

This is a repository copy of *Creating Audio Object-focused Acoustic Environments for Room-Scale Virtual Reality*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/189360/>

Version: Accepted Version

---

**Article:**

Popp, Constantin and Murphy, Damian Thomas [orcid.org/0000-0002-6676-9459](https://orcid.org/0000-0002-6676-9459) (2022)  
Creating Audio Object-focused Acoustic Environments for Room-Scale Virtual Reality.  
Applied Sciences (Switzerland). 7306. ISSN 2076-3417

<https://doi.org/10.3390/app12147306>

---

**Reuse**

This article is distributed under the terms of the Creative Commons Attribution (CC BY) licence. This licence allows you to distribute, remix, tweak, and build upon the work, even commercially, as long as you credit the authors for the original work. More information and the full terms of the licence here:

<https://creativecommons.org/licenses/>

**Takedown**

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing [eprints@whiterose.ac.uk](mailto:eprints@whiterose.ac.uk) including the URL of the record and the reason for the withdrawal request.

# Creating Audio Object-focused Acoustic Environments for Room-Scale Virtual Reality

Constantin Popp <sup>1,†</sup>  and Damian Murphy <sup>1,‡</sup>

<sup>1</sup> University of York <sup>1</sup>; damian.murphy@york.ac.uk

\* Correspondence: constantin.popp@york.ac.uk

‡ These authors contributed equally to this work.

**Abstract:** Room-scale Virtual Reality (VR) affordance in movement and interactivity causes new challenges in creating virtual acoustic environments for VR experiences. Such environments are typically constructed from virtual interactive objects that are accompanied by an Ambisonic bed and an off-screen ("invisible") music soundtrack, with the Ambisonic bed, music, and virtual acoustics describing the aural features of an area. This methodology can become problematic in room-scale VR as the player cannot approach or interact with such background sounds, contradicting the player's motion aurally and limiting interactivity. Written from a sound designers perspective, the paper addresses these issues by proposing a musically inclusive novel methodology that reimagines an acoustic environment predominately using objects that are governed by multimodal rule-based systems and spatialized in six degrees of freedom using 3D-binaural audio exclusively while minimizing the use of Ambisonic beds and non-diegetic music. This methodology is implemented using off-the-shelf, creator-oriented tools and methods and is evaluated through the development of a standalone, narrative, prototype room-scale VR experience. The experience's target platform is a mobile, untethered VR system based on head-mounted displays, inside-out tracking, head-mounted loudspeakers or headphones, and hand-held controllers. The authors apply their methodology to the generation of ambiances based on sound-based music, sound effects, and virtual acoustics. The proposed methodology benefits the interactivity and spatial behavior of virtual acoustic environments but may be constrained by platform and project limitations.

**Keywords:** Virtual Reality; sound design; spatial audio; six degrees of freedom; game audio; room-scale; spatial interaction; human-computer interaction; audio and haptic interfaces;

**Citation:** Popp, C.; Murphy, D.; Creating Audio Object-focused Acoustic Environments for Room-Scale Virtual Reality. *Appl. Sci.* **2022**, *1*, 0. <https://doi.org/>

Received:

Accepted:

Published:

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Copyright:** © 2022 by the authors. Submitted to *Appl. Sci.* for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Room-scale VR using head-mounted displays (HMD) allows the player to move around in virtual environments, causing new challenges in creating engaging user experiences [1]. This affordance in movement creates a sense of agency in the player that can be addressed through participatory, process-oriented narratives [2]. This affordance for movement and interactivity stems from modern VR systems, such as the Meta Quest 2 [3], being able to track the position and rotation of the player's limbs in six degrees of freedom (6-DoF). However, creating audio that supports these affordances requires new methodologies [4, Ch. 3].

Current recommendations for VR sound design suggest separating an audio mix into foreground and background sounds and applying dedicated spatialization techniques [5]. For example, foreground sounds would be spatialized into a sound-field representation according to their relative position and rotation to a virtual listener ("object-based rendering") and accompanied by an additional sound-field recording of an acoustic environment ("Ambisonic bed"), such as a recording of a forest or empty office [5,6]. The resulting sound-field would be transformed into a 3D-binaural representation for a listener's left and right ear, based on the listener's head rotation [7, p. 9].

However, such Ambisonic beds can become problematic in large, interactive virtual environments, as the player cannot approach or interact with the sounds of the Ambisonic bed. This is due to the latter being fixed to the player's position [5]. This behavior can break the impression of the player's movement in space, as illustrated in [Example 1](#) via a fly-through of a virtual outdoor environment that uses a single looped Ambisonic recording [8]. This issue can also be observed at the beginning of the VR game *Virtual Virtual Reality 2* [9], in which the position of the visual representation of a flock of birds does not align with its acoustic representation [10, [timecode 3:44](#)].

The potential for contradicting the player's movement in space via positionally unresponsive audio can be further aggravated using invisible, non-diegetic music. Here, the music would be placed outside the game world via stereophonic panning [11]. This placement prevents the player from approaching its source and restricts the music's response to the player's head movement, causing it to appear head-locked [12]. The spatial cues of the music, therefore, may differ from the spatial cues of the game world, which could lead to confusion among the players.

Unfortunately, replacing the use of Ambisonic beds or non-diegetic music through an alternative methodology is non-trivial.

Recordings not only just capture sounds, but also their behavior over time, such as the density of occurrence of a particular sound or their change in timbre [13]. Furthermore, the composition of acoustic environments can vary over time, affected, for example, by the time of day [14]. A recording of such an environment may capture such changes, which a composer could choose to use creatively [15]. In this sense, recordings also capture the sonic organization of an environment at one or more moments in time. Consequently, if the use of Ambisonic beds/non-diegetic music would be replaced by another methodology the latter needs to include a method to describe the temporal (and spatial) evolution of its associated sound sources.

Furthermore, the type of interactivity to which audio reacts must be considered. Interactivity can be differentiated between active and passive player participation [16]. Passive participation would require the rendering of sounds relative to the user's head orientation and position in space; active participation would require direct manipulable elements to the user [16]. An experience should cater to both types of participation to "achieve a total experience" [17].

Additionally, the multimodal nature of VR must also be considered, since humans generally experience the physical world through more than one sense [17]. For this reason, interaction feedback should be provided using multimodalities, such as mapping the change in object properties to visualizations, vibration, and audio simultaneously [17].

To address the issues of interactive spatial audio for room-scale VR, this paper investigates the creation of virtual acoustic environments through the establishment of virtual objects whose behavior can be described by rules and whose audio content is spatialized in 6-DoF using 3D-binaural audio exclusively. This methodology aims to reimagine the temporal structure of an acoustic environment using rule-based systems wherever possible rather than relying on Ambisonic beds or non-diegetic music as surrounding ambiances. Firstly, this recreation allows the sound designer the injection of new, multimodal behaviors to cater to interactivity. Secondly, an object-focused, 6-DoF 3D-binaural audio approach to acoustic environments allows a player to understand their proximity to the environment's objects. Similar approaches may have been used in VR games, such as *Half-Life: Alyx* [18]; however, documentation of such approaches may be lacking, and this paper aims to help fill this gap. Written from a sound designer's perspective and using the taxonomy of sounds in VR presented in [19], this paper bases its novel methodology on off-the-shelf tools and methods, while targeting mobile VR as presentation platform.

The structure of the paper is as follows: Section 2 reviews the literature to find possible solutions to the issues raised in the paper. Section 3 describes the methodology of the paper's approach. Section 4 describes the implementation of the methodology based on

example applications, which are evaluated with comparable alternative solutions in Section 5. Section 6 discusses the limitations of the proposal. Finally, Section 7 concludes.

## 2. Background

The introduction raised issues that a suitable methodology for spatial audio design for room-scale VR should address. The literature and tools for game engines propose several approaches to these issues.

### 2.1. 6-DoF audio

Object-based and sound-field rendering approaches have been used to implement passive player participation ("6-DoF audio") [6]. For example, 30 time-aligned third-order Ambisonic recordings of an orchestra performance are interpolated based on the listener's position and rotation in their room-scale VR music experience *Zylia Concert Hall* using the sound-field rendering technique [20,21]. [22] uses an object-based rendering method of multiple mono-recordings of an orchestra performance. Méndez *et al.* use both rendering methods in their experience depending on the source material [6]. In contrast to sound-field rendering, object-based rendering also considers a sound source's radiation pattern and interaction with the "virtual" acoustic environment during the rendering process [6]. This approach gives the sound designer deep control over the acoustic appearance of a sound source in space.

