Sen, Sourya; Tahiroğlu, Koray; Lohman, Julia

Sounding Brush: A Tablet based Musical Instrument for Drawing and Mark Making.

Published in:
Proceedings of the International Conference on New Interfaces for Musical Expression

Published: 25/07/2020

Document Version
Publisher's PDF, also known as Version of record

Please cite the original version:
Sen, S., Tahiroğlu, K., & Lohman, J. (2020). Sounding Brush: A Tablet based Musical Instrument for Drawing and Mark Making. In *Proceedings of the International Conference on New Interfaces for Musical Expression* (pp. 331-336). (Proceedings of the International Conference on New Interfaces for Musical Expression). International Conference on New Interfaces for Musical Expression (NIME).
https://www.nime.org/proceedings/2020/nime2020_paper64.pdf

Sounding Brush: A Tablet based Musical Instrument for Drawing and Mark Making

Sourya Sen
Department of Media
Aalto University
School of ARTS
FI 00076 AALTO Finland
ayruos@gmail.com

Koray Tahiroğlu
Department of Media
Aalto University
School of ARTS
FI 00076 AALTO Finland
koray.tahiroglu@aalto.fi

Julia Lohmann
Department of Design
Aalto University
School of ARTS
FI 00076 AALTO Finland
julia.lohmann@aalto.fi

ABSTRACT

Existing applications of mobile music tools are often concerned with the simulation of acoustic or digital musical instruments, extended with graphical representations of keys, pads, etc. Following an intensive review of existing tools and approaches to mobile music making, we implemented a digital drawing tool, employing a time-based graphical/gestural interface for music composition and performance. In this paper, we introduce our Sounding Brush project, through which we explore music making in various forms with the natural gestures of drawing and mark making on a tablet device. Subsequently, we present the design and development of the Sounding Brush application. Utilising this project idea, we discuss the act of drawing as an activity that is not separated from the act of playing musical instrument. Drawing is essentially the act of playing music by means of a continuous process of observation, individualisation and exploring time and space in a unique way.

Author Keywords

NIME, mobile music, mobile app, Sounding Brush, act of drawing

CCS Concepts

•Applied computing → Sound and music computing; Performing arts;

1. INTRODUCTION

Musical instruments, and the field of electronic music, is highly tied to its surrounding technological landscape. In recent years, that has led to the development of digital musical instruments (DMIs) and research into new forms of interaction, performance and composition. DMIs can be defined as digital sound generators with an associative but possibly dissociable control interface bound together through parameters which employ various mapping strategies [16, 7]. Through the employment of computers and portable computing devices, different forms of interaction, both spatial and gestural, have been employed as control surfaces for musical instruments and in turn, has led to various forms of performances, compositions and expressions [4, 5, 7]. However, when it comes to the interface itself, specially in the case when using graphical user interfaces (GUIs), the common strategy employed revolves around digital representations of physical controls, such as buttons and faders, and the use of a linear timeline for time-based events. The on screen interface is rarely a part of a performance and usually only serves as a control surface to the composer or performer. Touch capable computing devices like tablets are also powerful digital drawing and mark making tools. Combined with the availability of styluses alongside flexible drawing focused applications, they are commonly used by illustrators and graphic designers. These applications are purely for drawing and they are used for creating traditional two dimensional drawings and illustrations. In relation to building musical instruments, Zbyszynski [19] notes the gestural advantages a tablet tool entails to both the audience and the performer with any action and its sonic reaction immediately understandable.

