

Multimodal Metronome — Rhythm game for musical instruments

Tuan Son Hoang^{1,*} and Michael Cohen^{1,**}

¹Computer Arts Lab; University of Aizu; Aizu-wakmatsu, Fukushima; Japan

Abstract. The goal of this project is to create an application that uses auditory, visual, and tactile elements to assist and guide musical instrument learners, using features available on the Nintendo Switch device and its paired “Joy-Con” detachable interface, to increase the novelty as well as intuitiveness of musical performance while stimulating user experience. By integrating visual music, spectral analysis, and motion capture, we obtain a metronome system that synchronizes visual, acoustic, and haptic display.

1 Introduction

1.1 Motivation

Music has been indispensable in human society across millions of years of formation and development. Music is a tool to connect people with each other. Like language, music has syntax, rules for arranging elements — such as notes, chords, rhythm, and intervals — into complex structures. Music is also a tool to express and affect human psychological state, conveying and inducing emotions. There are many people who have a strong passion for music, and they put that passion into action, of which singing or playing a musical instrument are prime examples.

Guitarist Richie Sambora once said: “Be your own teacher. Let life write your textbook.” Self-study is a natural learning method to quickly and maximize proficiency. Instead of taking musical instrument courses, people can study at home learning instruments they love to satisfy their passion. However, learning to play musical instruments is not as simple as learning to walk or speak. Practice often becomes a kind of competition, and stories are told of great musicians who practiced incredible hours each day. In contrast, there are people who give up after only a few hours because there are no clear direction or specific instructions.

There have been many devices and products that help learners and performers, such as metronomes, tuners, software, etc. There is also a large number of music- and rhythm- related video games for music enthusiasts. It would be interesting if there was a program that could help people who are passionate about music both practice and learn the instruments they love, and also serve as entertainment after a tiring day at work. Real action interactive rhythm games are an attractive trend; there are many successful developers with that type of game, such as Beat Saber [2], Virtuoso [3], Guitar Hero [4], and Just Dance [5]. With this background, we created a project called Multimodal Metronome.

* e-mail: s1262009@u-aizu.ac.jp

** e-mail: mcohen@u-aizu.ac.jp

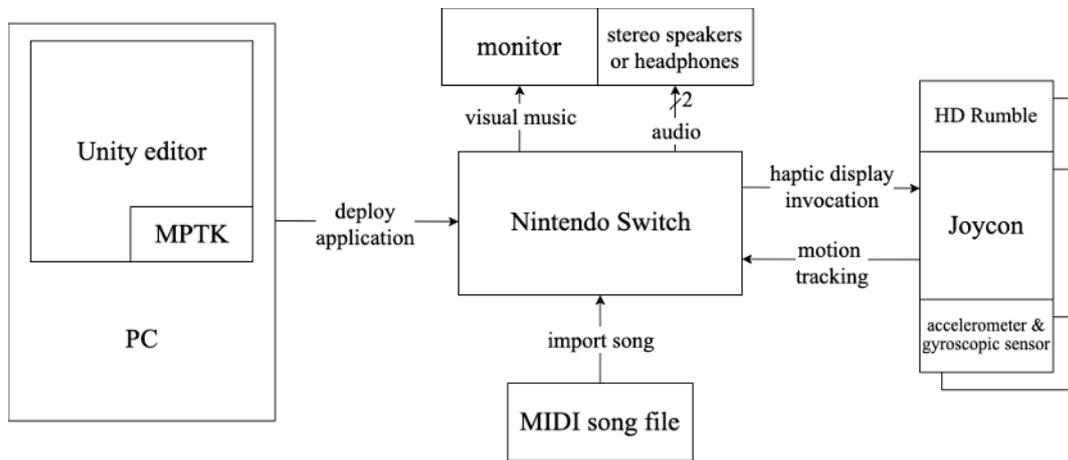


Figure 1. Software Functional Diagram

2 Methodology

2.1 System features

The architecture of our system is shown in Figure 1. Our application features these interface modalities:

2.1.1 Display (output)

1. Auditory

Read and play an audio source or MIDI file into an audio signal through stereo speakers, recreating the functionality of a metronome with audio.

2. Visual

Visualizing audio spectrum data, metronomic click track, or mo-cap rigging from an audio source or rendered MIDI file as explicit projection or abstract 3D “visual music” art scenes.

