

“Pool of Toys”: Multimodal Visual Music Rendering of Sequenced and Real-time Control Events

Satsuki Kawahara¹ & Michael Cohen¹

¹ Spatial Media Group, Computer Arts Lab., University of Aizu; Aizu-Wakamatsu, Fukushima 965-8580; Japan

Abstract. The goal of this study is to display music visually to aid recognizing each component, enjoying music not just aurally but visually as well. Elements of western music comprise multiple aspects (e.g., rhythm, melody, harmony, and orchestration). To express them, we introduce a “live song” called “Pool of Toys.” A pool filled with toys is used as a motif in a game engine (Unity). A multitimbral song including musical elements was created using DAWs (GarageBand and then Ableton Live). Each part is imported as a MIDI track into Unity, controlling the attributes of visual elements (floating toys) at runtime. A surface grid controller (Launchpad Pro) is used at runtime along with the precomposed music so that performers can synchronously play the MIDI terminal with output displayed in realtime.

1 Introduction

Music is basically auditory information. Vibrations in air reach our ear and move the eardrum. By the complex structure of ear, this vibration is converted into neural signals and processed in the brain. The percentage of each human sense occupied in our brain information is approximately as follows: Taste 1.0%, Tactile 1.5%, Smell 3.5%, Hearing 11.0%, and Visual 83.0%. Visual modality occupies the largest portion (83.0%), while auditory modality is slightly above 1/10 of it. Sometimes we feel people who have never learned musical instrumentals experience difficulty in recognizing detailed parts in songs. Visualizing music can support recognizing song details. At some concerts, performers often project video that provides visual effects matching the rhythm and the story of songs. Precise visual effects are possible thanks to the technological advancement. Recently, there is research on visual effects in hearing [3] and improved performance in music using visuals and sounds together [4]. In this research, we aim to develop a system that allows people to enjoy multisensory experience (visual and auditory) when listening to music. We implement the system using the game engine Unity and various accompanying tools. Our system supports audition with visuals, and it can help listeners recognize music details.

2. Methodology

2.1 Multitimbral Synthesis

Multitimbrality describes the multiplicity of parts that can be played by an instrument or synthesizer [5]. Generally,

these parts are assigned different MIDI channels to be controlled independently. (This is not the same as polyphony, or the number of notes that can be generated at the same time). When musical instruments (synthesizer, sampler, etc.) can generate more than one timbre, they are called multitimbral.

2.2 Polyphony

Polyphony in music is simultaneously combination of multiple tones or melody lines [6]. Therefore, simplex interval composed of harmony of multiple simultaneous tones, such as a chord, is polyphony. Counterpoint is pattern of polyphonic voices, such as parallel melody lines.

2.3 DAW (Digital Audio Workstation)

A DAW (Digital Audio Workstation) is music creation software. It integrates various functions (MIDI sequencer, musical composition, multitrack recording, audio editing, mixing, mastering, etc.) and completes musical composition to output audio files. In this study, a song was composed using Ableton Live.

2.4 MIDI

MIDI is a protocol for recording and playing music on digital devices [7]. It encodes and distributes information related to the production of music, using assumptions about audio synthesis timbre or “patch,” but doesn’t directly indicate sounds. The command set includes note on (with key velocity), note off, pitch event, and other ways of controlling synthesis.



Figure 1: Novation Launchpad Pro

Novation Launchpad Pro [8] is a MIDI surface controller triggering multitrack instruments and tone for arbitrary music software, using an 8x8 gridded RGB touchpad (Figure 1). For example, it can control drums and keyboard played at the same time. There are a variety of layouts (Note, Drum, Fader, and Programmer). In this study, Ableton Live is chosen to use Launchpad Pro. Launchpad Pro switches layouts automatically, remapping MIDI commands sent by the respective buttons.

2.5 Unity

Unity [9] is a cross-platform game engine. It is a development environment used for making games that integrates 2D/3D description, playing sounds, management of user interface, and data management/creation.