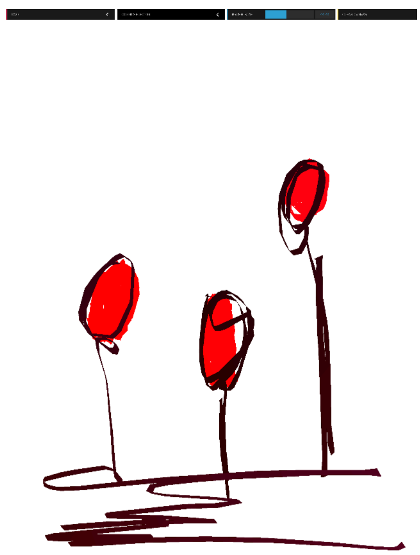

Figure 1: Drawing applied to Sounding Brush

When it comes to the direct and indirect relationships between visuals and music, it has been a matter of interest for many. Outside of the realm of moving image, commonly discussed under the umbrella term of visual music, illustrations and graphic design for musical ideas has been explored by various composers through graphic scores or graphic notations. Graphic score is the use of graphical representations of musical forms which are not related to any standardised notation systems [6]. Being purely personal graphical expressions of musical ideas, these do not necessarily form an evolving time-based musical performance as they are static in nature and represent the overall musical piece despite its temporal dimension.

NIME'20, July 21-25, 2020, Royal Birmingham Conservatoire, Birmingham City University, Birmingham, United Kingdom.

In this paper, we present Sounding Brush application, a tablet based musical instrument that merges with with digital drawing tools, employing a time based graphical gestural interface for music composition and performance (Figure 1). This tablet application abstracts the natural process of drawing as the primary control mechanism of different sound synthesis engines through various temporal, spatial, and gestural mapping strategies. Sounding Brush provides a platform for both musical expression and time based graphic notations with the visuals becoming a part of the performance itself. In turn, this helps us in exploring alternative ways of playing music with a tablet. This paper is divided into sections and covers the state of the art, the design and implementation of the instrument, and an overall discussion of the implications of this application. Finally, it also provides a brief conclusion and future plans for the project.

2. STATE OF THE ART

Computationally powerful handheld mobile device became a popular platform for music creation and performance applications, and has given rise to the field of mobile music. Gaye et al. [9] define mobile music as a term covering musical activities using portable devices without constraints of any particular physical location. These musical activities can cover a wide domain of applications [9]. While Kell and Wanderley [10] defined ten main categories in commercially available mobile music applications, when it comes to music creation, composition, and performance applications, these can be further simplified into three broader groups.

First of all, complete digital audio workstations (DAWs) like Garageband and FL Studio are available for such devices and offer similar functionality as their desktop versions. These applications provide a complete environment to produce music, with multiple tracks, virtual instruments, ability to record and compose loops, etc. The similar idea is also extended towards DJing applications on these kind of devices. However, while there are some additional affordances that are granted in the use of such applications on a tablet as compared to their desktop counterparts, essentially they are very similar and follow similar workflows for music productions. This is also evident within newer applications like Medly [1] and Auxy [2] which do not have a history in Desktop counterparts.

The second kind of musical instruments that are available can be described as virtual emulations of real world instruments. These include multiple types of pianos, string instruments, percussion instruments and even synthesisers. These usually take advantage of the multitouch capabilities of the tablet computer, offering a tactile interface missing within desktop environments. Both these classes, therefore, are based on "known metaphors" and "emulations of electronic music interfaces" as concluded by Kell and Wenderley [10] within their classification. Emulating real world instruments within mobile devices have often been criticised [15] and have failed to meet users expectations [14]. On the other hand, treating mobile devices in their own right and taking advantage of the affordances of such platforms tend to have a more positive reaction from users [15].

Therefore, it is inevitable that truly unique digital musical instruments also exist within this landscape. In general mobile applications offer interactions and interfaces which are far removed from any existing instrument, taking advantage of the technologies available on these devices while also recontextualizing traditional ideas of collaboration, performance and social interaction. For example, Smule's Ocarina [17] leverages both the multitouch capabilities as well as the microphone input, GPS, etc. Applications and projects like echobo [11], Daisyphone [2] and PESI [13] open up opportunities for audience participation, remote access collaborations and spatial and embodied explorations. Other applications, taking the advantage of the hardware and implementing interfaces that are not based on traditional instruments or desktop software, are also increasingly common. Playground app "reframes music making as a fluid, gestural process, making it easy for people who are intimidated by musical scales to create impressive compositions" [1] and offers an interface and experience that is truly unique to the platform. Within this class of applications, as the strategies implemented vary widely, it is hard to define common metaphors that govern their functions. However, a common theme of playfulness and simplifying traditional control schemes can be noticed.

