

Multi-Version Song Exploration in Web VR Using A-Frame and Tone.js

Tom Collins
Frost School of Music
University of Miami
tomthecollins@gmail.com

Ashay Dave
Frost School of Music
University of Miami
ashaydave@gmail.com

Raina Murnak
Frost School of Music
University of Miami
r.murnak@umiami.edu

ABSTRACT

Music audio concepts and design for VR are relatively under-explored compared to visual analogs. This paper addresses the design and evaluation of a dynamic experience of music in Web VR called VRTGO, in which a user explores three alternative versions of the same song called “VTGO (Vertigo)”. Each song version is represented visually by a spiral of blocks emanating from separately located vortices on the ground. The closer the avatar is to one of the three-block spirals, the closer the audio experience is to one of the three intended versions of the song, in terms of tempo and instrumental mix. We describe the use of granular synthesis to achieve smooth tempo transitions, and present findings from a user study exploring engagement and enjoyment. The study finds that participants spend significantly longer in the dynamic version of the experience and report significantly greater engagement, though no significant differences in enjoyment are found between the dynamic and static versions. While users’ stated benefits of VR music experiences include enhanced creativity and emotional engagement, drawbacks such as accessibility and over-immersion risks are identified. The technical contributions and results of this work could be of interest to music artists and producers looking to explore the possibilities associated with rendering their creations in VR, and how these experiences enable novel ways to engage and interact with their audience.

1. INTRODUCTION

Well-known artists such as Travis Scott, Ariana Grande, and Elton John have begun embracing the concept of live or recorded concerts rendered in virtual reality (VR), or, more generally, extended reality (XR) [9]. The musical content in these experiences tends to be largely predetermined: while it may change in volume or perceived source location as an avatar moves around a virtual space, fundamental musical components such as tempo, melody, lyrics, or constituent accompanying tracks do not tend to be subject to change, either by the artist or due to actions of audience members.

One reason for this status quo is that it mirrors the general structure of the recording industry – a “one-way street”

where artists make recordings, record companies distribute them, and listeners consume. Music content is more-or-less fixed by the artist, and typically consumers cannot interact with or modify it. A second reason, which we will explore further in the review below, is that existing audio frameworks for games and XR tend to be quite limited in the dynamic or on-the-fly changes to/experiences of music that they afford.

This “one-way street” of musicians making, companies distributing, and fans consuming music is a – perhaps *the* – defining characteristic of the recording industry, but it emerged and remains for reasons of economic expediency, rather than out of a love of music and the benefits it confers, and it contrasts with the way music both secular and sacred was made and heard previously. It also contrasts with the concept of *prosumption* – a *prosumer* being someone who is both a producer and consumer – which is a term coined by Toffler [20] but goes back as far as Karl Marx [15, 12].

What might transpire if we were to “drive the other way up the one-way street” – if an audience member with novice or expert musical skills were able to access and manipulate elements of a musical experience such as the tempo and constituent tracks, rather than them being mixed down and so hidden away? Based on the premise that this may be of interest to music artists, producers, and their fans, this paper develops the conceptual foundations for such experiences, as well as an A-Frame and Tone.js implementation via which they are realized.¹ It is less about experiencing a “live concert” in VR, and more about dynamic experiences of alternative song versions, which may achieve different kinds of musical immersion and engagement, and/or increase dwell time on music releases. Research in adaptive audio and music for interactive media seeks to establish audio as a dynamic and integral part of the interactive experience, enhancing player engagement and emotional resonance. This exploration aligns with the industry’s realization that true immersion extends beyond the visual domain and encompasses a multisensory engagement that includes audio.

Our implementation does not contain visual avatars, and generally it is visually reductive, enabling focus on and exploration of the musical possibilities. We include a UI/UX evaluation study as part of the paper, placing the user in a web VR experience called VRTGO.² The user study was designed to empirically evaluate the proposed dynamic mu-



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

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

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

¹<https://aframe.io/>; <https://tonejs.github.io/>

²<https://vrtgo.onrender.com>. For the source code, see script.js at <https://github.com/tomthecollins/vrtgo>. For the released version of the song, <https://vtgo.onrender.com>.

music experience, focusing on two primary objectives: (1) to assess user engagement and enjoyment when experiencing music through an interactive, spatially dynamic system compared to a static musical presentation, and (2) to explore users’ perceptions of the potential of dynamic musical experiences in VR. We employed a single-factor within-subjects experimental design with two conditions: a dynamic musical experience where participants’ spatial movement directly influences musical tempo and mix, and a static musical experience with a fixed musical rendition. By collecting both quantitative data (dwell times, enjoyment ratings, engagement ratings) and qualitative feedback, we aimed to understand how spatial interactivity might transform musical listening experiences and identify potential benefits and limitations of such an approach. As such, we claim the approach and its results contain some promise. The remainder of the paper is structured as follows: we review existing systems for delivering music in XR, describe the design concepts behind and implementation of VRTGO, present the findings from the user study to evaluate the system, and conclude with discussions of the contribution, an artist’s perspective, limitations, and future work.

