Drum-Dance-Music-Machine: Construction of a Technical Toolset for Low-Threshold Access to Collaborative Musical Performance

Dominic Becking
University of Applied Sciences Bielefeld, Department for Computer Science
Minden, Germany
dbecking@fh-bielefeld.de

Christine Steinmeier
University of Applied Sciences Bielefeld, Department for Computer Science
Minden, Germany
csteinmeier@fh-bielefeld.de

Philipp Kroos
University of Applied Sciences Bielefeld, Department for Computer Science
Minden, Germany
pkroos@fh-bielefeld.de

ABSTRACT

Most instruments traditionally used to teach music in early education, like xylophones or flutes, encumber children with the additional difficulty of an unfamiliar and unnatural interface. The most simple expressive interaction, that even the smallest children use in order to make music, is pounding at surfaces. Through the design of an instrument with a simple interface, like a drum, but which produces a melodic sound, children can be provided with an easy and intuitive means to produce consonance. This should be further complemented with information from analysis and interpretation of childlike gestures and dance moves, reflecting their natural understanding of musical structure and motion. Based on these assumptions we propose a modular and reactive system for dynamic composition with accessible interfaces, divided into distinct plugins usable in a standard digital audio workstation. This paper describes our concept and how it can facilitate access to collaborative music making for small children. A first prototypical implementation has been designed and developed during the ongoing research project *Drum-Dance-Music-Machine (DDMM)*, a cooperation with the local social welfare association AWO Hagen and the chair of musical education at the University of Applied Sciences Bielefeld.

Author Keywords

NIME, hci, gestures, early childhood education, simple interfaces

ACM Classification

H.5.2 [Information Interfaces and Presentation] User Interfaces, H.5.5 [Information Interfaces and Presentation] Sound and Music Computing, K.3.1 [Computers and Education] Computer Uses in Learning

1. INTRODUCTION

We argue that early music education should begin with a playful exploration of basic properties of harmony, rhythm and melody, such as timing, loudness and tonality. Traditional instruments (e.g. stringed or keyboard) don't have suitable affordances in this sense for very young children. The actions and motions necessary to produce (harmonic) sounds are artificial, and often do not reveal themselves from the shape of the interface itself. On the other hand, percussive instruments, e.g. drums, have a particularly simple interface and interpret intuitive and familiar movements. This simplifies basic interaction and action and invites children to explorative usage. Unfortunately, they are generally designed to produce transient sounds, with few or no tonal information. However, perception of pitch, consonance and dissonance is known to be a neurologically stimulating experience[9, 10], and their creation generates a rich sense of gratification, which is vital to motivate further music making. It is therefore reasonable to devise an instrument with a low learning barrier but rich musical feedback, in order to support and enable early access to basic musical performance and intuition.

The understanding and correct usage of musical terms and concepts is also important in order to communicate about collaborative music making, and thus to achieve consonance. Rather than by theoretical explanations, this can be accomplished better and faster during the process of music making, when abstract concepts are directly applied to current events. Musical ideas can be transferred from common speech to a technical language by gestural interaction, such as posture (happy, sad) and pointing (high, low), in combination with appropriate audiovisual feedback (e.g. major/minor keys, dark/light/colorful projection).

In order to achieve the desired learning outcomes (musical intuition and expressiveness), each interaction with the instrument must result in an immediate and appropriate musical feedback. However, consonance should be guaranteed simultaneously, to not discourage explorative usage. For these reasons, we propose a musical interface that allows manipulation of music on the harmonical and structural level. Rather than producing single or multiple notes, we suggest providing a prepared but dynamic composition as a basis, responding suitably to simple user actions (like pointing to darker/brighter projection areas or crouching/jumping), with a change in the harmonic or melodic structure.

2. GOALS

The objective of this project is to develop a system in order to support a low-threshold access to making music in the context of musical education for children. It is divided into three lines of research.

The first research line is concerned with the development of an instrument with the interface of a drum, which, however, produces notes and sounds. The first idea for realization

NIME'16, July 11-15, 2016, Griffith University, Brisbane, Australia.

is to design virtual software instruments, which are driven by conventional electronic hardware drum pads connected to a MIDI converter. A second strand consists of developing a system that controls virtual instruments through movement, dance and gestures, so that the terminology and concepts which should be taught, can be experienced.

