



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

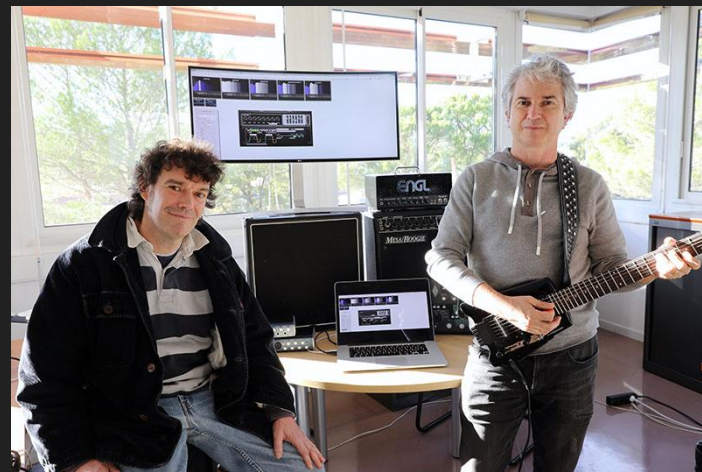
11 June 2022

Michel Buffa,
Université Côte d'Azur, France,
I3S/CNRS/INRIA labs
buffa@i3s.unice.fr, @micbuffa



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](#)
- [Buisness WWW sites](#)

← → ↻ ⚙ Non sécurisé | users.polytech.unice.fr/~buffa/#courses

Applications 10 of the best way... Get started | Lear... Zenodo (88) SMC-22 Live... A

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...](#)

Created in 1993, last updated:1996, HTML 1

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!



0

Sign
in to
vote

I assume you are using the bgsound tag.

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.



http://www.jw.org/download/?output=html



jw.org



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



dreamstime.com

ID 154897294 © Vasilyrosca

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

INTERNET ARCHIVE
waybackmachine

148 captures
6 May 2010 – 6 Jun 2022

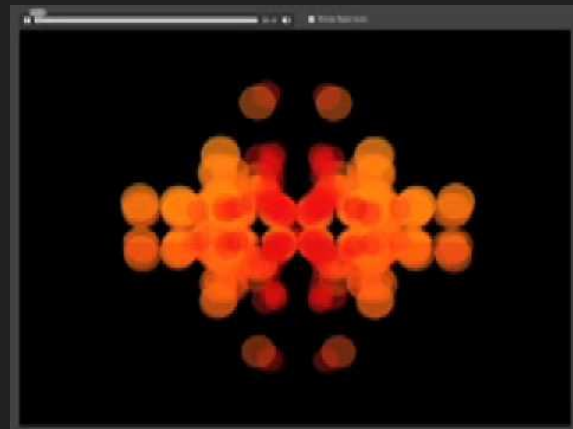
Go MAY JUL SEP
2009 06 2010 2011

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>
- 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-firefox-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 W

Proposed b

WebAudioDesigner | g200kg

https://g200kg.github.io/WebAudioDesigner/?b=W3tuOiJkZXN0aW5hdGlvbilseDo2OTQseToxMT

WebAudio Designer <https://github.com/g200kg/webaudiodesigner>

StopGraphAdd NodeAdd ControllerAbout

The screenshot shows the WebAudio Designer web application. The interface has a top navigation bar with buttons: Stop, Graph, Add Node, Add Controller, and About. The main workspace displays a signal flow graph on a dark blue background. The graph consists of several nodes: a buffer source node (bufsrc1) with parameters like playbackRate (1), loop (true), loopStart (0), loopEnd (0), and buffer (loop.wav); a delay node (delay1) with a delayTime of 0.25; three gain nodes (gain1, gain2, gain3) with gain values of 1, 0.5, and 0.5 respectively; and a destination node (destination). Green lines represent the signal flow connections between these nodes.

lists.w3.org/Archiv
Applications 10 of the best way...

