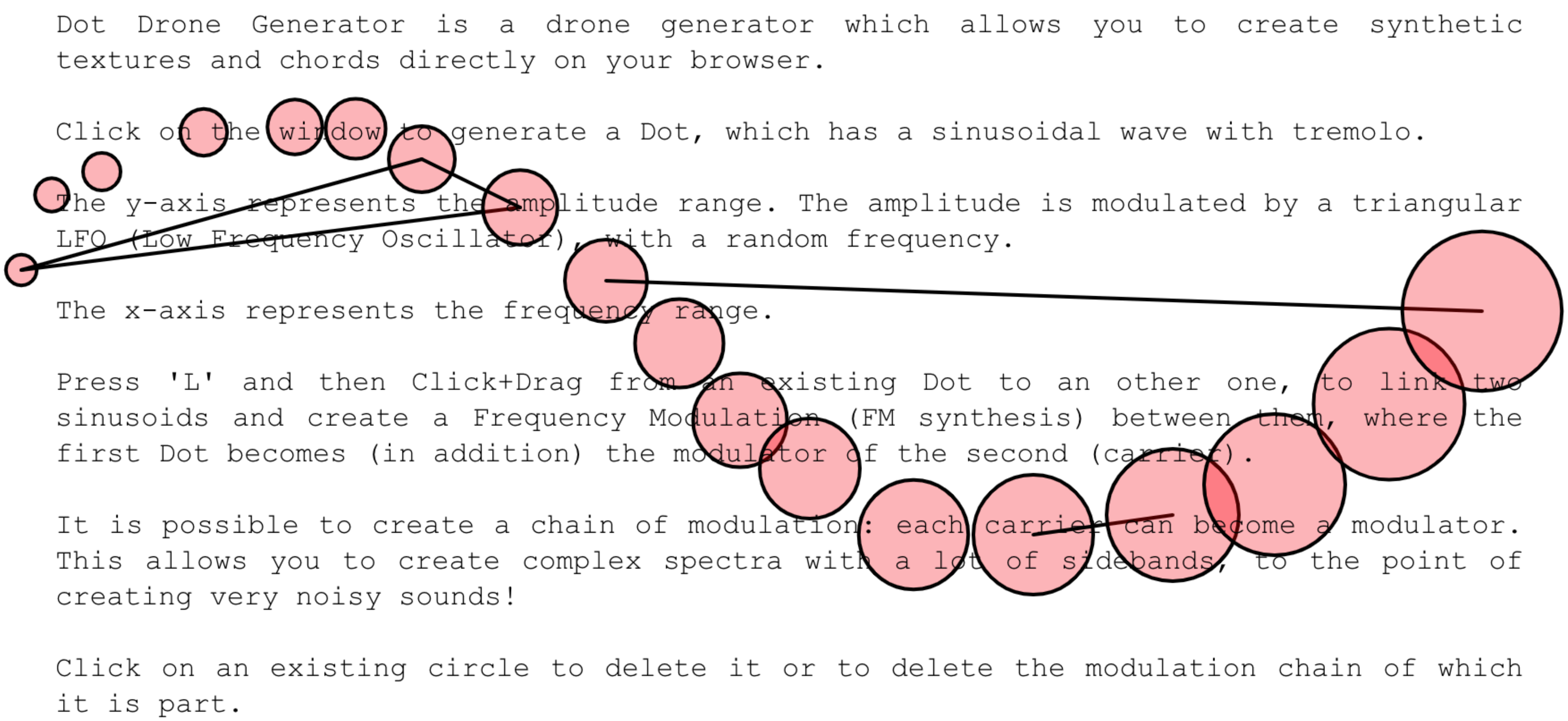
an introduction to the WEB AUDIO API

Day 1

highSCORE



Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.

Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

Alberto Barberis
2021

highSCORE

HELLO :)

Alberto Barberis -> www.albertobarberis.it

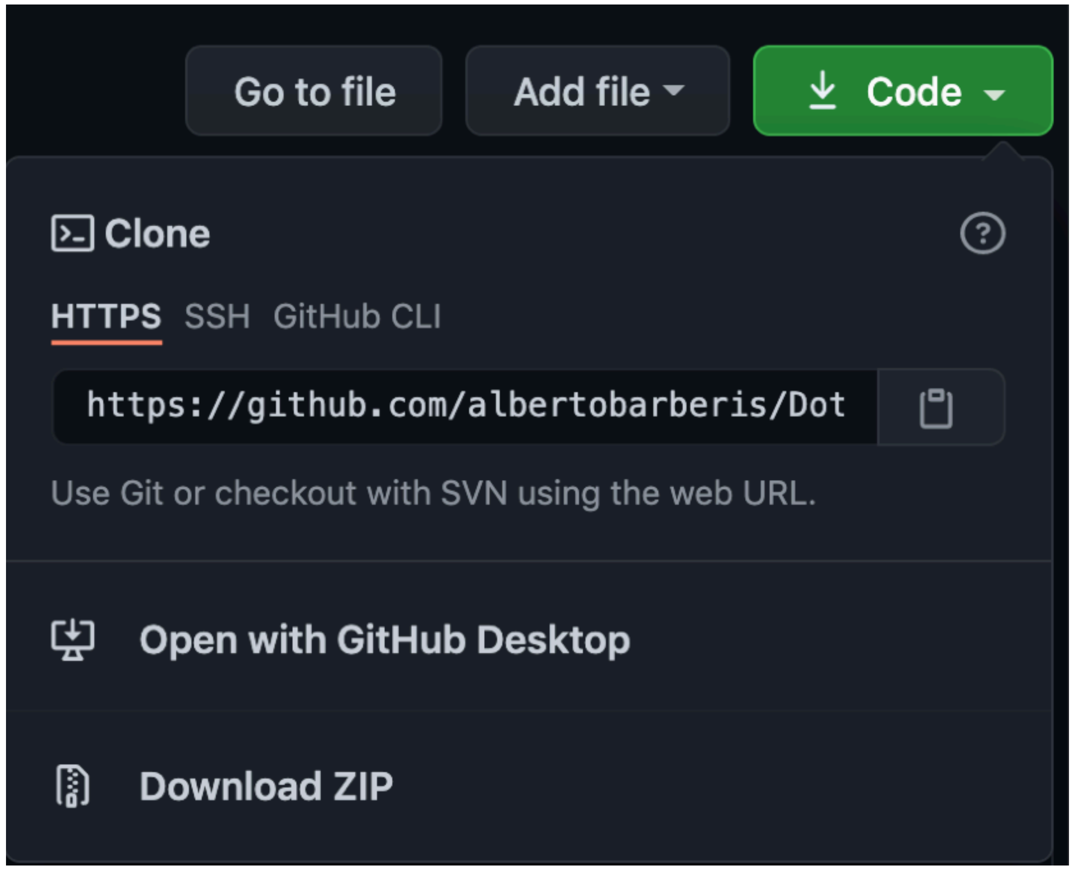
Lecturer in Electronic Music at the Conservatorio della Svizzera Italiana of Lugano (CH)

composer (instrumental and electroacoustic)
engineer (information technology and new media)
guitarist (my origin…)
electronic music performer
sound designer
code artist and developer

MATERIALS

You find all the **materials** (pdf slides, and code) for the workshop at the following link of a **Github repository**.

https://github.com/albertobarberis/DotDroneGenerator



```
https://github.com/albertobarberis/Dot
```

Use Git or checkout with SVN using the web URL.

Click on the green **Code** button and then on **Download ZIP** to download all the materials of the repository.

GOALS

1. understand what is the **web audio** and what are the basic concepts of the **web audio Application Programming Interface** (**API**);

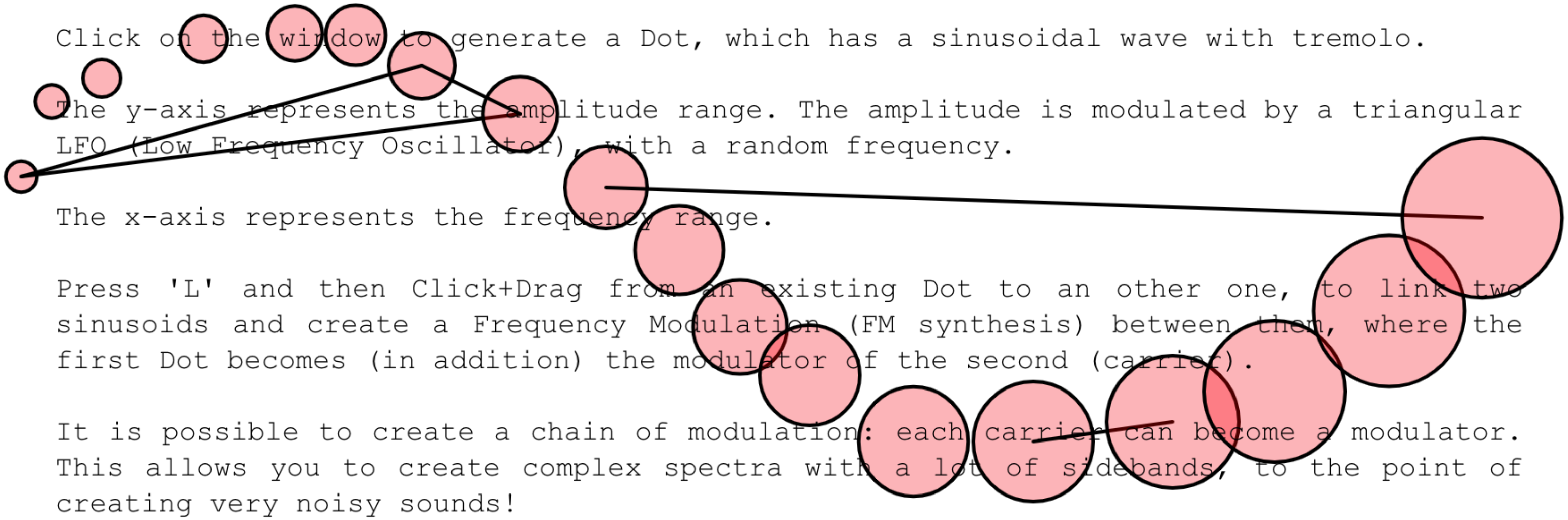
2. analyze together **some examples** of **web audio applications**;

3. summarize what are the **possibilities** offered by the web audio for sound design and music composition;

4. understand **what you need to know** to deal with the web audio (web programming: HTML, css, javascript; music theory and sound design);

5. **develop** a simple **web audio app**: the ***Dot Drone Generator***.

1. WHAT IS THE WEB AUDIO



Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.

Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

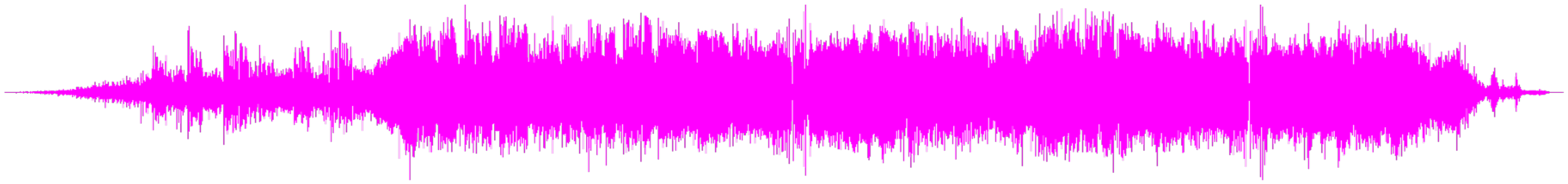
The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

1.1 WHAT IS THE WEB AUDIO



With the expression **web audio** we refer to the technologies that allow to **process and/or synthesize audio signals in web applications** (running on a browser).

The web audio allows you to **create and manipulate music in real-time in different devices** (smartphone, computer, tablet, smart TV, etc.) which use a browser that supports the **web audio**.

1.1 WHAT IS THE WEB AUDIO

The **web audio** is a rather new technology, made available thanks to the release of the **web audio API** by the **World Wide Web Consortium (W3C)** in 2011.

W3C is the main **international standards organization** for the World Wide Web.

Previously, audio could be embedded in browsers, but more complex audio effects were only available through the use of plugins.

Dealing with web application design means primarily dealing with **web programming languages**.

What is a programming language?

1.2 WHAT IS A PROGRAMMING LANGUAGE

A **programming language** is a formal language used to develop *software programs*, *scripts*, or other sets of instructions for computers to execute.

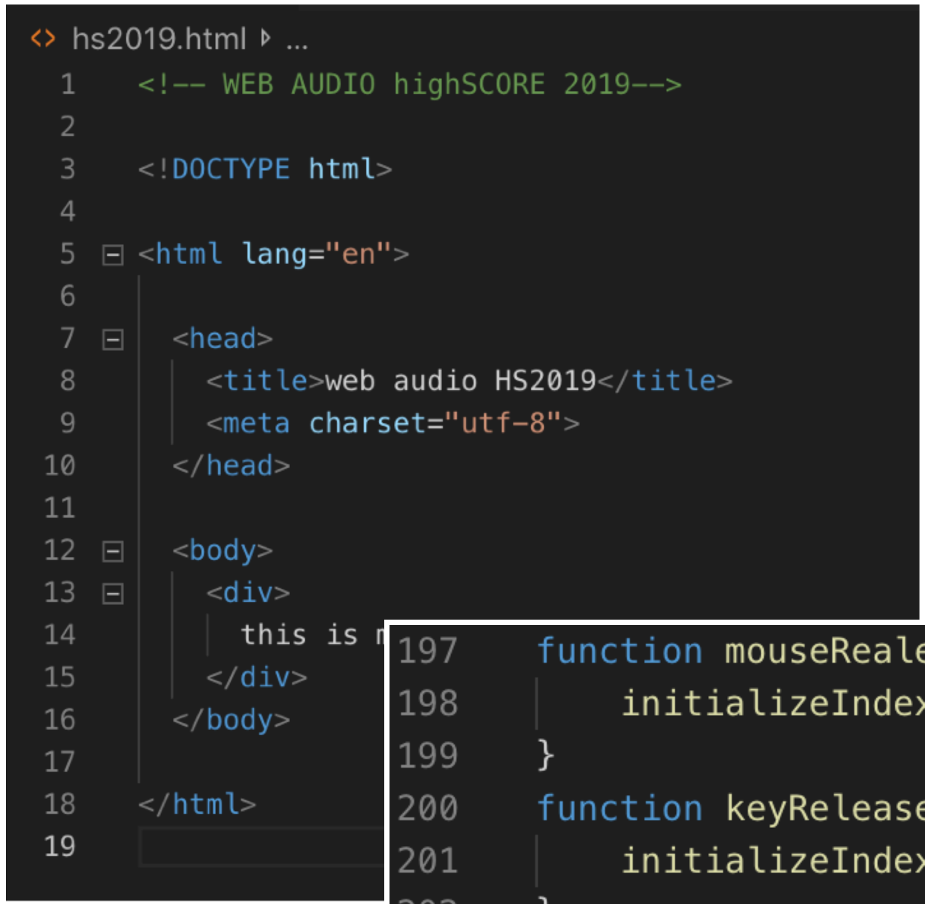
Thousands of different programming languages have been created, and more are being created every year.

Many languages share similarities, but each has its own:

- **syntax** (rules, structure, vocabulary, etc.);

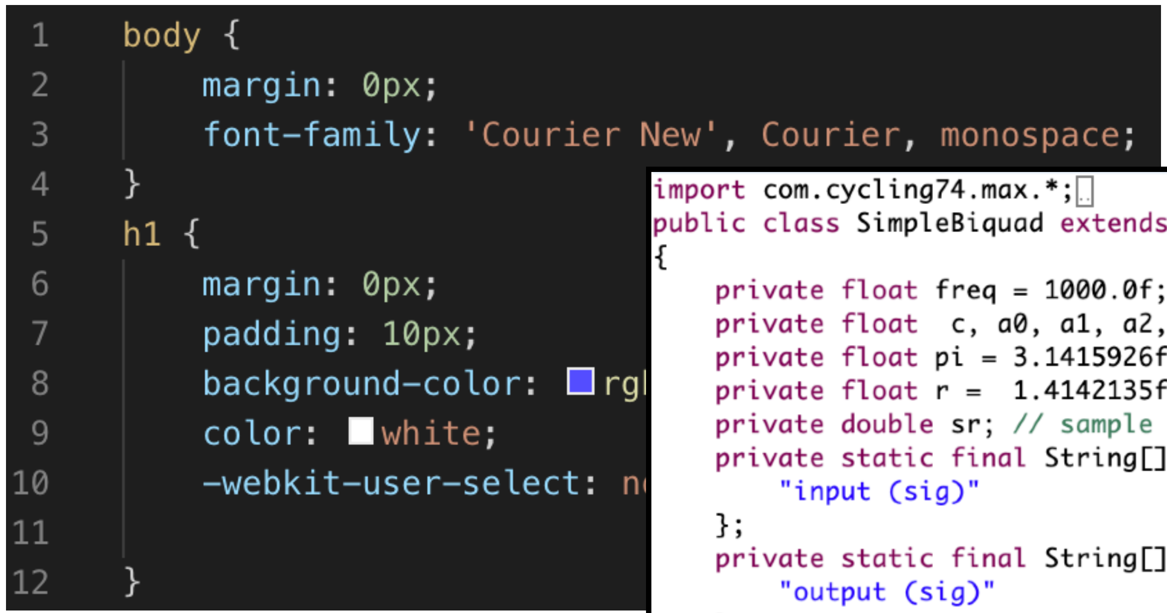
- **semantics** (meaning, logic, characteristics, etc.).

1.2 WHAT IS A PROGRAMMING LANGUAGE



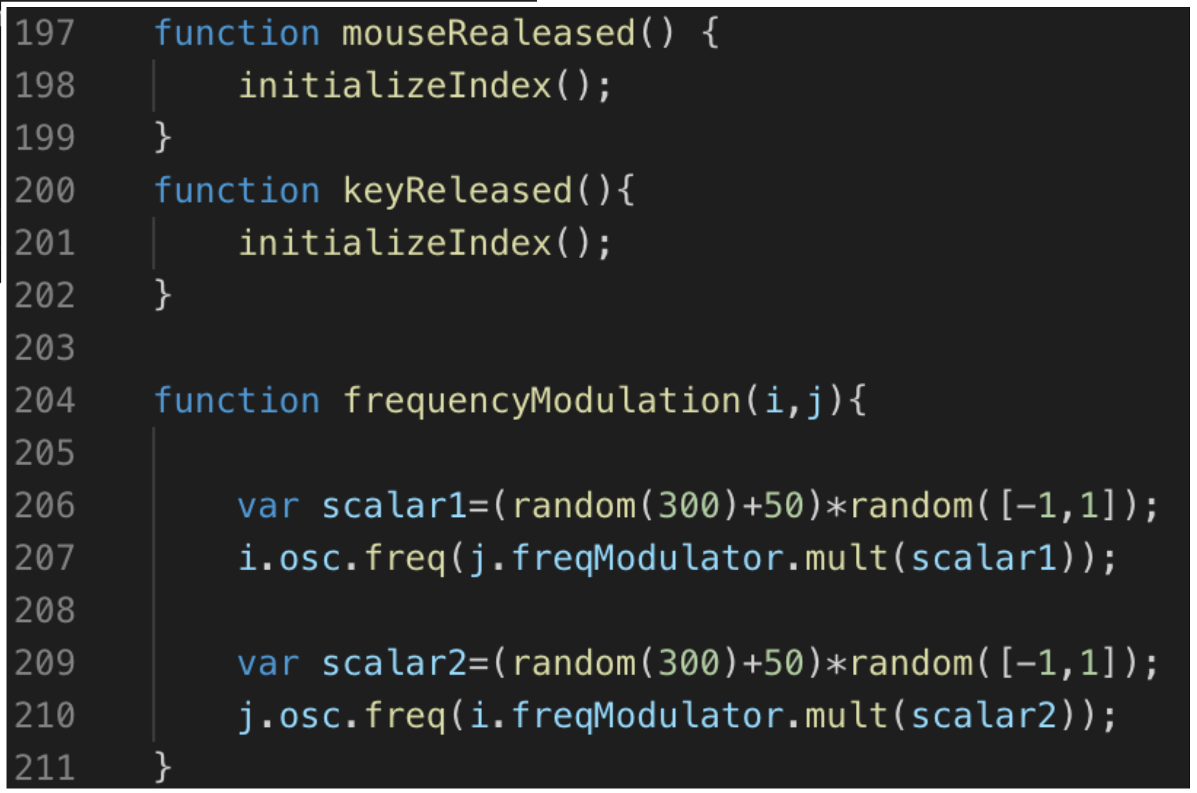
```
<> hs2019.html ▸ ...
<!-- WEB AUDIO highSCORE 2019-->

<!DOCTYPE html>

<html lang="en">

  <head>
    <title>web audio HS2019</title>
    <meta charset="utf-8">
  </head>

  <body>
    <div>
      this is r
    </div>
  </body>

</html>
```