Using and based upon existing solutions for generation of adaptive music, software for the interpretation of a simple dynamic composition should be developed. In particular, the program should be able to set didactic degrees of freedom for this dynamic composition, and to provide a direct musical feedback on movement or drum actions, according to a predefined educational specification.

Overall, the three described approaches will be realized as a modular, combinable system and should be based on VST technology preferrably, allowing to integrate them into any DAW (Figure 1).

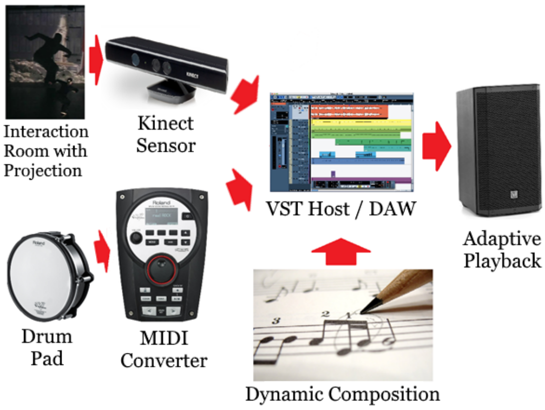

Figure 1: Intended System Structure

It is essential that all modules are able to operate independently, but also communicate when necessary. Beyond that, they should work in sync and ensure the strict real-time demands of audio data transmission for music performance while providing acceptable audio quality. According to [21], the upper bound for the bearable delay for interactive audio applications is 120 ms.

3. RELATED WORK

Some basic principles that may be useful for DDMM have already been described in other research papers (e.g. [22] or [6]), with objectives and results reflecting our observations at least partly. However, the focus is usually not early music education, the audience is usually older. No system exists so far which addresses all desired objectives and possibilities. Pachet and Adessi[15] describe a system primarily designed for the creation of improvisational phrases in jazz music. These are used to respond to phrases recorded by children. However, the system operates asynchronously, and is therefore not suitable for DDMM. Varni et al. use dialogues in a computer game to introduce children to an interface in which they are able to use their entire body to produce sound[23]. Results from this approach may be partly modified and applied to DDMM. In [4], Desainte-Catherine et al. use a joystick, in order to allow playful access to music making for children. In this case, the results in the development of specific models for virtual instruments may be of particular interest for DDMM.

3.1 Simple Interfaces for Music Making

Interesting for the development of DDMM are the results of Blaine and Perkis[2], who also choose pounding on a surface as the simplest interaction for collective, improvised music making. In their study the musicians are united using a shared interface. This approach puts the idea of collaborative music making in the foreground. They renounce, however, to grant the responsibility for their own instrument and its dominance to every musician and thus they are abstracting strongly from the actual feeling of drumming. This high abstraction in combination with the unusual auditory feedback, which is also intended in DDMM (you hit on a surface, but might not hear percussive sound), may lead to an overburdening of children of kindergarten age. Blaine and Perkins notice: *„Hitting is a most unforgiving gesture that demands immediate feedback. When someone hits a drum pad, there is a precise expectation of a reaction at a specific time. Anything other than that expected reaction intuitively sounds wrong, and makes it more difficult for players to identify their influence on the system.“* [2]

Similar ideas were used in the now commercial system “reactable”[8], even though the focus is not the simplification of the surface alone – which can be complicated almost arbitrarily by placing certain physical interaction elements – but rather on the novelty of the interaction with an electronic musical instrument. Also of interest is the approach of Zhou et al., who created MOCGLASS[24], a system for dedicated use in the classroom, which was implemented with sliders, keyboards and tap metaphors in smartphone displays. The resulting solution addresses significantly older children at the age of about 10-11 years, but individual aspects of their work may be useful for DDMM.

Even highly topical papers still deal with the issue of creating simple interfaces. This shows that the development of such aspects is absolutely essential, but also that no approach solving each use case exists. In particular, the system ”WamBam”[7], which has been developed recently is notable in this context. Jense and Leeuw describe a hand-drum for music therapy sessions with severely intellectually disabled clients. Although the described system has only a few similarities with DDMM, their results show that our concept is valid and will probably be well accepted by users.

3.2 Gesture Recognition and Music

The idea of using motion or facial expressions as a musical interface for computers is not new. The earliest example is probably the Theremin[20], which is played without direct contact: gestures in an electric field produce capacitance variations that control the output of the instrument. Later, Lyons and Tetsuta[12] describe a system that controls individual parameters of musical effects with the help of facial expressions and movements of the facial muscles.

