OPEN ACCESS Freely available online

A. Stolfi and C. Figueiró,
"Audio on Web: From Sharing to Production,"
J. Audio Eng. Soc., vol. 73, no. 12, pp. 837–848 (2025 Dec.).
https://doi.org/10.17743/jaes.2022.0251.

Audio on Web: From Sharing to Production

ARIANE STOLFI,[1]* AND CRISTIANO FIGUEIRÓ[2]
(arianestolfi@ufsb.edu.br) (cristianofigueiro@ufba.br)

[1]*Centro de Formação em Artes e Comunicação, Universidade Federal do Sul da Bahia, Porto Seguro, Bahia, Brazil*
[2]*Instituto de Humanidades, Artes e Ciências, Universidade Federal da Bahia, Salvador, Bahia, Brazil*

The rapid evolution of web technologies has fundamentally transformed the creation, distribution, and experience of audio and music. This paper explores the historical trajectory and current state of audio on the web, from early peer-to-peer file sharing (e.g., Napster) to modern browser-based tools that democratize music production and performance. Key technological milestones are examined, including the transition from proprietary plug-ins (Flash) to open standards (Hypertext Markup Language 5, Web Audio API), and the rise of JavaScript libraries (Tone.js, Magenta.js) that collaborate to sophisticated audio processing and generative music directly in the browser. The paper traces a brief history of audio on the web, from early music distribution systems to modern browser-based production environments. It highlights emerging paradigms, such as live coding, collaborative digital audio workstations and artificial intelligence–driven composition, which redefine creativity as a collective, networked practice. By analyzing these developments, the dual role of the web as both a technical infrastructure and a cultural platform is revealed, fostering new forms of artistic expression while challenging traditional boundaries of ownership, accessibility, and interaction. Finally, the authors discuss ongoing tensions between open-source ecosystems and proprietary solutions and speculate on future directions for web-based audio, including immersive environments and decentralized collaboration.

0 INTRODUCTION

Audio technologies have been intensively transformed during the last century, from the ability to record and reproduce sounds to the digitization of sound and its transmission through the web. The sound energy had an ephemeral characteristic since the beginning of time, but the first recording technologies could allow its crystallization into material objects that could carry it beyond the present time. For a while, music could be sold in the form of tangible objects: vinyl discs, tapes, and compact discs (CDs). Then, the digital technologies that followed transformed it again into volatile objects.

Trading objects always involves changing the ownership of something. To buy an item, someone must lose some money, but with digital information, it is possible to multiply things without losing the ownership of them. With the first digital audio format, which was first developed to write music on CDs, together with the popularization of personal computers at the end of the last century, it also started a culture of domestic multiplying of sound files.

Alongside this process of digitization of information and physical media, the digitalization of communication mediums was also taking place. Marshall McLuhan [1], in his important book *Understanding Media*, points to the fact that societies change when the communication media change. With this new media, the digital media, there is also the rise of what the philosopher Daniel Bell [2] called a "post-industrial society," where information has become one of the most important strategic resources, as capital and labor had been for the industrial society before. In this context, the communication medium becomes the biggest bounding structure to unify and amplify the arena where the social interaction happens.

In that context, computers were developing not only as tools but also as a communication medium. First developed for military purposes as the Advanced Research Projects Agency Network, a large network of computers called the Internet started to connect computers worldwide with the Transmission Control Protocol/Internet Protocol (IP), which made computers accessible through IP addresses. In 1989, this network gained another dimension with the advent of the World Wide Web by Tim Berners Lee at the European Organization for Nuclear Research Center [3].

With the advent of the Hypertext Markup Language (HTML), the computer browser, and a service to connect

*To whom correspondence should be addressed, e-mail: arianestolfi@ufsb.edu.br.

computers by addresses, the World Wide Web was an important step to organize Internet data and to transform it into the most important mass communication media. In the first site example, there was already an audio file that cannot be found any more.[1]

In 1994, Tim Berners Lee founded the World Wide Web Consortium (W3C) in collaboration with international organizations with the mission "to lead the World Wide Web to its full potential by developing protocols and guidelines that ensure the long-term growth of the Web" [4]. Since then, the W3C has been responsible for developing specifications and recommendations for HTML and other languages and formats to give support for the web, such as Cascading Style Sheets (CSS), JavaScript, JavaScript Object Notation, Scalable Vector Graphics, Portable Network Graphics, and others.

As a new communication medium, the Internet can be considered a "medium of media." It started as a primarily text-based information exchange system and gradually became a platform that supports various other communication media, such as mail, telephone, radio, and cinema. It can be seen as a metamedium capable of supporting content from all previous media [5]. For the first time, it also provided a space in which users could create content without interference from the communication industries that previously dominated cultural production.

Everything that was digital could be shared over the Internet; however, the main limitation was the file sizes, because the speed at which information could travel was determined by the capacity of the network. This constraint led industries to develop new formats to compress data and transmit it more efficiently over wired networks.

By the end of the 1990s, commercial modems that allowed dial-up access to the Internet from personal computers could reach speeds of up to 56 kbps. This was sufficient for text exchange but not for transferring large files, such as Waveform Audio files from CDs. This began to change with the introduction of the Moving Picture Experts Group (MPEG) format, an open standard for digital audio and video developed in the late 1980s, which enabled audio compression based on psychoacoustic models[6]. Whereas a standard CD could hold approximately 74 minutes of uncompressed audio, the same CD could store more than nine hours of MPEG-1 Audio Layer 3–encoded files, depending on the compression quality.

1 AUDIO ON THE INTERNET: PEER-TO-PEER NETWORKS

Although the World Wide Web was invented in 1989, it took a few years before it became accessible to the general public. In its early years, due to the limitations of the first version of the HTML language, websites had a rather basic appearance. There were no search engines, and web navigation relied primarily on web rings, groups "of websites that are linked together in a continuous linear loop" [7].

During that era, most web pages were coded by hand, with individuals writing their own HTML files and providing static content. Fans produced a significant amount of music-related content, such as pages with pictures and lyrics. The web relies on websites, which implies ownership of a web server, domain, and IP addresses. As a result, hosting illegal content, such as third-party MPEG-1 Audio Layer 3 files, can be easily traced by the owners of its digital rights. However, it was the advent of peer-to-peer (P2P) technologies, such as Napster, that revolutionized the presence of music on the Internet.

P2P networks allowed users to connect directly with each other and exchange private files between their computers without having to upload them to any personal web space. Napster's servers merely provided an index of users' computers and did not host any illegal content themselves. Users contributed by "uploading or ripping songs from their CDs, (...) tapes, and vinyl records" [8].

Other systems, such as Audiogalaxy and Kazaa, also emerged, making music a significant portion of the data transferred over the Internet. Along with these platforms, new forms of music discovery have developed. When searching for something in Napster, users could also browse other users' files to discover new artists, songs, and genres. Audiogalaxy featured a taxonomy-based search engine that allowed users to browse genres and find related artists from an audio file [9].

