THE EYEHARP: AN EYE-TRACKING-BASED MUSICAL INSTRUMENT

Zacharias Vamvakousis
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
zackbam@gmail.com

Rafael Ramirez
Universitat Pompeu Fabra
Roc Boronat 138
08018 Barcelona, Spain
rafael.ramirez@upf.edu

ABSTRACT

In this paper we present the EyeHarp, a new musical instrument based on eye tracking. The EyeHarp consists of a self-built low-cost eye-tracking device which communicates with an intuitive musical interface. The system allows performers and composers to produce music by controlling sound settings and musical events using eye movement. We describe the development of the EyeHarp, in particular the construction of the eye-tracking device and the design and implementation of the musical interface. We conduct a preliminary experiment for evaluating the system and report on the results.

1. INTRODUCTION

Traditionally, music performance has been associated with singing and hand-held instruments. However, nowadays computers are transforming the way we perform and compose music. Recently, music performance has been extended by including electronic sensors for detecting movement and producing sound using movement information. One early example of this new form of music performance is the theremin and terpsitone [1]. More recent examples of new music performance paradigms are systems such as The Hands [2] and SensorLab [3]. The creation of these kinds of musical electronic instruments opens a whole new door of opportunities for the production and performance of music.

Eye tracking systems provide a very promising approach to real-time human-computer interaction (a good overview of eye tracking research in human-computer interaction can be found in [4]). These systems have been investigated in different domains such as cognitive psychology where eye movement data can help to understand how humans process information. Eye tracking systems are also important for understanding user-device interaction and to allow physically disabled people to communicate with a computer using eye movements.

In this paper, we present the EyeHarp, a new music instrument based on eye tracking. We have built a low-cost tracking device based on the EyeWriter project [5] and implemented various musical interfaces for producing sound. The resulting system allows users to perform and compose music by controlling sound settings and musical events using eye movement.

The rest of the paper is organized as follows: Section 2 describes the background to this research. Section 3 presents the EyeHarp, in particular it describes the construction of the eye-tracking device, the design and implementation of the musical interface, and the evaluation of the system. Finally Section 4 presents some conclusions and future work.

2. BACKGROUND

2.1 Eye tracking systems

Several approaches for detecting eye movement have been proposed in the past. These have included electrophysiological methods [6,7], magnetic search coil techniques [8], infrared corneal reflectance and pupil detection methods.

Electrophysiological methods involve recording the difference potentials generated between electrodes placed in the region around the eyes. However, this method has been found to vary over time, and is affected by background activation of eye muscles [7]. The disadvantages of search coil systems are that its use involves quite invasive procedures, and it relies on expensive hardware (i.e. around US$40,000).

In recent years video-based eye movement detection has gained popularity due to the fact that it offers a solution to some of the limitations of other methods. For instance, it allows reliable tracking of the pupil as well as tracking of the iris as it rotates torsionally around the optic axis [9,10] at rates of up to 250 frames per second. However, one limitation of this type of system is the need for greater intensity of infrared illumination to allow adequate passing of light from the eye to the camera sensor.

Combined pupil detection and corneal reflection techniques are becoming more and more popular lately for interactive systems. The reason is that with this combined method the head of the user does not have to be fixed.

2.2 Eye-tracking-based music systems

The first system using eye tracking devices to produce music in real time was proposed by Andrea Polli in 1997 [11]. Polli developed a system which allowed performers to access a grid of nine words spoken by a single human voice by making saccadic to nine different directions. After trying different artistic implementations Polli concluded that improvising with the eye-tracking instrument could produce the same feeling for the performer as improvisation with a traditional instrument [11].

In 2001 she performed "Intuitive Ocusonics", a system for sound performance using eye tracking instruments to be performed live. Instruments were played using distinct eye movements. Polli's compositions responded to video images of the eye, not specifically the pupil center which are parsed and processed twelve times per second using the STEIM's BigEye software (www.steim.org). With this technology it is impossible to calibrate the pupils position to the computer screen coordinates, thus the user does not have precise control of the system.

