CLOUD TANK:

Virtual Reality Performance Instrument for Embodied Audiovisual Remixing

by Asya Ulger

Advisor: James M. Mahoney

Submitted to the Department of Computer Science

Dartmouth College

In Fulfilment of the Requirements

For a Bachelor of Arts Degree

Hanover, New Hampshire

June 2026

Abstract

Traditionally, live audiovisual performances are made behind a desk using a laptop and an audio controller. This setup restricts the performer's movements and separates the artist from their work. Moreover, it narrows what counts as the performance to the output alone — the audience is preconditioned to treat the projected image and sound as the whole event, and not the artist producing it. Cloud Tank challenges this by positioning the performer's body as the instrument, using virtual reality. As the performer, I mix spatialized audio and drive beat-aligned, reactive video layers through hand movement and hand-pose recognition. The audiovisual output is streamed and projected for the audience in real-time. Thus, the artist's movement and choreography become an integral part of the performance, adding a layer for both the audience and the artist to explore. Cloud Tank is built in Unity for the Meta Quest 3; this thesis presents its architecture, three live Pieces performed on it, and what designing the instrument revealed about embodied performance in VR.

Preface

I would like to express my deepest gratitude to my advisor, James Mahoney, for his continuous support, mentorship, and encouragement throughout this project.

Contents

List of Figures

CHAPTER 1

Introduction

Cloud Tank is an instrument for embodied audiovisual remixing in virtual reality, designed for live performances. The performer, I, wear the headset and mix spatialized audio and drive reactive visuals through hand movement and pose recognition. The audio and the resulting visuals are streamed in real time. The audience does not wear a headset, instead watches the performer move to make music, as they see the streaming output being projected. Thus, Cloud Tank combines audio, visual art, and physical/dance-like movements to create one performance. This chapter explains why I built it, the problem it responds to, and why I describe it as an instrument rather than a tool. Particularly, why I describe it as my instrument.

1.1 Background and Motivation

Over the past four years, I have been drawn to explore the intersection of technology and art. I believe the most interesting work is developed when a technical system is not just treated as a tool but is a part of the artwork itself. For me, Cloud Tank has been the fullest expression of that interest: I wanted to build an instrument for myself. I decided on a VR environment as it inherently makes accessible the two crucial components of this system, body/hand tracking and gesture recognition, without the need for additional hardware. The headsets are also designed to allow and encourage movement, which made it a natural fit for the project. Building Cloud Tank thus became a way to explore both what the technology can offer a live performer and what kind of artistic practice it makes possible.

The idea of creating an instrument stemmed from a frustration with how live audiovisual performance is often produced, and a personal interest toward alternative instruments. Traditionally, when it comes to audiovisuals, the performer is behind a laptop and an audio controller, building the sound on a flat surface, while the audience watches a projection. This arrangement works but also creates separation — voluntarily or involuntarily — between the performer and the art piece. The setup also limits the artist to sit or stand with minimal movement of the body. This is a well-documented problem, especially from the audience's perspective. Caleb Stuart, an experimental music scholar, argues that in laptop concerts the audience cannot perceive any causality between the performer's understated actions — typing, moving a mouse — and the resulting sound [1]. In a study of how listeners perceive laptop music, Bown and colleagues found that audiences associate liveness with visible performer activity, such as audible gesture and improvisation [2].

However, it is important to clarify what this critique is not. The laptop and the audio controller are not limited instruments: they offer far more raw capability than the system described in this thesis, in the range of sound they can produce, and the precision of their control. My argument is not that these are inadequate performance tools, but that they decide what a performance is allowed to be. My aim is to rethink that arrangement and explore what a performance can be when the performer's body becomes the visible instrument for the audience to see.

Furthermore, over the last two decades, a growing community of artists and researchers has pushed back on the desk-and-fader model and built alternatives. The New Interfaces for Musical Expression (NIME) conference has grown into a large field dedicated to this exact question, with a steady increase in research output and new gestural instruments year over year [3]. Musicians have built their own answers: Imogen Heap, designer of the *Mi.Mu* gloves, describes the

conventional setup as “pushing buttons and twiddling dials” and argue that “it is not very exciting for me or the audience” [4]. The live-coding community has taken a different approach, bringing the normally hidden computation part of computer-music performance to the foreground and treating programming itself as a public practice, so that the labor of the performance is made visible to the audience [5]. Different as they are, these efforts share a single conviction: that the audience should be able to see the performance being made.

While acknowledging the shortcomings of a VR space compared to traditional audiovisual equipment, there is precedent that new and “awkward” technology can grow into a real instrument. The Theremin — one of the very first electronic instruments and an inspiration for Cloud Tank — was played by moving the hands through space, without touching anything [6]. When it first appeared in the late 1920s, critics were confused as it was difficult to play; Theremin remained a novelty for years [7]. However, in the right hands, it proved to be genuinely expressive and eventually paved the way for Robert Moog's synthesizer as well as the electronic music that followed [7]. Looking at the Theremin, we can argue that the early roughness of a technology is not evidence against it. Extended reality is, today, roughly where the Theremin once was — a young and imperfect medium for performance. But XR offers something unique: the body is already tracked, in space, and legible to the system, so the movements I make are the movements the headset can read. Cloud Tank is my attempt to explore that opportunity and investigate what a body-first audiovisual instrument can be when it is built in VR.

1.2 Cloud Tank as an Instrument

Throughout this thesis, I refer to Cloud Tank as an instrument rather than a tool, and that distinction is deliberate. Tools are operated: the performer issues a command, and the system carries it out the exact same way each time. Instruments are played: they have a slow learning curve; they are practiced with the body and often performed from memory. Wessel and Wright describe what separates the two as control intimacy: instruments have a low-variance coupling between a performer's gesture and the resulting sound, and they are paired with initial ease of use and a long-term potential for virtuosity [8]. An instrument, within this framework, is something a performer grows into and takes ownership of. I designed Cloud Tank to be played in this manner. Its responsiveness, which Chapter 3 describes in detail, exists so that the system answers a movement quickly enough that performing feels continuous, and so that it can be rehearsed and eventually performed as choreography, without menus.

I argue that Cloud Tank is an instrument, and specifically, it is my instrument. During the design process, before even starting to design the interface, I began with the movements I wanted a performance to contain. I acted out the broad movements and hand gestures — reaches, sweeps and turns — and drafted the types of effects I wanted the gestures to map to. Then, I placed each interactive element where the chosen gesture would naturally land, retuning the layout for each performance. This choreography-first method, described in Chapter 3, means the instrument's layout is fitted to my body and my way of moving. I am, moreover, both its designer and performer: the process that built the system and the process that learned to perform with it were the same, carried out by the same person. The three Pieces documented in Chapter 4 are the

repertoire that came out of that process. This single-performer condition is also a limitation, one I return to in Chapter 5, but it is inseparable from what makes Cloud Tank the instrument it is.

1.3 Research Questions

Central Question. How can virtual reality be used to build a live audiovisual instrument in which the performer's body, rather than a control surface, becomes the visible site of the performance? This question breaks down into three smaller ones, each corresponding to a commitment that shapes the system described in Chapter 3:

1. ***Embodied movement.*** How can audiovisual control be mapped onto whole-body and hand movement, such that the instrument is played through motion rather than operated from a panel?
2. ***Legibility.*** What makes a performer's gestures readable to an audience, and which kinds of movement communicate across the gap between the headset and the projection?
3. ***Responsiveness.*** What is required of the system's architecture for it to answer a performer's movement quickly enough that performing feels continuous rather than transactional?

The first and second questions concern how Cloud Tank is performed; the third concerns how it is built. The chapters that follow take them up in turn: Chapter 3 describes the architecture and design decisions through which the system attempts to answer them, Chapter 4 documents what performing the instrument revealed, and Chapter 5 evaluates the system against each commitment.

1.4 Contributions

This thesis makes three contributions, corresponding to the three senses in which Cloud Tank is a piece of work: a system, a practice, and a set of findings.

System. Cloud Tank is a virtual reality architecture, built in Unity for the Meta Quest 3, for embodied audiovisual remixing. It combines per-bus real-time audio analysis, beat-aligned reactive video layers, and hand- and pose-driven gesture mapping, and outputs the result into a single frame streamed to the audience in real time. Chapter 3 describes this architecture and the design decisions behind it.

Practice. Explains the ways of performing with the system, developed through the choreography-first method. This contribution includes the three Pieces documented in Chapter 4 and the account of how an instrument of this kind is rehearsed and brought to an audience.

Findings. Explores and evaluates the process of designing this instrument while learning to perform it. Specifically, what this process revealed about embodied performance in virtual reality. Chapter 5 develops these reflections and considers their limitations.