The cited experiences, however, differ from our work in that they focus on the re-creation of existing music performance and its performance space, rather than arbitrary (imaginary) acoustic environments or active player interaction. As the corresponding standard for 6-DoF audio (MPEG-I) is currently being developed [23], [21–23] use their own custom toolchain, while [6] uses game technologies for both visuals and audio.

### 2.2. Adaptive audio

Within the context of game audio, active player participation can be addressed with adaptive audio, that is, audio that responds to changes in game states [24]. This adaptation is made by separating audio into a collection of sound files and rules, the latter defining the former's playback in the audio middleware or game engine [25, Ch. 8,9]. Game states can trigger the playback of specific audio files, as used in Byrns' *et al.* adaptive music therapy in VR [26], or affect the processing of sounds in real-time, as seen in the audification of the state of virtual objects [27, [timecode 10:33](#)] in the VR climbing simulation *The Climb 2* [28].

The separation of audio content and its rules of playback can also be used for interactive sound effects in VR. For example, in the VR game *The Mage's Tale* [29], [30] links a collection of "click sounds" to conditions of an angular movement: The player rotates a "wheel", with the latter responding with a clicking sound every 30°.

### 2.3. Rule-based systems

Rule-based systems, generative processes, or procedural content generation have been widely used for content generation in games. Applications of procedural content generation include animation, level design, stories or games, as well as music, soundscapes, and sound effects [31–35]. Recent advances apply Artificial Intelligence (AI) in procedural music to compose music in real-time that is affected by player preferences and/or the environment [36], or responsive to state changes of non-player characters (NPC) [37].

The non-VR game *Spore* [38] is an example of a rule-based system that composes and performs music "automatically" [33, p. 11]. It generates multiple independent musical phrases which can be randomly combined into complex combinations [33, p. 11]. To exert control over the resulting music to ensure aesthetic validity, the composer (Brian Eno) creates rulesets that predetermine important musical parameters, such as restricting pitches to a specific scale or creating patterns of rhythms and melodies that the player can randomly combine [39]. Some of these patterns are associated with visual representations through which the player can affect the music being generated [40, [timecode 5:34–6:47](#)].

However, the examples of this section focus on non-diegetic music, stereophonic audio playback through sound synthesis, and their generative algorithms affect only the music's modality. In the case of *Spore*, the music responds to the player's interaction with visual objects, but is presented non-diegetic and is not spatially correlated with the visual objects.

#### 2.4. Spatial implementation of music in room-scale VR

The spatial impact and spatial implementation of non-diegetic music in VR are debated [11,12]. Despite the potential to detract from the spatial audio experience [12], non-diegetic music can guide players through several game spaces, potentially reducing the impact of the event boundary phenomenon [41] and, therefore, improving the spatial cohesion of games [42]. Composers have also sought to soften or exaggerate the spatial disconnect between diegetic sound effects and non-diegetic music, depending on a game's context [43].

Visualizations of non-diegetic music can be found mainly in music-centered games [19, p. 166], but tend not to correlate spatially with the music's "instruments". For example, *Beat Saber* [44] visualizes various features of non-diegetic music, such as rhythm, tension, or contrasts, but limits spatial audio only to sound effects caused by hand interaction [45, [timecode 0:17](#)].

Alternatively, diegetic music would allow the player to approach its source and share its acoustic space with the player. This can be seen in the two music examples in Section 2.1. However, placing the music's instruments across the acoustic environment is challenging in the author's development platform, as FMOD stores audio as events and links them to a single position in space when used with a 3D audio spatializer [46, chap. 16.2.3]. Thus, implementing multi-source diegetic music in FMOD requires an alternative solution.

#### 2.5. Authors' contribution

Table 1 summarizes the similarities and differences between the cited and our work. Although our work is most similar in the construction of acoustic environments to the methodology used in *Half-Life: Alyx*, *The Mage's Tale*, and *Vacation Simulator* [47], it differs in the combination of exclusive use of 3D-binaural audio in 6-DoF, multi-source diegetic music, and rule-based systems that generate (sound-file-based) audio and map its behavior to several modes simultaneously. Furthermore, our work targets the Meta Quest 2 as the presentation medium and implements velocity-sensitive audio objects in select cases. [26,36,37] do not specifically mention the music's spatial format or reference any point / surrounding ambiances.

A methodology based on multimodal, object-, and rule-based systems is not new per se. For example, [48] describes a similar methodology in the context of Ambient Intelligence, where a collection of "cognitive", context-aware agents perform functionalities of a service-oriented architecture, such as controlling the playback and type of music based on the user's mood and environmental factors (temperature, outdoor light). Such systems have also been applied, for example, to intelligent music recommendation systems [49] or to context-aware music service for gym users [50].

However, considering Table 1, our work can be considered new in the context of spatial audio and the design of acoustic environments presented in room-scale VR. Although the technologies and methods used are not new, we propose a novel methodology based on their combination. This approach may leverage existing knowledge of sound designers and thus facilitate adoption.

### 3. Methods

The paper proposes a multimodal, (audio) object-focused methodology for the creation of acoustic environments presented in room-scale VR. Generally, the methodology builds upon the production cycle of game audio, as described in [25,54], but affects steps in the sound generation and implementation process.

**Table 1.** Comparison of differences and similarities of the cited experiences / papers regarding 6-DoF audio, generative audio, diegetic music, and the mapping of the music's features to other senses or objects, i.e. visualizations of rhythms or representation by visible objects. "N/a" is entered if there is not sufficient information available to identify a characteristic.

| Title                              | Type          | 6-DoF | 6-DoF audio          | Diegetic music | Generative music          | Surrounding ambiances | Music mapping  |
|------------------------------------|---------------|-------|----------------------|----------------|---------------------------|-----------------------|----------------|
| Virtual Reality 2                  | Adventure     | yes   | partial              | no             | no                        | yes                   | no             |
| Half-Life: Alyx                    | FPS shooter   | yes   | partial              | partial        | no                        | yes                   | object         |
| Zylia Concert Hall                 | Music         | yes   | yes                  | yes            | no                        | no                    | avatars        |
| Settel <i>et al.</i> , 2021        | Music         | yes   | yes                  | yes            | no                        | no                    | level meters   |
| Méndez <i>et al.</i> , 2018        | Music         | yes   | yes                  | yes            | no                        | no                    | avatars        |
| Byrns <i>et al.</i> , 2020         | Music therapy | n/a   | n/a                  | n/a            | no                        | n/a                   | visualizations |
| The Climb 2                        | Sport         | yes   | partial              | no             | no                        | yes                   | no             |
| The Mage's Tale                    | Action RPG    | yes   | partial              | no             | no                        | yes                   | no             |
| De Prisco <i>et al.</i> , 2016     | n/a           | n/a   | n/a                  | no             | evolutionary              | n/a                   | visualizations |
| Washburn and Khosmood, 2020        | n/a           | n/a   | n/a                  | n/a            | multi-agent expert system | n/a                   | visualizations |
| Spore                              | Simulation    | no    | no                   | no             | rule-based                | yes                   | objects        |
| Fail Factory                       | Arcade        | no    | partial <sup>1</sup> | partial        | no                        | n/a                   | visualizations |
| Beat Saber                         | Music         | yes   | partial              | no             | no                        | no                    | multimodal     |
| 12 Sentiments                      | Music         | n/a   | partial              | partial        | n/a                       | no                    | multimodal     |
| Rez Infinite [51]                  | Music         | no    | no                   | partial        | rule-based                | music                 | multimodal     |
| Eleven Table Tennis [52]           | Sport         | yes   | yes                  | yes            | no                        | n/a                   | object         |
| Vacation Simulator                 | Simulation    | yes   | yes                  | partial        | no                        | yes                   | object         |
| Please, Don't Touch Anything! [53] | Puzzle        | yes   | partial              | partial        | no                        | yes                   | object         |

<sup>1</sup> The game's description in [43] is unclear regarding the differentiation of 3D audio supporting 3-DoF or 6-DoF.