Hornof et al. [12] propose a system based on a commercial eye tracking system, the LC Technologies Eyegaze System, which provides accurate gazepoint data using the standard pupil-center corneal-reflection technique. In the system, the coordinates of the user' s gaze are sent to MAX/MSP for generating sound. They study both the case of using fixation detection algorithms for choosing an object and the raw data form the eye tracker. When trying to implement an eye-piano they report that the musicians that tried the system preferred to work with the raw data instead of a dispersion-based fixation-detection for playing the notes. The problem with fixation detection is that it reduces the temporal control, which is very critical in music. A velocity-based fixation-detection algorithm is suggested instead. They do not consider other techniques, such as blink detection, as a method for choosing objects. The authors consider designing more interactive tools using Storyboarding. The performer moves an eye-controlled cursor around on the screen, and makes the cursor come into direct visual contact with other visual objects on the screen, producing a visual and sonic reaction. The user interacts with objects that appear on the screen, through a series of interaction sequences (like a scenario).

Hornof and Vessey in a recent technical report evaluate four different methods for converting real-time eye movement data into control signals (two fixation based and two saccade-based methods). They conduct an experiment comparing the musicians' ability to use each method to trigger sounds at precise times, and examined how quickly musicians are able to move their eyes to produce correctly-timed, evenly-paced rhythms. The results indicate that fixation based eye-control algorithms provide better timing control than saccade based algorithms, and that people have a fundamental performance limitation for tapping out eye-controlled rhythms that lies somewhere between two and four beats per second [13]. Hornof claims in [12] that velocity-based (as opposed to dispersion-based) fixation-detection algorithms work better for rhythmic control with the eyes. Fixation-detection algorithms typically employ a minimum fixation duration of 100 ms which would impose an upper bound of ten eye-taps per second.

Kim et al. [14] present a low cost eye-tracking system with innovative characteristics, called Oculog. For selecting objects, blink detection is implemented. The data from the eye tracking device are mapped to PureData for generating and interacting with four sequences. In their user interface the performer's field of vision is divided into four discrete quadrants. The direction of eye movement detected by the Oculog camera software is encoded as a combination of horizontal position (pitch) and vertical position (velocity): pitch 0 is produced by looking to the extreme left, note number 127 to the extreme right; velocity 0 is produced by looking down, velocity 127 by looking up. Assigned to each quadrant is a real-time tone generator. Each tone generator is driven by a cyclic sequence. Oculog also detects torsional movement of the eye, but this is not mapped to any control feature. The authors claim that eye tracking systems are appropriate for micro-tonal tuning (they used a 15 note scale).

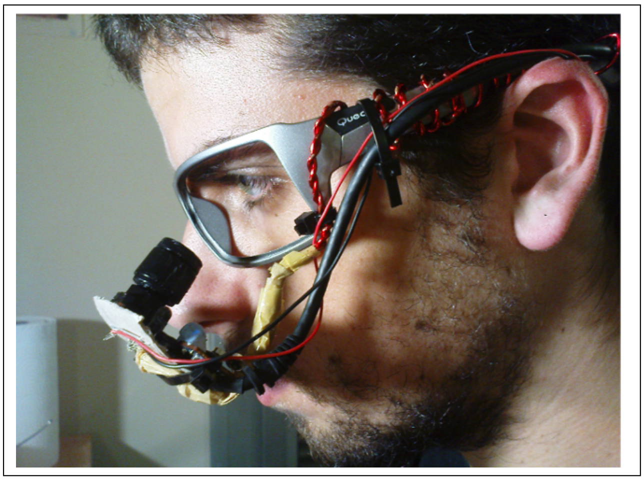

Figure 1. The PlayStation Eye digital camera is modified so as to be sensitive to infra-red light and mounted along with two infra-red leds on a pair of sun-glasses.

3. THE EYEHARP

3.1 Eye tracking device

There are a number of commercial systems available specifically designed to enable people to communicate using their eyes. However, these systems are expensive, costing in the range of US$20,000. In order to create a reproducible system we decided to make the most simple and inexpensive eye-tracking head-set possible. We built our own eye tracking system based on the EyeWriter project [5] . Thus, the resulting system emphasizes low-cost and ease of construction and as a consequence has several limitations such as robustness and appearance. Figure 1 shows the eye tracking device used in this work.