More advanced solutions consider the whole body of the musician as musical interface and controller. Motion sensors are used by Pellegrini et al. to produce music in a process called “Soundpainting”[16], which is exceptionally interesting for the development of DDMM. It seems appropriate to build upon their results and combine them with the findings from Marinos and Geiger[13], who use a Kinect instead of multiple movement sensors. A similar approach is described by Sarasua[19], who also uses the Kinect gesture detection system, and also demonstrates possible ways of analyzing retrieved data.

Taking the specific requirements of early music education into account, the works mentioned can be included in further iterations of DDMM, and may especially lead to the development of a connected visual projection as an improved feedback method.

3.3 Adaptive Music and User Interaction

There are some interesting approaches for the creation of adaptive music from dynamic composition combined with new methods of user interaction. For example, Bauer and Waldner[1] describe possibilities of applying adaptive music to lifestyle and health. Particularly interesting for DDMM is the approach previously mentioned by Marinos and Geiger[13], who developed a system, in which gestures from an orchestra conductor has a direct impact on the performance of a composition. This is quite close to the idea of processing gestures and movement of children in a similar way. A necessary further development, however, is a mapping of the highly formalized conductoral gestures to the spontaneous, less specific gestures and movements of a child.

Livingstone et al.[11] define a system that is able to modify annotated MIDI-files based on a music-emotion rule system, which may be used to improve the expressiveness of the proposed system, though their audience consists of musically trained individuals and may therefore be too complicated. Mion et al.[14] identify timbral features of individual sounds and their relation to emotions, a musical dimension which is currently ignored by DDMM.

4. CONCEPT AND DESIGN

DDMM aims to be a tool for musical education in an intuitive rather than functional way. We support the assumption that the perception and emotional interpretation of basic musical properties, such as rhythm and timbre, are inborn to neurologically intact humans[17]. We also acknowledge that emotional interpretation of basic properties of (western) musical structure, such as melody or major and minor tonality, can be traced back to cross-cultural origins[5].

Behaviour	Property	Effect
stomping	on beat	play current note
	off beat	skip current note
	height	loudness up/down
drum hitting	on beat	play current note
	off beat	skip current note
	velocity	loudness up/down
arm moving	up/down	play parallel tonality
	left/right	quint up/down

Table 1: Actions, Variables and Effects

We therefore focus on rhythmically, harmonically and melodically simple musical performances played back in a Digital Audio Workstation (DAW) and manipulated in realtime by an extendable set of plugins. Each plugin is able to process input of an attached sensor or physical instrument, which is used as the actual interface by participating children. The current system defines two input methods and corresponding *instrument*-plugins. They have been chosen to bridge the gap between childrens natural reaction when hearing the beating of a drum – such as clapping and stomping – and creating melodic music. That is to say, we define a *drum*-plugin (section 4.2.2) which processes rhythmical *hit*-events, and a *dance*-plugin (section 4.2.3), analyzing simple and intuitive motions such as *stomping* and *raising* or *lowering* the arms.

We also define a basic metronome which generates a simple rhythm and serves as a trigger to engage the children in the interactive musical performance.

An additional plugin generates a predefined polyphonic piece described by standard MIDI events. The actual conducting of the instrument lines is varied depending on the "performance" of an associated musician. A set of current musical properties is communicated to the instrument-plugins at audioblock-rate, which can manipulate individual parameters according to their received input.

We want to motivate participation, musical ambition and intuition, and therefore reward correct timing by the playing of notes, change tonality by movement of arms and adjust the loudness of the instrument by intensity, as summarized in table 1.

4.1 Software Architecture

We use JUCE[1] as an intermediate plugin format and compile to VST2[2]. JUCE is used for rapid development of appealing user interfaces and also provides an interprocess communication (IPC) layer which is exploited in DDMM for non-standard plugin communication. This would also allow creation of other plugin formats, however, since the instrument plugins communicate with attached hardware, we focused on Windows operating systems and VST2.