2. REVIEW OF EXISTING SYSTEMS FOR MUSIC IN XR

At present, in terms of development environments, the three main approaches to building a musical experience in XR are to use Unity, Unreal Engine, or a web-based framework. In terms of creating dynamic audio experiences, Stevens and Raybould [18] identify three main, non-mutually exclusive approaches: **transitional**, **parallel**, and **ornamental**.

The term **transitional** refers to how one passage of pre-composed music can be cross-faded convincingly with a second passage, often to accompany a game character moving from one (part of a) level to another. Another term for this is **horizontal** change in the music – considering how to manipulate and deliver audio resources at times a and $b > a$.

The term **parallel** refers to how two or more audio resources may be playing back in sync, and changes in the game make it appropriate to dynamically fade up or down these various layers of sound.

The term **ornamental** refers to having one main passage of music playing, while other shorter musical sounds – sometimes called **stingers** – play on top of the main passage, in response to certain in-game events. While ideally the timing of each stinger will be in sync with the main passage, this is not of primary importance. Another term for both parallel and ornamental is **vertical** change in the music – considering how to manipulate and deliver two or more audio resources that sound simultaneously at time a .

Unreal Engine offers MetaSounds for improved musical timing control. Unity has an audio plug-in software development kit that allows one to develop custom audio plug-ins.³ Both Unity and Unreal support audio middleware like Wwise and FMOD that extend their native audio capabilities. These solutions make precise musical synchrony possible, but they typically involve additional configuration or low-level coding. By contrast, in Web Audio (e.g., with Tone.js), such timing is accessible with relatively simple

function calls, which makes rapid prototyping and development more convenient.

If a developer has one passage of music that they want to play back at time a and another passage of music that they want to play back at time $b > a$, often it is not possible to achieve precision in this regard. Musically, the aim may be for the first and second passages to overlap in sync (Figure 1A), or for the pulse established in the first passage to continue unadulterated into the second (Figure 1B), but often issues to do with clocks and resources result in the passages being out of sync (Figures 1C and D), which corresponds to a suboptimal musical experience.

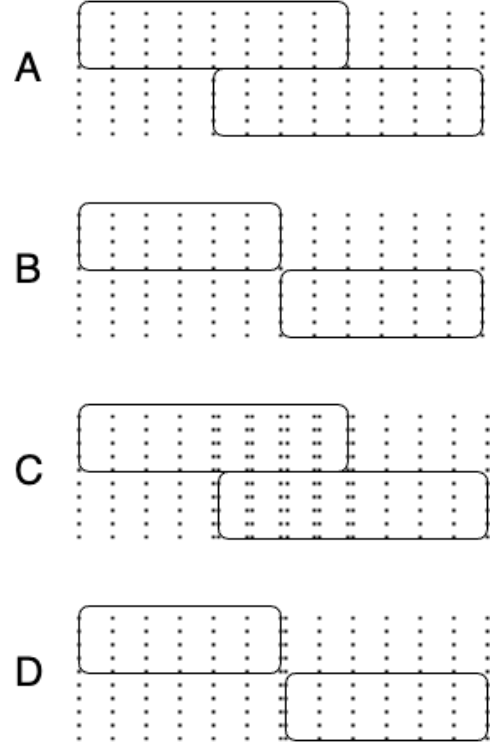


Figure 1: (A) In-sync sequencing of two overlapping passages of music; (B) In-sync sequencing of two non-overlapping passages; (C) Out-of-sync sequencing of two overlapping passages of music; (D) Out-of-sync sequencing of two non-overlapping passages. In all of the above, oblongs represent the passages of music and dashed lines indicate the perceived pulse.

A developer’s more advanced musical objectives – such as dynamic and independent time-stretching and pitch-shifting, or creation/alteration of note content – are often more complex to implement and less commonly featured in audio engines. Some games represent exceptions to this, such as *Genesis noir* [3]. In one of this game’s levels, the saxophone is played by the human player, and a rule-based AI derived from a system called the Impro-Visor [7] uses the player’s output to condition the response. As the improvisatory conversation is monophonic (not played on top of an accompaniment), timing issues are not so acute.

The Web Audio API (WAA) [1] has been in development as a specification since 2011, with implementations and more widespread use emerging in the mid-2010s. The WAA provides the developer access to the audio subsystem’s hard-

³<https://docs.unity3d.com/Manual/AudioMixerNativeAudioPlugin.html>

were clock [21]. Via lookahead scheduling [21], this means the WAA offers timing capabilities on par with or exceeding other game or XR audio frameworks. Paired with frameworks for the delivery of visual XR, such as Three.js or A-Frame, there is potential to support a developer’s basic and more advanced musical objectives. An early and non-XR example of the WAA’s capabilities for scheduling and sync is Madeon’s *Adventure machine* [10]: the user is able to select and deselect various components of a music track to play back dynamically (on the fly). While this could have been achieved by the transitional approach introduced above – starting all tracks and fading down all but the selected components – it is possible using the WAA to actually start and stop these resources at precise times. An example of a dynamic experience where the note content itself is determined in the moment is [8].

2.1 Matters of Evaluation

Prior research establishes strong connections between immersion and engagement in interactive experiences. McMahan [13] emphasizes that immersion occurs when users become mentally absorbed in an experience’s narrative elements, highlighting the psychological rather than technological aspects of engagement. This perspective is particularly relevant to our study, where engagement centers on the audio experience rather than complex visual or haptic interactions. Jennett et al. [5] further demonstrate that this type of engagement can be measured both subjectively and objectively, defining immersion as an experience characterized by engagement, engrossment, and decreased awareness of the real world. This framework aligns with our methodology of measuring engagement through both self-reported ratings and objective dwell times.