Exploring relationships between drawing and sound characteristics have a long history. In the digital environment, the earliest examples can be seen in the work of John Whitney [18] and Daphne Oram [12]. While Whitney was exploring translation of audio frequencies to graphic material, Oram modeled the Oramics Machine to synthesise sound based on user drawings. Most of these examples, however, are approaches towards visualising sound or synthesising sound through graphical input and explore a one-to-one connection in the time or frequency domain. Within the field of personal computing, the Kid Pix application (see [3]) is one of the earliest sonified drawing applications. In recent years, with the popularity of tablet devices, there have been various implementations of interactive music applications that use drawing or drawing-like gestures as an input method for musical applications. For instance, Fluxpad [3] offers a drawing based interface for composing music. Nevertheless, Fluxpad still follows a looping sequencer based approach using samples and drawing is only used to insert note and gate information within this sequencer. A common theme, however, seem to be in the usage of either drawing as an input method in a simplified musical interface context or treating sound as effects within primarily drawing applications.

Within this context, we approached Sounding Brush from both audio and visual directions, giving equal importance to drawing and sound synthesis. We wanted the application to be both a competent drawing application as well as a flexible musical interface. Mark making and sound synthesis would be mapped to each other but not in a way that it would be strictly visualising sound or sonifying visuals. Both are mapped together in various ways through alternative strategies - such as, the user can draw in different directions or with different stroke lengths for different outcomes. From an interface perspective, we also aimed to present the tablet as a drawing surface and not a musical instrument emulating known interfaces or based on known metaphors. The act of drawing itself would define the time-based nature of sound and music, both in continuous and discrete forms, building up sequences, events and soundscapes through the process in a two-dimensional space.

3. DESIGN AND IMPLEMENTATION

Sounding Brush is implemented as an iOS application, aimed for use on an iPad, using open source development frameworks openFrameworks, Pure Data and ofxPd. OpenFrameworks is used to develop the host application as well as for handling all the drawing functions, Pure Data is used for the audio synthesis modules which run within the main ap-

[1]https://medlylabs.com
[2]https://auxy.co
[3]https://mominstruments.com/fluxpad/

plication and the two communicate with each other using libPd embedded openFrameworks addon ofxPd. In terms of the software architecture, it has been developed with modularity in mind so that it can be extended further in the future. A few different discrete components make up the application and in order of execution, they are, the sound synthesis module, the mesh generation algorithm and two stages of shaders for colouring and post-processing the generated mesh. Once the user starts drawing, a Pure Data patch is executed and a mesh is generated based on the touch inputs. As the user keeps drawing, parameters in the sound synthesis engine and the mesh are updated in real time while the shaders provide the final colour and texture of the stroke. Each of these elements can be easily assigned and reassigned within the source code. Within the application interface, the user is free to choose among a few different brushes, each having colour and character parameters which affect both the look and the sound depending on the brush type and the mapping strategies implemented within each brush. A notable exception is the *Gesture brush* which does not provide any mesh generation or drawing facility, but use the sensor inputs of the device to produce sound. Furthermore, certain memory management strategies have been implemented for smooth performance as well as three different types of erasing options for added flexibility towards building up an audiovisual canvas.

Figure 2: Across brush - pen like mesh and shader

A common concern within most audiovisual platforms is the issue of parameter mapping between the two domains. Our approach involved a number of sound synthesis engines parameterised through the action of drawing. With the possibility of each sound synthesis algorithm requiring different parameters for initialisation and propagation, it was important to have a generalised approach for dealing with this issue. Certain features of the drawing are identified and updated through the act of drawing, namely, the length of the stroke, the position of the last inserted vertex, the variance and standard deviation of the points in the two axes, the colour of the stroke as chosen by the user, etc. Parameters of these features are then communicated to the Pure Data patch on a brush-by-brush basis during both initialisation, drawing and finishing the process of the drawing instance. Within each Pure Data patch, the incoming parameters are mapped to various features of the sound synthesis algorithm discussed below. Additionally, certain audio features are also calculated within some of the patches, like overall amplitude for example, and sent back to the instance of the stroke for visual feedback and helps towards building a dynamic audiovisual brush which can evolve over time.

