Open band: Audience Creative Participation Using Web Audio Synthesis

Ariane Stolfi
Universidade de São Paulo
05508-020 São Paulo
arianestolfi@gmail.com

Mathieu Barthet
Queen Mary University of London
E14NS Mile End Road
m.barthet@qmul.ac.uk

Fábio Goródscy
Universidade de São Paulo
05508-090 São Paulo
fabiogorodscy@gmail.com

Antonio Deusany de Carvalho Junior
Universidade de São Paulo
05508-090 São Paulo
dj@ime.usp.br

Fernando Iazzetta
Universidade de São Paulo
05508-020 São Paulo
iazzetta@usp.br

ABSTRACT

This work investigates a web-based open environment enabling collaborative music experiences. We propose an artifact, Open Band, which enables collective sound dialogues in a web "agora", blurring the limits between audience and performers. The systems relies on a multi-user chat system where textual inputs are translated to sounds. We depart from individual music playing experiences in favor of creative participation in networked music making. A previous implementation associated typed letters to pre-composed samples. We present and discuss in this paper a novel instance of our system which operates using Web Audio synthesis.

1. INTRODUCTION

Recent advances in web audio and open source technologies provide new grounds to reconfigure interactive audio art. In this work we propose a web-based platform for musical creative participation bridging the gap between audience and performers. We present an implementation, Open Band, where participants can play sounds in response to textual messages entered in an on-line chat. Our implementation is based on web audio technologies since they support the development of accessible audio interfaces [25], are ubiquitous and easy to distribute [18]. Furthermore they can easily be applied to design interactive systems, independent of any extra software installation [31].

We are interested in creating "open" works, in the sense advanced by Umberto Eco [9], were the final arrangement of the piece is given to publics to decide. We use music as our domain as music is known for its social functions of cohesion and communication [16]. Our web-based instrument has been used to create experimental participatory music performances where everyone in the audience can join through its cell phone, tablet or laptop, and can playing through typing.

The first version that we developed converted letters into sounds using audio samples. In the version presented in this paper, we are building a piece entirely based on web audio synthesis to reduce bandwidth requirements for the audience participation, and also to experiments with possibilities given by the web audio API.

Web Audio Conference WAC-2017, August 21–23, 2017, London, UK.

2. RELATED WORKS

Participatory art and Eco's Open Works.

Contemporary Western music performances are predominantly presentational, with performers preparing and providing music for another group, the audience [27]. In contrast to presentational performances, which intrinsically create a divide between audience and performers, participatory performances [15] seek to blur the boundaries between audiences and performers by giving equal role to participants, building up on the notion of "communitas" (unstructured communities in which all members are equal allowing them to share a common experience). Such aspiration from participatory art corroborates Eco's conceptions of "open works", which, according to Robey in [8], require of the public *"a much greater degree of collaboration and personal involvement than was ever required by the traditional art of the past."*. In Open Works, it is *"the artist's decision to leave the arrangement of some of their constituents either to the public or to chance, thus giving them not a single definitive order but a multiplicity of possible orders"*. These motivations interestingly resonate with the contemporary perception of audience's role changing from primarily passive to one "co-creating values" [23], with audiences who increasingly want to "shape" their own experiences. Another important rationale of participatory art is to improve audience cognitive engagement in performance through an active form spectatorship involving a physical engagement in the performance [15].

Participatory music performances with Web technologies.

Human-computer interaction (HCI) and communication technologies provide great potential to facilitate participatory art forms in our digital age as they can be used to mediate and transform creative information remotely, in (quasi) real-time and scalable ways. Within the field of digital participatory art, several authors investigated how to create participatory music performances using web technologies (see [30] for a review of some of them). Web applications for participatory performances have been proposed e.g. in [28, 10, 32, 3, 17], each of these studies manifesting specific artistic and audience creative agency models. The massMobile framework [29] was used to let audiences control stage lighting effects or create musical loops from instructions provided through a graphical user interface (GUI) on their smartphones. Mood Conductor [10] also relies on the use of smartphones to enable audience members to conduct performers in terms of emotional directions. In Open Symphony [32, 30], audience members can generate live graphic scores interpreted by performers by voting for music playing modes on their smartphones. In this work and others, such as A.bel [3] and the CoSiMa framework [17], smartphones are employed both as a sound control and diffusion interface leveraging the loudspeakers embedded in the devices.