HTML



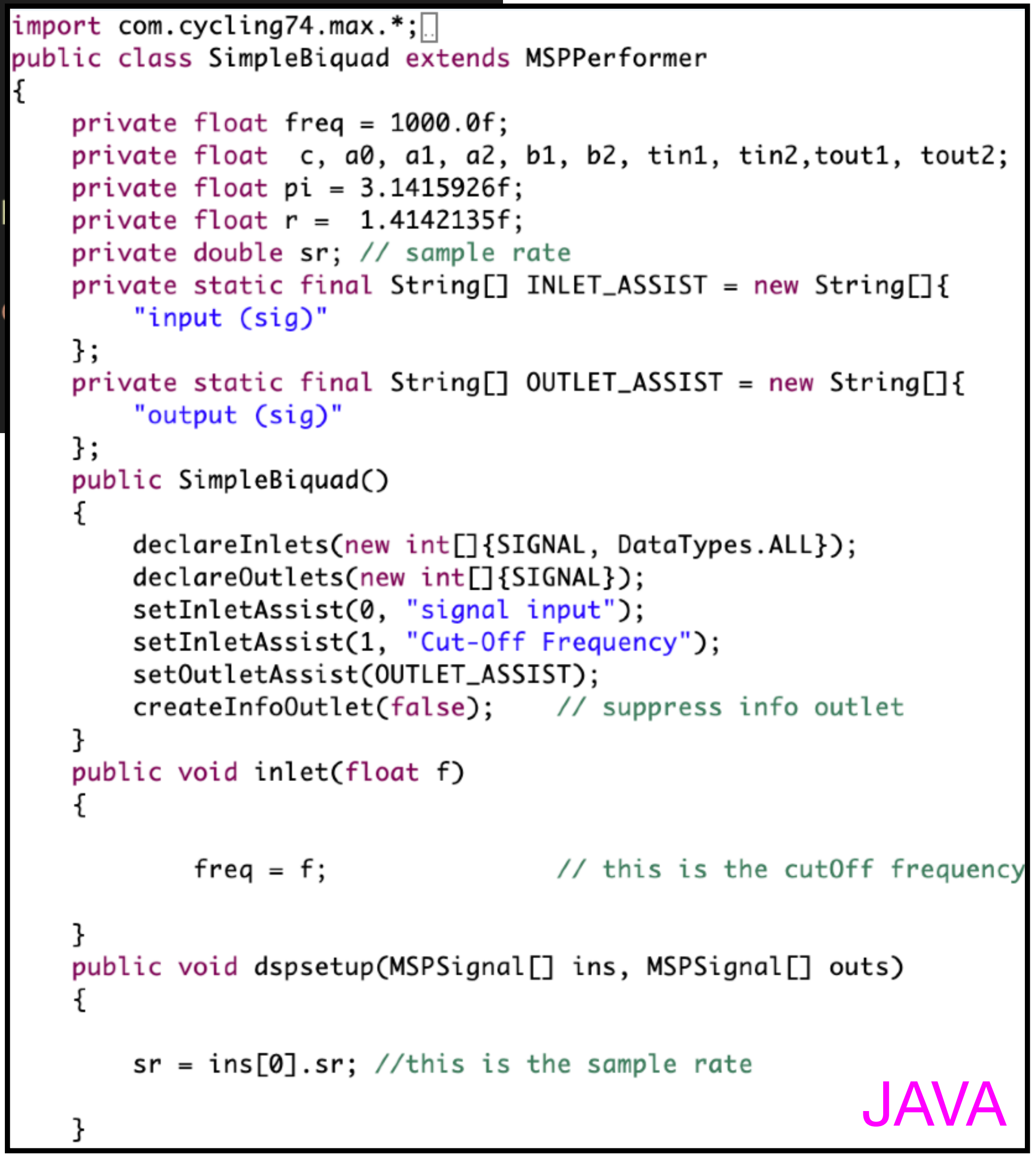
```
body {
    margin: 0px;
    font-family: 'Courier New', Courier, monospace;
}
h1 {
    margin: 0px;
    padding: 10px;
    background-color: 🟦rgl
    color: ⬜white;
    -webkit-user-select: n

}
```

CSS



```
function mouseRealeased() {
    initializeIndex();
}
function keyReleased(){
    initializeIndex();
}

function frequencyModulation(i,j){

    var scalar1=(random(300)+50)*random([-1,1]);
    i.osc.freq(j.freqModulator.mult(scalar1));

    var scalar2=(random(300)+50)*random([-1,1]);
    j.osc.freq(i.freqModulator.mult(scalar2));
}
```

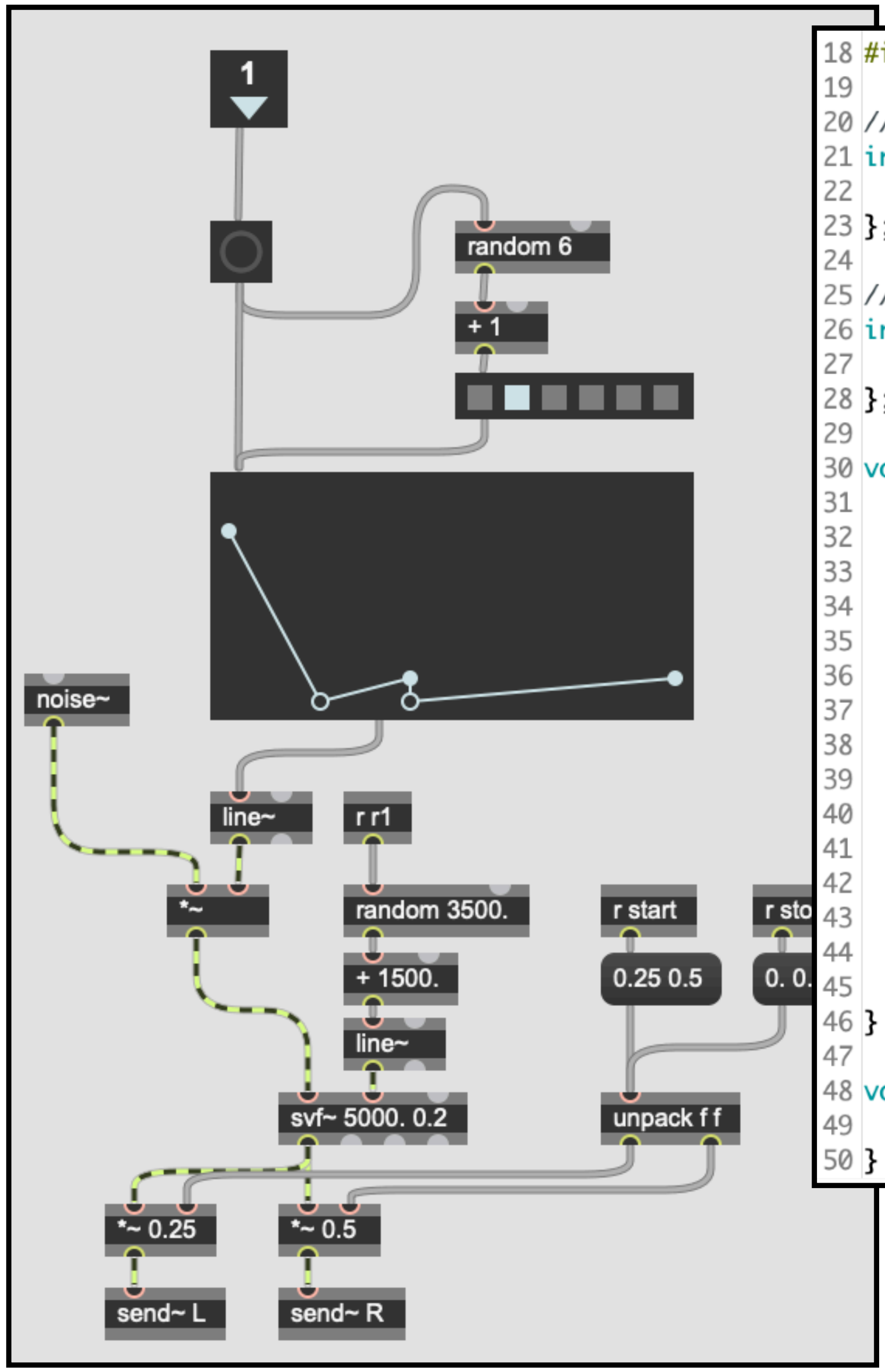
JAVASCRIPT



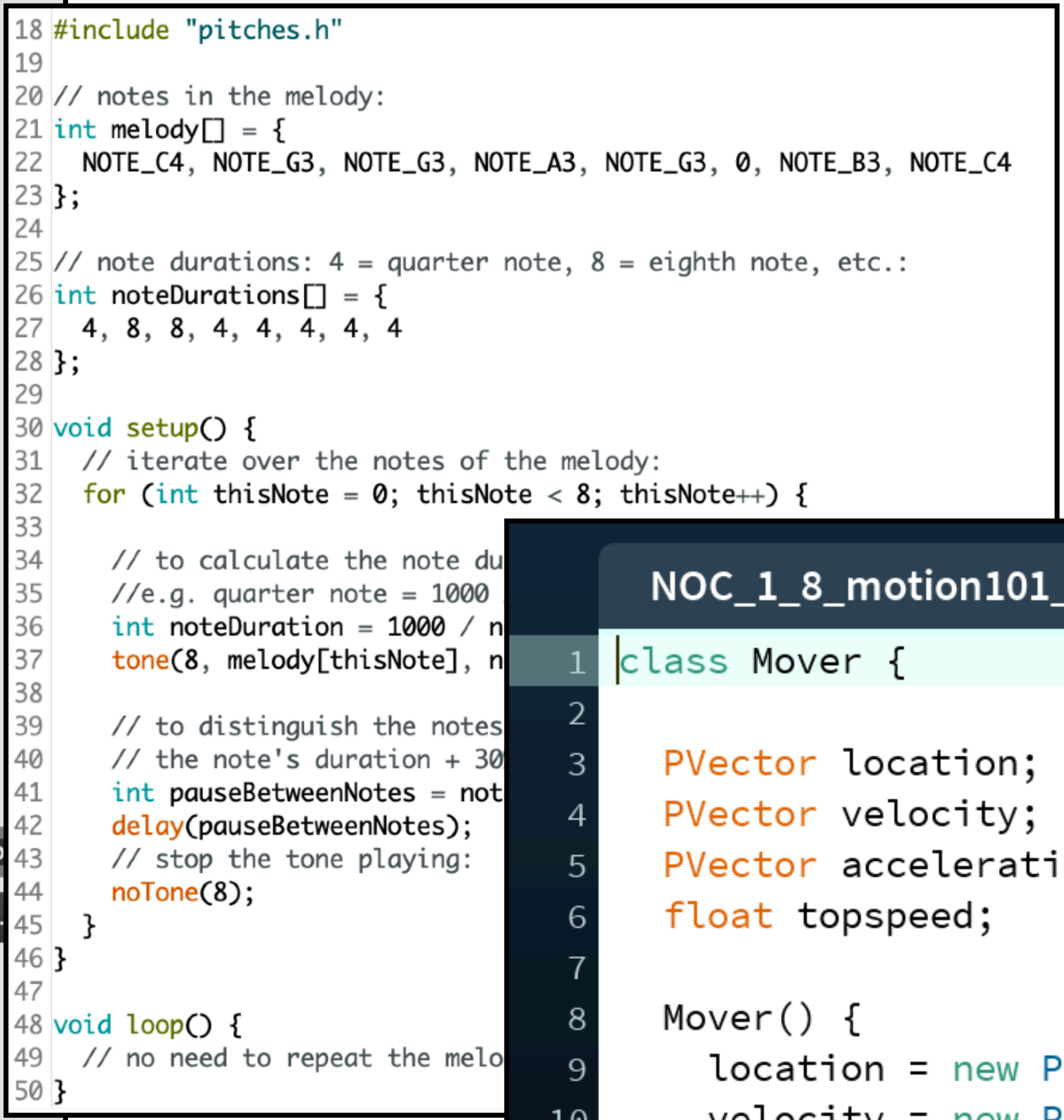
```
import com.cycling74.max.*;
public class SimpleBiquad extends MSPPerformer
{
    private float freq = 1000.0f;
    private float  c, a0, a1, a2, b1, b2, tin1, tin2,tout1, tout2;
    private float pi = 3.1415926f;
    private float r =  1.4142135f;
    private double sr; // sample rate
    private static final String[] INLET_ASSIST = new String[]{
        "input (sig)"
    };
    private static final String[] OUTLET_ASSIST = new String[]{
        "output (sig)"
    };
    public SimpleBiquad()
    {
        declareInlets(new int[]{SIGNAL, DataTypes.ALL});
        declareOutlets(new int[]{SIGNAL});
        setInletAssist(0, "signal input");
        setInletAssist(1, "Cut-Off Frequency");
        setOutletAssist(OUTLET_ASSIST);
        createInfoOutlet(false);    // suppress info outlet
    }
    public void inlet(float f)
    {

            freq = f;               // this is the cutOff frequency

    }
    public void dspsetup(MSPSignal[] ins, MSPSignal[] outs)
    {

        sr = ins[0].sr; //this is the sample rate

    }
```

