Perspectives on Practical Implementations in the Web-based Musical Metaverse

Damian Dziwis*, Alberto Boem†, Matthias Nowakowski‡, Luca Turchet†,
* Detmold University of Music
* TH Köln - University of Applied Sciences
* Technische Universität Berlin, Germany
Email: damian.dziwis@th-koeln.de
† Department of Information Engineering and Computer Science University of Trento, Trento, Italy
Email: alberto.boem@unitn.it, luca.turchet@unitn.it
‡ Musicological Seminar, University Paderborn, Paderborn, Germany
Email: matthias.nowakowski@uni-paderborn.de

***Abstract*—The technical advances that immersive environments are experiencing through virtual and augmented reality, combined with the global networking possibilities offered by the internet, are making shared virtual environments increasingly attractive as a platform for musical practice in the so-called Musical Metaverse (MM). The growth of dedicated tools and systems, especially for web-based metaverse environments, has made the web-based MM particularly relevant in recent years. Browsers and web technologies have particular advantages for live performances and sound installations in the MM. They are platform-independent and can be experienced on a wide range of devices, including PCs, smartphones, and standalone head-mounted displays. Users do not need to install additional software; a link is sufficient to participate in the next metaverse performance. However, all the advantages of web-based MM applications also come with various challenges and limitations, such as incompatibilities and inefficiencies. Drawing on extensive experience in development, research, and artistic practice in the realm of the web-based MM, the authors of this paper address the challenges encountered in the development of web-based MM applications and explore avenues for future advancement to bridge the identified gaps.**

***Index Terms*—Musical Metaverse, Web Technology, WebXR, Web Audio, Networked Music Performances**

I. Introduction

With the growing possibilities that metaverse technologies offer for musical practice, the resulting interest and the number of artistic explorations in the field of the Musical Metaverse (MM) [1] are also increasing. The first musical performances in shared virtual environments (SVEs) [2] date back to the early days of virtual worlds inspired by the idea of the metaverse [3], most notably the ones performed in Second Life [4], [5]. At the beginning of the 2020s, it was primarily the restrictions imposed by the COVID-19 pandemic that led to an increase in musical performances being held online [6]. In addition to live streams with two-dimensional video and audio, the increasing possibilities offered by immersive technologies such as virtual reality (VR) combined with SVEs for musical art practice were also explored [7]–[10]. Nowadays, various musical performances have taken place on platforms such as VRChat [11], Mozilla Hubs [12]—now Hubs Foundation [13] (Hubs)—, Somnium Space [14], or specialized VR applications for music such as PatchWorld [15] and CSoundMeta [16].

In particular, web-based and open-source platforms such as Hubs provide several key advantages that make them especially beneficial for the metaverse. First, the idea of the metaverse as a group of networked virtual worlds [17] in the World Wide Web (WWW) is supported by the fact that the WWW is already one of the most prevalent services on the internet. This provides immediate benefits: broad distribution, established infrastructure, and universal accessibility for all internet users. Unlike proprietary platforms such as VRChat or Meta Horizon [18], the web-based metaverse is decentralized and accessible to everyone without control by a company. There are no restrictions on who can develop technologies and content for it or who can access them (with the exception of living-areas with restricted and censored internet access). Second, the basic development resources for creating content for the WWW, such as programming languages for browsers (e.g., HTML, JavaScript), are generally open source and/or free. The same applies to many other necessary tools, including browsers, server software, as well as libraries and frameworks for developing content. Third, with open standards such as the WebXR Device application programming interface (WebXR API) [19], immersive content can be created on the WWW for virtual and augmented reality (AR) devices, such as head-mounted displays (HMDs). The Web Audio API (WAA) [20] also has given browsers the capability to process real-time audio, including synthesis, analysis as well as encoding and decoding spatial audio [21], [22]. These technical possibilities further solidify the concept of the metaverse as the next evolutionary stage of the WWW in the Web3 era [23].

The advantages of web-based technologies have led to an increasing number of artistic realizations in the MM. In addition to general-purpose tools for SVEs on the web, various tools and instruments have been developed specifically for musical practice within the web-based MM [24]–[32].

The work we present here is based on the authors' experience on developing web-based MM systems informed by practice-led research methodologies [33]. Our insights emerge from three years of creating artistic realizations, staging 6 public performances and 2 installations with 15-50 concurrent users, and developing and maintaining tools that use web-based metaverse technologies (see Sec. II). This paper contributes a systematized practice-based knowledge organized through three complementary stakeholder perspectives: developer (Sec. III), creator (Sec. IV), and audience (Sec. V). These perspectives were captured through post-performance discussions, demonstration sessions, and ongoing development issue tracking. Based on systematic cross-platform testing across Chromium-based browsers, Firefox, and HMD browsers, we analyze issues such as incompatibilities and inefficiencies and address possible starting points for further development and improvement of the current state of the art. Our analysis addresses an under-documented area in the literature by foregrounding the experiential aspect of building and deploying web-based MM applications, complementing existing technical and theoretical work with grounded practitioner knowledge.

II. Practical Implementations

The authors' experience includes a variety of practical implementations from the research and development of tools, experiment environments and artistic experiences in web-based MM environments. These environments were developed using the A-Frame [34] and Networked-Aframe (NAF) [35] libraries. These are a well-established foundation for web-based metaverse environments, the same one used by Mozilla in the design of Hubs.

Some of the tools developed are included in the Orchestra toolbox [28]. It contains: PdXR [31], a metaverse implementation of the Pure Data (PD) language [36], the Aframe-Kinectron component [37] for volumetric video streaming in live performances [9], the VERSNIZ live coding environment [30], a custom implementation of the IBNIZ language [38], and two other live coding environments [29] based on the Bytebeat concept [39]. Another project involved the development of StackBeat L.O.V.E. [32], a visual live coding language for the MM. For a number of studies [26], [27], five environments were implemented with collaborative instruments using the Tone.js [40] and Essentia.js [41] frameworks, as well as PdXR.

In addition to developing and researching dedicated tools, the authors evaluated such tools by using them to create a series of performances, sound installations and experiment environments for the web-based MM. These are described hereinafter.

Performances:

Fig. 1. A screen capture showing the OCANIA multi-instrument metaverse environment, which involved an implementation of the PdXR and StackBeat L.O.V.E. live coding environment.