The design and development of the application followed an iterative development process. Additionally, as our team had members from various disciplines, there were various rounds of discussion and feedback sessions from each that contributed to the current version of the application. Working within a multidisciplinary team with expertise over various domains helped towards addressing design components, namely, the drawing experience, the user interface, the synthesis engines and the overall usability, mapping and user experience for the application. With different vocabularies arising from the different backgrounds within the team, it was occasionally difficult to communicate ideas, provide feedback and discuss technical limitations. Overall, we did find common ground throughout the process. The difficulties often also led to alternative ways of approaching certain issues, which may not have been possible within a team consisting of members from the same field.

The brushes which are currently implemented in the application are called *Across, Line, Three Waves, Kar+Paint, Particles, Crackling*, and the aforementioned *Gesture*. Details of each are provided below.

3.1 Across

Across utilises the simplest of strategies in all aspects. Each instance of Across is a sinusoidal wave whose frequency is mapped with the horizontal coordinates of the tablet. While conceptually simple, this brush provides the user with implementing additive synthesis techniques. Additive synthesis is the process of creating rich timbres by combining different sinusoidal waves of differing frequencies. Figure 2 shows the visual features of the Across brush type, combined with a static, almost pen like mesh and shader combination. This brush allows the user to draw as they would regularly on any digital surface without any added implications to be aware of. In a performance setting, if the user draws a continuous stroke horizontally, the frequency of the sinusoidal wave updates to the last horizontal coordinate of the vertex of the stroke. Vertical coordinate of the stroke has no effect on the audio waveform. The colour settings affect the stroke colour and the character settings affect the stroke width, but are not mapped to any aspect of the sound. Finally, once the user has finished drawing the stroke, neither the stroke nor the sound fade away but rather remain visible and audible until the memory management limitations are met or are erased deliberately, allowing the user to build up both a drawing and a soundscape progressively.

Figure 3: Line brush - continuous waveform

Across was one of the earliest brushes that was imple-

Figure 4: Stroke colour in Three Waves brush

Figure 5: Kar+Paint brush - animated stroke texture

mented in the development process and initially served as a barebones example of the software architecture. However, through the development period, it has developed further to its current form and remained in the available list of brushes as its simplicity offers a good introduction to any user approaching this instrument. However, it is not so simple either as stacking up various instances of Across allows an easy way to build up complexity through additive synthesis techniques.

3.2 Line

Line is similar to Across with one marked difference - it is not a continuous waveform. As soon as the user finishes drawing a stroke (Figure 3), the sound generation decays. This was one of the last brushes that was added as through the development process the necessity for discrete sound events based on additive techniques was identified. As the visual stroke does not fade away, it also allows for adding details to the drawing without necessarily having to build up a continuous soundscape, as is the case with the Across brush. Therefore, in combination, both Across and Line allow the user to explore additive techniques while building up a drawing with discrete or continuous sound events.

3.3 Three Waves

Three Waves utilises subtractive synthesis with a decay and filter envelop mapped to both the coordinates of the screen as well as colour and character settings which mix different waveforms. Subtractive synthesis is the process through which a harmonically rich waveform is sculpted by filtering frequencies to shape a sound. Additionally, a noise based shader is utilised to provide a texture to the strokes that are drawn creating a rich spectrum in both audio and visual domains. A mix of a sawtooth wave, a pulse wave with a random pulse width, and a noise source are generated at initialisation. Their amplitudes mapped to the colour characteristics that the user selects the stroke colour shown in Figure 4. The coordinates of the stroke are used to determine the frequency as well as the filter cutoff parameters within the synthesis engine, each mapped to the horizontal and vertical positions of the last inserted vertex respectively. Finally, once the user finishes drawing, a decay envelop proportional to the length of the stroke is applied to the brush, making it decay to silence over time once the drawing is completed. The stroke, however, remains on screen. This brush was developed towards the second half of the the development process. As the latter brushes available to the user utilise somewhat advanced computer music techniques and algorithms, a simpler synthesis technique to complement Across was required to have a better balance within the drawing, synthesis and performance environment. However, it differs from Across as by employing subtractive synthesis as well as textured shaders, it is both aurally and visually richer than Across and Line.

