



HAL
open science

Guitarists will be happy: guitar tube amp simulators and FX pedals in a virtual pedal board, and more!

Michel Buffa, Jerome Lebrun

► To cite this version:

Michel Buffa, Jerome Lebrun. Guitarists will be happy: guitar tube amp simulators and FX pedals in a virtual pedal board, and more!. Web Audio Conf 2018, Sep 2018, Berlin, Germany. hal-01893681

HAL Id: hal-01893681

<https://hal.univ-cotedazur.fr/hal-01893681>

Submitted on 11 Oct 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Guitarists will be happy: guitar tube amp simulators and FX pedals in a virtual pedal board, and more!

Michel Buffa
Université Côte d'Azur
CNRS, INRIA
buffa@i3s.unice.fr

Jerome Lebrun
Université Côte d'Azur
CNRS
lebrun@i3s.unice.fr

OUTLINE

In this demo, we propose to showcase what we've been developing recently: a versatile, fully functional, low-latency virtual guitar tube amp designer, able to emulate each stage of many famous tube guitar amps or to create your own personal Web Audio tube amp. We have also included a large set of FX pedals, newly created or adapted to work with the Web Audio API. Finally, we have also developed a pedal board host application that mimics a real guitarist pedal board, where one can assemble with a graph the amp simulators, the FX pedals or many other virtual instruments.

SOME DETAILS

In the two previous Web Audio conferences [1,2,3], we presented, with quite some success, two versions of our Web Audio implementation of a classic tube guitar amp, the Marshall JCM800. We detailed how its different stages had been recreated and let everybody around to play with it¹.

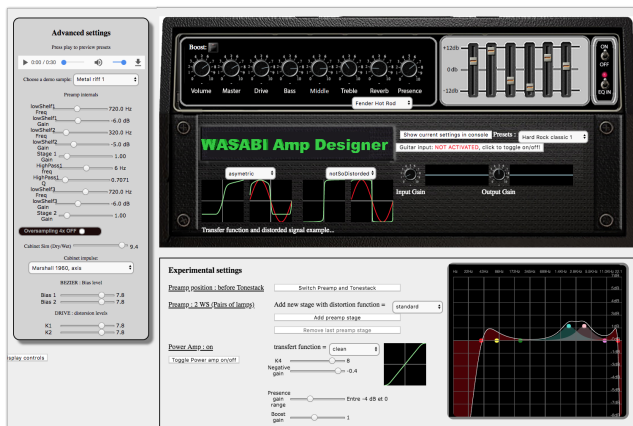


Figure 1: The tube amp designer GUI

The quality of the simulations was very positively evaluated, with low latency and robustness, up to the standard of the best native commercial competitors.

Here, we will detail first how we built on these previous works in order to generalize our approach towards any tube amp simulations. Instead of simply emulating a single guitar amp, we have now a versatile amp designer where one can select and tune the different parts that compose an amplifier: preamp, tonestack, power amp, speaker simulation, reverb, ... and then assemble those in different topologies, using filter banks between or inside these stages (Fig. 1). This modular tools make it possible to create specific amp designs targeted to get clean, crunch/bluesy, classic rock distortion, hi gain/metal sounds, etc. The final result can be a single specific amp or some multi channel amp with switchable designs² (ex: a clean and a crunch channel).



Figure 2: Different plugins connected inside a virtual pedal board. In black: some amps simulators

Parallely, we've been working with other researchers [4,5,6] on an open plugin standard for Web Audio plugins

² video available at <https://youtu.be/-NdMdJQx2Bw>

[7]: the pedal board is a host and the amps, as well as FX pedals and instruments are plugins. These plugins are identified by their unique URI and located on local or remote repositories. The host application, i.e. the pedal board, scans these repositories and gets the plugin metadata. Through the pedal board GUI, it is now possible to create plugin instances by drag and dropping thumbnails to the main pedal board area. When this occurs, the Plugin's code and assets are then loaded dynamically and a plugin instance is returned as a JavaScript object. Then, this object can be used to get the "audio processor part of the plugin" and connect it to the Web Audio graph, or to get the plugin GUI as a unique HTML container element.

We did the proof of concept that many plugins could be easily included from multiple origins: (1) written in pure JavaScript / Web Audio -we made a dozen of such plugins, including the guitar tube amp simulators, and ported some others created by other developers (such as a general midi synth, etc.), (2) WebAudioModules (VST/JUCE native plugins ported to WebAssembly/AudioWorklet) or (3) written in FAUST, a popular DSL for writing DSP code, here again ported to WebAssembly/AudioWorklet. Other sources / importers are planned (eg. MAX DSP/Pure Data). Most plugins are controllable using midi controllers. We will demonstrate how easy this scheme works³.



Figure 3: A typical benchmark/demo setup: a guitar, a low latency sound card, some speaker and a Web browser with our Web Audio apps running.

We finally also performed some blind tests of our framework with professional guitarists (Fig. 3). We asked

³Some videos are available at <https://youtu.be/PiOD7n3g-Qs> or <https://youtu.be/elbjh6tBK6U> (for the pedal board and plugins)

them to play guitar in our Web Audio pedal board with amps and pedals and also on native simulations of similar amplifiers so as to benchmark our simulations and to provide feedback. The results proved that we are up to the level of native simulations in terms of low latency, sound quality, timbre, dynamics, and more generally of playing comfort. For example, our simulations were much better evaluated than those of GarageBand. The latency on Mac OS is quite low and comparable to the best native apps, and most important was not perceptible by the testers.

DEMO SETTINGS

For the demo we would need a big screen or TV, decent speakers and a low latency sound card.

A typical demo is: look for a song in our WASABI⁴ songs database, switch to the embedded multitrack player, mute the guitar track and open the pedalboard, build your preferred virtual guitar rig, plug your guitar, adjust the sound and play along with the other tracks, with no noticeable latency.

ACKNOWLEDGMENTS

EIMahdi Korfed and Guillaume Etevenard who helped developing these tools. French Research National Agency (ANR) and the WASABI project team (contract ANR-16-CE23-0017-01).

REFERENCES

- [1] Buffa, M., Demetrio, M. and Azria, N. 2016. Guitar pedal board using WebAudio. In Proc. 2th Web Audio Conference (WAC 2016). Atlanta, USA.
- [2] Buffa, M. and Lebrun, J. 2017. Real time tube guitar amplifier simulation using WebAudio. In Proc. 3rd Web Audio Conference (WAC 2017). London, UK.
- [3] Buffa, M. and Lebrun, J. 2017. Web Audio Guitar Tube Amplifier vs Native Simulations. In Proc. 3rd Web Audio Conference (WAC 2017). London, UK.
- [4] Buffa, M., Lebrun, J., Kleimola, J., Larkin, O. and Letz, S. 2018. Towards an open Web Audio plug-in standard. The WebConf2018 (WWW'18), Mar 2018, Lyon, France.
- [5] Kleimola, J. and Larkin, O. 2015. Web audio modules. In Proc. 12th Sound and Music Computing Conference (SMC 2015), Maynooth, Ireland.
- [6] Letz, S., Orlarey, Y. & Fober, D. 2017. Compiling Faust audio DSP code to WebAssembly. In Proc. 3rd Web Audio Conference (WAC 2017), London, United Kingdom.
- [7] Jillings, N. & al. 2017. Intelligent audio plugin framework for the Web Audio API. In Proc. 3rd Web Audio Conference (WAC 2017), London, United Kingdom.

⁴ <http://wasabihome.i3s.unice.fr>