JAVA

1.2 WHAT IS A PROGRAMMING LANGUAGE



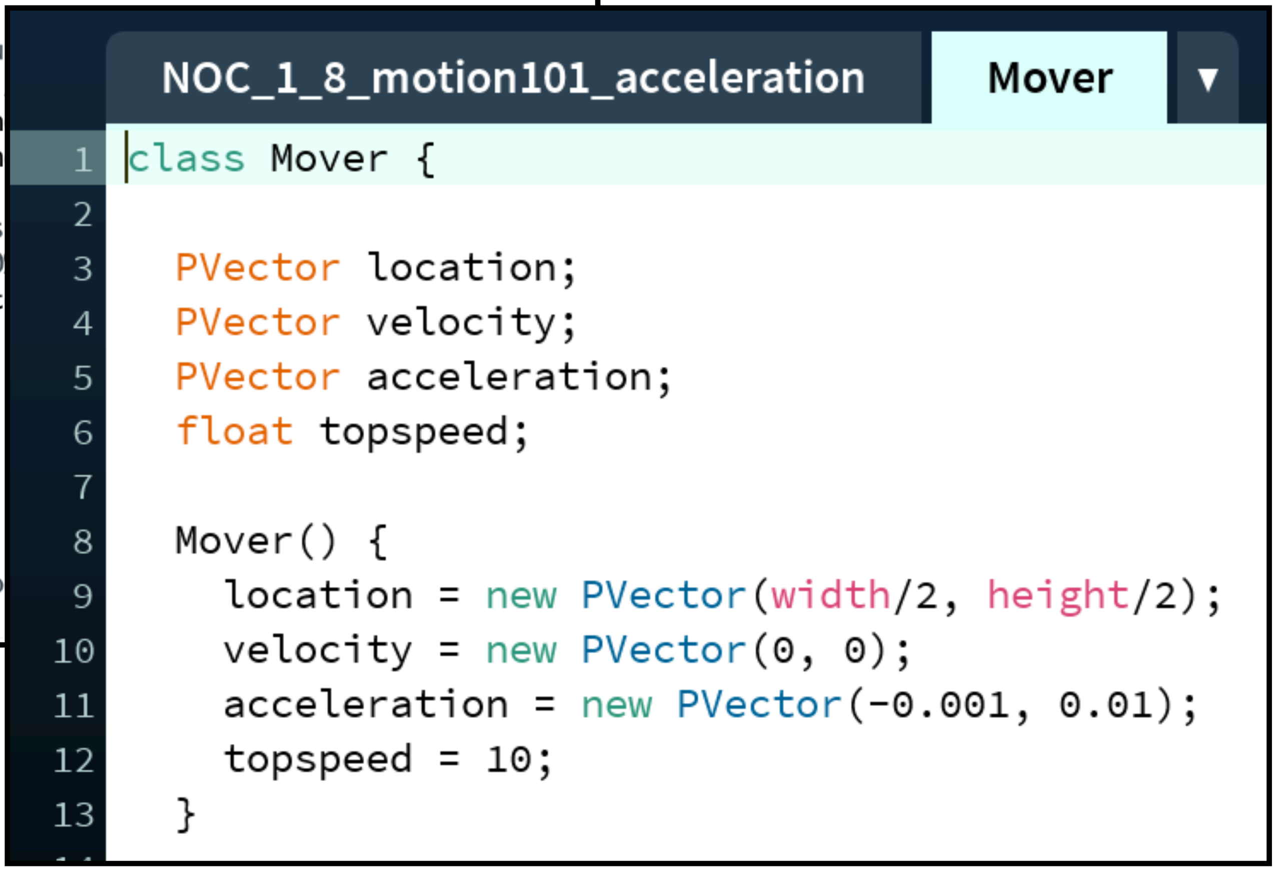
MAX/MSP



```
#include "pitches.h"

// notes in the melody:
int melody[] = {
  NOTE_C4, NOTE_G3, NOTE_G3, NOTE_A3, NOTE_G3, 0, NOTE_B3, NOTE_C4
};

// note durations: 4 = quarter note, 8 = eighth note, etc.:
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};

void setup() {
  // iterate over the notes of the melody:
  for (int thisNote = 0; thisNote < 8; thisNote++) {

    // to calculate the note du
    //e.g. quarter note = 1000
    int noteDuration = 1000 / n
    tone(8, melody[thisNote], n

    // to distinguish the notes
    // the note's duration + 30
    int pauseBetweenNotes = not
    delay(pauseBetweenNotes);
    // stop the tone playing:
    noTone(8);
  }
}

void loop() {
  // no need to repeat the melo
}
```

ARDUINO IDE (C++)

PROCESSING (JAVA)



```
class Mover {

  PVector location;
  PVector velocity;
  PVector acceleration;
  float topspeed;

  Mover() {
    location = new PVector(width/2, height/2);
    velocity = new PVector(0, 0);
    acceleration = new PVector(-0.001, 0.01);
    topspeed = 10;
  }
```

highSCORE

1.2 WHAT IS A PROGRAMMING LANGUAGE

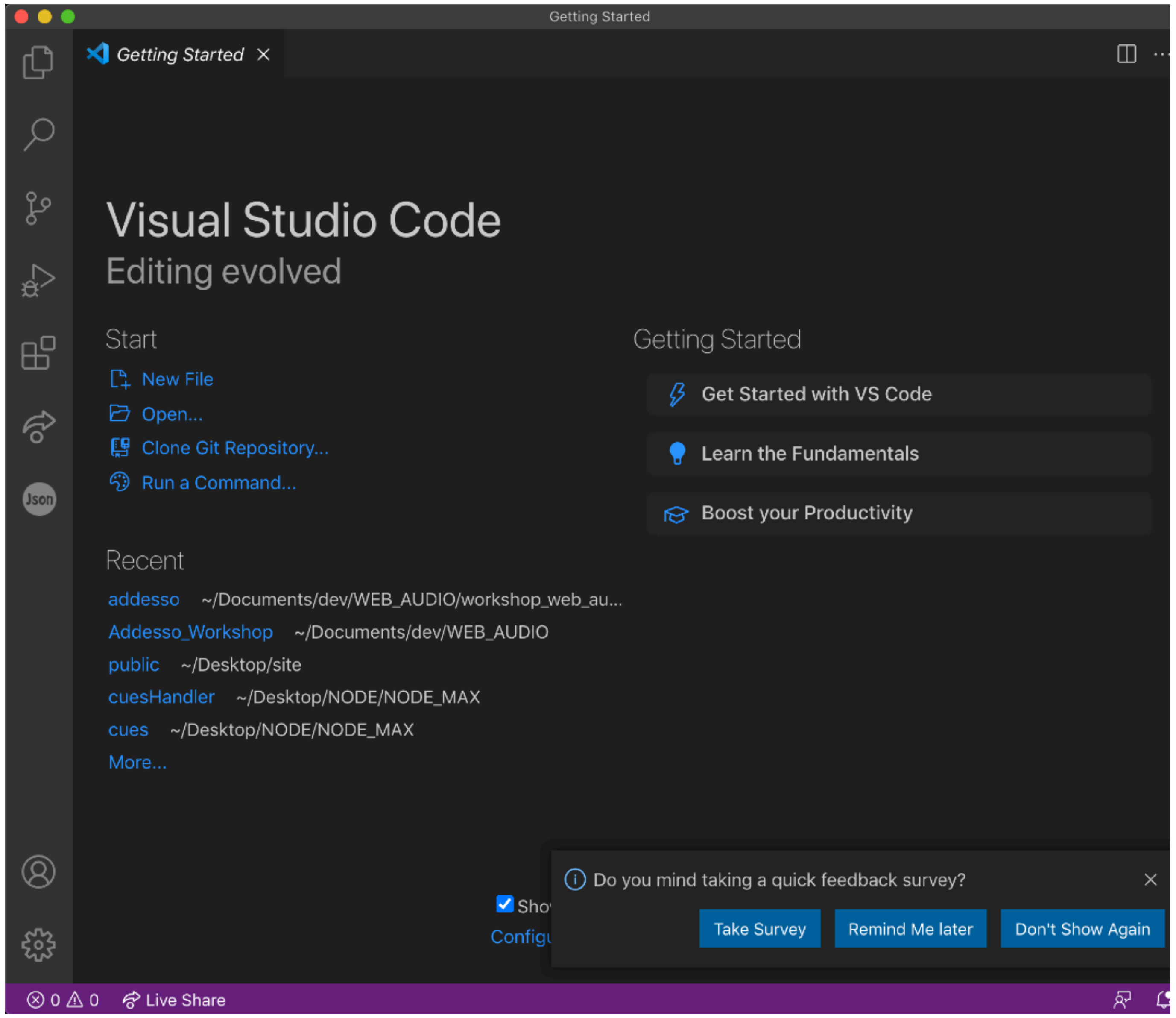
Usually a **developer** writes the source code in a **text editor** or **IDE** (Integrated Development Environment) and creates a source file (.html, .js, .css, .java, etc.).