This allowed people to access a vast collection of sound files and facilitated the circulation of artists' work without the mediation of large corporations such as radio and TV stations. Although record companies considered this circulation piracy and deemed it illegal, Honigsberg [8] points out that record sales increased during the Napster era and declined after its shutdown. However, he also notes that the industry sought to maintain control over the business, rejecting Napster's proposals to provide compulsory licenses for distributed music.

In April 2000, Metallica, one of the biggest rock bands of the time, and Dr. Dre, the owner of a record label that produced some of the most popular rap music of the era, filed a lawsuit against Napster. They tracked down the usernames of people sharing their songs and compelled Napster to ban them. Soon after, other artists and record labels followed, eventually forcing Napster to shut down its services in 2001 after multiple lawsuits. Although other P2P networks and decentralized technologies emerged, such as Soulseek and torrent services (which still operate today), the web also began to change over time.

2 AUDIO ON THE WEB: WEB 2.0

In the first phase of the web, which some refer to as Web 1.0, most of its content and infrastructure were provided by small users and companies. There were already some companies exploring music-playing technologies on the web using, such as Thomas Dolby's "Beatnik" that used Rich Molecular Format, a music file format that could use

[1] A missing sound example can be found at the 'What is Hypertext' link on the first webpage at https://info.cern.ch/hypertext/WWW/WhatIs.html.

MIDI and Waveform Audio files [10] based on Java Virtual Machines, but all of them were based on third party technologies and plug-ins.

However, as the web grew rapidly, investors rushed to find business opportunities that could fit within this new medium. As Dimitry Kleiner and Brian Wyrick point out, "money was thrown around quite randomly as investors struggled to understand what this medium would actually be used for" [11].

Soon, this independent, peer-based system began to be replaced by platforms controlled by big technology companies, as Wyrick notes:

> "Capitalism, rooted in the idea of earning income by way of idle share ownership, requires centralized control, without which peer producers have no reason to share their income with outside shareholders. Capitalism, therefore, is incompatible with free P2P networks, and thus, so long as the financing of Internet development comes from private shareholders looking to capture value by owning Internet resources, the network will only become more restricted and centralized" [11].

The so-called Web 2.0 was built on backend programming languages such as Personal Home Page and ASP.NET and with databases that allowed pages to be dynamically generated based on user interactions. This new era allowed people with no coding experience to create their own web pages using services such as social networks. These platforms became a primary means for people, especially youth, to stay in ongoing social contact.

One of the first social networks was myspace.com, launched in 2003, which became the most visited website in the United States in 2006 [12]. In addition to allowing users to write posts and share images, it also provided a platform where users could upload their own songs to be played directly in the browser. This created a network of independent producers and served as a showcase for new artists to present their work to the public. MySpace also supported user-customized pages, allowing each user to create a unique visual identity, as shown in Fig. 1.

It is important to mention that although MySpace is still online, the former personal pages are no longer accessible, nor are the more than 50 million songs uploaded by users before 2016. These songs were permanently lost due to a faulty server migration in 2019 [13].

Following the decline of MySpace, other social networks emerged, such as Orkut and later Facebook, but these were not focused on sound files like MySpace was. Other services more specifically focused on distribution of sound files, such as Bandcamp[2] and SoundCloud[3], were created and are currently active and provide space for artists who want to self-publish their music.

During the 1990s, the concept of "information superhighways" emerged as "high speed global data exchange networks," where people could "do business, be entertained

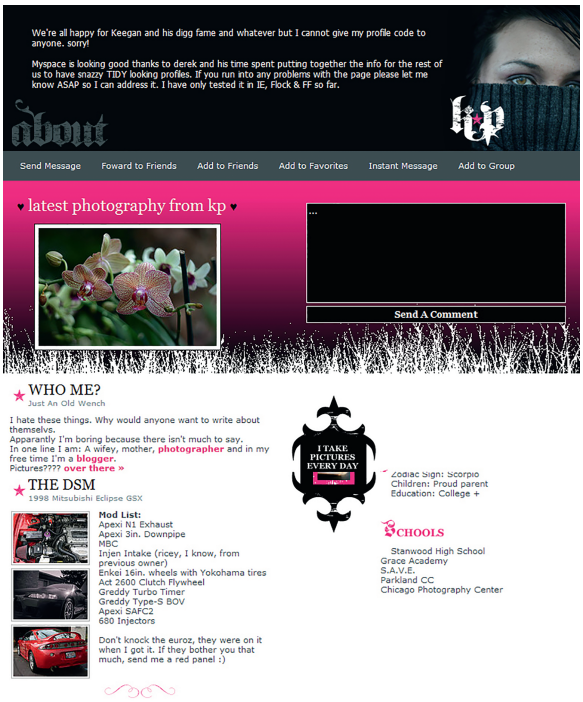


Fig. 1. Example of a customized MySpace music profile by Kristin Andrus, Creative Commons, 2006. Source: https://www.flickr.com/photos/wiphey/100693273/.

and meet friends" [14], leading to governments, organizations, and companies to invest in infrastructure to provide high-speed Internet to the world population, such as continental and underwater fiber-optic cables [15]. By the 2000s, this infrastructure was already available to a significant amount of the world consumers, leading to the streaming era that allowed people to watch media directly from the web for each individual user according to their own needs.

At the same time that this physical infrastructure was built, technologies for media streaming were also being developed, such as Shoutcast, launched in 1998, which allowed the creation of several web radios [16]. Some initiatives for collaborative music playing through the web were also developed at the time as high-end research projects, such as the SoundWIRE project from the Center for Computer Research in Music and Acoustics that "developed a set of tools for high-quality uncompressed bidirectional audio streaming" [17].

In 2002, Macromedia company launched Flash Player 6, which included a streaming server that was renamed to Flash Media Server in 2005. It was the technology behind YouTube, which was launched in the same year as a platform to broadcast user-generated content focused on videos [18]. Providing free space to host videos on their servers, YouTube began to be populated not only with original content produced by users, but also with ripped content such as TV shows, video clips, and music content, many of them converted into videos with static images. With this convenience, users started to prefer to listen to songs online

[2] https://bandcamp.com/
[3] https://soundcloud.com/discover

rather than downloading files that occupied space on their personal hard drives.

In a lawsuit against YouTube filed by Viacom, a U.S. media conglomerate that holds copyrights for audiovisual works [19], in 2007 it was argued that 75% to 80% of all YouTube streams contained copyrighted material, including music files uploaded as videos to the platform. In 2015, YouTube made commitments with several governments to combat piracy, taking steps to protect copyrights, including the development of technologies to identify content and ensure proper compensation for copyright holders. YouTube compensates content providers through advertising revenues and subscriptions from YouTube Premium users.