In order to read the input from the eye tracking device, we have used the libraries developed in the EyeWriter project. The eye-tracking software detects and tracks the position of a pupil from an incoming camera or video image, and uses a calibration sequence to map the tracked eye/pupil coordinates to positions on a computer screen or projection. The pupil tracking relies upon a clear and dark image of the pupil. The eye tracking device includes near-infrared leds to illuminate the eye and create a dark pupil effect.

This makes the pupil much more distinguishable and, thus, easier to track. The software dealing with the camera settings allows the image to be adjusted with brightness and contrast to get an optimal image of the eye. When initializing the system, calibration takes place displaying a sequence of points on the screen and recording the position of the pupil at each point. The user focuses on a sequence of points displayed in the screen presented one by one. When the sequence is finished, the collected data are used to interpolate to intermediate eye positions.

3.2 Music interface

The ultimate goal of this project is to create a real musical instrument with the same expressive power as traditional musical instruments. The implemented instrument should be suitable for being used as a musical instrument for performing in a band, as well as a standalone composition tool. The following decisions have been taken in the EyeHarp design:

- More than one different layer should be available. One of them could be used for building the rhythmic and harmonic musical background, and another for playing accompanying melodies on top of the musical background.
- The performer should be able to control in real time the rhythmic, harmonic and melodic aspect of his/her composition, as well as to control the timbre of the instrument. The instrument's timbre is determined by having control over (i) the spectral envelope, (ii) the attack-decay time of the produced sound. In addition, the performer should have control over the articulation and other temporal aspects of sound such as glissando and vibrato.
- The buttons on the screen for playing a note should be big enough in order to reduce the possibility of playing neighbor notes, due to errors of the eye tracking system. To save space and avoid dissonant notes the produced instrument should be diatonic (like e.g. the harmonica). The user should be able to determine the musical mode while performing.
- Temporal control in music is crucial. This is why we should avoid using blink detection or fixation detection algorithms for playing real-time melodies. Music should be controlled by making use of just the user' s gaze. Thus, the process of designing an eye tracking musical instrument is similar to designing an instrument in which the input is a pencil (eye gaze) drawing on a paper (screen), where the pencil should always be in touch the surface of the paper. Consequently the performer should be able to play every pair of notes with a straight saccade eye movement without activating any other note. This would allow working with the raw data of the eye tracker and skip the use of any fixation detection algorithm that would increase the response time of the instrument [12].

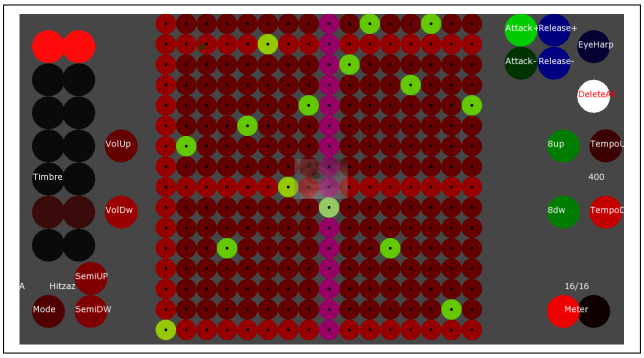

Figure 3. The EyeHarp Melodic Step Sequencer. Time Signature 16/16

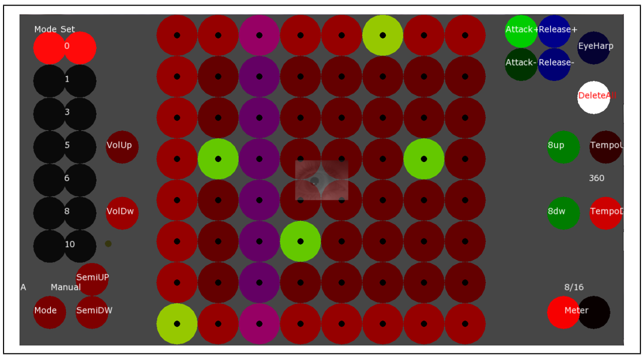

Figure 4. Setting the musical mode manually. In this case [0,1,3,5,6,8,10] corresponds to the mixolydian mode.

3.2.1 The EyeHarp Melodic Step Sequencer: Building the harmonic and rhythmic background

Various interfaces based on step sequencers are available in different environments. Two commercial examples are the Tenori-on [15] and Max For Live Melodic Step Sequencer [16]. The EyeHarp Melodic Step Sequencer is implemented using similar ideas (see Figure 2).

