VERSNIZ - Audiovisual Worldbuilding through Live Coding as a Performance Practice in the Metaverse

Damian Dziwis[1 2] ⋆

[1] TH Köln - University of Applied Sciences
[2] Technische Universität Berlin
damian.dziwis@th-koeln.de

Abstract. Even before the circumstances the global pandemic forced, a diverse ecosystem of technologies and artistic practices for performances in digital and virtual media was raising. Thus, not only is there a sustained interest in transferring existing performance practices into said media, but it also enables the emergence of new practices and art forms. In particular, immersive, networked, virtual multiuser environments (summarized under the term "metaverse") offer many possibilities for creating new art experiences that need to be explored. In this paper, we present VERSNIZ, a system for audiovisual worldbuilding, the spatial shaping of virtual environments, as a collaborative real-time performance or installation practice. It combines gamification concepts, known from popular sandbox video games, with the performance practice of live coding based on the esoteric programming language IBNIZ. We describe the technical implementation of the system, as well as the resulting artistic concepts and possibilities.

Keywords: Live Coding, Metaverse, Networked Music Performance, Virtual Installations, IBNIZ

1 Introduction

The term "metaverse" has become a real hype, and today all kinds of virtual and augmented reality (VR, AR) applications are often promoted as "metaverse". Used as a marketing term, the question of how these called metaverse applications differ from other VR/AR applications often remains unclear. Derived from the dystopian science fiction novel "Snow Crash" [1], the definition of the metaverse as "an interconnected web of social, networked immersive environments in persistent multiuser platforms" ([2], p. 1) seems to be gaining acceptance. According to this definition, the term metaverse environments is suitable for classifying online, multiuser, interactive, and interconnected virtual worlds, in contrast to other virtual applications. As virtual,

⋆ I thank Christoph Pörschmann (TH Köln), Stefan Weinzierl and Henrik von Coler (TU Berlin) for supervision and support in carrying out the research and writing this paper.

freely experiential environments, metaverse worlds offer a high degree of immersion and many creative possibilities. The manifold ways of multimodal interaction via text, speech, and movement allow for diverse social exchange and self-expression [3]. Thus, they have many properties that make them suitable for art experiences [4], also in the context of music practice [5]. A particular challenge in the context of metaverse environments for art expression is real-time performances with multimedia content of music and visuals. Although there is a long history of realizations and concepts for live performances in metaverse environments [6], in recent years, they have come into focus for a broader audience due to the limitations imposed by the global pandemic. But as all related areas of the metaverse continue to grow and advance technologically, we can expect to see continued interest in these topics. Metaverse environments can provide an immersive environment for telematic/networked music performances (NMPs) [7] and are an environmentally friendly and barrier-free alternative for bringing together audiences and artists from around the world.

Here, not only the issue of suitable streaming technologies is in the foreground [8], but also the resulting artistic possibilities. It is not only about how to transfer existing performance practices into such virtual environments but also about which new practices can emerge from these environments. Virtual environments, such as in VR, already allow for expanded possibilities in terms of virtual instruments, composition, and performance practice [9,10,11]. They also offer the possibility of creating entire worlds under the concept of worldbuilding/worldmaking [12,13]. The technological overlap with video games also allows for the incorporation of gamification elements for composing or performing in such environments [14,15].

In comparison to the above-mentioned virtual instruments or systems for performances in enclosed virtual environments, we present a system for real-time performances and installations incorporating the audience in online, multiuser, metaverse environments. Because they are shared virtual environments that are also accessible to the public via the Internet, they enable audience participation in virtual performance and composition processes and thus new art practices. How can the role of the performer, the stage, and the audience be redefined in this process? To explore the emerging possibilities of such environments, we combine the gamification concept of worldbuilding with the practice of live coding. As a result, we propose a performance and composition system that is only feasible in metaverse environments.

In the following, we describe the development of the metaverse environment ”VERSNIZ” for audiovisual live coding using the IBNIZ programming language [16]. The implementation integrates the above-mentioned concepts, to dissolve conventional ideas of performer, audience, and stage and enable a new performance practice in metaverse environments.