2.5.1 Sequenced Control: MIDI Tool Kit

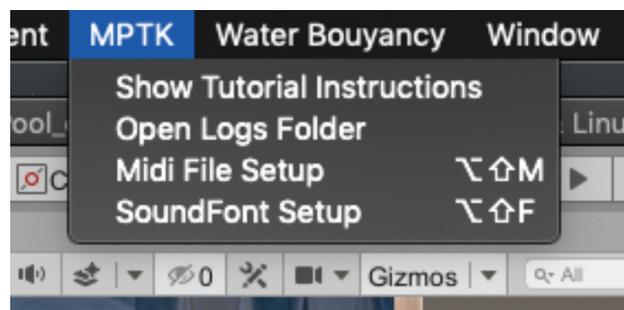


Figure 2: MIDI Player Toolkit

MIDI Tool Kit [10] is a Unity component (available from the Asset store) for playing sounds from a MIDI file (.mid) without requiring C# or JavaScript programming. It defines music using custom inspectors and synchronizes MIDI events and application responses. As shown in Figure 2, it is installed as a Unity plug-in module including a menu, the name of which header is the initials of “MIDI Player Tool Kit”.

2.5.2 Realtime Control: MIDI Jack

MIDI Jack [11] is used as MIDI input API for Unity. Using MidiMaster Class, it performs basic functions such as parsing note on/off opcodes, obtaining note velocity,

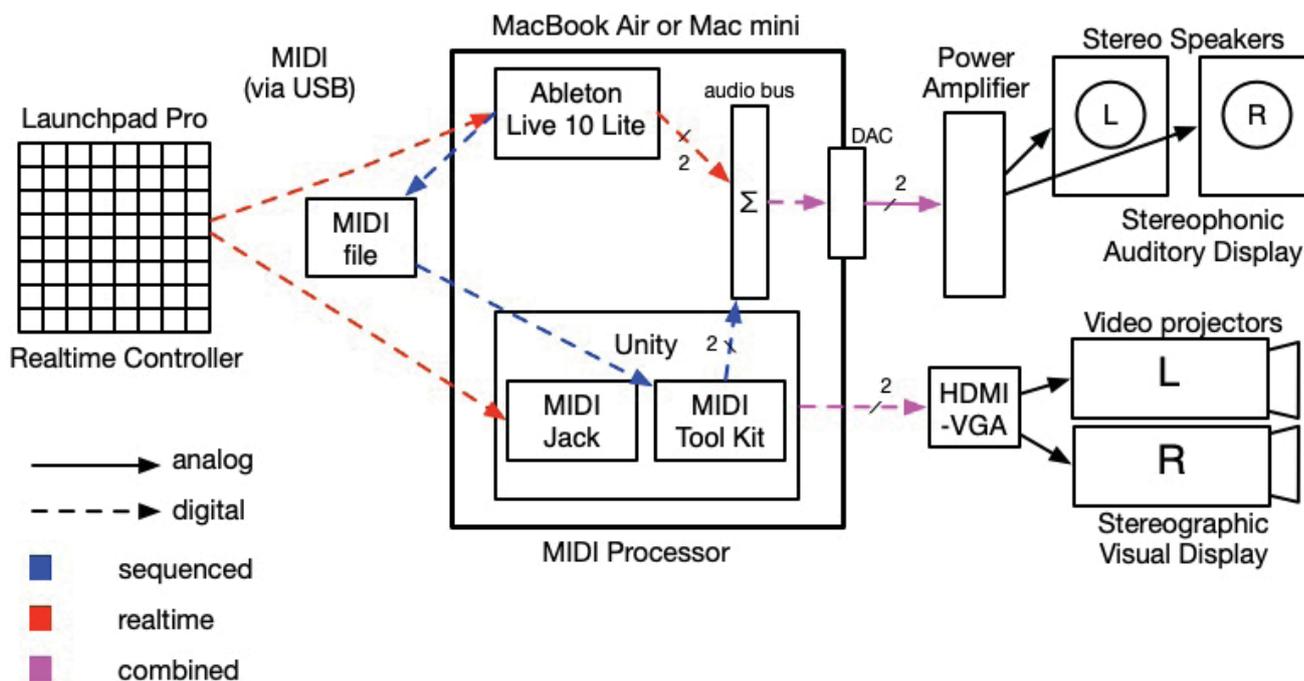


Figure 3: System architecture and data flow

interpreting performed key presses/releases, and obtaining controller values. Such given functions are displayed in a MIDI monitor window list of active devices and MIDI messages received.