3. Haptic vibration feedback:

Using HD Rumble, Nintendo’s brand of the haptic technology, creating a tactile metronome experience with more subtle expression than first-generation haptic displays with only crude vibration.

2.1.2 Control (input)

1. Motion tracking:

Using built-in gyroscope and accelerometer on the Nintendo Switch’s Joy-Cons to monitor player movement and provide analytic during instrumental performance.

The primary goal of software development is that all functions in the system must be integrated. For example, when MIDI files are imported into the software, the software should render that MIDI file to an audio source as it provides tempo information to the metronome system. From there, the software converts the audio source into “visual music” on the screen of the Nintendo Switch. At the same time, the metronome system receives the tempo information of the MIDI file and converts it into a tactile metronome effect as well as an audible metronome effect while the song is playing. When the player starts playing the instrument along with the song, the motion tracking system monitors whether the player is playing in the right tempo, comparing information provided from the metronome system with data from the tracking system.

2.2 Technical Method

2.2.1 Platform framework: Unity

Unity [11], developed by Unity Technologies, is a comprehensive game and simulation development tool that allows developers to create interactive content such as 3D video games, architectural visualization, simulation, and realtime 3D animation. It features a fully integrated professional game engine for modeling physical interaction and rendering graphics and sound.

2.2.2 Programming language: C#

Unity supports only C# language programming scripting now, so for flexibility in the development of features, we use C# as the programming language.

2.2.3 MIDI file processing: MIDI Player Tool Kit

MIDI Player Tool Kit (MPTK) [8] is an asset for Unity that supports reading, playing, and rendering MIDI files, using MIDI messages and streams to trigger and modulate synchronous events.

2.2.4 Device: Nintendo Switch

The Nintendo Switch (seen in Figure 2) is a video game console developed by Nintendo, released in 2017. The device is a tablet that can either be docked in a set-top cradle as a TV console or hand-held as a portable device, making it a hybrid console. It has removable game controllers that can be used as companion joysticks for interactive games, much like the Nintendo Wii, but which also attach to the Nintendo Switch tablet itself [9].

In blurring the distinction between portable gaming device and console, the Nintendo Switch straddles some usual distinctions. Not only must “on-the-go” games be console-quality, but the Nintendo Switch must offer an online component to remain competitive against the Sony PlayStation and Microsoft Xbox platforms, which heavily feature networked functionality.

2.2.5 Controller: Joy-cons

Joy-Con Controllers (seen in Figure 3) are the primary controllers for the Nintendo Switch. As separate left and right units, the controllers attach to the sides of either the Switch’s screen or the Joy-Con Grip¹. They feature Wii/Wii U Remote-esque gyro/motion controls, HD Rumble, and NFC reader amiibo² support. Detached, they can be held in a player’s hands, strapped to one’s wrist, or even worn on the ankles.

¹Joy-Con Grip is a plastic frame that combines the left and right Joy-Con into one larger controller with this handy grip.

²Amiibo is a toys-to-life platform by Nintendo, launched in November 2014. It consists of a wireless communications and storage protocol for connecting figurines to the Wii U, Nintendo 3DS, and Nintendo Switch video game consoles.



Figure 2. Nintendo Switch (with attached Joy-Con controllers)



Figure 3. Joy-Cons (detached)

2.3 Audio Spectrum

The audio spectrum is the frequency range of auditory signals, including all sounds in human audible hearing range. To help players understand sound theory in general and music in particular, we developed a system called “Audio Spectrum Visualizer,” collecting data from frequency bands in an audio source and visualizing them as visible shapes on the screen.