2 Background

A long and wide-ranging history of virtual environments as a medium can be found in video games. The perspective of the recipient is significant here. Usually, the user does not take the passive role of a spectator, but is an active protagonist in narrative scenarios or collaborative games. In concepts like sandbox games, such as the popular

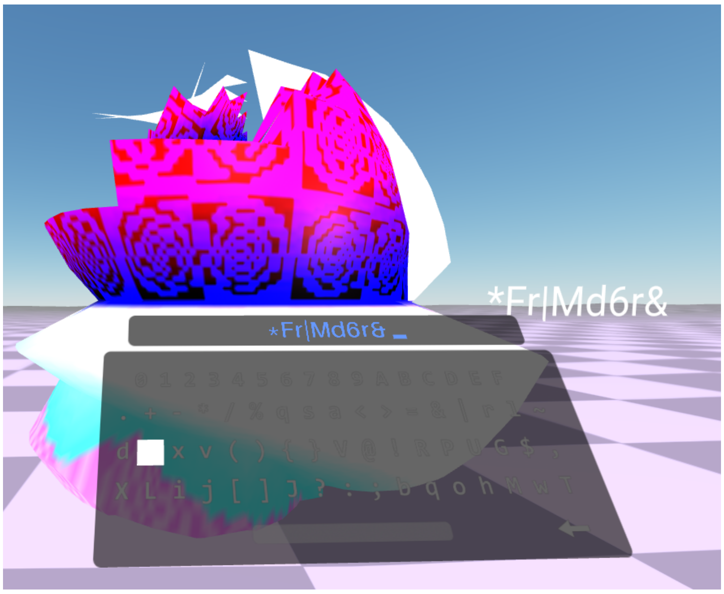

Fig. 1. A screen capture from the VERSNIZ environment, showing a live coding object placed in the default virtual world. It is rendering music and an animated visual from an example algorithm.

Minecraft [17], the player also takes on the role of the creator, creatively shaping the environment and the game experience. Here, players build virtual worlds by placing static or interactive elements as building blocks that shape the entire environment, referred to here as the concept of worldbuilding.
Thus, especially in the medium of virtual environments, the long practice of video games has led to a familiar blending of the recipient and the performing actor. The surrounding world is thereby both, the material as well as the stage for the creation of these experiences. When it comes to live performances of audio/visual content, there are a growing number of musical live performances in multiplayer games like Fortnite or metaverse environments such as VRChat or Mozilla Hubs [18]. In its most basic application, these are concerts with 2D audio and video live streams on such platforms [19,20]. The experience, with the audience viewing a large virtual screen, is more akin to a public screening than an actual live performance. More complex approaches allow artists to perform as virtual avatars, partly based on motion capture [21,22]. High-quality, pre-recorded live performances with 3D sound and video are another form of virtual live performances [23].
This century has seen a rise in the concept of the composer-programmer, which manifests itself especially in the performance practice of live coding [24]. A practice in which the on-the-fly programming of algorithms for the generative composition of music and/or visuals is performed live in front of an audience [25,26]. As programming languages, these can also be embedded well in new technologies, such as web-based applications in the browser [27,28,29]. Therefore also into web-based metaverse envi-

ronments. Embedding live coding into such environments allows live performances to be executed within the platform without the need to stream audio and video from local computers. For VERSNIZ we implemented an appropriate programming language for audiovisual live coding in metaverse environments and present a performance practice that combines the advantages of metaverse environments with the gamification concept of worldbuilding.

3 Concept

Our objective for VERSNIZ was to create a metaverse environment where multiple users can create a transforming virtual world as a collaborative performance or installation through audiovisual live coding. For this purpose, as in many worldbuilding video games, objects can be placed in virtual space by the user (see Fig. 1). These objects have a programming terminal and can be live-coded in the IBNIZ programming language. The algorithms simultaneously create the visual form, the animation, and the music. The placement and movement of the live coding objects, as well as the design of the acoustics of the virtual environment, allow for the additional application of spatial composition techniques [30].
Through movement in six degrees of freedom (6-DoF), the individual selection of what is seen (field-of-view) and heard is influenced by the head orientation and the position in space. Different visual details and a different ”mix” of auditory components have a significant and individual influence on the perceived art experience.
Various criteria were considered during the implementation of a suitable system to meet the definition of a metaverse environment: along with the fundamental requirement of being a multiuser virtual platform, particular focus was put on the resulting immersion. In addition to the 3D rendering of the visual environment, special attention was also paid to the spatial audio rendering. Here, three-dimensional 6-DoF audio reproduction is realized with binaural synthesis for headphone-based auralization [31]. An adequate auditive room simulation was additionally implemented. Compliance with the WebXR [32] standard ensures the use of common VR/AR end devices. While compatibility with VR and AR devices using head-mounted displays (HMDs), controllers, hand or room tracking enables immersive experiences, the ability to use a conventional computer or mobile device ensures a low barrier to entry for the widest range of users. As a web-based application, users do not need to perform platform-specific installations, and the environment is automatically networked as the metaverse concept intends. Integrating audio and video streaming or text-based chats enables additional forms of interaction and enhances the social component of the experience. In addition, the possibility of modifying the virtual environment allows new types of stage and design concepts that can be realized with low effort compared to physical reality, and would otherwise be difficult or impossible to realize.