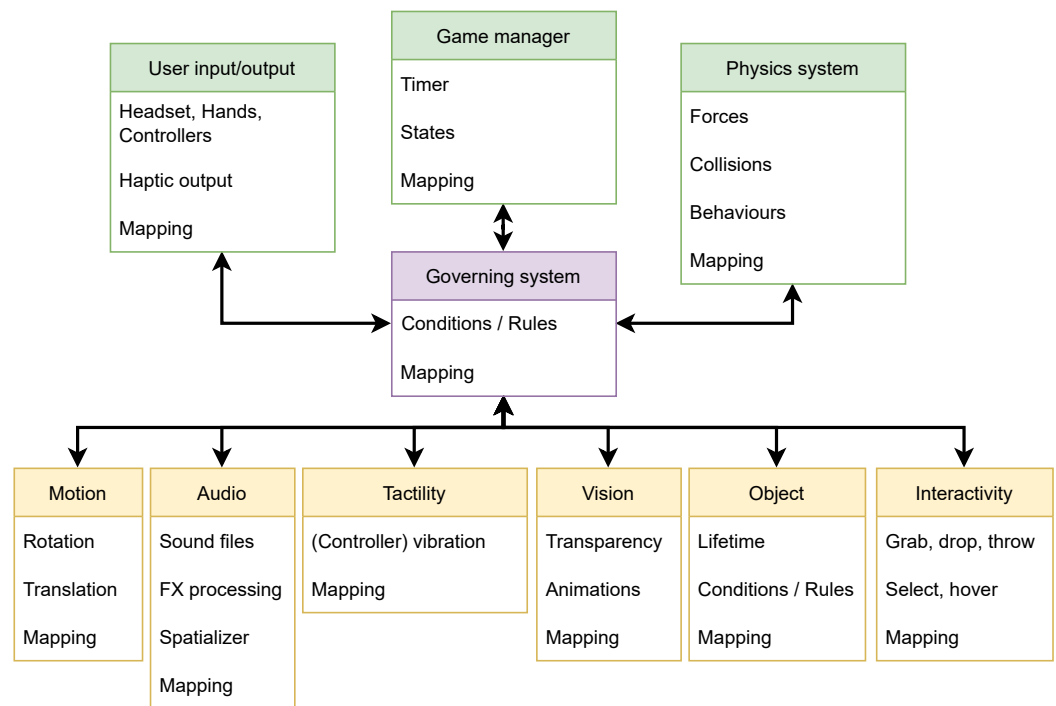
Traditionally, surrounding ambiances would be created in a digital audio workstation (DAW) and the mixed result is imported as a single sound file (or a collection of sound files corresponding to stems/tracks of this audio mix) into the audio middleware/game engine for playback [5]. This approach means that the structure and parts of the mix of the acoustic environment would be defined in the DAW, with the audio middleware providing some degree of flexibility over audio playback via adaptive audio. This process would lead to the issues mentioned in the paper's introduction.

In contrast, the paper's approach uses the DAW to produce a (detailed) collection of sound files corresponding to localized, single audio objects or point sources of an acoustic environment instead. Each object would be associated with an event in the audio middleware, and its playback and real-time effects are defined by rules, which are specified in the game engine. The game engine, thus, implements the structure of the acoustic environments and uses the audio middleware predominantly as a sample player rather than as a sequencer.

In this way, the paper combines an object-focused approach with multimodal rule-based systems and adaptive and 3D-binaural audio in 6-DoF. The governing system defines

191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206





**Figure 1.** Schematic overview of the proposed system. Data generated by the physics system, game states, or user input/output are mapped to a governing system that coordinates the interpretation and relation of this data to the multimodal representation of objects. The governing system is responsible for the sonic organization and the definition of the latter's interactive behavior.

an object's behavior on all its sensory, interactive, and spatial representations using rules, as conceptualized using example representations in Figure 1. In this way, object behaviors can be defined in many modalities and mapped to sounds by the designer. The governing system could be replicated for each object within a collection of objects, or a single system could control the entire collection. An acoustic environment can be formed from a single or several governing systems that affect many objects individually or simultaneously.

Each behavior representation in Figure 1 lists **example** data sources, processes, and an internal mapping system. The use of AI is omitted due to project scope but could be added in the future. The mapping system transforms received parameter data into internal data formats, such as clipping and scaling data, and associates external parameters with internal parameters. The figure's arrows indicate data exchange between the representations.

To illustrate the mapping between state changes and resulting object behaviors, here is an example. A timer initiates a state change, and the governing system then maps this state change to its internal rules. This mapping might result in an object moving in space, producing a sound, and gradually becoming transparent. Behaviors can also influence each other (via the governing system), such as an object's movement velocity could affect the playback rate of a sound file.

By experimentation and observation of the author's prototype room-scale VR experience, the authors established four key design rules that enable the rule-based system to generate (evolving) acoustic environments:

1. The system should transition between its states over time to allow the player to observe these transitions.
2. The system should affect or be replicated across several objects to cause complex inter-object interactions, aiming to cause subsequent state changes.
3. The system should proactively initiate subsequent state changes to avoid the system's premature termination / falling silent, depending on the scenario, narrative, and acoustic context.

4. Methods of virtual room acoustics, such as reverberation and simulation of (early) reflections, should be used wherever possible to give a sense of space. Multiple, diffuse-sounding Ambisonic field recordings without localizable sound sources can be used alternatively in outdoor environments.

Design rule 4 requires discussion. The taxonomy of acoustic environments presented in [55] differentiates between indoor and outdoor acoustic environments and lists possible generators of sound sources. One key difference between these environments is their acoustics. While reverberation provides the listener with aural cues indicating room dimensions and surface materials in indoor spaces, this option may not be suitable for outdoor environments [5,56]. For example, open spaces may not contain walls to reflect sound to the listener. While simulation methods of sparsely reflecting outdoor acoustics have been proposed [57], they may not have been implemented in 3D-binaural spatializers such as Google Resonance Audio [56].

Instead, (very) diffuse Ambisonic field recordings could be used, such as recordings of wind or very distant, constant traffic hum. They might assist in the evocation of a sense of space. If the sound sources in the field recording remain un-localizable ("diffuse") and respond to the player's head rotation, they may not contradict the player's motion in space. This impression could be further supported by blending dynamically between different diffuse Ambisonic recordings based on the listener's position. This approach aims to reduce contradiction to the player's spatial motion that Ambisonic beds could introduce.

#### 4. Implementation

The feasibility of this audio object-focused approach to the design of VR acoustic environments, and the spatial and interactive benefits that consequently emerge have been evaluated through the development of a prototype room-scale VR experience called *Planet Xerilia*, (soon) available on SideQuest [58]. In the prototype, a single player moves through eight virtual environments in 6-DoF, interacts with its virtual "sound installations", and uncovers a story told by audio and the environment.

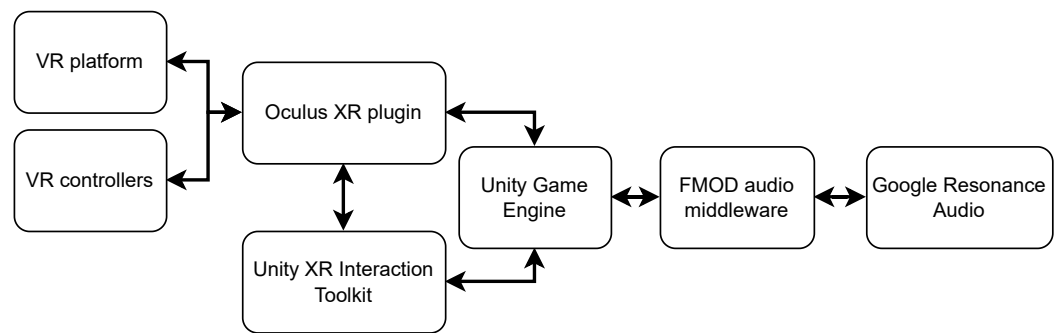
##### 4.1. Platform

The authors developed the prototype for the Meta Quest 2 using Unity [59] and additional packages, as shown in Figure 2. The Quest 2 features head-mounted displays, inside-out tracking, a Qualcomm Snapdragon XR2 SoC, and miniature loudspeakers embedded in the head-strap [3]. The authors primarily listened to the prototype's audio via closed and open headphones. Unity's audio engine was replaced with audio middleware FMOD to facilitate sound design and production [46]. FMOD ran within Unity as a plugin. Additionally, Google Resonance Audio [7] is used as the 3D-binaural audio spatializer and ran as a plugin in FMOD [60]. It was chosen for efficiency reasons [61] and ease of use. Unity's Oculus XR Plugin [62] is used to interface with the Quest 2's hardware. Unity's XR Interaction Toolkit [63] provides means of enabling player locomotion and interactivity. The proposed methodology is not specific to the author's development platform and may be possible to implement elsewhere using similar tools. The visuals are all based on Unity's standard primitives, i.e., rectangular cuboids and quads, and baked and real-time lighting.