1.5 Conclusion

This chapter has made the case for Cloud Tank as an instrument, and as my instrument. The conventional setup for live audiovisual performance — a laptop and a controller on a flat surface — works, but it keeps the performer's body still and hides the connection between action and sound from the audience. Cloud Tank is my response to that arrangement: an

attempt to make the performer's movement the visible site of the performance, built in a medium where the body is already tracked and legible to the system. It is an instrument because it is meant to be practiced and played rather than operated, and it is mine because it was shaped around my own movement, by the same person who learned to perform with it. The chapters that follow describe how it is built and how it is performed.

CHAPTER 2

Related Work

2.1 Digital Musical Instruments and the Body

Cloud Tank can be classified as a digital musical instrument, or DMI. DMIs are electronic or computer-based instruments where the physical act of playing (gestural control), and the actual sound production are separated [10]. They differ from acoustic instruments as the relationship between a performer's action and the resulting sound is not fixed by physics; it is designed. In the case of acoustic instruments, the interface and the sound source are inseparable: a violin string is both what the musician plays, and what acts as the sound source through vibrations. In contrast, in a DMI, the input device and the sound generator are separate pieces of equipment; the relationship between a performer's action and the resulting sound must be explicitly designed. Miranda and Wanderley frame this separation as a move “beyond the keyboard” — an attempt to find other forms of control than the piano-derived interface that electronic instruments inherited by default [9].

This freedom is also the central challenge for DMIs. The mapping between input and sound is not an implementation detail; it is what defines “the essence” of the instrument [10]. This challenge is explored further in Chapter 3 as it describes Cloud Tank's gesture-to-parameter mapping, and it is the reason designing the instrument was inseparable from deciding how it should be played.

The idea that an instrument should be played with the body has a long lineage. Michel Waisvisz's *The Hands*, a pair of sensor gloves first shown in 1984, is among the most cited early examples; Giuseppe Torre and colleagues note that Waisvisz kept its design fixed across two decades of revisions because he wanted it to "become second nature" and to be practiced as an acoustic

instrument is [11]. The same commitment is also shown through the work that followed. Jan Schacher, surveying the lineage from Waisvisz to Laetitia Sonami's *Lady's Glove* and Atau Tanaka's muscle-sensing performances, describes a shared effort to bring the body back into an electronic music that had grown dissociated from it [12].

There is an extensive theoretical basis for positioning the body essential to music-making. Marc Leman and Rolf Inge Godøy argue that the body is central to how musical meaning is made and that listeners make sense of sound partly by relating it to the movements that would produce it [13, 14]. This is important for Cloud Tank. It supports the intuition that movement and sound belong together, and it connects to the legibility raised in Chapter 1. An audience that reads sound through implied movement will read a performer who produces the sound more easily than one seated behind a controller.

Cloud Tank inherits this aspiration but departs from it in one respect. *The Hands, the Lady's Glove*, and the *Mi.Mu* gloves all place sensing hardware on the hands themselves; the playing limbs are instrumented. In the case of Cloud Tank, while the performer wears a headset, the hands are sensed remotely through its cameras, with no glove, marker, or device on them. Within this framework, Cloud Tank is closer to the virtual instruments proposed by Axel Mulder, played by moving through space and existing only through their visual and acoustic representation, with no physical object to hold [15]. Cloud Tank pursues the body-as-instrument goal of the glove tradition, but in a medium where the playing hands do not touch anything physical.

2.2 Liveness and Audience Legibility

If the mapping between gesture and sound defines a digital instrument for its performer, it is also what an audience must read to follow a performance. This is the problem of legibility. Alternatively, challenge of legibility is described as such: any public interface can be characterized by how far a performer's actions, and the effects they produce, are hidden or revealed to a spectator [16]. Within the same framework, Reeves and colleagues identify four strategies of legibility — secretive, where both the action and its effect are concealed; magical, where the effect is seen but its cause is not; suspenseful, where the action is visible but its effect is withheld; and expressive, where action and effect are both revealed, so the audience can follow the performer's interaction in full [16]. Traditional audiovisual performance tends toward the secretive; Cloud Tank is an attempt to design for the expressive, where the movement and its audiovisual result are visible at the same time.

It is also important to explore how spectators perceive digital instruments. Michael Gurevich and Cavan Fyans report that audiences often do not understand the relationship between a performer's gestures and the resulting sound as they would with an acoustic instrument and may not perceive the interaction as instrumental at all [17]. In a related study, Fyans, Gurevich, and Stapleton found that spectators judged skill and error more accurately when they held an accurate mental model of how the instrument worked [18]. Legibility, then, is not a property an instrument has automatically; it depends on whether an audience can build a working picture of how movement produces sound.

Cloud Tank aims to make the relationship between movement and sound visible enough for an audience to read. Chapter 4 returns to this in practice, where I describe which of Cloud Tank's gestures proved legible to audiences and which did not.

2.3 Audio-Reactive Visuals and Beat Synchronization

Cloud Tank is not only an instrument for sound; it drives visual material from that sound, so that the image responds to what is heard. This places it within a tradition of audiovisual performance in which image and sound are produced together. Golan Levin's audiovisual instruments are a central reference: he describes a set of systems for the "real-time and simultaneous performance of dynamic imagery and sound," in which the visual and aural dimensions are tied together by perceptually motivated mappings [19]. Cloud Tank shares this premise but differs in form: where Levin's systems are screen-based and the performer draws the sound into being, Cloud Tank is a body instrument in which authored video layers react to audio that is already playing.

Making visuals respond to sound requires extracting features from the audio as it plays. Two techniques from music information retrieval are relevant here. The first is beat tracking, the estimation of musical pulse from an audio signal. Masataka Goto's real-time system is a standard reference, and he lists the synchronization of real-time computer graphics with music among its intended uses [20]. The second is onset detection, the identification of the moments at which events begin. Bello and colleagues surveyed onset detection in terms of amplitude-envelope and spectral methods [21]. Together, these define how a system can know where it is in musical time, and when something has happened in the sound.

Cloud Tank uses these ideas selectively. Rather than infer the beat from an arbitrary signal, it works from authored tracks whose tempo is known, maintaining a continuous beat phase that visual events can follow. Its reactive behavior, by contrast, draws on feature extraction as Bello describes: each audio bus is analyzed for amplitude, frequency content, and onsets, and these values drive the visual layers.

2.4 Performance in Extended Reality

Cloud Tank belongs to a growing field of musical instruments built for virtual and extended reality. Serafin and colleagues, define a virtual reality musical instrument as one whose interface exists inside an immersive environment rather than as physical hardware, and set out design principles for instruments of this kind [22]. Within that space, systems differ along three axes that matter for this thesis: how the performer provides input, what is performed, and how, if at all, an audience experiences the result. The systems below represent the main approaches, and how Cloud Tank is positioned against them.

System	Type	Performer Input	What is performed	Audience experience
Tribe XR	Commercial (VR DJ)	Hand controllers on replica decks	DJ mixing on virtual professional gear	Livestream / web (Tribe Live), shared rooms
Electronauts	Commercial (VR remix)	Controllers as drumsticks / pads	Remixing pre-produced stems, auto-quantised	Projected visuals; played at festivals
Drile	Research (NIME 2010)	3D manipulation of audiovisual "worms"	Hierarchical live-looping	Staged, legibility-focused performance
Wave	Commercial (concert platform)	Motion-capture suit + face tracking (avatar)	Live avatar concert	In-VR avatar crowd; also 2D livestream
Cloud Tank	This thesis	Markerless hand + body movement	Spatial-audio remix + beat-aligned reactive video	Streamed composite, projected to a co-located audience

Table 1.1. *Comparison of Cloud Tank to similar instruments.*

Tribe XR. Tribe XR is a virtual DJ platform in which the performer operates detailed replicas of professional DJ decks — the Pioneer CDJ-3000 and its mixer. Performer uses the hand controllers to grip, turn, and press as on the physical gear, while the performance is streamed to a web and social audience [24]. Its strength is fidelity: it is unmistakably a DJ set up, and an

audience familiar with the form can read it. But it imports the desk-and-controller paradigm, and with it the limits described in Chapter 1; the body does little that the physical booth would not.

Electronauts. Like *Tribe XR*, Survios's *Electronauts* is controller-based, but it lowers the skill required to produce a good result. The two hand controllers become drumsticks for striking virtual pads and rearranging the stems of pre-produced tracks; its engine then automatically aligns every action to the beat and the key, so the output sounds polished regardless of what the performer does [25]. This makes the system easy to pick up, and it has been performed publicly with projected visuals. The trade-off bears directly on this thesis: if the system will not let the performer produce a wrong result, then skill and mistakes are not visible to an audience. The link between a performer's action and the outcome, the control intimacy discussed in Section 2.1, is intentionally given up.