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

Web Audio API

This message: [M]
Related messages:

From: Chris Rogers <cro
Date: Mon, 14 Jun 2010
Message-ID: <AANLkTik69Rr
To: public-xg-audio@w3

Hi Everyone,

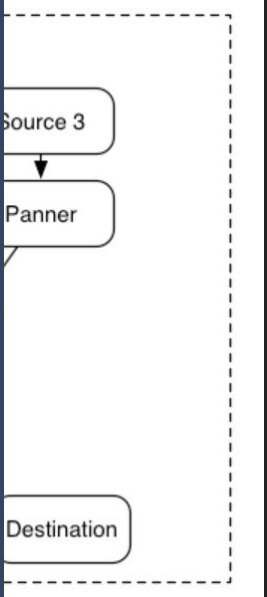
Since it looks like th
this opportunity to in
developing over the pa
at Google as part of t
at Apple as an enginee

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

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

<http://chromium.google>

DSP part



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.

[Sampler](#) (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)

cg?id=webaudio

Learn... Zenodo (88) SMC-22 Live... Audit Qualiopi - 2... Demo audio - Goo... Rapport CSI - Goo... Thèse Maroua Tik... Méthodes Informa...

Bugzilla Search Bugs Browse Advanced Search >> New Account Log In Forgot Password

Copy Summary View

Closed Bug 779297 (webaudio) Opened 10 years ago Closed 9 years ago

Implement and ship Web Audio

Categories

Product: Core ▼ Type: defect
Component: Web Audio ▼ Priority: Not set Severity: normal

Tracking

Status: RESOLVED FIXED Tracking Flags: relnote-firefox Tracking Status: --- 25+
Milestone: mozilla25

Michel Buffa 1m · 🧑

Vous voulez savoir comment se font certains standards du Web ? Prenons l'exemple de WebAudio, chaque semaine on fait une visio, depuis quelques années. Ce soir avec Google, Mozilla, la BBC, le moteur audio de jeu wWyse, Ableton, le W3C, l'Université Côte d'Azur, que je représente...

Meet - wgg-wjtd-kdu

meet.google.com/wgg-wjtd-kdu

Applications (6) - Since I've Bee... derpyiz/babyplots... (4) Deep Sensori... [TUTORIAL] [OPL... Free Mp3 Downlo...

Raymond Toy Philippe Milot Jack Schaedler

Hongchan Choi Jeff Switzer Paul Adenot

Matthew Paradis Ruth John Chris Lilley

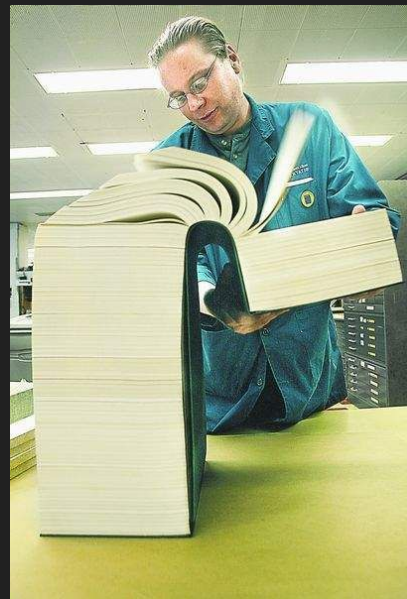
Some milestones

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...

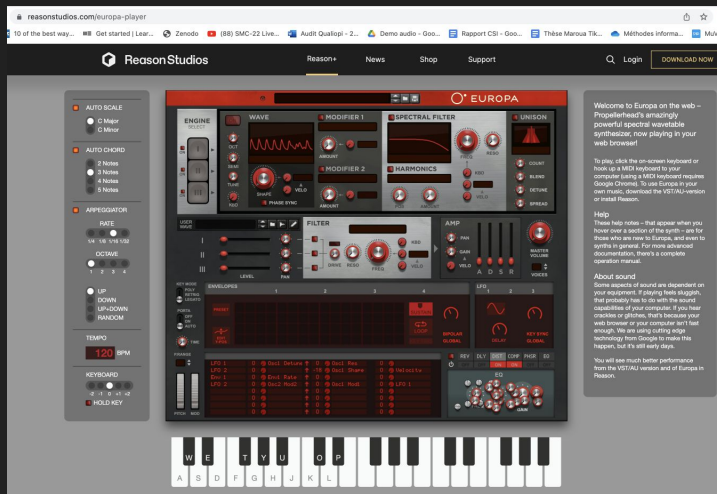
TL;DR? Watch the video!

