Gesture-based Collaborative Virtual Reality Performance in *Carillon*

Rob Hamilton
Rensselaer Polytechnic Institute
hamilr4@rpi.edu

Chris Platz
Savannah College of Art and Design
leonplatz@gmail.com

ABSTRACT

Within immersive computer-based rendered environments, the control of virtual musical instruments and sound-making entities demands new compositional frameworks, interaction models and mapping schemata for composer and performer alike. One set of strategies focuses compositional attention on crossmodal and multimodal interaction schema, coupling physical real-world gesture to the action and motion of virtual entities, themselves driving the creation and control of procedurally-generated musical sound. This paper explores the interaction design and compositional processes engaged in the creation of Carillon, a musical composition and interactive performance environment focused around a multiplayer collaboratively-controlled virtual instrument and presented using head-mounted displays (HMD) and gesture-based hand tracking.

1. INTRODUCTION

For as long as computers have been purposed as real-time generators and controllers of musical sound, composers and performers have researched methods and mappings through which performative gesture can intuitively drive computer-based instruments and procedures [1]. Traditional instrumental performance practices, developed over centuries of musical evolution, have by their very nature been based in the physical control of physical interactive systems. While digital music systems have freed musical generation and control from the constraints of physical interaction, there exists a strong desire amongst contemporary composers, performers and researchers to develop idiomatic performance mappings linking musicians' physical gestures to computer-generated music systems [2].

As commercial high-resolution virtual reality systems become commonplace components of an already digitally immersed 21st Century culture, a natural reaction for composers seeking to use rendered space for musical exploration is to look to existing instrumental performance paradigms and gestural mappings to guide interaction models for musical control in VR space. In that light, digital artists and researchers have been exploring modes of crossmodal interaction that allow users to control and manipulate objects in a rendered reality using interfaces and physical interaction models based in their own physical realities.

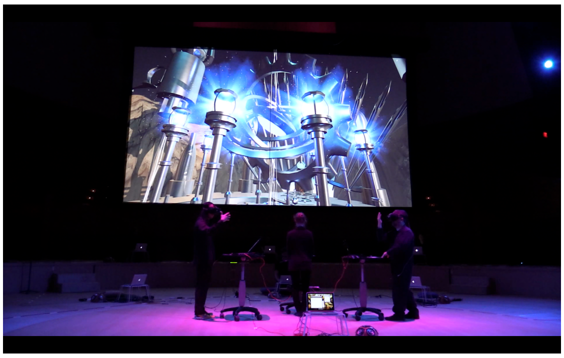

Figure 1. Live performance of *Carillon* featuring the Stanford Laptop Orchestra at Stanford University, May 30, 2015.

2. OVERVIEW

Carillon is a mixed-reality musical performance work composed and designed to marry gesture in physical space, avatar and structure motion and action in virtual space, and the procedural musical sonification and spatialization of their resultant data streams within a multi-channel sound space. Premiered on May 30, 2015 at Stanford University's Bing Concert Hall by the Stanford Laptop Orchestra [3], *Carillon* allows performers in VR space to interact with components of a giant virtual carillon across the network, controlling the motion and rotation of in-engine actors that themselves generate sound and music.

Visually, *Carillon* incorporates rendered three-dimensional imagery both projected on a large display for audiences to view as well as presented stereoscopically in a Head Mounted Display (HMD). Performers wearing Oculus Rift head-mounted displays view the central carillon instrument from a virtual location atop a central platform, overlooking the main set of rotating rings. Each performer's viewpoint aligns with one of three avatars standing in the virtual scene. Using Leap Motion devices attached to each Oculus Rift headset, each performer's hand motion, rotation and position are mapped to the motion, rotation and position of the hands of their respective avatar, creating a strong sense of presence in the scene. Floating in front of each performer is a small representation of the main set of rings that can be "activated" by touching one or more rings. A hand-swipe gesture is used to expand or collapse the set of rings, and each ring is visually highlighted with a distinct red color change when activated

Sound is generated in *Carillon* procedurally, by mapping data from the environment to parameters of various sound models created within Pure Data. The parameters of motion of each ring - speed of rotation in three-dimensional coordinate space - are mapped to parameters of a model

