Blendwave: A sound design tool for audiovisual authors

Werther Azevedo
PUC-Rio
Rua Marquês de São Vicente, 255 – RJ - Brazil
werther@puc-rio.br

José Bidarra
Centro de Investigação em Artes e Comunicação (CIAC), Universidade Aberta, Portugal
jose.bidarra@uab.pt

ABSTRACT

This paper presents Blendwave, an online sound design application inspired by the ease of use of samplers and the "sfxr family" of sound creation software. Starting from the hypothesis where current software tools have made the creation of sound inaccessible for non-specialist users, we analyze the problems with the musical bias ingrained in DAWs and Patchers, the standard sound design tools. After identifying the Web Audio API as an appropriate technological backbone for an accessible tool, recent online applications are surveyed, followed by popular software used in a rapid prototyping scenario. By acknowledging the merits and shortcomings of these sound creators, we present Blendwave, the byproduct of a practice-based research and development effort towards democratizing sound design as a rapid prototyping activity for audiovisual authors. The idea of a sampler architecture is proposed as a way to augment the sound possibilities of sfxr-based tools while keeping their simplicity. Finally, we report on the current state of development and outline the next steps.

CCS CONCEPTS

• **Information systems** → **Multimedia information systems**; • **Applied computing** → Arts and humanities; • **Information systems** → World Wide Web → Web Applications;

KEYWORDS

Sound design, samplers, sound effects, practice-based research, creativity support tools, web audio

ACM Reference format:

W. Azevedo and J. Bidarra. 2017. Blendwave: A sound design tool for audiovisual authors. In *Proceedings of Artech2017, Macau, China, September 06-08, 2017, 4* pages.
http://dx.doi.org/10.1145/3106548.3106596

ARTECH2017, September 06-08, 2017, Macau, China

ACM ISBN 978-1-4503-5273-4/17/09.
http://dx.doi.org/10.1145/3106548.3106596

1 INTRODUCTION

The rise of Homo Sapiens as one of the planet's most successful species is undoubtedly related to the ways by which we process and generate sound. From a relatively small interval of audible air disturbance frequencies, human beings have found and constructed sophisticated communication devices. However, while generating sounds through body moment or the vocal tract may be a trivial task for the average person, the employment of sound as a means of artistic expression remains a process surrounded in awe and mystery. What should be a simple, even intuitive activity, has become a highly specialized craft in the age of digital technology.

In audiovisual media, whether traditional or interactive, sound is usually an inseparable part of an artifact, its employment ubiquitous in sewing the emotional canvas that wraps the author's discourse. If audio is as common an element as imagery, one would expect its generation to be equally intuitive. Still, modern sound design practice encompasses a series of complexities that keep audiovisual authors from experimenting with sound as a sketching tool, like they usually would with images. By skipping the act of creating rapid sound prototypes during the initial stages of creative ideation and, thus, consigning audio to an obligatory post-production position, authors might be missing the opportunity to achieve maximum expressiveness in their works.

1.1 The Misconception of Musical Prerequisites

As Digital Audio lecturer in the Digital Media curriculum, one of the authors has the opportunity of experiencing first-hand the problems faced by future audiovisual authors. At the start of each new term, potential movie makers, animators and game designers are questioned about their motivations to have chosen that course unit. Most declare to be ignorant about the process of creating audio for media, and want to learn more. These young people are usually quite familiar with pen, pencil, graphic design software and video editors, but have never cared to register the soundscape that surrounds them. A common claim is that, by not having had proper musical education in the past, they feel unequipped to understand the creative processes revolving sound. Such assumption is a reasonable one, since music and sound design can be interpreted as different manifestations of a common discourse. Besides, it's no secret that a significant part of audio researchers

and professional practitioners indeed come from a musical background. Nonetheless, to restrict the sound design practice to musicians would be akin to constraining photography to visual artists. In fact, in the Foreword to "The Sound Effects Bible" introduction, Charles Maynes [1] states that "working with sound presents challenges not dissimilar to photography. In the process of capturing sound, we have many similar concerns and production techniques available". The book, a popular sound design manual, follows with no mention of musical prerequisites whatsoever.