**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

The screenshot shows the Tone.js v14.7.77 web interface for the pitchShift example. The left sidebar contains a menu with categories: BASIC, INSTRUMENTS, EFFECTS, and SEQUENCING / TIMING. Under BASIC, there are links for Oscillators, Envelope, Noise, Player, Microphone, and Mixer. Under INSTRUMENTS, there are links for Synth, MonoSynth, FMSynth, AMSynth, PolySynth, FatOscillator, MetalSynth, Granular Synthesis, and Sampler. Under EFFECTS, there are links for LFO Effects, PingPongDelay, Buses, Reverb, Spatialization, and PitchShift. The main content area displays the text "This example demonstrates the Pitch Shift effect." followed by a link to "Tone.Pitch Shift docs." Below this is a "pitch" slider set to -0.03 hz and a waveform visualization.

The screenshot shows the Tone.js v14.7.77 web interface for the daw example. The left sidebar contains a menu with categories: BASIC, INSTRUMENTS, and EFFECTS. Under BASIC, there are links for Oscillators, Envelope, Noise, Player, Microphone, and Mixer. Under INSTRUMENTS, there are links for Synth, MonoSynth, FMSynth, AMSynth, PolySynth, FatOscillator, MetalSynth, Granular Synthesis, and Sampler. Under EFFECTS, there are links for LFO Effects, PingPongDelay, Buses, Reverb, Spatialization, and PitchShift. The main content area displays the text "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." Below this is a visual representation of a beat with three independent players, each with a different loop length, synced to the Transport.

Used in TONS of projects!

2015: Web Audio debugger in Firefox

The screenshot displays the Firefox Web Audio debugger interface. The top toolbar includes tabs for Inspector, Console, Debugger, Style Editor, and Web Audio (which is currently selected). The main area on the left shows an audio graph with the following nodes and connections:

- OscillatorNode** connects to **BiquadFilterNode** and **DynamicsCompressorNode**.
- AudioBufferSourceNode** connects to **DynamicsCompressorNode**.
- BiquadFilterNode** connects to **GainNode**.
- GainNode** connects to **AudioDestinationNode**.
- DynamicsCompressorNode** connects to **AudioDestinationNode**.

The **BiquadFilterNode** is highlighted in blue. On the right, the **Parameters** panel for **BiquadFilterNode (conn0.audionode161)** is shown, displaying the following settings:

- AudioParams** (expanded)
- 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

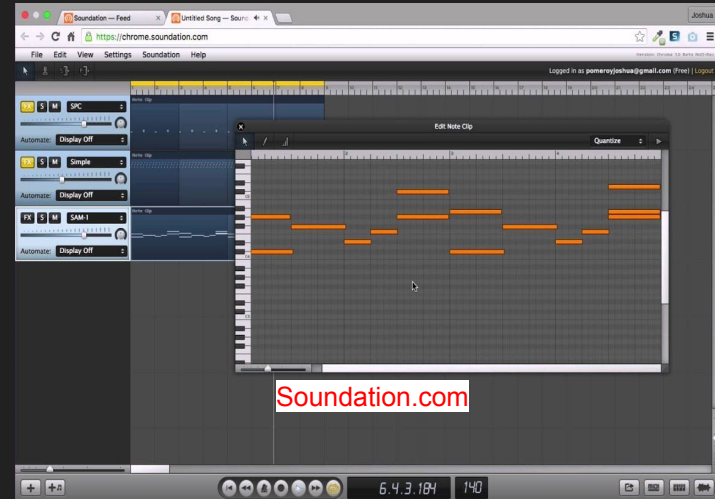
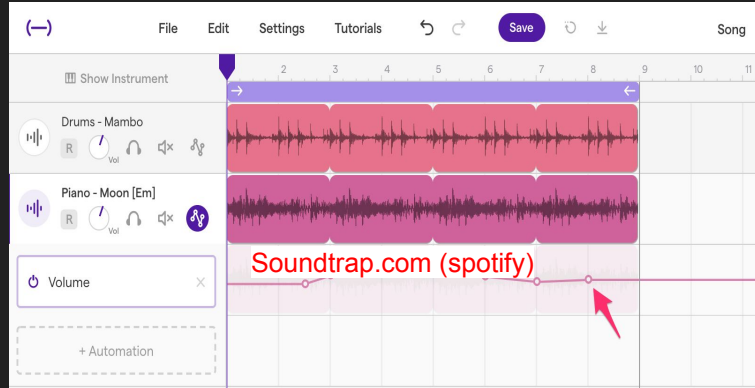


