Playing Music with the Eyes Through an Isomorphic Interface

Nicola Davanzo
Department of Electrical, Computer and Biomedical Engineering, University of Pavia
Via A. Ferrata 5 – 27100 – Pavia – Italy
nicola.davanzo01@universitadipavia.it

Piercarlo Dondi
CISRiC, Arvedi Laboratory of Non Invasive Diagnostics, University of Pavia
Via Bell'Aspa 3 – 26100 – Cremona – Italy
percarlo.dondi@unipv.it

Mauro Mosconi
Department of Electrical, Computer and Biomedical Engineering, University of Pavia
Via A. Ferrata 5 – 27100 – Pavia – Italy
mauro.mosconi@unipv.it

Marco Porta
Department of Electrical, Computer and Biomedical Engineering, University of Pavia
Via A. Ferrata 5 – 27100 – Pavia – Italy
marco.porta@unipv.it

ABSTRACT

Playing music with the eyes is a challenging task. In this paper, we propose a virtual digital musical instrument, usable by both motor-impaired and able-bodied people, controlled through an eye tracker and a "switch". Musically speaking, the layout of the graphical interface is *isomorphic*, since the harmonic relations between notes have the same geometrical shape regardless of the key signature of the music piece. Four main design principles guided our choices, namely: (1) Minimization of eye movements, especially in case of large note intervals; (2) Use of a grid layout where "nodes" (keys) are connected each other through segments (employed as guides for the gaze); (3) No need for smoothing filters or time thresholds; and (4) Strategic use of color to facilitate gaze shifts. Preliminary tests, also involving another eye-controlled musical instrument, have shown that the developed system allows "correct" execution of music pieces even when characterized by complex melodies.

CCS CONCEPTS

• **Human-centered computing** → **User interface design**; *User studies; Sound-based input/output* • **Applied computing** → **Sound and music computing** • **Social and professional topics** → **People with disabilities**

KEYWORDS

Eye tracking, gaze communication, musical isomorphic interface

COGAIN '18, June 14–17, 2018, Warsaw, Poland

ACM ISBN 978-1-4503-5790-6/18/06...$15.00
https://doi.org/10.1145/3206343.3206350

ACM Reference format:

N. Davanzo, P. Dondi, M. Mosconi, M. Porta. 2018. Playing Music with the Eyes Through an Isomorphic Interface. In *Proceedings of COGAIN '18: Workshop on Communication by Gaze Interaction, June 14-17, 2018, Warsaw, Poland*, 5 pages.
DOI: 10.1145/3206343.3206350

1 INTRODUCTION

Playing a musical instrument entails manual skills, precision, and sense of rhythm. Likewise, effective music performance requires dedication, training, and concentration.

Although the eyes can potentially be fast and accurate, gaze input through ordinary eye trackers suffers from well-known issues, such as imprecise fixation detection (due to several factors, among which eye tracker design, recording environment, calibration procedure, participants, etc. [Holmqvist et al. 2012]), with consequent difficulties in selecting small screen items. Moreover, it has been demonstrated that there is an upper limit to the number of fixations that a subject can voluntarily perform [Hornof 2014]: even if such limit is relatively high (one fixation every 300 ms), it constrains the rate of playable notes if fixations are used to trigger musical events.

When developing new interaction modalities, "natural" solutions with clear connections to traditional interaction modes are generally preferable (especially for the casual user), as they require limited learning efforts. However, less natural interaction techniques may be better suited for users with specific needs and training, like people with restricted motion capabilities [Jacob 1995]. This is the main target of our work.

In this paper, we present *Netytar*, an isomorphic virtual musical instrument playable using the eyes and a "switch" (for example, a button, a sip-and-puff tool, a pedal, or even eye blinking) exploited to activate the selected note.

Unlike other gaze-controlled musical instruments, Netytar is not characterized by any Delayed Audio Feedback (DAF, a pause between the action of the physical input and the generation of the related sound), which may reduce the quality of the musical performance. Although our application is mainly designed for motor-impaired people, who cannot play a standard musical instrument, it can be also employed by able-bodied users for playing two instruments at the same time.

2 RELATED WORK

Very few gaze-based methods for playing music have been developed to date. Interesting analyses of strengths and weaknesses of these approaches, as well as limits and challenges that future solutions should address, can be found in the works by Hornof et al. [2008] and Hornof [2014].

