Building hybrid performances with DMIs, Hubs and Faust

Michał Seta
RedSpills, Kaon'CPT
mseta@pm.me

Dirk Johan Stromberg
LASALLE College of the Arts
School of Contemporary Music
1 McNally Street
Republic of Singapore
ldirk.stromberg@lasalle.edu.sg

D. Andrew Stewart
University of Lethbridge
4401 University Drive
Lethbridge, AB, Canada
andrew.stewart@uleth.ca

ABSTRACT

In this article, we describe the challenges of an artistic residency that included: a distributed improvisation in VR, performances using Digital Musical Instruments (DMIs), and Open Source software as much as possible. For this residency, we were constrained to using Mozilla's Hubs as the Metaverse platform. We describe the shortcomings of the platform as a performance space in light of our experience, musical cultures, and the social aspects of a musical performance. We also address select technical issues pertaining to the context of a hybrid musical performance (simultaneously in Virtual Reality (VR) and in-real-life (IRL)) using this particular technology stack. Furthermore, we describe the challenges and surprises that occurred with Faust (Function Audio Stream), which was our choice of synthesis engine for the project. We conclude this paper by identifying some possible avenues for future research, exploration, and performances of a similar nature. We wish to clarify that although we will be talking a lot about Hubs, which was the Virtual Reality (VR) platform used for the residency, we were not endorsed by Mozilla.

Author Keywords

Artistic Telepresence; Telematic music performance; Metaverse; Hybrid performance; Music-related human-computer interaction;

CCS Concepts

•**Applied computing** → **Sound and music computing;** Performing arts;

1. INTRODUCTION

In recent years, particularly since the COVID-19 pandemic outbreak, the community has seen a surge of interest in various types of online music practices and collaborations [19] as in-real-life (IRL) interactions were greatly limited. We are referring to musical practices specifically, and we acknowledge that this is hardly a new phenomenon in the musical domain. It began with the advent of computer networks in the early 70s [4] and has become a regular practice for many, evidenced by the rise of festivals and organizations such as Network Music Festival [18] and NowNet Arts [3]. The improvement in consumer internet connection bandwidth allows a more wide-spread use of internet technologies for telepresence, including music [22]. VR and computer-game technologies encourage the development of virtual instruments and new styles of musical experience [16]. In spite of these developments, audio takes a back-seat in most modern commercial VR systems [7].

This paper presents a subjective look at a specific case of creating and performing a musical work utilising exclusively DMIs, designed to be performed as a hybrid experience, simultaneously for IRL and VR audiences, using a popular, widely available VR environment. We present the challenges we faced and solutions we have settled upon.

2. METAVERSE AND MUSIC

Music is a social activity and is very often presented with many extra-musical elements. In its many forms and incarnations, the sound of music(s) accompanies human activities related to entertainment, communication and ritual [17]. Inevitably, it has made its way into virtual environments. The 21st century has seen an increase in the interest of hybrid musical experiences, where the physical and virtual worlds fuse [10]. In some cases, the musicians, or performers, combine both their VR and IRL presence, the attending audience, as well, may exist physically and virtually. Moreover, the physical and virtual worlds may converge by having both performers and the audience as concurrent VR and IRL participants [25].

3. RELATED WORK

One of the most spectacular recent incarnations of the first approach, was *A Bowie Celebration* [1] world tour, which was interrupted by the COVID-19 pandemic and became a hybrid concert as a live stream[1], featuring co-present and distant performers. Adding audiences to the mix, both virtual and physically present, is an additional challenge. Jean-Michel Jarre has attempted this using VR Chat platform in *Alone Together* VR concert [2]. Pop culture is not, however, the only one partaking in the VR excursion. On a more experimental exploration side of the VR space, we will highlight a few examples using Mozilla Hubs, since they more closely relate to the present paper, although there are many more popular VR platforms [9]. Live-coding community has a more permanent space Hubs, called *Algorave*

[1]`https://www.youtube.com/watch?v=7QPW2MZZ8zk`
[2]`https://www.youtube.com/watch?v=omyFUTO4Phc`

NIME'23, 31 May–2 June, 2023, Mexico City, Mexico.