asm.js / ScriptProcessor



2015: Web Audio Modules (Kleimola & Larkin)

2015: first commercial DAWs



2016: why not some guitar?



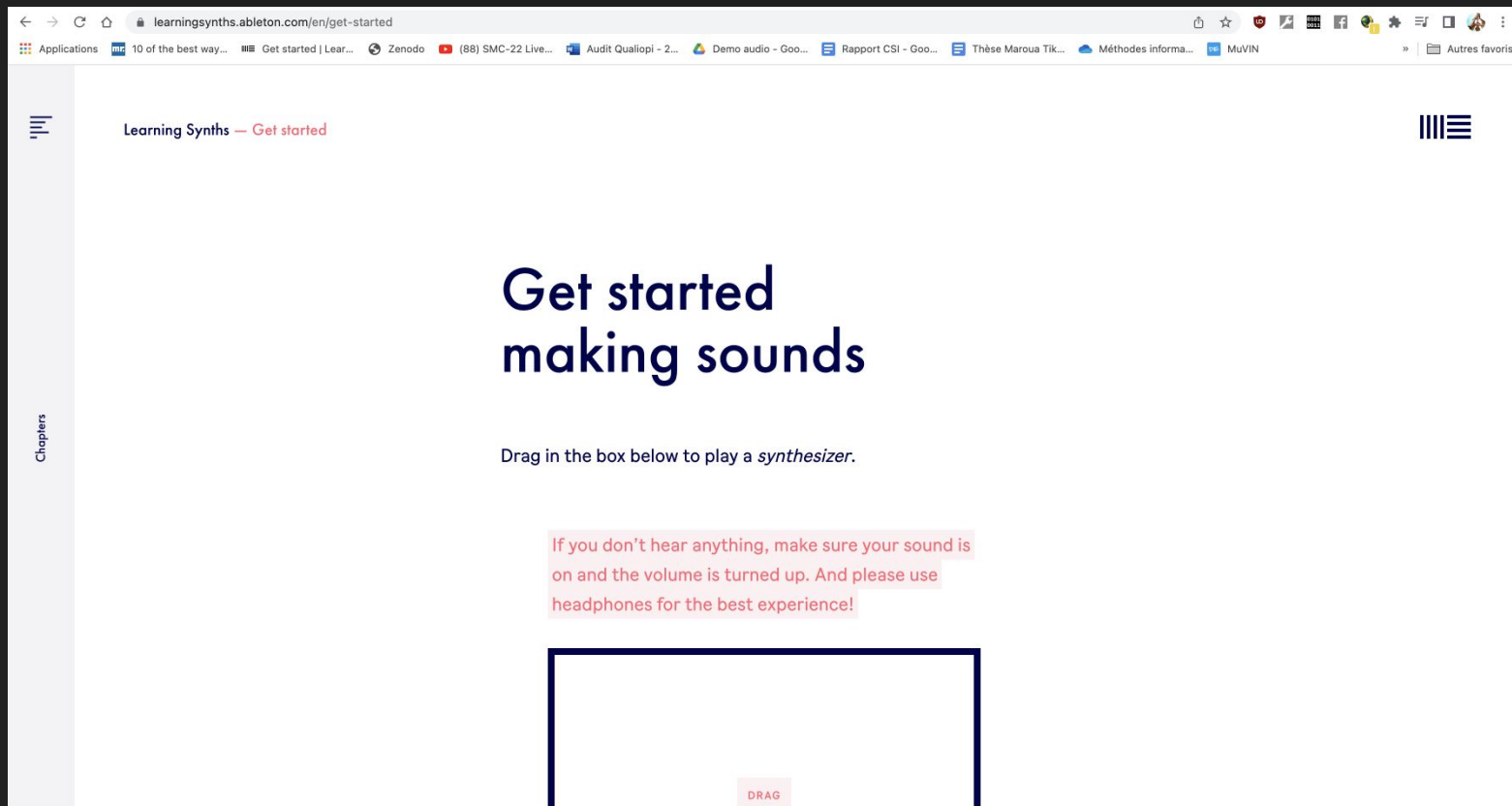
2014/2015: DSLs compile to ASM.js and later to WebAssembly

