



Audio on the Web (a brief history)

Michel Buffa

► To cite this version:

Michel Buffa. Audio on the Web (a brief history). SMC 2022 - Sound and Music Computing, Jun 2022, Saint-Etienne, France. hal-03871527

HAL Id: hal-03871527

<https://inria.hal.science/hal-03871527>

Submitted on 25 Nov 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Audio on the Web (a brief history)

Sound and Music Computing 2022

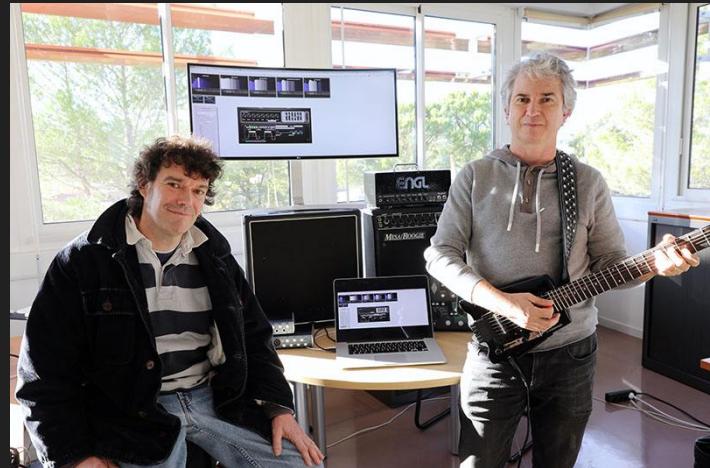
11 June 2022

Michel Buffa,
Université Côte d'Azur, France,
I3S/CNRS/INRIA labs
buffa@i3s.unice.fr, [@micbuffa](https://twitter.com/micbuffa)



Who am I?

- Professor / researcher at Université Côte d'Azur, in the South of France
 - Member of the WIMMICS research group common to INRIA and I3S lab from CNRS
 - W3C Advisory Committee Representative for UCA
- I participate to W3C working groups (HTML5, WebAudio...)
- I'm the co-creator of the
Web Audio Modules 2.0 (WAM or WAM2),
aka VSTs for the Web.
- Some WebAudio plugins I wrote are
now commercialized by CNRS (tube guitar
amplifier simulators + audio FXs)



Before 1996
we barely could hear
any sound from a browser

Non sécurisé | users.polytech.unice.fr/~buffa/

Applications 10 of the best way... Get started | Lear... Zenodo (88) SMC-22 Live... Audit Qualiopi - 2... Demo audio - Goo... Rapport CSI - Goo... Thèse Maroua Tik... Mé

Welcome to Michel's home page!

buffa@unice.fr



[Who am I? What is my work...](#)
[My coordinates](#)
[My hobbies](#)
[My courses](#)

You want to see me live? Check [this funny anim](#) and also [this new one!](#)

Table of contents

- [Internet, WWW, etc...](#)
- [Teaching computer techniques, University of Nice related stuff](#)
- [Research centers I have worked with](#)
- [Culture...](#)
- [Unix related stuff, X11/Motif/Tk, ftp, fsp, etc...](#)
- [Virtual Reality/3D/Computer Graphics](#)
- [Dictionnaires, maps, etc...](#)
- [Interesting personal Home Pages](#)
- [Pure Fun](#)
- [Business WWW sites](#)

Non sécurisé | users.polytech.unice.fr/~buffa/#courses

Applications 10 of the best way... Get started | Lear... Zenodo (88) SMC-22 Live... Audit Qualiopi - 2... Demo audio - Goo... Rapport CSI - Goo... Thèse Maroua Tik... Mé

My courses

- [Computer Graphics and OpenGL Programming](#)
- [Internet and WWW](#)
- [Java](#)
- [X11 and Motif Programming](#)
- [VRML](#)
- [Zsh](#)
- [An old WWW tutorial I wrote in 1994](#)
- [others...](#)

After 1996

Horrors and Joys

The infamous <bgsound> tag from IE4! (1996)

HTML <bgsound> tag

please plugin speaker/headphone to listen to the sound.

Supported Browsers: The supported browsers by **HTML | <bgsound> tag** are listed below:

- Internet Explorer

**...also known as the
worst possible way
to use audio on the
web!**



I assume you are using the bgsound tag.



0

Sign
in to
vote

The bgsound tag is non-standard and IE-specific. Firefox does not support it.

I would strongly suggest that you do not use a background sound, unless this is a personal/family site where your friends and family know what to expect.

As a rule, visitors do not appreciate their family, co-workers, other coffee shop patrons, or sleeping babies being blasted with sound from their computer that they had no control over.

If you want to play sound, do it with a method that lets the visitor control starting it and stopping it; don't try to do it automatically.

Jehovah's Witnesses—Organized to Share the Good News

Error loading media:
File could not be played 

However, with a little struggle, we had great applications!



FXExperiencePlayer / Java FX (video from 2012)



Audiotools / Flash(video from 2013)

1996-2011

HTML



CSS



JS



dreamstime.com

ID 154897294 © Vasiliosca

2008-2010: HTML5 audio tag with
rather complete JavaScript API

ANOTHER CODEC WAR, until 2015
no way to play a mp3 on all major
browsers!

HTML <audio>



```
<audio controls>
  <source src="sound.ogg" type="audio/ogg">
  <source src="sound.mp3" type="audio/mpeg">
  Your browser does not support the audio tag.
</audio>
```



0:00 / 0:02



For serious audio applications: Flash or Die!



