Making a guitar rack plugin -WebAudio Modules 2.0

Michel Buffa, Pierre Kouyoumdjian, Quentin Beauchet, Yann Forner, Michael Marynowic

Making a guitar rack plugin - WebAudio Modules 2.0

Quentin Beauchet, Pierre Kouyoumdjian, Michael Marynowicz,
Yann Forner, Michel Buffa
Université Côte d'Azur
(firstName.lastName)@univ-cotedazur.fr

Figure 1: A composite plugin that acts as a host

ABSTRACT

WebAudio Modules 2.0 (aka WAM) is a standard for developing interoperable host and plugins. that can be written using different development approaches (JavaScript, build systems, TypeScript, FAUST, Csound, front-end frameworks, etc.). In addition, the WAM standard includes the notion of group of plugins, in order for example to send MIDI events to a group of plugins or to unify their parameters in a single set. The demo we propose is about a plugin that acts as a host, that exploits the WAM group feature of the WAM SDK and manages a set of plugins. This plugin regroups guitar pedal effects and tube amp simulators plugins to propose a large set of sound presets organized in banks. It can be loaded by any WAM compatible host, be it a single HTML page with a few lines of JavaScript or a complete DAW. Both cases will be demonstrated.

Web Audio Conference WAC-2022, July 6–8, 2022, Cannes, France.

1. Introduction

In 2015 Jari Kleimola and Oliver Larkin created Web Audio Modules (WAM) [3], a standard for creating reusable and interoperable plugins implemented as WebComponents, the core of which was written in C/C++.

This initiative was aimed primarily at developers of native plugins (i.e. VST plugins). Since then, this standard has evolved in a new "Web Audio Modules 2.0" (WAM2) version [1], taking into account feedback from developers of the first version and the evolution of web Ssandards (with WebAssembly, WebComponents, WebMidi etc.). The WAM distribution is open source and distributed as four GitHub repositories and as npm modules (MIT License)[1]. One of these repository, the wam-example github repository, contains different plugins and hosts, that cover different programming languages and

[1] https://github.com/webaudiomodules and https://www.npmjs.com/settings/webaudiomodules/packages

approaches. One of these examples is a proof of concept of a guitar pedalboard plugin written using the React front-end framework. This example is incomplete and the use of React adds an extra layer of complexity. WAM2 creators asked us to develop a simpler (pure JavaScript) example that can show some remarkable features of the standard, in particular the use of WAM groups and in particular, how to achieve parameter automation of a group of plugins that can be added/removed on the fly. Furthermore, as a large amount of our plugins are AudioWorklet / WebAssembly based, showing how the WAM standard could perform while automating a large amount of parameters without leaving the audio thread (with a host also written as an AudioWorklet) was something we wanted to experiment with.

2. Related works

Several people tried to develop guitar effects and amplifier simulations with WebAudio. However, the vast majority of these projects does not use reusable, interoperable plugins, or used proprietary formats just usable with their own host. Among the oldest open source projects are pedalboard.js[2] and the Guitar Pedal Board[4]. JS-Rocks[3] proposed in 2018 a set of guitar pedal effects and amplifiers (using Angular for the GUI). Other projects have been recently presented at different JavaScript developer conferences[4], but most of them use "rather" simple effects coded using high level WebAudio nodes, and do support more complex approaches involving Web Assembly, Domain Specific Languages (DSL) or ports of VST/C++ cross compiled code, like our version does, that open the doors to sample accuracy audio processing. We are also using a large set of WASM/AW based plugins in our pedalboard. The most advanced examples you can find are in commercial online DAWS (bandlab.com, soundtrap.com or ampedstudio.com -this last example uses WAMs that some authors of this paper have developed). In 2018 a pedalboard project used interoperable plugins [2] and has been presented at the WebAudio Conference in Berlin. It was built with an older version of the WAM standard. The demo we proposes, while having less GUI features, proposes more advanced capabilities.