Drile. The research literature offers a more considered model. Drile, by Berthaut, Desainte-Catherine, and Hachet, is an immersive instrument based on hierarchical live-looping. In Drile, musical processes are represented as three-dimensional audiovisual objects that the performer manipulates in space to build and transform loops [23]. Drile is notable here for two reasons. It treats the visual and the sonic as a single object, as Cloud Tank does; and it comes from a research program explicitly concerned with audience legibility. It is the same program that produced the mechanism-revealing work discussed in Section 2.2. Where it differs from Cloud Tank is in its input: Drile is played by manipulating abstract widgets with 3D interaction techniques rather than through free movement of the body, and it predates consumer hand-tracking.

Wave XR. A fourth approach addresses not the instrument but the audience. Wave XR is a platform for live virtual concerts, in which a performer appears as an avatar performing to an

audience gathered as avatars in a shared virtual venue [26]. It uses motion-capture suits and face tracking. Wave solves the problem of co-presence that the other systems leave aside — it places the performance and the audience inside the same space. However, it abstracts the instrument away almost entirely; the artistry lies in the show and the venue, not in a played object.

Each of these systems do an in-depth implementation of many components shared with Cloud Tank. However, none of these existing systems combine markerless hand and body input, real-time analysis of the audio being performed, beat-aligned video layers, and a streamed image for the audience in one instrument.

2.5 Conclusion

This chapter places Cloud Tank within four bodies of work. The literature on digital musical instruments frames it as an instrument defined by the mapping between gesture and sound (Section 2.1). The work on liveness and legibility explains why that mapping must be readable to an audience, not only to the performer (Section 2.2). The research on audio-reactive visuals and beat synchronization supplies the techniques by which its visuals answer its sound (Section 2.3). The survey of XR performance systems locates the gap it sets out to fill (Section 2.4). Together these establish both the foundations Cloud Tank builds on and the space it occupies. The chapters that follow turn from this context to the system itself, beginning with its design and architecture.

CHAPTER 3

System Design

3.1 Design Philosophy

Cloud Tank originated from the frustration with the dominant frameworks of live audiovisual performance. Traditionally, the performer stands behind a laptop and uses some type of DJ controller to create sound, while the audience watches the projector. The musician and the designer of Mi.Mu wearable musical instruments Imogen Heap states this paradigm clearly: "Pushing buttons and twiddling dials is not very exciting for me or the audience" and she prefers to (with Mi.Mu gloves) "make music on the move, in the flow and more humanly, and more naturally engage with my computer software and technology." Cloud Tank is an attempt to bring Heap's observation inside virtual reality – where hand/body movements are already visible to the system and can be tracked. The technical decisions outlined in the rest of this chapter are all made around three main design commitments: embodied movement, legibility of the performer gestures, and responsiveness of the audio-visuals.

Cloud Tank centers body movements as an instrument. However, there is an important clarification to make: Cloud Tank does not have a built-in body gesture recognition component. Since it runs on Meta Quest Link for streaming purposes, a full-body skeletal tracking was unavailable. Rather than letting that constraint shrink the interaction, the user interface was designed around the body. The broad movements (reaches, sweeps, and leans) that all the performances contain were initially acted out. Each interactive element was then placed where the chosen gesture would land. Hence, the UI was designed and iterated on to prioritize natural full-body movements. Moreover, Cloud Tank utilizes the reliable hand joint-data extensively to

match hand-gestures to audiovisual tracks. Thus, the hands – whether through gesture or UI – are the signal that the system reads; the body is the instrument the audience watches being played.

The second commitment is legibility. Cloud Tank establishes an asymmetry within the performance: performer wears a headset while the audience experiences the rendition through a projector and speakers. Moreover, the performer can see the UI, but the audience is only shown the visuals through the streaming camera. This means that every meaningful interaction must be readable from the outside through the performer's body movements. When an arm lifts or hand sweeps, there should be an audible or visible change. Cloud Tank is designed with that intention: projected audiovisual output is the audience-facing interface, and the performer's body is the input device they can watch being played. This also creates a separation of the artwork and the workspace.

The third commitment is responsiveness. VR space combined with the streaming component naturally amplifies any decoupling between gesture and the result. Even a small delay between a hand sweep and corresponding audio change could spoil the audience perception of the performance. This single constraint shapes several decisions across the rest of the chapter: per-bus audio analysis runs every frame rather than at audio rate; envelope smoothing uses asymmetric attack and release so that visuals snap up on transients; gesture-driven volume changes interpolate over a configurable window rather than jumping to a new value. It is a crucial part of Cloud Tank that the performer feels as if they are a part of a live instrument.

These three commitments – embodied movement, legibility, and responsiveness – drive almost every architectural choice in Cloud Tank. The rest of the chapter describes the resulting system: an architecture overview (§3.2), the audiovisual content lifecycle managed by the layer system (§3.3), the temporal scaffolding provided by the beat driver (§3.4), per-bus audio analysis and

reactive binding (§3.5), embodied input and hand-pose recognition (§3.6), the gesture-to-parameter mapping layer (§3.7), and the evolution of the user interface from flat sliders to embodied control (§3.8).

3.2 Architecture Overview

Cloud Tank is built in Unity 6 for the Meta Quest 3, running over Meta Quest Link during performance so that the system can stream a real-time projection feed to the audience. The codebase is organized around six subsystems that operate concurrently during a performance: an audiovisual content layer, a temporal layer, an audio analysis layer, an embodied input layer, a gesture-to-parameter mapping layer, and a streaming output layer. Each subsystem is described in its own section in the rest of this chapter.

It is also important to introduce the concept of *Pieces* – currently Piece 1, Piece 2, and Piece 3. Each Piece is a complete audiovisual movement with its own visual vocabulary, track list, and gesture mapping, built on the same six subsystems described below. Chapter 4 describes the Pieces themselves; this chapter describes the shared architecture. Additionally, figure 3.2 defines related vocabulary for this chapter.

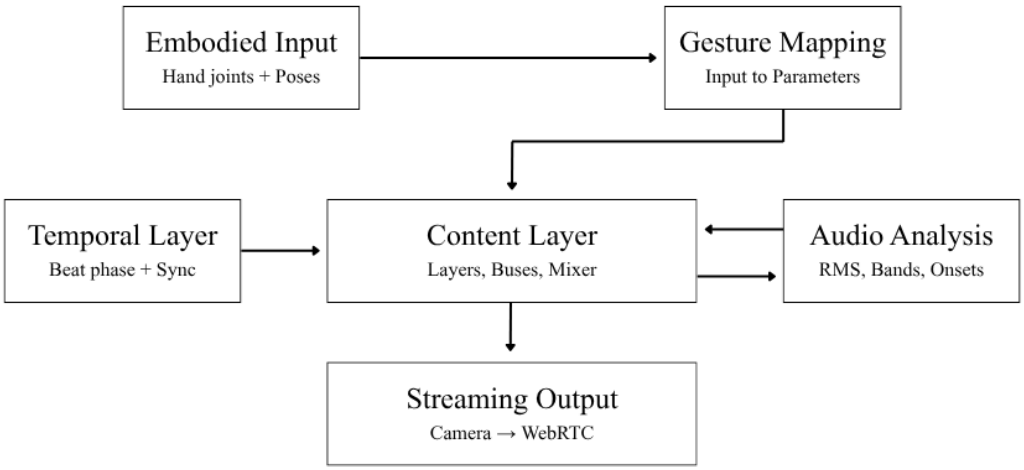

Figure 3.1. *Cloud Tank systems architecture.*

Term	Definition	Location
Track	The authored audiovisual unit. Audio + visual asset + metadata. An entry in the library. Visual assets include prefabs, Unity Particle Systems, and VFX-Graphs.	On disk, in Assets/Resources/Media/Library, and a corresponding *TrackEntry* in the *VideoLayerManager*
Layer	A track spawned into the scene at runtime. The live, playing instance.	In the Unity scene during performance
Prefab	The visual form a track takes. 3D mesh modeled in Maya with a video texture attached. The prefab is what gets instantiated to create a layer.	On disk, in Assets/Prefabs/VideoLayers/Piece 2/, etc.

Figure 3.2. *Systems vocabulary.*

Embodied Input and Gesture Mapping. The embodied input and gesture mapping subsystems form the input chain. Embodied input reads continuous hand-joint data and discrete hand-pose events from the Meta XR framework. Gesture mapping interprets those signals: hand/wrist position inside a slider prompts a bus volume change, a hand movement drags an object's position across audience view, and a recognized hand pose fires its associated audio and visual triggers. This is the section where the performer's body and movements act as system input.