EyeMusic [Hornof and Sato 2004] is a first attempt at creating tools for generating sounds with the eyes, although they cannot be strictly considered real musical instruments. *Eye play the piano* [Dredge 2015] allows to select notes and harmony by looking at hexagonal graphical shapes that control the keys of a real piano. *EyeJam* [Morimoto et al. 2015] proposes a method for note selection called "context-switch", where sound is produced only when the gaze crosses a horizontal line.

Eye.breathe.music [Bailey 2010], similarly to our solution (but using a completely different interface), is probably the first to propose a hybrid method involving both gaze and an additional switch (a sip-and-puff controller): a note is selected by gaze, and then its actual playing occurs by blowing into the activation tool.

Eye Play Music [Tobii Dynavox], *EyeConductor* [Ceurstemont 2016], and *EyeHarp* [Vamvakousis and Ramirez 2016] introduce pie-shaped interfaces in which the central area is a "pause" region. EyeConductor also exploits facial expressions, such as raising eyebrows or opening the mouth to change octaves or to control filters. EyeHarp deserves special attention because, like our proposal, is a complete musical instrument that allows to play notes on several octaves and to control sound dynamics. Its interface (Fig. 1a) is formed of buttons with white dots used to guide the gaze, and the central area is exploited for pauses and note repetition.

3 OUR PROPOSAL: *NETYTAR*

Usually, musical instruments are built to be played by means of the hands. Similarly, a gaze-based musical interface should consider the characteristics of eye movements to guarantee comfortable and effective interaction. In fact, a simple imitation of the layout of a traditional musical instrument can lead the user to involuntary activate intermediate notes when playing (the well-known Midas Touch problem). When gaze is used to select notes, it is not possible to stop the musical execution in the same way a musician can remove his or her hands from an ordinary instrument.

A possible solution to this problem is to apply a delayed selection method (dwell time). In musical terms, however, this is not an appropriate solution, because it introduces a DAF between the action of the physical input and the generation of the related sound. A DAF may alter the quality of a musical performance, impeding correct play of rhythmic pieces [Pfordresher and Palmer 2002]. This is the reason why, to allow fast musical executions and "virtuosities", we avoided dwell time in the implementation of our solution.

Most recent systems, such as EyeHarp (Fig. 1a), arrange virtual "keys" (i.e., gaze-sensitive regions corresponding to single notes) in a circular layout, which reduces the probability of intermediate key crossing. However, in a pie-shaped interface where keys have a finite area, the gaze may still involuntarily cross (wrong) keys before reaching a specific target (Fig. 1b) — in EyeHarp, the problem is solved through a (short) dwell time. Also, increasing the number of keys to augment the sounding range of the instrument would cause a loss of accuracy, since key size would be reduced accordingly (Fig. 1c).

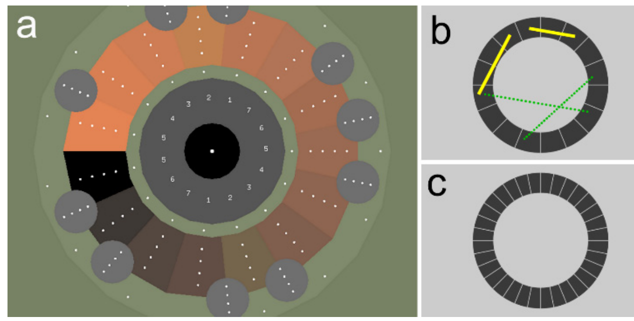

Figure 1: (a) EyeHarp interface with a neutral area in the center; (b) Keys with non-negligible areas may be involuntarily activated (yellow, solid lines); (c) Increasing the number of keys affects precision.

The Netytar interface (Fig. 2) is formed of an array of points, or *nodes*, each corresponding to a note, arranged in a grid of squares.

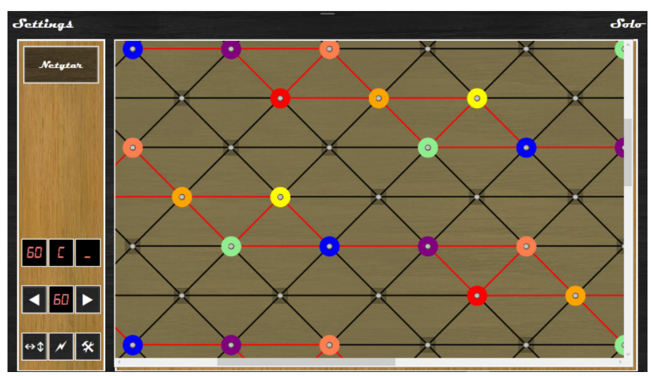

Figure 2: The *Netytar* interface.