4 Implementation

Implementing a system with the mentioned features requires a complex interchange of various programming languages, frameworks, libraries, and interfaces (see Fig. 2).

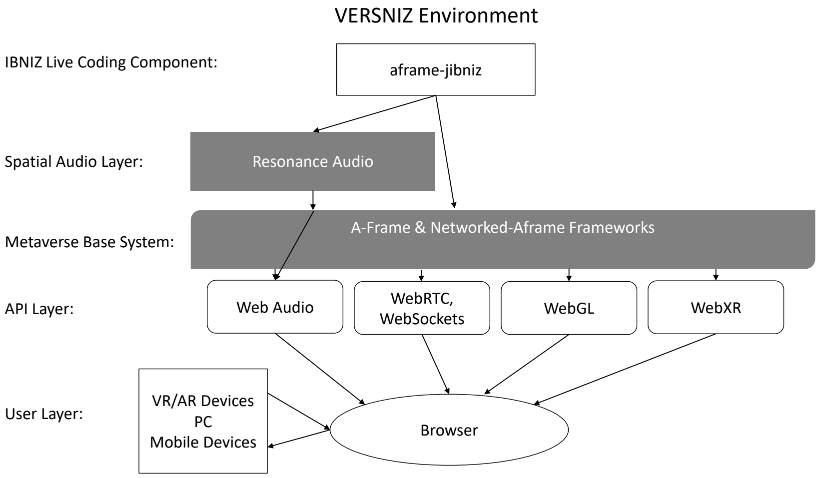

Fig. 2. The architecture of the VERSNIZ metaverse environment. The graphic shows the involved frameworks, libraries, and programming interfaces.

To comply with the described requirements for a metaverse environment, a selection of suitable web technologies was made. Thus, the virtual environment was developed based on the A-Frame framework [33]. Together with the library Networked-Aframe [34], shared, multiuser virtual environments can be implemented. This combination is well established, being the basis of the popular Mozilla Hub metaverse systems. The binaural rendering of A-Frame was extended with an implementation of the Resonance Audio spatializer [35]. The most critical component for the audio-visual composition is the programming language for live coding. The programming language IBNIZ unites various features that are particularly beneficial for programming in VR or AR and enables simultaneous programming of audiovisual algorithms. The following describes the development of these elements in more detail.

Metaverse Base System The web-based multiuser virtual environment was mainly realized with the A-Frame framework. A-Frame is a framework developed in JavaScript that enables the programming of virtual environments for VR and AR abstracted in HTML-like syntax. It is based on the JavaScript library Three.js [36] for programming WebGL applications. The HTML-like abstraction allows it to start designing 3D virtual worlds easily, while the access to Three.js makes it still very powerful. Embedding the WebXR application programming interface (API) enables integration with common VR/AR hardware and mobile devices - with stereoscopic rendering on HMDs and interaction via controllers and tracking. A-Frame applications can still be used with conventional computer hardware via screen, mouse, and keyboard. This allows most users to experience it without requiring special hardware. A-Frame provides various so-called ”components” that can be used to program 3D geometries and models, mate-

