Reflex-in: Generate Music on the Web with Real-time Brain Wave

Shihong Ren, Michel Buffa, Laurent Pottier, Yang Yu, Gerwin Schalk

Reflex-in: Generate Music on the Web with Real-time Brain Wave

Shihong Ren
shihong.ren@univ-st-etienne.fr
Shanghai Conservatory of Music, SKLMA
Université Jean Monnet, ECLLA
China, France

Michel Buffa
buffa@univ-cotedazur.fr
University Côte d'Azur, CNRS, INRIA
France

Gerwin Schalk
gerwin.schalk@shanda.com
Tianqiao and Chrissy Chen Institute
China

Laurent Pottier
laurent.pottier@univ-st-etienne.fr
Université Jean Monnet, ECLLA
France

Yang Yu
yuyang@shcmusic.edu.cn
Shanghai Conservatory of Music, SKLMA
China

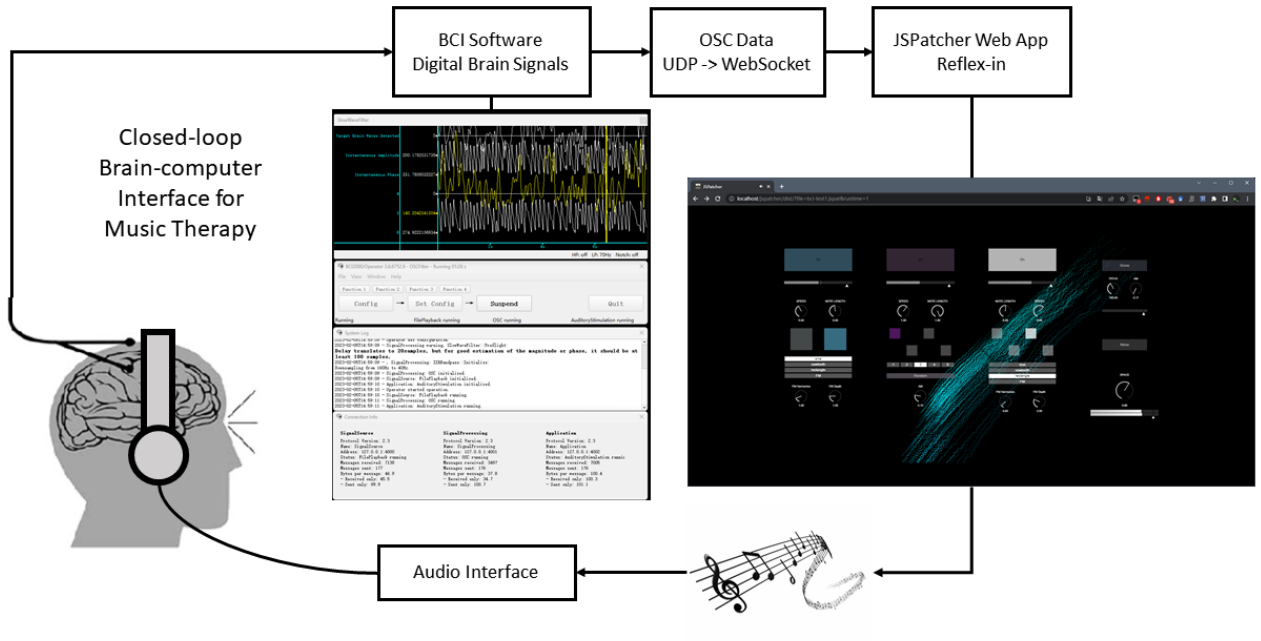

Figure 1: Diagram of the proposed medical use of Reflex-in

ABSTRACT

Reflex-in is a sound installation that uses brain-wave streams to create music composition within the Web environment in real time .

The work incorporates various state-of-the-art Web technologies, including Web Audio, WebSocket, WebAssembly, and WebGL.

The music generated from the algorithm - mapping brain wave signal to musical events - aims to produce a form of furniture music that is relaxing and meditative, possibly therapeutic. This effect can be further enhanced through binaural beats or other forms of auditory stimulation, also known as "digital drugs," which can be enabled through the user interface. The system represents a potential avenue for the development of closed-loop brain-computer interfaces by using the listener's own brain waves as the source of musical stimuli, which can be used for therapeutic or medical purposes.