In the center of the screen there is a small transparent section which shows an image of the eye as captured by the camera in real-time. This is crucial for live performances, as it helps the audience to correlate the eye movements to the produced music. A small green circle indicates the user's detected gaze point. Each circle corresponds to a note. A note is selected when the user remains looking at it for more than one second. When a note is active, the color of the corresponding circle is green. Only one note can be selected for each step of the sequence. To deactivate a note the user has to look at it again for more than a second. At the center of every circle, each of which corresponds to a note, there is a black dot which helps the user to look at the middle of each circle. In every column we have the notes of the selected key with their pitch rising with direction from down to up.

The purple line in Figure 2 is moving from left to right with a speed related to the selected tempo. When the line hits one of the green circles, the corresponding note is played. So in this grid, in the horizontal dimension we have time and in the vertical dimension pitch (down→ low pitch, high→ high pitch). The horizontal brighter lines are for

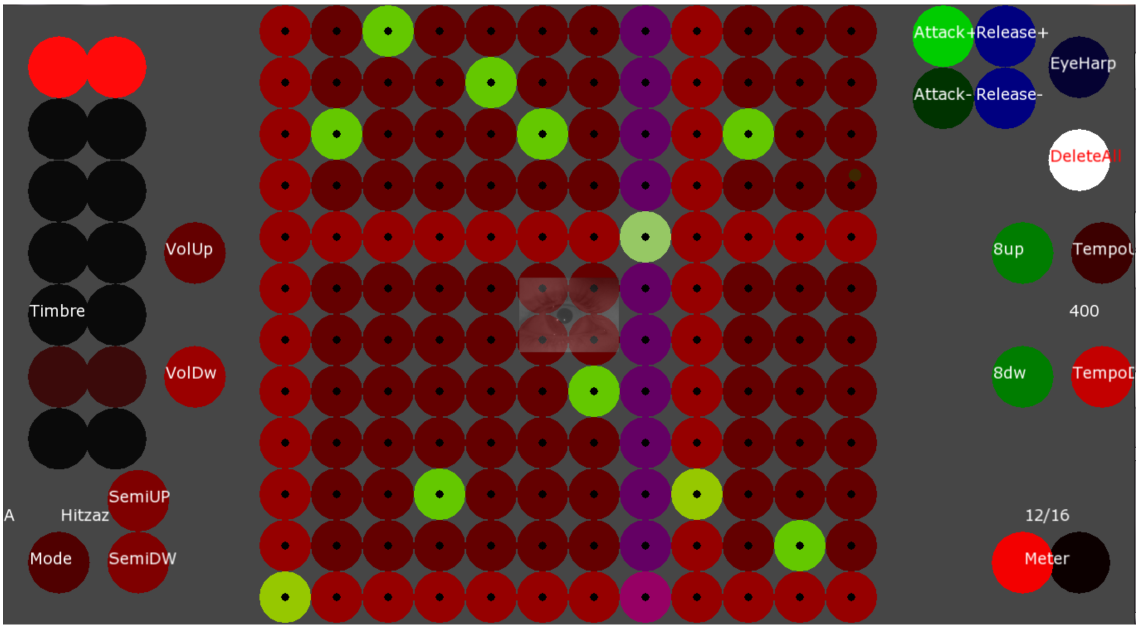

Figure 2. The EyeHarp Melodic Step Sequencer. Time Signature 12/16

helping the user to understand where a new octave starts (every 5 or 7 notes depending on the mode) The vertical brighter lines repeat every eight time steps and help to visualize beats.

On the left and right of the ”score” region described above, there are various buttons (i.e. circles) affecting different sound and musical aspects of the composition. On the left of the score region there are two circle buttons for controlling the volume. Again a time threshold is used for triggering a volume change: every 0.25 seconds that the user keeps looking at the “volumeUp” button, the volume is increased one step. The color of the corresponding button gets brighter or darker corresponding to its value. When the volume reaches its maximum value the “VolumeUp” button is bright red and “VolumeDown” button is black. That way the user has feedback about when a control parameter has reached its minimum or maximum value. The same applies to most of the circle buttons for controlling all different input variables: “SemitoneUp” - “SemitoneDown”, “AttackUp” - “AttackDown”, “ReleaseUp” - “ReleaseDown”, “TempoUp” - “TempoDown”, “OctaveUp” - “OctaveDown”, “MeterUp” - “MeterDown”. The “MeterUp” - ”MeterDown” buttons change the dimensions of the sequencer’s grid. That way the time signature of the composition changes as well (ranges from 1 to 64).

