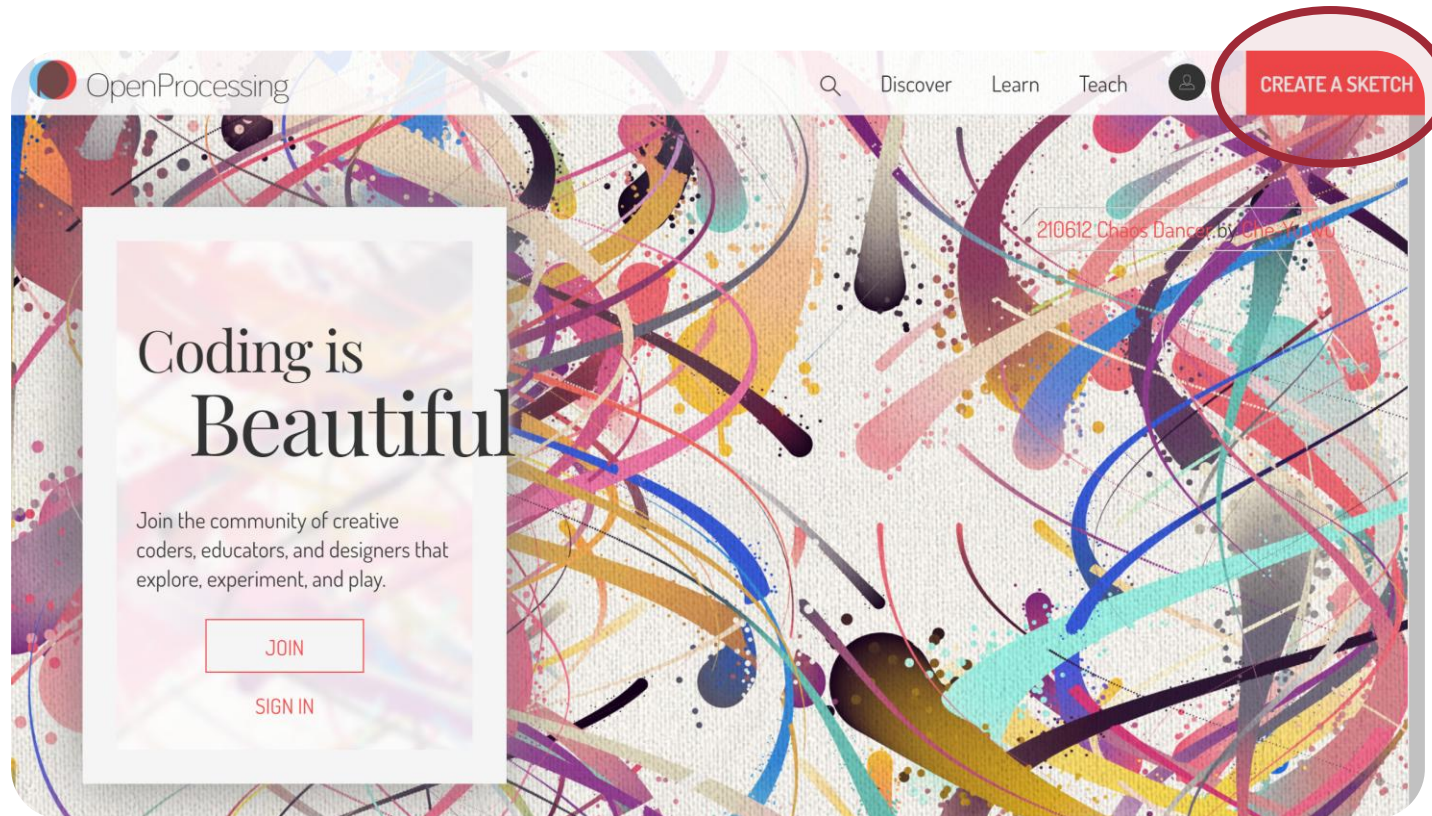
```
  for(int i = 0; i < bands; i++){  
    line(i, height, i, height - spectrum[i] * height * 5);
```