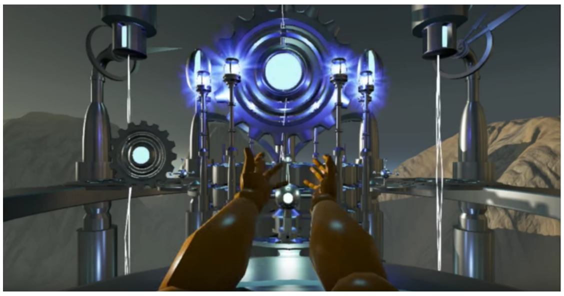

Figure 2. Avatar hands, the large central set of rings and the smaller ”HUD” rings to the avatar’s left.

of a ”Risset bell” (formed using additive synthesis) [4] in Pure Data [5]. Open Sound Control [6] is used to export data from the Unreal Engine 4 [1] and pass that data to . Rules governing the speed of rotation for each ring mediate user gesture and intentionally remove each performer from having precise control over the performance’s output. Similarly, by allowing multiple performers within the environment to interact with the same set of rings, the instrument itself and any performance utilizing it become inherently collaborative experiences. During performance, clients or ”soloists” connect to a game server and control individual instances of their character avatar within a shared and networked virtual space on the server.

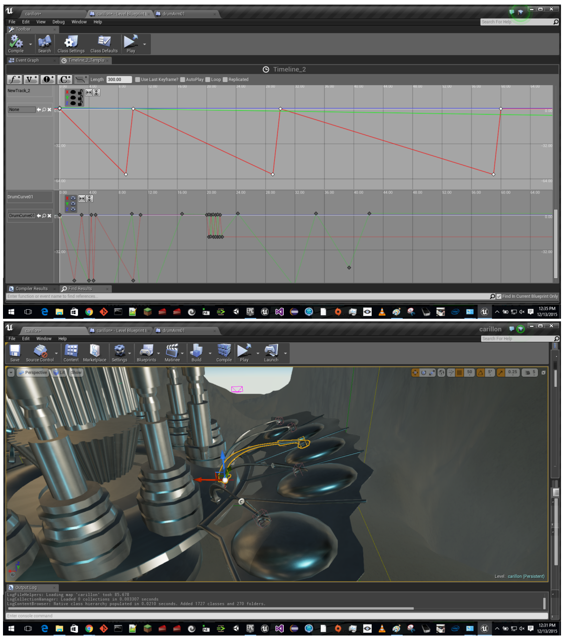

Figure 3. Event timeline (top)in Unreal Engine 4 triggers the motion of OSC-generating bell strikers (bottom).

In addition to the interactive sound-generating processes controlled by performers, a second musical component to *Carillon* is performed by a series of bell-plates and animated strikers attached to the main bell-tower structure. A series of scripts generated in the Unreal Timeline editor drive the animation sequence of each striker according to a pre-composed sequence created by the composer. When triggered by the Timeline, strikers swing and contact bell-plates, triggering an OSC message which drives the bell model in Pure Data. In this manner, composed rhythmic and harmonic material interacts with the live performance gestures controlled by each performer, adding structure to the work while still allowing a great deal of improvisation and interplay during any performance.

An accompaniment to both live soloists and pre-composed bell sequences is provided by a third sound component to *Carillon*, namely an ensemble laptop orchestra layer. Performers control an interactive granulation environment written in ChucK [7] while following a written score and a live conductor. Material for the granulator instrument is generated from pre-recorded samples of percussive steel plate strikes and scrapes, and the striking and scraping of strings within a piano.

3. MODES OF PERFORMANCE

In addition to the original ensemble performance configuration featuring soloists wearing head-mounted displays *Carillon* has been successfully presented in two additional configurations including a solo multi-channel performance and an interactive gallery installation.

Figure 4. Solo performance of *Carillon* featuring two Leap Motion controllers and sixteen-channel audio.

3.1 Solo-performance

One drawback of *Carillon* performances with HMD-wearing performers has been the disparity between the immersive three-dimensional views presented to each performer and the two-dimensional rendering of the instrument/environment presented to the audience. During a number of solo performances of *Carillon*, the performer’s in-engine view was presented to the audience, allowing them to see the complete interplay between Leap-controlled avatar arms and the performative gestures driving the musical system (see Figure 4). In this configuration, a Windows shell script triggered both dynamic camera views of the environment as well as the bell sequence. A multi-channel version of the ChucK patch performed by the laptop orchestra was driven by a second Leap Motion controller, with gestures based on hand motion, location and relative placement all mapped to parameters of the ChucK instrument. Output from Pure Data and ChucK instruments were spatialized around a sixteen-channel speaker environment.