3 Implementation

3.1 Overview

3D rendering was developed using macOS (on MacBook Air and Mac Mini) using Unity version 2019.3.04a with the programming language C#. Each imported MIDI track controls (modulates) the attributes of scene objects. Performers can play MIDI terminals, and the output is reflected in real-time. System diagram and flow are shown in Figure 3.

3.2 Design



Figure 4: Swimming pool (basic empty space)

The motif of the titular pool is a swimming pool, as illustrated in Figure 4, and we used game objects arranged in Unity. Toys and water objects were imported from Unity Asset Store. The expression of buoyancy was created using Unity-Water Buoyancy Simulation for Unity Engine [12], simulating toys floating in the water.

3.3 Sequenced MIDI Event

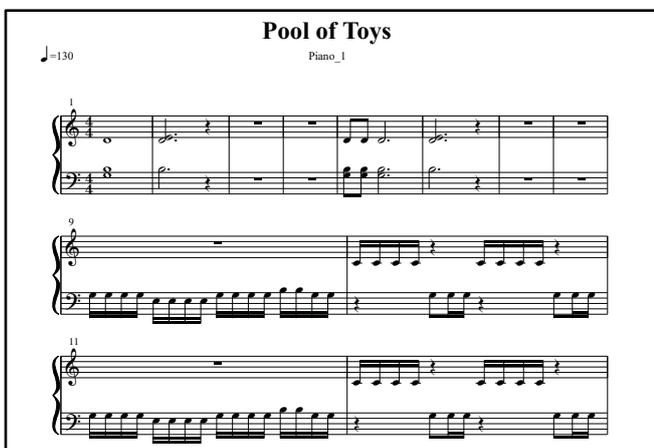


Figure 5: Piano part 1 for “Pool of Toys”; 「おもちゃのプール」

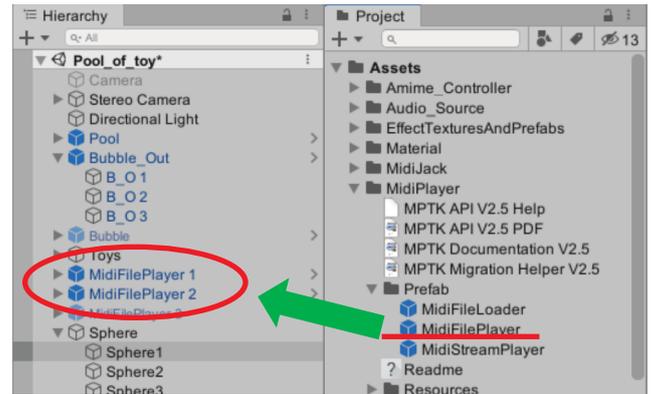


Figure 6: MIDI File Player added to Unity scene

We originally used Garage Band (DAW in Mac), but it had trouble exporting MIDI files (loop → MIDI). The song “Pool of Toys”; 「おもちゃのプール」 (Figure 5) was composed using Ableton Live, imported into Unity as a MIDI file (.mid). Ableton Live enables direct export of separated parts as .mid tracks. The earlier version was composed using Apple GarageBand on macOS, but it did not allow us to directly export MIDI tracks. The composition contains two tracks (both piano parts).