The architectural design (figure 2) abstracts the commonalities of the plugins in an intermediate `DDMM`-namespace which defines the general plugin as a direct extension of JUCE' generic audio processor. As can be seen, we choose a mixture of object oriented and template based abstraction techniques to exploit the features of the implementation language C++ in the most efficient and performant way, while maintaining an intuitive and logical software design. Where possible, abstractions and type definitions are realized at compile time with template deduction. Where useful or necessary, implementation choice is delayed to runtime to provide dynamic types and flexible interfaces. All VST2-

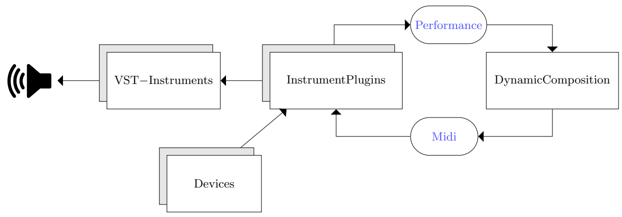

Figure 3: Signal flow

specific calls, e.g. the sample processing routine which is already wrapped and generalized in JUCE, are redefined in our general plugin to simplify the lower software layers and the actual plugin implementation. The interface is further extended for all instrument plugins, which need to provide device, buffer and musician specializations. The purpose is a unified handling of note and event history in the common code, while the exact definition of how data is retrieved from the actual device and how a buffer-event is generated from the raw input, is left to the implementation.

Our concept leans towards a modularized implementation, which is the main reason why we chose to design distinct plugins. The VST2 specification facilitates integration into a broader context and thus was chosen as our underlying protocol definition. However, our approach requires a two-way communication between the different components, as shown in figure 3. A dynamic composition is predefined and passed through the individual instrument plugins as output of our system. We don't want to distribute control over the actual composition to seperated plugins. This would most likely result in diverging output and thus dissonance, which we want to avoid. We therefore allow compositional changes only in one unique controller or (music) plugin. Since this plugin has no information about the performance of the children, this choice requires a feedback channel from the actual instrument plugins to the controller, which is not available in standard VST2. To dynamically adjust the composition

[1] Jules' Utility Class Extensions, `http://www.juce.com/`
[2] Virtual Studio Technology, `https://www.steinberg.net/de/products/vst.html`

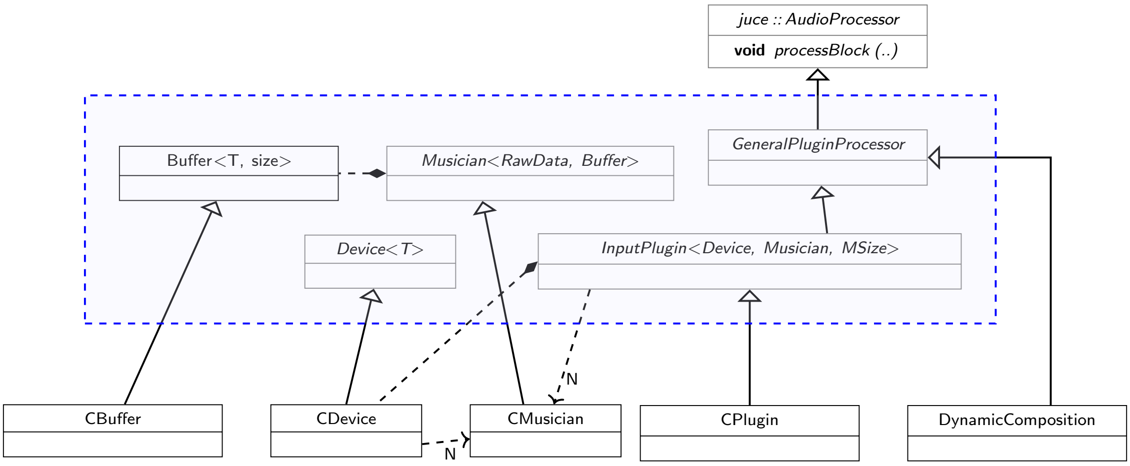

Figure 2: Software Architecture

according to the performance of the musicians, we therefore define a binarized representation of valid and appointed control data using Google's Protocol buffers[3], and pass it to the controller via IPC.

4.2 Plugins

The system is designed to be open and extendable and currently two plugins are defined to validate the general applicability with two different input methods. The third plugin, which is mandatory, emits and varies musical notes dynamically.