Witmer and Singer’s [22] definition of immersion as a psychological state of being enveloped by an environment that provides continuous stimuli is also relevant to our audio-focused study. In our context, the musical experience serves as the primary environmental stimulus, making engagement with the audio elements analogous to the immersive states described in traditional VR research. While our virtual environment offers limited interaction, the continuous stream of musical stimuli creates opportunities for psychological engagement similar to those found in more interactive virtual experiences.

3. DESIGN CONCEPTS

3.1 Smooth Tempo Transitions

We use granular synthesis to achieve smooth tempo transitions. Granular synthesis [23, 16, 14] involves pre-computing the **short-time Fourier transform (STFT)** of an audio signal, so that the frequency content of short temporal segments of a sound file – known as *grains* – are available, and can be recomposed via the **inverse STFT (iSTFT)** on the fly (in real time).

Common applications of granular synthesis are **time-stretch** (pitch-independent tempo change) and **pitch-shift** (tempo-independent pitch change).⁴ The former is achieved by applying the iSTFT at a rate different to the original analysis; the latter is achieved by a simple operation in the frequency domain, followed by application of the iSTFT at

⁴<https://tomcollinsresearch.net/mc/ex/grains/>

the original rate. Time-stretch and pitch-shift can also be applied in combination. There are limits to the perceptual effectiveness of granular synthesis, however, parametrized by the grain size (the duration of each temporal segment) and the overlap (the amount of cross-fade between consecutive grains). If one slows down a drum track too much, for example, one begins to hear multiple instances of a drum hit where the original recording contains only one.

In our web VR experience VRTGO, we wanted the avatar’s location in 3-dimensional space, notated $\mathbf{a} \in \mathbb{R}^3$, to determine the playback tempo. We had three alternative song versions at our disposal:

1. The released version, with an initial tempo of 90 bpm, falling to 85 bpm halfway through, and then increasing to 156.5 bpm towards the end;
2. A “sparkling pop” version of the opening half of the song, with a tempo of 95 bpm;
3. A “lo-fi guitar” version of the opening half, with a tempo of 85 bpm.

Web-based and native software provide time-stretch and pitch-shift either offline or in real-time (e.g., Traktor for DJs by Native Instruments). It is the real-time time-stretch capability we require here, enabling us to align the timing of the playback of the three versions on-the-fly, as an avatar moves around the space.

We conceptualize the three song versions as being located distinctly in \mathbb{R}^3 at \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 , respectively. We notate the tempo of song version i by t_i . While the tempo changes in the released version mean that t_1 is also a function of position in the song (in terms of measures, beats, and subdivisions of the beat), we will not notate this dependency explicitly.

Additionally, we conceptualize that the closer the avatar \mathbf{a} is to song version i , the closer the playback tempo t ought to be to t_i . The formulae we use to derive t are

$$t = t_j + \frac{d_j}{d_j + d_c}(t_c - t_j) \quad (1)$$

$$t_c = \sum_{i=1, i \neq j}^n \left(1 - \frac{d_i}{s}\right) t_i \quad (2)$$

where t_j is the tempo of the song version j to which \mathbf{a} is closest, d_j is the Euclidean distance between \mathbf{a} and \mathbf{x}_j , d_c is the Euclidean distance between \mathbf{a} and the centroid of the versions to which it is not closest $(\mathbf{x}_i)_{i=1, \dots, j-1, j+1, \dots, n}$, and s is the sum of the Euclidean distances between \mathbf{a} and the versions to which it is not closest.

In combination, Eqs. (1) and (2) mean that if the avatar \mathbf{a} is right at the locus \mathbf{x}_j of song version j , then d_j in Eq. (1) tends to zero and the tempo is exactly t_j . If the avatar is at the centroid of the other song versions, then d_c in Eq. (1) tends to zero and the tempo is exactly t_c . For any other location, it is a linear combination of t_j and t_c . Equation (2) represents a weighted mean of the tempi of song versions to which the avatar is not closest, weighted by a function of its Euclidean distances to those versions.

Advantages of the above formulae are:

- When the avatar is a minimal distance from a particular song version, the tempo (and beyond this, other

features of the song) is exactly as intended in that particular version. If it was *not* possible for an avatar to experience an intended version of a song, this could be perceived as a drawback by artists and users.

- They apply to $n = 2, 3, 4, \dots$ song versions as effectively as they apply to the $n = 3$ song versions utilized here;
- Equation (2) can be thought of as a **feature centroid**, potentially extending to other features beyond tempo that can be extracted from the music;

A disadvantage of the above formulae is that discontinuities can be encountered when the avatar’s location switches from being closest to some version location \mathbf{x}_j to another version location \mathbf{x}_k instead.