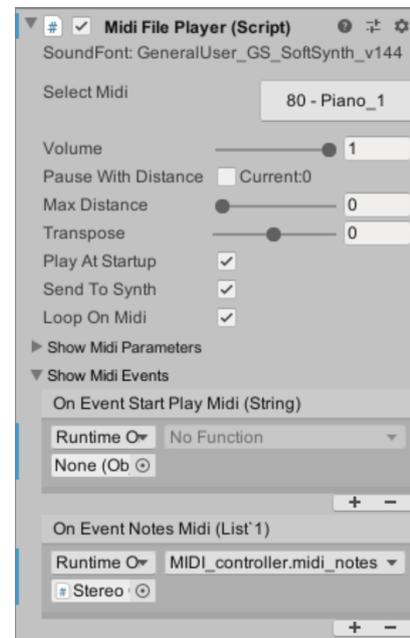


Figure 7: Select MIDI and add Event Note MIDI

Sequenced MIDI Event handler was created using MIDI Tool Kit on Unity. .mid was added to Unity via MPTK Midi File Setup (Figure 3). MidiFilePlayer (object can play .mid) from Prefab was added to Hierarchy (Figure 6), and a function (MIDI_controller.cs) that modulates objects (such as spherical bubble) when getting note from .mid was devised. The function was attached to the Main Camera,

and event was added and event (MIDI_controller.midi_notes) was selected (Figure 7). During this procedure, MidiFilePlayer references both piano sounds because it is possible to input only one file. Spherical bubbles were designed as Game objects. Each bubble is generated dynamically upon the recognition of respective note on event and disappears (“times-out”) after “deathspan,” determined as a fraction of its sustained lifetime.

3.4 Realtime MIDI Event

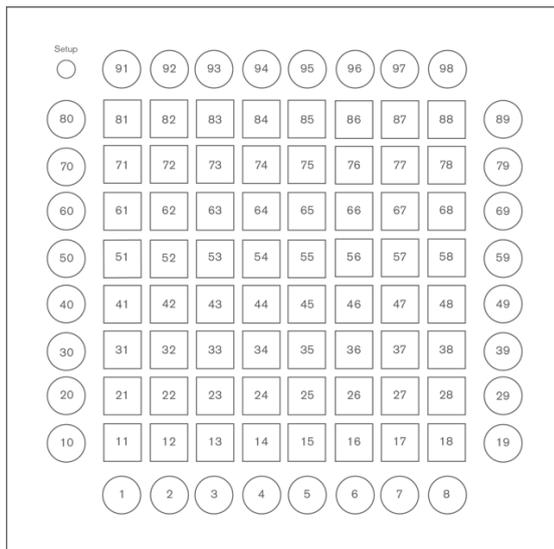


Figure 8: Launchpad Pro grid controller lay-out MIDI values

Real-time MIDI events arise from controller signal, outputting audio from Ableton Live (not from Unity). The sound is polyphonic. Programmer Layout [13] of controller was used on Unity. Numbers used and an example of how they can be allocated are shown in Figure 8. Function changes moving objects (toys) and used function in Midi Jack identifies the control based on the signal from the controller. At this stage, tones are generated as one-shot or looping sounds. Notes played by performers are displayed visually as well as aurally. Sounds include drums, synthesizer, bass and guitar, synthesized by Ableton Live. Using functions such as Animator in Unity, what is visually expressed is the movement of objects or the changing colors of them.

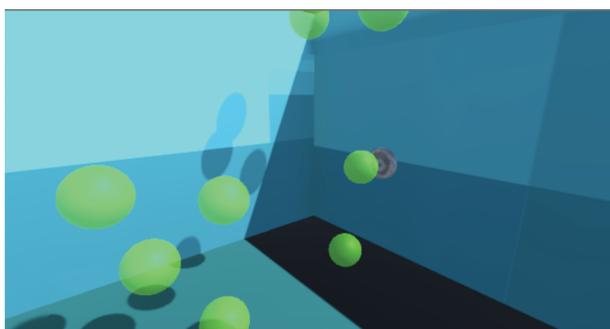


Figure 9: Sequenced MIDI events: outputting bubbles

4 Results

With our system and Pool of Toys, one can listen to and see the visual effect of both events (composed offline and online MIDI events). Performers push a controller button, and the colors and movement of rigged toys change. The prototype supports both sequenced and real-time events, generated by controller, as shown in Figures 9 and 10.