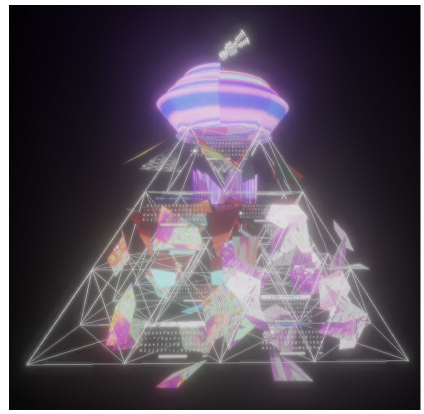


Fig. 2. A screen capture showing the T R I P L O R A T I O N live coding performance by Markus van Well and Damian T. Dziwis. The experience uses the VERSNIZ metaverse live coding implementation of the IBNIZ language.

- **OCANIA (2025)**: A multi-instrument metaverse environment (see Fig. 1). It encompassed shared virtual instruments [27] developed in PD using the PdXR system, as well as the StackBeat L.O.V.E. live coding language. It was first used in 2025 for an improvisational concert together with students of the Distributed Performance Course led by Sarah Weaver, professor in the Music Technology Program at the Steinhardt School of Culture, Education, and Human Development at New York University.
- **T R I P L O R A T I O N (2025)**: An audiovisual installation-performance by Markus van Well and Damian T. Dziwis, with a "worldbuilding" approach that creates a nonlinear experience through audience reception [42]. The title is a neologism composed of the words "triangle" and "exploration," which are used as metaphors for a live coding performance that incorporates algorithms for fractal compositions with Sierpinski triangles (see Fig. 2)[1]. That performance used the VERSNIZ live coding environment.
- **Mnemonic Garden (2023)**: A live coding performance[2], using the MetaBeat live coding environment [29]. The performance was premiered at the "die digitale" festival in Düsseldorf (Germany)
- **The Entanglement (2022)**: An audiovisual, volumetric, live performance realized as a telematic [43] improvisation of a quantum computer composition, staged in the

[1] T R I P L O R A T I O N (excerpts), https://www.youtube.com/watch?v=aJilUDWpKn0, accessed: 2025-06-16

[2] Mnemonic Garden, https://www.youtube.com/watch?v=HJGbvXs45sk, accessed: 2025-06-16

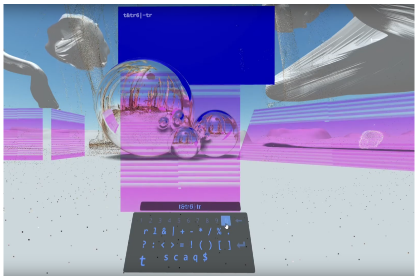


Fig. 3. A screen capture showing the “Mnemonic Garden” live coding performance. It used the “MetaBeat” metaverse implementation of the Bytebeat concept.

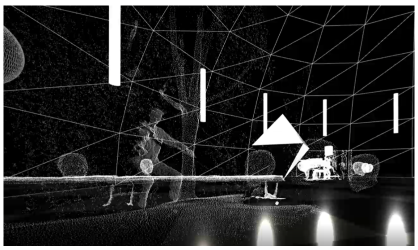


Fig. 4. A screen capture showing the “The Entanglement” networked music performance in the metaverse. The image shows one of the violin players being live-streamed as a volumetric point cloud, surrounded by visitors displayed as point-cloud-head avatars.

metaverse [9][3]. It uses Microsoft Kinect v2 3D cameras to live-stream two violinists performing in a virtual environment (VE) as 3D point clouds (see Fig. 4), along with their sound as virtual sound sources. The volumetric streaming is implemented as an A-Frame component for the Kinectron API [44]. The project was realized as part of the “Telematic Artistic Research Residency” at the Institute for Computer Music and Sound Technology at Zurich University of the Arts (ZHdK). It was created in collaboration with Frantz Loriot and HannaH Walter, who played the violin, as well as with the technical support of Joel De Giovanni, Martin Fröhlich, and Patrick Müller.

Installations:

- **STATIC (2023)**: A VR and AR interactive installation for multiple users (see Fig. 5). It was created using the PdXR system, adding customizations in the graphical user interface objects and adaptations to allow using VR and AR HMD systems.

Collaborative Instruments:

- **It Takes Two (2024)**: A group of three collaborative virtual instruments (see Fig. 7) that need to be used by two performers [27][4]. The three instruments explore specific features of SVEs, such as spatial audio, data sonification, and embodied interactions. The first is the “Spatial Instrument”, where a performer acts as an instrument player, responsible for creating a sequence of tones, and the second acts as a spatial sound conductor, moving the sound in space. The second is the “Sonification Instrument”, where the position in space of the two performers and their movements drive an FM synthesizer. The third is the “Body Instrument”, where users use each other’s avatars as a drumming surface. By changing their avatar’s size, performers can also affect the sound they emit. All three instruments were developed using PdXR and are best used through compatible HMDs.
- **Musical Metaverse Playgrounds - Synth Land (2023)**: An SVE (see Fig. 6) where participants can spawn a potentially infinite number of synthesizers in space and control their parameters through a 3D User Interface [26][5]. This application was developed using NAF for the networked VE, while for audio synthesis Tone.js was used. The experience is compatible with both, desktops and HMDs supporting WebXR.
- **Musical Metaverse Playground - Sound Space (2023)**: A SVE (see Fig. 6) where musicians can visualize and animate 3D objects with their voice [26][6]. The voice of each peer is captured in real-time by a microphone (whether it is in a laptop or the one integrated in a HMD) and a series of features are extracted. These are mapped to features of the VE. For the real-time analysis a combination of Tone.js and Essentia.js was used.

[3]The Entanglement, https://www.youtube.com/watch?v=yKGc8dYJSLs, accessed: 2025-06-16

[4]It Takes Two, https://youtu.be/d3W0ZxfDjMY

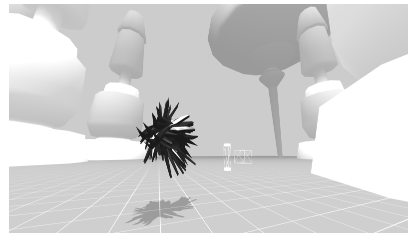


Fig. 5. A screen capture showing the virtual and augmented reality metaverse installation STATIC. It was created using the PdXR system.