3.2 Blending and Isolation of Sound Sources that Represent the Alternative Song Versions

We utilize a head-related transfer function (HRTF) [11] to provide the perception that sounds are emanating from certain locations as the avatar moves around the various sound sources in the XR space. Brandenburg et al. [2] suggests that individualized HRTFs may not be necessary for plausible sound reproduction: “Contrary to many older publications, new research gives hints that when the other cues mentioned here are applied correctly, mean HRTFs like from a commercial dummy head are sufficient for a fully plausible sound reproduction via headphones” [2, p. 2]. Nevertheless, our current solution [11] employs such a mean HRTF.

As mentioned in Section 3.1, the three song versions are conceptualized as being located distinctly in \mathbb{R}^3 at \mathbf{x}_1 , \mathbf{x}_2 , and \mathbf{x}_3 , respectively. Therefore, when the avatar’s location \mathbf{a} changes to be much closer to version $j \in \{1, 2, \dots, n\}$ than the other versions, our HRTF solution ensures that the sound sources associated with version j are perceived as closer than those associated with the other versions, due to an increase in loudness and decreases in interaural time difference (ITD) and interaural level difference (ILD).

That being said, during the design process, we perceived that even when the avatar is positioned directly at the location \mathbf{x}_j of a sound source for a particular song version, sound sources for other, relatively distant song versions at locations $(\mathbf{x}_i)_{i=1, \dots, j-1, j+1, \dots, n}$ are too noticeable. While we retain our HRTF solution, we incorporate gain components that reduce the loudness by 14 dB of the sound sources associated with song versions to which the avatar is not closest. When the avatar moves such that the closest version of the song switches from \mathbf{x}_j to \mathbf{x}_k instead, the gains alter gradually over 5 sec, to smoothen the transition.

3.3 Serial-recursive Loading

In any XR experience involving music/audio, it can be useful for developers to segment the audio, both in terms of vertical layers (e.g., into different instruments or tracks) and horizontal layers (e.g., into separate files based on whether they are likely to be played back sooner or later). The former may help avatars perceive different versions of the music more readily (part of what we refer to as *dynamic experience*), as the different vertical layers can be remixed on the fly; the latter may help reduce loading of and seeking through resources, as an experience can be started sooner if

an audio file of size 100 MB, say, is split into four files each of 25 MB in size, and the loading of segments 2, 3, and 4 continues while playback of segment 1 begins.⁵ While horizontal splitting reduces the amount of time a user has to wait before beginning an XR experience, vertical splitting has the opposite effect – on average, n vertical layers will take n times longer to load than one vertical layer.

For VRTGO, we split the audio resources both horizontally and vertically. Our decision to implement both horizontal and vertical splitting was driven by specific technical goals: vertical splitting (separating the audio into different instrumental tracks) was chosen to highlight the perceptual contrast between different versions of the song, allowing users to more readily identify the distinct musical characteristics of each version as they move through the space. This design choice supports our aim of creating a dynamic musical experience where the avatar or the user’s position meaningfully affects what they hear; horizontal splitting (dividing the song into temporally sequential segments) was implemented primarily to optimize loading times and resource management. By loading only what’s immediately needed while buffering upcoming segments, we reduce the initial wait time for users while maintaining audio continuity. These splitting strategies work together to support both the technical and artistic goals of creating a responsive, dynamic musical environment.

Overall, we have 210 audio files that need to be loaded (more details on the splitting given below). We implement serial-recursive loading to achieve this: for a given track in a given song version, we write a recursive function whose recursive call loads the next segment of the track, enabling us to prevent segment $n + 1$ from being loaded until segment n is loaded. Aside from this, song versions and tracks within versions are loaded in parallel. This ensures that the user can begin the experience as soon as possible.

Since we produced all three versions of the associated song “VTGO (Vertigo)”, we had at our disposal the stems for the versions listed in Section 3.1. We made 5 vertical layers for versions 1 and 2, and 3 vertical layers for version 3. Since we also had a 5-letter abbreviation for the experience’s name – VRTGO – we made the following letter-track associations:

- V for drums;
- R for bass;
- T for guitar/other;
- G for backing vocals;
- O for lead vocal.

We represent the association visually by using alphabet blocks, which bear each of these letters in the rendered experience.

Mostly, we split the tracks into 4-measure segments, though some segments are slightly longer or shorter than 4 measures, depending on the musical structure. For example, a particular alphabet block bearing the letter R in the

⁵Of course, there is a risk that if the connection speed is not fast enough, segment $n + 1$ may not be loaded into memory when the playback of segment n is complete [21]. The library used for playback also has to offer sample-level accuracy, otherwise artifacts may degrade the audio playback experience at the boundaries between segments.

experience may represent the bass part from measures 41–45 of version 1 of the song.

In the current version, we do not place much emphasis on the vertical layering – in other words, it is not possible for an avatar to hear the bass track in isolation from other tracks, say; the horizontal layering is more evident, however, as the alphabet blocks associated with each version form separate spirals of blocks in the 3D space, and they are highlighted/low-lighted as their playback begins/ends, from the base of the spiral of blocks and upwards and outwards as the song proceeds. One such block spiral is shown in Fig. 2.