VR [3]. Artists Claudia Hart and Matthew Gantt have collaborated [4], and *The Fountain* by Matthew Gantt [5] explored the use of Hubs for live performances as well. These kinds of hybrid projects are paving the way for a new musical culture and a need for innovation that are not limited to advancements in music technology, but also concern how we relate to musical performance via our bodies (i.e., sense of physical presence), our ears (i.e., auditory perception), and complex cultural and social processes [5].

4. PROJECT GOALS VS. CONSTRAINTS

As a trio of digital musical instrumentalists and improvisers, we were given the opportunity to do a research residency with Satellite[6], a department at the Society for Arts and Technology (SAT)[7] in Montréal, Canada. The residency took place in the first half of 2022. The current focus of Satellite is enhancing the Mozilla Hubs[8] platform to facilitate VR experiences/artworks in the form of XR installations and/or live performances. Our focus was on building a network multimedia performance (NMP) utilizing only DMIs (Karlax [9], T-Stick [10] and "a new instrument" by Dirk Stromberg [11]).

The artistic work was framed by the following considerations:

- the geographical location of performers (three locations spanning 14 time zones)
- social, logistical and musical considerations based on acquired musical and performance culture
- Mozilla Hubs' technical limitations
- a strong relationship between gesture, sound, and visual representation
- ability to perform the work in different VR vs. IRL configurations

The issues outlined above, concern the integration of live music technologies and production workflows into a hybrid (i.e., VR and IRL) performance. They require thoughtful and dedicated attention from many perspectives: technologist, musician, performer, etc.. As a DMI trio, we had had prior experience working with telematics, particularly in Hubs[12, 13].

4.1 Mozilla Hubs

Mozilla Hubs was the principal constraint for the performance component of our residency. Of course, there are many present-day platforms that can cater — for better or for worse — to a community of artists wishing to make and present musical performances online; however, Hubs was central to our residency because it served as the final delivery platform, the *venue*. Its integration with various music, and particularly, live music technologies and production workflows was paramount. Hubs is, first and foremost, a social platform and is based on the idea of people meeting in the same space. A Hubs experience parallels the IRL encounter; for instance, a Hubs scene provides a first-person perspective of a virtual room in which others — the interlocutors — are represented by avatars. One moves around the room — one interfaces — by using the computer keyboard, while computer mouse movements affect the orientation of one's avatar. For those who experienced first-person shooter (FPS) video games, the Hubs experience is a very familiar way of moving and looking around and navigating the scene. One of the most important and attractive features of Mozilla's Hubs, is that it runs in the everyday internet browser and can be used on various platforms, including mobile devices and head-mounted displays (HMDs). The Hubs environment is also programmed to simplify scenes in order to accommodate users with less powerful interfaces/devices. Additional factors that promote the integration of live music technologies and production workflows into Hubs are considered below.

4.2 Social

Hubs creates an environment that allows participants to mimic some fundamental social interactions. For example, participants can gather in groups or break out and interact in almost natural ways. Once inside a Hubs room, one can see another person's avatar move, pivot around their vertical axis, and animate when the avatar's user is speaking. Plus, avatar "head" movements are also visible – nodding up and down and, thus, one can pinpoint the general direction in which another user is looking inside the room. These features can contribute to a better perceptive quality of musical performance [14]. However, there are still numerous problems with Hubs, that prevent the interactions with other people to be completely smooth and without problems [11]. For a community of artists interested in music and sound integration into Hubs, each avatar features audio spatialisation; a person's avatar will project the IRL voice of the avatar's user (e.g, microphone signal; sound electronically rewired to the user's internet browser sound input). Moreover, the loudness of each avatar is directly proportional to distance; the closer you get, the louder the sound emanating from the avatar. Sound localisation and spatialisation parameters are also available for most sound sources placed inside a Hubs scene. Controlling these parameters helps to contribute somewhat to the social aspect of the platform because the sound can be made to behave, following the principles of real-world physics. Hubs offers an audience member some response, or reaction, buttons, allowing the audience to show primitive icons (e.g., hand clap) in response to the performance; however, this requires the performer to include the additional step of watching for these ephemeral reactions within the Hubs scene, while also playing the music.