CCS CONCEPTS

• **Human-centered computing** → *Interaction design*; • **Information systems** → *Web applications.*

KEYWORDS

music, brain wave, web audio, visual programming language

ACM Reference Format:
Shihong Ren, Michel Buffa, Gerwin Schalk, Laurent Pottier, and Yang Yu. 2023. Reflex-in: Generate Music on the Web with Real-time Brain Wave. In *Proceedings of The Web Conference 2023 (Conference WWW '23).* ACM, New York, NY, USA, 3 pages. https://doi.org/XXXXXXX.XXXXXXX

1 ABOUT THE WORK

Generative music from physiological sensors or EEG devices (brain wave) has been implemented since the middle of the 20th century. [1, 12] Controlling a human-computer interface using these signals is often more challenging than using gestures or physical objects due to the need for greater precision and the presence of measurement noise. The difficulty and complexity of the music creation process has hindered its development in recent decades.

However, this type of music may have potential benefit for human health. Its impact has been explored through various perspectives, including the famous "Mozart effect." [6] By using physiological measurements to assess changes in symptoms of illnesses, emotions, and sleep, researchers are able to conduct more precise experiments on the ways music can impact human health.

EEG signal is one of the main measurement tools for sleep and brain related diseases. Analysis of brain waves enables the direct identification of different sleep cycles, making it easier to conduct research on sleep quality. This property facilitates the quantitative research on sleep quality and makes the effect of auditory stimulation or music on sleep measurable in real time. [2]

Based on this context, our work aims to use Web technologies to transform smart devices into a brain-controlled music box. Our previous researches focused on web-based DSP techniques, sound design systems, visual programming languages for real-time audio processing and interactive music [8, 9, 11], and led to the design and implementation of a "patcher" web application named JSPatcher [10], a high-level visual language similar to the one of Max/MSP [4, 5], that enables users to visually, interactively, construct audio graphs using the Web Audio API. For this new project, we also rely on JSPatcher. Users can graphically design and execute DSP algorithms using domain-specific languages (DSL) for audio processing such as Faust or Gen. These algorithms are executed in a dedicated high-priority thread called an AudioWorklet. This application can also be utilized to design interactive programs and shareable digital artworks online, incorporating other JavaScript language features, Web APIs, web-based audio plugins, or external JavaScript modules.

Typical BCI software such as BCI2000 [13] or OpenBCI[1] can be used for transferring brain wave data through the UDP network protocol with the OSC format. We first forward the data through WebSocket protocol, possibly with a local server, then capture it in the browser. The peaks in the brain wave signal are then used to trigger the web-based synthesizer built in JSPatcher, which has access to the WebSocket API for processing the incoming brain wave signal.

To process the signal, a patcher is created with a web browser using the JSPatcher web application[2]. When a patcher is hosted on an online server, a URL can be used to prompt JSPatcher to open the patcher program in run-time mode, displaying user interface elements and providing a usable application for end-users, as opposed to the "editing mode."

The generated web application[3] can be used to adjust the rhythm, harmony, and timbre of the sound. These adjustments can be performed manually (using GUI elements like buttons, knobs, and sliders, etc.) or automatically over time.

The synthesizer part is created from Faust DSP. Faust is a functional, synchronous, domain-specific programming language designed for real-time audio signal processing and synthesis [3]. Thanks to the Emscripten transpiler and the WebAssembly format, the Faust compiler is available as a JavaScript module named faustwasm [7] which can compile Faust code to a fully functional WebAudio AudioWorklet node, directly in the browser. The main Faust DSP in this work is a a 16-voice additive synthesizers with envelopes, frequency modulators and amplitude modulators, Parameters affecting timbres can be controlled from the user interface.

A real-time WebGL visualization is displayed as the background of the user interface. The graphics are generated using the three.js library[4] and are based on the brain wave and the music generated. Two real-time signal analyzers are used for reporting the peak values of the brain wave signal and the audio signal. The values are used by the WebGL shaders to modify the position of the particles rendered as a real-time 2-dimensional waveform visualization.

Reflex-in is an original creation experience, implementing a set of standard web technologies. It shows that it is possible to develop original audio/musical applications with the web platform. It has mainly been designed as an artwork, while we also discussed its potential therapeutic applications. This work is a prototype for people who need to hear a real-time musical feedback from their brain wave in order to eventually improve the sleep quality. It will be tested soon with neuroscientists to assess its effect with different music parameters on a closed-loop BCI system. This work is commissioned by Shanghai Conservatory of Music and Shanghai Key Laboratory for Music Acoustics.