1.2 Software Dependent Sound Effects

Not every type of sound used in audiovisual work depend on software tools. Apart from Foley sounds, Opolski [2] divides audio effects into three categories: "1) backgrounds, the sounds that constitute the environment, 2) hard effects, effects that are directly related to a specific on frame sound source, and 3) sound effects, which are non-literal, non-indicial effects that (...) are created by the sound designer from digital processing and synthesis." Although modern sound design techniques usually employ some manner of digital processing for each of these categories, Foley sounds, backgrounds and hard effects are usually achievable through the use of quality microphones and good recording practices. Furthermore, the need for such sounds on a prototype stage are usually satisfied by sound libraries. Sound effects, on the other hand, tend to be abstract representations of fictional elements and, as such, depend on software tools to be conceived. This research is therefore focused on this category of audiovisual sound.

1.3 Problems with Sound Design Tooling

According to Magnusson [3], "Technological objects are (...) never neutral, they contain scripts that we subscribe to or reject according to our ideological constitution. The problem is that the scripts are often well hidden and concealed, which can result in an uncritical use of creative technologies – technologies inherent with ideological content." This notion helps to understand the current state of sound design software, where the developers' ideologies, far from being concealed, are biased towards musical and/or specialist usage.

1.3.1 Digital Audio Workstations. In the digital domain, sound design is mostly practised through the usage of Digtal Audio Workstations – the DAWs. It's "where all the magic happens", according to Viers' Sound Design Bible [1]. While these applications indeed offer a vast array of audio processing and manipulation capabilities, they are designed, developed and marketed for musical usage. The timeline paradigm that governs the user interface might be familiar to audiovisual authors used to video editors and animation software but, in DAWs, the timeline is often at the service of musical time units such as bars, beats and notes. Therefore, while DAWs have been proven efficient sound design tools, their establishment as the de facto method to create sound effects contributes to reinforce the myth that sound design is a musician-only activity.

1.3.2 Patchers. Another user interface paradigm often employed on sound design software is that of the Patcher, defined by Pluckette [4] as "a graphical environment for making real-time computer music". Popularized by tools such as MAX/MSP and Pure Data (Pd), it abandons the timeline as a fundamental metaphor, focusing instead on the signal flow through sound units, operators and modifiers. Unlike DAWs, Patchers, in their Node Graph Architecture, are somewhat dissociated from musical metaphors. However, these are not tools designed with the novice user in mind. As Max Mathews states in Pluckette's Pd companion book, [4], "Max and Pd allow almost anyone to synthesize uninteresting timbres almost instantly. Making interesting timbres is much more difficult and requires much additional knowledge.". Therefore, while Patchers present an inviting environment for the non-musician while compared to DAWs, they lack the simplicity that could help audiovisual authors with no prior sound synthesis experience to generate sounds that are meaningful for their creative aspirations.

2 WEB AUDIO API

The Web Audio API is "high-level JavaScript API for processing and synthesizing audio in web applications" [5]. It was first specified in 2011 and is currently in "Working Draft" status, being supported by a group of web browsers that accounts for most of the connected population [6]. While the browser cannot match specialized native tools in terms of features or performance, Wyse and Subramanian [7] identify a series of advantages of the web platform which are relevant to this investigation:

1. A huge developer community, which results in a rich ecosystem of computer music libraries
2. Connectedness
3. Ease of Access
4. Portability

As much as open source tools like libpd share some of those characteristics, no native solution can match the web browser in terms of ubiquity and effortless access to an application. In a research that looks for ways to democratize a creative process, conceiving a tool that does not require installation or configuration processes and can be launched from an URL is an appealing proposition. Additionally, the cross-platform nature of the web means an author could seamlessly and persistently design sound on multiple devices. Accessibility has been a major directive on the web since its inception, and that alone justifies its use as a platform for Blendwave. Moreover, the vast availability of free web analytics tools will be of great importance in evaluating and iteratively improving the tool, while gaining insight from the process.