4.3 Technical

While it is easy to transmit live audio signal inside Hubs, signal quality will often be disappointing even for the most tolerant lo-fi audio lover. Mozilla has optimized Hubs for the human voice (i.e., speech). So, preserving the audio quality of, say, the music inside a Hubs scene can be hopeless. One solution for transmitting reasonably good quality audio is uploading audio assets using Mozilla's Spoke interface/software scene editor, which can then be used to

[3] `https://hubs.mozilla.com/LYmhUf3/algorave-vr/`
[4] `https://www.youtube.com/watch?v=WlTfRN-9I18`
[5] `https://www.youtube.com/watch?v=ueUw-uNYFpo`
[6] `https://sat.qc.ca/fr/satellite-espace-virtuel`, visited on 2023-01-22
[7] `https://sat.qc.ca`
[8] `https://hubs.mozilla.com/`, visited on 2023-01-22
[9] `http://www.dafact.com/fonctionnalites.php?id_product=1`
[10] `https://www.idmil.org/project/the-t-stick/`
[11] `https://dirkstromberg.org/2021/10/08/a-new-instrument/`
[12] http://www.youtube.com/watch?v=udIiQn1idY8
[13] http://vimeo.com/529719034

populate a Hubs scene with *fixed* media sources: videos and audio files (or URLs pointing to the fixed media). A participant in a Hubs scene can also "paste" the URL of a fixed media source into a scene using Hubs' standard scene controls; however, the source is deleted when the participant quits, or leaves, the scene. Working with Spoke is the preferred method of fixed music integration for artists because the scene editor offers parameter controls over aspects such as: sound attenuation in relation to distance, sound attenuation curves and attenuation rate factors, and different sound diffusion patterns. A fixed media source may also be configured to have a consistent loudness throughout the entire Hubs scene, exactly like a 2D audio asset in a video game.

The integration of sound and music that is not fixed (i.e., live music improvisation) is more challenging with respect to preserving signal quality. Using an embedded video host (e.g., Twitch TV) can be leveraged in order to stream audio inside a Hubs scene. Similar to a fixed media asset, a URL to the video host can be instantiated via Spoke or through a participant pasting the link directly into a Hubs scene. While the Hubs environment is programmed to retain a stream's audio and video quality, the result is excessive latency; a hybrid performance becomes untenable because the VR presence of the performer is substantially late – desynchronised, when compared to the IRL presence of the performer.

4.4 Logistical

Performing live online is not easy. For instance, the physical distance between performers, and between performers and their audiences, particularly when everyone is in different, remote, geographical locations, can have an alienating effect; a performer may have no idea whether a member of the audience can hear the music.

4.5 Spatial

During a performance, certain practices include in situ actions that are only visible within the performance space. Actions include communicating with body language [8] and eye contact [2]. During an online, distributed, performance, these in situ actions are nearly impossible to integrate with live online music performances. In Hubs, we leveraged the presence of avatars in response; that is to say, performers' avatars were equipped with surface textures derived from each performer's webcam signal into Hubs. Additional work is still required to enhance a performer's spatial presence.

4.6 Musical

One of the most prominent difficulties for musicians, while playing over the internet, is a lack of synchronisation [21] and multiple different perceptions of musical timing (e.g., meter, beat, tempo). Additionally, an ensemble of musicians may find it difficult to hear each other; they may not be able to identify how, or by whom, a musical sound is made [13]. During our residency, we recognised these challenges. While we explored approaches [6] to transmitting sound with reasonably low levels of latency and, thus, improving synchronisation, we also considered that accepting the limitations of present-day telematic technologies is part of the creative process. In other words, the limitations and constraints were matter, or materials, for our musical thinking.