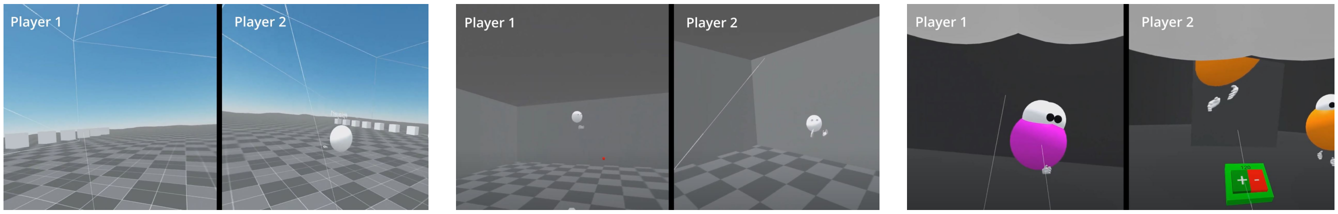
III. Developer Perspective

Developing technologies for use in the the MM, such as for creating virtual musical instruments (VMIs) [45] or networked music performances (NMPs) [46], poses several technical challenges. These include dealing with latencies when streaming audio and video in real time, as well as synchronizing data, and the lack of standards for these types of applications [47].

In particular, developing web-based technologies for the MM brings additional web-specific challenges and limitations. Although web-based applications offer advantages such as easy accessibility due to platform independence among browsers and end devices, as well as the elimination of the need to install additional software, these advantages pose

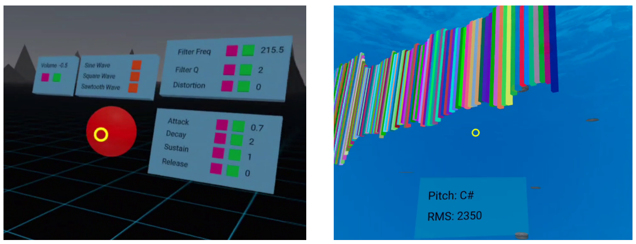
[5]Synth Land, https://youtu.be/FYyrrHzwGqE?si=Gxg6IbUMbHh7zDzs

[6]Sound Space, https://youtu.be/Pix2LNEWe8g?si=qe3ScUefQENHqPjf



(a) (b) (c)

Fig. 6. Three screenshots of the instruments of “It Takes Two”: (a) Spatial Instrument, (b) Sonification Instrument, c) Body Instrument.



(a) (b)

Fig. 7. Two screenshots of playing sessions using (a) Synth Land, and (b) Sound Space.

additional challenges that are generally easier to solve with native applications for specific target devices.

Based on the practical implementations (see Sec. II) realized by the authors, the following list of challenges and limitations in the development of web-based MM applications emerges.

A. *Cross-browser compatibility*

A significant challenge in the field of web development is cross-browser compatibility. The wide variety of browsers (e.g., Google Chrome, Mozilla Firefox, Microsoft Edge, Apple Safari, and more), operating systems (Microsoft Windows, Apple MacOS, Linux, Android, iOS, and more), as well as end devices (desktop and laptop PCs, smartphones, tablets) pose a problem for web developers to make web applications look and run consistently on all these end devices [48]. Especially relevant for our discussion is the growing market of extended reality (XR) standalone HMDs with integrated browsers compatible with the WebXR API, which creates another layer of complexity. Standalone commercial HMDs come with integrated browsers that support WebXR API, including the Chromium-based Meta Quest Browser for Meta Quest devices, the open-source browser Wolvic (formerly Firefox Reality) supported on many devices including Lynx and Pico, and a dedicated version of Safari for the Apple Vision Pro.

While the WebXR API serves as the fundamental interface enabling web applications to access HMD hardware functionalities through standardized browser implementations, it should ensure cross-platform compatibility. However, each platform implements distinct approaches, especially with Android-based devices (e.g., Meta Quest, Pico, Lynx) and Apple exposing different functionalities and limiting others, especially in terms of VR and AR support, along with the interaction techniques and input modalities available to the users, such as mouse/keyboard, VR tracked controllers, hand and eye tracking.

Along with all of the technologies for VR/AR, another important element is how audio is handled in web browsers. The WAA serves as the foundational technology for creating immersive audio experiences in XR applications, and it shows issues similar to the ones of the WebXR API. Quest browsers implement the WAA through the Chromium audio engine with platform-specific optimizations for mobile VR hardware, while the Wolvic browser delivers an implementation through its Gecko engine foundation (while having also a Chromium version), adapting to available audio APIs on different headsets. Differences in the execution order and priority of audio threads across browsers can cause disruptive artifacts, such as dropouts and jitter, when using real-time audio in one browser, while audio processing works flawlessly in another browser. This has especially an impact on how immersive audio is handled by each browser [22].

The lack of full support for web standards and APIs (especially WebXR), even in widely used browsers, as well as the different implementations and prioritization of critical technologies, such as JavaScript engines and audio threads, leads to differences and issues in displaying and interacting with developed web applications. Despite the goal of achieving compatibility and integrated support across popular browsers, remaining cross-browser incompatibilities can quickly lead to situations in which a web-based MM application functions perfectly on one browser or device, yet becomes unusable on another. This can result in additional, and sometimes extremely high, levels of testing and development work.

During the development of the VERSNIZ live coding environment, a cross-browser incompatibility issue in the rendering of textures resulting from the IBNIZ code remained unsolvable. While Chromium-based browsers (e.g., Chrome and Edge) can render and use textures resulting from the IBNIZ code as intended (see Fig. 2), the Mozilla Firefox browser fails to do so when using the texture as a source for the designated 3D geometries, despite a valid and error-free implementation. A seemingly unrelated change to the sound synthesis engine allowed Firefox to use the texture as intended but made the sound synthesis engine unusable.

Complex audio algorithms processed with PdXR in the browser resulted in quicker dropouts and jitter when using Chromium-based browsers compared to Firefox. This suggests differences in the processing and/or prioritization of audio threads in Chromium. Therefore, adapting or simplifying complex PD algorithms may be necessary when aiming for problem-free audio processing in all browsers.

Given these cross-browser compatibility challenges we observed with custom-made tools such as PdXR, we have also explored other ways of implementing cross-compatible audio processing. An alternative strategy involved using the WAA directly through established web audio libraries, such as Tone.js, which can in theory provide more consistent cross-browser performance. This approach was explored in the two projects of the "Musical Metaverse Playgrounds" project [26]. While this direct approach achieved consistent audio processing results across different browsers and platforms (including both PC and HMD environments), it required significantly more time and resources dedicate to development, particularly when integrating Tone.js with the NAF, especially for synchronization purposes.