A note is played when its node is looked at and a switch is triggered. The switch can be of diverse types, depending on the kind of disability and needs of the performer. It could be a pedal, a button (e.g., a key on the keyboard), a mouth-driven device (e.g., a sip-and-puff controller, which would also allow to control volume and vibrato effects), etc. Using a switch eliminates the need for pause areas in the interface.

Netytar also introduces an auto-scrolling approach: the grid moves automatically, and "gently" places the node corresponding to the just played note at the center of the visualization area. The speed at which the interface moves is proportional to the square of the distance between the observed point and the center of the visualization area. This allows to have a theoretically infinite playing region available, regardless of the screen size. This solution also increases accuracy, as gaze detection is usually more accurate in the central screen area [Holmqvist et al. 2012].

The basic principle behind the layout of Netytar is twofold. On the one hand, we wanted to minimize the eye movements necessary to play most common intervals in music. On the other hand, we desired to reduce the risk of unintentionally looking at wrong nodes. The arrangement of nodes in Netytar obeys the following rule: starting from any node (i.e., note), a rightward shift corresponds to a positive jump of two semitones, while a leftward move corresponds to a negative jump of two semitones. Analogously, a downward shift causes a positive jump of four semitones, while an upward shift produces a negative jump of four semitones. Moving on the top-right/bottom-left diagonal produces negative (towards top-right) and positive (towards bottom-left) jumps of a single semitone. Moving on the top-left/bottom-right diagonal produces negative (towards top-left) and positive (towards bottom-right) jumps of three semitones. All other intervals are obtained as vectorial sums of the "vectors" given by the different directions. For example, Fig. 3 shows (red line) a jump of seven semitones (4 + 3).

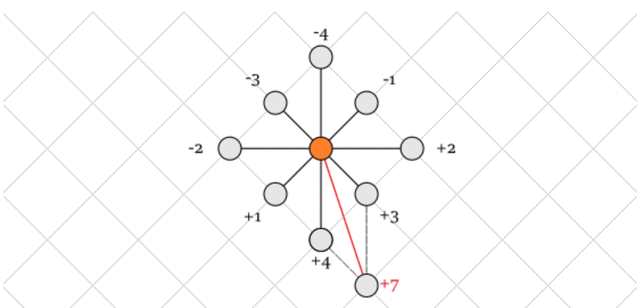

Figure 3: Intervals (indicated in number of semitones) from the (orange) central node to its neighbors.

Musically speaking, this means that the interface of Netytar is *isomorphic*: the transposition of any music piece, scale, or chord to a different musical key does not change the "shape" of the corresponding path, as shown in Fig. 4a (in a non-isomorphic instrument like the piano, for example, different combinations of white and black keys would be necessary).

Another consequence of the Netytar layout is that different nodes can be used to play the same note, and therefore there are several "paths" to play the same sequence. As stressed by Maupin et al. [2011], the specific isomorphic arrangement chosen affects the melodic and harmonic capabilities of an instrument.

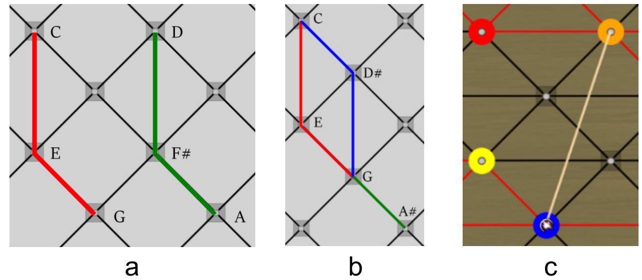

Figure 4: (a) Since the layout of Netytar is isomorphic, the C major arpeggio (C-E-G, red) has the same "shape" as the D major arpeggio (D-F#-A, green); (b) The paths required to play C (C-E-G, red), Cm (C-D#-G, blue), and C7 (C-E-G-A#, red + green) arpeggios; (c) A white line flashes indicating the gaze movement from the orange (top-right) to the blue (bottom-center) node.

Using the evaluation method described in Maupin et al. [2011], it is possible to highlight the melodic capability of Netytar, showing how simple the execution of chromatic and diatonic scales is. In particular, playing the diatonic scale is like reading text (Fig. 5): from left to right and from top to bottom.