Although a significant portion of YouTube's content was provided by traditional media, a great deal of original content created by YouTube users also flourished on the platform. Several homegrown channels and independent music artists grew through the "viral" distribution of their content on the web, giving rise to "new modes of cultural production and new forms of enterprise" [18]. This marks the beginning of the streaming era, which changed the paradigm of the music industry again.

3 FROM FLASH TO HYPERTEXT MARKUP LANGUAGE 5

Although the web is built on open and free technologies and languages, such as HTML, CSS, and JavaScript, the first version of HTML supported only text and images. Audio and video players relied on proprietary technologies provided by companies such as Macromedia and Java. YouTube's first video player, for example, was based on Macromedia Flash, a vector-based drawing software capable of generating timeline-based animations. It used ActionScript, a proprietary language for building interactive algorithms. Flash could generate applets that were embedded in HTML pages using the `<embed>` tag and was used to create web pages, advertisements, games, and various web art experiences.

The ease of use of Flash's builder interface led to a generation of web art experiences, such as those created by Yugo Nakamura on the Yougop.com website [20], which is no longer active. Although the sound processing capabilities of ActionScript were basic, some of these experiences included simple sound interactivity, such as playing and stopping sound files [21]. Kip Parker [22], for example, explains how to build a stylophone-like interface with Flash technology by looping sound samples.

However, one important aspect of Flash applets is that they are essentially closed, in contrast to the main web technologies, such as HTML, JavaScript, and CSS, which are open, noncompiled languages. Flash content cannot be inspected by users, a feature that was often exploited to insert malicious code. Additionally, because Flash relied on proprietary technologies, all Flash (and other vendor-specific user interface [UI] technologies, such as Silverlight) content was subject to the vendors' control, who were required to provide free plug-ins for browsers. This led the W3C to develop new standards that would enable HTML to support multimedia and provide solutions for media interaction without the risk of vendor lock-in (W3C, 2008).

In 2008, W3C released the first version of HTML5, which included the `<video>` and `<audio>` elements. With these elements, browsers could directly render and control media files without the need for third-party plug-ins. Additionally, HTML5 introduced the `<canvas>` element, which serves as a visual display for JavaScript code. These elements made it possible to create an interactive viewport to control media and graphics, such as Scalable Vector Graphics files, images, and video files. With these HTML objects, it became possible to create interactive media pages directly in the browser.

HTML5 could process multimedia elements on the web without the need for any plug-ins, which led to a boycott of Flash plug-ins on Apple's devices due to alleged security issues. In December 2020, Adobe discontinued its Flash media plug-in for the web, rendering much artwork obsolete [23].

4 WEB AUDIO API

Although these HTML5 elements allowed one to play audio media on the web, they were only sufficient to replace the audio processing capabilities of previous technologies because they essentially support basic streaming and audio playback. With HTML5 implementation, browsers also started to work on using JavaScript to extend the `<audio>` element possibilities. Initiatives such as Audio Data API, released in 2010 by Mozilla Foundation, started to implement tools to allow reading and writing audio[4].

Following these initiatives, the W3C began working on an application programming interface (API) to control audio digital signal processors through JavaScript, to provide standards for developing more complex audio applications, such as "sophisticated web-based games and interactive applications" [24], with special methods and objects specifically designed for audio processing and synthesis. The first version of this API, the Web Audio API (WAA), was released at the end of 2011.

Far more advanced than Flash audio processing capabilities, it was designed to support modular routing, scheduled sound playback, automation of audio parameters, audio stream synthesis and processing, spatialized audio, convolution engines, dynamic compression, real-time time domain and frequency analysis for sound visualization, efficient filters, and other nonlinear effects such as wave shaping. Although this was not enough to replace current digital audio workstation (DAW) software, it was sufficient to create several types of web application, such as online games, participatory performances, and musical instruments. Later versions included oscillators, flexible channel handling, live audio input processing, and integration with WebRTC to process and send sound to and from remote peers.

The WAA is conceptually influenced by decades of research in computer music synthesis, particularly the

[4] https://wiki.mozilla.org/Audio_Data_API

MUSIC-N series of programs and his descendants (e.g., Csound, Pure Data, SuperCollider). These systems were designed to describe and generate sound using unit generators, a modular approach where audio processing is broken down into interconnected components. They established the paradigm of modular synthesis, where audio is generated and processed by connecting functional blocks (oscillators, filters, delays, etc.). This idea directly informs the WAA design that allows digital signal processing based on connections and routing between AudioNodes inside AudioContext.

At the beginning of the Web Audio Working Group, there was at least one alternative proposal for an audio processing web API, the Mozilla Audio Data API, which aimed to allow developers to read and write raw audio data directly. The final choice of node-based architecture highlights a specific technical and cultural trajectory for web audio development, prioritizing the computer music paradigm. This level of complexity is often more familiar to researchers and academics than to typical web developers. To address this gap, several libraries were created, including Tone.js, a framework designed to enable interactive music creation in the browser aimed at audio engineers and musicians accustomed to DAWs [25], and WebAudioXML, a modular system intended for integration into larger applications designed to make web audio tools more accessible to artistic creators [26]. These libraries have significantly facilitated the development of many subsequent web audio applications.

It is important to note that the web standards published by the W3C are not pieces of software but public recommendations. Browser developers, including those behind Google Chrome, Mozilla Firefox, and Internet Explorer, are expected to implement these standards within their rendering engines, although they retain the freedom to do so at their own discretion. As observed by Lazarini and Yi [27], variations in the way these APIs are implemented can result in different sound outputs for the same code in different browsers. In practice, these standards are not always fully compatible with all browsers at all times, although most of them are now supported by the major browsers that are freely available.

5 WEB AUDIO EXPERIMENTS

Since the release of the WAA, many researchers have dedicated significant effort to advancing technologies for playing and producing audio on the web. In addition to the Web Audio Conference that has presented web audio–related technologies and applications since 2015, there is also the Ubiquitous Music Workshop, held annually, which serves as a showcase for technologies and art projects in the field of ubiquitous music. This research area "combines methodologies from music, computer science, education, creativity studies, human sciences, and engineering" [28], with a strong emphasis on web and mobile applications, technologies, and art pieces.

When the first author started investigating web technologies to produce music for her Ph.D. in 2015, she made a study of all the sound-related experiments available at the time on Google Chrome Experiments website, a website that is a "showcase of work by coders who are pushing the boundaries of web technology, creating beautiful, unique web experiences" [29]. At that time, the experiences were more fun oriented and quite simple, both in terms of audio processing capabilities and artistic aspects.[5] Although most of the experiments are offline at present, some of them are still functional.