Choosing between higher-level tools like PdXR with its simplified development workflows and ready-made NAF-compatible modules, as well as lower-level approaches that can provide better cross-browser consistency, creates a demand for more extensive custom integration work. This is due to the fact that every new synth or audio process needs to be tightly integrated with the logic of NAF.

Ultimately, these two examples reflect a more broader challenge: balancing development efficiency with cross-platform compatibility. Here the choice between advanced functionality and reliable performance across diverse browser implementations remains a critical consideration for web-based multimedia applications.

B. Browser security features

In addition to cross-browser compatibility issues in MM web application development, individual browser settings and security restrictions pose an additional technical barrier. In the context of the web-based MM, the security feature that restricts sound and video autoplay is especially relevant. This browser feature is intended to protect users from unintentionally being disturbed by unwanted sounds and videos from websites. Browsers implement various permission policies to prevent this. As a rule, sound autoplay only works after the user has interacted with the website.

However, this is not uniformly implemented across browsers, which leads to serious problems, especially in complex audio environments, such as those usually found in MM applications. Specifically, if audio engines or sound spatializers must be loaded and executed before user interactions, subsequent execution of the audio context due to those interactions can cause complications. Although changing the sound autoplay settings can usually solve the problem, this cannot be expected from users for an MM application. Therefore, in practical implementations, it is essential to ensure that the applications work smoothly for the audience automatically without requiring any action from the user.

The most recent version of PdXR, revealed an issue with the PD sound engine not reliably starting on Chromium-based browsers (e.g., Google Chrome and Microsoft Edge) when used in the OCANIA metaverse environment (see Sec. II). This problem did not occur in Firefox, and it could not be completely resolved in Chromium browsers, even when forcing user interaction before starting the audio engine. Some cases still exhibited undeterministic behavior, resulting in the PD sound engine failing to start. In these cases, the only solution was to reload the page. This problem could be traced back to the autoplay policy; changing the settings to permanent autoplay reliably eliminated it. However, it was unreasonable to expect audience users to change their settings for a live performance event.

C. Cross-platform compatibility

Developing a web-based immersive musical application requires accommodating a wide range of devices from desktop computers with mouse and keyboard input to smartphones with touch interfaces, standalone VR/AR HMDs with hand tracking, to PC-VR systems with dedicated controllers. Each offers to users fundamentally different experiences in terms of display resolutions, processing power, sensory feedback mechanisms, and interactions. Whereas the first ones require a screen in front of the user and buttons that are arbitrarily mapped to the avatars movements (such as a keyboard or buttons shown on the display), HMDs have the entire field of vision enclosed and hand movements are directly responsible for the perception and interaction with the VE.

Developing musically meaningful interactions for these applications is complicated by the fundamental differences between input and display technologies across platforms. Each device type presents distinct limitations that directly impact how users perceive and interact with musical interfaces and virtual instruments.

By using HMDs musicians can perceive virtual objects, such as instruments or mixing consoles, as complete three-dimensional entities. Moreover, through hand tracking or input controllers user can directly manipulate such objects directly as they will do in their physical reality [45]. Compared to this rich and intuitive type of interaction, desktop computers or laptops, further restrict available input methods and limit musical expression: computer keyboards and mice lack the nuanced, continuous control essential for expressive musical performance. While mobile devices with touchscreens provide direct tactile feedback and simultaneous multi-point control, screen size and performance is further limited in this class of devices. As a result, ensuring interaction compatibility with keyboard/mouse and touchscreens, can lead to potentially losing expressive potential and musical authenticity compared to HMD-only implementations.

These constraints can lead to two primary design strategies, each with distinct compromises between cross-platform compatibility and expressivity. The first approach prioritizes cross-platform compatibility through interaction abstraction. For instance, the “Musical Metaverse Playgrounds” project [26] was designed for both PC-based browsers and HMD browsers using a symmetric interaction model: users employ a point and select mechanism across both platforms, pointing at objects with a cursor (controlled by mouse movements on PC, and gaze in VR) and confirming selections with simple actions (left-click on PC, controller trigger in VR). Locomotion follows similar principles, using WASD keys on PC and joystick controls in VR.

While this approach provides a straightforward and seamless experience between devices and types of browsers, it delivers a significantly simplified musical interaction model. Basic functions like spawning objects or sending a Note ON/Note OFF events off work effectively, but controlling continuous parameters (such as adjusting a filter’s cutoff frequency) becomes problematic.

The second approach prioritizes platform-specific optimization, accepting reduced cross-platform effectiveness. In “It takes two” [27], the focus was primarily on embodied interactions optimized for HMDs and tracked controllers, leveraging the full potential of spatial awareness and hand gestures for musical expression. While the experience remains technically accessible on PC with mouse and keyboard input, this method does not reflect the immediacy of the interaction allowed by a VR systems.

However, this can be also seen in the development phase. Even if WebXR-compatible browsers on PCs include an emulator for testing HMD applications, this can lead to a disconnect between developers and users, and can extend development cycles if developers remain in lower-fidelity on-screen testing for too long.

D. Software sustainability

In web development, many development resources are open source/format. This includes languages (such as HTML, JavaScript, CSS, and PHP) and popular frameworks and libraries (such as React and Node.js). The web-based metaverse follows the same open-source approach, with frameworks and libraries like A-Frame and NAF, as well as entire platforms like Hubs, being freely available to both developers and users.

Web-based MM applications generate additional demand for specialized resources. Immersive sound reproduction requires a sound spatializer with room simulation. Developing virtual instruments requires specialized tools for sound synthesis and interaction. NMPs with acoustic instruments require adequate network technology for low-latency audio and video streaming. New performance practices, such as live coding, require the porting and adaptation of programming languages for the MM. Recent developments have made a variety of necessary resources available for the development of experiences for web-based MM experiences, also as open source. This creates broad access to developments and resources within the web-based MM, enabling the efficient evolution of the MM and its creations.

The fast pace of the internet, however, means that the software in this ecosystem is constantly advancing. This applies to browsers (see Sec.III-A) and their security features (see Sec.III-B) but also to the used resources such as programming languages and, above all, libraries and frameworks. In order to guarantee the technical functionality of developed tools, constant further development and adaptation to dependencies and resources used is also required here. This means a substantial development and maintenance effort that some open source projects from nonprofit and academic developers cannot cope with.