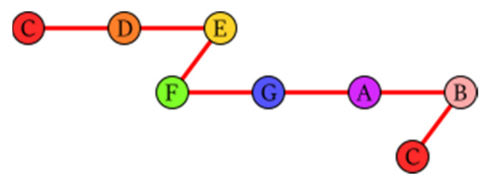

Figure 5: The path for playing a C major scale in Netytar.

As regards the harmonic capabilities of Netytar, Fig. 4b shows the paths necessary to play, respectively, a major, a minor, and a 7-dominant arpeggio. Unfortunately, there is no way to play an octave interval without traversing several nodes. A possible solution could be using an ad-hoc switch that allows to modify the octave, as happens in a recorder, such as a sip-and-puff device.

In Netytar, nodes are connected by horizontal and diagonal lines that act as "guides" for the gaze and highlight the various scales. The color of these lines depends on the type of scale: for example, red for major and blue for minor scales. Colors of nodes highlight the position of the corresponding notes within the scale (e.g., red nodes for a tonic note and blue nodes for the 5^{th}). The use of colors improves the overall accuracy of the system, since it is proved that the human peripheral vision is particularly sensitive to color variations [Lou et al. 2012].

The trace of the gaze is displayed by means of white flashing lines that connect nodes each other. A flash in the center of the node provides a visual feedback of the actual execution of the note (Fig. 4c), working as a sort of "discrete cursor". This way, the execution of a music piece can be associated with paths inside the interface, thus offering a mnemonic help to the performer. To reduce gazing errors, each node has an actual (hidden) gaze-sensitive area that is larger than the node itself.

A common concern of eye-controlled musical interfaces is the repetition of the same note. Instead of repeating gazing actions, like in EyeHarp and EyeJam, Netytar implements two alternative solutions, namely a repeated activation of the switch and the blink of one or two eyes.

4 TESTS AND RESULTS

We have carried out some preliminary tests with eight able-bodied expert musicians. Four of them were pianists, three were guitarists, and one was a saxophonist.

Designing a test for a musical instrument is a complex task, which starts with the selection of the parameters to be measured: for a tool intended for an artistic activity, it is difficult to set specific objectives, thresholds, etc. For this reason, we preferred to compare Netytar to a previous work that we consider the actual current "state of the art", i.e., *EyeHarp* [Vamvakousis and Ramirez 2016].

The test session was subdivided into three parts. In an initial training phase, the tester could play each of the two instruments for half an hour, performing free play and guided exercises. During the last ten minutes, a metronome (80 bpm) was also used. The second part of the test was a timed performance. Using a metronome at two different tempos per song (70 and 100 bpm), the tester played the initial parts of two well-known melodies, "Twinkle,Twinkle, Little Star" (Fig. 6) and "Silent Night" (Fig. 7), while the MIDI output was recorded.

Figure 6: Music sheet for the exercise "Twinkle".

Figure 7: Music sheet for the exercise "Silent Night".

The MIDI recording software had a resolution of 192 PPQ (Pulses Per Quarter note). The tester could try an exercise for five times: only the trial he or she thought was the best was considered for the analysis. The third part of the test required the tester to answer a *usability questionnaire*, providing a rating about the characteristics of each of the two instruments. Four testers tried at first Netytar and then EyeHarp, while the other four tried at first EyeHarp and then Netytar. Testers were also asked to "think aloud", and their most relevant comments were transcribed.

Tests were performed using the Tobii EyeX eye tracker. Each subject initially performed a calibration procedure using the standard tool supplied with the SDK of EyeX, which consists in fixating a few circles appearing in various positions of the screen. Calibration accuracy was checked through an interface (containing nine target points) showing gaze position in real-time. If, during the different test phases, the tester perceived a decreased accuracy, the calibration was repeated.

EyeHarp was tested in its "standard" configuration, using the latest version available (December 2017). Netytar was tested using a keyboard button as a switch. The sound produced by the instruments was the combined sound of a piano and a flute (to provide an accurate feedback on note timing and length).

In the timed performance, Netytar worked better than EyeHarp in all the tested musical pieces. Fig. 8 shows the average timing error per note (number of notes / total timing error) expressed in milliseconds. Fig. 9 depicts, for each exercise, the number of testers who played at least one wrong note. As can be seen, with EyeHarp the maximum number of wrong notes was 4, while with Netytar it was 2.