4.7 Performance context

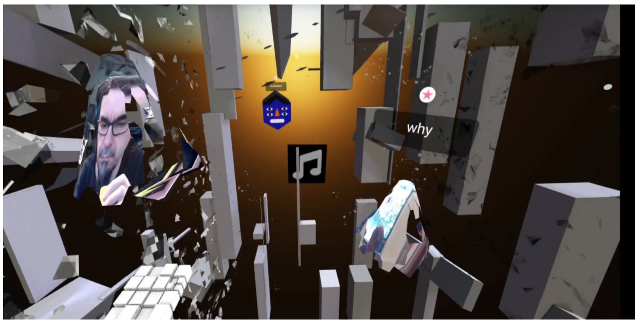

Figure 1: Screenshot of the Hubs scene showing two of the players' avatars

As the residency developed, the final performance format, which had been scheduled as a remote presentation due to COVID-19, was shifted to a hybrid model; as the residency progressed, COVID-19 confinement measures were slowly relaxed worldwide. Consequently, our interest in expanding the residency to include a hybrid presentation was increasingly stronger. Armed with prior experiences performing in the (vanilla) Hubs VR space, we developed an approach to a hybrid delivery that expanded our earlier Hubs performances, which relied on embedded video hosting for conveying our visual and sonic presence (e.g., Twitch TV). In particular, we explored a low, or less, audio latency-prone solution with the Satellite development team at the SAT.

5. IMPLEMENTATION

We concluded the residency with a system that corresponded to our initial projections. Our take on Hubs' limitations in handling interactive audio was to synthesize the audio directly in the browser. The synthesis was handled by Faust (Functional AUdio STream) [20] and we used its ability to export to WebAssembly [23, 15] (WASM), which was then integrated into the webpage served to participants' web browser. The control signals from our DMIs were sent to the page using WebSockets [24] and affected various synthesis parameters and interacted with some 3D objects in the "room". Some aspects of musical gestures were mapped to control some of the visual aspects in the room. For instance, *note-on/note-off* type of messages were triggering UV texture animation on objects associated with each performer, and *amplitude* parameter was mapped to the animation speed. The performers had the ability to map some arbitrary parameters to control the rotation of the "room" in which the performance was taking place.

Our final rendition relied heavily on the custom version of the Hubs instance hosted by Satellite which uses a technique called "code injection", based on William Freeman's (clinfizgig on GitHub[14]) approach to customizing Mozilla Hubs components[15]. Their own injection server[16] provided an opportunity to deploy custom JavaScript code to add functionality to the running Hubs instance.

In order to promote the hybrid performance, the audio output from the Hubs scene was routed from one of the performer's computers to the PA in the performance venue. In this way, the IRL audience could also experience the sounds being generated in Hubs. A short video documentation of our work, *Oscuterium* can be seen at https://youtu.be/udIiQn1idY8

[14]https://github.com/colinfizgig
[15]https://github.com/colinfizgig/Custom-Hubs-Components
[16]https://gitlab.com/sat-mtl/satellite/hubs-injection-server

6. IMPLEMENTATION DETAILS

6.1 Faust

Seduced by Faust's ability to compile DSP code to different target architectures [12], we decided to use Faust WebAssembly builds to create the virtual instruments for our project. Thus, all sounds associated with the DMIs were synthesized on the client computer (i.e. directly in the web browser of the person accessing the Hubs room were the performance was held).

The feature of being able to convert code allowed us to modify and test the DSP code, use native exports (i.e. VST plugin, Max or Pd external, jack client, etc.), develop mapping strategies to control synthesis parameters, and practise our instruments off-line – independently of the web browser and its stack.

The Faust approach turned out to be not without flaws. We have discovered that the WASM export creates some global variables in the accompanying JavaScript code and we had to leverage this with some build scripts that consumed the Faust files for each instrument and generated some additional JavaScript code to avoid variable names collisions. Those scripts can be found in our GitLab repository [17]. Unfortunately, this code is specific to the Satellite flavour of Hubs, which is not yet available publicly, so our code may not be useful as is, but could certainly help in overcoming similar issues if someone chooses to follow similar path.