3.4 Kar+Paint

Kar+Paint utilises a Karplus Strong algorithm. The Karplus Strong algorithm is a physical modelling technique that is used to generate string like sounds. This is the only brush which generates a sound only once the user has finished drawing, in the fashion that the user is building up tension and releasing it, as they would in a stringed instruments like a guitar. This brush also uses dynamic, animated shaders with bi-directional mapping between the graphic and sound, and the notes produced are quantised to a musical scale for playing pitched harmonies. Unlike the previous brushes, the two dimensional coordinates of the stroke have no impact on the frequency of the sound produced, instead, the length of the stroke and the speed of drawing determines its pitch related to the fundamental and the scale, which are both defined in the C++ source code. In practise, the user can draw the stroke visually and hold on to the touch for as long as they desire to add visual characteristics to the canvas. As soon as they let go, the sound is triggered. As the sound fades away, so does the visual while a texture animates within the stroke (Figure 5).

3.5 Particles

Particles is based on granular synthesis and uses a modified version of a patch provided as an example by Farnell [8]. Granular synthesis is a technique in which sound is broken down into tiny grains and are redistributed and played back with different parameters of time, scale and pitch reshaping the original sample in unique ways. Keeping with the abstract nature of sound that can be produced through granular synthesis, the visual strokes complement the sound by animating over time both within the mesh and the colour. The Pure Data patch for this brush was developed quite early in the process, however, the visual characteristics were developed and tweaked later in the development cycle. Both the aural and visual parameters randomise and/or evolve other time giving the user certain unpredictability but is controllable to an extent that it is not purely random. Additionally, the aim of the Particles, as well the Crackling brush (described below) is to provide a texture and an at-

Figure 6: Particles brush - gradient feature on the alpha channel

mosphere to the performance. As the user draws with this brush, the Pure Data patch initialises a sample and chooses a random starting point as the grain playback window, as well as the overlap between successive grains. The grain pitch is determined by the character settings from the interface, which in turn also affects the stroke width of the visual. As the user continues drawing, the length of the stroke is used to determine the duration of the grain playback. Visually, the stroke animates over time, with colour and texture that subtly morphs over time. Figure 6 shows the gradient feature applied on the alpha channel of the stroke which evolves over the length of the stroke making the stroke fade out at its ends.

3.6 Crackling

Crackling procedurally produces crackles, hisses and drips based on ideas illustrated by Farnell [8]. Procedural audio is a technique in which computationally generating sound through processes and modules that generate in real time. The brush also morphs in and out of its original shape with animated textures and colour gradients that evolve over time . Overall, the brushes, the synthesis engines, and the mapping strategies implemented grow in complexity and this brush is the most unpredictable within the whole group. The sound is synthesised with a noise source sent through multiple resonant and non resonant filters, with different response types and cutoff frequencies, additionally shaped by two envelopes, one controlled by a random gate and the other, updated through the drawing process, parameterised through the length of the stroke and the variance on the minor axis. Visually, both alpha channels and the colour channels animate over time, mapped to the character settings from the interface. In practise, Crackling, and to some extent Particles, work well as fill colours, complementing the first three brushes which work better as strokes, as these are disconnected to spatial coordinates and unlike Kar+Paint, do not decay. Due to the memory utilisation of these brushes, the total number of instances as permitted within the memory management strategy are rather limited as compared to the first four brushes, however, due to the atmospheric and textural nature of their sonic and visual properties, provide spatial and temporal characteristics which offset the limitations to an extent.