5 Future Work

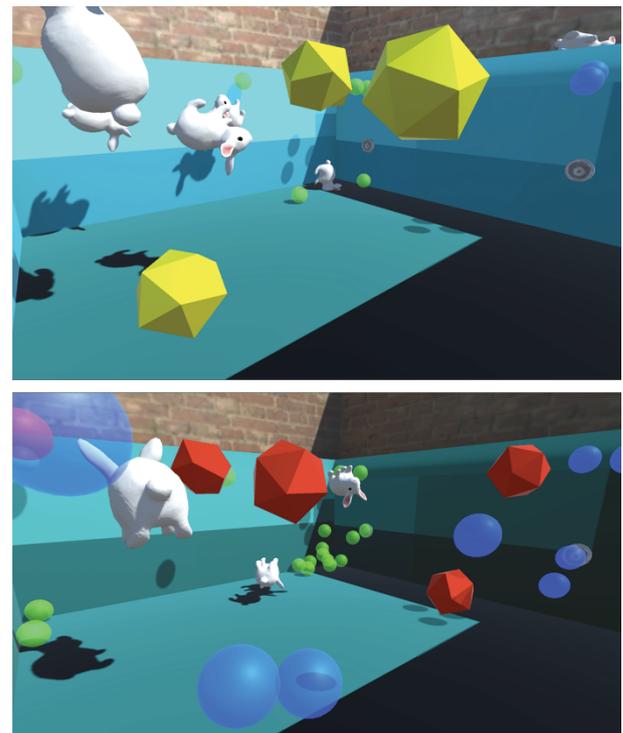


Figure 10: Realtime MIDI events: modulating scene color of toys (changing from yellow to red)

Remaining issues include the quality of the moving objects, such as water surface animation (shadows, caustics, etc.). We would also like to experiment with spatial sound and perspective (standpoint) adjustment so that users can virtually walk or swim around the underwater scene, enjoying various visual and auditory stimuli. In a song, there is a chord (harmony) progression. With each changing chord, listeners can feel the changing mood with the changing sound. So, performers can play displayed chord whose position is set by the controller more easily on Game Display of Unity. For doing this, we hope to make a system to read the chord progression from MIDI and modulate the scene accordingly.

5 Acknowledgements

We thank Camilo Arévalo and members of the Spatial Media Group for helping us.

References

1. 音が聞こえるとは
<https://www.toa.co.jp/otokukan/otomame/theme1/1-1.htm>
2. 情報の伝わりやすさと視覚の関係
『産業教育機器システム便覧』P4「図 1.2 五感による近くの割合」教育機器編集委員会編 日科技連出版社 1972
3. P. Friedrich and K. Reinhard, 2012. *When the Eye Listens: A Meta-analysis of How Audio-visual Presentation Enhances the Appreciation of Music Performance*. *Music Perception: An Interdisciplinary Journal*, Vol. 30, No. 1, Sept. 2012; (pp. 71–83). DOI: [10.1525/mp.2012.30.1.71](https://doi.org/10.1525/mp.2012.30.1.71)
4. E. G. Sharon, A. G. Lee, and B. Randolph, 2005. *Hearing What the Eyes See: Auditory Encoding of Visual Temporal Sequence*. *Psychological Science* 16(3): 228–235, March 1, 2005, DOI: [10.1111/j.0956-7976.2005.00808.x](https://doi.org/10.1111/j.0956-7976.2005.00808.x)
5. Multitimbral Synthesis
<https://www.sweetwater.com/insync/multitimbral/>
6. Polyphony
<https://www.britannica.com/art/polyphony-music>
7. MIDI
<https://whatis.techtarget.com/definition/MIDI-Musical-Instrument-Digital-Interface>
8. Launchpad Pro
<https://novationmusic.com/launch/launchpad-pro>
9. Unity
<https://unity.com/>
10. MIDI Tool Kit
<https://paxstellar.fr/>
11. MIDI Jack, Keijiro Takahashi, 2013-2015.
<https://github.com/keijiro/MidiJack>
12. Floating Object in Water, Denis Rizov, 2015.
<https://github.com/dbrizov/Unity-WaterBuoyancy>
13. Launchpad pro programmer's Reference Guide
https://d2xhy469pqj8rc.cloudfront.net/sites/default/files/novation/downloads/10598/launchpad-pro-programmers-reference-guide_0.pdf