The particular difficulty addressed by our code, is the production of `.wasm` code, accompanied by JavaScript integration code. The latter, contains a global variable `const dspName` which gets a synth name, as string, assigned. The said string is derived from the Faust synth definition filename. Trying to use more than one instrument leads to JavaScript errors and unusable synths. We have addressed this issue by writing a script that renames `const dspName` variable to a more meaningful and unique name, i.e. it replaces `const dsp` prefix by wrapping it into a class that will be understood by the injection server, as seen in this excerpt from the `build.sh`:

```
for i in $(grep -l dspName wasm/*.js)
do
  filename=$(basename $i .js)
  n=$(awk '/const dspName/{print NR}' $i)
  sed -i $( expr $n - 1 )q $i
  echo "SAT.Utils.faust.classes['${filename}'] = \
    ${filename};" >> $i
done
```

We also needed to write some custom JavaScript to handle instantiating of the `.wasm` code specifically in the injection server. The Python code handling that can be found in the `generate_plugin_js.py` of the same repository.

We also discovered, that Faust code was exported to WASM lost its polyphonic capacity. residency and for example, we tried to balance out the constraint of having a single voice with distinctive sound synthesis techniques; although each Faust instrument produced one voice, we tried to enable a rich playing experience in both sound colour and how the instrument was controlled (using a DMI).

6.2 Hubs limitations

As mentioned previously, working with customised code and the Hubs stack took concerted skill and effort — the stack is quite complex and inflexible with respect to live audio input and audio quality within a Hubs scene. Consequently, we were constrained by present-day web browser technology. For example, on the one hand, synthesizing all of our sounds on the client computer was very appealing because it eliminated the need to route compressed audio signals and having to deal with compression management issues inside Hubs. On the other hand, this approach required us to accept limited control over the efficiency of the code running on the user's computer. Some devices may not be powerful enough to compute the audio signals required. We were mindful of this.

In addition to these concerns, we were unable to ascertain that the synthesis actually executed on the client's browser. On some rare occasions, no sound was produced and we did not find a way of detecting such issue nor addressing it, save for reloading the webpage.

[17] `https://gitlab.com/redspills/oscuterium-room-injection`

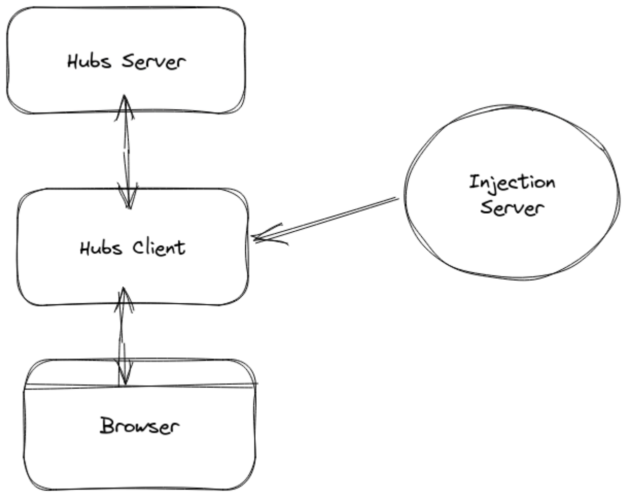

Figure 2: Hubs stack with injection server

6.3 Injection server

We wish to underline that using any customised code in Mozilla's Hubs stack is not trivial. Whether incorporating custom code into the server or the client, one needs to deploy one's own server and modify the Hubs code – serious effort is required to create a functional system. SAT's Satellite developers implemented an injection server[18] in order to facilitate injecting client servers without having to modify the upstream Hubs code. The overall functionality was based on the interaction between server and client, where the client functionality was intercepted and augmented by customised code before being presented to a web browsing client – browser of performer, Hubs scene participant, audience member, etc. (see Figure 2). Of course, it is unlikely that the code developed for this particular injection solution will work on another Hubs instance.