3 RELATED WORK

3.1 Sound Design Possibilities with Current Web Audio Tools

The advent of the Web Audio API has ushered a new era of powerful online audio software. Many of the current tools are showcased in the "Web Audio Weekly" newsletter [8]. The applications that are commonly listed by this publication can be crudely categorized into synthesizers [9], sequencers [10], modular environments [11] and live coding IDEs [12]. Most of these are designed around music related goals, usually within the fields of performance or composition. As is the case with DAWs, many of these can be adapted by an experienced user to function as a sound design tool. Blokdust [11], for instance, provides a selection of sources and effects that can be freely arranged and combined in a modular setting to generate a variety of sound effects. Still, the fact that is has been idealized as a "web-based music making app" [13] is once again an open invitation for musicians and "music makers", but not so much for aspiring sound effects creators. This musical bias in current Web Audio applications not only strengthens the musical requirement myth, but also leads to tools that are not streamlined for sound effects creation in terms of usability and workflow.

3.2 Sfxr and Derivatives

A tool that has presented itself as a sound effects creator from its inception is sfxr [14]. Created for the quintessential video-game prototyping environment, the game jam, it is widely used by the game development community as an accessible tool for rapid sound effects generation, being displayed as such on popular game jam websites such as Ludum Dare [15]. Being open source software, it has spawned Flash-based online ports [16] and Web Audio derivatives [17]. These tools offer basic subtractive synthesis capabilities, akin to those of 1980s computers and video-game consoles, which results in sound effects with equivalent aesthetic characteristics. Therefore, despite being popular alternatives for non-musicians, these applications cannot generate a wide gamut of sound effects, their use being usually constrained to the video-game and retro thematic.

With Blendwave, the primary goal is to attain the accessibility and ease of use of sfxr and its derivatives, while also harnessing the sample-based capabilities and processing power present in the aforementioned Web Audio API projects.

4 BLENDWAVE

Blendwave is a creativity support tool (CST) being developed to investigate the hypothesis that basic digital sound design techniques are inaccessible to audiovisual authors with no prior musical or digital audio processing knowledge. The effort to build a new tool to further understand this particular problem is characteristic of a Practice-based research, "an original investigation undertaken in order to gain new knowledge partly by means of practice and the outcomes of that practice" [18]. This methodological approach is suitable to the development of CST's, where the goal is "to facilitate users in generating innovative ideas related to existing open problems, or to new problems that appear on the scene" [19]. Through iterative software enhancement [20], we aim to develop, test and revise multiple versions of the tool in order to evolve our own understanding about the needs of audiovisual authors within the sound design creative domain.

4.1 Sampler Architecture

The design of sound effects on a professional basis is mostly realized by using recorded samples as source material. When describing his sound design process for the series "Stranger Things" [21], Craig Henighan mentions how he combined a great assortment of real world samples to achieve each fictional sound. The reasons for the prominence of sampled sounds over synthetic are straightforward, since many of the abstract entities that sound effects illustrate, such as creatures and weapons, have counterparts in the tangible world. It comes as a surprise, then, that most sound effect creators such as sfxr don't offer sample-based synthesis. By recognizing this contradiction, and its consequential opportunity to merge the popular aspects of sfxr with the authenticity of audio samples, Blendwave is conceived as a sample-based synthesizer or, put simply, a sampler. The ease-of-use of samplers as instruments is often presented as one of the cornerstones of widely popular urban music genres, such as rap and hip-hop, where artists with little traditional music background have found creative ways of artistic expression through the remix culture. As well stated by Navas [22], "Remix as a proper concept and eventual discourse has its birth in the development of music samplers". Blendwave dwells on the possibility that the same process can be applied to sound design, where the remix and processing of audio samples can enable anyone to achieve expressive new sounds. Feature-wise, Blendwave offers some of the capabilities found in software-based samplers (Fig.1): Amplitude Envelope, Amplitude Oscillator, Pitch Shifting, Pitch Oscillator and a range of DSP Effects.

