Matthew Walak

Konrad Kaczmarek

Fundamentals of Music Technology (MUSI 315)

13 December 2022

Music Tech Final Project – The Musical Ball Box

Abstract:

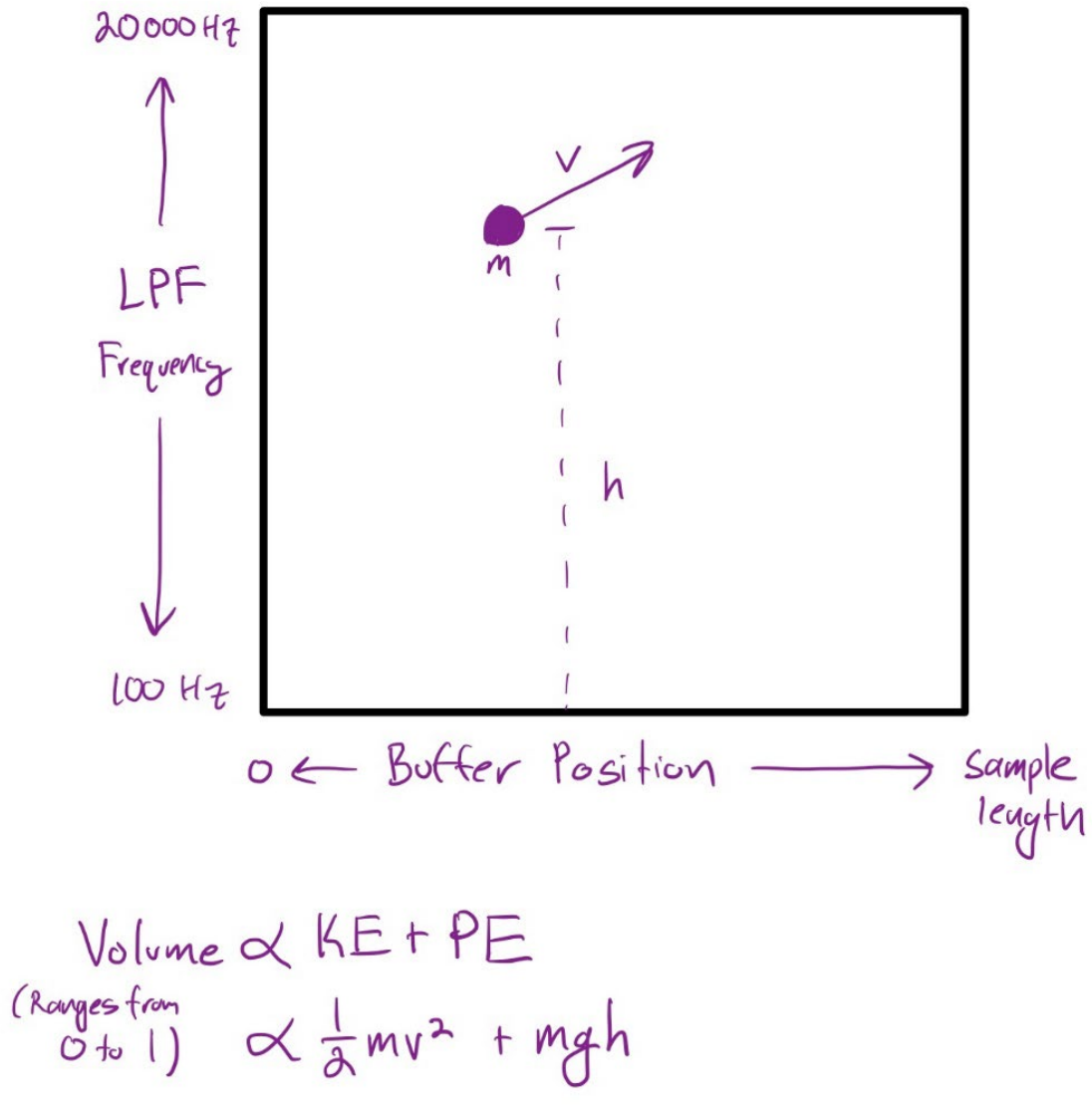
I created an interactive audio-visual instrument using Unity as a graphics engine and Chunity (ChucK for Unity) as an audio engine. This instrument leverages the periodic nature of simulated 2-dimensional motion to produce sample-based music. This instrument also uses techniques inspired by early tape-based manipulations of recorded sound such as looping, varying playback speed, and filtering to complicate the musical texture. The software produced from this project can not only be used as a tool to create pieces of music, but also as a tool to explore properties of a sound in a musique concrete context.