Temporal Layer, Content Layer, and Audio Analysis. The temporal, content, and audio analysis subsystems form the system's core. The content layer manages tracks. At runtime, the content layer spawns the tracks into the scene as layers: live, playing instances routed through four content busses (Music, Percussion, Atmos, FX) and a global Master fader. The temporal layer maintains a continuous beat-phase signal between 0 and 1 that any other component can subscribe to, and it resets on every beat. The audio analysis layer reads the live signal on each bus every frame, computes amplitude, frequency-band energy, and onset events. It feeds those values back into the layers to drive material properties, particle systems, and motion. The

relationship between content and audio analysis is bidirectional and continuous: audio out, visual parameters back in.

Streaming Output. The streaming output subsystem sits on top of the rest. A second camera – positioned in front of the UI to frame the audience-facing composition – renders to a render texture, which is broadcast via Unity Render Streaming over WebRTC at 2560×1440 and 60 frames per second. The performer sees the full workspace, including trigger zones and UI elements; the audience only sees what the streaming camera frames.

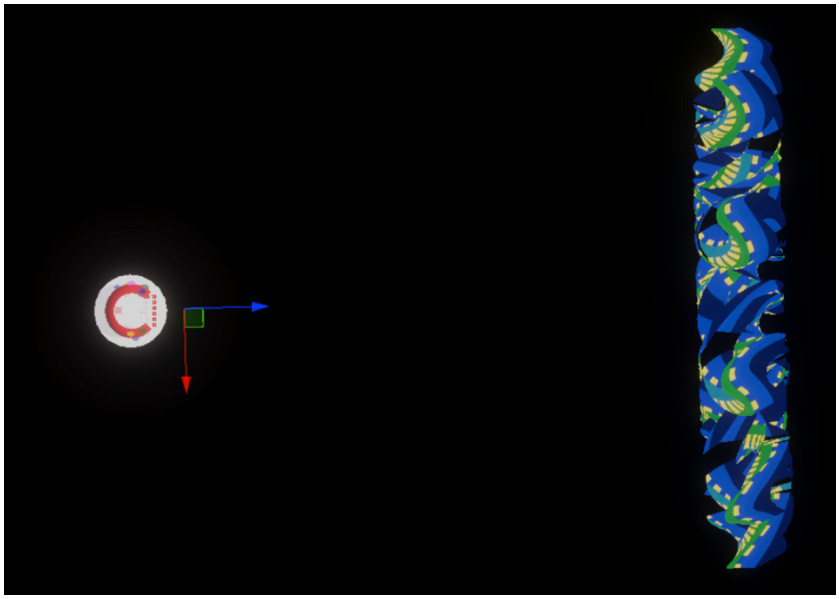

Figure 3.3. *The streaming camera and a layer positioned in its view; the performer's interface is excluded from the framing.*

A piece/performance moves through these layers as follows. The performer's hand movements and gestures produce input. Gesture mapping interprets this input as parameter changes and triggers. Each layer's audio plays into its assigned bus. The audio analysis layer reads per-bus signal, smooths it, and shapes the values into visual parameters that animate the layers. The temporal layer advances the beat phase. The streaming camera renders the result and sends the

frame to display for the audience. The whole pipeline runs every frame at the headset's native refresh rate.

3.3 Audiovisual Content Lifecycle

This section explains how audiovisual content is created, organized, and initiated into the scene during performance. The content layer is the foundation of the architecture: every other subsystem operates on layers. The temporal layer (§3.4) drives layer behavior by beat. The audio analysis layer (§3.5) reads each layer's audio and animates its visual response. The gesture mapping layer (§3.7) decides which layers are spawned, removed, or modified.

Authoring a Track. The design process of each track begins outside of Unity. The audio comes first and is produced in Ableton Live. Pieces 2 and 3 use Ableton Live sample sounds; Piece 1 is assembled from sampled recordings and sound effects. The visual asset is created for the audio: a video is produced in After Effects, edited and timed to match the audio it will accompany, using copyright-free source video and imagery as the base material. The result is a single audiovisual file – a video clip carrying its associated audio – that becomes the playable content of the track. The visual asset is then prepared for Unity. A 3D mesh is modeled in Autodesk Maya, low-poly and UV-unwrapped to receive the video as a surface texture. The mesh is brought into Unity as a prefab (a reusable GameObject) with a Unity Video Player component attached. At runtime, the Video Player streams the video into a render texture, which the prefab's material samples as its surface. A single render texture asset is shared at the project level so that prefabs can be authored and previewed against a common preview stream. However, at runtime, each spawned layer instantiates its own Video Player and renders its own video, so any number of layers can play different content simultaneously without interfering.

It is important to note that not every track follows this video-on-prefab pipeline. Some tracks use a Unity Particle System or VFX Graph as the visual asset instead. In those cases, the track carries no mesh and no video; the visual is generated procedurally and driven by audio analysis (§3.5). The audio half of the track is identical; only the visual form changes. These are still tracks. They simply do not use a prefab.

Finally, beat metadata for the audio is pre-computed offline and saved as a *BPMMetadata ScriptableObject*, which carries the track's BPM and an array of beat times. The complete track is therefore three or four assets, depending on its visual form: an audio clip, a BPMMetadata asset, and either a prefab (with its associated video clip) or a Particle System / VFX Graph asset.

The Library. All the tracks are located on disk in *Assets/Resources/Media/Library*. A single track, such as *DrumBeats*, exists as two parallel files: *DrumBeats.mp4* (the video clip with the audio) and *DrumBeats_BPM* (the ScriptableObject carrying its BPM and beat times). Prefabs are stored separately under *Assets/Prefabs/VideoLayers/*, with one subfolder per Piece. Each Piece has its own set of unique prefabs, except for two shared prefab elements across performances. Thus, the visual vocabulary differs across Pieces, but the *audio/video/metadata* library shares the same structure throughout.

Loading and Registration. When a Piece is loaded, *VideoLayerManager* reads its track list. Each track within that list — a *TrackEntry* — pairs an identifier with a video clip, a *BPMMetadata* asset, and a bus assignment.

Spawning a Layer. When a track is triggered during runtime — by gesture, hand pose, slider interaction, or beat-aligned event — a layer is created. The *VideoLayerManager* instantiates the track’s prefab into the scene under a designated parent transform, places it according to the current layout, and attaches a *LayerAudioHandle* component that records which bus the layer

belongs to. The audio source on the prefab is routed through the *CloudTankMixerRouter* to the *AudioMixerGroup* for its assigned bus. From this point forward, the layer is a live instance: its video plays, its audio mixes through the bus, and any audio-reactive components on the prefab subscribe to the audio analysis layer (§3.5) and the temporal layer (§3.4) for their behavior. The layer remains in the scene until it is explicitly removed by a trigger or the end of the Piece.

Placement and Motion. Each layer's position in the scene is determined by two systems that operate at different timescales. When a layer is first spawned, the *VideoLayerManager* places it at either a fixed position (a slot) or a procedural location chosen from one of three fallback layouts: Grid, Arc, or Ring. A small per-layer jitter is added to prevent identical placements when two layers share a slot. This handles initial seeding only.

Once placed, most layers are driven by a *CameraSpaceMotor* component on the prefab, which continuously repositions them in streaming-camera space. Rather than using world coordinates, the motor expresses position as a fraction of the audience-facing camera's visible area: (0, 0) is the center of the audience's view, (-1, -1) is the bottom-left corner, regardless of where the camera physically sits.

Bus Routing. A bus is a labeled audio channel that groups tracks of similar type, sums their signals, and exposes the result both as a controllable mixer output and as an analysis target. The term comes from analog mixing consoles, where a bus is the wire that carries a summed signal. Cloud Tank uses four content buses — Music, Percussion, Atmos, and FX — plus a global Master that sums them into the final output. Every track is assigned to exactly one of the four content buses, and that assignment determines three things: which AudioMixerGroup it routes through (and therefore which volume slider the performer controls), which *BusAudioAnalyzer*

reads its audio for reactive visuals (§3.5), and which gestures and triggers act on it. A layer's bus is its identity in the system.

3.4 Temporal Structure

In Cloud Tank, each track carries its own pre-computed beat times and reports their progress during runtime, per layer. This section describes how layer-specific information is produced and exposed to the rest of the system, and how multiple playing layers in runtime are kept in sync.

Per-layer Beat Phase. Each layer has a *BeatDriver* component attached alongside its Video Player. This driver references the track's list of beat timestamps — *BPMMetadata* asset — and reads Video Player's current time on every frame. Through these two values, a discrete "*OnBeat*" event fires every time playback reaches the next beat time. Between those events the driver exposes a continuous phase signal between 0 and 1, representing how far the playback has progressed from the last beat to the next. The phase signal is the temporal layer's main product: visual properties animate accordingly and sync with the track. Since the phase is computed from the Video Player's own playback time, it remains accurate even if playback is paused, sped up, or slowed down during runtime.