Figure 1: FL Studio Sampler EZ Sampler are references for Blendwave's feature-set

4.2 User Interface

The User Interface was designed with simplicity in mind. As popular as sfxr is, the amount of sliders and parameters encourage a workflow where the user presses the "randomize" button several times until a usable sound is generated. With Blendwave, we wish to encourage critical thinking and experimentation. The first version of the User Interface was designed by means of low fidelity wireframes (Fig.2) that showcase the overall architecture, intended features and basic interaction of the tool. The main features are clearly grouped into sections: Sound Source, Envelope, Filter, Effects and Output. Unlike sfxr, where 22 sound manipulation parameters are exposed in a single screen, Blendwave employs a tabbed interface to linearly guide the user through the traditional sound synthesis workflow: selecting the

source sound sample from a collection, manipulating envelopes and filters, applying additional DSP effects, and then exporting. By logically dividing the workflow into stages, we intend to avoid confusion and analysis paralysis by authors. In terms of content strategy, we have chosen to follow standard computer music terminology with the pedagogical intent of gently introducing them to authors, therefore easing their transition into more specialized tools in the future. To attain such goal, the wireframes propose visual ideas such as the employment of ludic icons, images and labels to help illustrate specialist terms like "envelope" and "filters". By the same reasoning, we also plan to employ animations to help explain DSP effects particularities like filter types, delay feedback, etc. These ideas are to be implemented during the graphic design stage of the development process.

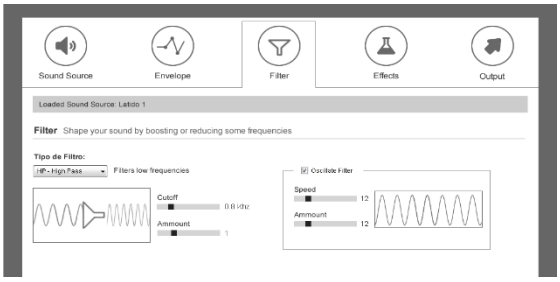

Figure 2: The filter section mockup, represented by a low fidelity wireframe.

4.3 Prototype Implementation

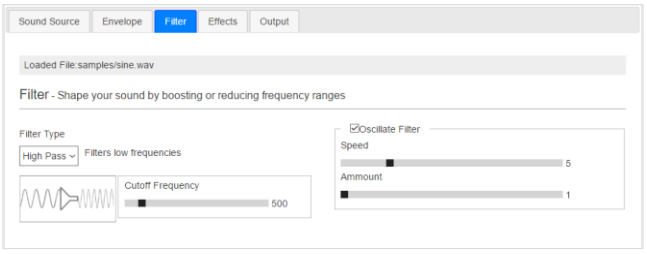

Figure 3: Blendwave prototype, implemented with the Pizzicato Javascript Library

The prototype (Fig.3) is being implemented with the Pizzicato library, which aims to "simplify the way you create and manipulate sounds via the Web Audio API" [23]. Aside from featuring a straightforward API, which speeds up the prototyping process, this library is focused on creating sounds from wave files, by using the Web Audio API AudioBuffer object [5]. Such characteristics make Pizzicato a good match for building a sample-based synthesizer such as Blendwave. Another important feature of Pizzicato that has proven helpful during development is the possibility to access the audio context. Through this method, some functionalities that aren't provided by the library, such as pitch shifting, can be implemented by directly using the lower level functionality provided by the Web Audio API.