4.2.1 Music Plugin

The Music plugin serves as the controller in our design, reading song information from a supplied MIDI-file and constantly passing extracted and dynamically adjusted data to the connected instrument plugins. Since the intended users are (small) children, songs with a simple and short phrase structure are provided in the format of a standard MIDI file. The plugin extracts note informations for the main voice and derives the key of the piece. Accomponying parts are created dynamically for currently two voices from octave and quint of the current transposition, which can be controlled by the attached instrument plugins. To produce a steady consonance and an intuitive association between control and composition in a motivational way, the basic melodic structure is always kept intact. Dynamic is

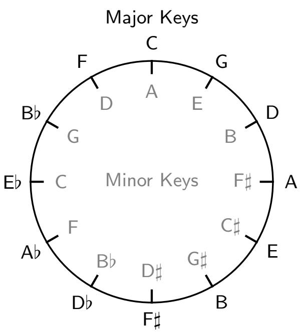

Figure 4: Circle of Fifths

introduced into the composition by consonant harmonical variations of the tune by a moving through the circle of fifth (figure 4). Assuming the tonica derived from the piece is C, transposition is possible to neighbouring keys to the left, right or bottom, e.g. F, G or A minor. This allows an expressive yet simple musical variation which can easily be mapped to corresponding motions of participating children, e.g. movement of the arms to the left, right, up or down. In addition, to provide the children with an easy to follow rhythm, a metronome sound is generated. The according clock signal is sent to attached instrument plugins together with the note information for the voices, allowing them to either silence or play the associated channel, depending on the rhythmical accuracy of the child.

4.2.2 Drum Plugin

A drum is a conceptually simple device with a particularly useful interface in the given context. Drumming is a basic movement used by children as a way to experience self-efficacy and motion[18]. It is also simple in usage, requiring no prior learning or introduction phase. It is therefore logically consistent to utilize this behaviour in early music education.

In DDMM, standard electronic drum pads are connected to a trigger interface, converting beats into standard MIDI messages. An instance of the instrument plugin is specialized to interpret timing information of these messages to either block notes or pass them through to an attached VST2-instrument. MIDI Velocity is tracked and its trend over time can be passed to the music generating plugin as loudness control through standard IPC. To allow for several drum pads per child or several children, the individual channel of each pad is associated with an own `Musician` object inside the plugin.

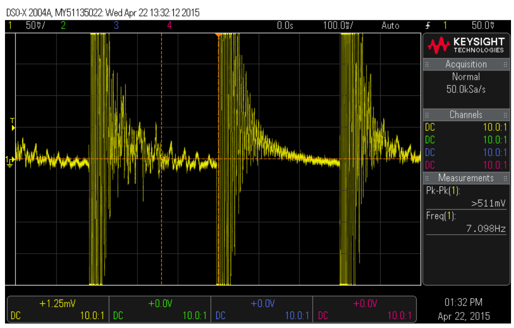

Figure 5: Oscilloscope recording of three drum beats

Figure 5 shows analog output signals from the drum pads measured with a digital oscilloscope. The signal provides information such as impact time and strength, however, it does not contain information about the position on the membrane. This is due to the fact that the piezo sensor is

[3]https://github.com/google/protobuf

mounted punctually at one location on the membrane and thus no spatial analysis can be performed.

In order to perform an accurate evaluation of beats, which is necessary for a comparison with timestamps from the metronome, reverberant vibrations of the drum pad have to be supressed. For this reason, a debouncing algorithm is carried out for every MIDI event received, and a single event is pushed to the buffer of the respective `Musician`. Currently, the recognized events are used to give out or supress notes, depending on the rhythmical correctness of the musician.

4.2.3 Kinect Plugin

In order to determine gestures and motion of music-making children, the Microsoft Kinect sensor[4] is used, a hardware system for detecting motion originally devised as a video game controller. It recognizes up to 6 persons and detects their body data. For this purpose, the different joint types and their position in three-dimensional space are stored for each person. With these 25 joint values the body information can be tracked and analyzed to determine individual poses and movement.

In the context of DDMM, the location of hands and feet are particularly interesting, as well as the position of spine and shoulders, in order to identify the center of the body. Furthermore, the height of the floor has to be detected. All other attributes are not necessary for the first prototype and are currently ignored in the process. Therefore, the extensive data provided during runtime by the connected Kinect system is read and then stored in a smaller, more efficient way. For this pupose we implemented a data structure consisting of the basic classes `Body` and `PoseEvent`, which use the additional helper classes `PoseType`, `JointType` and `BodyPart`.