##### 4.2. Player movement in space

To enable room-scale movement the authors implemented an avatar-based movement system [64]. This allows the player to physically move in the environment within the limits of their play space. The player's movement area is virtually extended using a teleportation system and joystick-controlled smooth locomotion [65], as well as virtually rotated via joystick-controlled stepped rotation ("snap turn"). The experience uses the joysticks of the Quest 2's handheld controllers. The player can ignore smooth locomotion and can use teleportation instead to move around the virtual environment's platforms. Strategies such as redirected walking techniques, as presented in [66], have not been implemented due to the project scope.





**Figure 2.** Schematic overview of the packages used in the prototype to interface with the VR hardware (Oculus XR plugin, Unity XR Interaction Toolkit), enable adaptive audio (FMOD) and 3D-binaural audio (Google Resonance Audio).

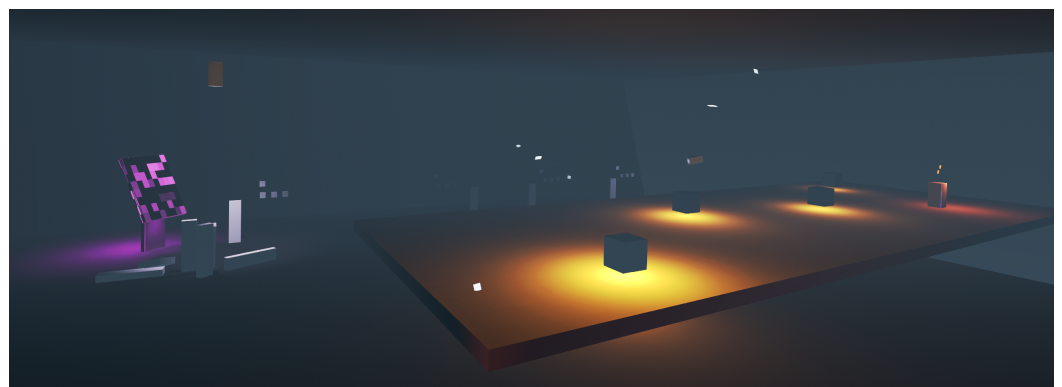
#### 4.3. Multimodal applications

Three **example** implementations of rule-based systems and their application are explored in *Planet Xerilia*. They are based on collisions, motion patterns, or lists. Screenshots available in the supplementary section show the system's audio-visual appearance and interactive potential.

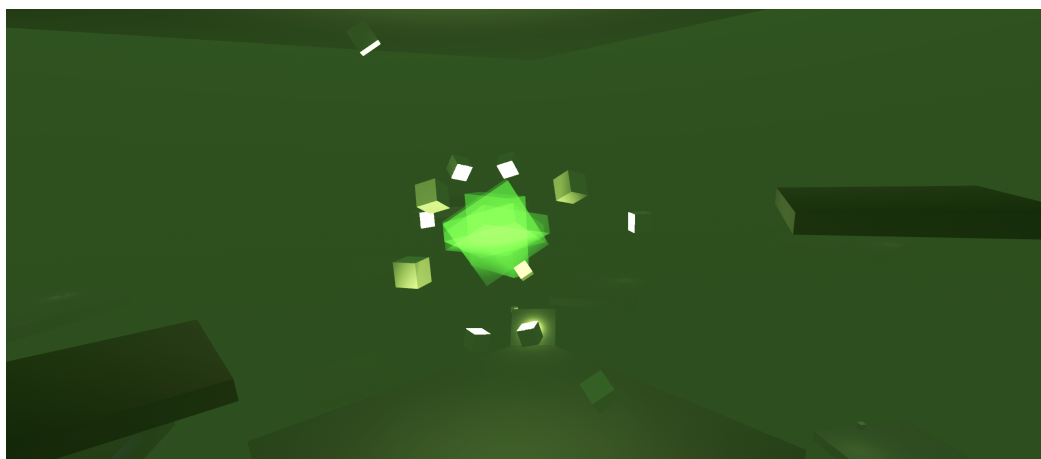
##### 4.3.1. Collision-based system

In the collision-based system, forces push objects to move and, eventually, collide with other objects, with the collision resulting in sound. For instance, *Planet Xerilia* uses this system to create an acoustic environment in which water drips down from a ceiling (Figure 3, Example 2). A timer causes the random instantiation of "water droplets" in time and space (Design Rules 2/3). The droplets, affected by gravitation, will fall, and collide with the floor or the player (Design Rule 1). Each collision triggers a sound, fades out the object's visual representation, and ultimately destroys the object. The collision's force (correlating with impact velocity) is mapped to a matching sound that implies a similar impact velocity. The droplets excite the environment's reverberation (Design Rule 4), aiming to evoke the impression of a cavernous, wet room. The timer's frequency of droplet instantiation can be set by the sound designer, providing control over the density of the resulting texture.

Alternative applications of this process push an existing object in a "random" direction with both the push and the collision creating a sound, leaving the object's lifetime unaffected (Example 2, [timecode 0:35](#)). Interactivity can be implemented, for example, via letting the player exert forces onto an object either by grab-and-throw actions or collisions between the player's virtual hands/head.



**Figure 3.** Screenshot showing one area in *Planet Xerilia*. The collision-based system is implemented in the "water" droplets, here visible as randomly rotated white, small squares, that fall from the area's ceiling. Once they collide with the floor they disappear and make a "splash" sound.



**Figure 4.** Screenshot of one of *Planet Xerilia's* areas. The matte green small cubes that hover around a glowing, bright green center implement the motion-pattern-based system. These cubes constantly emit sound, with their motion and sound affected by the physics system.

#### 4.3.2. Motion-pattern-based system

In the motion-pattern-based system, a group of objects continuously move within the defined space, always generating sound. An object's motion is induced by (constantly) applied forces (via the physics system) or direct translation in space. Each resulting motion pattern has its specific duration, movement speed, parameter settings for real-time processing effects, and sound file variation (Design Rule 1). Through the spatial layering of several of these objects (Design Rule 2) and careful tuning of their sound radiation range and behavior, the overall impression of a complex texture emerges as each object moves in and out of the player's proximal hearing range at different times.

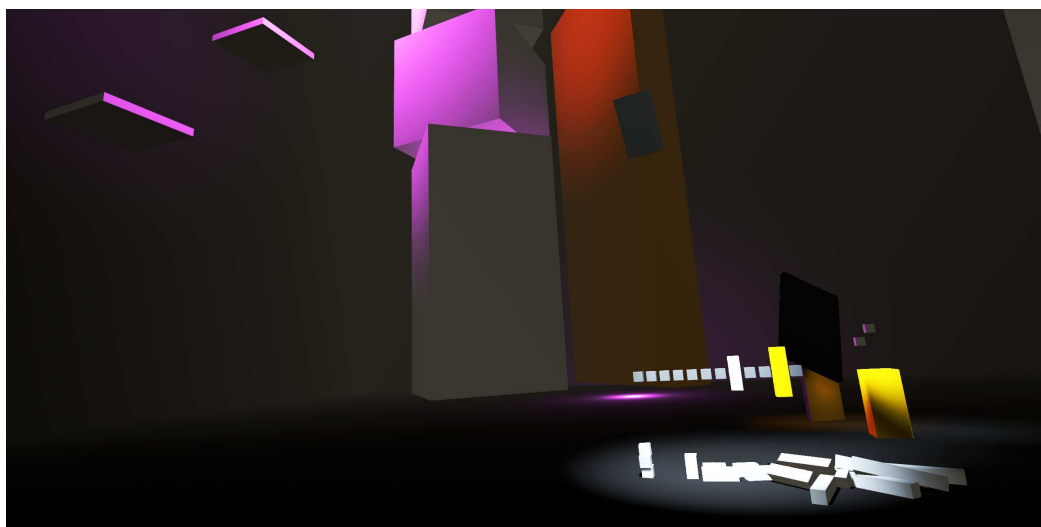
The physics engine can create complex motion patterns in this system and affords rudimentary player interaction. For example, in [Example 3](#) (Figure 4), several objects ("bees") orbit around a glowing center. A constant force pushes each object continuously and separately, with the object's motion constrained by an invisible hinge. A collision with the player's head, hands, or other objects can cause this hinge to break, initiating a "scream" of the object and throwing it into another orbit. This collision also adds additional force, causing the object to accelerate, resulting in subsequent collisions occurring more quickly. Additionally, the whole system slowly rotates around itself, which adds further potential for collisions. Combining these strategies creates an "unstable" scenario, resulting in complex, emergent motion patterns. The instability of the setup satisfies Design Rule 3, causing the emergence of an evolving system that the player can influence. This scenario can also be seen as an example where the features of different systems are combined.