rials, lights, shadows, and multimedia content such as images, videos, and sounds into a virtual environment. The framework is also arbitrarily extendable by programming custom A-Frame components in JavaScript.
The Networked-Aframe library builds on A-Frame and enables the programming of multiuser environments for VR and AR. Networked-Aframe offers various adapters for data exchange via WebRTC with a WebSockets fallback. The data is transferred from user to user in a so-called peer-to-peer (P2P) network. The integration of the WebRTC standard [37] enables the low-latency transmission of audio and video streams, so that video and audio chats can be realized. Text can also be transmitted in-between users, as can the parameters of all A-Frame components, including custom-developed ones. In addition to sharing data, Networked-Aframe also provides templates for implementing avatars and synchronized interaction with the environment to ensure interactivity and persistence of the virtual environment.
For VERSNIZ, Networked-Aframe was integrated using the default EasyRTC adapters. Besides the synchronization of avatars and user interaction, the IBNIZ source code of the audiovisual live coding objects is shared. The virtual world unfolds its immersive potential when immersive end devices are used [38]. Stereoscopic rendering on HMDs and tracking head and movement in the room, create the feeling of presence in virtual environments. Not only is the image rendered to match the user's perspective, but so is the sound. A-Frame, in combination with the Web Audio API [39], already creates a dynamic three-dimensional sound experience through binaural rendering for headphone playback using real-time convolution with head-related transfer functions [31]. Sound sources are rendered at the appropriate location depending on the head orientation and the user's position in the room. To increase acoustic plausibility, a spatial room simulation for reverberation that considers the material properties of reflective surfaces, was added. This makes it possible to match the visually designed environment acoustically and to increase audiovisual coherence [40]. As there is already an A-Frame port [41] of the Resonance Audio spatializer, and Resonance Audio can be considered an appropriate choice for web-based applications [42], the existing port was extended and implemented in conjunction with the IBNIZ live coding component [43]. Based on these technologies, any desired immersive world can be created to represent the stage in VERSNIZ.

IBNIZ Live Coding Component IBNIZ, ”Ideally Bare Numeric Impression giZmo”, is a virtual machine for low-level programming of audiovisual algorithms [16] which is closely related to the Bytebeat concept [44]. It was developed by Ville-Matias ”Viznut” Heikkilä and is linked to ideas present in the Demoscene. This is reflected in the minimalistic design of the language, resulting in the reduced instruction set, consisting of only one character per instruction.
Through this minimalist approach, IBNIZ is often considered an esoteric programming language [45]. A kind of programming language developed out of the motivation to implement experimental, weird, or sometimes artistic concepts rather than to pursue a practical use. Here many results can be already considered software art themselves.
The language design of IBNIZ also has artistic characteristics in it, and it is at the same time a domain-specific language [46] for the programming of further 2D video and au-

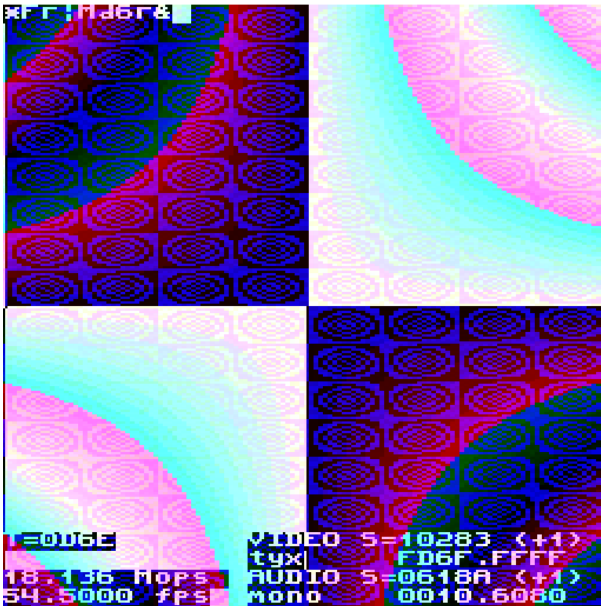

Fig. 3. A screen capture from the original IBNIZ code editor. It shows an example of the same algorithm from Fig. 1.