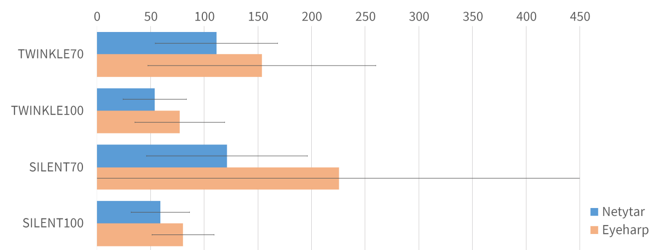

Figure 8: Average timing error (ms) per note.

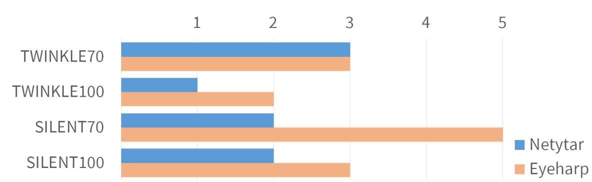

Figure 9: Number of testers who played at least one wrong note.

Regarding the usability survey, our system received almost always higher scores compared to EyeHarp. Fig. 10 shows the results of the first part of the questionnaire. Scores were provided using a five-level Likert scale: (1) I do not agree at all; (2) I do not agree; (3) I neither agree nor do not; (4) I agree; and (5) I totally agree. The complete sentences were:

- *Feeling*: In general, I had a good feeling with [...];
- *Scales*: It was simple to play scales with [...];
- *Arpeggios*: It was simple to play arpeggios with [...];
- *Complex melodies*: It was simple to play complex melodies with [...];
- *Fatigue*: It was more tiring (ocular effort) to use [...];
- *Frustration*: It was more frustrating to use [...];
- *Improvisation*: It was simple to improvise melodies with [...];

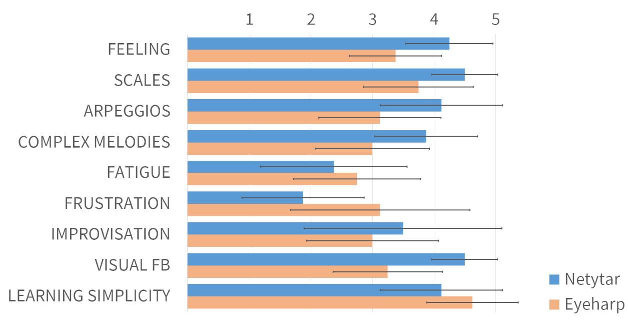

Figure 10: Results of the first part of the questionnaire.

- *Visual FB*: I appreciated the visual feedback of […];
- *Learning simplicity*: It was easy to learn how to play with […].

Fig. 11 shows the results of the second part of the questionnaire, which does not use a Likert scale but simply shows "judgments" in a scale from 1 (lowest) to 5 (highest). The full sentences were:

- *Timing difficulty*: Express the difficulty you had in playing in time using […];
- *Eyes vs. hands*: Express how hard it seemed to you to play in time using your eyes, compared to playing with your hands, using […];
- *Scrolling nuisance:* I found the Netytar grid scrolling annoying.

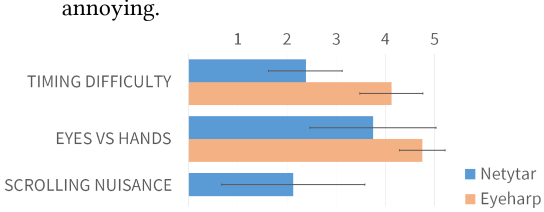

Figure 11: Results of the second part of the questionnaire.

The results shown in Fig. 11 seem to confirm the lower difficulty of playing in time with Netytar ("Timing difficulty", "Eyes vs Hands"). Also, the automatic scrolling system ("Scrolling nuisance") does not appear to bother the performer.

5 CONCLUSIONS

Preliminary tests seem to confirm that the absence of delayed audio feedbacks in Netytar is positively perceived by the performer. Probably, the use of a hybrid eye tracker + switch solution also plays a key role.

Several features included in the interface of Netytar can be traced to well-known usability principles, that guided our design process. The increased size of the gaze pointing area for notes, in compliance with Fitts's Law [Fitts 1954], aims to improve the speed and the accuracy of the system with respect to other similar tools. The isomorphic layout, while different from traditional keyboards, makes the transposition of sequences of notes more immediate (thus reducing the mental load) and allows to memorize musical phrases as visual paths. In addition, the isomorphic layout offers a certain degree of flexibility: indeed, the performer can choose among different paths to play the same sequence.