The two columns of circles at the left of the screen are for controlling the amplitude of each of the seven first harmonics of the synthesized sound. The left column is for decreasing the contribution of each harmonic and the right for increasing it. The “mode” button is for switching between different musical (scale) modes. The available modes are the major, Hitzaz [1] and pentatonic. The “mode” can also been set to “manual”. In this case the buttons that normally were for setting the amplitude of the harmonics can be used for setting the musical mode manually: each note of the scale can be assigned to any semitone. For example in Figure 4 the mode is set manually to the mixolydian. Transposition to all the different semitones is available as well. Finally “DeleteAll” sets all the notes to inactive. The “EyeHarp” button is for switching to the EyeHarp layer for playing a melody on top of the composed loop.

The EyeHarp Step Sequencer is not designed for playing real time melodies, but for building the harmonic and temporal background of the composition. The decisions mentioned at the beginning of this section apply mostly to the next proposed layer for playing real time melodies.

3.2.2 The EyeHarp

Playing Melodies in Real Time The EyeHarp interface was designed having in mind that it can be controlled with or without a gaze fixation detection algorithm. A velocity based fixation detection algorithm can be optionally activated. The velocity is computed by two successive frames and is given by the equation:

$$Velocity = \frac{\sqrt{(x_{t+1} - x_t)^2 - (y_{t+1} - y_t)^2}}{dt} \quad (1)$$

where $x_t, y_t, x_{t+1}, y_{t+1}$ are the screen coordinates of the gaze detected in each frame, and dt is the time between two successive frames. If the fixation-detection algorithm is not activated, the response time is expected to be equal to dt. If the fixation-detection algorithm is active the response time of the system will range from $2 \cdot dt$ to $4 \cdot dt$. In any eye tracking device, noise will be registered due to the inherent instability of the eye, and specially due to blinks [17]. The EyeWriter software used for tracking the gaze coordinates, provides some configurations that can help to reduce that noise. By setting the minimum and maximum

[1] Hitzaz is a mode used in Eastern music (e.g. Greece, Turkey and some Arabic countries) and flamenco music

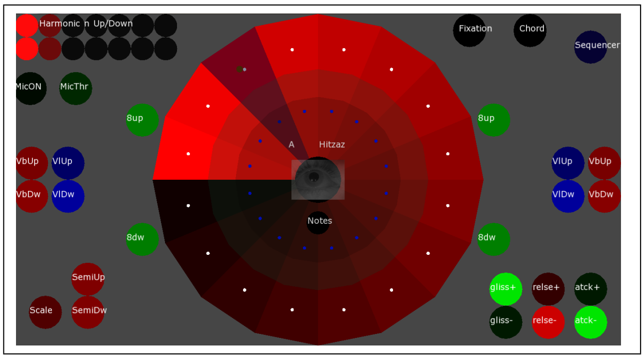

Figure 6. The EyeHarp without chords.

of the pupil' s size to the proper values the system might ignore the blinks in most of the cases. Another possible adjustment is the smoothing amount. The smoothed coordinates are given by the equation:

$$x_n = S * x_{n-1} + (1 - S) * Gx_n$$
$$y_n = S * y_{n-1} + (1 - S) * Gy_n$$

where x, y are the smoothed gaze values, Gx_n, Gy_n are the raw data of the gaze detection and S is the smoothing amount. $0 \leq S \leq 1$. For maximum temporal control, the smoothing amount should be set to zero.

The PlayStation Eye camera that is used in this project is capable of capturing standard video with frame rates of 75 hertz at a 640×480 pixel resolution. The program has been tested on a Intel Core i5 460M processor with 4GB of RAM and an nVIDIA GeForce GT 330M graphic card. For the sound to be generated smoothly, the refresh rate is set at 30 frames/second. Thus, without the fixation-detection algorithm the response time is 25ms, while with the fixation algorithm it is 50-100ms.