The same applies to critical resources required for the web-based MM. Of particular note is the demand for an audio spatializer with room simulation. Suitable open source solutions (unlike commercial/closed source solutions such as Atmoky or Superpowered) for the MM [22], like the Resonance Audio A-Frame port [49], have not been undergoing further development for many years. Resonance Audio for A-Frame serves as the audio foundation for the web-based MM tools of the Orchestra toolbox [28]. As the development of the Resonance Audio Spatializer and the A-Frame port has not been pursued for many years, only a fork within the Orchestra toolbox is maintained to ensure compatibility with current A-Frame versions. This backlog in this crucial dependency may lead to future issues with the Orchestra toolbox.

There is also a certain development backlog for other components within the Orchestra toolbox. PdXR is still working with a libPd [50], [51] audio backend based on an outdated Pure Data version (v. 0.48, 2017). A-Frame Kinectron for low-latency streaming for volumetric video in the web-based metaverse, is only compatible with a legacy version of Kinectron [44] (version 0). The live coding systems included in the Orchestra toolbox use an outdated version of the A-Frame Superkeyboard component [52] as the virtual keyboard. Also here, a forked version of the component is maintained in the toolbox to keep the component compatible with current A-Frame versions.

Although there are a variety of specialized resources for development and creation in the web-based MM, some of them have not been actively developed for a long time. Particularly in the case of complex components with a development backlog, there is a high risk of incompatibilities, especially if these contain dependencies that are themselves no longer being actively developed.

E. Server limitations

Although web-based MM applications and experiences eliminate the need for creators to distribute their applications through stores or custom solutions and eliminate the need for users to install additional client-side software, they impose a constraint on developers by requiring them to host applications on servers. Setting up and maintaining server software can be challenging, especially for developers and artists with little experience in system administration. It also imposes additional costs on development and artistic projects.

Hosting web-based MM projects requires renting servers and domains, as well as significant system administration and maintenance efforts. Setting up the popular open-source metaverse environment Hubs requires knowledge of Kubernetes container installation, management of firewalls and certificates, and more. A compatible server architecture for the NAF library, which is widely used in web-based MM tools, requires the installation of a Node.js environment with web and RTC servers and also the management of firewalls and certificates. Alternatives to self-hosting, such as the free hosting offered by the Mozilla Foundation for Hubs, have been discontinued. The same applies to Glitch [53], a popular free hosting option with minimal installation requirements for NAF-based metaverse projects.

However, free hosting options are usually limited in terms of available resources such as disk space, bandwidth or accessible server hardware. Due to these limitations, e.g. platforms like Glitch only provided 200 MB of disk space and 512 MB of RAM in their free options, these services are only scalable to a limited extent. This affects the amount of data and concurrent users with corresponding WebSocket and WebRTC connections. The latter is particularly noticeable in terms of the robustness of the experience with a large number of users. Having too many users connect simultaneously can affect the performance of data exchange and lead to synchronization problems between users.

During the performance in the OCANIA environment hosted on Glitch (see Sec. II), synchronization issues with data occurred occasionally when there were more than 10 concurrent users. These issues even caused a temporary server crash during the premiere. In conclusion, to ensure a robust experience during live performances with many concurrent users, it is not recommended to use servers with severely limited resources.

IV. Creator perspective

Similar to developing basic technologies, systems, and tools for the MM, creating and implementing practical musical applications, such as virtual musical instruments, installations, and VEs for live performances, also presents challenges and technical limitations [54]. While the possibilities for creating musical applications for the MM are potentially increasing and simplifying with the development of dedicated systems and tools for implementing such experiences, there are still many limitations and technical challenges, especially in the web-based MM.

The authors' process of creating artistic experiences such as performances and installations (see Sec. II), revealed a number of challenges and limitations specific to the web-based MM, which are described in the following subsections.

A. Web-app efficiency

Although VEs offer seemingly endless possibilities for the visual and sound design of artistic experiences in the MM, the ability to create is quickly limited by the performance of the resulting applications. Since WebGL applications are less performant than native OpenGL applications [55], low frame rates and audio rendering dropouts quickly become noticeable, especially with regard to immersion and XR devices. Consequently, many compromises must be made when creating web-based MM experiences. When creating the VE, artists must limit the amount of generative algorithms, 3D objects, and media, such as textures and videos, that they use. Sound synthesis and spatialization may need to be reduced to a few sound sources to prevent jitter and dropouts. These restrictions not only lead to artistic limitations, but can also result in an impression of the experience that does not reflect the current state of the art, particularly when compared to native applications created with popular game engines like Unreal Engine or Unity.

B. Available hardware

In addition to the limitations caused by the lower performance of web-based immersive applications due to inefficient web rendering (see Sec.IV-A), the hardware commonly used by artists and audiences further limits the creation of complex virtual content in the web-based MM. Similar to current video games, rendering virtual 3D content requires high-performance devices with powerful graphics processing units (GPUs). However, unlike for the gaming community, this cannot be assumed for the art and music-savvy users. Less powerful mobile devices, such as smartphones or standalone HMDs, must also be considered, especially if the whole bandwidth of the platform-independent web-based MM needs to be utilized. This necessitates further compromises and limitations in the design of visual and auditory 3D content.

V. Audience perspective

With numerous practical implementations in the web-based MM (see Sec. II), the authors can draw on extensive user feedback. In addition to findings from conducted studies [26], [27], various challenges and limitations have also emerged from informal audience evaluations of performances and installations.

A. Limited experience

Distributing MM performances on the WWW is a pragmatic way to reach a large audience. However, there are different levels of what one can expect from the audience, and what the audience can expect from the performance. Some may not be tech-savvy, so they need a low technical barrier to entry. To access a web-based MM experience on their own, visitors need to know at least how to use a web browser. While this can typically be assumed by internet users, navigating a VE in a browser can be challenging for many inexperienced visitors. Using arrow keys or WASD to move around in a 3D environment while simultaneously dragging the mouse to control the viewing perspective is generally not intuitive for users with no previous experience with 3D video games or other similar applications. Getting used to the navigation then typically requires a training period that cannot effectively be anticipated during a performance. Since creating MM experiences for a highly technical, specialized audience would result in performances with just a few visitors, inexperienced audiences must always be considered and included in the process.