Normalized to [0, 1]

```
  }  
}
```

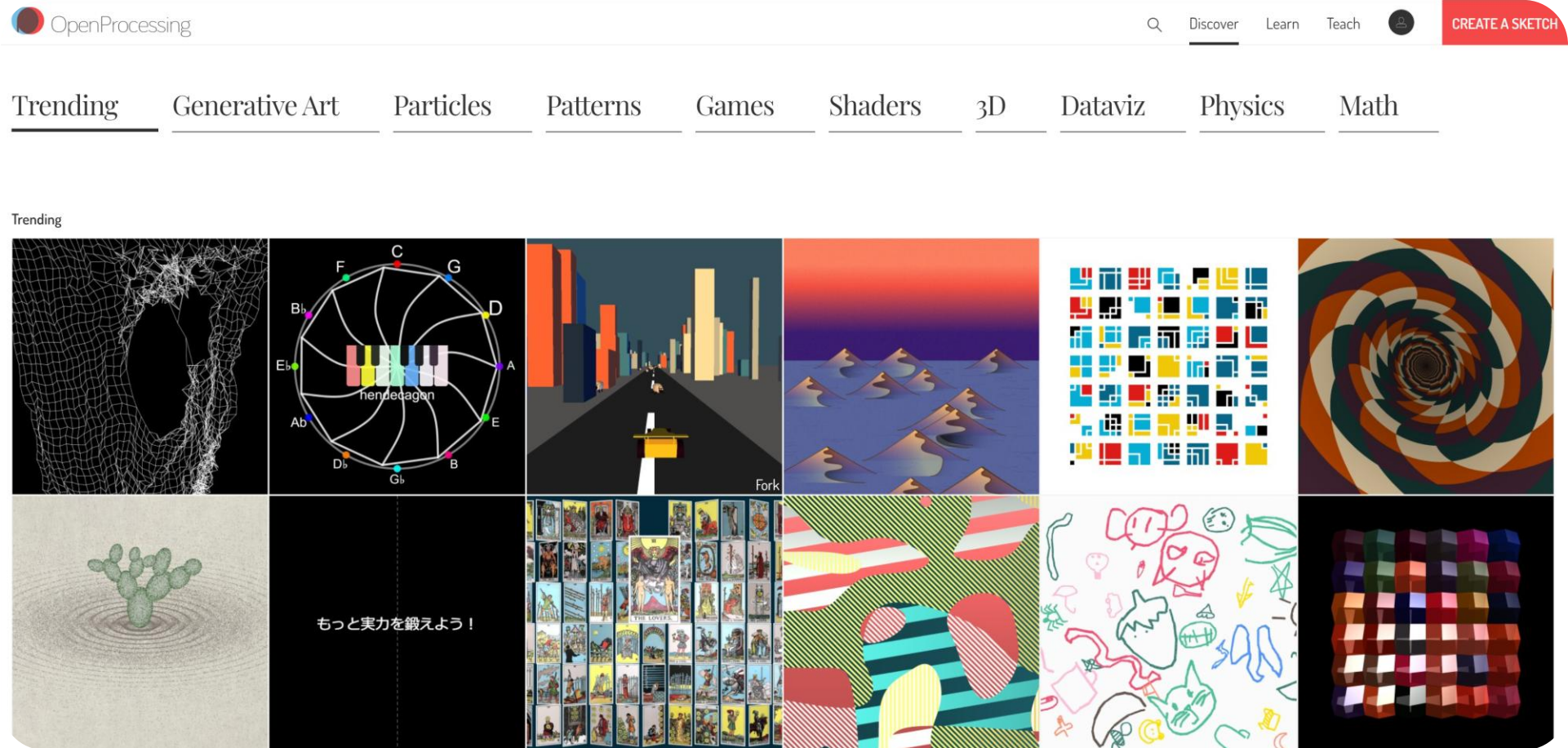
OpenProcessing

- Large community of creative coders, educators and designers!



openprocessing.org

OpenProcessing Gallery



openprocessing.org/discover

Processing on Different Platforms

- | | | |
|----------------|---------------|---|
| • JavaScript | p5.js | p5js.org |
| • Python | processing.py | py.processing.org |
| • Android | | android.processing.org |
| • Raspberry Pi | | pi.processing.org |

Next Lecture

Transformations & 3D Graphics