As seen in [Section 4.3.1](#), the physics engine allows the mapping of an object's velocity to sound. In this instance, the authors exaggerated this mapping to make the object's change in velocity aurally apparent to the player and appear to be "faster-than-life".

#### 4.3.3. List-based system

In the lists-based system, a timer initiates a state change of an object from a predefined list of objects (Design Rule 3). The state change triggers the aural, motion, and visual behavior of an associated object. Interactivity can be implemented by giving the player control over aspects of the list, such as which objects are to be included. Alternatively, the player may indirectly / directly start or stop the timer. Collisions between the associated objects and the player can add the potential for interactivity, as seen in the discussion of the collision-based system.

*Planet Xerilia* uses the list-based system as a note-/chord-based step-sequencer that advances within a pattern, i.e., the list. [Figure 5](#) shows an example of such an implementation ([Example 4](#)). The list is represented as hovering, partially transparent cubes ("sockets")



**Figure 5.** This screenshot shows an example implementation of the list-based system. The list affects the rotation and playback of sounds emitted by large, grey/purple cubes, shown in the center of the screenshot. The list is player accessible for data entry via its representation as bright grey, small hovering cubes (bottom right, "sockets"). A yellow cuboid that overlaps with one of the list's sockets corresponds to the list's play position. The white cuboids on the floor represent the objects associated with the list. Their size corresponds to the vertical position of one of the large grey cubes.

in which the player can place larger cuboids. The latter cuboids correspond to vertically stacked large cubes. When the play-head object, here represented as a yellow cuboid, passes through a socket, the state of the socket's associated object changes, resulting in the rotation of the large cuboid and the playback of its sound source. This system's implementation results in a player-driven bass line. An alternative implementation combines two such step-sequencers, each associated with their sounds, objects, and trigger frequency, resulting in evolving chord progressions (Example 6, [timecode 1:05](#)).

The application of the lists-based system in Figure 5 can be seen as an approach to multi-source diegetic music in 6-DoF audio. The objects' position in space targeted by the list corresponds to the spatialization of the notes/sounds; the note sequence also corresponds to a sequence in space. This allows the player to approach each note. The sound files and the data implementation of the list need to ensure musicality, i.e., that the sounds and their representation in a data structure are musically meaningful. For this reason, the authors restricted player control to the order of the sequence's notes. The pitch of each note and the tempo of the sequence is controlled by the authors. This approach shares similarities with the approach used in *Spore*, as described in Section 2.3.

## 5. Evaluation/results

The evaluation focuses on the advantages of the (audio) object-focused approach to acoustic environment design in VR, whereas limitations/disadvantages will be discussed in Section 6. A comparison is made with alternative implementation methodologies, where applicable, to indicate the spatial and interactive benefits of the former.

### 5.1. Collision-based system

To demonstrate the spatial benefits and audio-visual synchronization, the authors compare the application of the collision-based system with an alternative approach based on the playback of a static Ambisonic recording. Both are built from one element ("water droplets") of one of the prototype's acoustic environments. The authors manually approximated the aural features of the Ambisonic version with the original object-based version in a DAW. Both audio representations share the same visual scene; the water droplet's visual representation in time and space is generated randomly in both cases.

The object-based version is created using the following steps. A dry, in-situ recording of water droplets is segmented into several small sound files; the resulting files are grouped into four levels representing different impact forces; impact forces are assigned to a parameter exposed to Unity and mapped to the impact collision velocity of the water droplet's visual representation; the visual representation of the droplet's collision with the floor or the player triggers the water droplet's sound; the position of the collision is matched to the spatialized sound's position; the player's distance to the spatialized sound's position affects its direct signal level and a high-pass filter's cutoff frequency re-moves the sound's low-frequency content. As a result, the water droplet will sound brighter, quieter, and more reverberated as its proximity to the player decreases.

The Ambisonic recording method approximates the object-based version via the following steps. First, the same source recording is divided into small segments; each segment is assigned to one of five positions around the virtual listener in the DAW at different distances; each position is spatialized using 1st-order Ambisonic encoding via dearVR pro [67]; the resulting Ambisonic recording is reverberated using an impulse response generated from within the environment's real-time room simulation; the recording is looped, placed in the center of the environment and starts playing at the beginning of the scene.

[Example 5](#) shows the spatial, interactive benefits of the audio object-focused approach. The screencast presents a fly-through through the environment of the Ambisonic version first, then followed by a fly-through of the object-focused version (visible at [timecode 0:37](#)). In the object-focused version, the collision sound of each water droplet matches its visual representation in space. Their distinct aural localization facilitates transitions across acoustic environments: when the player stands on the threshold of one environment, all the drop-lets sound distant, whereas in the middle of that environment, some droplets sound close, while others sound distant. Furthermore, integrating sound file playback in the collision system aligns the visuals with the audio in time and space (see also [Example 2](#)). The transient nature of the water droplets limits active player participation in both cases. However, the object-focused approach provides interactivity to the sound designer, making the acoustic environment malleable in real-time. For instance, the density of the water drop-lets can be easily adjusted in real-time by changing the generator's trigger frequency.

In contrast, while the Ambisonic version also creates a sense of space, it does not respond to the player's position dynamically. This means some water droplets sound always close, while others sound always distant. Furthermore, the sounds cannot be aligned in time and space to the water droplet's visual representation due to the randomness of their generation (unless their pattern of appearance would be predetermined). A change in density of water droplets generation would require the sound designer to re-design, re-render, and reimport the Ambisonic recording.

## 5.2. Motion-pattern-based system

[Example 3](#) shows the spatial and interactive benefits of the motion-pattern-based system when used in combination with a collision-based system, as described in [Section 4.3.2](#). The screencast first presets the system from a distance, then shows the player stepping close to the system and hitting one of its associated objects which results in the system spinning chaotically. The motion-pattern-based system allows the player to spatially experience their approach and withdrawal to/from each object individually via 3D-binaural audio. The embedded collision system acknowledges the player's existence in the virtual environment via intentional or unintentional collisions caused by the player. The player can actively influence the configuration of the system by hitting some of its objects.

Due to the unpredictability of player interaction and behavior of the system, the authors do not compare this application with another non-object-based solution. Alternatively, the movement of the objects could be defined and fixed in time through a timeline-based animation, which could be set to sound in a DAW. This sound could then be rendered as an Ambisonic sound file and played back synchronously with the animation. However, this

approach would exclude player interaction. The complexity of the objects' motions would also be challenging to manually match in a DAW.

### 5.3. List-based system

The authors compare two spatial implementations of an environment's object-based music to demonstrate the differences in spatial response. The music contains three elements: two drone-based sounds are accompanied by bass-drum-like collision sounds. In one version, the 3D-binaural audio spatializer embeds the drone-like sounds in the same space as the visual objects, with the music becoming diegetic. In the second version, randomized, stereophonic amplitude-based panning locates these sounds outside the game world, with the music becoming non-diegetic. In both cases, the sound of the collisions between the visual representations of the objects is kept in 3D-binaural audio. The spatialization strategy implemented in the non-diegetic version shares similarity with the approach taken by [43] (see Section 2.4).

While the music evokes a similar atmosphere in both cases, its spatial impression changes, as heard in Example 6. The screencast first presents a fly-through through the environment using the non-diegetic version, followed by a version using the diegetic version. The diegetic approach makes the music localized and associates each sound with its visual representation. Due to the spatial correlation between the visual and aural representation, the collision sounds appear to be part of the music. The music becomes responsive to the environment's room reverberation and the player can approach the sounds contained in the music.

This diegetic placement contrasts with the non-diegetic version. There is a spatial disconnect between the collision sounds, the drone-like sounds, and the acoustic environment's remaining sounds in the non-diegetic version. Furthermore, due to this spatial disconnect, the collision sounds may not appear to be part of the music but rather just the sounds of the colliding objects.

The diegetic or non-diegetic versions of the music may both be suitable; however, their narrative meaning may change [68]. Ultimately, whether the effect of the music's spatial dislocation appears problematic or appropriate would depend on the game's contexts, narrative goals, or overall approach, as suggested by [11].

## 6. Discussion

The application of the object-focused approach presented in this work has revealed implementation complexities that require consideration. These complexities stem from platform constraints and the multimodal nature of the object-focused approach.