Instrument Design:

To run the instrument, Launch ‘Music_Tech_Final.exe’ in the ‘Windows_Build’ folder (Sorry, no Mac version yet)

The Musical Ball Box instrument consists of a 2-dimensional square box, filled with 2-dimensional balls. The balls follow 2D kinematics as simulated by Unity’s Physics2D engine. Each ball within the box represents an audio buffer in memory. A ball’s x position determines the position of the playback head in the audio buffer, meaning a ball’s x velocity is proportional to the playback rate of the buffer. Before the sound from an audio buffer is sent to the speakers, it is sent through a low pass filter whose cutoff frequency is determined by the ball’s y position. The volume of a given audio buffer is determined by the total energy (kinetic plus potential) of its corresponding ball.

Figure 1: Instrument Design



Both volume and low pass filter frequency are interpolated exponentially to better align with human perception of pitch and volume. This is achieved using ChucK's built-in rmstodb() function (To convert a linear, 0-1 amplitude to decibels), and dbtopow() function (To convert dB to a power value).

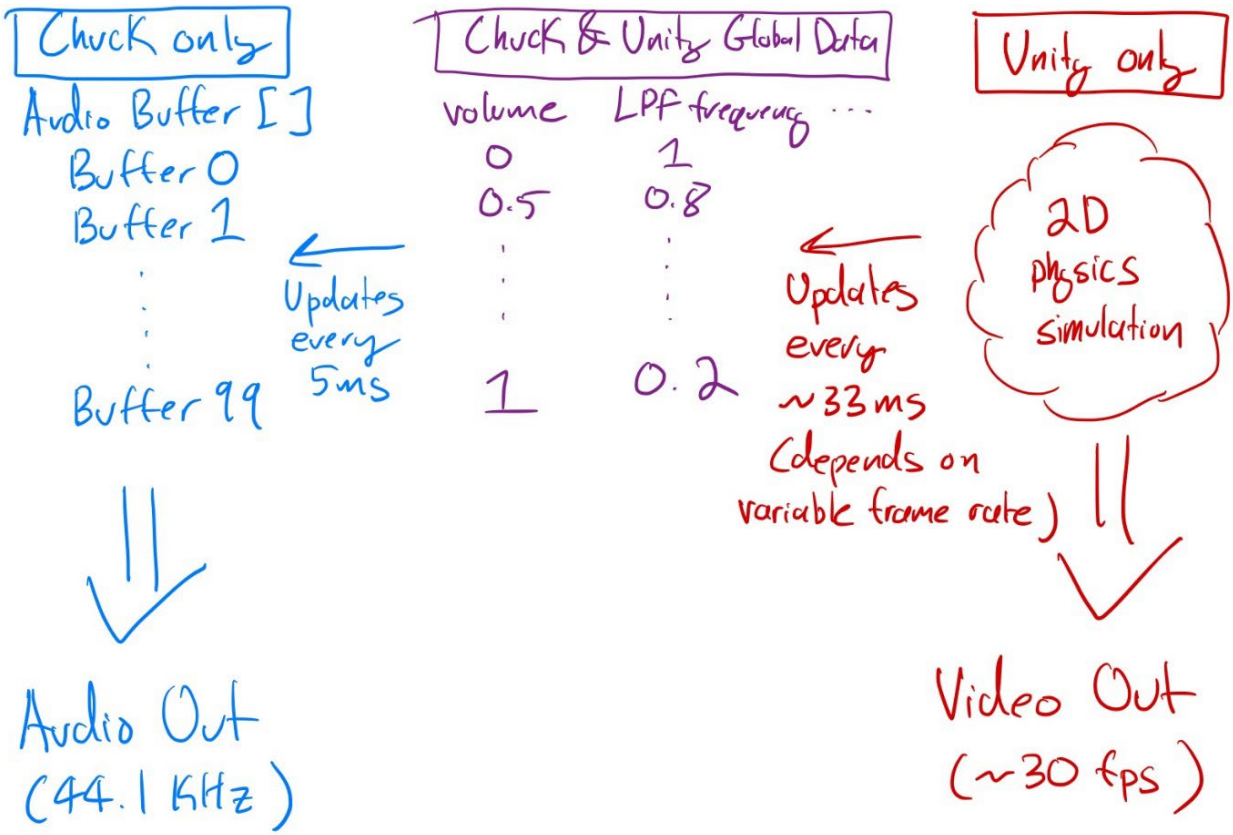
System Architecture:

The complete set of project files (Including all code) can be found here:

https://github.com/mrwalak/MusicalBallBox

Because the graphics of Unity and audio of ChucK run on different threads and at different rates, a system was developed with buffer IDs to manage cross thread communication. Upon initializing the program, ChucK instantiates 100 audio buffers, each with a unique ID. Each audio buffer has a set of data stored in global arrays for all properties of the buffer that should be accessible from Unity. Every 5 milliseconds, ChucK updates all sounding audio buffers with data from these global arrays. This allows Unity to update the global arrays at any rate and have those changes reflected on the audio thread in 5ms or less.

Figure 2: System Architecture



Discussion:

This instrument heavily draws inspiration from early electronic music and musique concrete pieces which used tape-based manipulations of sound to create music. In a similar approach to musique concrete composers, this instrument allows a user to recontextualize a sound by playing it at a different speed, in reverse, and with different filtering effects. Creation of this instrument was also inspired by the works of Steve Reich. Many Steve Reich works highlight how looping can change the meaning of sound, and how multiple loops running at different rates can further complicate this meaning. This informed the decision to make the instrument a closed 2D box, which allows for balls to ricochet back and forth in a periodic fashion. This also informed the decision to include gravity in the simulation, which produces periodic bouncing motion.

The instrument largely succeeded as a tool to recontextualize sound and succeeded in removing technical barriers for creating music based on manipulating digital audio. In future iterations, the instrument could benefit from changes to how a ball's total energy impacts its sound. Currently, there are certain sounds that are hard to produce because of how energy is treated. For example, it is hard to hear a sample with a very slow playback rate and very low filter cutoff. This is because a slow playback rate corresponds to low kinetic energy. A low filter cutoff corresponds to low potential energy. Because of how energy is treated in the instrument, this sound would be barely audible because the volume would be set very low. Conversely, it is difficult to make a sample sound at a low volume when it has a high playback speed and high filter cutoff. The overall result is that higher frequencies are generally favored by this instrument, making it harder to experiment with the lower end of the spectrum.

Additionally, this instrument could benefit from different types of filter effects. With the current low pass filter, it is often difficult to distinguish between when a sound is attenuated by the low pass filter or when it is attenuated by volume, because the results of extreme attenuation by a low pass filter is silence, which is the same result as extreme attenuation by volume. Switching the low pass filter to a bandpass filter, notch filter, resonant low pass filter, or frequency boosting filter could have interesting effects.

Lastly, the ability to record and import customs sounds was experimented with, but not fully implemented in this iteration of the instrument. Having this feature would greatly expand the instruments creative potential and would give users and even broader view of sound through the lens of musique concrete.