3. OPEN BAND DESIGN

During the development of this new version we opted to evaluate some ready-made solutions for web audio synthesis like Gibber [24], Waax [2], and meSpeak.js [1]. These solutions are able to facilitate the use of web audio technology and other browser facilities, but also create barriers and sometimes limitations that weren't imposed by pure web audio. Furthermore, sometimes frameworks could offer more than what was required by the application, which can lead to more confusion during development. We tried to use, meSpeak.js as a simple and interesting text-to-speech library but the resulting sound was not musically satisfactory, and we couldn't make messages play one on top of each other, due to limitations of the framework, so we choose to focus on pure web audio for sound programming.

We also decided to follow some aesthetic concepts based on type design [26] to propose "an audio typograph", and also to experiment with new kind of music, avoiding traditional chords and scales. These concepts surrounded the whole development of this current version of Open Band and had a great impact in the new result obtained.

3.1 Genesis

The phonetic alphabet was as pointed by McLuhan [21], a fundamental technology for the development of our culture, being easy to learn and adjustable to many languages. Here we are making use of the alphabet as a technology to produce experimental music. Since text messages are one of the main forms of communication on smartphones today, we choose to experiment possibilities to experiment with type as input to produce interactive music.

Interfaces using typing as input are used in various applications such as live coding [4], audience participation pieces, to receive feedback [2], to get user data as a source for algorithmic composition [5], or using keyboard as music controllers [11]. In this project, the interaction process happens through an open multi-user chat interface (see Figure 4), where there is no distinction between users. No one can tell who is playing what and each message sent to the chat is loaded into a chat room and is played back as a sequence of sounds. In an inter-semiotic translation [22], defined by Roman Jakobson as 'transmutation of signs', [6], we decode the text messages in musical information, and all messages are played as they are sent to the chat. In this particular version, we are also using a concept of audio typography as base to this inter-semiotic translation, to decode letters into sounds, building spectral designed "letters" made of predetermined sound blocks. Trough this path, we are abstracting the aural aspects of the typed characters and grabbing only the shape of the letters as information for the sound synthesis.

[1]meSpeak.js website: http://www.masswerk.at/mespeak/
[2]FLOrchestra at CMMR http://www.crisap.org/event/in-transglasphone/

3.2 Audio Typography

The effect of typography on written texts acted as a motivation in the present artifact to experiment with ways in which timbre could affect the sound produced by musicians. We use additive and noise synthesis to sonify "drawings" of the shapes of the letters.

3.2.1 Metafont

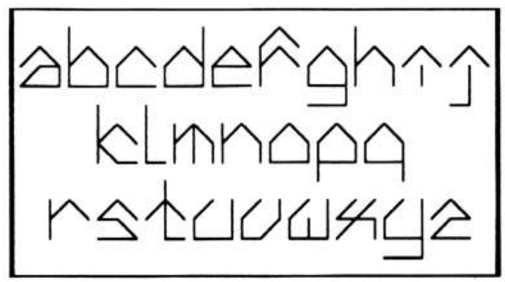

Figure 1: One of the modular alphabets proposed by Douglas Hoefstader in the book Metamagical Themas, page 90

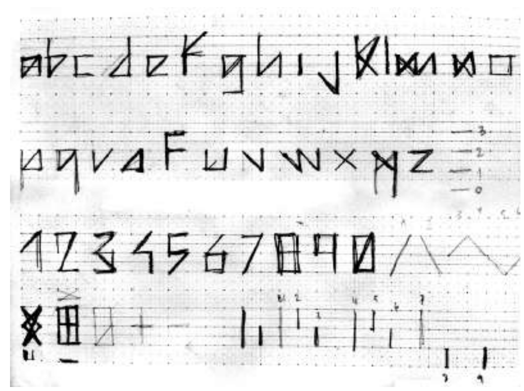

Figure 2: Sketch for the modular font to be base for audio synthesis.

Donald Knuth, a computer scientist and artist who worked on typesetting issues, developed the concept of a "meta-font", a typeface that was not statically drawn, but that could be changed trough typographical parameters[7]. Following the rules of meta-fonts, many different typefaces could be generated simply by varying the type parameters. As says Douglas Hoefstader in [14], *"Knuth's purpose is not to give the ultimate parametrization of the letters of the alphabet (indeed, I suspect that he would be the first to laugh at the very notion), but to allow a user to make "knobbed letters" – we could call them letter schemas. This means* *that you can choose for yourself what the variable aspects of a letter are, and then, with Metafont's aid, you can easily construct knobs that allow those aspects to vary.".*

> Knuth's purpose is not to give the ultimate parametrization of the letters of the alphabet (indeed, I suspect that he would be the first to laugh at the very notion), but to allow a user to make "knobbed letters" – we could call them letter schemas. This means that you can choose for yourself what the variable aspects of a letter are, and then, with Metafont's aid, you can easily construct knobs that allow those aspects to vary [14].

