Designing VR Applications with CsoundUnity

Pedro Sodre*

NYU
pls9788@nyu.edu

Abstract. This paper presents two interactive music systems built for the Oculus Quest that use the CsoundUnity package. The first system explores different ways to control Csound instruments and samples by interacting with 3D objects using object collision and the grip buttons of the VR controller. The second system transforms Boulanger's classic "Trapped In Convert" into an interactive system that allows users to play and spatialize adaptations of the original Csound instruments used in the piece - to remix it and recompose it and play along with it.

Keywords: Csound, Unity, CsoundUnity, VR

1 Background and Goal

With the Unity XR Interaction Toolkit it is possible to create applications for the Oculus Quest that use the different types of input from the device's controller. Version 3.0 of CsoundUnity allows the developer to manipulate Csound instruments inside Unity and create standalone apps that run on the Oculus Quest. The author's first explorations of the CsoundUnity package involved creating multi-user systems for remote jamming during the Covid19 pandemic (Fig. 1), and designing sounds for a virtual creature (Fig. 2) using Csound instruments. After these initial experiments, the author started to explore VR interactive music systems. The goal of this paper is to document some of the design decisions made while designing these VR Applications and point to some of the next steps toward improving them as well as share some ideas they have inspired.

2 System 1 - SoundXplorer

2.1 Introduction

The author's undergraduate thesis included a VR system called *SoundXplorer* (Fig. 3) . This system featured different sets of gestural instruments that users could explore and perform with collide, grab, drag and throw actions.

* I would like to thank Dr. Richard Boulanger for the mentoring in the development of these projects and the writting of this paper. I would also like to thank Rory Walsh and Giovanni Bedetti for the development of CsoundUnity and their inspiring examples.

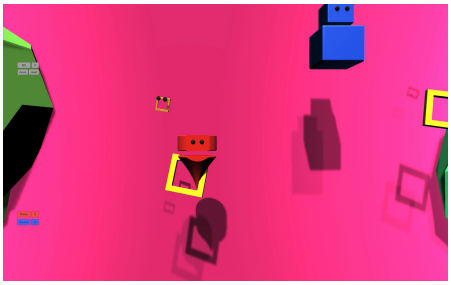

Fig. 1. Connect[1]: A CsoundUnity System for Remote Musical Interaction

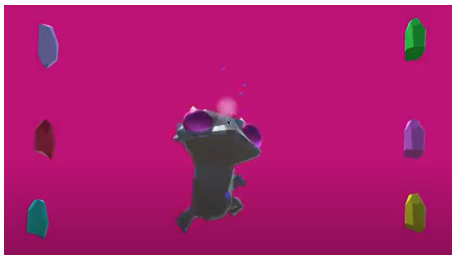

Fig. 2. Tasha[2]: A CsoundUnity Virtual Pet

2.2 Main Instruments

In *SoundXplorer*, users choose between three sets of instruments; each featuring three sound-producing 3D objects that appear at the center of the interface.

The first set features three spheres that users touch to produce sound, and grab and drag to modify the sound. In this instrument, by placing the hands, that represent the users controllers, inside spheres and pressing the grip button, the instrument generates notes that the user can transform the timbre by moving the controller around. The rotation of the two of the spheres on the X and Y axis is scaled and mapped to parameters of two FM instruments (that use the foscil opcode), and the other is mapped to a Karplus-Strong instrument. The rotation also alters the 3D shape by altering parameters of its shader.

The second set features a pandrum instrument with three pans that use mode opcodes to simulate the sound of bells. By colliding the controller's game object with the pans, a note is generated and the shader produces a corresponding visual movement of the object. Averaging the change in the values of the hand positions is mapped to the cutoff frequency and amplitude of the modal synthesis algorithm. Further, by dragging the controller while grabbing the pandrum object, the user alters the base frequency of the modes used to generate the sound.

The third set features a set of three floating vertical bars that respond to collisions. When the user's controllers (or virtual hands) collide with the vertical bars, the algorithm randomly picks notes from an arpeggio. Above and below

Fig. 3. SoundXplorer[3]: A CsoundUnity VR Interactive System

are cubes that control octave and pitchbend. When pressing the grip button and dragging the one below, you change octave, and when grabbing the one above, you do a pitchbend. In this set the shaders are also animated and wobble when you collide with the rods.

2.3 Playback Instruments

In addition, two sets of "playback instruments" were created - one more active and the other more passive. These are meant to accompany the users as they interact with the playable instruments described above. The playback instruments were positioned the top part of the interface and the user could toggle between both sets by pressing the grip button while colliding with UI elements on the right side of the interface. Thus, each of the sets contains six sound making game objects. The total collection of playback instruments consist of eight that feature Csound made samples and four employing Csound instruments that a renders in real time. These can be divided into the following three categories: 1) A sample-based pad instruments that can be toggled on and off with the grip button. 2) A real time generative Csound drum instrument that can also be toggled on and off with the grip button and whose tempo is changed by pressing the grip button and dragging the controller. 3) A percussive shaker (that is spatialized using Unity's native audio-source spatializer), that can be grabbed, moved around, and thrown away, and over time, will return to it's home position.

2.4 Mixer

Additionally, the system features a mixer in which the user can balance the levels of all the main instruments, and set the master level as well. The mixer

also includes mute buttons and slow fade-in and fade-out buttons to make for a more musical and elegant performance experience.