By using these structures, analyses can be carried out repeatedly in order to derive poses and movements from the stored data in `Body`. Thereby the pose of the arms and the pose of the legs are checked in each cycle. If a fundamental change is detected, e.g. a previously raised leg is back on the ground, a new `PoseEvent` will be registered in the buffer of the particular musician. Minor changes in contrast only result in a change of the value of the last pose. At the current stage of development, there are 6 different arm poses implemented (top left, top center, top right, bottom left, bottom center, bottom right) and 4 different leg poses (both feet down, right foot up, left foot up, both feet up). Among others, the moments of stomping can be detected using these poses. This would require that at least one leg was raised previously and in the following pose both feet are on the floor (figure 6).

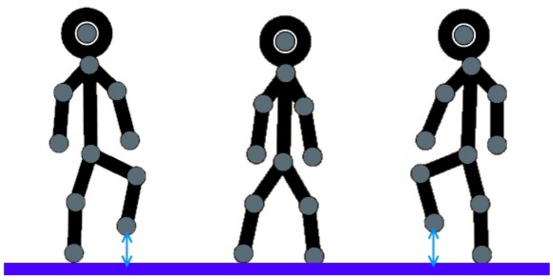

Figure 6: Feet Positioning and Stomping

If detected, the precise timepoint of a stomping event is compared with the most current MIDI note using an adjustable tolerance. For matching events, the corresponding note is then propagated to the output channel. For unmatched events, the current note is supressed. Changes in the arm pose are communicated to the music plugin via IPC, which uses the contained information to derive updated note values.

Furthermore, an accuracy value is transmitted to the music plugin in regular intervals, reflecting the overall rythmical correctness of the musician. This value is currently ignored, but may be used in future iterations to dynamically adjust the “difficulty” of the played tune. E.g., we could slow down or speed up, depending on the current accuracy, or introduce other rhythms and rhythmical variations.

[4] `http://www.xbox.com/de-DE/xbox-one/accessories/kinect-for-xbox-one`

5. CONCLUSIONS AND FUTURE WORK

In this paper we have addressed the challenge of creating a low-threshold system which aims at providing access to collaborative music making. Though we focused our work on the needs and particularities of early music education during the ongoing research, our approach is generic and may well be adapted to less specific use cases.

Basically, the achieved results show the feasability of the described concept. Followed by a short initial setup needed to e.g. associate our instrument plugins with actual sound generators, the operation of the prototype is very simple. Due to technical decisions, the initialization-parameters can be stored as a general DAW-project. Once loaded, up to two drummers and dancers can create and manipulate music at the same time, following the beat of the metronome. The implementation in the common VST2-format and the use of separate MIDI channels allow an easy and intuitive assignment of the individual instruments to each user. The decision whether an instrument plays is based on the rhythm of the musician and made in the assigned plugin. Therefore, feedback for the user is immediate and comes naturally. Harmonical and melodical variations are controlled in a central plugin, guaranteeing constant consonance and therefore motivating yet adaptive output.

After first tests, it is apparent that the general conflict between real-time processing of audio data and the parallel analysis and evaluation of external hardware inputs is not an issue for modern computer systems. Furthermore, it has been found that the use of VST2 technology is generally recommendable. Improvements are, however, possible by expanded interfaces and the definition of a meta plugin-container, which could allow for free communication between our plugins. Further improvements are intended for the dance plugin. In order to ease the process of supplementing it with other gestures, it would be helpful to use findings from machine learning. Useful approaches already exist (e.g. [3]) but have to be customized to be reasonable for the concept of DDMM.

We conclude that the current prototype can be used for the planned experiments, and if necessary, be subject to further extensions or portations to different use cases. Following a first evaluation of the prototype together with educationalists, the applicability of the implementation to the objectives of the research project should be reconsidered. However, the results demonstrate the potential of our design approach for further development. Any reader interested in being involved is encouraged to get in touch.

6. REFERENCES

[1] C. Bauer and F. Waldner. Reactive music: When user behavior affects sounds in real-time. In *CHI ’13 Extended Abstracts on Human Factors in Computing Systems*, pages 739–744, New York, NY, USA, 2013. ACM.