We propose to present this artwork at the conference, using a large display, ideally large TV with touch screen. For the demonstration/exhibition of the artwork We can simulate real-time brain wave by replaying EEG recordings on the computer.

REFERENCES

[1] Alvin Lucier. 1965. Music for a Solo Performer, live performance. *Brandeis University* (1965).
[2] Hong-Viet V. Ngo, Thomas Martinetz, Jan Born, and Matthias Mölle. 2013. Auditory Closed-Loop Stimulation of the Sleep Slow Oscillation Enhances Memory. *Neuron* 78, 3 (May 2013), 545–553. https://doi.org/10.1016/j.neuron.2013.03.006
[3] Yann Orlarey, Dominique Fober, and Stéphane Letz. 2009. FAUST : an Efficient Functional Approach to DSP Programming. In *New Computational Paradigms*

[1] OpenBCI homepage: https://openbci.com/

[2] We call a "patcher" a program written online using the JSPatcher web application. JSPatcher can run a patcher in "editing mode" that allows editing the program, or in "run-time mode" that shows the patcher like a web application.

[3] A live demo can be found at https://jspat.shren.site/dist/?projectZip=https://static.shren.site/reflex-in/project.zip&file=main.jspat&runtime=1

[4] Three.js Homepage: https://threejs.org/

for Computer Music, Editions Delatour France (Ed.). Paris, France, 65–96. https://hal.archives-ouvertes.fr/hal-02159014

[4] Miller Puckette. 1986. The patcher. In *Proceedings of the International Computer Music Conference*. Computer Music Association, San Francisco, United States, 420–429.

[5] Miller Puckette and David et al. Zicarelli. 1990. Max/msp. *Cycling* (1990).

[6] Frances H. Rauscher, Gordon L. Shaw, and Catherine N. Ky. 1993. Music and spatial task performance. *Nature* 365, 6447 (Oct. 1993), 611–611. https://doi.org/10.1038/365611a0

[7] Shihong Ren, Stéphane Letz, Yann Orlarey, Dominique Fober, Romain Michon, Michel Buffa, Laurent Pottier, and Yang Yu. 2022. Modernized Toolchains to Create JSPatcher Objects and WebAudioModules from Faust Code. In *Proceedings of the International Web Audio Conference*. Université Côte d'Azur, Cannes, France. https://doi.org/10.5281/zenodo.6767596

[8] Shihong Ren, Laurent Pottier, and Michel Buffa. 2020. From Diagram to Code: a Web-based Interactive Graph Editor for Faust DSP Design and Code Generation. In *Proceedings of the International Functional Audio Stream (Faust) Conference*. GRAME, Saint-Denis / Virtual, France.

[9] Shihong Ren, Laurent Pottier, and Michel Buffa. 2021. Build WebAudio and JavaScript Web Applications using JSPatcher: A Web-based Visual Programming Editor. In *Proceedings of the International Web Audio Conference (WAC '21)*, Luis Joglar-Ongay, Xavier Serra, Frederic Font, Philip Tovstogan, Ariane Stolfi, Albin A. Correya, Antonio Ramires, Dmitry Bogdanov, Angel Faraldo, and Xavier Favory (Eds.). UPF, Barcelona, Spain. ISSN: 2663-5844.

[10] Shihong Ren, Laurent Pottier, Michel Buffa, and Yang Yu. 2022. JSPatcher, a Visual Programming Environment for Building High-Performance Web Audio Applications. *Journal of the Audio Engineering Society* 70, 11 (Nov. 2022), 938 EP – 950. https://doi.org/10.17743/jaes.2022.0056

[11] Shihong Ren, Laurent Pottier, Michel Buffa, and Yang Yu. 2022. Making Computer Music on the Web with JSPatcher. In *Proceedings of the Sound and Music Computing Conference*, Romain Michon, Laurent Pottier, and Yann Orlarey (Eds.). SMC Network, Saint-Etienne, France, 541–548. https://doi.org/10.5281/zenodo.6573552

[12] David Rosenboom. 1990. The Performing Brain. *Computer Music Journal* 14, 1 (1990), 48. https://doi.org/10.2307/3680116

[13] Gerwin Schalk, Dennis J McFarland, Thilo Hinterberger, Niels Birbaumer, and Jonathan R Wolpaw. 2004. BCI2000: a general-purpose brain-computer interface (BCI) system. *IEEE Transactions on biomedical engineering* 51, 6 (2004), 1034–1043. Publisher: IEEE.