Soundation



Audio Sauna

2010
First browser APIs
for audio processing

Firefox Audio Data API (2010)

Simple: nearly no native code in the browser, all sound processing delegated to JavaScript, short spec!

web.archive.org/web/20100706012324/https://wiki.mozilla.org/Audio_Data_API

148 captures | 6 May 2010 – 6 Jun 2022

INTERNET ARCHIVE Wayback Machine

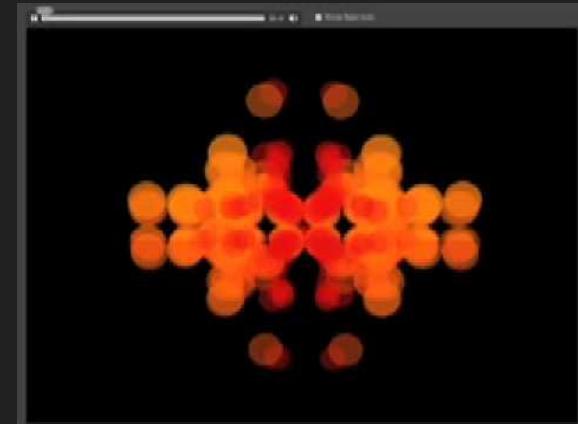
https://wiki.mozilla.org/Audio_Data_API

A number of working demos have been created, including:

NOTE: If you try to run demos created with the original API using a build that implements the new API, you may encounter [bug 560212](#). We are aware of this, as is Mozilla, and it is being investigated.

Demos Working on Current API

- FFT visualization (calculated with js)
 - <http://weare.buildingsky.net/processing/dsp.js/examples/fft.html> (video [here](#))
- Beat Detection (also showing use of WebGL for 3D visualizations)
 - <http://cubicvr.org/CubicVR.js/bd3/BeatDetektor1HD-13a.html> (video [here](#))
 - <http://cubicvr.org/CubicVR.js/bd3/BeatDetektor2HD-13a.html> (video of older version [here](#))
 - <http://cubicvr.org/CubicVR.js/bd3/BeatDetektor3HD-13a.html> (video [here](#))
 - <http://cubicvr.org/CubicVR.js/bd3/BeatDetektor4HD.html> (video [here](#))
- Writing Audio from JavaScript, Digital Signal Processing
 - JavaScript Text to Speech engine
<http://scotland.proximity.on.ca/dxr/tmp/audio/tts/index2.html>
 - JavaScript Audio Sampler
<http://weare.buildingsky.net/processing/dsp.js/examples/sampler.html>
 - Experimental JavaScript port Pure Data <http://mccormick.cx/dev/webpd/> with demo <http://mccormick.cx/dev/webpd/demos/processingjs-basic-example-with-audio/index.html>
 - Delay effect <http://code.almeros.com/code-examples/delay-firfox-audio-api/> (video of older version [here](#))
 - Csound shaker instrument ported to JavaScript via Processing.js
<http://scotland.proximity.on.ca/dxr/tmp/audio/shaker/instruments/shaker.htm>



The Web

Proposed by

lists.w3.org/Archives

Applications 10 of the best way...

[W3C home](#) > [Mailing lists](#)

Web Audio API

This message: [[Message](#) | [Related messages](#)]