[2] T. Blaine and T. Perkis. The jam-o-drum interactive music system: A study in interaction design. In *Proceedings of the 3rd conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 165–173, New York, NY, USA, 2000.

[3] B. Caramiaux and A. Tanaka. Machine learning of musical gestures. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 513–518, Daejeon, Korea, 2013.

[4] M. Desainte-Catherine, G. Kurtag, S. Marchand, C. Semal, and P. Hanna. Playing with sounds as playing video games. *Comput. Entertain.*, 2(2):16–16, 2004.

[5] T. Fritz, S. Jentschke, N. Gosselin, D. Sammler, I. Peretz, R. Turner, A. D. Friederici, and S. Koelsch. Universal recognition of three basic emotions in music. *Current biology*, 19(7):573–576, 2009.

[6] J. Harriman. Start'em young: Digital music instruments for education. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 70–73, LA, USA, 2015.

[7] A. Jense and H. Leeuw. Wambam: A case study in design for an electronic musical instrument for severely intellectually disabled users. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 74–77, LA, USA, 2015.

[8] S. Jordà. The reactable: tangible and tabletop music performance. In *Proceedings of the 2010 ACM SIGCHI 28th ACM Conference on Human Factors in Computing Systems*, pages 2989–2994, Atlanta, Georgia, USA, 2010.

[9] S. Koelsch, T. Fritz, K. Schulze, D. Alsop, and G. Schlaug. Adults and children processing music: an fmri study. *Neuroimage*, 25(4):1068–1076, 2005.

[10] S. Koelsch, T. Grossmann, T. C. Gunter, A. Hahne, E. Schröger, and A. D. Friederici. Children processing music: electric brain responses reveal musical competence and gender differences. *Journal of Cognitive Neuroscience*, 15(5):683–693, 2003.

[11] S. R. Livingstone, R. Muhlberger, A. R. Brown, and W. F. Thompson. Changing musical emotion: A computational rule system for modifying score and performance. *Computer Music Journal*, 34(1):41–64, 2010.

[12] M. J. Lyons and N. Tetsutani. Facing the music: A facial action controlled musical interface. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems*, pages 309–310. ACM, 2001.

[13] D. Marinos and C. Geiger. Facilitating the creation of natural interactions for live audiovisual performances: An authoring-by-demonstration approach. In *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, New York, NY, USA, 2014. ACM.

[14] L. Mion, G. D'Incà, A. de Götzen, and E. Rapanà. Modeling expression with perceptual audio features to enhance user interaction. *Computer Music Journal*, 34(1):65–79, 2010.

[15] F. Pachet and A. R. Addessi. When children reflect on their own playing style: Experiments with continuator and children. *Comput. Entertain.*, 2(1):14–14, Jan. 2004.

[16] T. Pellegrini, P. Guyot, B. Angles, C. Mollaret, and C. Mangou. Towards soundpainting gesture recognition. In *Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound*, New York, NY, USA, 2014. ACM.

[17] I. Peretz. The nature of music from a biological perspective. *Cognition*, 100(1):1–32, 2006.

[18] J. L. Reifinger. Skill development in rhythm perception and performance: A review of literature. *Update: Applications of Research in Music Education*, 25(1):15–27, 2006.

[19] A. Sarasua. Context-aware gesture recognition in classical music conducting. In *Proceedings of the 21st ACM International Conference on Multimedia*, pages 1059–1062, New York, NY, USA, 2013. ACM.

[20] L. Sergejewitsch. Theremin: Method of and apparatus for the generation of sounds. In *US Patent*, 1928.

[21] R. Steinmetz. Human perception of jitter and media synchronization. *IEEE Journal on Selected Areas in Communications*, 14(1):61–72, 1996.

[22] C. Trappe. Making sound synthesis accessible to children. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, Michigan, USA, 2012.

[23] G. Varni, G. Volpe, R. Sagoleo, M. Mancini, and G. Lepri. Interactive reflexive and embodied exploration of sound qualities with besound. In *Proceedings of the 12th International Conference on Interaction Design and Children*, pages 531–534, New York, NY, USA, 2013. ACM.

[24] Y. Zhou, G. Percival, X. Wang, Y. Wang, and S. Zhao. Mogclass: A collaborative system of mobile devices forclassroom music education. In *Proceedings of the 18th ACM International Conference on Multimedia*, pages 671–674, New York, NY, USA, 2010. ACM.