3 Trapped Xplorer

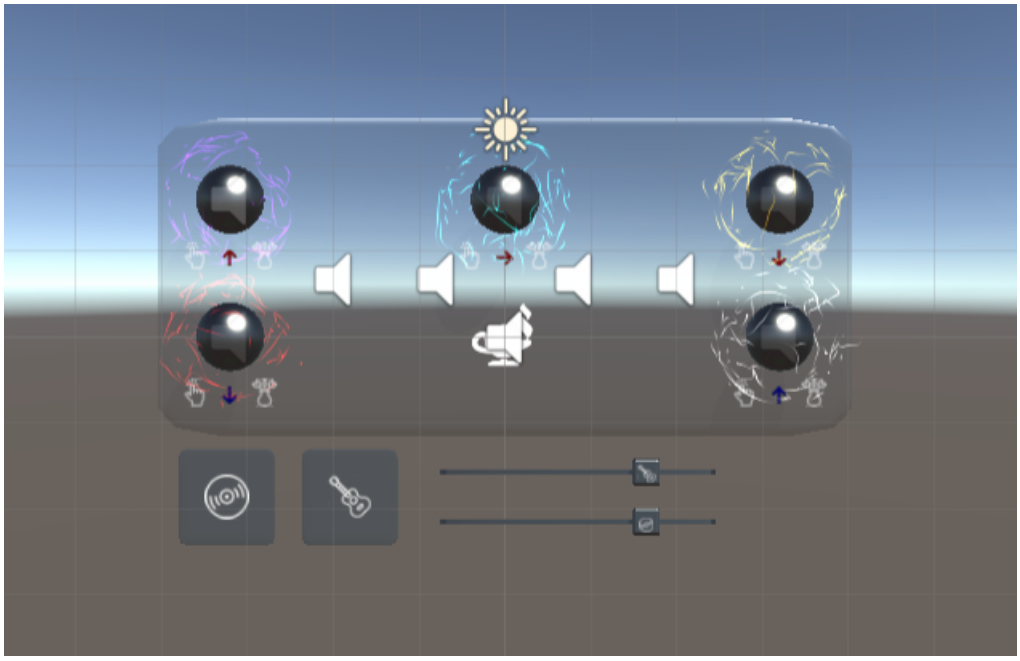

Fig. 4. TrappedXplorer: An Interactive VR Playground for *Trapped In Convert*

3.1 Introduction

TrappedXplorer (Fig. 4) is a system that allows users to trigger, playback and spatialize sections of the classic work by Richard Boulanger. The system also allow users to interact with, modify, clone, and spatialize adaptations of his classic Csound instruments. A goal of the design was to adapt the instruments in a way that they were still recognizable, but could also be malleable. In the system, all the audio is rendered in real time by CsoundUnity. Currently, the system gives users two menus from which they can activate, spatialize, and clone either playback or mutable instruments. Currently, users can create (clone) up to three copies of each interactable instrument.

3.2 Playback Instruments

In TrappedXplorer there are four different playback instruments, each associated with a small metallic spheres. While hovering over or touching them and pressing the A or the X buttons, the user toggle them to play or stop. When the instrument is toggled on, the sphere turns blue. The spheres can be grabbed, pulled out of the menu, and spatialized in the virtual world. (Each game object from the playback instrument group has a CsoundUnity component rendering a particular section of "Trapped in Convert" in real time, and can be transformed in the virtual world changing their pitch and time.

3.3 Interactable Instruments

Currently, there are five different interactable instruments which are associated with larger metallic spheres with different colored particles that surround them (as seen in Fig. 4). By pointing at them you get some vibrational feedback, and then pressing the trigger, the particles becomes bigger and the instrument produces sound. These can be toggled on and off with the X or A button, or they can be set to stay on continuously while the user is touching them. This way you can position them without making sound or play them while moving them. It is also possible to group the interactable instruments and trigger them all with a single gesture. Many audible parameters are mapped to position, location, orientation and specific combinations of buttons to fully explore and exploit the full range of sonic possibilities found in each of the adapted Trapped instruments.

3.4 Future Enhancement of TrappedXplorer

The author plans to continue developing the TrappedXplorer system. This will allow users to create new versions of the piece as a part of live music performances or just for fun. To that end, additional functionalities will be implemented. The Y position of the selected object will control the pitch of the instrument. Instruments will be adapted to further support their appropriate and controllable spatial location cues, volume and EQ will better model distance cues. Of note, the trigger button on the Quest is actually sending continues values and will be used to modify the envelopes of the instruments to allow for more nuanced articulation of the sounds. Grouped objects will respond uniquely to the controllers, and therefore, the system is being modified to respond appropriately when a group is selected.

One of the loftier aspiration is for this system to serve as a compositional tool. To that end, we plan to capture Xplorations so that they can be revisited, and shared. The hope is that expert users might enjoy making music in this space. But also, the hope it that by adding more visually responsive elements and behaviors, visual and haptic output, associated with the changes in the sound and their locations, that everyone will also enjoy playing in the world of this piece, and making it their own.

4 Beyond

As a result of working on the systems described in this paper, systems that harness the visual power of Unity with the audio power of Csound, the author is now, more than ever, motivated to build interactive systems for the Oculus Quest that create multi-user playful music-making experiences that look, feel, react, and sound beautiful, and are at the same time, engaging, inspiring, invigorating, and relaxing to both the casual and the expert user.

References

1. Connect (2020) - shorturl.at/EORU6
2. Tasha (2020) - shorturl.at/drBLU
3. SoundXplorer (2021) - shorturl.at/qtUY6