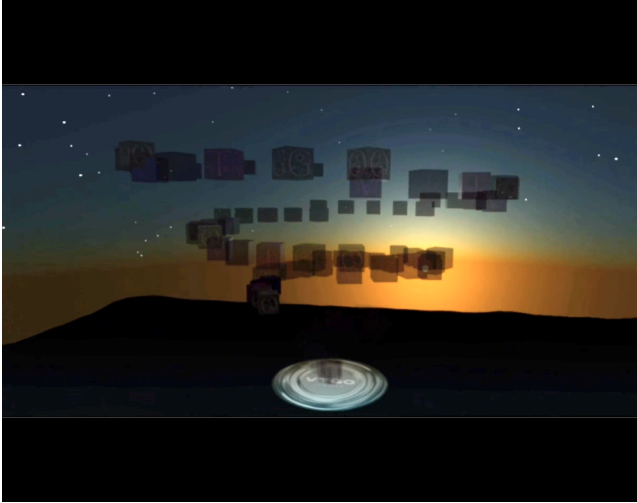


Figure 2: A spiral-shaped collection of alphabetic blocks that bear the letters V, R, T, G, and O. Each spiral corresponds to one of the three song versions, and each block corresponds to approximately 4 measures of music on drums, bass, guitar/other, backing vocals, or lead vocal.

3.4 Implementation in A-Frame and Tone.js

A-Frame is a JavaScript library for rendering of (mainly) visual components of a web VR experience. We utilized an extension of A-Frame called the Environment Component [19] to render the ground and sky shown in Fig. 2. The alphabet blocks on the cubes are from the Westminster Press circa 1925.

Tone.js [11] is a JavaScript library that makes some of the underlying WAA [1] more easy to use, as well as extending it in certain respects, such as with the inclusion of granular synthesis and syntax for scheduling/timing of events, as well as converting between musical terminology such as [bars (measures), beats, sixteenths] and absolute time in seconds.

“The primary paradigm [of the WAA] is of an audio routing graph [also known as a signal flow graph], where a number of `AudioNode` objects are connected together to define the overall audio rendering” [1]. In our implementation, each of the blocks depicted in Figure 2 (as well as those in the other two block spirals) is an instance of the class `MusicCube`. Each instance of this class contains an instance of the Tone.js class `GrainPlayer`, which provides control over time-stretch (tempo manipulation) for each segment of each track. These are the main constituents of our audio routing graph. Instances of `MusicCube` are properties of another class we implement called `SongSource`. The `SongSource` class is respon-

sible for a particular block spiral, both visually and sonically.

The other two classes in our implementation are `GamePlayer` and `GameState`. The former contains little other than Tone.js’ `Listener`, which is the avatar’s head location and orientation;⁶ the latter has an instance of `GamePlayer` and three instances of `SongSource` as properties. It also has all the methods that support calculation of the feature centroids and updating of the tempo and gains. Throughout the experience, Equations (1) and (2) are applied every 16th-note subdivision of the beat, which we found adequate to produce the perception of smooth tempo transitions, as well as the gain changes mentioned in Section 3.2.

4. EVALUATION

To evaluate our proposed system, an IRB-approved user study is conducted. We evaluate user behavior in a web VR environment comprising a dynamic music system according to the approach proposed above, compared to a web VR environment with identical visuals but comprising a static music system.

The intent of this listening study is to evaluate the user’s enjoyment of the experience as well as the amount of time in which they dwell in dynamic versus static conditions. The user study was designed with two primary objectives: (1) to assess quantitatively whether users exhibit different engagement behaviors and report different enjoyment levels when experiencing music through an interactive, spatially dynamic system compared to a static musical presentation; and (2) to explore qualitatively users’ perceptions of the potential benefits and limitations of dynamic musical experiences in VR. These objectives give rise to the following research question regarding how spatial interactivity might transform traditional musical listening experiences. We employed a within-subjects experimental design with two conditions (dynamic and static musical experiences) to allow direct comparison of the same participants’ responses to both interaction modes.

RQ: To what extent do participants report higher enjoyment ratings for the dynamic musical experience compared to the static musical experience?

Based on this research question, we developed two hypotheses regarding participant behavior:

H_A: Participants spend longer in the dynamic versus the static musical experience, so we predict longer dwell times in the dynamic versus static musical experiences.

H_B: Participants will rate the dynamic musical experience as more enjoyable than the static musical experience.

While we have no a priori hypothesis concerning engagement, ratings of engagement constitute the other quantitative data point that we collect.

4.1 Design

The study uses a single-factor within-subjects design in which the independent variable (IV) is musical experience (dynamic or static version of VRTGO) and the DVs are dwell time, rating of enjoyment, and rating of engagement. We collect some qualitative data also, as described below in Materials. By collecting both quantitative data (e.g., engagement ratings) and qualitative feedback (e.g., thoughts

⁶A-Frame and Tone.js’ `Listener` utilize different coordinate systems, so it is necessary to convert between the two in order to make these packages interoperable.

on specific points where a use felt most or least engaged), we aim to develop a comprehensive understanding of how our proposed approach affects the user experience. This evaluation methodology was chosen to balance objective behavioral measures with subjective experiential feedback, providing insights that could inform future development of dynamic music experiences in XR environments.