dio art (see Fig. 3).
The language's minimalism is not only ideal for live coding in real-time performances, allowing the artist to program expressive algorithms quickly and with few characters only, but also for typing with virtual keyboards on VR/AR devices. While the use of virtual keyboards in VR is often limited to short texts due to the difficulty of using them with the available input devices [47], the advantage of IBNIZ as the chosen language is particularly evident here. The entire instruction set can be placed on a single virtual keyboard view to code the desired algorithms quickly, even with controllers, the mouse, or hand tracking. Building on a web-based port 'jibniz' [48] in JavaScript, we realized an implementation of IBNIZ as an audiovisual live coding language for metaverse environments. The IBNIZ virtual machine is implemented as an A-Frame and Networked-Aframe compatible component 'aframe-jibniz' [49] using the Web Audio API. The resulting audio is linked to a Resonance Audio source and can be rendered spatially depending on its position in relation to the listener. The visual output serves as a displacement texture for arbitrary 3D geometries; in this way, using algorithms, an animated, constantly changing object is generated. In combination, three-dimensional audiovisual objects are created that continuously evolve in real-time. Each object can be programmed independently with a programming terminal and a virtual keyboard, implemented using the Aframe-Super-Keyboard component [50]. The written code and the objects' position are also synchronized P2P using Networked-Aframe with all other users. Since the rendering is client-based for each user, neither audio nor video needs

to be streamed, only the text of the programmed code. This way, low bandwidth is used to transmit only text, enabling low-latency, real-time performances.

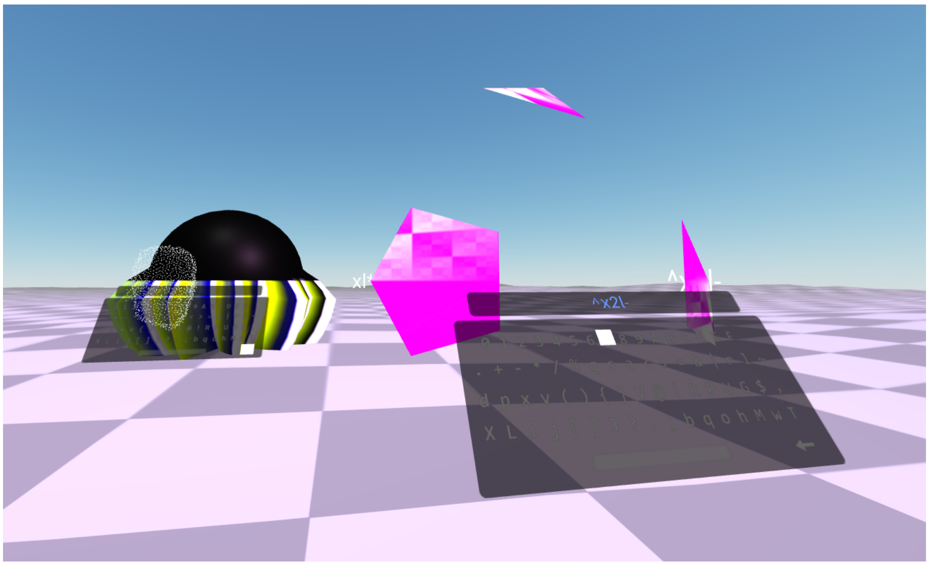

Fig. 4. A screen capture of a collaborative live coding scenario in VERSNIZ. Two people represented as a head point-cloud avatar performing together in the default environment setup.

5 Use Cases

With VERSNIZ we provide an exemplary template implementation of the 'aframe-jibniz' live coding component into an A-Frame/Networked-Aframe metaverse base system. Using this template, users can create arbitrary virtual worlds and art experiences. Within the possibilities offered by A-Frame, artists can freely design virtual environments, also specifically for use with VR or AR systems. It is intended for performances or installations using the proposed worldbuilding concept (see Sec. 3): the placement and live coding of multiple IBNIZ audiovisual objects in this environment, to enable the algorithmic composition of a constantly changing virtual world. The composition can then take place in real-time within the virtual experience, rather than being produced in advance. The resulting music and the 3D visualization of the objects change constantly depending on the algorithm. The location of their placement adds a spatial component to the composition and allows for extensive integration of spatial composition techniques. Placement and live coding can be done by any user or restricted to specific performers, allowing for different levels of audience engagement. The audience can not only simultaneously take on the role of the performer by live coding themselves, but they can also have their individual experience as passive spectators by interacting with the environment, for example by freely roaming around the world of various audiovisual objects. The recipient is free to move around the virtual environment at any time; the artist has no control over the time and place where the objects are experienced. This adds a spatial component to a time-continuous performance whose control is entirely in the hands of the recipient.