3.7 Gesture

Gesture brush, like the Particles brush, utilises a granular synthesis algorithm. However, it is different to all the above brushes as it does not constitute any visual material. Instead, it maps the movements of the tablet to parameters of the synthesis engine to create sounds. Additionally, this is the only sound synthesis engine that does not spawn multiple instances. Different parameters of the synthesis engine are mapped to the accelerometer of the tablet and offers the user scope to go beyond the two-dimensional space of the screen to further take advantage of both the form factor and the in-built sensors that the tablet offers.

4. DISCUSSION

Digital musical instruments and mobile music applications primarily concern themselves towards composing and performing music. Rhey are targeted towards a certain group of people with a certain vocabulary and function within established norms of time and space. For example, most sound synthesis applications would use nomenclatures and have interface elements like 'frequency', 'filter cutoff', etc. Most digital instruments based on real instruments would provide graphical representations of keys, pads, etc. Our application, on the other hand, by introducing drawing as the primary focus of user input, break a lot of these ideas and strive towards a different understanding of performability.

First of all, using drawing as the interface for music composition and performance and forgoing a traditional control schemes, it adds to it the visual dimension and can appeal to both musicians as well as illustrators. This adds to performance as well as the visuals can also be used, thus enabling a form of audiovisual performance. The act of drawing is the same as playing the instrument, not separated from each other. In this way, it becomes an immediate feedback loop where drawing and playing sound is inseparable and intertwined. As soon as the performer draws a stroke, an immediate sound heard and this enactive loop continues through the entire process of composition and performance, visually and aurally interconnected.

Second, by utilising the two dimensional surface of a tablet device, it opens up time and space in compositions and performance in a unique way. Music and sound are bound together in time, and mark making, within a two dimensional surface. By combining both, a performance can evolve over both time and space. The temporal act of drawing as well the spatial positioning feed into the process of sound generation, which in itself is a time-based process evolving through every successive instance of drawing.

Third, the barrier of entry is kept low. The act of drawing, or scratching a surface, is an act that is a natural human activity that approaching this instrument does not need any prior understanding of musical instruments or interfaces. By utilising both direct and indirect mapping strategies within each brush, learnability is also easily achieved. As the first few brushes use direct one-to-one mapping between space and sound, it is easily understood by the user and provides a good introduction to the application. Each successive brush builds on the mapping strategies used and progressing through them, the user gains more understanding and learns the application better. For example, *Across* uses the simplest of mapping strategies and the performer easily understands how the positing of their drawing affects the sound. When it progresses to *Kar+Paint*, however, the spatial location is not used to set the frequency of the sound, but the length of the stroke is utilised. Here, the understanding or learning can take slightly longer. However, as mentioned previously, with slowly building on the mapping strategies used, a certain level of learnability is maintained. Additionally, by not using similar mapping strategies between different brushes, scope is provided in the visual domain for experimenting further on the drawing aspect. In order to build soundscapes within similar frequencies, the drawing would need to be concentrated within one zone.

Finally, there are performance strategies built into this application to take advantage of its approach in working within both time and space. Treating the tablet device as a canvas, different brushes are limited within their own palettes. This gives the user opportunity to move between different brushes while freezing the instances of the other brushes to build a performance strategy. For example, the performer can draw two strokes with Line and create an additive soundscape with them. As Line is treated as a separate palette, they can be erased or cleared without affecting the soundscape created with Across. Finally, instances of Line can be frozen, the performer can move back to Across and change the earlier soundscape without affecting the instances of Line. This gives the performer opportunity to create a composition or build a performance, both in sound and visual domains.