Multi-layer Synchronization. Layers are initiated at different moments during a performance, and thus, it is hard to achieve phase alignment. Phase alignment refers to when the beat grids of different tracks land on the same wall-clock moments and make up for a fuller sound. The *BeatAlignmentManager* resolves this issue by designating one layer as the master and aligning the layers spawned after to the master's beat grid. When the alignment is enabled and a new track is triggered, the manager computes the time until the master's next beat, finds a beat in the new track's metadata whose offset would land on that moment, and schedules the new layer to start playback at the corresponding video offset. As a result, the new layer's audio starts on the

master's downbeat. After spawning, the new layer's *BeatDriver* continues to follow its own video time, but its beat grid now agrees with the master's. Beat alignment is optional, and the performer can enable/disable it anytime through a runtime toggle. It depends on the performer's confidence with the tracks and preference.

Beat Events and the Global Bus. Discrete beat events are also published to a project-wide singleton, *GlobalBeatBus*, which any component can subscribe to without holding a reference to a specific layer. This is used by visual elements that pulse/move on the beat regardless of which layer triggered it. The bus does not track phase; phase remains a per-layer concept read from its own driver.

3.5 Per-Bus Audio Analysis and Reactive Binding

Cloud Tank's commitment to responsiveness (§3.1) is important as the system needs to extract usable signal from the audio fast enough to act on it within a frame. This section describes how that signal is produced and how visual properties subscribe to it.

Per-Bus Analysis. A single *BusAudioAnalyzer* runs in the scene and is responsible for all four content buses — Atmos, Percussion, Music, and FX. Each frame, the audio analyzer goes through every audio source assigned to each bus and reads the recent samples directly from the source (~23 ms at 1024 samples). It computes an RMS amplitude, a frequency-band decomposition into bass, mid, and high, and a spectral flux value used for onset detection. These sources are registered and unregistered automatically as layers spawn, so the performer never configures audio analysis manually.

Term	Explanation
RMS Amplitude	Overall loudness on the bus.
Band Decomposition	Decomposes the signal into bass, mid, and high bands. Bass peaks on kick drums, mid-energy on melodic instruments, high-energy on hi-hats and sibilance.
Spectral Flux	How much the spectrum (frequency content of a sound) changed since the last frame. Spikes on transients (drum hits, synth onsets, sample triggers). Basis for onset detection.

Table 1.2. *Dictionary*

Asymmetric Smoothing. Raw, frame-to-frame values from any of the signals mentioned above are too jittery to drive the visuals. Hence, the analyzer applies asymmetric exponential smoothing to each: a short time constant (~50 ms) lets the smoothed value rise quickly toward a peak, and a longer release time constant (~300 ms) lets it decay gently. Visually, this means a single drum kick produces a fast brightening followed by an eventual fade, not a one-frame flash and immediate darkness. The smoothing values can be adjusted per track for visual preference.

Onset Detection. An onset is the starting moment of a discrete sound event; onsets are derived from spectral flux. Sustained sound produces low flux; new events produce sudden spikes. The analyzer maintains a baseline of recent flux values and fires an onset when two conditions are met: the current frame's flux exceeds the baseline by a configurable multiplier (default 2.5×), and it exceeds an absolute noise floor that prevents false triggers during near silence.

Reactive Binding. A binding is the connection between an audio signal and a visual property; in Cloud Tank, *AudioReactiveBinding* manages the said connection. Any *GameObject* that responds to audio carries one or more bindings. Each of these bindings declare a source — a bus

and a signal, such as Percussion and Smoothed RMS —— and a target, a visual property to drive. The target can be a material color or property, a transform scale, a value inside a VFX Graph, or any public method that accepts a number.

Between source and target exists the shaping stage. Shaping is where each binding tunes how the signal input translates into an output. This is important as the raw audio values are not directly useful as visual parameters: spectral band energies are small numbers — often below 0.01 — while smoothed RMS is closer to 0.3, and visuals generally expect values between 0 and 1. The shaping stage solves this in two steps. First, the raw value is multiplied by an *inputScale* to bring it into the 0–1 range. The scaled value is then passed through an *AnimationCurve*, which remaps the input range to the output range using an arbitrary curve drawn in the inspector. Through this process, the same audio signal can produce noticeably different visual behavior. For instance, a linear ramp tracks amplitude flatly, whereas a concave-up only snaps on peaks.

Furthermore, a single layer typically carries several bindings, and each are tuned independently. In practice, for instance, a *Capsule* prefab runs four bindings at once. It maps bass value of a track to its horizontal movement across the streaming camera's frame and maps the mid-energy value to vertical movement. It connects a track's smooth RMS value with its prefab's scale, and the master beat phase value to prefab's spinning movement. Each of these bindings has its own input scale, response curve, and target on the *CameraSpaceMotor*. The bass binding, in this case, uses a steep curve so the capsule snaps sharply on each kick drum. The RMS binding uses a smoother curve, so the capsule's scale does not change with sharp pulses. The beat-phase binding uses a linear ramp, so the spin movement stays locked to tempo of the track. Together, these four bindings shape the same prefab through four different audio channels, and the composite

behavior is what makes the layer respond to the music. Each layer functions as its own sound visualizer, with its character defined by the tuning of each binding's curve in the inspector.

3.6 Embodied Input and Hand-Pose Recognition

Cloud Tank is designed with embodied movement as a main component (§3.1). This section explores how the embodied movement is captured as input — including the continuous (hand position, finger curl, wrist orientation) and the discrete data (recognized hand poses). It also explains the design constraints that shaped such configuration.

The Link Constraint. Cloud Tank runs over Meta Quest Link during performance so that the headset can stream a real-time video feed to a projector (§3.2). However, Meta's full-body skeletal tracking, which would otherwise expose torso, shoulder, and hip joints, is not available in this streaming configuration. The only reliable joint data comes from the hands and the head. Rather than working against this constraint with imprecise estimations, the system uses what is available reliably and designs the rest of the interaction around it (§3.1).

Hand-joint Tracking. The Meta XR component *OVRSkeleton* provides continuous hand input: Cloud Tank reads the position and rotation tracking values of each hand during runtime. The wrist joint is used as position reference for sweep and motion gestures, and the finger joints are used for trigger and pose detection. Each frame, gesture-mapping components (§3.7) read the joint positions from the skeleton and translate them into parameter changes or trigger events. As the joint data is in the world space, it is directly compared against the positions of the UI elements without coordinate system conversion.

Hand-pose Recognition. Discrete hand input is read through Meta XR's *Active State Framework*, a publish/subscribe system for hand poses. Each hand pose is defined by a combination of two recognizers, *ShapeRecognizerActiveState* and

TransformRecognizerActiveState. The former is used to check finger curl and spread against a target shape; the latter checks the wrist's orientation in space. When both conditions are met, the framework's *ActiveStateSelector* fires a Unity event that declares the pose has been entered; when either condition fails, it fires a corresponding exit event. Cloud Tank wires these events directly to the audio and visual triggers described in §3.7.

Two distinct poses are recognized in the final performances – Paper (open hand, palm forward) and Rock (closed fist). Each pose functions independently on the left and right hand; so, Paper-left and Paper-right are separate triggers. Earlier prototypes also recognized various other poses, but they were cut from the final performances during rehearsal. Additional pose vocabulary tended to fire false positives during natural movement, and the two-pose vocabulary proved sufficient for the performance's expressive needs.

3.7 Gesture-to-Parameter Mapping

The previous sections described what the system reads (§3.6) and what it produces (§3.3–§3.5). This section explains the portion in between: how hand input, once captured, becomes audio and visual change. The mapping layer is where the performance happens. Each interactive element in Cloud Tank — sliders, trigger boxes, hand poses — is a mapping from a gesture to a parameter or event in the underlying system. There are five families of mappings, described in turn.

Bus Volume Sliders. The most direct mapping in Cloud Tank is the bus volume slider. There are six trigger boxes in front of the UI labeled Master, Atmos, Music, Percussion, Reverb, and FX. Only Master, Atmos, Music, Percussion are active in the final performances, as noted in §3.3. Each active box is a *MixerBusSliderTrigger*: it includes a collider associated with a bus, a hand filter (left, right, or either), and a configurable volume range (default 0.05 to 1.0). When a tracked hand enters the box, the slider becomes active. Within the slider collider volume, the

hand's local X position is read every frame, smoothed with an exponential filter to remove tracking jitter, mapped to the volume range, and sent to the *CloudTankMixerRouter* as a new bus level. A visible marker inside the box slides horizontally to reflect the current value, so the performer can see the volume position.