Even if a simple text editor can be used, there are **dedicated editors**, which contain many support features: Atom, Visual Studio Code, Notepad++, Sublime Text, etc.

Then the **source code** is *compiled* or *interpreted* to be transformed into a code understandable and executable by the machine itself (the *machine code*).

1.2 WHAT IS A PROGRAMMING LANGUAGE

https://visualstudio.microsoft.com/



If you don't have a text editor you can download and install **Visual Studio Code** (it's free).

For the workshop is better to use **Google Chrome** (better for the web audio API compatibility).

1.3 WHAT IS THE WEB PROGRAMMING

The term **web programming** refers to all programming activities, languages, practices, and technologies that enable the creation of **web applications**.

A **web application** is a **dynamic website**, where is required a high user interaction (opposed to static web pages).

Web development is divided into several areas:

- Web design;
- GUI (graphic user interface) development;
- Client-side programming and scripting;
- Server-side development;
- Database management;
- Network security;
- ...

1.3 WHAT IS THE WEB PROGRAMMING

The coding involved in Web development includes different languages:

- **markup languages** (HTML, CSS): define the structure, the organization, and the look and feel of a site.

- **client-side scripting languages** (javascript): transform the website from a static page to an interactive application; client-side code runs in the browser.

- **server-side scripting languages** (PHP): server-side code lives on a server, and serves as go-between architecture, transferring data to the browser, etc..

Do we need all of this for making music on the browser?

1.3 WHAT IS THE WEB PROGRAMMING

We do not need to be a full-stack web developer to make music on the web! **thankfully.... :)**

We only need a **client-side code** and a **web browser** able to read it (we don't need to manage a database, to organize a large amount of data, or to store user data, to implement security protocols, etc.).

In the development phase, working locally on our machine, we won't even need a server where we can save (and request) our client-side code because the browser can load directly from our file system the *entry point* (the *index.html* file) of the client-side code, **a simple HTML file with some javascript**.

1.4 WHAT IS AN API

In the context of computer programming, an **Application Programming Interface** (API) represents a **set of procedures** for carrying out a given task.

An API lists a **bunch of operations** that developers can use, along with a description of what they do (the **API documentation**).

The developer doesn't necessarily need to know how they work internally, they just **need to know how to use them**.

The term can also indicate a **software library** of a programming language.

1.4 WHAT IS AN API

An API **reduces the amount of code developers need to create**, and also helps create **more consistency apps** for the same platform.

In last years, the specification and implementation of new APIs in web browsers has allowed for

envisioning the web platform as a fertile play-ground for artists and musicians / composers.

1.5 WHAT IS THE WEB AUDIO API

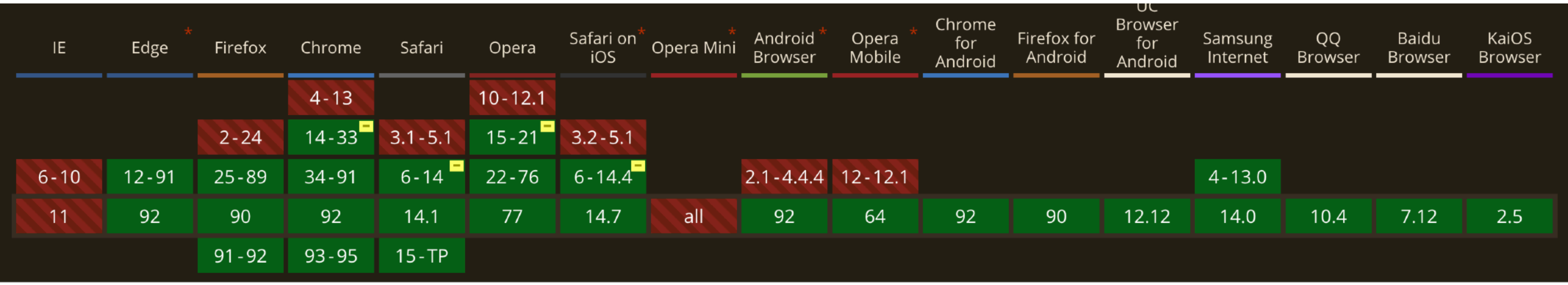
The **web audio API** is a **high level javascript API** developed by W3C **used for processing and synthesizing audio in web applications**.

The **web audio API** provides a powerful system for controlling audio on the Web, allowing to choose audio sources, add effects to audio, create audio visualization, compose with algorithmic procedures, apply spatial effects, be creative!

OFFICIAL DOCUMENTATION: https://www.w3.org/TR/webaudio/

MOZILLA DOCUMENTATION: https://developer.mozilla.org/en-US/docs/Web/API/Web_Audio_API

1.6 WEB AUDIO API BROWSER SUPPORT



IE	Edge *	Firefox	Chrome	Safari	Opera	Safari on iOS *	Opera Mini *	Android Browser *	Opera Mobile *	Chrome for Android	Firefox for Android	UC Browser for Android	Samsung Internet	QQ Browser	Baidu Browser	KaiOS Browser
			4-13		10-12.1											
		2-24	14-33	3.1-5.1	15-21	3.2-5.1										
6-10	12-91	25-89	34-91	6-14	22-76	6-14.4		2.1-4.4.4	12-12.1				4-13.0			
11	92	90	92	14.1	77	14.7	all	92	64	92	90	12.12	14.0	10.4	7.12	2.5
		91-92	93-95	15-TP												

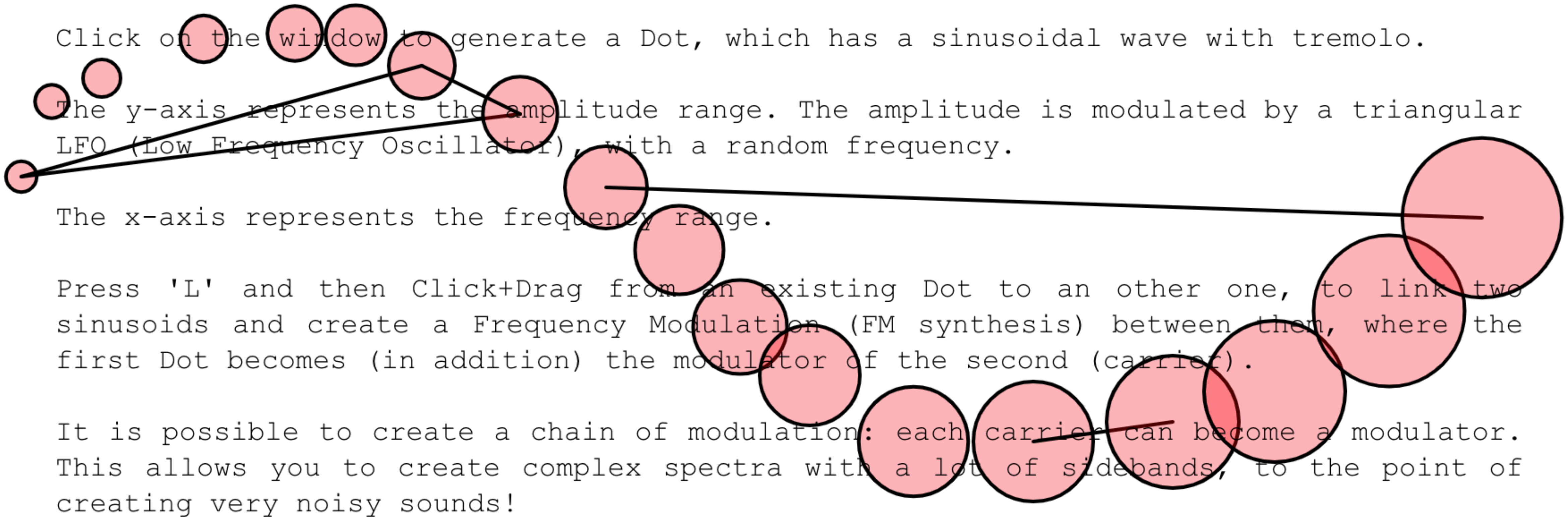
https://caniuse.com/?search=web%20audio

The **web audio API** is supported in most of the modern browsers (desktop or mobile).

highSCORE

2. USES OF THE WEB AUDIO

Dot Drone Generator is a drone generator which allows you to create synthetic textures and chords directly on your browser.



Click on the window to generate a Dot, which has a sinusoidal wave with tremolo.

The y-axis represents the amplitude range. The amplitude is modulated by a triangular LFO (Low Frequency Oscillator), with a random frequency.

The x-axis represents the frequency range.

Press 'L' and then Click+Drag from an existing Dot to an other one, to link two sinusoids and create a Frequency Modulation (FM synthesis) between them, where the first Dot becomes (in addition) the modulator of the second (carrier).

It is possible to create a chain of modulation: each carrier can become a modulator. This allows you to create complex spectra with a lot of sidebands, to the point of creating very noisy sounds!

Click on an existing circle to delete it or to delete the modulation chain of which it is part.