4.2 Participants

Participants ($n = 8$) were recruited for this study from an undergraduate and graduate college student population enrolled in music-technological majors at the Frost School of Music, University of Miami. These participants represent a particularly relevant sample population, as they study in a musical environment where individual and ensemble lessons/rehearsals occur on a daily basis, and 2-3 concerts in various campus spaces take place each day during the semester. This musical background gives them a heightened sensitivity to and critical awareness of audio experiences. Participants had sufficient computer literacy to navigate virtual environments using a VR headset. The sample size, though modest, is adequate for the initial evaluation of the system and our hypotheses, given the specialized nature of the participant pool and their relevant expertise in both music and technology.

4.3 Materials

Two versions of the virtual environment are developed and hosted online: a dynamic version (<https://vrtgo.onrender.com>) and a static version (<https://srtgo.onrender.com>). Both environments are visually identical and feature the same controls, including play/pause and stop functionality in addition to ordinary navigation of a space. The dynamic version incorporates real-time mixing of sources and HRTF for sound localisation, as well as granular synthesis for time-stretching, allowing the music to respond to player location and movement. The static version plays the original, unmodified version of VTGO regardless of player position or activity. The virtual environment consists of geometric structures arranged in a spiral pattern, around and through which participants can navigate.

The questionnaires presented to participants are as follows:

- After each experience (dynamic and static music):
 - To what extent did you enjoy this experience? [Likert scale 1–7]
 - Rate your level of engagement with this experience. [Likert scale 1–7]
 - Was there a specific point where you felt **most engaged** or **least engaged**? If so, explain where and whether you’re talking about “most” or “least”. [Open-ended text response]
- After completion of both experiences and post-experience questionnaires:
 - In terms of music, which experience did you prefer? [2-alternative force-choice]
 - Do you think the ability to control musical features **enhances the experience** compared to a static version? If yes, why? [Open-ended text response]

- What do you see as potential **benefits** of experiencing music in VR? [Open-ended text response]
- What do you see as potential **drawbacks** of experiencing music in VR? [Open-ended text response]

Dwell time is measured as the duration participants spend in each version of the experience, serving as an objective measure of engagement.

4.4 Procedure

The study employs a counterbalanced design to control for order and fatigue effects. Following informed consent, participants are assigned randomly to experience either the dynamic or static version first. After each experience, participants complete the first questionnaire above. Finally, they complete the second questionnaire and are debriefed.

4.5 Apparatus

Meta Quest 2 and 3 VR devices are utilized for the experience, and a pair of Audio Technica M40X headphones are provided to the participants.

The participants switch to a standard computer for completion of the questionnaires via Qualtrics.

4.6 Results

To test H_A , we conduct a paired-samples t -test comparing dwell times between conditions. Participants spent significantly longer in the dynamic condition ($M = 7.25$ min, $SD = 2.96$ min) compared to the static condition ($M = 5.13$ min, $SD = 2.70$ min), $t(7) = 2.49, p < .05, d = 0.879, 95\% \text{ CI} = [0.031, 1.685]$. This finding supports our hypothesis that participants spend longer in the dynamic versus static musical experience.

To test H_B , we conduct a Wilcoxon signed-rank test (non-parametric being appropriate for Likert-scale data) comparing enjoyment ratings in the dynamic condition (median = 4.5, $IQR = 1.75$) compared to the static condition (median = 4.5, $IQR = 3.75$). It reveals no significant difference in enjoyment ratings ($W^+ = 5, SE = 3.54, p = .480$).

While we have no hypothesis regarding engagement ratings, we conduct a Wilcoxon signed-rank test also comparing engagement ratings in the dynamic condition (median = 5.5, $IQR = 2.5$) compared to the static condition (median = 4, $IQR = 2.5$). It reveals a significant difference in engagement ratings ($W^+ = 0, SE = 3.62, p < .05$). This aligns with the observed difference in dwell times, suggesting that the dynamic music system creates a more engaging experience, even though this did not translate to higher enjoyment ratings.

As for the qualitative questions and responses regarding the proposed dynamic music system, participants report feeling most engaged during specific interactive or novel moments, such as when blocks move with the music or when sounds shift spatially. However, the least engagement occurs during moments of confusion, unpredictability in sound transitions, or when the music ended with no further actions available (even though stop and play was available, perhaps this was not clear to all participants). For the static version, engagement peaks in the initial moments, often tied to observing block patterns or attempting to effect changes in the music. However, engagement wanes when participants realize that movement or actions have no auditory effects.

Most participants agree that the ability to control musical features enhances the experience, noting that it introduces interactivity, spatial audio dynamics, and deeper immersion. They feel it transforms the activity from passive listening to active exploration. However, clarity and synchronization are identified as areas for improvement, and two out of eight participants still prefer a static mix for traditional music enjoyment.

Participants noted moments of confusion during transitions between song versions, indicating that while variation engages listeners, too much unpredictability may negatively impact enjoyment. This suggests that composers creating for such environments might benefit from designing recognizable motifs or elements that persist across variations, providing continuity in dynamically changing environments.

The lack of significant difference in enjoyment ratings between conditions, despite increased engagement, suggests that composers should view this approach as complementary to rather than replacing traditional listening experiences. Music creators might consider developing compositions specifically designed for spatial interaction, rather than simply adapting existing works.

Based on the observation of user behavior during the study, we noted that participants tended to move more deliberately and purposefully in the dynamic version of the virtual environment, suggesting they were actively exploring the sonic possibilities. This behavior implies that composers might benefit from designing clear sonic landmarks or destinations within their spatial compositions to reward exploration and discovery—elements that traditional linear compositions typically cannot incorporate.