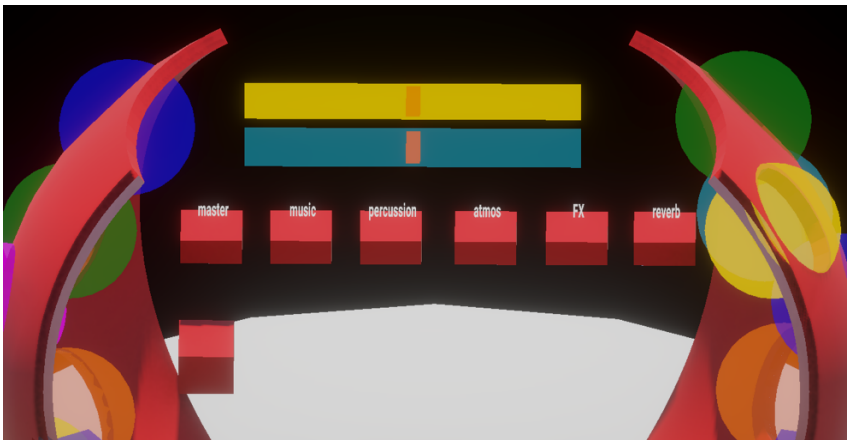

Figure 3.4. *Bus, motion, and volume sliders.*

Continuous Motion Control. The same slider mechanic is used to drive non-audio parameters via *MotionSliderTrigger*. In the documented performances, this is wired to the Percussion and Music buses and shows as another slider on top of the volume one (Figure 3.4). It emits a normalized float that is forwarded to a *MotionBroadcaster*. The broadcaster reads that value once per frame and pushes it into every active layer on a chosen bus, calling configurable *CameraSpaceMotor* setters. The slider itself widens the horizontal and vertical sweep range of every layer on the bus at once, taking each from near-zero movement at the slider's left edge to nearly the full streaming frame at the right. Since the broadcaster pulls active layers every frame, layers spawned after the slider was configured pick up the vertical/horizontal range values automatically. Similarly, whenever the performer moves the motion slider for a bus, it controls whatever Percussion — or Music — layers happen to be alive at that moment during the performance.

Manual Mode. Manual mode is the most direct mapping of the system: instead of reading hand position relative to a slider, the system takes the hand's wrist position and uses it directly to position the layers within the frame. It is enabled/disabled through a toggle box during runtime. Once the manual mode is active, every frame, *HandMotorController* reads the wrist position from *OVRSkeleton*, computes the frame-to-frame world space delta (in meters), converts it to viewport units (~30 cm of hand motion corresponds to the full streaming frame), and overrides the *baseScreenPosition* of every *CameraSpaceMotor* on the target bus. When the manual mode is enabled, the motor's autonomous Lissajous orbit (§3.3) is significantly damped. Consequently, the performer's hand motion and position become the dominant signal. When manual mode is toggled off, the visuals return to their resting position over half a second and the Lissajous orbit resumes. Thus, this mode turns the music-bus visuals into something the performer drags across the audience's view by hand; it is a deliberate moment of authorship over the visual output.

Hand-pose Triggers. Recognized hand poses (§3.6) are wired into the system through two scripts that mirror the audio/visual split: *PoseAudioController* fires audio events and *PoseVFXController* fires VFX Graph events. When a pose enters, the corresponding start methods are invoked; when it exits, the end methods fade the audio and stop the visual. In the final performances, Paper triggers a paired audio clip and VFX graph as a single composite element. Rock triggers an audio clip alone, with no paired visual. Each pose is wired per hand, so Paper-left and Paper-right can be mapped to entirely different audiovisual content.

Beat-aligned and Runtime Triggers. During the performance, a few smaller mappings handle one-shot performance actions that are not tied to a pose or slider. *BeatAlignedTrackTrigger* initiates a track at the next beat boundary rather than immediately, using the master beat phase from §3.4. *TrackFadeOutTrigger* does the inverse, fading an active layer to silence on contact.

BeatAlignmentToggleTrigger flips the alignment mode on or off (§3.4). *ClipPlaneTrigger* enables or disables a global clipping plane that hides part of the visual stage. *VideoEchoTrigger* increments a video-echo effect. Each is a small, single-purpose interaction. Together, they add another layer to the improvisation of the performance.

3.8 The Performer's Interface

The interface the performer faces is itself part of the instrument, and it is built around the body rather than laid out as a flat panel. Figure 3.5 shows the performer's view. A row of six trigger boxes — labeled Master, Music, Percussion, Atmos, FX, and Reverb — sits across the lower-middle of the field; as noted in §3.3, only Master, Music, Percussion, and Atmos are active in the final performances. Above them are the volume and motion sliders (§3.7), with a marker that reflects the current slider value, and the visuals spawn and move in the space the layout frames. The placement of these elements is deliberate. Earlier versions of Cloud Tank arranged the controls as flat panels, operated much like an on-screen mixer. Over development, the layout was rebuilt around the movements each gesture required: each element was positioned where the performer's hand would naturally land when making the gesture mapped to it, following the choreography-first method described in §3.1. The interface is therefore fitted to the body that plays it. It is also private to the performer — as established in §3.2, the performer sees this full workspace, while the audience sees only the visuals framed by the streaming camera. The interface never reaches the projection.

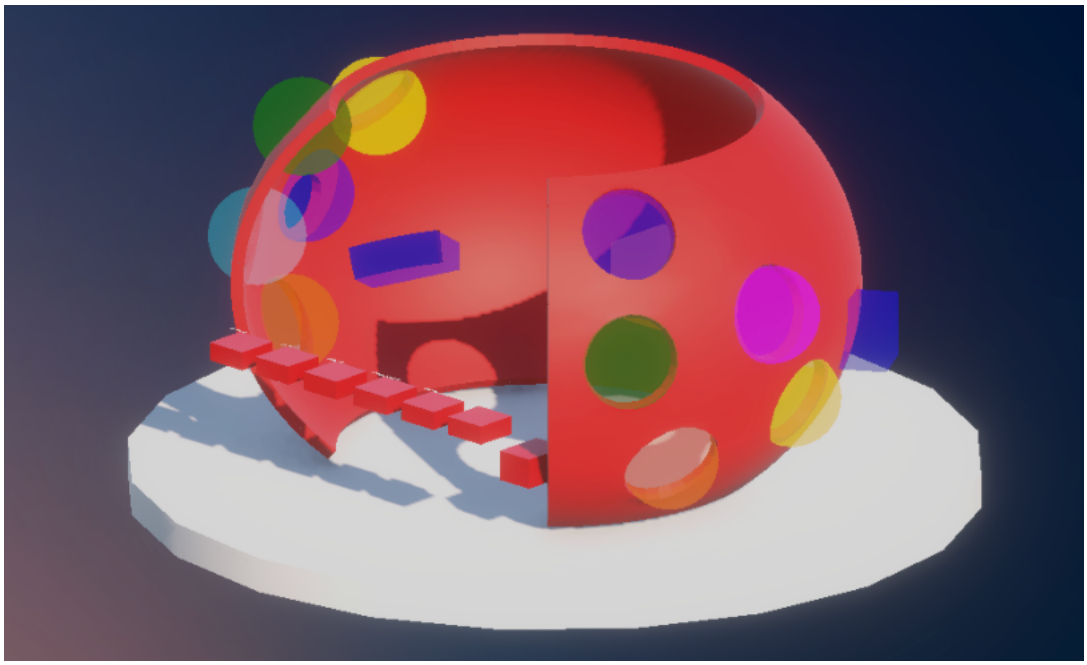

Figure 3.5. *The performer's interface.*

Chapter Summary

Chapter 3 has described Cloud Tank's six subsystems — content, temporal, audio analysis, embodied input, gesture mapping, and streaming output — and the architectural decisions that follow from the three commitments laid out in §3.1. The next chapter turns from the system to the performance: how the three Pieces use this architecture, what they look and sound like, and how the system holds up under the demands of live performance.

CHAPTER 4

Practice and Documentation

The previous chapter described Cloud Tank as a system; this chapter turns to what it became when it was played. The three Pieces are the repertoire produced through the choreography-first method introduced in Chapter 1, each a complete audiovisual movement built on the shared architecture, and together the evidence that Cloud Tank is an instrument rather than only an artifact. This chapter profiles the three Pieces (§4.1), then describes how they are shaped in performance and what playing them in front of an audience revealed (§4.2) — observations that Chapter 5 takes up in its evaluation.

4.1 The Three Pieces

The three Pieces are built on the same six subsystems, but each is a distinct movement with its own visual vocabulary, sound, and gesture emphasis. Each has its own prefabs and its own VFX or particle graphs; two prefab elements recur across all three, reworked slightly each time, so that even the shared forms read differently from Piece to Piece (Figure 4.4). Together the three move from calm to dense: Piece 1 is ambient and sparse, Piece 2 builds from slow to percussive, and Piece 3 is the most beat-driven of the set.