6.4 Control

All DMIs in our project had the capacity to use the Open Sound Control (OSC) communications protocol. We developed a utility called *babyhands*[19], inspired by *allhands*[20], a utility for sending and receiving OSC data over WebSock-

[18] `https://gitlab.com/sat-mtl/satellite/hubs-injection-server`
[19] `https://gitlab.com/djiamnot/babyhands`
[20] `https://github.com/michaelpalumbo/allhands`

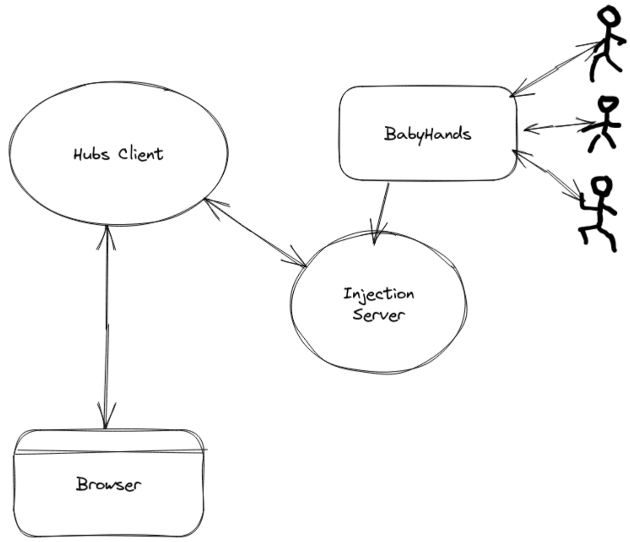

Figure 3: OSC routing with Babyhands

ets. *Babyhands* enables forwarding OSC messages to all performers, as well as the injection server via WebSockets. OSC data received by the injection server was then used to control all Faust instruments and the properties of virtual objects in the 3D Hubs scene (see Figure 3). While the websockets approach proved fruitful, we expected random failures of, or surprises from, some web clients, especially due to latency and unreliable network connections.

7. EVALUATION OF OUR EXPERIENCE

This evaluation is purely subjective and draws from our experience as artists dealing with a variety of technological hurdles during one residency.

Overall, our approach to a distributed improvisation in VR, using DMIs, proved fruitful and satisfying. We wish to highlight the following observations:

7.1 Latency

As mentioned in 4.6, latency is one of the prominent factors that participants in NMP must face. While we were able to drastically reduce the bandwidth needed to produce sound, by sending only control messages, the latency was still an issue. The (geographical) distances among Hubs participants and each participant's local internet infrastructure still played an important role. For instance, the Amazon Web Services (AWS) server hosting the Hubs scene, developed during our residency, was located in North America. During the development of the project and while practising for the final performance, two performers remotely connected to the Hubs scene from widely separated locations in Canada and one performer connected from the Republic of Singapore. For the final hybrid performance, the member from Singapore was in Montreal and performed for the IRL audience. Consequently, the impact of distance and infrastructure on a distributed performance was evident; for instance, latency was practically undetectable for the IRL performers.

7.2 Limitations of the system

Mozilla Hubs is a great solution for hosting low-key NMPs, especially, when access to the performance space is desired on widest range of devices, without installing any additional software. It simply runs in a web browser on mobile devices, laptops, and desktop computers. This availability comes at a price. First, the number of participants in one room is limited to 25 (including the performers), on most Hubs servers. Furthermore, the visual aspect of the room will be impacted by the limitations imposed by the platform in terms of total number of 3D objects' faces. The creation of visually interesting worlds requires a considerable effort in optimization of 3D objects and textures.

7.3 Audio quality

From the sonic perspective, in spite of the issues identified in 6.1, using Faust as a synthesis language for experiences hosted in Web VR has a potential for yielding interesting and rich results. For fixed media, or non-time-critical content that is streamed from other servers via WebRTC protocols, the quality will not be degraded.

8. CONCLUSIONS AND FUTURE DIRECTIONS

In this paper we presented our approach to building a Networked Multimedia Performance using DMIs within Mozilla Hubs VR environment. We've identified the goals and constraints that framed our project, which consisted of the musical, technical, social, and logistical considerations, all in relation to the chosen VR medium. We've explained our implementation of our project, mostly from technical standpoint. We described our innovation to dealing with sound in this context, which consists of using Faust to render audio directly in the web browser. We identified the technical challenges that we have encountered, which consisted mostly of the complexity of Hubs software stack and some surprising outcomes from using the Faust language for audio synthesis.