From: Chris Rogers <crogers@chromium.org>

Date: Mon, 14 Jun 2010

Message-ID: <AANLkTik69Rr-

To: public-xg-audio@w3.org

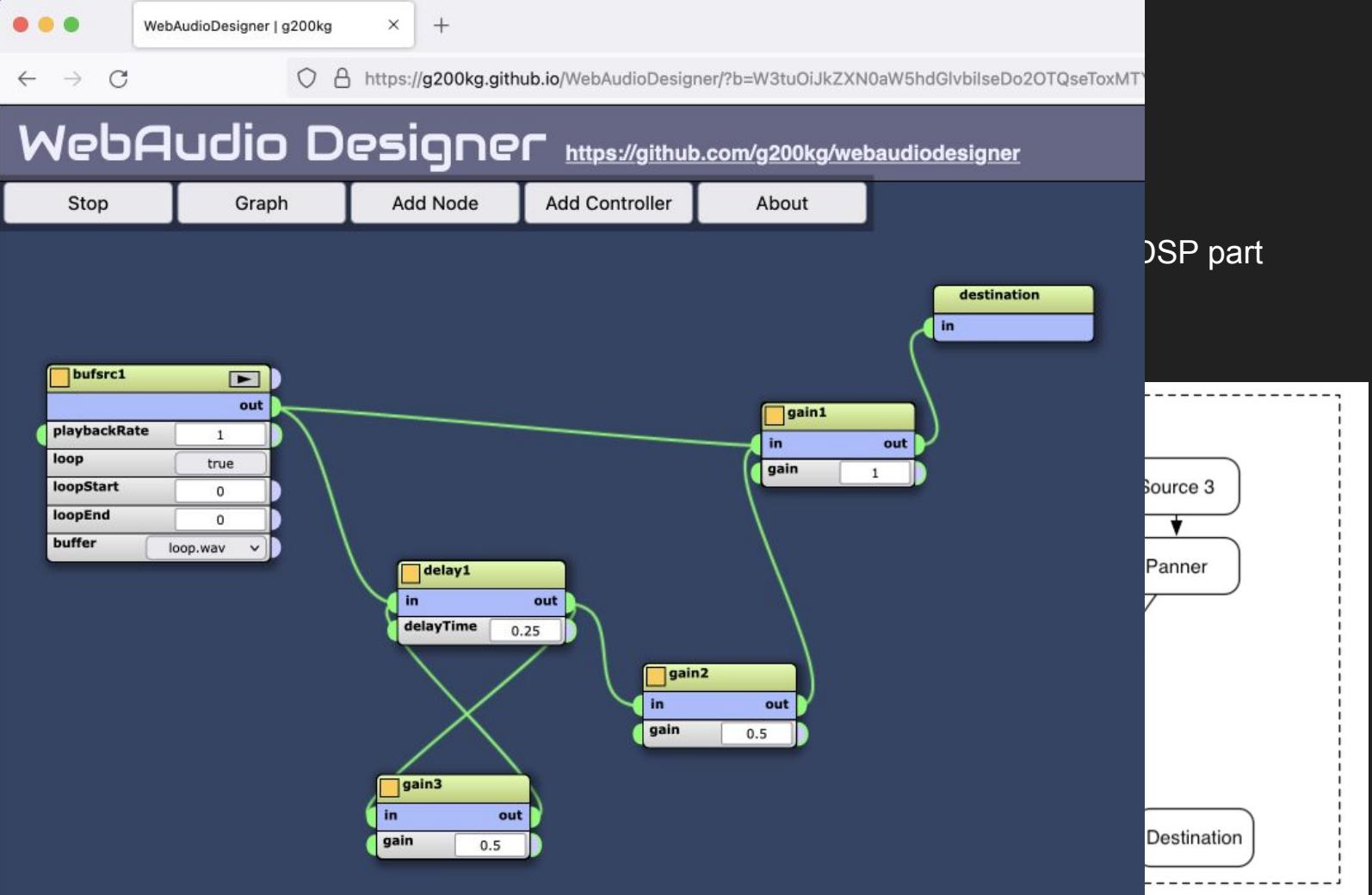
Hi Everyone,

Since it looks like there's this opportunity to improve developing over the past at Google as part of the team at Apple as an engineer.

I've been developing a <https://svn.webkit.org> (most of the interesting

I've been working with Carlsson, Chris Marrin, API to the point where group. Here's a preliminary

<http://chromium.google>



Web Audio Demos

[Playground \(a,i\)](#)

The Web Audio Playground helps developers visualize how the graph nodes in the Web Audio API work. Shown at I/O 2012.

[Vocoder \(a,i,m\)](#)

This complex audio processing app (shown at I/O 2012) implements a 28-band [vocoder](#) - a "robotic voice" processor.

["Analog" Synth \(a,m\)](#)

This application implements a polyphonic "analog" synthesizer, with a classic voice architecture. Playable via Web MIDI or onscreen keyboard.

[Input Effects \(a,i\)](#)

This demo lets you play with a few common effects on the audio inputs.

[AudioRecorder \(a,i\)](#)

This is sample code for recording audio from live input and downloading them as WAV files, built on Matt Diamond's excellent [RecorderJS](#).

[Oscilloscope \(a\)](#)

Oscilloscope waveform display for Web Audio, and pulse-width duty cycle control on a square wave.

[Monosynth \(a,m\)](#)

Example of a monophonic Web MIDI/Web Audio synth, with no UI.

[wubwubwub \(a,m\)](#)

This application implements a dual DJ deck, specifically intended to be driven by a Numark MIDI controller.

[Pitch Detection \(i\)](#)

This application performs monophonic autocorrelating pitch detection in realtime. This is suitable for a guitar tuner or other complex waveform source.

[The Game of Life \(m\)](#)

Conway's Game of Life, on an 8x8 grid, interfacing with a Novation Launchpad controller via Web MIDI.

[Happy Drums \(a,m\)](#)

Quick addition of MIDI controller support (using a Livid Instruments CNTRLR) to the [Shiny Drum Machine](#).

[TemplateSynth \(a,m\)](#)

Simple template for Web Audio synths, with a polyphonic voice architecture, on-screen keyboard (including touch), and Web MIDI.

[TouchPad \(a,t\)](#)

Pointer/multi-touch input to control Web Audio. Load it on a touch device, touch and slide around - to open/close the filter and modify Q.

[Volume Meter \(i\)](#)

This is a simple code example of how to properly implement a clip-detecting volume meter.

[Samplr \(a,m\)](#)

This is an in-progress example of how to implement a sample-playback synthesizer. Eventually should support looping and voice control, but simple code example of how to properly implement a clip-detecting volume meter.

[Slide Deck](#)

Slide deck on Web Audio and Web MIDI.