The main advantages of Netytar compared to other state-of-the-art alternatives can be summarized as follows:

1) Absence of filters, i.e. delayed audio feedbacks;
2) Minimization of gaze shifts needed to play the most common musical intervals;
3) Strategic use of color, to reduce "exploration gaze movements", which may lead to involuntary note activations;
4) Use of an automatic-scrolling system, which provides a potentially unlimited playing area regardless of screen size (without the need to reduce the size of interface elements to expand the tonal range of the instrument).

Furthermore, in the tests, Netytar exhibited a decidedly better performance in all the measured metrics (execution accuracy, user satisfaction, octave extension, etc.).

Future works will include further experiments to more deeply assess Netytar under different test conditions.

REFERENCES

S. Bailey, A. Scott, H. Wright, I. Symonds, and K. Ng. 2010. Eye.Breathe.Music: creating music through minimal movement. In *Proceedings of the 2010 International Conference on Electronic Visualisation and the Arts* (EVA'10), 254-258.

S. Ceurstemont. 2016. Virtual instruments let you play music using only your eyes. *New Scientist*, 16 March 2016 (https://www.newscientist.com/article/2081150-virtual-instruments-let-you-play-music-using-only-your-eyes/).

S. Dredge. 2015. Disabled boy learns to play piano with his eyes using virtual-reality headset. *The Guardian*, 14 January 2015 (https://www.theguardian.com/technology/2015/jan/14/disabled-play-piano-eyes-virtual-reality-headset).

P. M. Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology,* 1954 47(6): 381-391.

K. Holmqvist, M. Nyström, and F. MulveyI. 2012. Eye tracker data quality: What it is and how to measure it. In *Proceedings of the 2012 ACM Symposium of Eye Tracking Research and Applications* (ETRA 2012), 45-52

A. Hornof. 2014. The Prospects for Eye-Controlled Musical Performance. In *Proceedings of the 2014 Conference on New Interfaces for Musical Expression* (NIME'14), 461-466.

A. Hornof, T. Rogers, J. Stolet, and T. Halverson. 2008. Bringing to Life the Musical Properties of the Eyes. *University of Oregon Department of Computer and Information Science, Technical Report 08-05.*

A. Hornof and L. Sato. 2004. EyeMusic: Making Music with the Eyes. In *Proceedings of the 2004 Conference on New Interfaces for Musical Expression* (NIME'04), 185-188.

R. J. K. Jacob. 1995. Eye tracking in advanced interface design. In *Virtual environments and advanced interface design*, Oxford University Press, 258-288.

J. Kim, G. Schiemer, and T. Narushima. 2007. Oculog: Playing with Eye Movements. In *Proceedings of the 2007 Conference on New Interfaces for Musical Expression* (NIME'07), 50-55.

C. I. Lou, D. Migotina, J. P. Rodrigues, J. Semedo, F. Wan, P. U. Mak, P. I. Mak, M. I. Vai, F. Melicio, J. G. Pereira, and A. Rosa. 2012. Object Recognition Test in Peripheral Vision: A Study on the Influence of Object Color, Pattern and Shape. In *Proceedings of the 2012 International Conference on Brain Informatics* (BI'12), 18-26.

S. Maupin, D. Gerhard, and B. Park. 2011. Isomorphic Tessellations for Musical Keyboards. In *Proceedings of the 8th Sound and Music Computing Conference* (SMC'11), 471-478.

C. H. Morimoto, A. Diaz-Tula, J. A. T. Leyva, and C. E. L. Elmadjian. 2015. EyeJam: A Gaze-Controlled Musical Interface. In *Proceedings of the 2015 Brazilian Symposium on Human Factors in Computing Systems* (IHC'15), 360-368.

P. Q. Pfordresher and C. Palmer. 2002. Effects of delayed auditory feedback on timing of music performance. *Psychological Research*, 2002; 16: 71-79.

Tobii Dynavox. *Eye Play Music.* Software downloadable from http://www2.tobiidynavox.com/product/eye-play-music/.

Z. Vamvakousis and R. Ramirez. 2016. The EyeHarp: A Gaze-Controlled Digital Musical Instrument. *Front. Pychol.* 2016; 7: 906.