The Implementation of Augmented Reality and Low Latency Protocols in Musical Instrumental Collaborations

Qixiao (Jimmy) Zhu, Dr. Paul J. Botelho

Emerging Scholars

Department of Music, Bucknell University, Lewisburg, Pa.

Abstract

The project attempts to use augmented reality as a medium of sound production. A synthesizer and two modulars are created in the augmented reality environment. The user can tweak the synthesizer and modulars inside the headset while still being able to adjust the pitch of the synth by playing a physical MIDI instrument. The project was created using the Unity game engine. Various other packages and tools such as the Passthrough API, Oculus Interaction SDK were used. Functions for different kinds of wave forms were used for the synthesizer and the modulars. Further development such as adding multi-user interaction, other kinds of digital instruments, and improving on the existing control buttons and leavers are considered.

Introduction

This is an experimental project with a purpose of implementing sound producing mediums to the VR headset while still being able to play the actual musical instruments in the user's hands. The project is at its early stage with basic controls and augmented reality environment set up, and a built in synthesizer that produces several different sounds. The synthesizer works well with two modulars that are also implemented: frequency modular and amplitude modular.

Augmented Reality Environment and Hand Tracking
The environment was implemented so that the users can see the surroundings and the musical instruments through their VR headsets. Hand tracking is also implemented so that the users can use their hands to interact directly with the virtual objects instead of having to grab physical controllers to interact with the virtual objects.

Transmitting MIDI Signal from Computer to Headset
The keyboard outputs MIDI signals to the connected computer. The computer transmits the signals using a network protocol to the headset. This ensures that no wire is plugged to the headset when playing musical instruments.

Synthesizer
The synthesizer uses wave forms to produce sound. It produces four different kinds of wave forms: sine wave, triangle wave, square wave, and sawtooth wave. Each of them have different sounds.

Frequency Modular
The frequency modular is an object created to modulate the frequency of the oscillator inside the synthesizer. The modulation also has four different kinds of wave forms.

Amplitude Modular
The amplitude modular is an object created to modulate the amplitude of the oscillator inside the synthesizer. The modulation has the same four different kinds of wave forms.

Methods

Tools used throughout the project

❑ Oculus Quest 2 Headset
- Allows programs to run on the headset without connecting any wires to other devices

❑ Unity and C#
- The scene was set up in the Unity game engine and game objects are coded using Unity C#

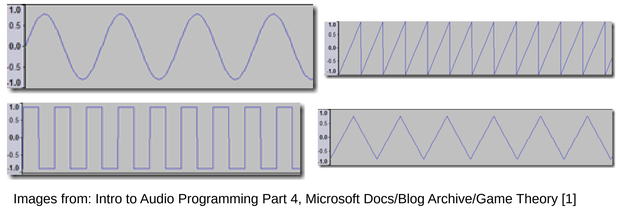
❑ Oculus Passthrough API
- The only augmented reality API available for Oculus headsets
- Does not allow passthrough when running the AR program on the computer

❑ Oculus interaction SDK
- Hand tracking and hand grab interactions

❑ UDP protocol using extOSC in Unity C#, PythonOSC in Python
- Transmitting MIDI signal in the format of an array

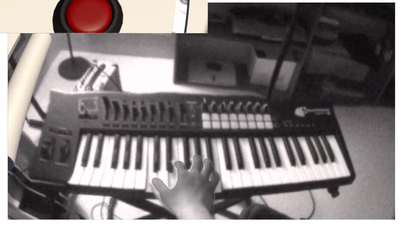
❑ VR Ready Controls
- An Unity asset with leavers and buttons and the basic code for them
- The buttons control the state of the synthesizer and the modulars (On/off), other buttons control the wave forms of the three game objects.
- Leavers control the volume, amplitude, and frequency.

❑ Formulas of wave forms are used to make oscillators inside synthesizer
- Sine wave: Sample = Amplitude * sin(t*i)
- Square wave: Sample = Amplitude * sgn(sin(t * i))



Images from: Intro to Audio Programming Part 4, Microsoft Docs/Blog Archive/Game Theory [1]

Results



The synthesizer works fine and produces sound through the headset. Tweaking the synthesizer with the modulars produces a variety of different kinds of sounds.

Discussion

Result and Drawbacks
- The result is a synthesizer implemented inside the headset.
- The synthesizer's interface is not movable by the user because of the physics of the buttons and leavers (they get stuck by wobbling when moved).
- The sawtooth wave forms inside the two modulars still has some problems functioning. I am trying to find simpler sawtooth functions that I can use.
- Due to the nature of the wave functions, heavy computation capability is required. The Quest 2 headset still lacks a bit in this field. There is noticeable latency when the synthesizer produces sound after user physically pressing the keys.

Possible Ways of Improvement and Future Plans
- The button and leavers should be re-implemented without physics so that the interface of the synthesizer can be moved without buttons ad leavers wobbling.
- Change the wave functions to pre-generated arrays of numbers that follow the wave forms.
- Will implement an arpeggiator for the synthesizer
- Will add in avatars for the user and connect two headsets so they can see and hear each other

Conclusion

The project in its state right now might not be a product that can be provided to people and function as a more effective way of producing music, but it can be something that is more effective in the future (with lighter and more powerful headsets and MIDI instruments with built in networking).

References

[1] kexugit, "Intro to Audio Programming Part 4: Algorithms for Different Sound Waves in C#." https://docs.microsoft.com/en-us/archive/blogs/dawate/intro-to-audio-programming-part-4-algorithms-for-different-sound-waves-in-c (accessed Jul. 20, 2022).

Acknowledgements

Thanks to my mentor Dr. Paul J. Botelho for helping and giving directions to the project.
Thanks to Emerging Scholars and the Schotz Family Fund for this opportunity and funding.