



HAL
open science

Embodied Virtual Instruments in Web-Based Multi-User VR: A Case Study with a 3D Drum Kit and Web Audio Modules

Michel Buffa, Marco Winckler, Adam Mir-Sadjadi

► **To cite this version:**

Michel Buffa, Marco Winckler, Adam Mir-Sadjadi. Embodied Virtual Instruments in Web-Based Multi-User VR: A Case Study with a 3D Drum Kit and Web Audio Modules. WAC 2025 - 9th Web Audio Conference, Nov 2025, Paris, France. Proceedings the Web Audio Conference 2025. <hal-05363346>

HAL Id: hal-05363346

<https://hal.science/hal-05363346v1>

Submitted on 13 Nov 2025

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.



Distributed under a Creative Commons CC BY 4.0 - Attribution - International License

Embodied Virtual Instruments in Web-Based Multi-User VR: A Case Study with a 3D Drum Kit and Web Audio Modules

Michel Buffa, Marco Winckler
University Côte d'Azur
firstname.name@univ-cotedazur.fr

Adam Mir-sadjadi
University Côte d'Azur
firstname.name@etu.univ-cotedazur.fr

ABSTRACT

WAM Jam Party is a web-based, multi-user VR application that enables participants to collaboratively build musical experiences by visually manipulating an audio signal chain composed of interactive audio nodes. Users can create, modify, and route sound by adding or connecting these nodes, each being a WAM (Web Audio Module) plugin with a 3D GUI. WAMs are often described as the web equivalent of VST plugins. Available 3D WAMs include synthesizers, samplers, audio effects, and pattern-based music note generators such as piano rolls or step sequencers, though traditional instrument metaphors have not yet been incorporated.

This article explores our efforts to enhance interactivity within this environment by introducing embodied virtual instruments — interfaces designed to replicate real-world instrumental interaction as closely as possible. Our initial focus is the development of a virtual drum kit (composed of several 3D meshes) that generates MIDI notes (including velocity), connected to a 3D drum sampler WAM and implemented using the Havok physics engine used in many AAA video games, for realistic collision detection with the virtual drumsticks that follow the user's movements. We aimed to simulate a drummer's physical gestures using VR controllers, while also addressing audio responsiveness, haptic feedback, and synchronized visual cues (i.e animating the 3D meshes during impacts) to reinforce the sense of embodied performance. During the demonstration, participants will be equipped with VR headsets to interact directly with the virtual drum kit environment. A secondary display will mirror the VR user's viewpoint, enabling observers to monitor the real-time immersive experience.

Keywords

Web Audio Modules, VR, Embodied interfaces, WebXR, Babylon.js, Havok, Musical Metaverse, WAM Jam Party



Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** owner/author(s).

Web Audio Conference WAC-2025, November 19–21, 2025, Paris, France.

© 2025 Copyright held by the owner/author(s).

1. INTRODUCTION

1.1 General Presentation

As the host to the "Musical Metaverse" — a virtual space where users can gather and create music together — WAM Jam Party [3] offers various contexts and approaches for collaborative music creation. Whether for performance or educational purposes, the need for realistic musical instruments as embodied interfaces quickly became evident.

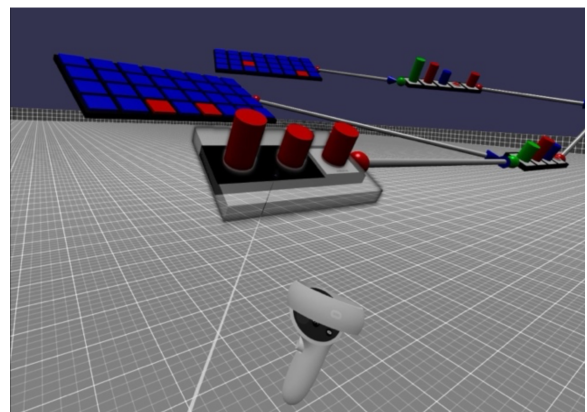


Figure 1: WAM Jam Party Audio chain viewed from a user headset

Embodied instruments offer several advantages: they reduce the learning curve for new users by relying on familiar physical metaphors (e.g., picking up drumsticks and hitting drums), they can serve educational purposes by replicating real playing conditions, and they enrich the overall experience by introducing natural movement, haptic feedback, and spatial cues.

This paper focuses on the design of a virtual drum kit in this environment — a playable 3D instrument implemented entirely in the browser, capable of responding to gestures with realistic (or as much as possible) timing, haptic, visual and sound response.