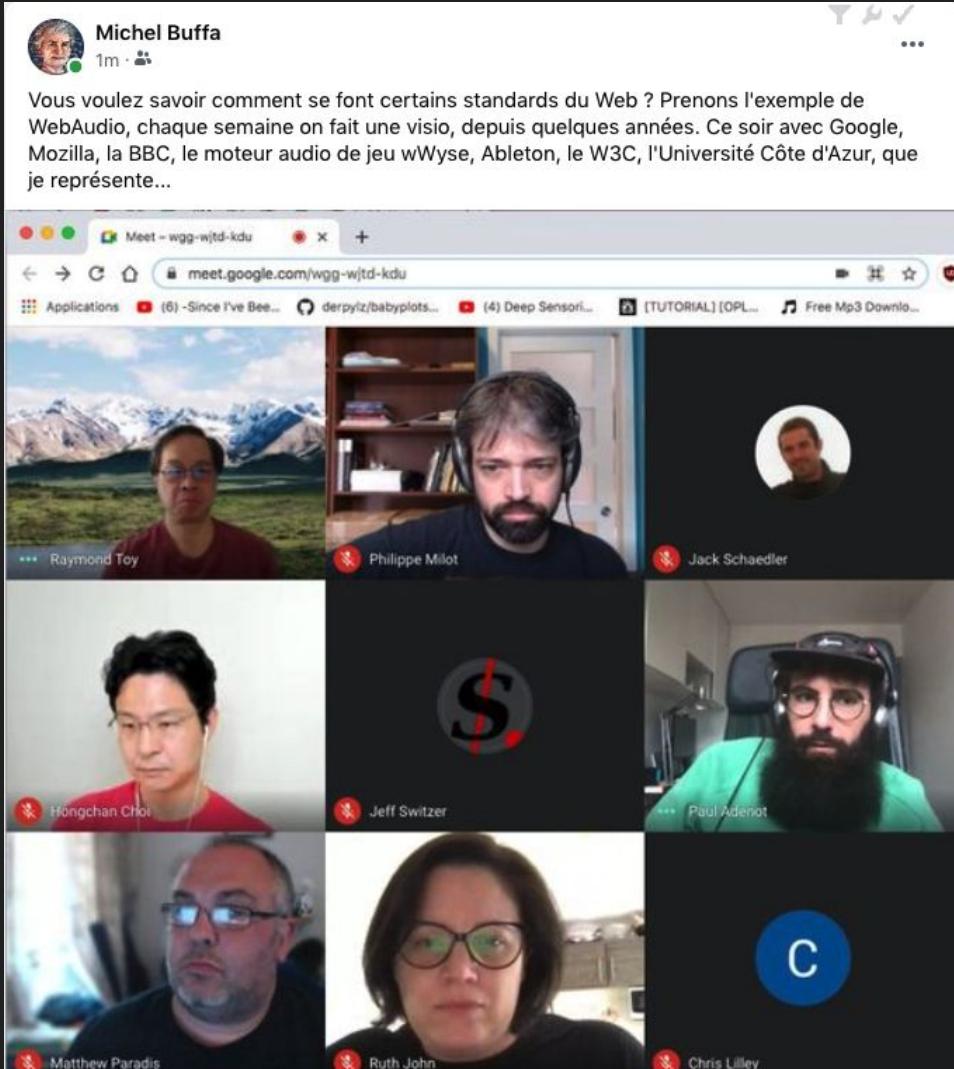
Creation of a W3C Web Audio Working Group

Meeting every 1-2 weeks

In person meeting 2-3 times a year

(right picture from 2021)

The screenshot shows a Bugzilla search results page with the URL <https://bugzilla.mozilla.org/cgi?id=webaudio>. The search term 'webaudio' is entered in the search bar. One result is displayed: Bug 779297 (webaudio), which was opened 10 years ago and closed 9 years ago. The summary is 'Implement and ship Web Audio'. The bug details show it's a defect of the Core component in the Web Audio product. Status: RESOLVED FIXED, Milestone: mozilla25, Tracking Flags: retnote-firefox, and Status: 25+. The tracking section indicates 25+ users are tracking this bug.



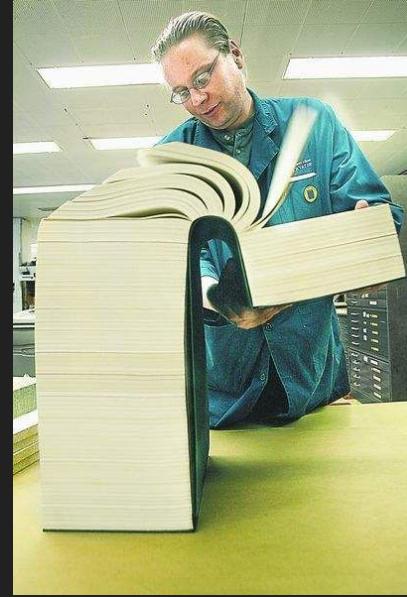
Some milestones

15/12/2011: first draft specification of the [WebAudio API](#)

- Lot of work to make sure it can be implemented and sound the same on all browsers (algorithms, etc.)

2012:

- Implementations in Firefox/Webkit browsers
- Support for real time input: [Media Capture and Streams API](#)
- First draft of the [WebMidi API](#), implementation in Google Chrome
 - (2022 in Firefox too!)



But... porting
existing DSP code
was not easy...

The WebAudio API was
incomplete!

Need for something better!

Monster Madness – creating games on the web with Emscripten



By [Jeremy Stieglitz, Robert Nyman \[Editor emeritus\]](#)

Posted on [December 12, 2013](#), [asm.js](#), [Audio](#), [Games](#), [IndexedDB](#), and [WebRTC](#)

When our engineering teams at Trendy Entertainment & Nom Nom Games decided on the strategy of developing one of our new Unreal Engine 3 games — [Monster Madness Online](#) — as a cross-platform title, we knew that a frictionless multiplayer web browser version would be central to this experience. The big question, however, was determining what essential technologies to utilize in order to bring our game onto the web. As a C++ oriented developer, we determined quickly that rewriting the game engine from the ground-up was out of the question. We'd need a solution that would allow us to port our existing code in an efficient manner into a format usable in the browser...

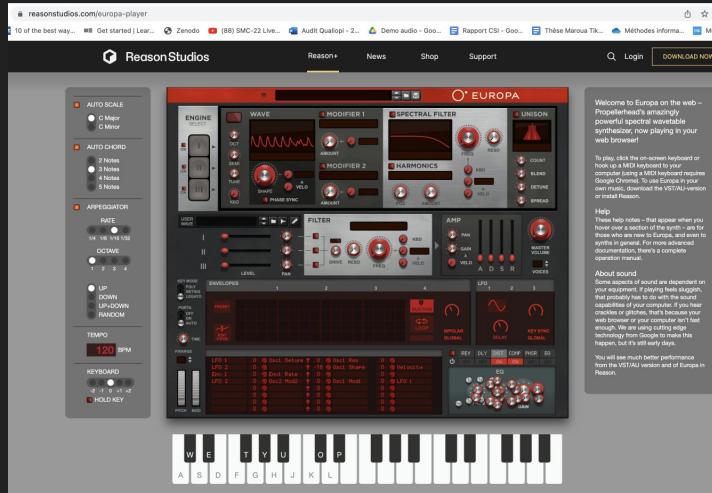
TL;DR? Watch the video!