Spatial Distribution of the Notes The EyeHarp layer is displayed in Figure 5 and Figure 6. In order to be able to play the instrument without a fixation-detection algorithm all the notes are placed at the periphery of a circle. In the middle of this circle there is a small black circle, where the performer' s eye is displayed. If the performer looks at this circle, then the played note is released. The fixation detection algorithm is always active for this specific region. The reason for this is that the user should be able to play any melodic interval without accidentally releasing the played note. So the release region in the center is triggered only when a fixation is detected. Using that spatial distribution, the user can have control over the articulation of the sound (staccato, legato). To play staccato, after triggering a note the user' s gaze should quickly return to the center of the circle in order to release it soon. One more advantage of that spatial distribution is that all the notes are relatively close to each other, so it is easy to play every possible melodic interval. At the center of every note there is a white spot that helps the user to focus on it. A note is triggered immediately when the gaze of the user is detected inside its region. Almost no controls are placed in the region inside the circle. The user' s gaze can move freely inside this region without triggering anything. A second row of blue dots are placed inside this region. Before playing a note the user can first look on the corresponding blue spot inside the circle and then play it by looking at the white spot placed at the periphery. This way of "clicking" provides an optimum temporal control, since the note is triggered exactly when the user looks at it. If the fixation-detection algorithm is inactive, the response time is only limited by the frame rate. The dark color indicates a low pitch, while a bright color indicates a higher pitch. So the pitch increases in a counter clockwise order, starting from the most left note of the circle. If the gaze of the performer is between the small black circle in the center and the notes at the periphery of the main circle, nothing happens. The last triggered note will keep on being generated until a new note is played or it is released. As already mentioned the instrument is diatonic, so every seven or five (for pentatonic) notes we have a new octave.

Spatial distribution of the control buttons All the control buttons work in the same way as described in the step sequencer layer: there is a time threshold -different for every button- for moving the corresponding variable one step up or down.

The only control button that is inside the main circle is the one that deactivates all the notes. Obviously, there was a need for the gaze to move outside the circle in order to change several aspects of the synthesized sound. If the fixation-detection is inactive, in order to go outside the circle without triggering a note, the notes should be deactivated first. They can be activated again by looking at the black circle in the middle of the interface.

On the upper right corner of the screen, there is the "fixation" button for activating or deactivating fixation-detection. Next to it, there is the "chord" button. When it is active the notes at the upper part of the circle are assigned for changing the harmonies of the Step Sequencer. The user can build an arpeggio in the sequencer layer and then change the harmonies of his composition in the EyeHarp layer. The closest buttons to the main circle are the ones for changing octaves, and they are placed close to the lowest and highest pitches of the interface. If the fixation-detection is not active, when pressing any of the buttons for changing octave, the notes are automatically deactivated, so the user can enter inside the main circle again without triggering any note accidentally.

As it can been seen in Figure 5, there are buttons for adjusting the glissando, attack, release, volume, vibrato, amplitude of each harmonic, tonality (semitone up, down and "mode"), and switching to the sequencer layer. The two layers have their own sound properties, apart from the ones related to the tonality. That means that the timbre, articulation, temporal aspects of the sound, octave of each layer can be set to different values (e.g. choose a percussive timbre for the sequencer and a harmonic timbre for the melody). The user can also activate the microphone input for blowing in the microphone and having control over the amplitude of the melody that he is performing (a very dynamic microphone is recommended). The minimum sound level to be considered as an input can be set through the "MicThr" button.