We used the basic parameters defined by Knuth [14] such as x-height, baseline, height of ascendants and descendants, as measures to establish frequencies for tone sounds (one low, one at the baseline, one at the x-height an one at the uppercase line of each letter). The horizontal parameters of the type are translated into temporal measures, to determine duration of the sound events and intervals between the letters.

A meta-font is a complex font with several typographic parameters. In this work, we are using a simpler analogy by proposing a modular type, similar to the ones Douglas Hoefstader [14] proposes in his book "Metamagical Themas", based on simple grid structures (see Figure 1). We propose a model that can work only with a few building blocks (see Figure2 in which they are sketched). The line of the letters was mapped into two basic types of functions: noise synthesis for the verticals and solenoids for the horizontal lines and glissandi, as defined by the composer project in Figure 3.

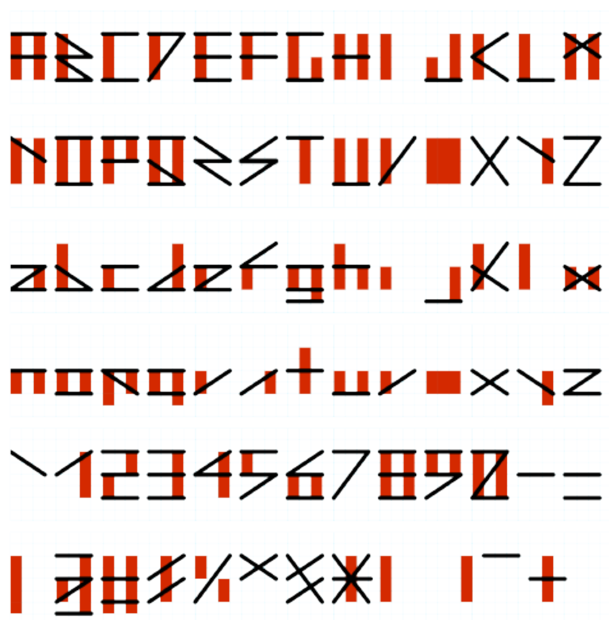

Figure 3: Project for audio typography, made of blocks of noise, sine waves and glissandi. The red blocks represent noise and the black lines solenoids and glissandi.

3.3 Project Development

This project started off as open source software in 2016. Although its applications are oriented towards the artistic domain, we have been using agile design methodologies principles generally used for products. We namely use the Lean Ux [20] methodology that includes early user evaluation, collaborative cross-functional design, solving use case problems, and nimble design.

Figure 4: Interface of the chat on mobile device

3.3.1 Front End

The interaction takes place trough a webchat interface an illustration of which is given in Figure 4. When users enter the site, they can type messages in a text box, with no distinction between past and present users. Once the messages are received they are played as a sequence of preprogrammed sounds. The composer and programmers decided to keep the interface is minimalistic for aesthetic reasons and in an attempt to follow a brutalist approach as it stands in architecture.

3.3.2 Special Commands

Besides sounds generated from people's interactions, there are also commands that can be live coded. Those commands are not shown in the public interface, for the conductor maintain certain level of control on the performances. They can change the audio frequencies, the global volumes or turn off the sound, for example. The commands follow a command line like program paradigm, which means that there are special words and characters that will run special actions, like programs. This kind of feature is also present in many of modern chat-like applications, so we expect this to be intuitive for users. Also, we have an administration interface with pre-programmed frequencies and intervals, that we can use to control sound flow and rhythm in real time.

3.3.3 Network Architecture

The project was built having two server components that are completely independent, one is a typical web server which is responsible to serve web content such as HTML, JavaScript and audio files, based on the Web Audio API [1] and the other is responsible for making messages deliverable (see Figure 5). This provides flexibility to the system, as it allows the messaging server to be replaced by any framework that supplies what is required by the application.

3.3.4 Back-end

In the current implementation, the messaging server is written in Ruby, making use of the Puma, Sinatra and Faye frameworks, as well as WebSockets for the server/browser communication. The server [3] tries to be easily reproducible on Linux platforms and is released open source. The messaging technology plays a main role in such application and early versions of the project frameworks like PubNub and Peer.js have been tested. The decision of keeping a Ruby server was made having in mind that it was important being able to make the performances even without any Internet access, what couldn't be made with these frameworks.