### 6.1. Multimodal behavior application constraints

While implementation complexity and scope management strategies are typical for game development [69,70], an object-focused approach to acoustic environments adds to this complexity. For example, each sound associated with an object's audio behavior needs to be recorded, edited, mapped to parameters, linked to a playback and behavior logic, balanced in signal level to other sounds in the same context, tested for functionality, and optimized for performance. If the sound has to be replaced at a later stage the whole process needs to be repeated. Variations of sounds are required to reduce the perceived repetition of sounds [54, ch. 3]. Velocity-sensitive sounds increase the sound pool further, potentially amounting to about 35+ sound files per object (see Section 6.4). For context, approximately 1572 sound files are used in total *Planet Xerilia* across its eight environments. Equally, the object's other modal representations, e.g. visualizations, require similar steps of implementation and testing relevant to their medium. For this reason, VR sound designers/programmers need to (re-)consider the depth of detail and interactivity required and balance these against other relevant project constraints.

Furthermore, the implementation of object behaviors on all their sensorial representations may not be necessary or applicable in all contexts. By way of example, sounds



representing aural hallucinations or ghosts may not require any visual representation due to reasons of narration and only the audio behavior would need to be implemented. Such a reduced-modal approach could limit opportunities for active player participation, as the player may not be able to associate and understand their influence over an object's state. However, restricting active player participation may facilitate production workflows as it removes the need for sounds confirming user action. Ultimately, the sound designer would have to balance the constraints of the project with the desired level of object-based player participation. However, if the sound designer ensures an object-focused approach in the audio representations, these audio objects would still respond to passive player participation.

### 6.2. VR platform constraints

The multimodal nature of the object-focused approach proposed in this work may place a high strain on the processing power available on the target VR platform, especially if untethered. Moving objects may require real-time calculations provided by the physics engine, rendering, or lighting system. Optimization strategies are suggested to offset such compound computational overheads, such as reducing the number of moving objects to gain the ability to pre-render their lighting and shadows [71]. In this way, the VR platform acts to constrain the complexity and usage of a multimodal object-focused approach.

Due to the processing limitations of the target platform in this work, the authors employed a compromise between a multimodal and reduced-modal approach in the creation of the acoustic environments (Example 7), with the decision between each depending on the object's narrative focus: objects that are critical for the experience's narrative feature multimodal behaviors, while others, used to augment the environment, feature reduced-modal behaviors. This approach facilitated production and reduced processing demands.

### 6.3. Implications of general-purpose programming language

Converting acoustic environments into object-focused, rule-based systems may require extensive programming knowledge beyond that of some sound designers or composers. Implementing behaviors may require custom code using general-purpose programming paradigms, such as object-oriented programming (OOP) or visual scripting, which is a methodology required by the game engine [72]. As these concepts are not intrinsically musical, the implementation of musical ideas requires translation and abstraction [73]. The aural result is then limited to the extent the sound designer can translate aural concepts to programming concepts used in a game engine, and available time. In this way, the game engine may constrain the musicality or complexity of an object-focused approach.

In the prototype discussed in this paper, custom code was necessary, for example, to implement the list-based system used in Section 4.3.3. This code included the description of a data structure that represents sounds and their configuration in time, as well as methods to describe their triggering, evolution in time, and visual behavior. Additionally, the code also needed to consider active player participation, such as the player affecting the contents or configuration of the data structure and the aesthetic/narrative aim.

### 6.4. Implementation complexities in collision-based systems

The complexity of implementing a collision-based system can increase as the level of desired realism increases. For example, the increase in sound level corresponding to an increase in input forces/collision velocities depends on the materials involved [74]. This finding requires a force-to-sound mapping on a per-material basis. Furthermore, higher impact forces can cause an object to crack, splinter, or break. Including these kinds of complexities necessitates creating, testing, and implementing many sound files. In the example of the water droplets from Section 4.3.1, the authors associated a total of 35 sound files to the four ranges of impact velocities. In another example, the collision sound is based on four components, including a synthesized base impact accompanied by different intensities of sand falling and stones rolling, resulting in 37 associated sound files.



Furthermore, the designer must ensure that the game engine detects collisions according to its multiple implementation contexts, requiring testing and adaptation of the underlying model or code design [75].

### 6.5. Conversion of Ambisonic field recordings

Ambisonic field recordings used as acoustic backdrops in outdoor environments may require conversion into an object-based representation to comply with the object-focused approach and to facilitate acoustic coherence. For example, the combination of bird-like sounds from different Ambisonic and non-Ambisonic field recordings found in a sound library may cause narrative/aural contradictions. Birds from one recording may not be compatible with the environment suggested by another recording through a mismatch of bird species or reverberation cues, e.g., dense vs sparse forest. Such mismatch could be avoided if the sounds captured in an Ambisonic field recording could be extracted and thus converted into an object-based representation.

Such conversion may be possible in cases where the recording features spectral or temporal separation between foreground and background sounds and the sounds of interest are in the foreground. For example, insects in an Ambisonic recording could be isolated from atmospheric background sounds, e.g., very distant traffic hum, via bandpass or FFT filters and then decoded to mono sound files. Each insect call could be separated into a sound file, spatialized using object-based rendering, and the frequency of the insects' utterances recreated using a rule-based system. More advanced machine learning approaches for sound isolation could be used alternatively [76]. The remainder of the Ambisonic recording could be recycled as a diffuse ambient Ambisonic bed if all identifiable gestural elements have been removed. The authors used this process in all outdoor environments in *Planet Xerilia* with [Example 7](#) demonstrating the result.

## 7. Conclusion

This paper has presented the design, application, and discussion of a multimodal, rule-governed, audio object-focused methodology for creating acoustic environments in room-scale VR. This methodology utilized adaptive, 3D-binaural audio in 6-DoF, rulesets, and a game engine to define the behavior of objects in space and time across different sensual representations (modalities). Its feasibility was demonstrated in a prototype room-scale VR experience and evaluated using an analysis of screencasts of the prototype.

The paper demonstrated the benefits and limitations of such an audio object-focused approach, allowing the injection of new behaviors, such as the audio's response to player movement in 6-DoF, active player participation driven by the physics system and custom code, and synchronized multimodal processes. The paper also highlighted the utility of a game engine's physics system in creating complex movements. However, platform or project constraints may restrict multimodal applications. These constraints may be addressed in part by concentrating on audio behaviors and 3D-binaural audio in 6-DoF while limiting the behavior of the remaining modalities by making objects immobile, non-interactive, or invisible. This approach also reduces the implementation complexities of audio behaviors as fewer sound files need creation, implementation, and testing. Ultimately, the designers must balance opportunities for audio-responsive player interaction against the larger framework of the game and its time and budget constraints.

Future work will investigate the effect of the multimodal, object-focused approach in a directed user experience study, as well as the potential application of AI in the audio production process.

**Supplementary Materials:** Supporting video examples are available at doi: <https://doi.org/10.5281/zenodo.6616582>.

**Author Contributions:** Conceptualization, Constantin Popp; Funding acquisition, Damian Murphy; Investigation, Constantin Popp; Project administration, Damian Murphy; Resources, Damian Murphy;

Software, Constantin Popp; Supervision, Damian Murphy; Writing – original draft, Constantin Popp; Writing – review editing, Constantin Popp and Damian Murphy.

**Funding:** This research was funded by XR Stories as part of the AHRC Creative Industries Clusters Programme, grant number AH/S002839/1.