Figure 2: Typical setup

3. The WAM pedalboard plugin

The WAM pedalboard plugin can be tried online[5] with an embedded audio player or using a real guitar and a sound card. It acts as a host and can handle a large set of plugins. According to the WAM specification, each WAM plugin is identified by a URI. From this URI it is possible to fetch metadata about the plugin (name, version, category, author, description, but also an image thumbnail). It is also possible to fetch the plugin code and to instantiate it using a single JavaScript dynamic import. Each plugin is a WebComponent. The WAM pedalboard uses a JSON configuration file that contains an array of URIs, each corresponding to the different WAM providers. Plugin providers can have an absolute URI (remote third party server) or a relative URI (in this case the plugins are located under the base URI of the pedalboard itself). Along the pedalboard we provide a local set of plugins, but we also set up a remote server with another set of plugins (source code also available in the wam-example GitHub repository[6]). When instantiated, the pedalboard queries all these WAM providers and builds an internal description of the available plugins. Thumbnails of these plugins are displayed on the top row, and can be sorted by categories using a drop down menu.

By clicking on thumbnails plugins are downloaded and instantiated and a plugin chain can be assembled (Fig 1.). Plugins in the chain can be dragged and dropped to change the order, or deleted. All controls are usable (knobs, switches etc.). When plugins are added or removed, the internal set of parameters of the pedalboard are updated and events are fired, that can be caught by the host (for updating the parameter automation menu, for example). Each plugin can be controlled using any MIDI control device (knobs, sliders, switches are all made with the webaudiocontrols library[7] that proposes a set of web components with midi learn features).

The pedalboard plugin also comes with a bank/preset manager. In the WAM specification, each plugin has a state (i.e the current value of its exposed parameters, that often is also the position of each know/slider/switch in its GUI). The state of the pedalboard is a composite set of all its internal plugin states, plus a description of its presets and banks. The order of each plugin chain in each preset is also part of the state of the pedalboard.

Like this, a DAW that would use one or more pedalboard plugin(s), will be able to save/restore its/their full state(s) with a song project, for example. At the time of writing we have about twenty audio effect plugins and three tube guitar amplifier simulators, but we are currently porting a much larger set of plugins we developed with a previous version of the plugin standard [5]. For the demo we plan to have more than seventy plugins available. Our proposed demonstration is a good test bench to assess the current limitations of Web Audio in terms of latency (driver, audio buffer size, sample rate frequency, etc.), audio effect sound quality, etc.

4. Settings for the demo

Our pedalboard can be played real-time with a real guitar. As of today, we recommend for the best experience to use MacOS and a low latency sound card. By using a real guitar, we propose to show that WebAudio plugins for guitarists are a reality and could be included in full-scale DAWs with a quality comparable to what we find in the native application world.

[2] https://dashersw.github.io/pedalboard.js/

[3] https://github.com/vitaliy-bobrov/js-rocks

[4] For example https://github.com/trys/pedalboard

[5] https://wam-bank.herokuapp.com/

[6] https://github.com/webaudiomodules/wam-examples

[7] https://github.com/g200kg/webaudio-controls

5. REFERENCES

[1] M. Buffa, S. Ren, O. Campbell, T. Burns, S. Yi, J. Kleimola, O. Larkin, "Web Audio Modules 2.0, an open Web Audio plugin standard", *In Companion Proceedings (Developer's track) of the The Web Conference 2022*, April 2022, Lyon, France

[2] M. Buffa, J. 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⟩

[3] J. Kleimola and O. Larkin. Web audio modules. *12th Sound and Music Computing Conference (SMC15). Maynooth, Ireland.*

[4] M. Buffa, M. Demetrio, and N. Azria. Guitar pedal board using WebAudio. *Web Audio Conference (WAC 2016). Atlanta, USA.*

[5] S. Ren, S.Letz, Y. Orlarey, R. Michon, D. Fober, et al. FAUST online IDE: dynamically compile and publish FAUST code as WebAudio Plugins. 5th Web Audio Conference, WAC 2019 Trondheim, Norway.