3.3.5 Front-end

In the first version, all the sounds are played as samples. In this case we have one sample for each ASCII character, and the mp3 files are named each with its ASCII number. This one-to-one mapping was chosen to make the system more recognizable by its players, as results can always be expected. Sounds are played one after another, as indicated by the sequence of the letters on the messages. The samples are rhythmically played through values that are determined by the code and that can be changed through special commands in real time. In the proposed version, sounds are all synthesized in the browser. This way there is no need for transferring files at the beginning of a user's participation. The stream of data is reduced and more parameters can be changed during a live performance.

3.3.6 Performance Set-up

The performances are designed to be made in any space that can fit a laptop and a network hub. Participants can join the network with their devices and connect to the the application in a web browser. As nowadays' web browsers are ubiquitous, the only limitation for connecting to the application is the device being able to join the network through WiFi. This way, the requirements are quite minimal for the audience. We usually use a projector to show the messages for everybody, so even who is not connected to the application can enjoy the action.

A regular laptop should be able to run the application without any issues. The last performances were made on a laptop with AMD A4-5000 processor and 4GB memory. Any devices that can install Ruby gems should be able to host the server but it was created and developed for Linux.

3.3.7 Audio Synthesis

In the first version, all the audio processing was made through sampling, which caused a long time for downloading the samples from the server when the user entered the website. Also, the samples had to be cut one by one, resulting in a large amount of time to prepare material for playing.

[3] Server code: https://github.com/fabiogoro/bandaserver

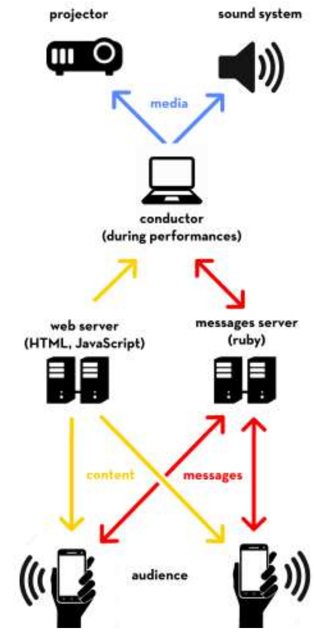

Figure 5: Servers Schema

The new version relies entirely on web audio synthesis, using the Web Audio API to generate sounds on users' devices. There could be many forms to do the mapping between letters and sounds, such as note/keyboard association, that used in many applications such as DAW software, or even speech synthesis mechanisms. The present artifact draws on the idea of "audio typography". We are drawing spectral "letters" made of modules of audio functions, that work as temporal semiotic Units (TSU) [13]; such as noise blocks, sine waves and glissandi. Now there are two types of basic functions that work as building blocks, as shown in Figures 6 and 7, one that uses FFT synthesis to build the blocks of noise based on variable values of tempo and base frequencies, using the framework Noisy.js [4], witch was written specially for this project, but can be used to other web-based music projects in the future.

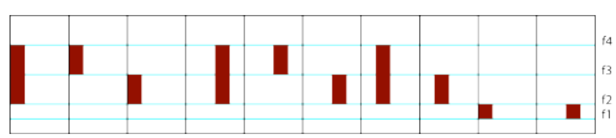

Figure 6: Functions for playing vertical blocks with noise synthesis

Figure 7: Functions for playing horizontal and diagonal lines with oscillators

Sine function has one oscillator buffer that can have static frequencies or linearly ramping frequencies. This creates diagonal and horizontal lines in the spectrum. Then, frequencies are defined as the bottom and top limit, some steps are defined in the middle. The length of spectral letters are as

[4] Source code: https://github.com/fabiogoro/noisy

well defined. By the end, letters are translated as a group of functions, and the result is played one after another. We built 20 predetermined functions to correspond at each "type block", and each letter plays as many functions as are determined by the typeface project. Thus, we created a system of mapping letters to the functions. Every output node goes through an ADSR class [19] that creates a more natural attack/release, and also goes through a gain node, that can be changed during a performance. On Figure 8 is shown a spectrogram generated by the application with the results obtained.