**IT DID NOT WORK
WELL !!!!!**



Some milestones

2018-2022: AudioWorklet : custom DSP processing in the Audio Thread



2012 : WebSockets (synchronous collaboration)

2015: Web Components

2018: WebAssembly

Recent history...

**2021: the WebAudio API V 1.0 became
a W3C Recommendation in June 2021
(aka a “frozen standard”)**

**2021-today : ongoing development of
Version 2.0**

- The working group took care of github issues/proposals
- Conducted a developers' survey

**2022 : W3C proposes help for higher
level standards (i.e for a plugin standard)**



Some notable demos

Web Audio Demos

[Playground \(a,i\)](#)

The Web Audio Playground helps developers visualize how the graph nodes in the Web Audio API work. Shown at I/O 2012.

[Vocoder \(a,i,m\)](#)

This complex audio processing app (shown at I/O 2012) implements a 28-band [vocoder](#) - a "robotic voice" processor.

["Analog" Synth \(a,m\)](#)

This application implements a polyphonic "analog" synthesizer, with a classic voice architecture. Playable via Web MIDI or onscreen keyboard.

[Input Effects \(a,i\)](#)

This demo lets you play with a few common effects on the audio inputs.

[AudioRecorder \(a,i\)](#)

This is sample code for recording audio from live input and downloading them as WAV files, built on Matt Diamond's excellent [RecorderJS](#).

[Oscilloscope \(a\)](#)

Oscilloscope waveform display for Web Audio, and pulse-width duty cycle control on a square wave.

[Monosynth \(a,m\)](#)

Example of a monophonic Web MIDI/Web Audio synth, with no UI.

[wubwubwub \(a,m\)](#)

This application implements a dual DJ deck, specifically intended to be driven by a Numark MIDI controller.

[Pitch Detection \(i\)](#)

This application performs monophonic autocorrelating pitch detection in realtime. This is suitable for a guitar tuner or other complex waveform source.

[The Game of Life \(m\)](#)

Conway's Game of Life, on an 8x8 grid, interfacing with a Novation Launchpad controller via Web MIDI.

[Happy Drums \(a,m\)](#)

Quick addition of MIDI controller support (using a Livid Instruments CNTRLR) to the [Shiny Drum Machine](#).

[TemplateSynth \(a,m\)](#)

Simple template for Web Audio synths, with a polyphonic voice architecture, on-screen keyboard (including touch), and Web MIDI.

[TouchPad \(a,t\)](#)

Pointer/multi-touch input to control Web Audio. Load it on a touch device, touch and slide around - to open/close the filter and modify Q.

[Volume Meter \(i\)](#)

This is a simple code example of how to properly implement a clip-detecting volume meter.

[Samplr \(a,m\)](#)

This is an in-progress example of how to implement a sample-playback synthesizer. Eventually should support looping and voice control, but simple code example of how to properly implement a clip-detecting volume meter.

[Slide Deck](#)

Slide deck on Web Audio and Web MIDI.

2011-2015: Chris Wilson's demos!

2014: first implementation of ToneJS library

Tone.js v14.7.77

BASIC

- Oscillators
- Envelope
- Noise
- Player
- Microphone
- Mixer

INSTRUMENTS

- Synth
- MonoSynth
- FMSynth
- AMSynth
- PolySynth
- FatOscillator
- MetalSynth
- Granular Synthesis
- Sampler

EFFECTS

- LFO Effects
- PingPongDelay
- Buses
- Reverb
- Spatialization
- PitchShift

SEQUENCING / TIMING

- Step Sequencer

This example demonstrates the Pitch Shift effect.

[Tone.Pitch Shift docs.](#)

pitch -0.03 hz

Tone.js v14.7.77

BASIC

- Oscillators
- Envelope
- Noise
- Player
- Microphone
- Mixer

INSTRUMENTS

- Synth
- MonoSynth
- FMSynth
- AMSynth
- PolySynth
- FatOscillator
- MetalSynth
- Granular Synthesis
- Sampler

EFFECTS

- LFO Effects
- PingPongDelay
- Buses
- Reverb
- Spatialization
- PitchShift

This beat is composed of 3 independent Players each with a different loop length, synced to the Transport to start at different times and different offsets. The players stay synchronized to the position and offset of the Transport.

Used in TONS of projects!

2015: Web Audio debugger in Firefox

The screenshot shows the Firefox developer tools interface with the "Web Audio" tab selected. On the left, a node graph displays the audio signal flow. The graph consists of several nodes: OscillatorNode, BiquadFilterNode, GainNode, AudioDestinationNode, and DynamicsCompressorNode. The OscillatorNode has two outputs: one goes to the BiquadFilterNode, and another goes to the DynamicsCompressorNode. The BiquadFilterNode has one output that goes to the GainNode. The GainNode has one output that goes to the AudioDestinationNode. The DynamicsCompressorNode also has one output that goes to the AudioDestinationNode.