The above-mentioned properties allow a performance practice specifically tailored to the possibilities of metaverse environments, where all actors (performers and audience) come together in an immersive virtual world. The spatial aspect is important here; multiple IBNIZ live coding objects allow an audiovisual composition to be placed as individual fragments in space. This concept, inspired by worldbuilding videogames, allows different composition and performance concepts in combination with audiovisual live coding:

1. virtual worlds can be used as a shared playground for free-form audiovisual creation by different users
2. as a multiuser environment, it can be used for networked music performances in the form of collaborative improvisations or rehearsed compositions (see Fig. 4)
3. artists can create immersive, persistent audiovisual installations in virtual worlds (see Fig. 5)

With VERSNIZ, a new dimension is added to live coding as a performance practice, while the medium of virtual environments offers a high degree of freedom in designing artistic experiences. The gamification concept of worldbuilding allows for a new way of spatially distributed collaboration. A demo of the default VERSNIZ template implementation can be found at: `https://versniz.glitch.me/`

A video with a brief demonstration of the concept and mechanics is available at:
`https://youtu.be/O4TmE1-bth4`

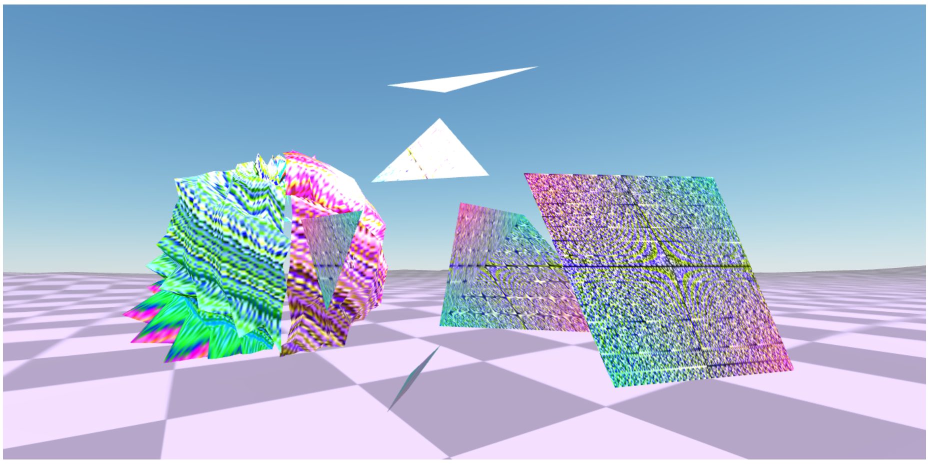

Fig. 5. A screen capture of multiple audiovisual objects composed into a sculpture, as an example for an audiovisual installation.

6 Conclusion & Future Work

With VERSNIZ we have created a metaverse environment for networked, collaborative live coding of audiovisual worlds. It allows for various novel performance and composition concepts characterized by their spatial aspects. This makes it possible to create art experiences that would be difficult or impossible to realize in physical reality. As an open-source environment, it allows artists a high degree of freedom in design, customization, and expansion. With the incorporation of the worldbuilding concept inspired by video games, we have described a specific performance practice that particularly benefits from the advantages of metaverse environments. The primary constraint is the programming language IBNIZ. While its minimalist design has significant advantages for programming in VR/AR, it is limited in stylistic variety. The properties of the IBNIZ virtual machine in terms of visual and audio resolution or the lack of external media integration, such as images and audio samples, limit the results to a lo-fi 8-bit-like aesthetic.
The current implementation of VERSNIZ works best for Chromium-based browsers and can be found at: `https://github.com/AudioGroupCologne/VERSNIZ` However, it is still under continuous development to increase compatibility with browsers and end devices, improve the user experience, provide additional features for artists, and improve stability and performance. Referring to the limitations of IBNIZ mentioned above, also other systems for performances in metaverse environments [51] are being developed parallel to VERSNIZ. This includes additional programming languages for live coding, systems for programming and performing with virtual instruments, and performances using real-time streaming of volumetric audio and video [52].