2. RELATED WORKS

In the gaming industry, several VR titles such as *Drums Hero*¹ and *PatchWorld*² incorporate embodied musical in-

¹https://store.steampowered.com/app/608370/Drums_Hero

²<https://patchxr.com/>

struments. However, these applications are not web-based, and their interaction models are often tightly coupled with proprietary ecosystems.

In the scientific literature, the *Musical metaverse playgrounds* by Boem and Turchet [1], and *Orchestra* by Dziwis, Von Coler and Pörschmann [5] present multi-user VR environments using WebXR and the Web Audio API. However, they use the Networked-Aframe 3D framework and do not integrate WAMs. Our approach differs in that it builds on Babylon.js and integrates inside a WAM environment.

Sequencer Party [2], WAM Jam Party’s 2D predecessor, served as a foundation for session-based, synchronized audio collaboration using WAMs. The current project presented here extends those capabilities to a VR context with gesture-based interaction.

3. DESIGNING THE VR DRUM KIT

3.1 Technical Stack and Architecture

WAM Jam runs entirely in a web browser and relies particularly on:

- **WebXR** API for VR headset/controller tracking and haptic feedback.
- **Babylon.js** for 3D rendering and WebXR integration.
- **Web Audio Modules (WAMs)** [4] to handle sound generation and transformations (high-level implementations based on the Web Audio API).

The drum kit consists in an object containing the different drum components as well as the drumsticks. The controllers and headset inputs and outputs are managed through an additional Handler class and debugging is made through a custom Logger which allows for debugging from within the application.

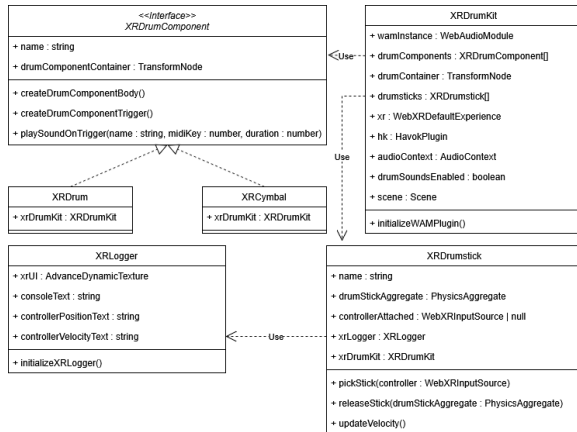


Figure 2: UML Class diagram for the VR Drum Kit

3.2 WAM API and physics interactions

Currently, a typical audio node that can be interacted with in WAM Jam Party consists in an imported WAM from which the usual 2D interface has been discarded. A custom 3D GUI is then generated through an automated process depending on the WAM parameters. This is made possible by a volumetric representation of the node parameters,

where you can manipulate cylinders or buttons to change the node’s parameters or behavior. For our embodied interfaces, this automated process obviously couldn’t be perpetuated as each instrument (at least while there are no prototypes for each instrument “family”) needs to be designed to respond as the user would expect it to. Therefore, we proceeded by importing a drum sampler WAM and implemented a custom 3D GUI on top of it to allow for collision detection.

To achieve this, we used the Havok Physics Engine which now comes with Babylon.js and is used in most of the AAA video games. We activated it and created triggers linked to the drums and cymbals meshes to allow for accurate collision detection. Triggers are naturally positioned only on the top surface of the drums (the drumheads), whereas cymbals have triggers distributed across their entire mesh and can be struck anywhere.



Figure 3: The Drum Kit 3D Model with Havok triggers bounding boxes highlighted

Each drumstick was assigned a physics aggregate that switches between an *Animated Motion* type when held and a *Dynamic Motion* type when at rest. Upon each relevant collision between a stick and a trigger, the WAM API is called to generate the appropriate MIDI Note ON and OFF events, based on the specific component that was struck. Collisions considered irrelevant—such as an upward strike from beneath a drum trigger—are ignored, as triggers are not intended to be activated from below.

3.3 Enhancing feedback

Detecting an upward or downward collision requires to know the linear and angular velocity of the controller when the collision occurred. This information is supposedly available through the WebXR API, but we still had to track at each frame the controllers position in order to compute their velocity from this information and the frame delta.

Additionally, velocity was required to adjust the produced sounds volume accordingly to the intensity of each collision caused by the user. Determining the right proportion between these two values with quite different scales requires further adjustments to reflect at best the volume intensity a given kinetic force would produce.

Haptic feedback through the controllers have been included and take velocity into account to produce an appropriate response for the user to sense when and how he is hitting the model.