On the right, a detailed view of the BiquadFilterNode is shown. The node is identified as "BiquadFilterNode (conn0.audionode161)". The "Parameters" section is highlighted in blue. It contains the following settings:

- type:** "highpass"
- frequency:** 1000
- Q:** 1
- detune:** 0
- gain:** 1

CATEGORY

abstract (4)

subtractive (2)

additive (0)

physical (0)

sample (0)

hybrid (0)

misc (0)

WAM SYNTHS

WebAssembly / AudioWorklet

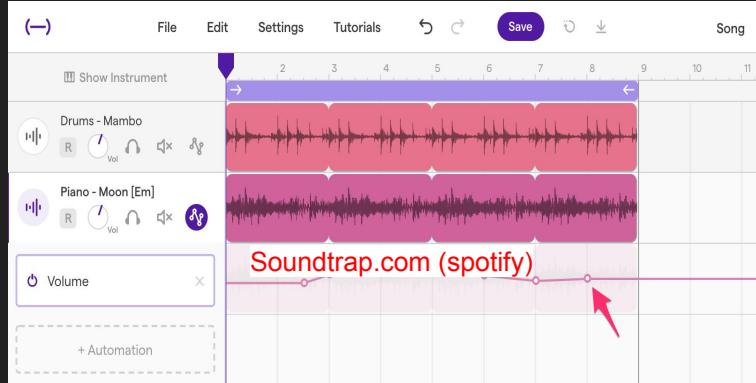


asm.js / ScriptProcessor



2015: Web Audio Modules (Kleimola & Larkin)

2015: first commercial DAWs



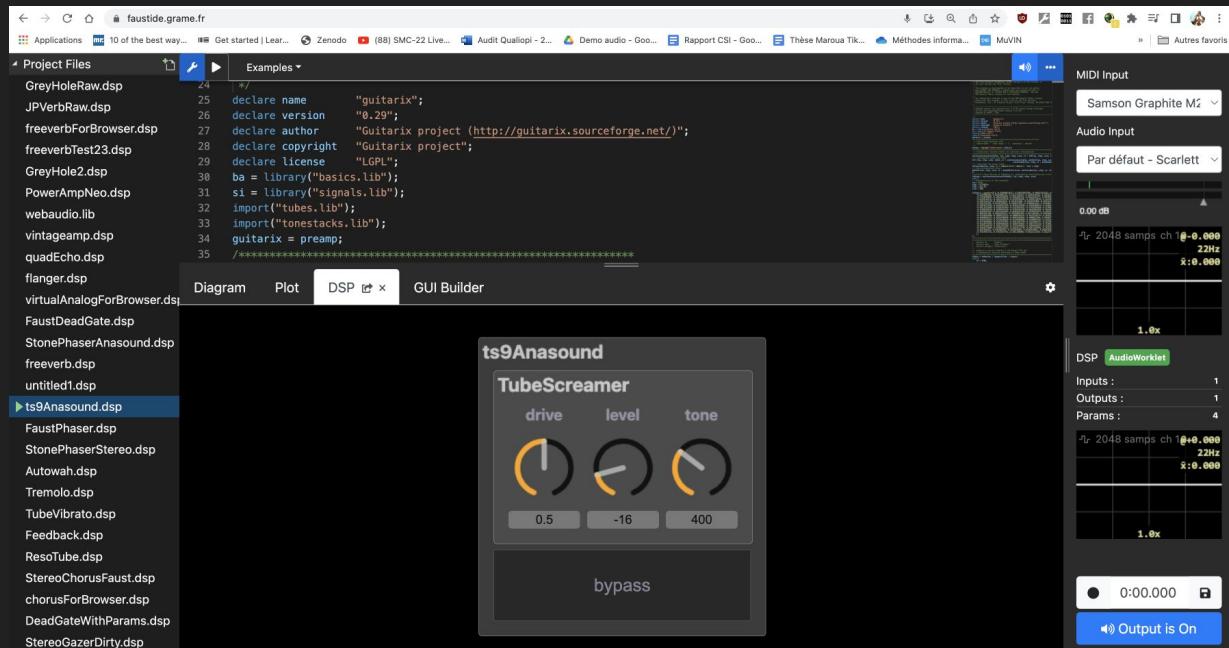
2016: why not some guitar?



2014/2015: DSLs compile to ASM.js and later to WebAssembly

FAUST, PureData,
Csound, others can be
run on the Web

2018 : FAUST online IDE



2018: Ableton comes in the game!

The screenshot shows a web browser window with the URL learningsynths.ableton.com/en/get-started. The page title is "Learning Synths — Get started". On the left, there's a vertical sidebar with a "Chapters" section. The main content area features a large, bold heading "Get started making sounds". Below it, a text instruction says "Drag in the box below to play a *synthesizer*." A pink callout box contains the note: "If you don't hear anything, make sure your sound is on and the volume is turned up. And please use headphones for the best experience!" At the bottom of the interaction box, the word "DRAG" is written in a small pink box.