Using HMDs without prior experience can be challenging, especially when they are provided to the audience in a local installation situation or in a hybrid concert format. In addition to general issues, such as motion sickness [56] and isolation from instructions due to the use of HMDs and headphones, the ad hoc use of an HMD with controllers or hand tracking can quickly overwhelm users without experience in VR or similar 3D applications. Unintuitive controller operation, such as stick control or teleportation in combination with 6-DoF motion tracking, can quickly lead to confusion, disorientation, and additional motion sickness. A lack of experience with the hardware can lead to incorrect controller or HMD button inputs (e.g., when adjusting the headset position), which can minimize or terminate applications, making it difficult for users to continue the experience independently [57]. During the presentations of the VR/AR metaverse installation STATIC (see Sec. II), many users had little to no experience with HMDs. Similarly, when the two "Musical Metaverse Pla-grounds" systems were presented in public demos, it became evident that strategies are required for implementing and introducing such experiences.

B. Guiding audience in space

An experience in the MM has virtually no bounds in space and movement is not just confined to a solid underground. Enabling the audience to move anywhere differs vastly from previous real life experience even if a performance has a more open conception, e.g., distributed musicians in a given space. As observed, e.g. in OCANIA (see Sec. II), the movement of the performers differs from the movement of the audience as they mostly adopt a stationary hearing mode. Performing in an unlimited space might sometimes require visual cues to lead the audience. Creating an interactive and walkable environment as part of a piece might be a call to action for the audience and also a viable contribution to the overall artistic display by the performer. The stage becomes an active element of the performance, instead of a static display piece.

The immobile audience behavior could be explained by the lack of a interactive elements that may structure the narrative for each member [58]. As long as nothing is immediately interactive for the audience, they mostly do what they would do in a personal setting in front of a display, tablet or mobile device. Designing the space for a static display is not just a developmental issue, but also one of audience experience. Foveal and peripheral sight are directed differently in HMDs. Having a disconnected field of view (i.e., a display with something different appearing outside the edges) might leave the viewers interaction more passive [59].

VI. Discussion and Conclusion

In this paper we presented a comprehensive analysis of the current state and challenges of web-based MM applications, drawing from our extensive practical experience in developing tools, creating artistic works, and presenting performances in SVEs (see Sec. II). Through examination of multiple perspectives (such as the developer, creator, and audience) we showed that while browser-based applications offer unprecedented platform independence and accessibility, eliminating the need for specialized software installation, these advantages create substantial technical challenges and limitations that can significantly impact the artistic and functional quality of MM experiences.

From a developer's perspective (see Sec. III), implementing and designing tools for web-based MM applications presents a paradox, since the major strengths of browser-based applications becomes their biggest challenges. While browser-based applications enable the realization of platform-independent experiences that are highly accessible to users (since no additional software is needed and available hardware can be used), this flexibility also necessitates substantial additional development and maintenance efforts. The wide range of browsers, systems, and hardware can lead to multiple compatibility issues, regarding target platforms or input and output interfaces, when developing and using tools for the web-based MM (see Sec. III-A - III-C). Therefore, it is crucial that both browsers and XR devices better and more consistently integrate web standards and APIs such as WebXR and WAA. Additionally, the execution of multimedia content must be adapted to immersive web use-cases in terms of autoplay and security policies. The policies and features implemented for the convenience of browsing 2D web pages do not necessarily translate well to use-cases like web-based video games, WebXR applications, or MM experiences. Standardizing and seamlessly executing such immersive web content would ensure consistent MM application execution on all platforms. This would allow developers to focus on flawless system development.

The open-source nature of the web and its tools provides a high degree of transparency and availability of software resources. However, the lack of monetary support for non-commercial software can quickly lead to a backlog in the development of open-source projects, such as the ones developed used by the authors (see Sec. III-D). This can result in further incompatibilities with the required state of technology such as current browsers, dependency frameworks, and hardware. This is particularly true for academic projects, which often depend on consistent funding that can support their maintenance during their lifetime. Thus, the discussion arises about how open-source projects could be utilized more efficiently and how acquired resources could be invested in existing projects instead of redeveloping them. In particular, a state-of-the-art audio spatializer with room simulation would be a necessary resource for numerous web-based MM tools.

The hosting of web-based MM applications (see Sec. III-E) would benefit from ready-made installers or containers with server architectures and frameworks that can be deployed by the user or hosted by special services. Further developments in the MM, like using backend architectures based on approaches like Conflict-Free Replicated Data Type [24] could prove to be beneficial for robust data synchronization and data persistence, which are fundamental for MM applications.

From the perspective of a creator, limitations in performance and optimization of web-based immersive applications represent the main factors that limits artistic goals and realizations (see Sec. IV). Current developments in web technologies, such as in WebAssembly (e.g., for audio signal processing) [60] and WebGPU (for rendering 3D content) [61], ensure that immersive web content runs more efficiently in the browser. To benefit from these optimizations, priority of developers should be given to implement browsers, 3D rendering frameworks and libraries, as well as resources such as audio spatializers and sound synthesis engines incorporating these technologies. This would ensure maximum efficiency and performance in the resulting web applications and reduce performance bottlenecks for end users and design compromises for artists when creating them.

Informal feedback gathered from public performances and presentations of installations (see Sec. V) has shown that certain parts of the art-interested audience have either limited or no prior experience with immersive technologies or suitable technical equipment for participation in the MM (especially HMDs). The lack of high-performance hardware or immersive devices available to the general public can limit the access or experience of a web-based MM application. Without prior experience in related technologies and applications, for many novice users even navigating in a VE and setting up a HMD were particularly challenging activities.

Orientation in a possibly new environment is a problem that needs to be addressed already in the design of a VE. Solutions are discussed extensively in the field of game design [62], and can act as a practical guideline for designing and implementing the same experience for different input modalities, as well as guiding the audience through space (see Sec. V-B). To address these issues when presenting our performances (see Sec. II), we provided a brief introduction to the controls on the landing page before entering the SVE to assist inexperienced visitors. Additionally, at the beginning of each performance, participants were given a short orientation phase to familiarize themselves with the controls.

When presenting VR/AR installations (see Sec. II) that required the audience to use HMDs, we found it particularly effective to reduce the amount of input modalities available to the users (e.g., removal of one or all controllers). This helped to reduce the amount of input errors that lead to terminations or problems during the experience.

By adopting solutions such as developing resources, optimizing existing tools, and implementing targeted experience design, developers and creators can overcome browser limitations and leverage web-based multimedia strengths. In conclusion, we have identified several promising directions for future development, that however requires coordinated efforts across multiple domains. This involves a better standardization and implementation of web APIs such as WebXR and WAA in modern browsers, the development of more efficient frameworks and libraries for MM applications, improved user experience design strategies that accommodate diverse user capabilities, and finally finding more sustainable funding models for the open-source projects that form the foundation of the growing web-based MM ecosystem.