3.2 Gallery Installation

As part of the ”Resonant Structures” gallery show at Stony Brook University’s Paul Zuccaire Gallery, an installation

[1] Epic Games, https://www.unrealengine.com

version of *Carillon* was designed and implemented (see Figure 5). Using two Leap Motion controllers and a single display, visitors could control the virtual carillon, listening to their collaborative sonic output over two sets of headphones. Like the solo-performance version, aspects of the installation including the bell sequence and camera views were triggered and controlled by a looping shell script.

Figure 5. Gallery Installation: *Carillon* on display at Stony Brook University's Paul Zuccaire Gallery, 2016.

4. GESTURE AND MOTION

The role of gesture in musical performance can be both structural and artistic. In traditional musical performance, physical gesture has been necessary to generate sufficient energy to put instrumental resonating systems into motion. Musical performers and dancers at the same time convey intention, performative nuance and expressivity through their gestures, shaping the performance experience for performers and audiences alike by communicating without language [8, 9]. In each case, the injection of energy into a system, be it via articulated gesture into a physical resonating system or through a sequence of motion into a choreographed or improvised pattern of action.

In *Carillon* a conscious effort was made to impart physicality into the control of the instrument, itself only existing in rendered virtual space. Hand motions in three-dimensions - respectively pushing into the screen, swiping left or right or moving up or down - inject energy into the selected ring or rings, causing them to rotate around the chosen axis. In this manner, human physical gesture as articulated through the hands and arms is used to mimic the physical grabbing and rotation of the gears of the main instrument. The angular velocity of performer gesture translates into directional rotation speed for the instrument, affecting different sonic parameters for each direction of rotation.

5. PARAMETER MAPPING

To create a tight coupling between the physically-guided rotation of *Carillon*'s central rings and the sonic output that they controlled, parameters representing the speed and current directional velocities for each ring were used to control a Pure Data patch based around Jean-Claude Risset's classic additive synthesis bell model.

For each ring, a set of starting partials was calculated and tuned by hand, to create a set of complementary pitch spaces. The root frequency of each bell model was driven by the speed of rotation in the X or horizontal plane (as viewed by the performer). The amplitude of each bell model was driven by speed of rotation in the Z or vertical plane. A quick gesture pushing into the Y plane to stops a selected ring or set of rings.

To add to the harmonic complexity of the work, the starting partials for each ring were varied for each individual performer, with each performer's output being spatialized to a different set of outputs. In this manner, performers had to cooperate with one another to create or preserve interesting timbral and harmonic structures. As it was not always immediately apparent as to which parameter from each performer was creating a particular desired timbre or harmonic structure, performers had to employ a great amount of listening and exploration to move the performance in desired directions.

6. SYSTEM ARCHITECTURE

The visual and interactive attributes of *Carillon* were developed using the Unreal Engine 4 by Epic Studios, a commercial game-development engine used for many commercial game titles. The Unreal Engine 4 is free to use for non-commercial projects and can be used in commercial projects based on a profit-sharing licensing model. Art assets including 3D objects, animations, textures and shaders were created using industry-standard tools including 3ds Max and Maya by Autodesk.

Within the Unreal Engine, the Blueprint scripting language - a workflow programming language for controlling processes within Unreal - was used to script the interactions between player and environments, the networking layer, and custom camera and player behaviors. External plugins, developed by members of the Unreal and Leap Motion developer communities were used to bind hand-tracking data from the Leap Motion devices to avatar limb skeletal meshes [2] as well as to output Open Sound Control messages from Unreal to Pure Data [3].

6.1 Bell Sequences

A precomposed array of struck Risset bell models is used as an additional compositional element. A sequence of pitches was composed and stored within Pure Data. Notes from this sequence were triggered by OSC messages generated by collisions between rendered hammers and a series of rendered bell-like plates. Collision boxes attached to each striker were scripted to generate unique OSC messages when they collided with each plate, turning each visual artifact into a functioning musical instrument.