Based on our project development and final output – a live hybrid VR and IRL performance, we made significant headway in three areas:

- Producing a hybrid performance during which the physical and virtual worlds fuse by having both performers and audience as concurrent VR and IRL participants.
- Integrating live music technologies and production workflows in Mozilla's Hubs.
- Combining WebAudio and Faust allowing us to synthesise sound – and perform – directly within a Hubs scene.

Future development will focus on exploring different approaches to utilizing WebAudio APIs, either directly, or via available wrappers, and comparing performance and user experience. Furthermore, we will focus on implementing monitoring tools to enable us to more easily pinpoint problematic code.

9. ACKNOWLEDGEMENTS

We dedicate this paper to the memory of Eric Poirier who was leading the injection server and Hubs customization developments with Satellite department. We thank the entire Satellite team for their support, enthusiasm and open mind throughout our residency.

10. ETHICAL STANDARDS

This research-creation project was held during an artistic residency at the Société des arts technologiques [SAT] focused on integration of diverse technologies to support Mozilla Hubs specifically. No compensation was received by the authors or the SAT in exchange for choosing the software stack, to the best of our knowledge. There were no observed conflicts of interest. All developers, researchers and artists participated consensually in the activities described in this document.

11. REFERENCES

[1] *A Bowie Celebration - The David Bowie Alumni Tour*. https://abowiecelebration.com/. (Visited on 04/01/2023).

[2] Alessandro Antonietti, Daniela Cocomazzi, and Paola Iannello. "Looking at the Audience Improves Music Appreciation." In: *Journal of Nonverbal Behavior* 33.2 (June 2009), pp. 89–106. ISSN: 0191-5886, 1573-3653. DOI: 10.1007/s10919-008-0062-x. (Visited on 04/12/2021).

[3] NowNet Arts. *NowNet Arts*. https://nownetarts.org. 2017. (Visited on 03/20/2023).

[4] Álvaro Barbosa. "Displaced Soundscapes: A Survey of Network Systems for Music and Sonic Art Creation." In: *Leonardo Music Journal* 13 (Dec. 2003), pp. 53–59. ISSN: 0961-1215, 1531-4812. DOI: 10.1162/096112104322750791. (Visited on 01/21/2023).

[5] Nicolas Bouillot and Michał Seta. "The Kaon'CPT Collective: Building a Musical Culture of Not-in-Real-Life Performance through Conducted Live Comprovisation." In: *Organised Sound* 26.3 (Dec. 2021), pp. 368–377. ISSN: 1355-7718, 1469-8153. DOI: 10.1017/S1355771821000455. (Visited on 12/22/2021).

[6] Juan-Pablo Caceres and Alain B Renaud. "Playing the Network: The Use of Time Delays as Musical Devices." In: *Proceedings of the International Computer Music Conference* (2008).

[7] Anıl Çamcı and Rob Hamilton. "Audio-First VR: New Perspectives on Musical Experiences in Virtual Environments." In: *Journal of New Music Research* 49.1 (Jan. 2020), pp. 1–7. ISSN: 0929-8215, 1744-5027. DOI: 10.1080/09298215.2019.1707234. (Visited on 04/03/2023).

[8] Andrew Chang et al. "Body Sway Reflects Joint Emotional Expression in Music Ensemble Performance." In: *Scientific Reports* 9.1 (Dec. 2019), p. 205. ISSN: 2045-2322. DOI: 10.1038/s41598-018-36358-4. (Visited on 04/12/2021).

[9] Ruizhi Cheng et al. "Are We Ready for Metaverse?: A Measurement Study of Social Virtual Reality Platforms." In: *Proceedings of the 22nd ACM Internet Measurement Conference*. Nice France: ACM, Oct. 2022, pp. 504–518. ISBN: 978-1-4503-9259-4. DOI: 10.1145/3517745.3561417. (Visited on 04/02/2023).