**Data Availability Statement:** Not applicable

**Acknowledgments:** Firelight Technologies Pty Ltd and Unity Technologies kindly provided non-commercial licenses for research purposes as part of this research project.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

- Schjerlund, J.; Hansen, M.R.P.; Jensen, J.G. Design Principles for Room-Scale Virtual Reality: A Design Experiment in Three Dimensions. In Proceedings of the Designing for a Digital and Globalized World. Springer International Publishing, 2018, pp. 3–17. [https://doi.org/10.1007/978-3-319-91800-6\\_1](https://doi.org/10.1007/978-3-319-91800-6_1).
- Aylett, R.; Louchart, S. Towards a narrative theory of virtual reality. *Virtual Real.* **2003**, *7*, 2–9. <https://doi.org/10.1007/s10055-003-0114-9>.
- Facebook Technologies, LLC.. Oculus Device Specifications. <https://developer.oculus.com/resources/oculus-device-specs/>, 2021. Accessed: 2021-11-29.
- Schütze, S.; Irwin-Schütze, A. *New Realities in Audio: A Practical Guide for VR, AR, MR and 360 Video*, 1st edition ed.; CRC Press: Boca Raton, 2018.
- Google LLC. Game Engine Integration. <https://resonance-audio.github.io/resonance-audio/develop/fmod/game-engine-integration>, 2018. Accessed: 2021-12-29.
- Méndez, D.R.; Armstrong, C.; Stubbs, J.; Stiles, M.; Kearney, G. Practical Recording Techniques for Music Production with Six-Degrees of Freedom Virtual Reality. *New York* **2018**, p. 5.
- Gorzal, M.; Allen, A.; Kelly, I.; Kammerl, J.; Gungormusler, A.; Yeh, H.; Boland, F. Efficient Encoding and Decoding of Binaural Sound with Resonance Audio. In Proceedings of the 2019 AES International Conference on Immersive and Interactive Audio. Audio Engineering Society, 2019.
- Popp, C.; Murphy, D. Video Examples from: Creating Object-focused Acoustic Environments for Room-Scale Virtual Reality, 2022. <https://doi.org/10.5281/zenodo.6616582>.
- Tender Claws. Virtual Virtual Reality 2. <https://www.oculus.com/experiences/quest/2731491443600205>, 2022. Accessed: 2022-6-26.
- 99th V R. Virtual Virtual Reality 2 | 1st Hour Gameplay | 60FPS - No Commentary. <https://www.youtube.com/watch?v=V2Lcq7PqB2k>, 2022. Accessed: 2022-6-26.
- Phillips, W. Composing video game music for Virtual Reality: Diegetic versus Non-diegetic. <https://www.gamedeveloper.com/audio/composing-video-game-music-for-virtual-reality-diegetic-versus-non-diegetic>, 2018. Accessed: 2022-3-21.
- Meta. Sound Design for VR. <https://developer.oculus.com/resources/audio-intro-sounddesign/>, 2022. Accessed: 2022-6-27.
- Smalley, D. Spectromorphology: explaining sound-shapes. *Organised Sound* **1997**, *2*, 107–126. <https://doi.org/10.1017/S1355771897009059>.
- Song, X.; Lv, X.; Yu, D.; Wu, Q. Spatial-temporal change analysis of plant soundscapes and their design methods. *Urban For. Urban Greening* **2018**, *29*, 96–105. <https://doi.org/10.1016/j.ufug.2017.11.002>.
- Chattopadhyay, B. Sonic Menageries: Composing the sound of place. *Organised Sound* **2012**, *17*, 223–229. <https://doi.org/10.1017/S1355771811000422>.
- Turchet, L.; Hamilton, R.; Çamci, A. Music in Extended Realities. *IEEE Access* **2021**, *9*, 15810–15832. <https://doi.org/10.1109/ACCESS.2021.3052931>.
- Atherton, J.; Wang, G. Doing vs. Being: A philosophy of design for artful VR. *Journal of New Music Research* **2020**, *49*, 35–59. <https://doi.org/10.1080/09298215.2019.1705862>.
- Valve Corporation. Half-Life: Alyx. <https://www.half-life.com/en/alyx/>, 2020. Accessed: 2022-1-21.
- Jain, D.; Junuzovic, S.; Ofek, E.; Sinclair, M.; Porter, J.; Yoon, C.; Machanavajhala, S.; Ringel Morris, M. A Taxonomy of Sounds in Virtual Reality. In Proceedings of the Designing Interactive Systems Conference 2021; Association for Computing Machinery: New York, NY, USA, 2021; DIS '21, pp. 160–170. <https://doi.org/10.1145/3461778.3462106>.
- Zylia Sp. z o.o.. ZYLIA Concert Hall on Oculus Quest. <https://www.oculus.com/experiences/quest/3901710823174931/>, 2021. Accessed: 2021-11-30.
- Ciotucha, T.; Rumiński, A.; Żernicki, T.; Mróz, B. Evaluation of Six Degrees of Freedom 3D Audio Orchestra Recording and Playback using multi-point Ambisonic interpolation. *Journal of the Audio Engineering Society* **2021**, p. 7.
- Settel, Z.; Downs, G. Building Navigable Listening Experiences Based On Spatial Soundfield Capture: The Case Of The Orchestre Symphonique De Montréal Playing Beethoven's Symphony No. 6. In Proceedings of the 18th Sound and Music Computing Conference (SMC), 2021, p. 8.

23. Quackenbush, S.R.; Herre, J. MPEG Standards for Compressed Representation of Immersive Audio. *Proc. IEEE* **2021**, *109*, 1578–1589. <https://doi.org/10.1109/JPROC.2021.3075390>. 632
24. Whitmore, G. Design With Music In Mind: A Guide to Adaptive Audio for Game Designers. <https://www.gamedeveloper.com/audio/design-with-music-in-mind-a-guide-to-adaptive-audio-for-game-designers>, 2003. Accessed: 2022-5-9. 633
25. Zdanowicz, G.; Bambrick, S. *The Game Audio Strategy Guide: A Practical Course*; Routledge, 2019. 634
26. Byrns, A.; Ben Abdesslem, H.; Cuesta, M.; Bruneau, M.A.; Belleville, S.; Frasson, C. Adaptive Music Therapy for Alzheimer’s Disease Using Virtual Reality. In Proceedings of the Intelligent Tutoring Systems. Springer International Publishing, 2020, pp. 214–219. [https://doi.org/10.1007/978-3-030-49663-0\\_25](https://doi.org/10.1007/978-3-030-49663-0_25). 635
27. Vizm. The Climb 2 VR FULL WALKTHROUGH [NO COMMENTARY] 4K 60FPS. <https://www.youtube.com/watch?v=n8Dr4HIQu8>, 2021. Accessed: 2022-6-26. 636
28. Crytek GmbH. The Climb. <https://www.theclimbgame.com/>, 2021. Accessed: 2022-6-23. 637
29. inXile entertainment Inc.. The Mage’s Tale on Oculus Rift. [https://www.oculus.com/experiences/rift/1018772231550220/?locale=en\\_GB](https://www.oculus.com/experiences/rift/1018772231550220/?locale=en_GB), 2017. Accessed: 2022-6-26. 638
30. Scioneaux, III, E. VR Sound Design for Touch Controllers. <https://designingsound.org/2017/09/14/vr-sound-design-for-touch-controllers/>, 2017. Accessed: 2022-3-25. 639
31. Horswill, I.D. Lightweight Procedural Animation With Believable Physical Interactions. *IEEE Trans. Comput. Intell. AI Games* **2009**, *1*, 39–49. <https://doi.org/10.1109/TCIAIG.2009.2019631>. 640
32. Shaker, N.; Togelius, J.; Nelson, M.J. *Procedural Content Generation in Games*; Springer, Cham, 2016. <https://doi.org/10.1007/978-3-319-42716-4>. 641
33. Plut, C.; Pasquier, P. Generative music in video games: State of the art, challenges, and prospects. *Entertain. Comput.* **2020**, *33*, 100337. <https://doi.org/10.1016/j.entcom.2019.100337>. 642
34. Salamon, J.; MacConnell, D.; Cartwright, M.; Li, P.; Bello, J.P. Scaper: A library for soundscape synthesis and augmentation. In Proceedings of the 2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA), 2017, pp. 344–348. <https://doi.org/10.1109/WASPAA.2017.8170052>. 643
35. Farnell, A. An introduction to procedural audio and its application in computer games. *Audio mostly conference 2007*. 644
36. De Prisco, R.; Malandrino, D.; Zaccagnino, G.; Zaccagnino, R. An Evolutionary Composer for Real-Time Background Music. In Proceedings of the Evolutionary and Biologically Inspired Music, Sound, Art and Design. Springer International Publishing, 2016, pp. 135–151. [https://doi.org/10.1007/978-3-319-31008-4\\_10](https://doi.org/10.1007/978-3-319-31008-4_10). 645
37. Washburn, M.; Khosmood, F. Dynamic Procedural Music Generation from NPC Attributes. In Proceedings of the International Conference on the Foundations of Digital Games; Association for Computing Machinery: New York, NY, USA, 2020; Number Article 15 in FDG ’20, pp. 1–4. <https://doi.org/10.1145/3402942.3409785>. 646
38. Electronic Arts Inc. Spore: What is Spore. <https://www.spore.com/what/spore>, 2009. Accessed: 2022-1-21. 647
39. Collins, N. Musical Form and Algorithmic Composition. *Contemporary Music Review* **2009**, *28*, 103–114. <https://doi.org/10.1080/07494460802664064>. 648
40. Sharples. Spore - No Commentary - Part 1. <https://www.youtube.com/watch?v=RiBmqTA85I>, 2018. Accessed: 2022-6-26. 649
41. Radvansky, G.A.; Krawietz, S.A.; Tamplin, A.K. Walking through doorways causes forgetting: Further explorations. *Q. J. Exp. Psychol.* **2011**, *64*, 1632–1645. <https://doi.org/10.1080/17470218.2011.571267>. 650
42. Phillips, W. Game Music and Mood Attenuation: How Game Composers Can Enhance Virtual Presence (Pt. 4). <https://www.gamedeveloper.com/audio/game-music-and-mood-attenuation-how-game-composers-can-enhance-virtual-presence-pt-4->, 2019. Accessed: 2022-6-27. 651
43. Phillips, W. Composing video game music for Virtual Reality: 3D versus 2D. <https://www.gamedeveloper.com/audio/composing-video-game-music-for-virtual-reality-3d-versus-2d>, 2018. Accessed: 2022-3-21. 652
44. Beat Games. Beat Saber. <https://www.oculus.com/experiences/quest/2448060205267927>, 2019. Accessed: 2022-6-27. 653
45. jonacreepy. Beat Saber all Songs on EXPERT mode | no commentary. <https://www.youtube.com/watch?v=ZcBcCxBRHXw>, 2018. Accessed: 2022-6-27. 654
46. Firelight Technologies Pty Ltd. FMOD Studio User Manual 2.02. <https://www.fmod.com/resources/documentation-studio?version=2.02>, 2021. Accessed: 2022-1-24. 655
47. Owlchemy Labs. Vacation Simulator on Oculus Quest. <https://www.oculus.com/experiences/quest/2393300320759737>, 2019. Accessed: 2022-6-29. 656
48. Acampora, G.; Loia, V.; Vitiello, A. Distributing emotional services in Ambient Intelligence through cognitive agents. *Service Oriented Computing and Applications* **2011**, *5*, 17–35. <https://doi.org/10.1007/s11761-011-0078-7>. 657
49. Wen, X. Using deep learning approach and IoT architecture to build the intelligent music recommendation system. *Soft Computing* **2021**, *25*, 3087–3096. <https://doi.org/10.1007/s00500-020-05364-y>. 658
50. De Prisco, R.; Guarino, A.; Lettieri, N.; Malandrino, D.; Zaccagnino, R. Providing music service in Ambient Intelligence: experiments with gym users. *Expert Syst. Appl.* **2021**, *177*, 114951. <https://doi.org/10.1016/j.eswa.2021.114951>. 659
51. Monstars Inc. and Resonair. Rez Infinite on Oculus Quest. <https://www.oculus.com/experiences/quest/2610547289060480>, 2020. Accessed: 2022-6-29. 660
52. For Fun Labs. Eleven Table Tennis on Oculus Quest. <https://www.oculus.com/experiences/quest/1995434190525828>, 2020. Accessed: 2022-6-29. 661