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

2018 : FAUST online IDE

The screenshot displays the Faust online IDE interface. On the left, a 'Project Files' sidebar lists various DSP modules, with 'ts9Anasound.dsp' selected. The main area is a code editor showing the source code for 'ts9Anasound.dsp', which includes declarations for 'guitarix' and imports for 'basics.lib', 'signals.lib', and 'tonesacks.lib'. Below the code editor, a 'GUI Builder' tab is active, showing a visual representation of the 'ts9Anasound' module. This GUI includes three knobs for 'drive', 'level', and 'tone', each with a corresponding numerical value (0.5, -16, and 400 respectively). A 'bypass' button is also visible. On the right side of the interface, there are controls for MIDI input (Samson Graphite M2), audio input (Par défaut - Scarlett), and a DSP section with 'AudioWorklet' and 'DSP' settings. The bottom right corner shows a timer at 0:00.000 and a button labeled 'Output is On'.

2018: Ableton comes in the game!



2018: the live coding community shines



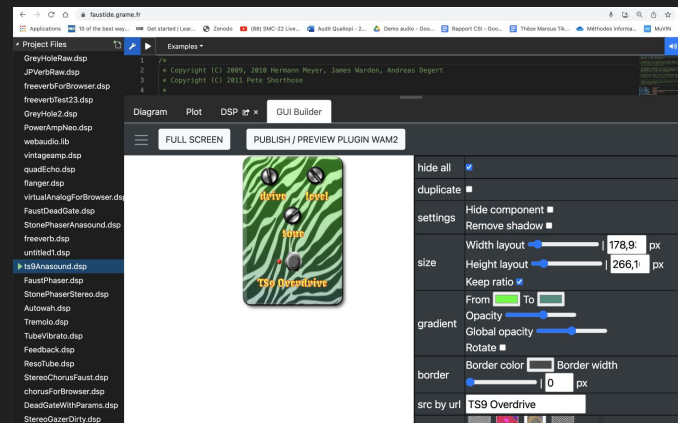
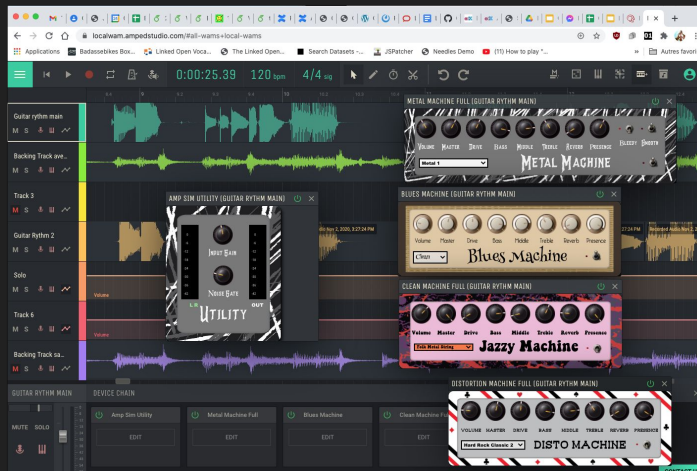
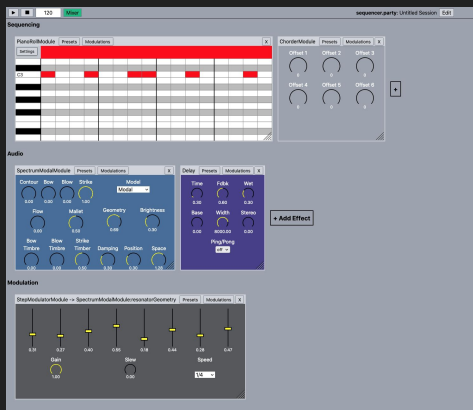
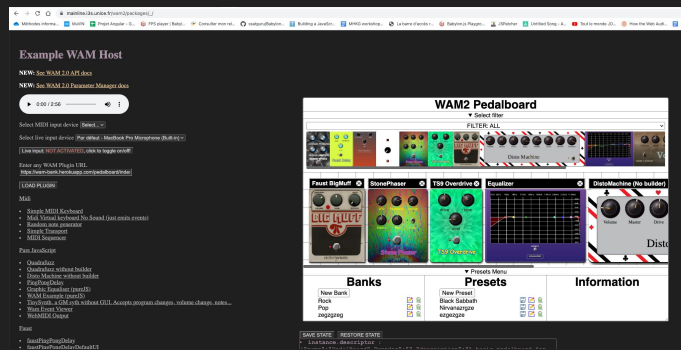
Strudel in 2022