The current state of the prototype offers working implementations of the Sound Source, Envelope and Filter Sections. Some of the wireframe's specifications had to be simplified due to the technical difficulties or the lack of support by Pizzicato, but we intend to use the testing stage to validate these simplifications before diving into sophisticated solutions for features that might not be needed.

5 CONCLUSION

We have presented the problems and misconceptions faced by audiovisual authors when addressing the need to design sound effects for their work. The state of the art of current sound design and Web Audio tooling was analyzed, and a bias towards music or specialist usage identified. Blendwave was presented as a middle ground alternative between popular sound effect creators such as sfxr and the traditional sample-based workflow employed by specialists through the usage of DAWs and Patchers. With it, we expect to contribute towards democratizing the sound design process in the same way that music samplers did for artists during the birth of the remix culture.

As we approach the final stages of the first prototype implementation, we prepare to perform usability tests among audiovisual authors and collect quantitative data via web analytics tools in order evaluate and subsequently evolve the tool by following an iterative enhancement paradigm.

REFERENCES

[1] Viers, R., 2008. The sound effects bible: how to create and record Hollywood style sound effects. Michael Wiese Productions, Studio City, CA.
[2] Opolski, D.R., 2013. Introdução ao desenho de som: uma sistematização aplicada na análise do longa-metragem Ensaio sobre a cegueira, Coleção Humanidades. Editora UFPB, João Pessoa.
[3] Magnusson, T., 2009. Of Epistemic Tools: musical instruments as cognitive extensions. In Organised Sound 14, 168. DOI = 10.1017/S1355771809000272.
[4] Pluckette, M., 1988. The Patcher. In Proceedings of International Computer Music Conference 420–429.
[5] Rogers, C., 2011. Web Audio API - https://www.w3.org/TR/2011/WD-webaudio-20111215/ 04/17/2017.
[6] Can I Use Webaudio? http://caniuse.com/#search=web%20audio accessed 04/17/2017.
[7] Wyse, L., Subramanian, S., 2013. The viability of the web browser as a computer music platform. Computer Music Journal 37, 10–23.
[8] Web Audio Weekly - http://www.webaudioweekly.com accessed 04/14/2017.
[9] WASYN-1 Web Audio Synth - http://www.errozero.co.uk/stuff/poly accessed 04/12/2017.
[10] Orbit Sampler Music App - http://www.dreampipe.io/orbit/ accessed 04/12/2017.
[11] BlokDust - http://blokdust.com/ accessed 04/12/2017
[12] Lissajouss - http://lissajousjs.com accessed 04/13/2017
[13] Femur - The creative work of musician & developer, Luke Phillips - https://www.femurdesign.com/ accessed 04/12/2017.
[14] sfxr – http://www.drpetter.se/project_sfxr.html accessed 04/12/2017.
[15] Tools | Ludum Dare - http://ludumdare.com/compo/tools/ accessed 04/13/2017.
[16] BFXR - http://www.bfxr.net accessed 04/13/2017.
[17] jfxr - http://jfxr.frozenfractal.com accessed 04/13/2017.
[18] Candy, L., 2006. Practice based research: A guide. CCS Report 1, 1–19.
[19] Sielis, G., Papadopoulos, G., & Granić, A. 2011. Design of a multi-interface creativity support tool for the enhancement of the creativity process. In Universal Access in Human-Computer Interaction. Applications and Services, 632–641.
[20] Basil, V.R., Turner, A.J., 1975. Iterative enhancement: A practical technique for software development. IEEE Transactions on Software Engineering 390–396.
[21] How the Outstanding Sound for 'Stranger Things' is Made - https://www.asoundeffect.com/stranger-things-sound/ accessed 04/17/2017.
[22] Navas, E., 2012. Remix theory: the aesthetics of sampling. Springer, New York.
[23] Pizzicato.js – https://alemangui.github.io/pizzicato/ accessed 04/17/2017