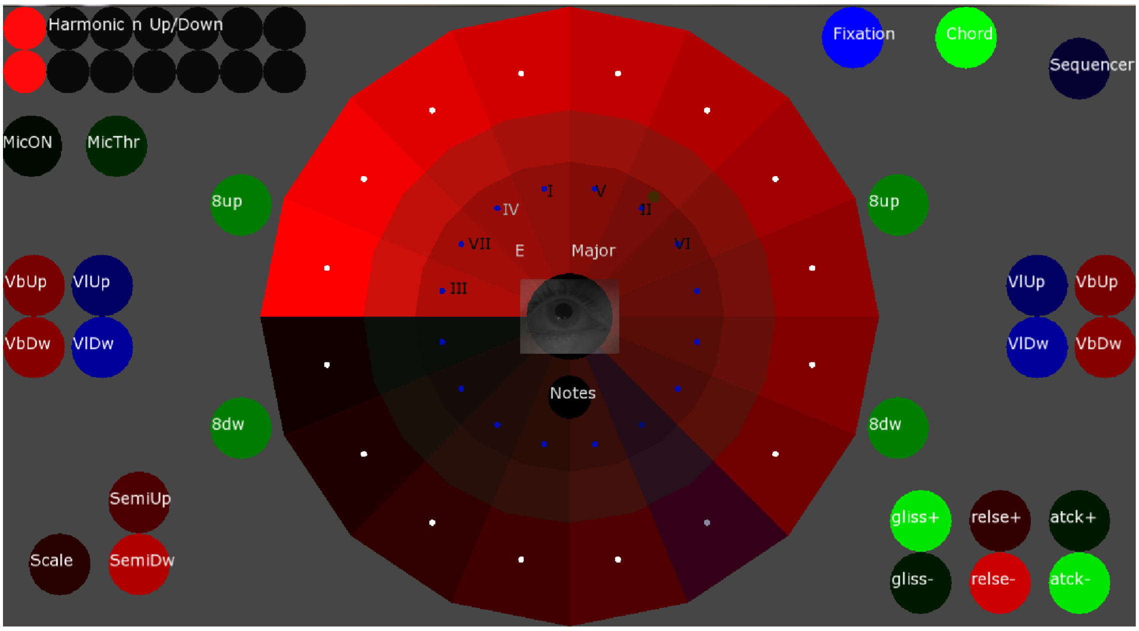

Figure 5. The EyeHarp layer. The selected scale is E major. The 4th chord of the key is played (A minor) and the melody note is do. The user is about to play the second chord of the key (F# minor).

3.3 Implementation

The interface and sound synthesis of the EyeHarp were implemented in Openframeworks [18], an open source C++ toolkit for creative coding. Openframeworks is used in all the stages of the system: (i) tracking the pupil of the eye and calibrating (based on the EyeWriter Project), (ii) designing the different modes of the EyeHarp (iii) synthesizing the sound.

3.4 Evaluation

Evaluating a new musical instrument is a difficult task. Ideally, the instrument should be evaluated at different stages. It should be evaluated on how accessible or "playable" it is for novice performers, how easy/difficult it is to improve with experience, and what is the potential of the instrument performed by experts. As a preliminary evaluation we have asked two people, one person completely novice to the instrument (playing the EyeHarp for the first time), and another more experienced person who had spent many hours using the EyeHarp, to each perform two tasks: perform a two octave scale using the EyeHarp interface as accurate and speedy as possible, and generate a note pattern on the EyeHarp melodic step sequencer as speedy as possible. in addition, for comparison purposes we have asked the same two people to perform the same tasks using a video-based head tracking software [19]. Figure 7 shows the results of the experiment.

Both (the experienced and novice) participants agreed that proficiency in the EyeHarp improves with practice. Observing the participants interact with the EyeHarp after the experiment, it seems that the fixation-detection algorithm is indeed very helpful for a novice user and can be activated for increasing the spatial accuracy of the system. The smoothing amount can be adjusted as well. The user can choose between better spatial (not pressing notes accidentally) or temporal control by adjusting these two parameters.

		Two octave scale in the EyeHarp		Arpeggio in the Step Sequencer
User	Tracking	Seconds	Accuracy	Seconds
Expert	Eye	12	100%	15
	Head	36	100%	18
Novice	Eye	18	94%	30
	Head	40	75%	29

Figure 7. Eye-Tracking and Head-Tracking for an experienced and a novice user.

It has to be noted that the accuracy of the implemented eye-tracking device was not explicitly evaluated (this is out of the scope of this paper). However, the EyeHarp interface can be used along with more accurate commercial eye tracking systems. It is very likely that the temporal and spatial control would be even better in that case.

Probably the best way to evaluate the potential of the EyeHarp as a musical instrument is to listen to performances produced using the instrument. The reader may listen (and watch) one such performance at:
http://www.dtic.upf.edu/~rramirez/eyeharp/EyeHarpDEMO.wmv

4. CONCLUSIONS

We have presented the EyeHarp, a new musical instrument based on eye tracking. We have built a low-cost eye-tracking device which communicates with a melody and step sequencer interface. The interface allows performers and composers to produce music by controlling sound settings and musical events using eye movement. We have described the development of the EyeHarp, in particular design and implementation of the melody and step sequencer interface. Finally, we have conducted a preliminary experiment for evaluating the system and compare its usability with a similar video-based head tracking controller. The results are encouraging but are still preliminary because the evaluation included only one experienced performer and one novice performer. The EyeHarp interface is still under development and many aspects, such as the choice of colors and spatial distribution of the control buttons, are still under reconsideration.