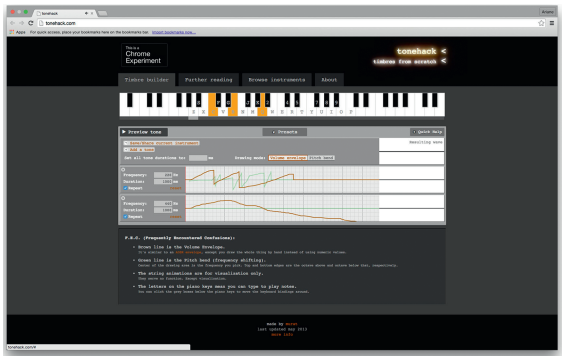


Fig. 2. Tonehack, an early web audio synth by Murat Ayfer. Open-source. Screenshots by the author on July 27, 2015. Source: http://tonehack.com/.

Some examples of early experiments from 2015 that are still operational include Plink, a multiuser sampling interface where users can play sound samples together, and Tonehack (Fig. 2), a web-based waveshaper synth. At that time, the WAA had just been released, and most of the experiments were still in the early stages of web audio processing and generation. Some of them, however, brought significant conceptual contributions as instruments for playing through the web, such as Multiplayer Piano, a website that allows users to play a virtual piano together with people from different locations, allowing real-time music collaboration. Boris Smus' Spectrogram and Oscillator is also very useful today for the practices of audio professors, by providing a sine oscillator that mapped vertical coordinates to frequencies ranging from 1 to 21.1 kHz together with real-time spectral analysis of sound input from the user's microphone.

With the community's effort, the volume and quality of web audio applications escalated very quickly. For example, as early as 2016, Radio.Garden, an impressive website that uses a 3D model of the planet, allowed users to browse and listen to online radio stations from all over the world. From a single window, users can navigate through the diverse multilingual audio cultures of the world [30].

At that time, most of these experiences were more about playing for fun than playing in real performance or production contexts by musicians or sound artists. Some functioned as interactive compositions created by code artists to

[5] The report categorized and analysed all the audio related experiments available at Google Chrome experiments at the time. It was part of the first author's research that was not published, but is available on the author's website at https://ariane.stolfi.org/textos/.

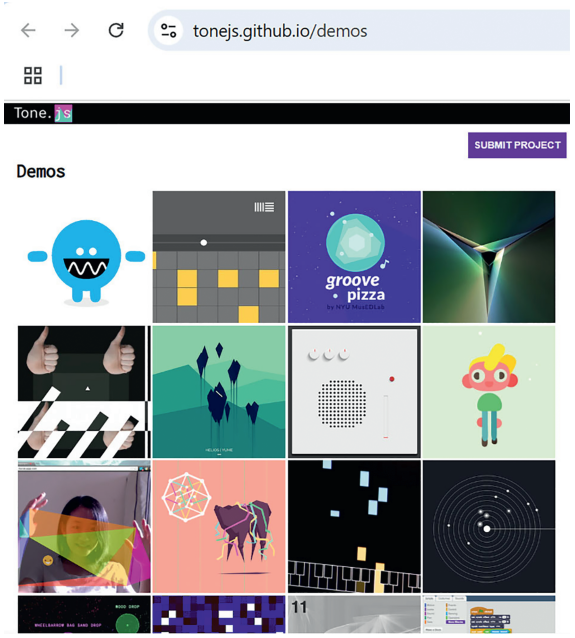


Fig. 3. Tone.js demos. Massachusetts Institute of Technology MIT license. Source: https://tonejs.github.io/demos. Screenshot by the author.

be played by users, offering a fixed selection of sounds and working with sound samples. Others tried to emulate traditional instruments, such as synthesizers or guitars, within a web interface. However, by being open and accessible in the browser, they also allowed nonmusicians to engage and play.

Both Chrome Music Lab[6] and Tone.js demos still offer a wide range of sound art experiments that inspire a strong creative flow. Chrome Music Lab is explicitly described as a website that makes music learning more accessible through fun and hands-on experiences. Teachers can use the experiences to explore music and its connections to other fields. One notable example is "Song Maker," which allows users to create and share music.

Similarly, Tone.js's Demos (Fig. 3) page features a diverse collection of projects, ranging from rhythm sequencers such as Groove Pizza[7] and Polymetric Drum Machine[8] to more interactive experiences such as AI Duet[9] and Weather Synth[10]. A key feature shared by both platforms is that all these experiments and demos are built using open-source web technologies such as the WAA and Tone.js. Many also have their source code available on GitHub, allowing users to study, modify, and incorporate them into their own projects. This transparency and accessibility makes these experiments valuable tools for learning and creative experimentation in the realm of music and digital sound.

[6] https://musiclab.chromeexperiments.com/
[7] https://apps.musedlab.org/groovepizza/
[8] https://react-drum-machine-6f184.firebaseapp.com/
[9] https://experiments.withgoogle.com/ai/ai-duet/view/
[10] https://psherrard.github.io/patrickSherrardProject5/

An example the authors would like to share is the Acoustic Atlas [31], developed by Dr. Corbi van Tonder and Mariana Lopez, which provides an interface that reproduces the acoustics of cultural heritage sites, such as caves, historical constructions, and spaces with significant acoustic properties, such as echo, reverb. They collect room impulse responses from the selected spots and reproduce them in web audio processing, allowing the users of the website to experience the sound of the real spaces.

6 MAKING MUSIC ON THE WEB

In 2015, Lazarini, Yi and Timoney [27] raised some questioning about the WAA, including the "abscense of the concept of instrument," that does "not offer much in terms of lower-level access to audio computation," there is no event data structure to schedule. Some of their considerations were addressed by the introduction of the AudioWorklet Object, which allowed communication to low-level applications for sound processing developed in other programming languages.

After the advancements proposed by the first versions of the WAA, new developments continued to be made until the 2018 version introduced the possibility of using AudioWorklets, an interface that "Ideally, it should be able to support any use case which could reasonably be implemented with an optimized C++ engine controlled via script and run in a browser" [32], allowing the development of far more complex audio applications. One example is the Csound implementation for web [33], using the WebAssembly AudioWorklet, "that provides a full Csound livecoding environment in the browser," or the FAUST implementation based on the same technologies [34], which allows the design of several audio applications, including Virtual Studio Technology simulation on the web.

The integration of web browsers into music production environments has emerged as a significant development in computer-mediated creativity, driven by advances in the ecosystem made on top of the WAA and JavaScript. These frameworks enable real-time audio processing, synthesis, and MIDI control directly within the browser, reducing the dependency on proprietary software and specialized hardware. It is simple to observe that browser-based tools enhance accessibility and facilitate music production in diverse contexts, from educational applications to remote collaborative projects. This paradigm shift aligns with broader trends in computational art, where lightweight, platform-agnostic solutions are increasingly favored for their flexibility and low entrance barrier.