2.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO USES

1. **play audio tracks** in the browser with a better control over panning and volumes, and process them (DSP);
2. performing **sound analysis** in the time domain and frequency domain (waveform and spectrum);
3. develop **sound visualization applications**; integrate the web audio API with web graphics libraries (webGL, three.js);
4. creating **web applications for music education** (music theory, ear-training, etc.);
5. creating **generative compositions** living in the browser (exploiting the algorithmic logic offered by the programming language javascript and the whole ecosystem of its libraries and extensions);

2.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO USES

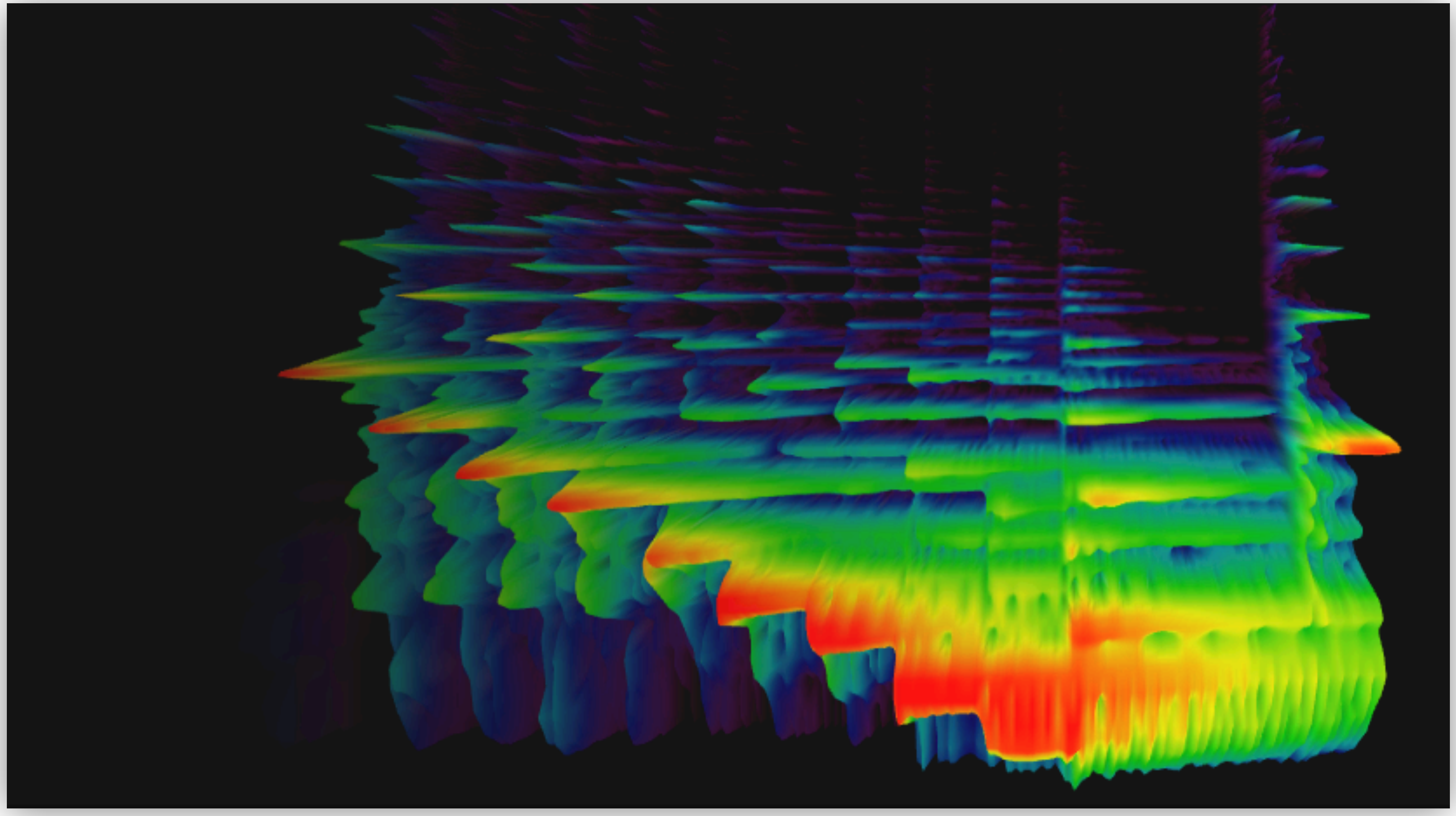
6. create **interactive compositions**, in which the users play an important role in the creative process;
7. real-time **data analysis and data sonification**; exploiting data analysis for sound synthesis and composition, taking advantage of the many APIs offered by the web to collect and receive data in real time;
8. create **not-fixed media music**; creating compositions that can be accessed anywhere and from different devices;
9. develop **synths and web musical instruments**;
10. use the web audio along with **web MIDI** to create integrated browser / desktop software / MIDI interface environments;

2.1 (NON-EXHAUSTIVE) LIST OF WEB AUDIO USES

11. develop **web audio effects for real-time** DSP; with the AudioWorklet it is possible to create high-performance DSP algorithms that work at the sample level (there are some free ones, such as *genish.js*);
12. create **environments for live-coding** and **online music performances** that take advantage of the web technologies and web support;
13. exploit the possibilities of **integration with the entire web ecosystem** (extensions and libraries) to do almost anything!
14. exploit the web audio to create **web IDEs** that interpret other audio languages (e.g. Csound: https://ide.csound.com/);

2.2 SOME EXAMPLES

AUDIO ANALYZER



https://musiclab.chromeexperiments.com/Spectrogram/

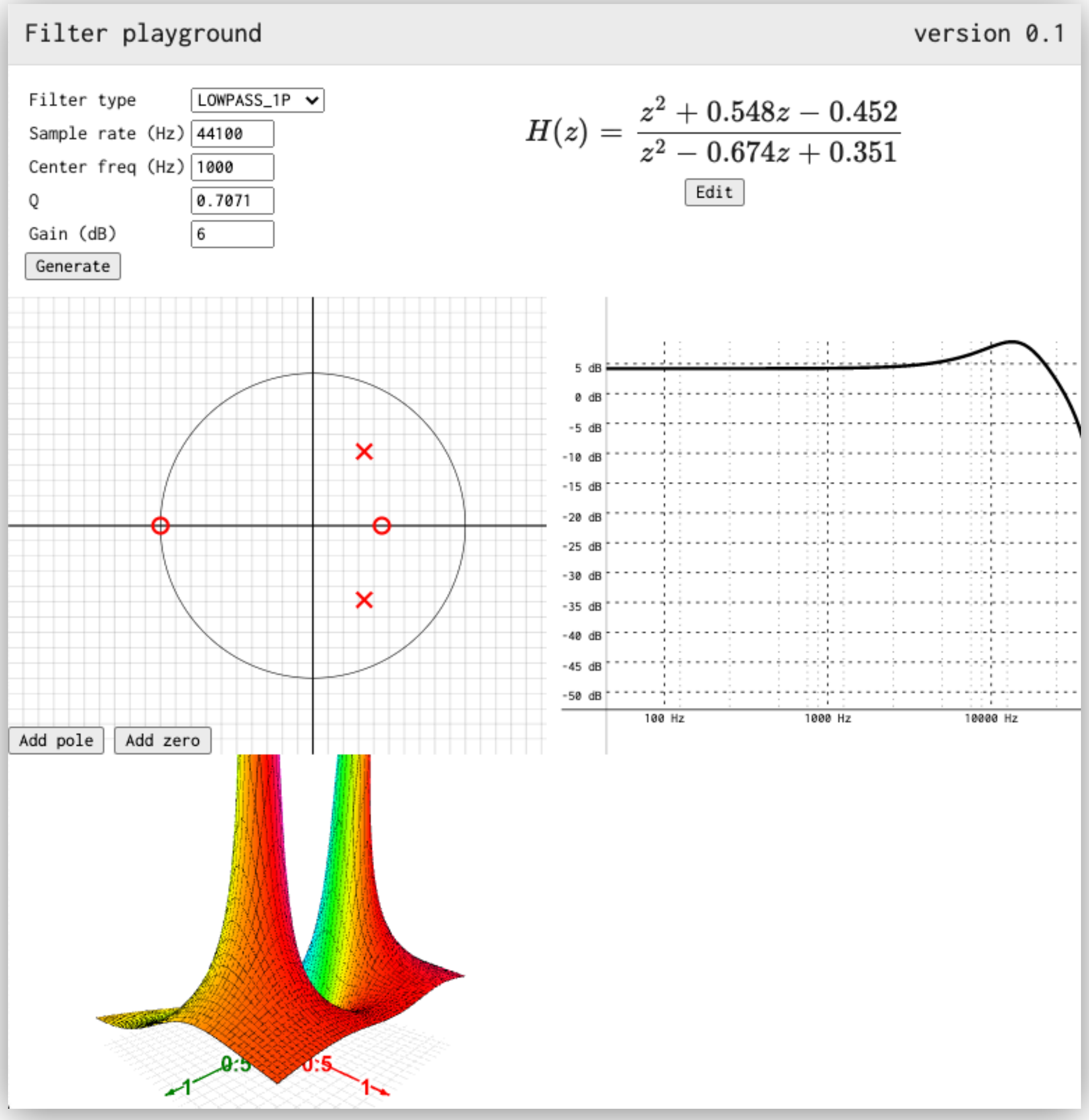
2.2 SOME EXAMPLES

VOICE - SPINNER

https://musiclab.chromeexperiments.com/Voice-Spinner/

2.2 SOME EXAMPLES

FILTER DESIGN


Filter playground version 0.1

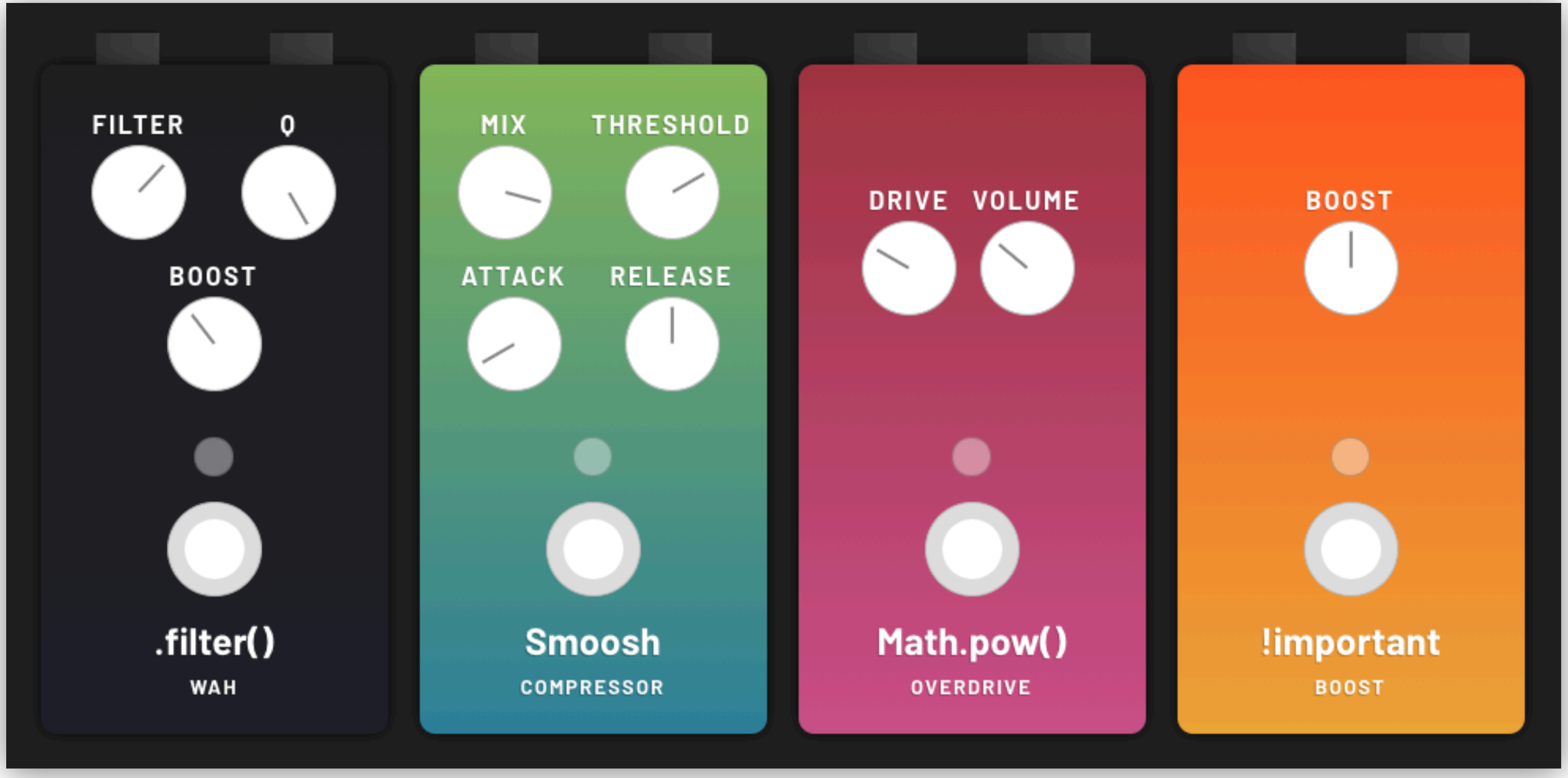
Filter type LOWPASS_1P
Sample rate (Hz) 44100
Center freq (Hz) 1000
Q 0.7071
Gain (dB) 6