[10] Adriana de Souza e Silva. "From Simulations to Hybrid Space: How Nomadic Technologies Change the Real." In: *Technoetic Arts* 1.3 (Dec. 2003), pp. 209–221. ISSN: 1477965X, 17589533. DOI: 10.1386/tear.1.3.209/1. (Visited on 04/08/2023).

[11] Thommy Eriksson. "Failure and Success in Using Mozilla Hubs for Online Teaching in a Movie Production Course." In: *2021 7th International Conference of the Immersive Learning Research Network (iLRN)*. Eureka, CA, USA: IEEE, May 2021, pp. 1–8. ISBN: 978-1-73489-952-8. DOI: 10.23919/iLRN52045.2021.9459321. (Visited on 04/06/2023).

[12] Dominique Fober, Yann Orlarey, and Stéphane Letz. "FAUST Architectures Design and OSC Support." In: (2011).

[13] Lauren Hayes and Christos Michalakos. "Imposing a Networked Vibrotactile Communication System for Improvisational Suggestion." In: *Organised Sound* 17.1 (Feb. 2012), pp. 36–44. ISSN: 1355-7718, 1469-8153. DOI: 10.1017/S1355771811000495. (Visited on 01/26/2023).

[14] Jay Juchniewicz. "The Influence of Physical Movement on the Perception of Musical Performance." In: *Psychology of Music* 36.4 (Oct. 2008), pp. 417–427. ISSN: 0305-7356, 1741-3087. DOI: 10.1177/0305735607086046. (Visited on 04/12/2021).

[15] Stéphane Letz, Yann Orlarey, and Dominique Fober. "FAUST Domain Specific Audio DSP Language Compiled to WebAssembly." In: *Companion of the The Web Conference 2018 on The Web Conference 2018 - WWW '18*. Lyon, France: ACM Press, 2018, pp. 701–709. ISBN: 978-1-4503-5640-4. DOI: 10.1145/3184558.3185970. (Visited on 04/01/2023).

[16] Ben Loveridge. "Networked Music Performance in Virtual Reality: Current Perspectives." In: 2 (2020).

[17] Jeremy Montagu. "How Music and Instruments Began: A Brief Overview of the Origin and Entire Development of Music, from Its Earliest Stages." In: *Frontiers in Sociology* 2 (June 2017), p. 8. ISSN: 2297-7775. DOI: 10.3389/fsoc.2017.00008. (Visited on 04/02/2023).

[18] *NMFLive*. https://live.networkmusicfestival.org/. 2020. (Visited on 03/20/2023).

[19] Kelsey E. Onderdijk, Freya Acar, and Edith Van Dyck. "Impact of Lockdown Measures on Joint Music Making: Playing Online and Physically Together." In: *Frontiers in Psychology* 12 (May 2021), p. 642713. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2021.642713. (Visited on 04/02/2023).

[20] Yann Orlarey, Dominique Fober, and Stephane Letz. "FAUST : An Efficient Functional Approach to DSP Programming." In: *New Computational Paradigms for Computer Music* (2009).

[21] Nathan Schuett. "The Effects of Latency on Ensemble Performance." In: *Bachelor Thesis, CCRMA Department of Music, Stanford University* (2002).

[22] Mel Slater and Maria V. Sanchez-Vives. "Enhancing Our Lives with Immersive Virtual Reality." In: *Frontiers in Robotics and AI* 3 (Dec. 2016). ISSN: 2296-9144. DOI: 10.3389/frobt.2016.00074. (Visited on 04/02/2023).

[23] Wikipedia contributors. *WebAssembly — Wikipedia, the Free Encyclopedia*. 2023.

[24] Wikipedia contributors. *WebSocket — Wikipedia, the Free Encyclopedia*. 2023.

[25] Clemens Wöllner, ed. *Body, Sound and Space in Music and Beyond: Multimodal Explorations.* First. Abingdon, Oxon; New York, NY: Routledge, 2017. | Series: SEMPRE studies in the psychology of music: Routledge, Apr. 2017. ISBN: 978-1-315-56962-8. DOI: 10.4324/9781315569628. (Visited on 04/03/2023).