5. CONCLUSIONS

In this paper, we presented our Sounding Brush application, its unique position within the current state of digital musical instruments and mobile music applications, its design and development and discussed what it enables for the performer or composer. By giving performers the natural interface of drawing to work with for music composition and performance, we hope to make digital musical instruments accessible to everyone. The application in its current state, was both presented at the Soundform symposium[4] as well as demonstrated [5]. There was a certain amount of interest that was expressed by the audience that reinforced our ideas, at the same time, the project needs further development be a comprehensive musical instrument for a wide variety of audiences and performers. Towards that end, the next steps would include comprehensive user testing, from people with different background, for evaluating the current state of the instrument and understanding the future developments it would need to go through. Furthermore, working within the limitations of the iPad as a platform has been a challenge and other means of implementing certain ideas for better memory handling needs to be investigated further. Various other features, discussed during the development process but not implemented due to limitations of time, would also be a point of focus, some of these being the ability to import samples for the granular engines, saving and/or loading a composition, in-built recording and sharing functionality. The open source code of the project is available at `https://github.com/SopiMlab/SoundingBrush`.

6. ACKNOWLEDGMENTS

This work was supported by the Academy of Finland (project 319946) and the Soundform - Instruments for All project.

7. REFERENCES

[1] *How to Make Music on Your iPad: The Best Synths, Samplers and Mores*, 2016 (accessed April 21, 2020).

[2] N. Bryan-Kinns and P. G. Healey. Daisyphone: Support for remote music collaboration. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Hamamatsu, Japan, 2004.

[3] B. J. Chan. *Kid Pix Around the World: A Multicultural Computer Activity Book*. Addison Wesley Publishing Company, 1993.

[4] N. Correia, K. Tahiroglu, and M. Espada. Pesi: extending mobile music instruments with social interaction. In *International Conference on Tangible, Embedded and Embodied Interaction*. Barcelona, 2013.

[5] G. Essl. Mobile phones as programming platforms. In *Proceedings of the First International Workshop on Programming Methods for Mobile and Pervasive Systems*, 2010.

[6] J. Evarts. The New Musical Notation: A Graphic Art? *Leonardo*, 1(4):405–412, 1968.

[7] M. Fabiani, G. Dubus, and R. Bresin. Moodifierlive : Interactive and collaborative expressive music performance on mobile devices. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Oslo, Norway, 2011.

[8] A. Farnell. *Designing Sound*. MIT Press, Aug. 2010.

[9] L. Gaye, L. E. Holmquist, F. Behrendt, and A. Tanaka. 2006: Mobile Music Technology: Report on an Emerging Community. In A. R. Jensenius and M. J. Lyons, editors, *A NIME Reader*, volume 3, pages 253–265. Springer, Cham, 2017.

[10] T. Kell and M. M. Wanderley. A quantitative review of mappings in musical ios applications. In *Proceedings of the SMC*, pages 473–480, 2013.

[11] S. W. Lee and J. Freeman. echobo : Audience participation using the mobile music instrument. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Daejeon, 2013.

[12] D. Oram. *An Individual Note: Of Music, Sound and Electronics*. Galliard Ltd, 1972.

[13] K. Tahiroğlu, N. N. Correia, and M. Espada. Pesi extended system: In space, on body, with 3 musicians. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 35–40, Daejeon, Republic of Korea, May 2013.

[14] K. Tahiroğlu, A. Tanaka, A. Parkinson, and S. Gibson. Designing musical interactions for mobile systems. In *Proceedings of the Designing Interactive Systems Conference*. Newcastle Upon Tyne, 2012.

[15] A. Tanaka, A. Parkinson, Z. Settel, and K. Tahiroglu. A survey and thematic analysis approach as input to the design of mobile music guis. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Ann Arbor, Michigan, 2012.

[16] M. M. Wanderley and P. Depalle. Gestural control of sound synthesis. *Proceedings of the IEEE*, 2004.

[17] G. Wang. Designing smule's ocarina : The iphone's magic flute. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Pittsburgh, PA, United States, 2009.

[18] J. Whitney. *Digital harmony*. Byte Books Peterborough, NH, 1980.

[19] M. Zbyszynski, M. Wright, A. Momeni, and D. Cullen. Ten years of tablet musical interfaces at CNMAT. In *Proceedings of the 7th international conference on New interfaces for musical expression - NIME '07*, page 100, New York, New York, 2007. ACM Press.

[4] `https://www.eucrea.de/current-projects/soundform-instruments-for-all-2018-2019`

[5] `https://www.youtube.com/watch?v=oRodExgEXU4`