Learning Synths — Get started

Chapters

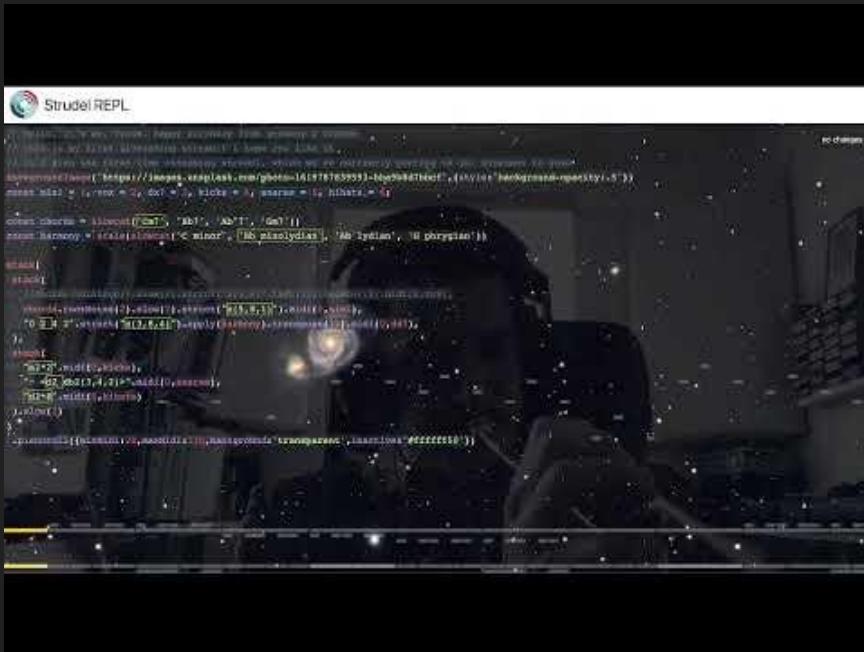
Get started
making sounds

Drag in the box below to play a *synthesizer*.

If you don't hear anything, make sure your sound is on and the volume is turned up. And please use headphones for the best experience!

DRAG

2018: the live coding community shines



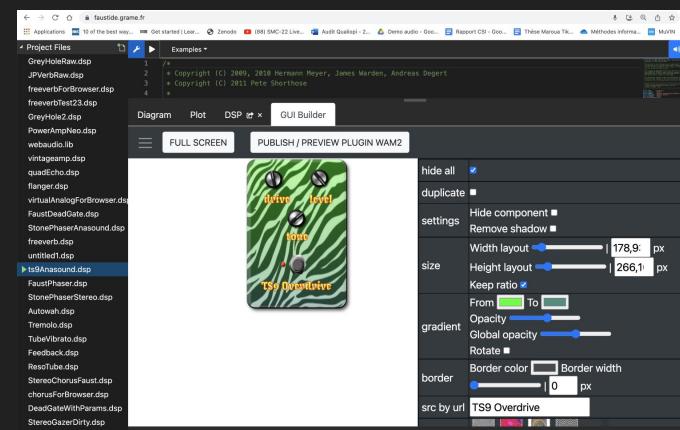
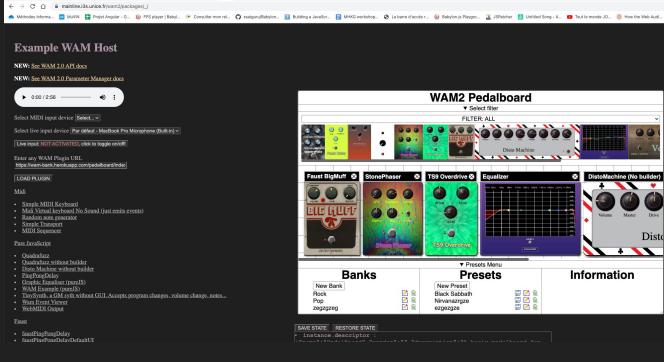
Strudel in 2022



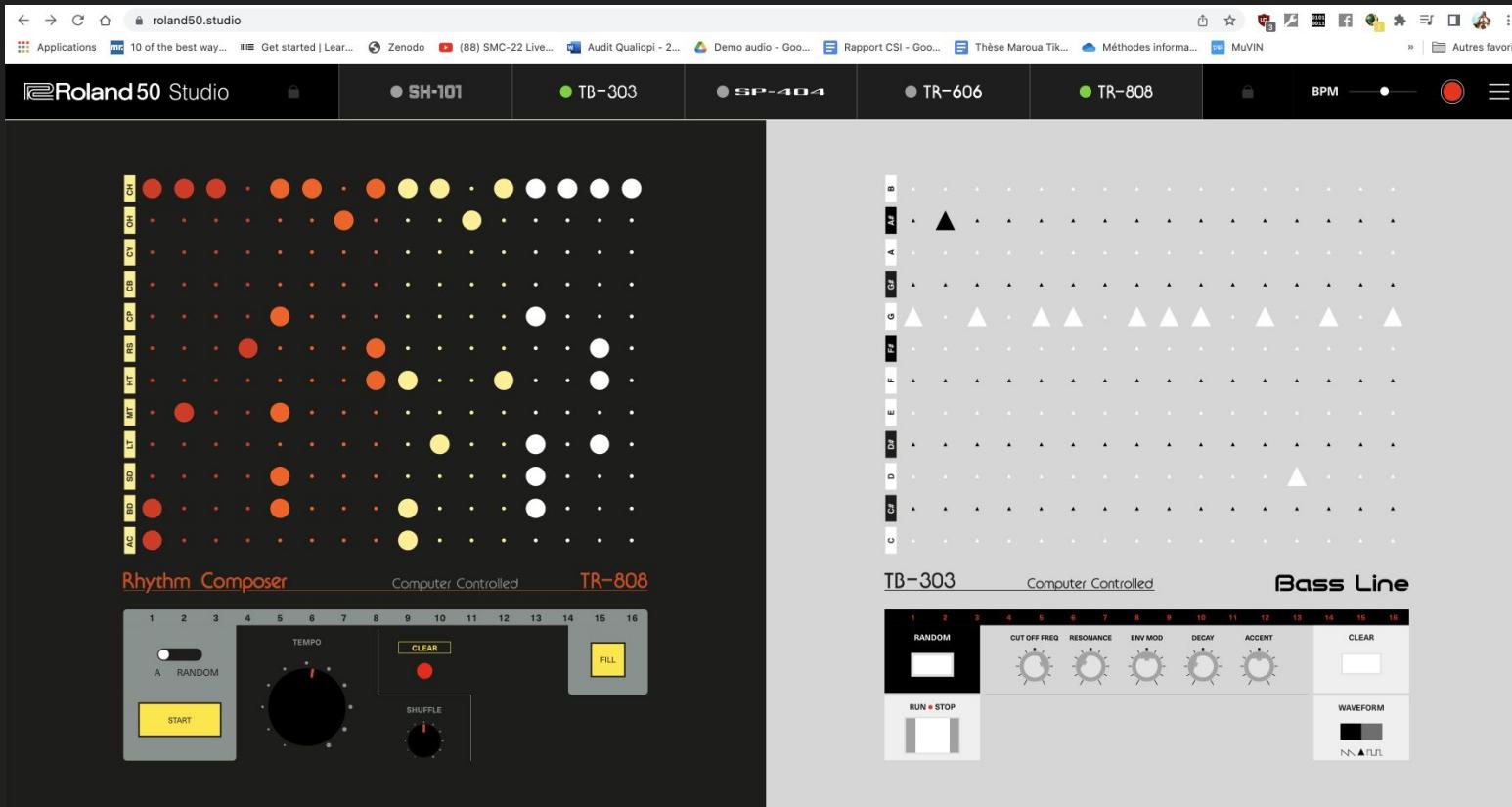
Strudel (Roos, Mclean), Gibber, (Roberts and Joann), Estuary (Ogborn and Beverley), Hydra, Wags, Feedforward, etc.