Figure 4: View of the velocity and position information displayed on the logger while drumsticks are held

Visually, we are using a 3D model for each drum component which can be animated to show the drum skins trembling and the cymbals moving accordingly around their pole when hit.

3.4 Multi-user design

Starting from this standalone functionality, the challenge arises of how to properly synchronize user sessions when an instrument produces sound in an irregular and unpredictable manner.

Several approaches were considered, and we ultimately opted for a “looper”-based solution, using an output format similar to that of the piano roll WAM.

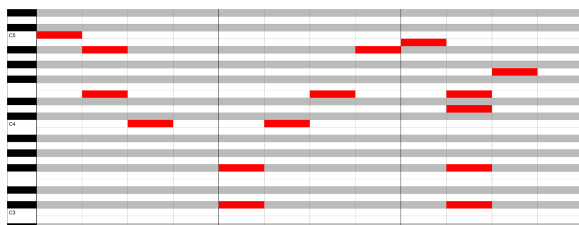


Figure 5: Piano Roll WAM Interface (notes per sample)

When recorded at 44.1 kHz, a user’s drum performance can be routed into a piano roll WAM, which captures the MIDI notes as a *Note Pattern* with high temporal precision. Once recorded, this pattern becomes part of the piano roll’s internal state, is synchronized across the network, and is integrated into the music heard by all participants. While this approach introduces a brief delay between recording and playback, it ensures consistent temporal alignment across all users. Furthermore, the extended automatically generated GUI enables the performer to fine-tune playback parameters as needed. The Piano Roll supports the WAM “Pattern Extension,” allowing users to load, save, and switch between patterns on demand.

4. FUTURE IMPROVEMENTS

While our model consists in a proof of concept for future embodied instruments inclusion inside WAM Jam Party, several enhancements will be brought to this project.

The custom GUI will be included inside a regular, automatically generated 3D WAM GUI to allow for standard

WAM parameters tweaking as well as positioning inside the audio chain. This will allow for user and scalability tests directly inside the main application for further fine-tweaking of the velocity impact on sound intensity. Any improvement to the 3D model could also enhance performances and prevent any computation overloading which might occur while scaling (more users, more complex installations...). Additionally, the position of each drumstick hit on the drumhead can be used to modulate parameters such as filter and pitch, producing subtle variations in sound—much like an acoustic drum kit.

Eventually, we consider adding MIDI pedals support for the kick and hi-hat, as well as changing the sound envelope of generated sounds to account for additional characteristics (impact precise location, angle of collision...).

5. CONCLUSIONS

By combining Web Audio Modules with the Havok physics engine, we introduced a way into WAM Jam Party towards new and perhaps more natural and immersive musical experience for the users. The combination of Havok physics and WebXR allowed for a satisfying audio, haptic, and visual feedback which are yet to be improved and fine-tweaked. This work demonstrates the feasibility and potential of embodied musical instruments based on WAMs in the browser, paving the way for further research and user studies in WAM Jam Party.

6. ACKNOWLEDGMENTS

This work was supported by the French government, through the France 2030 investment plan managed by the Agence Nationale de la Recherche, as part of the UCA DS4H project, reference ANR-17-EURE-0004.

7. REFERENCES

- [1] A. Boem and L. Turchet. Musical metaverse playgrounds: exploring the design of shared virtual sonic experiences on web browsers. In *2023 4th International Symposium on the Internet of Sounds*, pages 1–9, 2023.
- [2] M. Buffa and T. Burns. Real-Time Collaborative Music Creation on the Web: exploiting Web Audio Modules for Interactive Performance and Composition. In *30th IEEE Symposium on Computers and Communications (ISCC), workshop Next-Generation Multimedia Services at the Edge (NGMSE): Leveraging 5G and Beyond*, Bologna, Italy, July 2025. IEEE,.
- [3] M. Buffa, A. Hofr, and D. Girard. WAM Jam Party: Using Web Audio Modules in the Musical Metaverse. IS2 2024 - IEEE International Symposium on the Internet of Sounds 2024 / 1st IEEE International Workshop on the Musical Metaverse (IEEE IWMM), Sept. 2024. Poster.
- [4] M. Buffa, S. Ren, T. Burns, A. Vidal-Mazuy, and S. Letz. Evolution of the web audio modules ecosystem. In *Web Audio Conference 2024*. Zenodo, 2024.
- [5] D. Dziwis, H. von Coler, and C. Pörschmann. Orchestra: A toolbox for live music performances in a web-based metaverse. *Journal of the Audio Engineering Society*, 71:802–812, 11 2023.