53. Forward Game Studios. Please, Don't Touch Anything on Oculus Quest. <https://www.oculus.com/experiences/quest/2706567592751319>, 2019. Accessed: 2022-6-29. 691
54. Robinson, C. *Game Audio with FMOD and Unity*; Routledge, 2019. 692
55. Brown, A.L.; Kang, J.; Gjestland, T. Towards standardization in soundscape preference assessment. *Appl. Acoust.* **2011**, *72*, 387–392. <https://doi.org/10.1016/j.apacoust.2011.01.001>. 693
56. Kelly, L.; Gorzel, M.; Gungörmüsler, A. Efficient externalized audio reverberation with smooth transitioning. *Technical Disclosure Commons* **2017**. <https://doi.org/10.13140/RG.2.2.20335.33449>. 694
57. Stevens, F.; Murphy, D.T.; Savioja, L.; Välimäki, V. Modeling Sparsely Reflecting Outdoor Acoustic Scenes Using the Waveguide Web. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **2017**, *25*, 1566–1578. <https://doi.org/10.1109/TASLP.2017.2699424>. 695
58. SideQuestVR Ltd.. SideQuest: Oculus Quest Games & Apps including AppLab Games ( Oculus App Lab ). <https://sidequestvr.com/>. Accessed: 2022-5-31. 696
59. Unity Technologies. Unity - Manual: Unity User Manual 2020.3 (LTS). <https://docs.unity3d.com/Manual/>, 2021. Accessed: 2021-1-12. 697
60. Google LLC. FMOD. <https://resonance-audio.github.io/resonance-audio/develop/fmod/getting-started.html>, 2018. Accessed: 2021-12-7. 698
61. Gould, R. Let's Test: 3D Audio Spatialization Plugins. <https://designingsound.org/2018/03/29/lets-test-3d-audio-spatialization-plugins/>, 2018. Accessed: 2021-1-12. 699
62. Unity Technologies. About the Oculus XR Plugin. <https://docs.unity3d.com/Packages/com.unity.xr.oculus@3.0/manual/index.html>, 2022. Accessed: 2022-5-31. 700
63. Unity Technologies. XR Interaction Toolkit. <https://docs.unity3d.com/Packages/com.unity.xr.interaction.toolkit@2.0/manual/index.html>, 2021. Accessed: 2021-12-28. 701
64. ap Cenydd, L.; Headleand, C.J. Movement Modalities in Virtual Reality: A Case Study from Ocean Rift Examining the Best Practices in Accessibility, Comfort, and Immersion. *IEEE Consumer Electronics Magazine* **2019**, *8*, 30–35. <https://doi.org/10.1109/MCE.2018.2867971>. 702
65. Weißker, T.; Kunert, A.; Fröhlich, B.; Kulik, A. Spatial Updating and Simulator Sickness During Steering and Jumping in Immersive Virtual Environments. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR). IEEE, 2018. <https://doi.org/10.1109/VR.2018.8446620>. 703
66. Langbehn, E.; Steinicke, F. Space walk: a combination of subtle redirected walking techniques integrated with gameplay and narration. In Proceedings of the ACM SIGGRAPH 2019 Emerging Technologies; Association for Computing Machinery: New York, NY, USA, 2019; Number Article 24 in SIGGRAPH '19, pp. 1–2. <https://doi.org/10.1145/3305367.3327976>. 704
67. Dear Reality. dearVR PRO. <https://www.dear-reality.com/products/dearvr-pro>, 2022. Accessed: 2022-5-16. 705
68. Çamci, A. Exploring the Effects of Diegetic and Non-diegetic Audiovisual Cues on Decision-making in Virtual Reality. In Proceedings of the Proceedings of the 16th Sound & Music Computing Conference; Isabel Barbancho, Lorenzo J. Tardón, Alberto Peinado, Ana M. Barbancho., Ed., 2019, pp. 195–201. 706
69. Elliott, B. Anything is possible: Managing feature creep in an innovation rich environment. In Proceedings of the 2007 IEEE International Engineering Management Conference, 2007, pp. 304–307. <https://doi.org/10.1109/IEMC.2007.5235049>. 707
70. Petrillo, F.; Pimenta, M.; Trindade, F.; Dietrich, C. What went wrong? A survey of problems in game development. *Comput. Entertain.* **2009**, *7*, 1–22. <https://doi.org/10.1145/1486508.1486521>. 708
71. Arm Limited. Arm Guide for Unity Developers: Real-Time 3D Art Best Practices: Lighting. <https://developer.arm.com/documentation/102109/latest>, 2020. Accessed: 2022-5-11. 709
72. Unity Technologies. Coding in C# in Unity for beginners. <https://unity.com/how-to/learning-c-sharp-unity-beginners>, 2021. Accessed: 2021-12-3. 710
73. Dannenberg, R.B. Languages for Computer Music. *Frontiers in Digital Humanities* **2018**, *5*. <https://doi.org/10.3389/fdigh.2018.00026>. 711
74. Lietzén, J.; Miettinen, J.; Kylliäinen, M.; Pajunen, S. Impact force excitation generated by an ISO tapping machine on wooden floors. *Appl. Acoust.* **2021**, *175*, 107821. <https://doi.org/10.1016/j.apacoust.2020.107821>. 712
75. Shekhar, G. Detect collision in Unity 3D. <http://gyanendushekhar.com/2020/04/27/detect-collision-in-unity-3d/>, 2020. Accessed: 2022-1-21. 713
76. Li, H.; Chen, K.; Wang, L.; Liu, J.; Wan, B.; Zhou, B. Sound Source Separation Mechanisms of Different Deep Networks Explained from the Perspective of Auditory Perception. *NATO Adv. Sci. Inst. Ser. E Appl. Sci.* **2022**, *12*, 832. <https://doi.org/10.3390/app12020832>. 714