While the web-based MM represents a promising democratization of immersive musical experiences, realizing its full potential requires addressing fundamental technical and experiential barriers. Our extensive practical experience showed both the current limitations and the significant opportunities that exist for creating meaningful musical experiences in SVEs, provided that the gaps we have identified and discussed can be systematically addressed through targeted technical improvements and thoughtful user-centered design approaches. This calls for more research for the web-based MM field to blossom.

Acknowledgments

This work was realized as a part of KreativInstitut.OWL. It is a project in research cooperation of the OWL University of Applied Sciences and Arts, Detmold University of Music and University Paderborn, funded by the Ministry of Economic Affairs, Industry, Climate Action and Energy of the State of North Rhine-Westphalia, Germany. This work has been also supported by the Italian Ministry for University and Research under the PRIN program (grant n. 2022CZWWKP).

References

[1] L. Turchet, "Musical Metaverse: vision, opportunities, and challenges," *Personal and Ubiquitous Computing*, 2023.
[2] R. Waters and J. Barrus, "The rise of shared virtual environments," *IEEE Spectrum*, vol. 34, no. 3, pp. 20–25, 1997.
[3] N. Stephenson, *Snow Crash.* New York: Bantam Books, 1992.
[4] R. Elen, "Music in the metaverse," *Journal of the Audio Engineering Society*, April 2008.
[5] G. F. Martín, "Social and psychological impact of musical collective creative processes in virtual environments; The Avatar Orchestra Metaverse in Second Life." *Music/Technology*, vol. 75, pp. 75–87, 2018.
[6] K. E. Onderdijk, L. Bouckaert, E. Van Dyck, and P.-J. Maes, "Concert experiences in virtual reality environments," *Virtual Reality*, vol. 27, no. 3, pp. 2383–2396, 2023.
[7] A. Champlin, S. Knotts, J. Chicau, A. Xambó, and I. Saladino, "Community Report : LivecoderA," in *7th International Conference on Live Coding (ICLC)*, 2023, pp. 1–5.
[8] B. Loveridge, "An Overview of Immersive Virtual Reality Music Experiences in Online Platforms," *Journal of Network Music and Arts*, vol. 5, no. 1, 2023.
[9] D. Dziwis and H. von Coler, "The Entanglement – Volumetric Music Performances in a Virtual Metaverse Environment," *Journal of Network Music and Arts*, vol. 5, no. 1, pp. 1–12, 2023.
[10] J. Park, Y. Choi, and K. M. Lee, "Research trends in virtual reality music concert technology: A systematic literature review," *IEEE Transactions on Visualization and Computer Graphics*, vol. 30, no. 5, pp. 2195–2205, 2024.
[11] VRChat. (2025) Homepage. Accessed: June 17, 2025. [Online]. Available: https://hello.vrchat.com/
[12] Mozilla Foundation. (2025) Mozilla Hubs. Accessed: June 17, 2025. [Online]. Available: https://hubs.mozilla.com/
[13] Hubs Foundation. (2025) Homepage. Accessed: June 17, 2025. [Online]. Available: https://hubsfoundation.org/
[14] Somnium. (2025) Somnium Space. Accessed: June 19, 2025. [Online]. Available: https://www.somniumspace.com/
[15] PatchXR. (2025) PatchWorld. Accessed: June 17, 2025. [Online]. Available: https://patchxr.com/

[16] H. Vo and R. Boulanger, "Csound in the metaverse – from cabbage to csoundunity and beyond: Developing a working environment for soundscapes, soundcollages, and collaborative soundplay," in *Proceedings of the 7th International Csound Conference*. Vienna, Austria: ICSC 2024, September 2024, pp. 115–119.

[17] S.-M. Park and Y.-G. Kim, "A metaverse: Taxonomy, components, applications, and open challenges," *IEEE Access*, vol. 10, pp. 4209–4251, 2022.

[18] Meta. (2025) Horizon. Accessed: June 17, 2025. [Online]. Available: https://horizon.meta.com/

[19] W3C. (2025) WebXR. Accessed: June 17, 2025. [Online]. Available: https://immersiveweb.dev/

[20] W3C. (2025) Web Audio. Accessed: June 17, 2025. [Online]. Available: https://www.w3.org/TR/webaudio/

[21] A. McArthur, C. Van Tonder, L. Gaston-Bird, and A. Knight-Hill, "A survey of 3d audio through the browser: practitioner perspectives," in *2021 Immersive and 3D Audio: from Architecture to Automotive (I3DA)*. IEEE, 2021, pp. 1–10.

[22] M. Tomasetti, A. Boem, and L. Turchet, "How to Spatial Audio with the WebXR API: a comparison of the tools and techniques for creating immersive sonic experiences on the browser," in *2023 Immersive and 3D Audio: from Architecture to Automotive*. IEEE, 2023, pp. 1–9.

[23] L. Cao, "Decentralized ai: Edge intelligence and smart blockchain, metaverse, web3, and desci," *IEEE Intelligent Systems*, vol. 37, no. 3, pp. 6–19, 2022.

[24] M. Buffa, D. Girard, and A. Hofr, "Using web audio modules for immersive audio collaboration in the musical metaverse," in *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)*, 2024, pp. 1–10.

[25] C. Oh, D. Lee, and A. Smith, "Doongdoong.club: A Web-Based Metaverse Music Sequencer With Korean Onomatopoeic and Mimetic Words," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2024, pp. 622–626.

[26] 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*. IEEE, 2023, pp. 1–9.

[27] A. Boem, D. Dziwis, M. Tomasetti, S. Etezazi, and L. Turchet, """It Takes Two"-Shared and Collaborative Virtual Musical Instruments in the Musical Metaverse," in *2024 IEEE 5th International Symposium on the Internet of Sounds (IS2)*. IEEE, 2024, pp. 1–10.

[28] 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*, vol. 71, no. 11, pp. 802–812, 2023.

[29] D. Dziwis, H. von Coler, and C. Porschmann, "Live Coding in the Metaverse," in *2023 4th International Symposium on the Internet of Sounds*. IEEE, 2023, pp. 1–8.

[30] D. Dziwis, "VERSNIZ – Audiovisual Worldbuilding through Live Coding as a Performance Practice in the Metaverse," in *Proceedings of the 16th International Symposium on Computer Music Multidisciplinary Research*, Nov. 2023, p. 289–300.