This section systematically examines the role of web technologies in contemporary music production through three subsections. Online DAWs, the creative code ecosystem, and case studies. Online DAWs assess the functional capabilities and limitations of cloud-based DAWs. The creative code ecosystem investigates the proliferation of JavaScript libraries and browser-based frameworks that support algorithmic composition and interactive sound design. Case studies analyze some contributions of the authors to the field through real-world implementations and evalu-

ate their technical and artistic efficacy. Collectively, these subsections provide a structured analysis of how web-based tools are reshaping music production practices, offering insights into their potential for future innovation in the field.

6.1 Online Digital Audio Workstations

Proprietary platforms, such as BandLab[11] and Soundtrap[12], represent an important facet of the current web as a music creation environment, bringing to the browser the experience of a fully functional DAW with advanced production, editing, and collaboration features in real time. Unlike open-source and live coding tools, these solutions rely on a plug-and-play approach, democratizing access to music production for less technical users but without giving up professional features.

A big feature of these platforms is the synchronous collaboration, where multiple users can edit the same musical session simultaneously, whether composing, mixing, or recording instrumental parts, a dynamic that resembles studio work but is geographically distributed. BandLab and Soundtrap prioritize structured production, with integrated sharing and cloud-saving tools.

This trend reflects a commoditization of DAWs, making them accessible directly in the browser without the need for software installation or powerful hardware. However, unlike the open-source and live coding ecosystem, which value customization and algorithmic transparency, these platforms operate in a closed, cloudcentric model, which can limit creative freedom in exchange for convenience. Still, their commercial success shows that there is room for multiple approaches to web-based music creation, from highly technical solutions to more intuitive, mass market tools. They are especially valuable in contexts where software acquisition is difficult, such as in low-budget educational contexts.

We can also see a trend toward using the web as a platform for remote musical interaction that brings back the essence of in-person collective improvisation and live performance but is now mediated by technology. Projects such as NINJAM [35] (which allows musicians to play together in real time, even with latency, through synchronized loops), JackTrip (aimed at high-quality performances with ultra–low-latency audio streaming via high-speed networks) and Quacktrip [36] (a more accessible adaptation of JackTrip for less specialized scenarios) demonstrate how the Internet can be a space for collaborative and synchronous musical creation, overcoming geographical barriers.

6.2 Creative Code Ecosystem

In recent years, there has been a growing trend of creative coding environments and music synthesis tools moving to the web browser, using JavaScript and the WAA to democratize access to sound creation. Projects like Csound.js[13] (which port the powerful Csound to the web), WebPd[14] (a Pure Data JavaScript implementation), and SunVox.js[15] (a web version of the modular SunVox tracker) exemplify this movement, allowing artists to code and share interactive compositions directly in the browser without the need for installation. This approach not only facilitates collaboration and music education but also integrates with visual frameworks (like Three.js[16]) for immersive audiovisual experiences [37], aligning with the expansion of generative art on the web. The portability and open-source nature of these tools reinforce a culture of accessible experimentation that is essential to the evolution of algorithmic music and digital performance.

Beyond that, the live coding movement has radically transformed the relationship between programming and music creation, turning it into a collective, performative, and real-time practice [39]. Projects like Strudel.cc[17] (which uses JavaScript to create algorithmic patterns in an accessible and collaborative way), Flok.cc[18] (a shared live coding platform where multiple artists program and improvise together in a synchronous session), and Gibber[19] (that currently also has chat and visualization features) [39], exemplify this evolution. These tools not only democratize algorithmic music production but also spectacularize the act of programming [40], transforming it into a visible and participatory artistic gesture.

Festival performances, live streams, and collaborative sessions reinforce the idea that code is both an instrument and a performative language, where creation ceases to be a solitary process and becomes a collective, playful, and open experience, strengthening communities around creative programming and experimental music. This trend also dialogues with the hacker culture and the ethics of free software, where code sharing and networked improvisation expand the possibilities of generative art and human-computer interaction.

6.3 Web Audio Frameworks

Creating music directly with the pure WAA can be challenging because it requires explicitly defining subsymbolic musical aspects in JavaScript using high-level programming methods. Instead of working with ready-made musical abstractions, developers must manually implement concepts, such as note sequencing, tempo control, and audio manipulation, using the API's low-level nodes and structure. For example, even creating a simple metronome requires detailed programming of audio scheduling with JavaScript and the WAA.

Moreover, precise audio timing management, as discussed in various articles and projects on audio scheduling [41, 42], is crucial and complex. The existence of numerous libraries and frameworks, such as Tone.js, Gibber.cc[20] (Fig. 4), and Strudel.cc[21], built on top of the WAA sug-

[11] Bandlab.com
[12] Soundtrap.com
[13] https://ide.csound.com/
[14] https://sebpiq.github.io/WebPd_website/?target=app&build=1
[15] https://warmplace.ru/soft/sunvox/jsplay/
[16] https://threejs.org/
[17] https://strudel.cc/
[18] https://flok.cc/
[19] https://gibber.cc/
[20] https://gibber.cc/

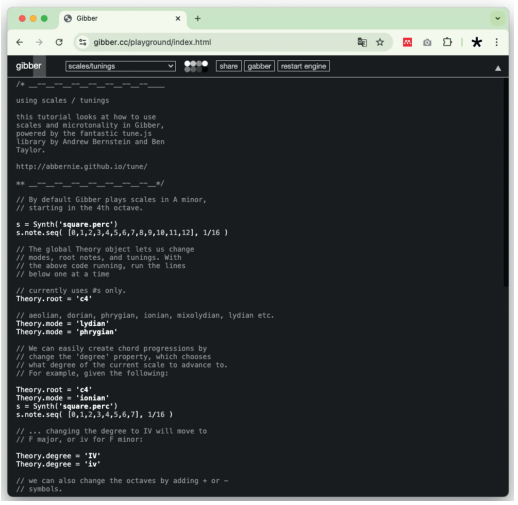


Fig. 4. Gibber interface by Charlie Roberts. Open-source. Source: https://gibber.cc/playground/index.html. Screenshot by the author.

gests that whereas the API itself is powerful it may be too low-level for high-level musical tasks. As a result, developing sophisticated music applications often involves constructing these subsymbolic musical abstractions within JavaScript code.

The Tone.js [43] library offers a solution to the problem of scheduling temporal events through a global clock to synchronize and coordinate musical events. It provides tools that allows one to schedule methods along a global timeline that can be repeated and traversed. It also facilitates dealing with musical notation creating tempo curves where the BPM gradually change over time and scheduling start and stop methods with sample-level precision. The p5.js project [44] inherited a knowledge base and code compatibility from the Processing programming language, which means that a lot of documentation on graphic creation and interactive art development with this library had been used.