Band	Bandwidth/Hz	Range/Hz
Sub bass	86	0 – 86
Bass	171	87 – 258
Low mid	343	259 – 602
Mid-range	687	603 – 1290
Upper mid	1375	1291 – 2666
Presence	2751	2667 – 5418
Brilliance	5503	5419 – 10922
Upper brilliance	11007	10923 – 21930

Table 1. Frequency bands

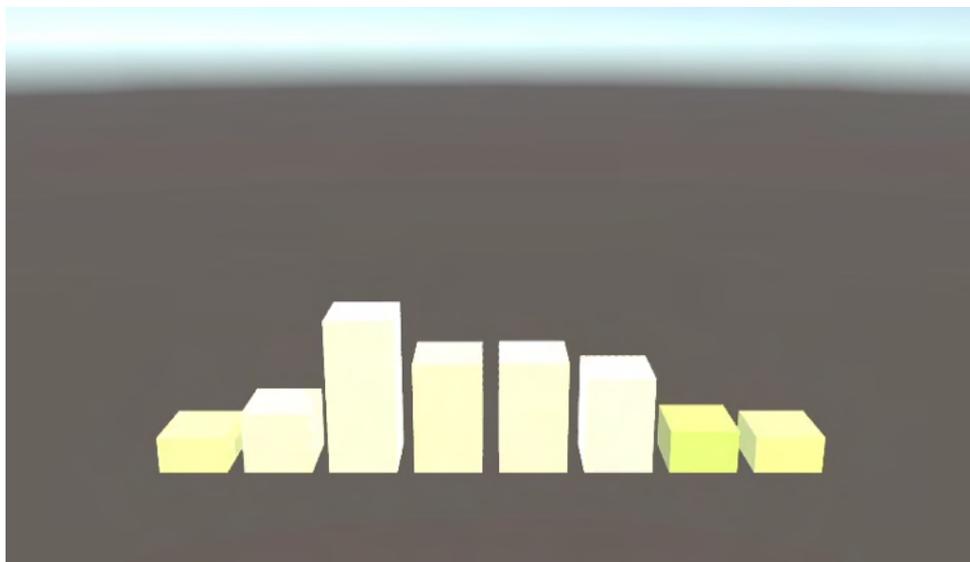


Figure 4. Real time spectral analysis

2.3.1 Visualization Experiments

Normally, the audible spectrum is a continuous band, ranging about 20 to 20,000 Hz. We divided this spectrum into eight bands, as shown in Table 1. After defining ranges of constituent bands as mentioned above, a template visualization was created using eight 3D objects representing each of the frequency bands. The scaled sizes of the objects are modulated by the amplitudes of the respective frequency bands. Figure 4 shows the spectral visualization.

2.4 Multimodal Metronome

2.4.1 Processing MIDI Player Data

MIDI Player Tool Kit can not only render MIDI files to audio, but also extract information about MIDI-encoded songs for various purposes. We extract the tempo data from metadata of the

song to configure the metronome function. By accessing the `MidiLoad` function we obtain the `MPTK_InitialTempo` parameter representing the tempo data of the song.

2.4.2 Auditory Metronome

Tempo is measured in BPM (beats/min.). When musicians use metronomes, they typically set them such that each click corresponds to a quarter note, so in 4/4 meter (the most common time signature), four clicks comprise a full measure. We imitate the sound of a real metronome by making “tok” and “tik” metronome click tracks during the music playing, the stronger “tok” sound represent each bar’s downbeat followed by three weaker “tik” sounds. The click track is stereo, so can be panned laterally (adjusting left-right “balance”).