References

1. Neal, S.: Snow Crash. New York: Bantam Books (1992)
2. Mystakidis, S.: Metaverse. Encyclopedia 2(1), 486–497 (2022)
3. Park, S.M., Kim, Y.G.: A Metaverse: Taxonomy, Components, Applications, and Open Challenges. IEEE Access 10, 4209–4251 (2022)
4. Lee, L., Lin, Z., Hu, R., Gong, Z., Kumar, A., Li, T., Li, S., Hui, P.: When creators meet the metaverse: A survey on computational arts. CoRR abs/2111.13486 (2021), `https://arxiv.org/abs/2111.13486`
5. Turchet, L.: Musical Metaverse: vision, opportunities, and challenges. Personal and Ubiquitous Computing (2023)
6. Elen, R.: Music in the metaverse. Journal of the Audio Engineering Society (April 2008)
7. Oliveros, P., Weaver, S., Dresser, M., Pitcher, J., Braasch, J., Chafe, C.: Telematic music: Six perspectives. Leonardo Music Journal 19, 95–96 (2009)
8. Rottondi, C., Chafe, C., Allocchio, C., Sarti, A.: An overview on networked music performance technologies. IEEE Access 4, 8823–8843 (2016)
9. Turchet, L., Hamilton, R., Camci, A.: Music in Extended Realities. IEEE Access 9, 15810–15832 (2021)
10. Loveridge, B.: Networked Music Performance in Virtual Reality: Current Perspectives. Journal of Network Music and Arts 2(1), 2 (2020)
11. Men, L., Bryan-Kinns, N.: Supporting Sonic Interaction in Creative, Shared Virtual Environments, pp. 237–267. Springer International Publishing, Cham (2023), `https://doi.org/10.1007/978-3-031-04021-4_8`

12. Wakefield, G., Ji, H.: Artificial nature: Immersive world making. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 5484 LNCS, pp. 597–602 (2009)
13. Wakefield, G., Smith, W.: Cosm : a Toolkit for Composing Immersive Audio-Visual Worlds of Agency and Autonomy. In: Proceedings of the International Computer Music Conference (2011)
14. Hamilton, R.: Collaborative and competitive futures for virtual reality music and sound. In: 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). pp. 1510–1512 (2019)
15. Ciciliani, M.: Virtual 3D environments as composition and performance spaces*. Journal of New Music Research 49(1), 104–113 (jan 2020)
16. V. Heikkilä, IBNIZ. `http://viznut.fi/ibniz/`, accessed: 2023-02-15
17. Microsoft: Minecraft (2023), `https://www.minecraft.net/`, accessed: 2023-02-15
18. Foundation, M.: Mozilla Hubs (2023), `https://hubs.mozilla.com/`, accessed: 2023-02-15
19. Toplap Berlin: Mozilla Hubs Concertspace (2023), `https://hubs.mozilla.com/oUQRigk/toplap-berlin/`
20. Champlin, A., Knotts, S., Chicau, J., Xambó, A., Saladino, I.: Community Report : LivecoderA. In: 7th International Conference on Live Coding (ICLC). pp. 1–5 (2023)
21. T. Scott, Epic Games: Travis Scott and Fortnite Present: Astronomical (Full Event Video) (2023), `https://www.youtube.com/watch?v=wYeFAlVC8qU`
22. VRROOM: Jean Michel Jarre - ZERO GRAVITY - IN VR - Welcome to the Other Side (2023), `https://vimeo.com/547145530`, accessed: 2023-02-15
23. VRROOM: Jean Michel Jarre, Oxymore 2 - Overview (2023), `https://vimeo.com/767098450/6fcf797c5a`, accessed: 2023-02-15
24. Nilson, C.: Live coding practice. In: Proceedings of the 7th International Conference on New Interfaces for Musical Expression. p. 112–117. NIME '07, Association for Computing Machinery, New York, NY, USA (2007), `https://doi.org/10.1145/1279740.1279760`
25. Collins, N., McLean, A., Rohrhuber, J., Ward, A.: Live coding in laptop performance. Organised Sound 8(3), 321–330 (2003)
26. Juan Romero, Borgeat, P.: Live-Coding – programming masterly music. In: TEDx KIT (2018), `https://www.ted.com/talks/juan_romero_patrick_borgeat_live_coding_programming_masterly_music_jan_2018`, accessed: 2023-02-15
27. Roberts, C., Kuchera-Morin, J.A.: Gibber: Live coding audio in the browser. ICMC 2012: Non-Cochlear Sound - Proceedings of the International Computer Music Conference 2012 pp. 64–69 (2012)
28. Ogborn, D., Beverley, J., Navarro Del Angel, L., Tsabary, E., McLean, A.: Estuary: Browser-based Collaborative Projectional Live Coding of Musical Patterns. In: International Conference on Live Coding (2017)
29. Lan, Q., Jensenius, A.R.: QuaverSeries : A Live Coding Environment for Music Performance Using Web Technologies. In: Web Audio Conference WAC-2019, (2019)
30. Baalman, M.A.: Spatial composition techniques and sound spatialisation technologies. Organised Sound 15(3), 209–218 (2010)
31. Møller, H.: Fundamentals of binaural technology. Applied Acoustics 36(3-4), 171–218 (1992)
32. W3C: WebXR (2023), `https://immersiveweb.dev/`, accessed: 2023-02-15
33. A-Frame: Homepage (2023), `https://aframe.io`, accessed: 2023-02-15
34. Networked-Aframe: Git Repository (2023), `https://github.com/networked-aframe`, accessed: 2023-02-15