$$H(z) = \frac{z^2 + 0.548z - 0.452}{z^2 - 0.674z + 0.351}$$

https://borismus.github.io/filter-playground/?equation=%28z%5E2%2B0.5475285649299622z-0.45247143507003784%29%2F%28z%5E2-0.6741842031478882z%2B0.3505924277806516%29

2.2 SOME EXAMPLES

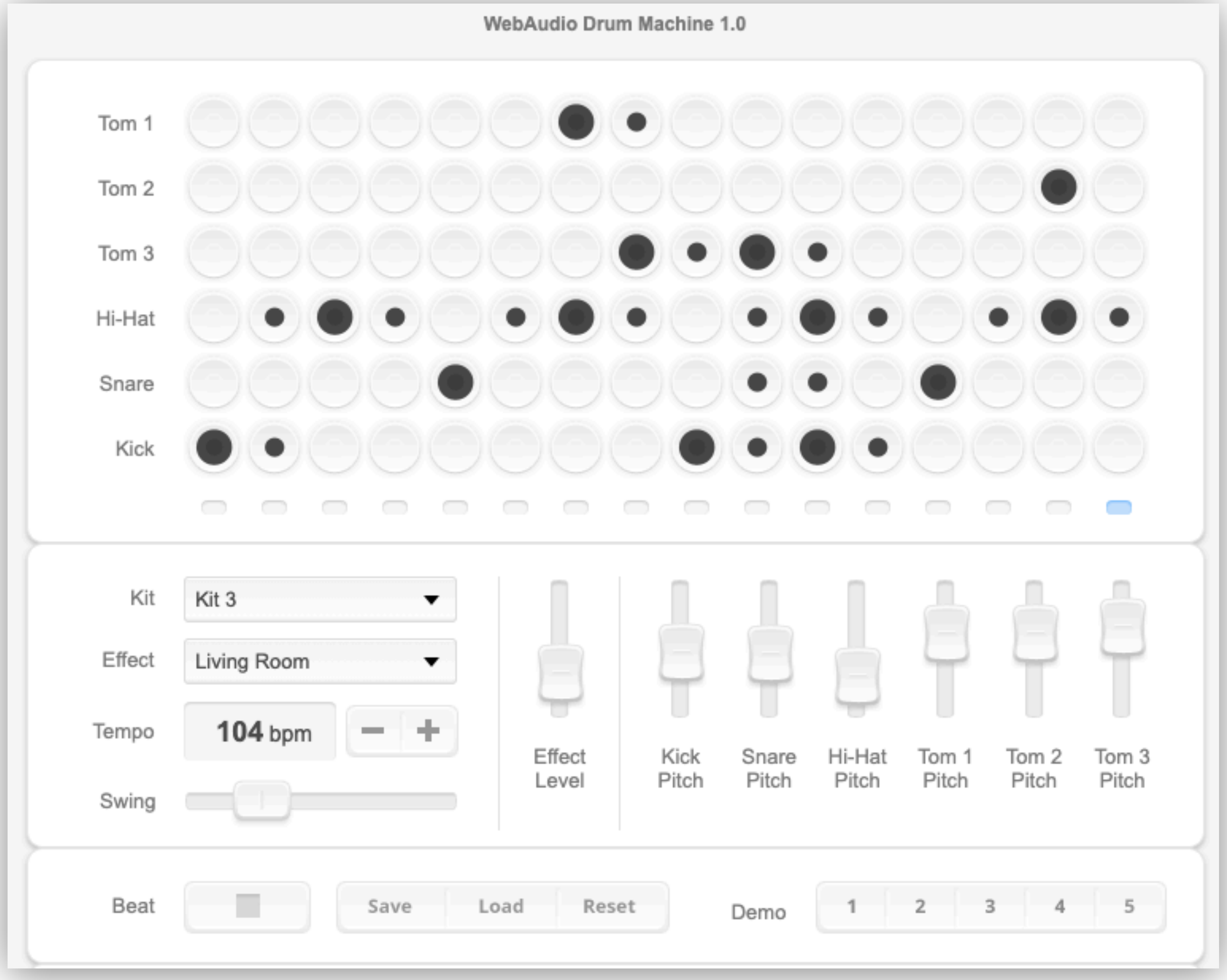
PEDAL BOARD



https://pedalboard.netlify.app/

2.2 SOME EXAMPLES

DRUM MACHINE

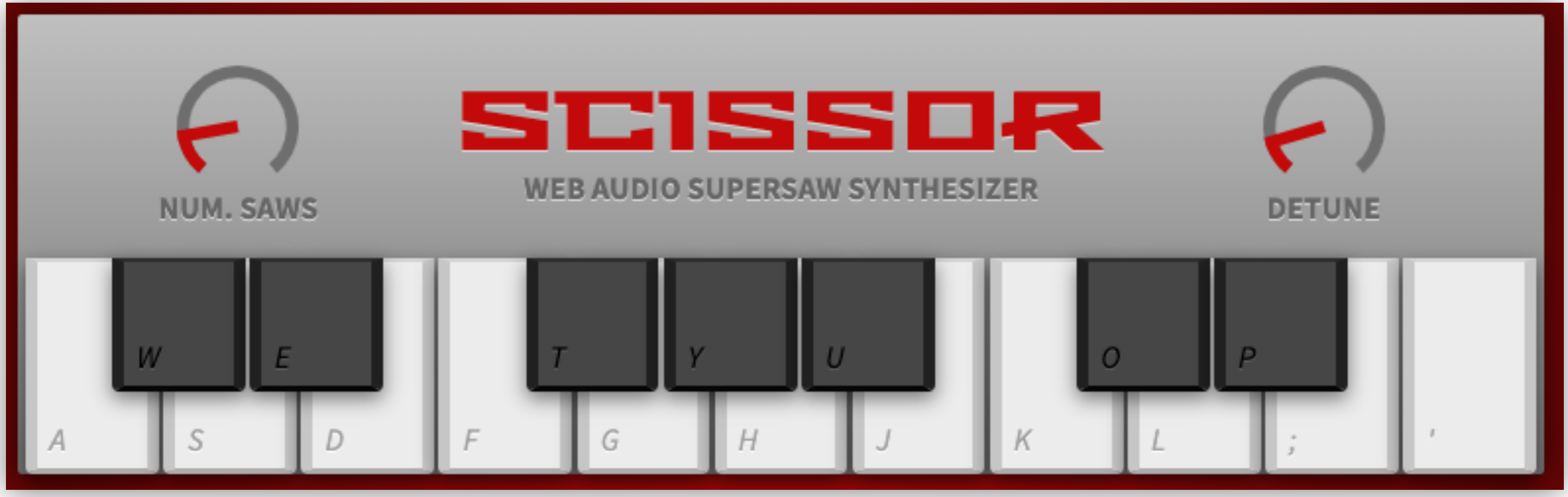


Kit: Kit 3
Effect: Living Room
Tempo: 104 bpm
Swing

https://webaudiodemos.appspot.com/MIDIDrums/index.html

2.2 SOME EXAMPLES

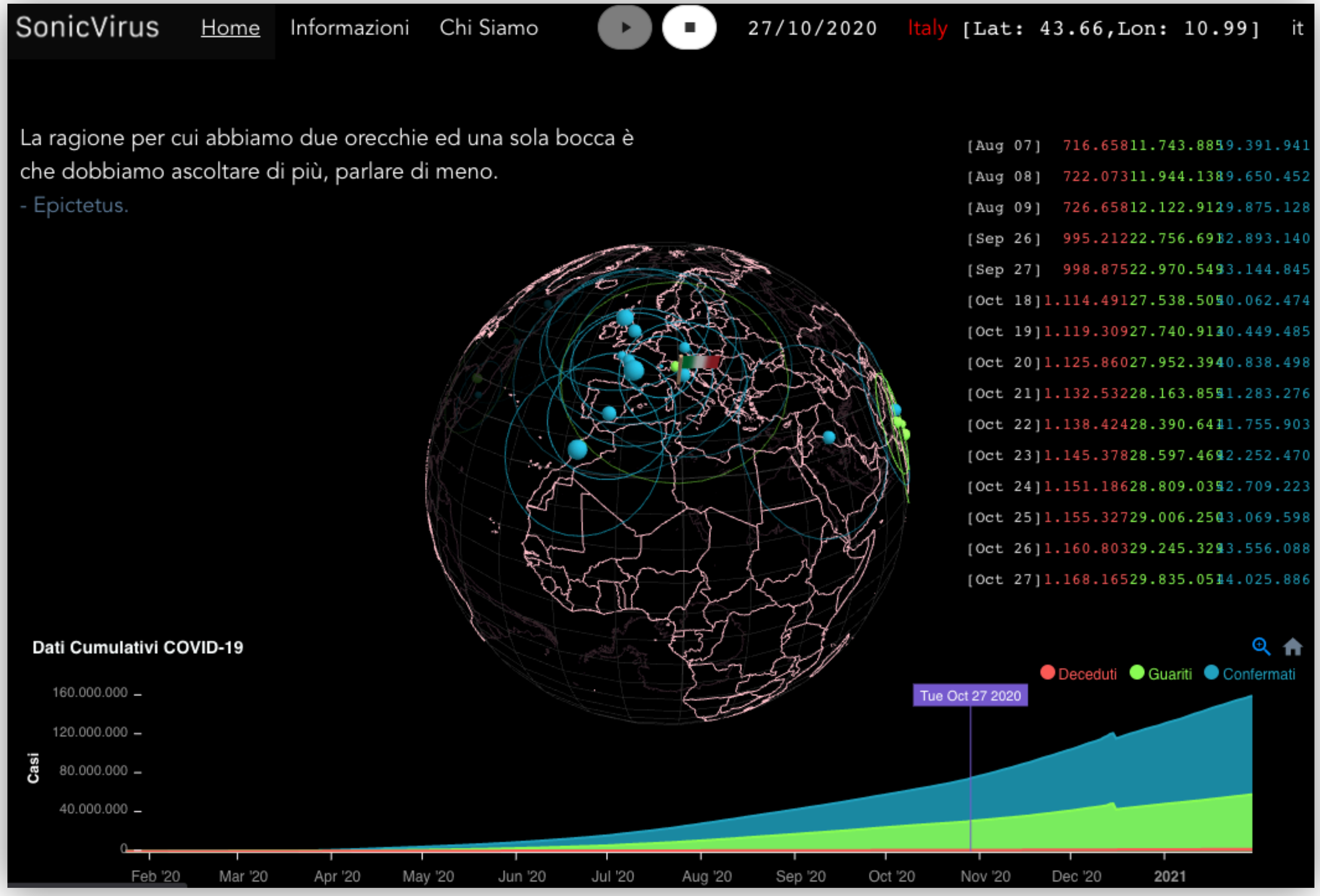
SYNTHESIZER



https://noisehack.com/scissor/

2.2 SOME EXAMPLES

DATA SONIFICATION



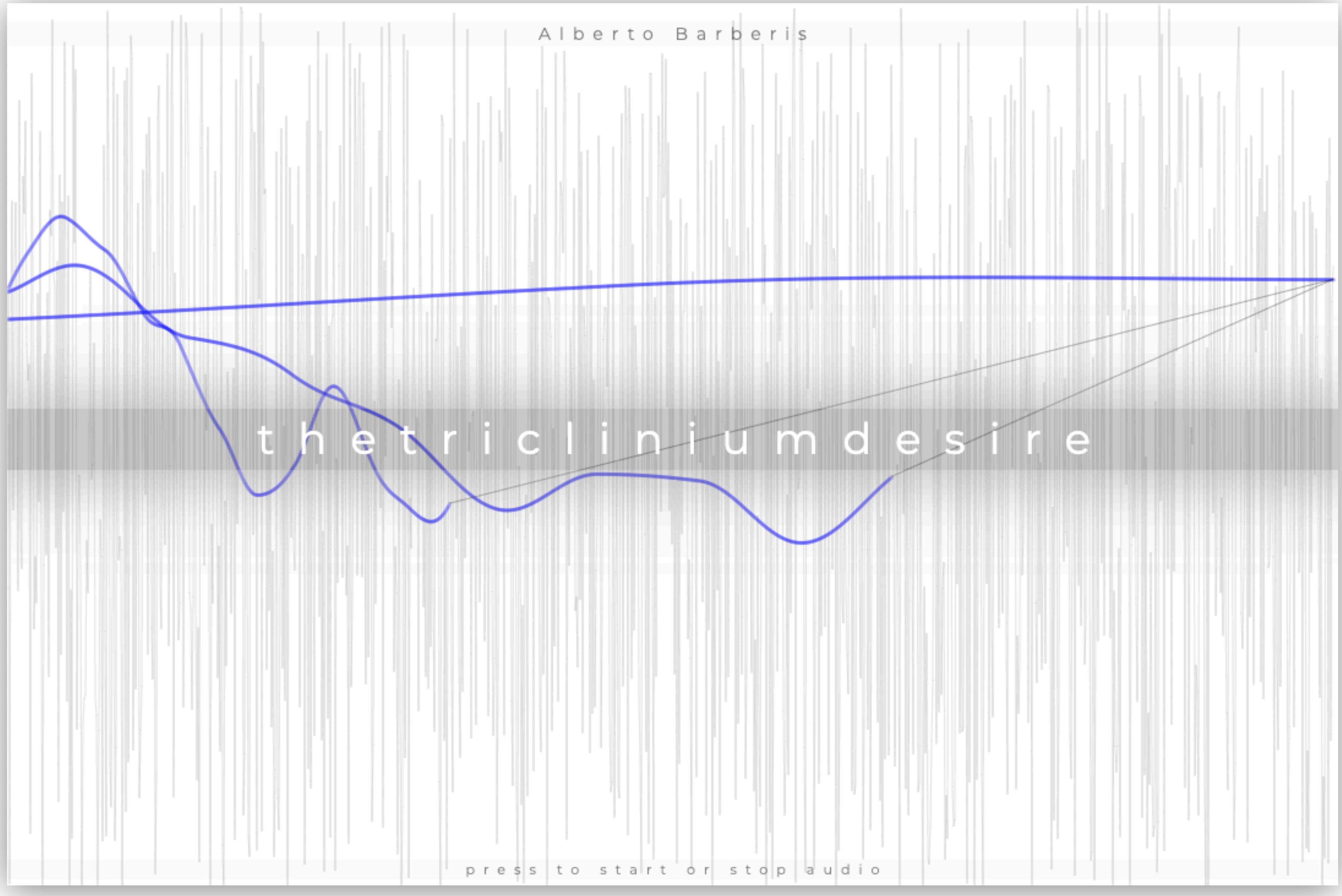
La ragione per cui abbiamo due orecchie ed una sola bocca è che dobbiamo ascoltare di più, parlare di meno.
- Epictetus.

```
[Aug 07]  716.65811.743.8859.391.941
[Aug 08]  722.07311.944.1389.650.452
[Aug 09]  726.65812.122.9129.875.128
[Sep 26]  995.21222.756.6912.893.140