6.4 Case Studies

The Música Movel [45] project was a process of creative interdisciplinary immersion to develop open-source music production tools for mobile devices with the collaboration of artists and programmers. The update of the Música Móvel took place between 2021 and 2023 with the aim of building experiments with WAA to create music and teach creative code. The first iteration in 2021–2022 (Fig. 5)[22], is described as an experiment in two-channel audio sampler slicing, offering the ability to redefine the size and quantity of audio slices and loops.

The first iteration of the Música Móvel project demonstrates how the temporal precision of Tone.js can be used to interactively manipulate audio. Choosing P5.js to build the graphical UI allows users to visually interact with the slicing and looping parameters while controlling the underlying audio engine built with Tone.js. In this way, the project concretely illustrates how the combination of these two libraries facilitates the creation of interactive visual music experiences.

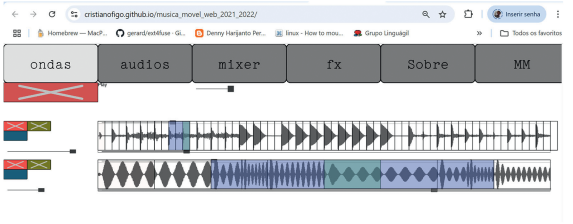


Fig. 5. Música Móvel 2021–2022 (Tone.js + P5.js) by Cristiano Figueiró. Screenshot by the author.

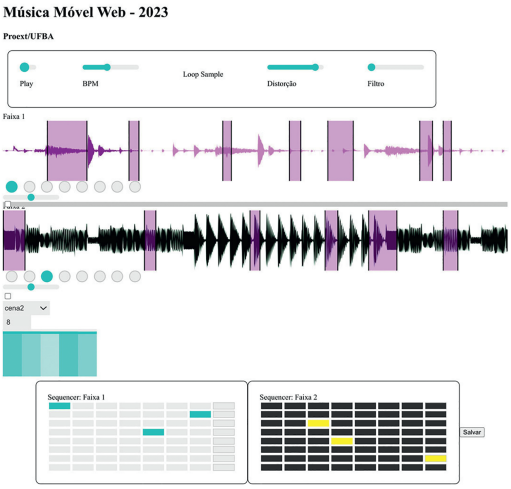


Fig. 6. Música Móvel 2022–2023 (Tone.js + wavesurfer.js + NexusUI.js) by Crisitano Figueiró. Screenshot by the author.

The second iteration of the Música Móvel project (Fig. 6)[23], represents a significant improvement in interface and functionality compared with the previous version. This version maintains the two-channel sample audio slicer and individual sequencers for each channel, allowing precise control over slice playback.

A notable improvement in this iteration is the visualization of waveforms through Wavesurfer.js, complemented by its Regions plug-in[24]. This choice was crucial for the visual and intuitive editing of audio regions, allowing users to select, resize, and move slices directly on the graphical representation of the sound wave using the mouse. This direct interaction with the waveform and regions provides highly effective temporal gestural control, aligning with the previous discussion of the importance of libraries that facilitate fluid temporal manipulation.

Additionally, the interface of this second iteration uses the NexusUI.js[25] library to build the sequencers, sliders, and menus. NexusUI offers a collection of HTML5 in-

[21] https://strudel.cc/

[22] Hosted on GitHub at https://github.com/cristianofigo/musica_movel_web_2021_2022.

[23] This version is located in the repository https://github.com/cristianofigo/MusicaMovel_22_23.

[24] https://wavesurfer.xyz/example/regions/

[25] https://nexus-js.github.io/ui/



Fig. 7. Playsound web interface by Ariane Stolfi. Screenshot by the author in November 2025



Fig. 8. Noise Symphony excerpt by Ariane Stolfi. Screenshot by the author

terfaces designed for web audio applications, making it a sensible choice to provide interactive and visual controls for sequencing slices and adjusting other slicer parameters.

This modular strategy not only enhances usability for musicians and sound artists but also reflects a broader trend in web audio development: leveraging the right tool for each subtask to balance creativity, responsiveness, and computational efficiency. In particular, this practice aligns with the education of contemporary arts [46], where creative coding experiments for artistic creation [47] have become integral components of undergraduate curricula in digital arts, music technology, and multimedia programs around the world.

Web technologies can also help implement ideas for music making that go beyond traditional interfaces. Playsound.space [48] (Fig. 7), for example, uses the Freesound API [49] to provide a real-time interface to play free improvisation music or to create live soundscapes. Built in Angular.js with the WAA without any audio framework, it allows users to query the Freesound database by spectrograms and create a personal sound collection that can be looped, cut, and processed in real time, breaking away from a timeline-based grid paradigm.

Playsound is based on a concrete music approach, using only prerecorded sound samples for playback. For special cases where users needed specific sounds, the recordings were previously submitted to the Freesound library. It has been used for seven years in live performances and pieces proposed by the author and their groups in various works, such as Noise Symphony (Fig. 8), Reverbera!, Telematic City Jam, and others, and remains the primary instrument of the first author for playing in live performances. Additionally, it provides a valuable environment for education, allowing students to quickly compose soundtracks for videos and images.

6.5 Artificial Intelligence Music Tools on the Web

Today, the use of artificial intelligence (AI) in music creation has gained prominence with projects such as Magenta.js, a library developed by Google that integrates machine learning models (such as MusicVAE and DDSP) into the web ecosystem [50, 51], allowing the generation of algorithmic melodies, harmonies, and rhythms directly in the browser. This approach expands the possibilities of creative coding by combining traditional programming with machine learning to explore new forms of generative and interactive composition.

At the same time, proprietary platforms, such as Suno AI[26] (which generates complete songs from textual prompts), represent a more automated but less flexible path, where creative control is traded for ease of use. This duality between open tools (Magenta.js) and closed solutions (Suno) reflects a crucial debate in the field of music and AI: although the former favors experimentation and integration with other web technologies (such as the WAA and Three.js), the latter focuses on immediate accessibility but with limitations in customization. By allowing music creation to people without any musical skills, these technologies certainly amplify general accessibility to music creation, but several ethics and aesthetic debates are emerging from this new perspective, such as copyright infringement practices form AI trainers, how to pay for automatic music, and even about the doubtful qualities of the music produced by these kinds of software.

For artists and developers, the choice between these paths depends on the objective: whether it is exploring the frontier of AI-assisted creation in a transparent and hackable way or adopting ready-made solutions for rapid prototyping, which are increasingly present in the music market. Questions about the role of the artist in the face of these new technologies are emerging, such as if robots are making art, what will be left for people to do?