35. Resonance Audio: Git Homepage (2023), https://resonance-audio.github.io/resonance-audio/, accessed: 2023-02-15
36. Three.js: Homepage (2023), https://threejs.org/, accessed: 2023-02-15
37. Developer, G.: WebRTC (2023), https://webrtc.org/, accessed: 2023-02-15
38. Slater, M., Lotto, B., Arnold, M.M., Sanchez-Vives, M.V.: How we experience immersive virtual environments: The concept of presence and its measurement. Anuario de Psicologia 40(2), 193–210 (2009)
39. W3C: Web Audio (2023), https://www.w3.org/TR/webaudio/, accessed: 2023-02-15
40. Werner, S., Klein, F., Mayenfels, T., Brandenburg, K.: A summary on acoustic room divergence and its effect on externalization of auditory events. In: 2016 8th International Conference on Quality of Multimedia Experience, QoMEX 2016. pp. 1–6. IEEE (2016)
41. Kungla, M.: A-Frame Resonance Audio Git Repository (2023), https://github.com/mkungla/aframe-resonance-audio-component/
42. McArthur, A., Tonder, C.V., Gaston-Bird, L., Knight-Hill, A.: A survey of 3D audio through the browser: Practitioner perspectives. In: 2021 Immersive and 3D Audio: From Architecture to Automotive, I3DA 2021. Institute of Electrical and Electronics Engineers Inc. (2021)
43. Dziwis, D.: A-Frame Resonance Audio Git Repository (2023), https://github.com/AudioGroupCologne/aframe-resonance-audio-component
44. Heikkilä, V.M.: Discovering novel computer music techniques by exploring the space of short computer programs pp. 1–8 (2011), http://arxiv.org/abs/1112.1368
45. Temkin, D.: Language without code: Intentionally unusable, uncomputable, or conceptual programming languages. Journal of Science and Technology of the Arts 9(3 Special Issue), 83–91 (2017)
46. Voelter, M., Benz, S., Dietrich, C., Engelmann, B., Helander, M., Kats, L., Visser, E., Wachsmuth, G.: DSL Engineering - Designing, Implementing and Using Domain-Specific Languages (2013)
47. Grubert, J., Witzani, L., Ofek, E., Pahud, M., Kranz, M., Kristensson, P.O.: Text Entry in Immersive Head-Mounted Display-Based Virtual Reality Using Standard Keyboards. In: 25th IEEE Conference on Virtual Reality and 3D User Interfaces, VR 2018. pp. 159–166. IEEE (2018)
48. L. Escot, jibniz. https://github.com/flupe/jibniz, accessed: 2023-02-15
49. Dziwis, D.: aframe-jibniz. https://github.com/AudioGroupCologne/aframe-jibniz, accessed: 2023-02-06
50. Supermedium: Aframe-Super-Keyboard (2023), https://github.com/supermedium/aframe-super-keyboard, accessed: 2023-02-15
51. Dziwis, D., von Coler, H., Pörschmann, C.: Orchestra: a Toolbox for Live Music Performances in a Web-Based Metaverse. Journal of the Audio Engineering Society pp. 1–11 (2023), (accepted for publication)
52. Dziwis, D., von Coler, H.: The Entanglement – Volumetric Music Performances in a Virtual Metaverse Environment. Journal of Network Music and Arts 5(1), 1–12 (2023)