Identified benefits include enhanced creativity, interactive dimensions, and deeper emotional engagement, with potential for innovative applications like interactive music videos. Drawbacks involve high VR costs, accessibility issues, reduced realism compared to live music, over-immersion risks, and potential impacts on the live music industry.

5. DISCUSSION

How music is created, produced, and disseminated is of great interest to practitioners and audiences, but also to those working in intermediary roles in the music industry, and those that study music, especially modern artistry development and entrepreneurship, musicology, and music technology. While major innovations of the last decades such as streaming have changed music creators' revenue streams [4], there has been little or no disruption in the "one-way street" of musicians making, companies distributing, and fans consuming music.

More potentially disruptive in this regard are XR and artificial intelligence, either in isolation or combination. We see via these technologies the potential for "driving the other way up the one-way street" – audience members might access and manipulate elements of a musical experience such as the tempo and constituent tracks, in ways that may be of interest to the artists, producers, and other fans. The conceptual bases for such experiences are developed in this paper, and a web VR implementation called VRTGO via which they are realized acts as a proof of concept.

We highlight how web VR, and the WAA and Tone.js in particular, offers developers the opportunity to fulfill some advanced musical objectives – such as dynamic and independent time-stretch and pitch-shift, or creation/alteration

of note content.

Our study explores the potential of dynamic musical experiences in web VR through a user study that employs both objective and subjective measures. The methodology was intentionally structured to provide an understanding of user interaction within a dynamic musical experience.

Participants spent significantly longer in the dynamic condition, with an average of 7.25 minutes compared to 5.13 minutes in the static condition. This difference suggests that the dynamic musical experience successfully captured and sustained user attention more effectively than the static version. Moreover, a statistically significant difference in engagement ratings was found, indicating that the interactive musical environment created a more immersive experience.

Despite the increased engagement, no significant difference was found in enjoyment ratings between the dynamic and static conditions. This finding demonstrates that increased interaction does not necessarily translate directly to increased enjoyment. Returning to our primary research question regarding enjoyment of dynamic versus static musical experiences, our results present a nuanced picture. While we did not observe significantly higher enjoyment ratings for the dynamic condition, the significantly longer dwell times and higher engagement ratings suggest that the dynamic musical experience captured participants' attention and interest more effectively. This distinction between enjoyment and engagement highlights the importance of considering multiple dimensions of user experience when evaluating novel musical presentation and/or design methods.

The qualitative questions provided further insights: participants saw potential for enhanced creativity and emotional engagement, however, challenges were also identified – particularly around the clarity of interaction and synchronization of musical elements. The study and its findings suggest that such innovative interfaces can create more engaging musical experiences, although the path to validating optimal user enjoyment remains challenging.

5.1 An Artist's Perspective on the Potential for Dynamic Experiences of Alternative Song Versions in XR

The concept of dynamically experiencing music within VR environments such as VRTGO introduces significant potential for innovative musical engagement and composition. Audiences moving through a digital environment where their proximity to specific points actively transforms the music they hear—from tempo shifts to instrumental prominence—all in real-time, opening up a whole new avenue of inventive composition.

This approach allows for various forms of physical and interactive engagement, offering valuable insights into how different user groups—including DJs, educators, and those seeking meditative experiences—may choose to engage with music in this adaptable format. Musical versions can continuously evolve, where each listener becomes an integral part of the song's structure by exploring the space. The continuously evolving nature of these musical compositions allows each listener to become an active participant in shaping the musical experience, transforming a static piece into a dynamic one where each interaction leads to a distinct version of the piece.

In addition, this interactive model has the potential to inspire collaborative listening experiences, enabling multiple

users to influence the soundscape collectively by adjusting or isolating specific musical elements as they explore together. Such possibilities redefine traditional concepts of audience engagement, positioning music not merely as a fixed composition to be received but as a shared, immersive experience with transformative and co-creative potential.

5.2 Limitations

While we kept the visual components of VRTGO relatively simple to help focus on the musical aspects, we would have liked to include an avatar of the main performer in the center of the experience, dancing to the music with movements scheduled to adapt to the dynamic tempo controlled by the user.

We also noticed that 'Tone.js' claim of sample-accurate scheduling does not hold when the avatar starts moving around the space before all audio resources are loaded. So even though we implement serial-recursive loading as described in Section 3.3 to enable the avatar to begin exploring the space before all audio files are loaded, pursuing such an option can lead to a suboptimal audio experience.

5.3 Future work

In other projects, we have explored the possibilities for multiuser interactions via the JavaScript library called Socket.IO, which enables bidirectional and low-latency communication between users. In a future version of VRTGO that supports networked interactions via Socket.IO, we would like to make it possible for users to hear the effects of actions of other users. One idea would be to use raycasting to enable users to isolate and hear the musical contents associated with individual alphabet blocks. They could also remove blocks from the block spiral, and deposit them elsewhere in the space [6, for example]. This would mean that temporarily at least, these blocks do not form part of an intended version of the song.

While the application of Equations (1) and (2) work well in the current experience, it would be worth investigating other formulae such as the softmax function for deriving tempo and other musical feature values from the avatar's location.