7 CONCLUSION

The evolution of audio on the web reflects a broader transformation in the way music is created, shared, and experienced in the digital age. From early P2P networks, such as Napster, to the democratization of production through the WAA and collaborative platforms, such as BandLab, the web has become a dynamic space for musical innovation. Technologies like Tone.js, Csound.js, and live coding environments (e.g., Strudel.cc) exemplify the shift toward

[26] https://suno.com/home

accessible code-based creativity, whereas proprietary tools like Suno AI highlight the tension between customization and convenience in AI-assisted composition. The rise of synchronous collaboration, whether through DAWs in the browser, live coding performances, or remote improvisation, demonstrates the web's capacity to bridge geographical and technical divides, fostering collective artistry.

As the web continues to mature as a metamedium for audio, challenges remain in balancing open standards with proprietary ecosystems, low-level control with high-level abstractions, and individual expression with collaborative workflows. Future developments may further integrate AI, spatial audio, and immersive technologies, pushing the boundaries of participatory music making. The enduring lesson is clear: the strength of the web lies in its duality as both a platform for technical experimentation and a global stage for shared human creativity.

8 ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which greatly contributed to improving the quality and critical depth of this article.

9 REFERENCES

[1] M. McLuhan, *Understanding Media: The Extensions of Man* (Signet Books, New York, NY, 1964).

[2] Bell Daniel, "Thinking Ahead," *Harvard Business Review*, pp. 20–42 (Jun. 1979).

[3] T. Berners-Lee and R. Cailliau, "WorldWideWeb: Proposal for a HyperText Project," W3C Proposal (2009 Nov.). http://www.w3.org/Proposal.html (accessed Nov. 15, 2025).

[4] W3C, "Our Mission," W3C Mission Statement (2013) https://www.w3.org/Consortium/mission (accessed Nov. 15, 2025).

[5] P. Levinson, *Digital McLuhan: A Guide to the Information Millennium* (Routledge, Oxfordshire, UK, 2001).

[6] K. Brandenburg, "MP3 and AAC Explained," in *Proceedings of the 17th AES International Conference: High-Quality Audio Coding* (1999 Sep.), paper 17-009.

[7] C. Casey, "Web Rings: An Alternative to Search Engines," *College & Research Libraries News*, vol. 59, no. 10, pp. 761–763 (1998 Nov.).

[8] P. J. Honigsberg, "The Evolution and Revolution of Napster," *University of San Francisco Law Review*, vol. 36, pp. 473–508 (2002 Nov.). https://papers.ssrn.com/sol3/papers.cfm?abstract_id=830884.

[9] A. Brennan, "How an Unknown MP3 Service Led to Tumblr's Fandometrics," https://web.archive.org/web/20241013144933/https://www.dailydot.com/unclick/audiogalaxy-mp3-service-tumblr/ (accessed Nov. 15, 2025).

[10] K. VanScoy, "Turning the Internet on Its Ear: Thomas Dolby's Euphonious Headspace," *The Austin Chronicle* (1997 Mar.). https://www.austinchronicle.com/screens/1997-03-07/527546/.

[11] D. Kleiner and B. Wyrick, "InfoEnclosure 2.0," *Mute Magazine*, Vol. 2, no. 4 (2007 Jan.). http://www.metamute.org/editorial/articles/infoenclosure-2.0.

[12] P. Cashmore, "MySpace, America's Number One," *Mashable* (2006 Jul.). https://mashable.com/archive/myspace-americas-number-one (accessed Nov. 15, 2025).

[13] Hern Alex, "Myspace Loses All Content Uploaded Before 2016," *The Guardian*," (2019 Mar.). https://www.theguardian.com/technology/2019/mar/18/myspace-loses-all-content-uploaded-before-2016 (accessed Apr. 24, 2025).

[14] S. Kamel, "Information Superhighways, a Potential for Socioeconomic and Cultural Development," in *Proceedings of the 6th IRMA Conference*, pp. 21–24 (Atlanta GA) (1995 May).

[15] J. Markoff, "Building the Electronic Superhighway," *New York Times* 1993 Jan.). https://www.nytimes.com/1993/01/24/business/building-the-electronic-superhighway.html.

[16] M. Matassa, "Internet Radio – Computers Help You Hear What Might Be Broadcasts," *The Seattle Times* (1999 Jun.). https://archive.seattletimes.com/archive/19990627/2968794/internet-radio—-computers-help-you-hear-what-might-be-broadcasts.

[17] A. Renaud, A. Carôt, and P. Rebelo, "Networked Music Performance: State of the Art," in *Proceedings of the 30th AES International Conference* (2007 Jan.), paper 16.

[18] J. Burgess, "YouTube and the Formalisation of Amateur Media," in *Amateur Media*, pp. 53–58 (Routledge, London, UK, 2012), 1st ed.

[19] K. Borgsmiller, "YouTube vs. the Music Industry: Are Online Service Providers Doing Enough to Prevent Piracy," *S. Ill. U.L.J.*, vol. 43, p. 647–663 (2018 Apr./May).

[20] Y. Nakamura, "CRASH," THA LTD (2009 Jun.). https://tha.jp/456 (accessed Nov. 15, 2025).

[21] Adobe, "Sound and ActionScript in Adobe Animate," (2023 May). https://helpx.adobe.com/animate/using/sound-actionscript.html (accessed Nov. 15, 2025).

[22] K. Parker, "Sound Toys*Byte-Size Flash MX: Adventures In Design Optimization*, pp. 131–173 (Apress, Berkely, CA, 2002).

[23] A. Mladentseva, "Responding to Obsolescence in Flash-Based Net Art: a Case Study on Migrating Sinae Kim's Genesis," *J. Inst. Conserv.*, vol. 45, no. 1, pp. 52–68 (2022 Jan.). https://doi.org/10.1080/19455224.2021.2007412.

[24] World Wide Web Consortium, "Web Audio API 1.1," (2024 Nov.). https://www.w3.org/TR/webaudio-1.1/ (accessed Nov. 15, 2025).

[25] Y. Mann, "Interactive Music With Tone.js," in *Proceedings of the 1st Annual Web Audio Conference (WAC)* (Paris, France) (2015 Jan.).

[26] H. Lindetorp, "Sonification for Everyone Everywhere–Evaluating the WebAudioXML Sonification Toolkit for Browsers," in *Proceedings of the 26th International Conference on Auditory Display* (Virtual) (2021 Jun.).

[27] V. Lazzarini, S. Yi, and J. Timoney, "WebAudio: Some Critical Considerations," in *Proceedings of the VI Ubiquitous Music Workshop* (Vaxjo, Sweden) (2015 Jun./Jul.).

[28] UBIMUS, "UBIMUS 2024 Macao," in *Proceedings of the 14th International Symposium of Ubiquitous Music* (Macao, China) (2024 Oct-Nov.). https://ubimus2024.fah.usj.edu.mo/.

[29] Google, "Chrome Experiments," https://experiments.withgoogle.com/collection/chrome (accessed Nov. 15, 2025).