Piece 1. First Piece is ambient and atmospheric, the calmest of the three and the one with the least percussion. Its palette is soft — blues, greens, and pastels — and its visual material is built mostly from video-on-mesh prefabs (§3.3), with a single VFX graph held back for one gesture. It stays sparse, with at most three layers alive at once, weighted toward the Atmos and Music buses. Piece 1 is played mainly through the bus volume sliders (§3.7), and its signature is the Paper pose (§3.6), which brings in its one VFX graph. From the outside, the audience sees me

swaying with the music, easing the volume sliders, and opening a hand into Paper to call in the visual — a gentler, less percussive-heavy body than the Pieces that follow. Its defining moment is that single Paper gesture lifting the VFX layer over the ambient bed, shown in Figure 4.1: the open hand and the burst of particles it produces are visible in the same frame.

Figure 4.1. *Piece 1 — the signature Paper gesture: an open hand at the headset and the particle burst it triggers, in the pastel palette.*

Piece 2. Second Piece is the arc of the set: it opens slow and atmospheric and builds into percussion, mixing slower tracks with beat-driven ones. Its palette is brighter and more textured, in oranges, blues, and yellows, and its layers combine video-on-mesh prefabs with VFX graphs. The Music and Percussion buses carry most of the weight, and it grows considerably fuller than Piece 1, with a few hard kicks to the FX trigger box. Its signature is manual mode (§3.7): I take the VFX-graph trails in my hand and drag them across the audience's view (Figure 4.2). The motion slider is the most prominent control in this Piece, widening and narrowing the sweep of the layers as the energy rises, with the volume sliders worked alongside it. The defining moment is that manual-mode passage — reaching into the frame and pulling the trails across it by hand.

Figure 4.2. *Piece 2 — manual mode: dragging the VFX-graph trails across the frame by hand, in the Piece's bright textures.*

Piece 3. Third Piece is the most percussive and the densest, and the one that leans hardest on particle graphs (Figure 4.3). Where Piece 2 introduced VFX graphs, Piece 3 pushes that experiment further: its signature visual form is the particle graph, which moves and bursts with the sound (§3.5), in bright cyan, magenta, and light blue. All four buses are in use, and it is the fullest of the three. Its signature gesture is the Rock pose (§3.6), which drops a bass beat, and from the outside the body here is more percussive — closer to drum-hitting than the swaying of Piece 1. The defining moment is that Rock gesture landing the bass as the particles scatter across the frame.

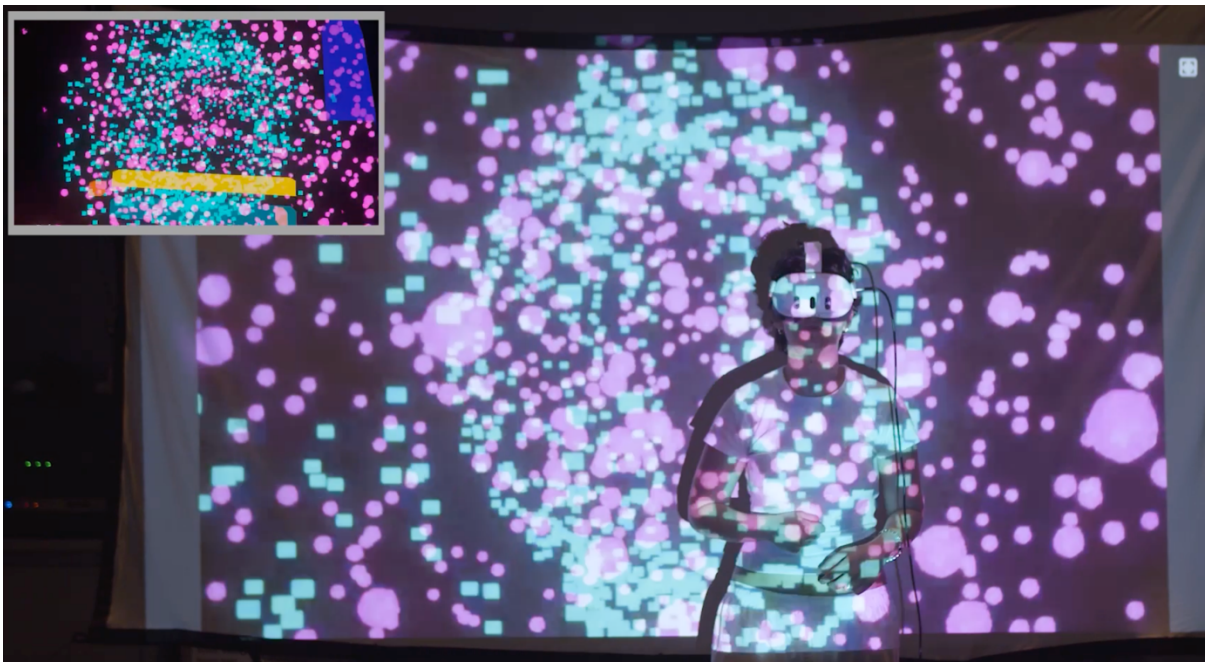

Figure 4.3. *Piece 3 — particle graphs reacting to the percussion, in cyan and magenta.*

Piece	Character	Dominant visual form	Signature gesture	Buses emphasized
Piece 1	Ambient, sparse, slow	Video-on-mesh prefabs; pastel	Paper pose → VFX graph; volume sliders	Atmos, Music
Piece 2	Slow-to-percussive build; bright	Prefabs + VFX graphs	Manual mode + motion slider	Music, Percussion (FX accents)
Piece 3	Most beat-heavy, densest	Particle graphs + VFX	Rock pose → bass; particle bursts	All four

Table 4.1. *The three Pieces at a glance.*

Figure 4.4. *A recurring capsule prefab across the three Pieces — recolored and retextured for each (Piece 1, Piece 2, Piece 3, left to right).*

4.2 Performance and Observations

Cloud Tank has been performed throughout its development. From the end of last summer through May, I performed it constantly but informally: regularly once a week for my advisor, as a way of gathering feedback and deciding what to build next. More formally, I performed at both the Fall and Winter *Technigala*, Dartmouth's end-of-term showcase of computer science and digital arts projects, and in two in-class performances. The system was therefore shaped in front of an audience from early on, rather than built in isolation and shown once at the end.

Each Piece begins from a library and an intended order, a rough idea of which tracks come first and how the Piece should build, but a performance differs every time. In practice, performing Cloud Tank is a combination of structure and improvisation: the library and the arc are fixed, while the timing of the triggers, the layering, and the movement are decided live. A Piece generally starts sparse and fills as it goes, and I shape that build through the choice between beat-aligned and immediate triggers (§3.4, §3.7).

Performing the instrument repeatedly is also what taught me which parts of it worked. The most important lesson was about legibility: large, whole-body motions read to the audience, while small or precise hand movements did not. Immediate audio and visual triggers — particularly on the FX bus — improved legibility further, because the audience could see a gesture and hear or see its result at the same moment, with no delay to break the connection. Manual mode landed especially well: dragging the visuals across the frame by hand was the most readable gesture in the system, and the one that felt most like playing. The finer hand poses, by contrast, misfired during natural movement and were cut from the final performances (§3.6). These observations are the basis for the evaluation in Chapter 5.

CHAPTER 5

Discussion and Conclusion

This chapter evaluates Cloud Tank against the three commitments that shaped it — embodied movement, legibility, and responsiveness — and through them, the research questions set out in §1.3. The evaluation draws on the performances described in Chapter 4. It is not a formal study; it is my own assessment as the designer and performer, based on building the instrument, learning to play it, and watching audiences respond to it. I then discuss the system's limitations and the directions the work could take next.

5.1 Evaluation

Embodied movement. The first commitment was embodied movement: the instrument should be played primarily through motion, without a complex UI. On this point, Cloud Tank succeeds adequately. Audiovisual control is mapped onto hand movements and gestures, and the three Pieces serve as evidence. Each Piece is performed as choreography, from memory, without menus. However, it is important to acknowledge the challenge: solidifying the movements to the music was difficult, and the choreography of each Piece had to be iterated many times before I was comfortable performing it. That difficulty, though, is not a failure but a property of the system. As argued in §1.2, an instrument is something with a learning curve, practiced with the body over time; the fact that Cloud Tank took real practice to play is itself evidence that it is an instrument rather than a tool.

Legibility. Legibility was the most challenging commitment. Two things stand out. First, before the streaming subsystem worked, the projected output was simply my own view: me looking at the UI and the resulting visuals. The audience, therefore, experienced mostly the interface rather

than the artwork — the secretive case in Reeves's terms (§2.2), the opposite of what I intended. The dedicated streaming camera (§3.2), which frames only the audience-facing composition and hides the UI, was the fix that resolved the problem in the final performances. Second, even with the visuals framed correctly, there was a clear legibility threshold. The broad hand sweeps used only to trigger the audiovisual tracks were not enough to create a convincing performance for the audience. After my performances, there would always be some amount of confusion I needed to clear up about what my hand movements meant. However, once I wired the hand gestures and the slider-bus logic, the feedback became more positive, and there has been less confusion. Immediate audio and visual triggers, especially on the FX bus, also helped, as the audience saw the gesture and its result at the same moment. Legibility was achievable then — but only above a threshold, and only through the body. This is the choreography-first method justified after the fact: the system is legible because it is performed with large, readable movement, not because the visuals alone communicate.