Figure 8: Spectrograms generated by the application

3.3.8 Challenges

As every letter from every message is translated into a group of playing Web Audio nodes, the number of oscillators that plays simultaneously can grow really quickly depending on the number of participants sending messages. As this application was intended to play in many different mobile devices, this led to the necessity of limiting the creation of nodes, as this could take more processing than some devices could take, and in some cases could make the device stop playing, which is undesirable behavior for such application. We have also met differences of implementations from web audio synthesis in different browsers which led to unexpected behavior between different executions. As an example, the way time is used in linearRampToValue in Safari and Firefox browsers is different from Chrome, and iOS devices request special actions for playing sounds in browsers. For now the system is working correctly on recent versions of Google Chrome, Mozilla Firefox, Safari and Opera browsers.

4. PERFORMANCES EVALUATION

We have presented the first version of Open Band as performance running the server in a local computer and a closed WiFi network, for reducing latency and keep the public together in a kind of web"agora". It has been performed in different occasions, in concerts and private spaces, most of them with a projection of e online chat on a shared screen. Depending on the audience's profile, the exchanges between participants could go different way. We observed that in musical contexts, the audience keeps more time experiencing with the samples and rhythm, typing meaningless phrases, and when the public is younger, the participants tend to play with the ability to speak anonymously. As the conversation "warms" up, layers of sounds are overlapped, turning the rhythm into something more frenetic.

Like in a real conversation, if people don't take time to listen to the others, the sound becomes more difficult to distinguish. We received a lot of feedback from the public on the chat, with people asking things about the project and supporting it.

One problem of the previous version was that been based on heavy volume of data from samples, we had always to create a local network and even with that we had a long time of download before playing. With the present version relying only on web audio synthesis, the size of the project reduced dramatically — from 60 megabytes to kilobytes — and also the downloading time when compared with the previous version. This new aspect of the Open Band facilitates users to participate using 4G data plan as the data consumption is now comprised of text messages only as no samples files are downloaded.

It is important to emphasize that the browser used by the participants can make a real difference on the performance itself. Most current browsers do enable web audio synthesis, however they are incompatible with many parameters. That said, the participants are recommended to use Google Chrome and take advantage of its full compatibility with current web audio standards.

Figure 9: people interacting within the piece

5. CONCLUSIONS AND FUTURE WORK

Free interaction and easiness to use bring a high degree of fun to the activity of making music with web audio applications, as we have seen in some of performances. It does make play music closer to the sense of playing that is also related to games, but, in this case, maintaining a socio-political standpoint, offering freedom of speech, and allowing individuals to interact without been pointed at, granting people to express personal ideologies without constraints. Besides this, since the messages are decoded into music, there is also a possibility of communication beyond symbolic written language, with the sharing of purely sound phrases, without any literal sense.

There are now two working versions of the project, one with fixed collections of samples, and the other with web audio synthesis, both of them with preprogrammed sounds for each letter. We seek in the future to expand it to be a framework to be used in different musical contexts. One of our goals is to integrate the project with the audio Commons API [12] to foster media re-purposing, creating an interface for uploading and mapping sounds to the letters. For the web audio synthesis version, we aim to develop form to control the base frequencies and rhythm trough gestural interfaces, to enhance the musical possibilities given to the audience.

6. REFERENCES

[1] P. Adenot, C. Wilson, and C. Rogers. Web audio api. https://www.w3.org/TR/webaudio/, 2017. Accessed March, 2017.

[2] H. Choi and J. Berger. Waax: Web audio API extension. In *Proceedings of the International Conference on New Interfaces for Musical Expression*,