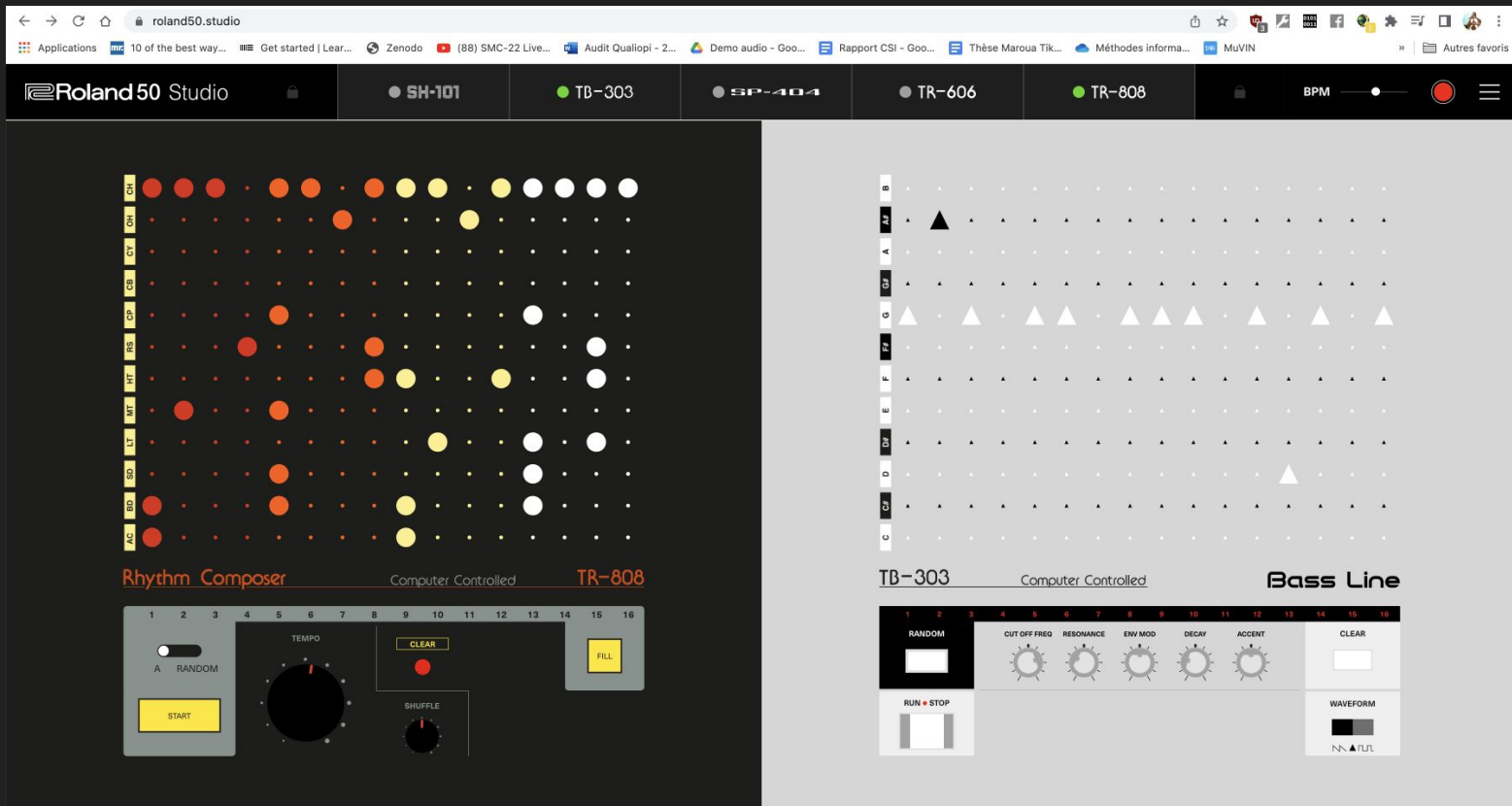
Strudel (Roos, Mclean), Gibber, (Roberts and Joann), Estuary (Ogborn and Beverley), Hydra, Wags, Feedforward, etc.