Responsiveness. The system needs to react to movement quickly enough that the performance feels continuous, and for Cloud Tank, it does. Per-bus analysis runs every frame, smoothing shapes the response, and gesture-driven changes interpolate rather than jump (§3.5), so the system reacts within a frame. For instance, in manual mode, dragging the visuals across the frame by hand felt continuous and direct, and it was the gesture that most resembled playing. In the final performances, I did not encounter the decoupling between gesture and result that would have broken the sense of a live instrument.

5.2 Limitations

The clearest limitation is that Cloud Tank has one performer, who is also its designer. Everything in this thesis comes from a single person building, playing, and observing the instrument; there is

no formal user study, and the claims about legibility are my observations of audience response in the room, not measured data. A study with other performers, and with audience surveys would test whether what I found holds beyond myself.

A second limitation is inherent to the choice of VR over AR: I cannot see the audience while performing. This was strange at first, and it removes the in-the-moment feedback a performer normally has. It is also, in a way, a benefit — without the audience in view, I can see the visual output clearly and focus on my craft — but it is a real cost, and one that the medium imposes rather than one I chose.

Finally, the instrument's expressive vocabulary is narrower than I first intended. The finer hand poses misfired during natural movement and were cut, leaving the two-pose vocabulary of Paper and Rock (§3.6), and the Link constraint meant the interaction had to be built around the hands and head rather than the full body (§3.6). Both were reasonable trades for reliability, but both bound what the instrument can express.

5.3 Future Work

Several of these limitations point directly to future work. Running Cloud Tank in augmented reality or passthrough, rather than fully enclosed VR, would let the performer see the room and recover the audience feedback that the current setup loses. Running it untethered and on-device, rather than over Link, would remove the constraint that disabled full-body tracking, opening the door to torso and limb movement and a larger pose vocabulary. The instrument could also be extended to more than one performer. And the user interface itself, which over the course of development moved from flat sliders toward embodied control, deserves a fuller account than this thesis gives it; tracing that evolution is work I would like to take up next.

5.4 Conclusion

This thesis set out to build a live audiovisual instrument in which the performer's body, rather than a control surface, becomes the visible site of the performance. Cloud Tank is the result, and it makes three contributions. As a system, it is a virtual reality architecture for embodied audiovisual remixing, combining per-bus audio analysis, beat-aligned reactive visuals, and gesture mapping into a single streamed frame. As a practice, it is a way of performing, developed through the choreography-first method and realized in the three Pieces. And as a set of findings, it is what designing and performing the instrument revealed — above all, that legibility in this medium is achieved through large, readable, whole-body movement, and that an instrument worth playing is one that takes practice to play.

Extended reality is, today, roughly where the theremin was a century ago: a young and awkward medium that is difficult to play and easy to dismiss. Cloud Tank does not resolve that. But it is an attempt to take the medium seriously as an instrument — to treat the tracked body not as a novelty but as the thing being played — and to make a performance the audience can watch being made.

Bibliography

[1] Caleb Stuart. 2003. The object of performance: Aural performativity in contemporary laptop music. Contemp. Music Rev. 22, 4 (2003), 59–65.

[2] Oliver Bown, Renick Bell, and Adam Parkinson. 2014. Examining the perception of liveness and activity in laptop music: Listeners' inference about what the performer is doing from the audio alone. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '14). London, UK, 161–164. DOI: https://doi.org/10.5281/zenodo.1178722

[3] Stefano Fasciani and Jackson Goode. 2021. 20 NIMEs: Twenty years of New Interfaces for Musical Expression. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '21). Retrieved June 3, 2026 from https://nime.pubpub.org/pub/20nimes

[4] James Pallister. 2014. These gloves will "change the way we make music," says Imogen Heap. Dezeen. (March 2014). Retrieved June 3, 2026 from https://www.dezeen.com/2014/03/31/imogen-heap-gloves-mini-frontiers-movie/

[5] Alex McLean, Julian Rohrhuber, and Nick Collins. 2014. Editors' notes. *Computer Music J.* 38, 1 (Spring 2014), 4–5. DOI: https://doi.org/10.1162/COMJ_e_00225

[6] Albert Glinsky. 2000. Theremin: Ether Music and Espionage. University of Illinois Press, Urbana, IL.

[7] Trevor Pinch and Frank Trocco. 2002. Analog Days: The Invention and Impact of the Moog Synthesizer. Harvard University Press, Cambridge, MA.

[8] David Wessel and Matthew Wright. 2002. Problems and prospects for intimate musical control of computers. Computer Music J. 26, 3 (2002), 11–22. DOI: https://doi.org/10.1162/014892602320582945

[9] Eduardo R. Miranda and Marcelo M. Wanderley. 2006. New Digital Musical Instruments: Control and Interaction Beyond the Keyboard. A-R Editions, Middleton, WI.

[10] Andy Hunt, Marcelo M. Wanderley, and Matthew Paradis. 2003. The importance of parameter mapping in electronic instrument design. J. New Music Res. 32, 4 (2003), 429–440. DOI: https://doi.org/10.1076/jnmr.32.4.429.18853

[11] Giuseppe Torre, Kristina Andersen, and Frank Baldé. 2016. The Hands: The making of a digital musical instrument. Computer Music J. 40, 2 (2016), 22–34. (
[12] Jan Schacher. 2022. Capture and express, question and understand: Gloves in gestural electronic music performance. Wearable Technol. 3 (2022), e5. DOI: https://doi.org/10.1017/wtc.2022.3
[13] Marc Leman. 2008. Embodied Music Cognition and Mediation Technology. MIT Press, Cambridge, MA.
[14] Rolf Inge Godøy and Marc Leman (Eds.). 2010. Musical Gestures: Sound, Movement, and Meaning. Routledge, New York, NY.
[15] Axel G. E. Mulder. 1998. Design of Virtual Three-Dimensional Instruments for Sound Control. Ph.D. Dissertation. Simon Fraser University, Burnaby, BC, Canada.
[16] Stuart Reeves, Steve Benford, Claire O'Malley, and Mike Fraser. 2005. Designing the spectator experience. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05). ACM, New York, NY, 741–750. DOI: https://doi.org/10.1145/1054972.1055074
[17] Michael Gurevich and A. Cavan Fyans. 2011. Digital musical interactions: Performer–system relationships and their perception by spectators. Organised Sound 16, 2 (2011), 166–175. DOI: https://doi.org/10.1017/S1355771811000112
[18] A. Cavan Fyans, Michael Gurevich, and Paul Stapleton. 2010. Examining the spectator experience. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '10). Sydney, Australia, 451–454.
[19] Golan Levin. 2000. Painterly Interfaces for Audiovisual Performance. M.S. Thesis. Massachusetts Institute of Technology, Cambridge, MA.
[20] Masataka Goto. 2001. An audio-based real-time beat tracking system for music with or without drum-sounds. J. New Music Res. 30, 2 (2001), 159–171.
[21] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B. Sandler. 2005. A tutorial on onset detection in music signals. IEEE Trans. Speech Audio Process. 13, 5 (2005), 1035–1047. DOI: https://doi.org/10.1109/TSA.2005.851998
[22] Stefania Serafin, Cumhur Erkut, Juraj Kojs, Niels C. Nilsson, and Rolf Nordahl. 2016. Virtual reality musical instruments: State of the art, design principles, and future directions. Computer Music J. 40, 3 (2016), 22–40.

[23] Florent Berthaut, Myriam Desainte-Catherine, and Martin Hachet. 2010. Drile: An immersive environment for hierarchical live-looping. In Proceedings of the International Conference on New Interfaces for Musical Expression (NIME '10). Sydney, Australia, 192–197. DOI: https://doi.org/10.5281/zenodo.1177721

[24] Tribe XR. 2026. Tribe XR DJ Academy. Retrieved June 3, 2026 from https://www.tribexr.com/

[25] Survios. 2018. Electronauts. Retrieved June 3, 2026 from https://survios.com/electronauts/

[26] Wave XR. Wave (formerly TheWaveVR). Retrieved June 3, 2026 from https://wavexr.com/

Appendix

Performance Documentation

Video recordings of the three Pieces are available at the links below.

Piece 1: https://www.youtube.com/watch?v=jLPIfmmiHnA

Piece 2: https://www.youtube.com/watch?v=Ko7qSA5g4o4

Piece 3: https://www.youtube.com/watch?v=OCaRtDuoOV8