[2] https://github.com/getnamo/leap-ue4

[3] https://github.com/monsieurgustav/UE4-OSC

Rather than control the motion of each bell striker from Pure Data over OSC, a novel technique native to the Unreal Engine was explored for this musical control system. Using the Blueprint "Timeline" object, multiple parameter envelope tracks representing the speed of motion and angle of articulation for each individual bell striker can be precisely set over predefined time periods (see Figure 3). Timeline object parameter tracks are typically used to automated variables for specified game entities over a given timeframe. The automation of an OSC enabled interaction, occurring when the rotation of a striker entity is driven by the timeline track to collide with a bell-plate entity, serves as a notated multi-part score wholly contained within the Unreal Engine's internal interface.

6.2 Laptop Orchestra Arrangement

For the premiere performance of *Carillon*, an additional laptop-based performance interface was created for the Stanford Laptop Orchestra. Following a notated score and a human conductor, the ensemble of nine performers controlled individual instances of a software granulator written in the ChucK programming language. Sound for each SLOrk member was produced from a six-channel hemispherical speaker alongside each performer, distributed around the concert stage in a semi-circle. Sound material for the accompaniment was composed using fragments of metallic percussion and string recordings and granulated in real-time by each performer. Gestures notated in the performance score matching temporal cues from the Timeline-driven carillon-bell tracks were performed en masse by the ensemble. In this manner live gesture performed by *Carillon's* soloists is married to both the pre-composed bell tracks as well as the real-time granulated performance by the laptop orchestra.

7. DISCUSSION

As an exploration of rendered space and that space's suitability to sustain performance interactions capable of driving music and sound, *Carillon* has been an extremely successful work. The integration of head-mounted display, hand-tracking sensors and procedurally-generated sound creates a novel yet physically intuitive interaction model that can be learned quickly yet explored to create nuanced sonic gesture. The client-server architecture of the work allows for multiple potential configurations ranging from ensemble to solo performance, from gallery installation to networked "game" play, as shared on the Leap Motion developers website.

Carillon was designed with the specific intent of exploring the nature of collaborative instruments controlled by human gesture while residing in a shared network space. Simple yet intuitive physical gestures as tracked by Leap Motion sensors allow performers to affect changes upon the instruments procedurally-realized timbre, frequency and amplitude at varying scale (from small to large) and in real-time. By presenting *Carillon* to soloist performers using HMDs, the feeling of depth and presence associated with functioning VR devices allows the performer to utilize depth in gesture more accurately and intuitively. The performative yet necessarily collaborative aspects of *Carillon*'s central ring-as-instrument metaphor not only allows each performer to improvise freely, but also adds a level of constraint in each performer's ability to either inhibit or augment one another's gestures. A more complex and articulated musical work is realized through the addition of pre-composed bell sequences and a live-yet-composed laptop orchestra accompaniment.

8. ACKNOWLEDGEMENTS

The design, development and production of *Carillon* has been made possible through generous hardware grants by NVIDIA and Leap Motion.

9. REFERENCES

[1] M. Mathews and R. Moore, "GROOVE: A Program to Compose, Store, and Edit Function of Time," *Communication of the ACM*, vol. 13, no. 12, 1970.

[2] M. Wanderley, "Performer-Instrument Interaction: Applications to Gestural Control of Music," Ph.D. dissertation, University Pierre Marie Curie - Paris VI, Paris, France, 2001.

[3] G. Wang, N. Bryan, J. Oh, and R. Hamilton, "Stanford Laptop Orchestra (SLORK)," in *Proceedings of the International Computer Music Association Conference*, Montreal, Canada, 2009.

[4] C. Dodge and T. Jerse, *Computer Music: Synthesis, Composition and Performance*. Schirmer, 1997.

[5] M. Puckette, "Pure Data," in *Proceedings, International Computer Music Conference*. San Francisco: International Computer Music Association, 1996, pp. 269–272.

[6] M. Wright, "Open Sound Control: an enabling technology for musical networking," *Organised Sound*, vol. 10, pp. 193–200, 2005.

[7] G. Wang, "The ChucK Audio Programming Language: A Strongly-timed and On-the-fly Environmentality," Ph.D. dissertation, Princeton University, Princeton, New Jersey, 2008.

[8] S. Dahl, F. Bevilacqua, R. Bresin, M. Clayton, L. Leante, I. Poggi, and N. Rasamimanana, *Gestures In Performance*. New York: Routledge, 2010, pp. 36–68.

[9] A. Jensenius, M. Wanderley, R. Gody, and M. Leman, *Musical Gestures: Concepts and Methods of Research*. New York: Routledge, 2010, p. 13.