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 microtonal tuning systems

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

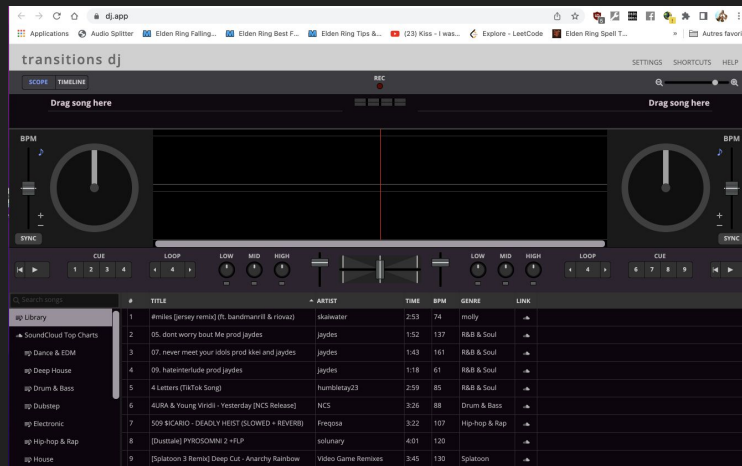
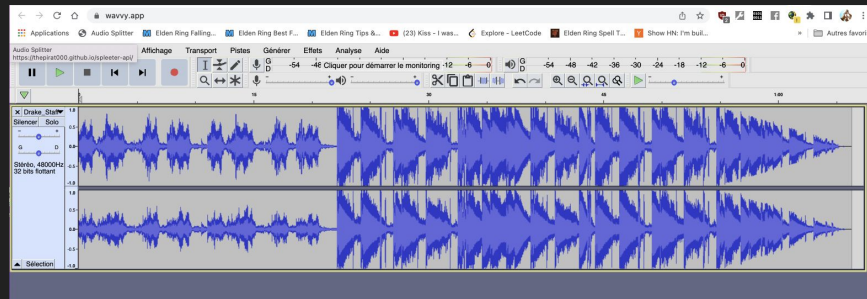


2022: Audacity (the real one) on the Web thanks to WASM

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.