[30] C. Mitchell, L. Peter, F. Golo, et al., "radio.garden," (2016). https://sure.sunderland.ac.uk/id/eprint/9477/ (accessed Nov. 15, 2025).

[31] C. Van Tonder and M. Lopez, "Acoustic Atlas - Auralisation in the Browser," in *Proceedings of the International Conference on Immersive and 3D Audio: From Architecture to Automotive* (Bologna, Italy) (2021 Sep.). https://doi.org/10.1109/I3DA48870.2021.9610909.

[32] P. Adenot and R. Toy, "Web Audio API - W3 Candidate Recomendation," (2018). https://www.w3.org/TR/2018/CR-webaudio-20180918/ (accessed Nov. 15, 2025).

[33] S. Yi, V. Lazzarini, and E. Costello, "Webassembly AudioWorklet Csound," in *Proceedings of the Web Audio Conference WAC* (Berlin, Germany) (2018 Sep.).

[34] S. Letz, Y. Orlarey, and D. Fober, "Compiling Faust Audio DSP Code to WebAssembly," in *Proceedings of the Web Audio Conference (WAC)* (London, UK) (2017 Aug.). https://hal.science/hal-03162898.

[35] E. V. Buskirk, "NINJAM: Near-Real-Time, Glitch-free Online Music Collaboration," *Wired* (2007 Apr.). https://www.wired.com/2007/04/ninjam-nearreal/.

[36] M. Puckette, "Quacktrip, Neety McNetface, and Music 101 - Network Audio for Musicians" (2020). https://msp.ucsd.edu/tools/quacktrip/doc.htm (accessed Nov. 15, 2025).

[37] A. McLean and E. Harlizius-Klück, "Fabricating Algorithmic Art," PENELOPE (2018 Sep.). https://doi.org/10.5281/zenodo.2155744 (accessed Nov. 15, 2025).

[38] T. Magnusson and A. McLean, "Performing With Patterns Of Time," in R. T. Dean and A. McLean (Eds.), *Oxford Handbook of Algorithmic Music*, Oxford Handbooks, pp. 245–256 (Oxford Academic, Oxford, UK, 2018). https://doi.org/10.1093/oxfordhb/9780190226992.013.21.

[39] C. Roberts, I. Hattwick, E. Sheffield, and G. Smith, "Rethinking Networked Collaboration in the Live Coding Environment Gibber," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, p. 28 (Auckland, New Zealand) (2022 Jun./Jul.) https://doi.org/10.21428/92fbeb44.38cb7745.

[40] A. McLean, G. Fanfani, and E. Harlizius-Klück, "Cyclic Patterns of Movement Across Weaving, Epiplokē and Live Coding," *Dancecult*, vol. 10, no. 1, pp. 5–30 (2018 Nov.). https://doi.org/10.12801/1947-5403.2018.10.01.01.

[41] C. Wilson, "Uma história sobre dois relógios," web.dev (2013 Jan.) https://web.dev/articles/audio-scheduling?hl=pt-br (accessed Nov. 15, 2025).

[42] M. van Dijk, "errozero/beatstepper," (2025 Mar.). https://github.com/errozero/beatstepper (accessed Nov. 15, 2025).

[43] Y. Mann, "Interactive Music with Tone.js," in *Proceedings of the 1st International Web Audio Conference (WAC)* (Paris, France) (2015 Jan.).

[44] L. McCarthy, C. Reas, and B. Fry, *Getting Started With P5.js: Making Interactive Graphics in JavaScript and Processing* (Maker Media, San Francisco, CA, 2016).

[45] B. Rohde, C. Figueiró, and G. R. Soares, "Explorações e Estratégias em Música Móvel: Processo Criativo de Produção de Código Aberto Para Experimentações Sonoras Interativas em Dispositivos com Android," in *Proceedings of the IV International Symposium on Innovation in Interactive Media* (Goiania, Brazil) (2016 May).

[46] T. M. Nakra, *Constructing Music: Musical Explorations in Creative Coding* (Oxford University Press, New York, 2024), 1st ed. https://doi.org/10.1093/oso/9780197669198.001.0001.

[47] E. Edmonds, "Art and Code: Programming as a Medium," in A. Brooks, E. Brooks (Eds.), *Interactivity, Game Creation, Design, Learning, and Innovation*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, pp. 3–12 (Springer, Cham, Switzerland, 2020). https://doi.org/10.1007/978-3-030-53294-9_1.

[48] A. S. Stolfi, A. Milo, and M. Barthet, "Playsound. Space: Improvising in the Browser With Semantic Sound Objects," *J. New Music Res.*, vol. 48, no. 4, pp. 366–384 (2019 Sep.). https://doi.org/10.1080/09298215.2019.1649433.

[49] V. Akkermans, F. Font Corbera, J. Funollet, et al., "Freesound 2: An Improved Platform for Sharing Audio Clips," in *Proceedings of the 12th International Society for Music Information Retrieval Conference* (Miami, FL) (2011 Oct.).

[50] J. Engel, L. H. Hantrakul, C. Gu, and A. Roberts, "DDSP: Differentiable Digital Signal Processing," in *Proceedings of the International Conference on Learning Representations* (Addis Ababa, Ethiopia) (2020 Apr.).

[51] A. Roberts, J. Engel, C. Raffel, C. Hawthorne, and D. Eck, "A Hierarchical Latent Vector Model for Learning Long-Term Structure in Music," in *Proceedings of the 35th International Conference on Machine Learning*, vol. 80, pp. 4364–4373 (Stockholm, Sweden) (2018 Jul.).

THE AUTHORS



Ariane de Souza Stolfi



Cristiano Figueiró

Ariane de Souza Stolfi is an architect, designer, programmer, and musician, transiting between languages and disciplines and researching interactive interfaces using web technologies such as Playsound.space. She is a professor at Centro de Formação em Artes e Comunicação (CFAC)–Universidade Federal do Sul da Bahia (UFSB), teaching in the Sound, Image, and Motion Program. She holds a Ph.D. in sonology (Centro de Música (CMU)-USP) and a Master's degree in architecture and design (Faculty of Architecture and Urbanism (FAU)–USP). She has been involved with the open-source community in Brazil, collaborating with several projects and organizations and has participated in numerous festivals and events, presenting interactive installations and performances in Europe and Brazil. With free music improvisation practice, she is currently the creative director of the Reverbera! project and collaborates with the Female Laptop Orchestra.

•

Cristiano Figueiró is a musician, professor and is dedicated to researching the interfaces between composition and interactive technologies with free software. He develops sound creation works in partnerships with various artists covering different genres and techniques using guitar and computer in the creative process. He works as a professor at Universidade Federal da Bahia (UFBA) in interdisciplinary bachelor's degrees, researching the intersections between the development of free technologies and the production of sound art from the perspective of culture and teaching.