pages 499–502, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.
[3] A. Clément, F. Ribeiro, R. Rodrigues, and R. Penha. Bridging the gap between performers and the audience using networked smartphones: the a.bel system. In *Proceedings of International Conference on Live Interfaces*, 2016.
[4] N. Collins, A. McLean, J. Rohrhuber, and A. Ward. Live coding in laptop performance. *Organised Sound*, 8(03):321–330, 2003.
[5] L. Dahl, J. Herrera, and C. Wilkerson. Tweetdreams : Making music with the audience and the world using real-time twitter data. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 272–275, Oslo, Norway, 2011.
[6] Daniella Aquiar and João Queiroz. Towrds a Model of Semiotic translation. *The Internationa Journal of Arts in Society*, 4, 2009.
[7] Donald E. Knuth. The Concept of Meta-Font. *Visble Language*, XVI:3–27, 1982.
[8] U. Eco. *The Open Work (Translated by Anna Cancogni With an Introduction by David Robey).* Harvard University Press, Cambridge, Massachusetts, USA, 1989.
[9] U. Eco. The poetics of the open work. 1962. *Participation: Documents of Contemporary Art*, pages 20–40, 2006.
[10] G. Fazekas, M. Barthet, and M. B. Sandler. The Mood Conductor system: Audience and performer interaction using mobile technology and emotion cues. In *Proc. of the Int. Symposium on Computer Music Multidisciplinary Research (CMMR)*, 2013.
[11] R. Fiebrink, G. Wang, and P. R. Cook. Don't forget the laptop : Using native input capabilities for expressive musical control. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 164–167, New York City, NY, United States, 2007.
[12] F. e. a. Font. Audio commons: Bringing creative commons audio content to the creative industries. In *Proceedings of Audio Engineering Society Conference: 61st International Conference: Audio for Games*. Audio Engineering Society, 2016.
[13] X. Hautbois. Temporal semiotic units (tsus), a very short introduction. *MiM.*, 29, 2013.
[14] D. Hofstadter. *Metamagical Themas: Questing For The Essence Of Mind And Pattern*. Basic Books, 1985.
[15] S. Kattwinkel. *Audience participation: Essays on inclusion in performance*. Number 101. Greenwood Publishing Group, 2003.
[16] S. Koelsch. Brain correlates of music-evoked emotions. *Nat Rev Neurosci*, 15(3):170–180, 03 2014.
[17] J.-P. Lambert, S. Robaszkiewicz, and N. Schnell. Synchronisation for distributed audio rendering over heterogeneous devices, in html5. In *Proc. of Web Audio Conference (WAC)*, Atlanta, USA, 2016.
[18] S. Lee and G. Essl. Web-based temporal typography for musical expression and performance. In E. Berdahl and J. Allison, editors, *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 65–69, Baton Rouge, Louisiana, USA, May 2015. Louisiana State University.
[19] S. W. Lee, A. D. de Carvalho Jr, and G. Essl. Crowd in c [loud]: Audience participation music with online dating metaphor using cloud service. In *Web Audio Conference*, Atlanta, Georgia, USA, 2016. Georgia Institute of Technology.
[20] L. A. Liikkanen, H. Kilpiö, L. Svan, and M. Hiltunen. Lean UX: The Next Generation of User-centered Agile Development? In *Proceedings of the 8th Nordic Conference on Human-Computer Interaction: Fun, Fast, Foundational*, NordiCHI '14, pages 1095–1100, New York, NY, USA, 2014. ACM.
[21] M. McLuhan. *Understanding media: The extensions of man*. MIT press, 1994.
[22] J. Plaza. *Tradução intersemiótica*, volume 93. Editora Perspectiva, 1987.
[23] J. Radbourne, K. Johanson, H. Glow, and T. White. The audience experience: measuring quality in the performing arts. *International Journal of Arts Management*, 11(3):16–29, 2009.
[24] C. Roberts and J. Kuchera-Morin. Gibber: Live coding audio in the browser. In *Proceedings of the International Computer Music Conference (ICMC)*, Ljubljana, Slovenia, 2012. IRZU - the Institute for Sonic Arts Research, International Computer Music Association.
[25] C. Roberts, G. Wakefield, and M. Wright. The web browser as synthesizer and interface. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 313–318, Daejeon, Republic of Korea, May 2013. Graduate School of Culture Technology, KAIST.
[26] E. Ruder. *Typography: A Manual of Design: A Textbook of Design*. Verlag Niggli, Niederteufen, 4th edition edition, Jan. 2009.
[27] T. Turino. *Music as Social Life: The Politics Of Participation*. University of Chicago Press, 2008.
[28] N. Weitzner, J. Freeman, S. Garrett, and Y.-L. Chen. massmobile -an audience participation framework. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012. University of Michigan.
[29] N. Weitzner, J. Freeman, S. Garrett, and Y.-L. Chen. massmobile-an audience participation framework. In *Proceedings of the international conference on New Interfaces for Musical Expression*, pages 21–23, 2012.
[30] Y. Wu, L. Zhang, N. Bryan-Kinns, and M. Barthet. Open symphony: Creative participation for audiences of live music performances. *IEEE Multimedia Magazine*, (48-62), 2017.
[31] L. Wyse and S. Subramanian. The viability of the web browser as a computer music platform. *Computer Music Journal*, 37(4):10–23, 2013.
[32] L. Zhang, Y. Wu, and M. Barthet. A web application for audience participation in live music performance: The Open Symphony use case. In *Proc. of International Conference on New Interfaces for Musical Expression (NIME)*, 2016.