Future studies might explore comparative designs across different musical genres, examining how the dynamic interaction method performs with various types of musical compositions. Studies could also investigate whether the novelty effect of interactive musical experiences diminishes over repeated exposures and how users' engagement might evolve with prolonged interaction.

A larger and more diverse sample size would enhance the statistical robustness of the findings. Implementing more granular measurement tools, such as physiological tracking (e.g., heart rate, galvanic skin response [17]) can provide additional objective measures of engagement beyond dwell time and self-reported ratings.

Incorporating more detailed variations of the design environment could help identify specific design elements that most effectively enhance musical engagement. Interdisciplinary collaborations between music technologists, cognitive psychologists, and user experience researchers could provide more comprehensive information on the psychological and perceptual mechanisms that pertain to dynamic musical experiences in virtual environments.

If we can address some of the above limitations and pursue

some of the ideas for future work, we see great potential for music artists and audiences in exploring dynamic web VR experiences of alternative song versions.

6. ACKNOWLEDGMENTS

This work is supported by a grant from the University of Miami's U-LINK fund to the project Concerts with Humans and Artificial Intelligence (CHAI).

7. REFERENCES

- [1] P. Adenot and H. Choi. Web audio API. <https://www.w3.org/TR/webaudio/>, 2021. Accessed: 2024-07-09.
- [2] K. Brandenburg, C. Fascella, N. Merten, R. Profeta, U. Sloma, T. Thron, and F. Wollwert. Implementation of and application scenarios for plausible immersive audio via headphones. In *Audio Engineering Society Convention 155*. Audio Engineering Society, 2023.
- [3] Feral Cat Den and Skillbard. Genesis noir. <https://genesisnoirgame.com/>, 2017. Accessed: 2024-07-09.
- [4] D. Hesmondhalgh, R. Osborne, H. Sun, and K. Barr. *Music creators' earnings in the digital era: Ground-breaking research into how creators earn money through streaming*. Intellectual Property Office, Newport, Wales, 2021.
- [5] C. Jennett, A. L. Cox, P. Cairns, S. Dhoparee, A. Epps, T. Tijs, and A. Walton. Measuring and defining the experience of immersion in games. *International Journal of Human-Computer Studies*, 66(9):641–661, Sept. 2008.
- [6] S. Jiang, L. Lim, and M. Sra. Spatializing music in virtual reality. In *Proceedings of the 2023 ACM Symposium on Spatial User Interaction*, pages 1–3, 2023.
- [7] R. M. Keller and D. R. Morrison. A grammatical approach to automatic improvisation. In *proceedings of the sound and Music computing conference*, pages 330–337, 2007.
- [8] A. Kobayashi, R. Ishino, R. Nobusue, T. Inoue, K. Okazaki, S. Sawa, and N. Tokui. MR4MR: Mixed reality for melody reincarnation. *arXiv preprint arXiv:2209.07023*, 2022.
- [9] Kwek, Nick for BBC Click. Music tech special. <https://www.bbc.co.uk/programmes/m001lpy1/>, 2023. Accessed: 2024-07-09.
- [10] Madeon. Adventure machine. <https://adventuremachine.4thfloorcreative.co.uk/>, 2015. Accessed: 2024-07-09.
- [11] Y. Mann. Interactive music with Tone.js. In *Proceedings of the Web Audio Conference*, Paris, France, 2015.
- [12] H. M. McLuhan and B. Nevitt. *Take today: The executive as dropout*. Longman, 1972.
- [13] A. McMahan. Immersion, engagement, and presence: A method for analyzing 3-d video games. *The Video Game Theory Reader*, pages 67–86, 2003.
- [14] S. Riedel, M. Frank, and F. Zotter. Perceptual evaluation of listener envelopment using spatial granular synthesis. *arXiv preprint arXiv:2301.10210*, 2023.

- [15] G. Ritzer, P. Dean, and N. Jurgenson. The coming of age of the prosumer. *American behavioral scientist*, 56(4):379–398, 2012.
- [16] C. Roads. *Microsound*. MIT Press, Cambridge, Massachusetts, 2001.
- [17] H. A. Shehu, M. Oxner, W. N. Browne, and H. Eisenbarth. Prediction of moment-by-moment heart rate and skin conductance changes in the context of varying emotional arousal. *Psychophysiology*, 60(9), 2023.
- [18] R. Stevens and D. Raybould. *Game audio implementation: a practical guide using the unreal engine*. Routledge, 2015.
- [19] Supermedium. A-Frame Environment Component. <https://github.com/supermedium/aframe-environment-component>, 2017. Accessed: 2024-07-09.
- [20] A. Toffler. *The third wave: The classic study of tomorrow*. Bantam, 2022. Originally published 1980.
- [21] C. Wilson. A tale of two clocks: Scheduling web audio with precision. <https://web.dev/articles/audio-scheduling>, 2013. Accessed: 2024-07-09.
- [22] B. G. Witmer and M. J. Singer. Measuring presence in virtual environments: A presence questionnaire. *Presence*, 7(3):225–240, 1998.
- [23] I. Xenakis. *Formalized Music: Thought and Mathematics in Composition*. Indiana University Press, Bloomington and London, 1971.