Some projects started in 2012...

2022: Web Audio plugins become mature (WAM)



2022: Roland comes in the game!



2022: Microtonal synthesizer (Khyam Allami / counterpoint)

LEIMMA: explore microtonial tuning systems

APOTOME: generative music composition tool, which lets you use all the tunings from LEIMMA

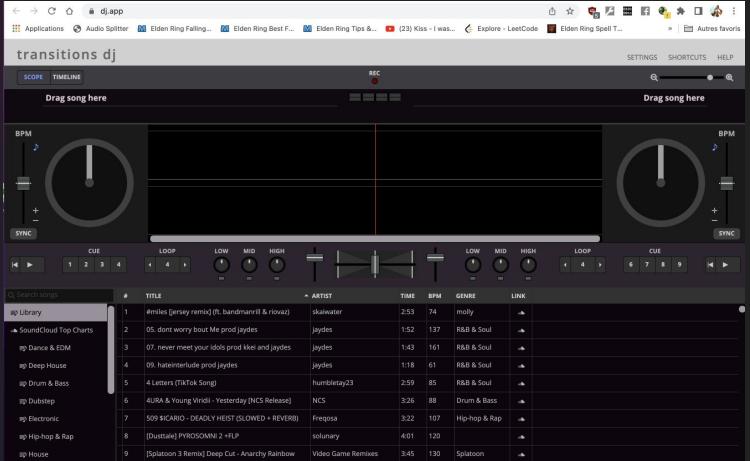
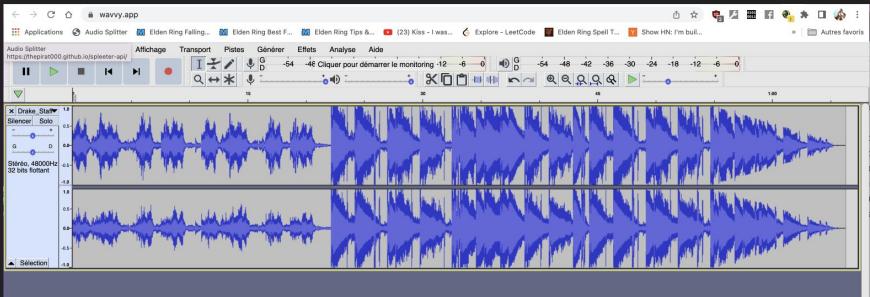
The screenshot shows the Counterpoint software interface. At the top, there's a black circular profile picture and the text "Counterpoint" followed by "@cpt_studio". Below this is a large text block: "We've just launched a family of projects that explore the world of microtonal tuning systems, developed in collaboration with [@KhyamAllami](#) over the past two years". Underneath this text is a blue link "Traduire le Tweet". The main area of the interface is a grid of five tracks labeled TRACK 2 through TRACK 6. Each track has sections for "TUNING & SUBSET", "SUBSET ROLE WEIGHTS", "ALLOWED MELODIC INTERVALS", and "OCTAVES WEIGHTS". There are also buttons for "Override Time Signature", "Override Accents", and "Exclusive Beat Division". To the right of the tracks, there's a sidebar titled "Book your performance for CTM Festival 2021" with a photo of Khyam Allami. The sidebar includes sections for "SNAPSHOT 1" (Name: Drone E (127) + Max + Melody, Duration: 12), "SNAPSHOT 2" (Name: Max + Melody, Duration: 12), "SNAPSHOT 3" (Name: Max + Melody, Duration: 12), and "SNAPSHOT 4" (Name: Max + Melody, Duration: 12). At the bottom right, there's a timestamp "23:03" and some other controls.

2022: Audacity (the real one) on the Web thanks to WASM

Port of audacity on the Web!

- **WASM compiled**

Also: WASM port of transitionDJ app



CONCLUSION

The Web platform is Magic :-)

Web Audio is a technology but it also became a new, multidisciplinary, research field



Web Audio lives along with all other APIs, and goes beyond the browser now...

WebML, WebCodecs, WebAudio API standalone in embedded systems, etc.