[31] D. Dziwis, "PdXR - the Evolution of Pure Data into the Metaverse," in *ICMC 2023 : Proceedings of the International Computer Music Conference*. ICMC, 2023, pp. 52–58.

[32] M. Nowakowski and D. Dziwis, "Disrupting Musical Behavior: Musicking Through Anti-Patterns in 3D," in *ISEA 2025: Proceedings of the 30th International Symposium on Electronic Art*. ISEA, 2025.

[33] H. Smith, *Practice-led research, research-led practice in the creative arts*. Edinburgh University Press, 2009.

[34] A-Frame. (2025) Homepage. Accessed: June 17, 2025. [Online]. Available: https://aframe.io

[35] Networked-Aframe. (2025) Git Repository. Accessed: June 17, 2025. [Online]. Available: https://github.com/networked-aframe

[36] M. Puckette, "Pure Data : another integrated computer music environment," in *second intercollege computer music concerts*, 1997, pp. 37–41.

[37] D. Dziwis. A-Frame Kinectron. Accessed: June 17, 2025. [Online]. Available: https://kinectron.github.io/#/aframe/getting-started

[38] V. Heikkilä. (2025) IBNIZ. Accessed: June 17, 2025. [Online]. Available: http://viznut.fi/ibniz/

[39] V.-M. Heikkilä, "Discovering novel computer music techniques by exploring the space of short computer programs," pp. 1–8, 2011. [Online]. Available: http://arxiv.org/abs/1112.1368

[40] Tone.js. (2025) Homepage. Accessed: July 6, 2025. [Online]. Available: https://tonejs.github.io/

[41] A. A. Correya, D. Bogdanov, L. Joglar-Ongay, and X. Serra, "Essentia.js: A javascript library for music and audio analysis on the web," in *Proceedings of the 21st International Society for Music Information Retrieval Conference*, 2020.

[42] M. Kallionpää, A. Berndt, D. A. Mauro, and D. Dziwis, "Non/Repeat: Three Case Studies of Non-linear Live-Music (accepted for publication)," in *20th Audio Mostly Conference 2025*. AM, 2025.

[43] P. Oliveros, S. Weaver, M. Dresser, J. Pitcher, J. Braasch, and C. Chafe, "Telematic music: Six perspectives," *Leonardo Music Journal*, vol. 19, pp. 95–96, 2009.

[44] Kinectron. (2025) Git Repository. Accessed: June 17, 2025. [Online]. Available: https://github.com/kinectron/

[45] S. Serafin, C. Erkut, J. Kojs, N. C. Nilsson, and R. Nordahl, "Virtual reality musical instruments: State of the art, design principles, and future directions," *Computer Music Journal*, vol. 40, no. 3, pp. 22–40, 2016.

[46] C. Rottondi, C. Chafe, C. Allocchio, and A. Sarti, "An overview on networked music performance technologies," *IEEE Access*, vol. 4, pp. 8823–8843, 2016.

[47] A. Boem, M. Tomasetti, and L. Turchet, "Issues and Challenges of Audio Technologies for the Musical Metaverse," *Journal of the Audio Engineering Society*, vol. 73, pp. 94–114, june 2025.

[48] M. Muneer, U. Rasheed, and M. M. Hussain, "A meta-model to support compatibility testing of cross-browser web application," in *2025 6th International Conference on Advancements in Computational Sciences (ICACS)*, 2025, pp. 1–8.

[49] M. Kungla. A-Frame Resonance Audio. Accessed: June 17, 2025. [Online]. Available: https://github.com/mkungla/aframe-resonance-audio-component/

[50] P. Brinkmann, C. Mccormick, P. Kirn, M. Roth, and R. Lawler, "Embedding Pure Data with libpd," *Proceeding of the Fourth International Pure Data Convention*, pp. 291–301, 2011.

[51] P. Brinkmann, D. Wilcox, R. Eakin, and R. Alexander, "libpd : Past , Present , and Future of Embedding Pure Data," in *PdCon2016∼*, 2016, pp. 1 – 6.

[52] Supermedium, "Aframe-Super-Keyboard," accessed: June 17, 2025. [Online]. Available: https://github.com/supermedium/aframe-super-keyboard

[53] Glitch. (2025) Homepage. Accessed: June 17, 2025. [Online]. Available: https://glitch.com/

[54] A. Boem, M. Tomasetti, and L. Turchet, "Harmonizing the musical metaverse: unveiling needs, tools, and challenges from experts' point of view," pp. 206–214, September 2024.

[55] M. Z. Khan and M. M. A. Hashem, "A comparison between html5 and opengl in rendering fractal," in *International Conference on Electrical, Computer and Communication Engineering*, 2019, pp. 1–6.

[56] P. Howarth and P. Costello, "The occurrence of virtual simulation sickness symptoms when an hmd was used as a personal viewing system," *Displays*, vol. 18, no. 2, pp. 107–116, 1997.

[57] J. Chen, Y. J. Lee, X. Lehto, and Y. Zhu, "Exploring experience design features in the metaverse concerts: a mixed-method approach," *Information Technology & Tourism*, vol. 27, no. 3, pp. 865–886, 2025.

[58] G. Karnchanapayap, "Activities-based virtual reality experience for better audience engagement," *Computers in Human Behavior*, vol. 146, p. 107796, 2023.

[59] K. Richards, N. Mahalanobis, K. Kim, R. Schubert, M. Lee, S. Daher, N. Norouzi, J. Hochreiter, G. Bruder, and G. Welch, "Analysis of peripheral vision and vibrotactile feedback during proximal search tasks with dynamic virtual entities in augmented reality," in *Symposium on Spatial User Interaction*, ser. SUI '19. New York, NY, USA: Association for Computing Machinery, 2019.

[60] A. Haas, A. Rossberg, D. L. Schuff, B. L. Titzer, M. Holman, D. Gohman, L. Wagner, A. Zakai, and J. Bastien, "Bringing the web up to speed with webassembly," *SIGPLAN Not.*, vol. 52, no. 6, p. 185–200, Jun. 2017.

[61] S. Chickerur, S. Balannavar, P. Hongekar, A. Prerna, and S. Jituri, "Webgl vs. webgpu: A performance analysis for web 3.0," *Procedia Computer Science*, vol. 233, pp. 919–928, 2024, 5th International Conference on Innovative Data Communication Technologies and Application (ICIDCA 2024).

[62] M. Dickey, "Engaging by design: How engagement strategies in popular computer and video games can inform instructional design," *Educational Technology Research and Development*, vol. 53, no. 2, pp. 67–83, 2005.