The eyeWriter project team has provided "a low-cost eye-tracking apparatus and custom software that allows graffiti writers and artists with paralysis resulting from Amyotrophic lateral sclerosis to draw using only their eyes". The eyeHarp is a musical instrument that could give to these people the opportunity to express themselves through music, but can also be used by anyone as musical instrument in a traditional way.

5. REFERENCES

[1] B. M. Galayev, "Light and shadows of a great life: In commemoration of the one-hundredth anniversary of the birth of leon theremin," *Pioneer of Electronic Art. Leonardo Music Journal*, pp. 45–48, 1996.

[2] M. Waiswicz, "The hands: A set of remote midi controllers." in *Proceeding s of the 1985 International Computer Music Conference*. San Francisco: Computer Music Association, 2003, pp. 573–605.

[3] A. Tanaka, "Musical technical issues in using interactive instrument technology." in *Proceedings of the International Computer Music Conference*. San Francisco: International Computer Music Association, 1993, pp. 124–126.

[4] R. J. K. Jacob and K. S. Karn, "Eye tracking in human-computer interaction and usability research: Ready to deliver the promises," in *The Mind's Eyes: Cognitive and Applied Aspects of Eye Movements. Elsevier Science*, . H. D. J. Hyona, R. Radach, Ed., 2003, pp. 573–605.

[5] The eyewriter project. [Online]. Available: http://www.eyewriter.org/

[6] J. A. Werner, "A method for arousal-free and continuous measurement of the depth of sleep in man with the aid of electroencephalo-, electrooculo- and electrocardiography (eeg, eog, and ekg)," in *Z Gesamte Exp. Med.*, 1961, vol. 134, pp. 187–209.

[7] S. Iwasaki, L. A. McGarvie, G. M. Halmagyi, A. M. Burgess, J. Kim, J. G. Colebatch, and I. S. Curthoys, "Head taps evoke a crossed vestibulo-ocular reflex." in *Neurology (in press)*, December 2006.

[8] D. A. Robinson, "A method of measuring eye movement using a scleral search coil in a magnetic field," *IEEE Trans Biomed Engineering*, pp. 137–145, 1963.

[9] S. T. Moore, T. Haslwanter, I. S. Curthoys, and S. T. Smith, "A geometric basis for measurement of three-dimensional eye position using image processing. vision research," in *Vision Research*, 1996, vol. 36, no. 3, pp. 445–459.

[10] D. Zhu, S. T. Moore, and T. Raphan, "Robust and real-time torsional eye position calculation using a template-matching technique." in *Comput. Methods Programs Biomed.*, 2004, vol. 74, pp. 201–209.

[11] A. Polli, "Active vision: Controlling sound with eye movements," *Leonardo*, vol. 32, no. 5, pp. 405–411, 1999, seventh New York Digital Salon.

[12] A. Hornof, "Bringing to life the musical properties of the eyes," University of Oregon, Tech. Rep., 2008.

[13] A. J. Hornof and K. E. V. Vessey, "The sound of one eye clapping: Tapping an accurate rhythm with eye movements," Computer and Information Science University of Oregon, Tech. Rep., 2011.

[14] J. Kim, "Oculog: Playing with eye movements," in . Nime 07, 2007.

[15] Y. Nishibori and T. Iwai, "Tenori-on," in *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME06), Paris, France*, 2006.

[16] Max for live. [Online]. Available: http://www.ableton.com/maxforlive/

[17] A. Duchowski, *Eye Tracking Methodology: Theory and Practice*, 2nd ed., Springer, Ed., 2007.

[18] openframeworks. [Online]. Available: http://www.openframeworks.cc/

[19] M. Betke, J. Gips, and P. Fleming, "The camera mouse: visual tracking of body features to provide computer access for people with severe disabilities," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 10, no. 1, p. 110, 2002.