[Sep 27]  998.87522.970.5493.144.845
[Oct 18]1.114.49127.538.5050.062.474
[Oct 19]1.119.30927.740.9130.449.485
[Oct 20]1.125.86027.952.3940.838.498
[Oct 21]1.132.53228.163.8551.283.276
[Oct 22]1.138.42428.390.6441.755.903
[Oct 23]1.145.37828.597.4692.252.470
[Oct 24]1.151.18628.809.0352.709.223
[Oct 25]1.155.32729.006.2503.069.598
[Oct 26]1.160.80329.245.3293.556.088
[Oct 27]1.168.16529.835.0514.025.886
```

https://sonicvirus.si.usi.ch/

2.2 SOME EXAMPLES

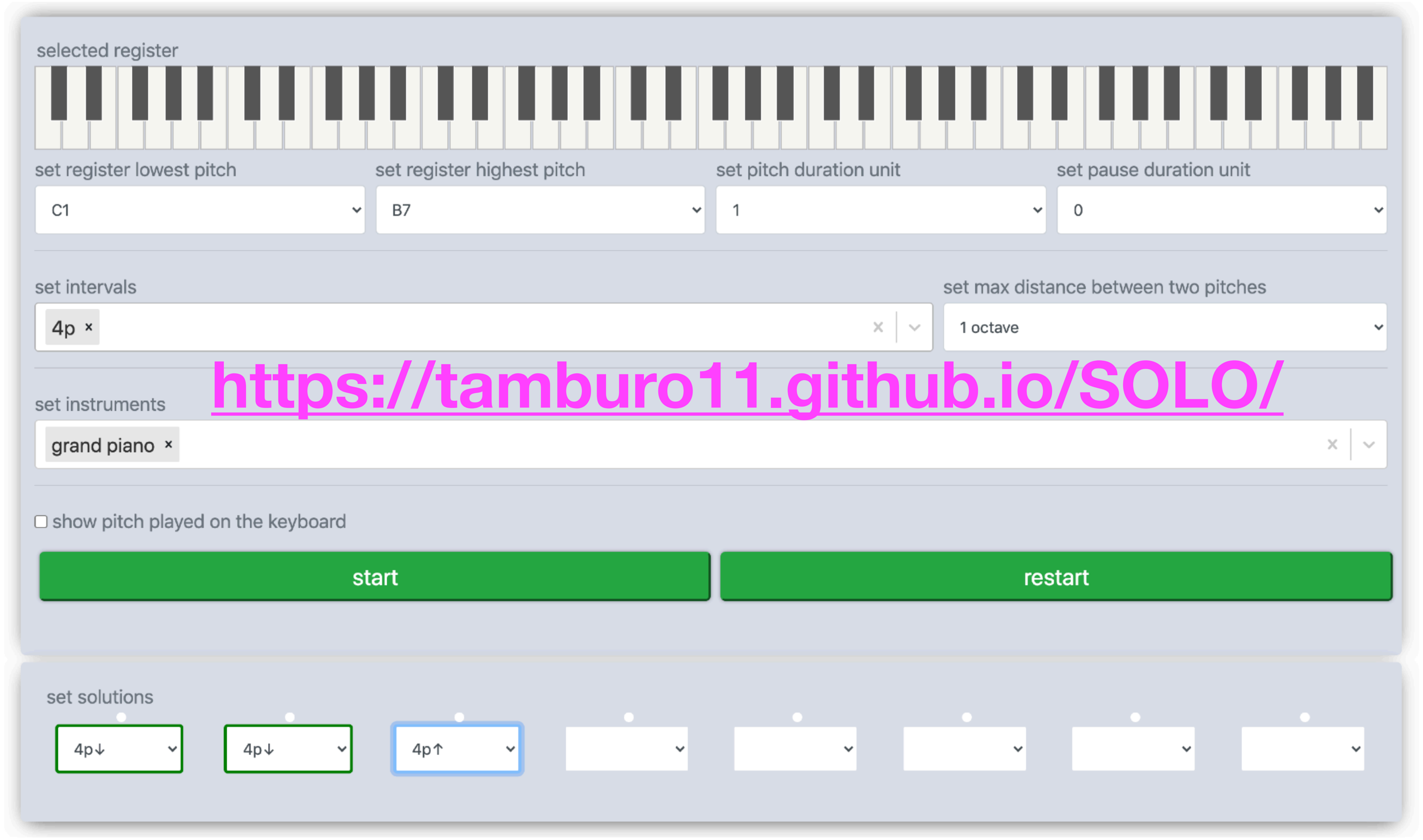
GENERATIVE MUSIC



https://tamburo11.github.io/thetricliniumdesireWeb/

2.2 SOME EXAMPLES

MUSIC EDUCATION


selected register

set register lowest pitch: C1
set register highest pitch: B7
set pitch duration unit: 1
set pause duration unit: 0

set intervals: 4p ×
set max distance between two pitches: 1 octave

https://tamburo11.github.io/SOLO/

set instruments: grand piano ×

☐ show pitch played on the keyboard

set solutions: 4p↓ 4p↓ 4p↑