2.4.3 Visual Metronome

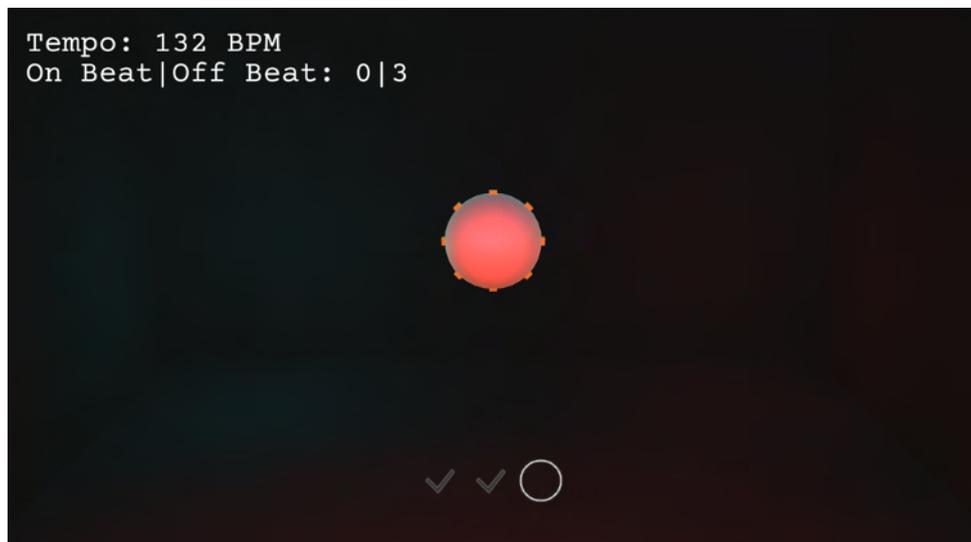


Figure 5. Visual Metronome

The metronome, beside serving as the pulsing clock ticks or beat track of the system, can also be visually displayed, as seen in Figure 5. Such overlapping cues (metronome displayed graphically as well as auditorily rendered) are a kind of synesthetic display, cross-modal expression for reinforced prompts.

2.4.4 Tactile Metronome

There is no operating system that supports drivers to access the Joy-Cons API besides the Nintendo Switch itself. Using `JoyconLib` framework, we can invoke the Joy-Con interface and create tactile display with `HD Rumble` vibration module available in both two Joy-Cons. By assigning the above function to the previously initialized metronome function we obtain a metronome system that synchronizes visual, acoustic, and haptic display. Because the program recognizes each Joy-Con separately, we can control both Joy-Cons at once based on their unique IDs.

2.5 Motion Tracking

To determine if the instrument player is hitting the right beat according to the metronome, we leverage motion tracking. Using JoyconLib we can access data from the gyroscope and accelerometer captured, exposed, and streamed by the Joy-Cons held in each hand. However, we found that when playing musical instruments, most of the player's performance movement is in the vertical direction, so in this program we focus only on the vertical motion tracking, because most of the remaining sensor data is almost redundant for our purposes.

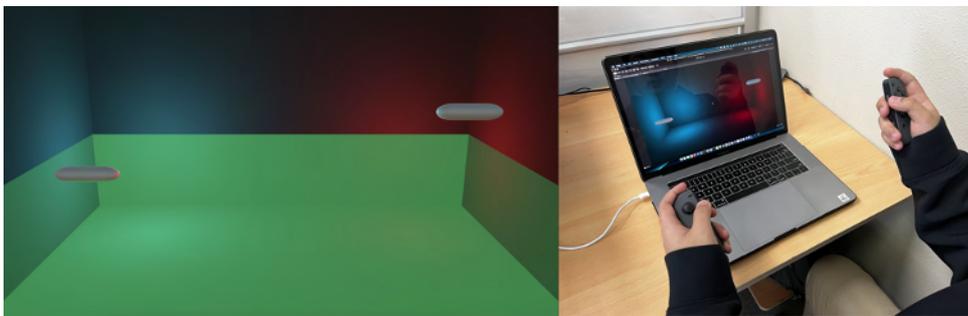


Figure 6. Hand-Tracking “puppets” or “digital twins” (rigged to separate Joy-Con controllers)

For more intuitive experience for players, we created two game objects, each representing a controller, as seen in Figure 6. As the player moves the controllers, the game objects also move according to the height of the respective controller. With this object we simplify the cadence testing of functionality by creating a boxy invisible game object so that when controller-rigged colliders acquire (come into range of) the targets, the program considers the event to be a struck beat.

The integrated application can be seen in Figure 7. To make it easier for players to track strike progress, some extra features indicate entrainment, such as a green light that illuminates when the player is on the right beat (“in the pocket” or “in the groove”), turning red when they're on the wrong beat. Our system records the number of right and wrong strokes of the player during a game.

3 Discussion and Conclusion

There are a lot of rhythm game products, and they can bring benefits to instrumental practice as well as entertainment. Applications like the Multimodal Metronome can potentially encourage musical study, exploration, and play. However, the Multimodal Metronome needs a lot of improvement. We hope that what we do will can inspire other developers in producing rhythm game products that are highly interactive with rich control and display.

In the future we want to develop Multimodal Metronome even more. Besides improving accuracy of the motion tracking system to help diversify the number of instruments that can be played, we hope to add new features such as spatial audio, recording and analyzing sound from the microphone for speech or singing, game control via physical buttons from Joy-Cons, and development of “eye candy” graphical effects.

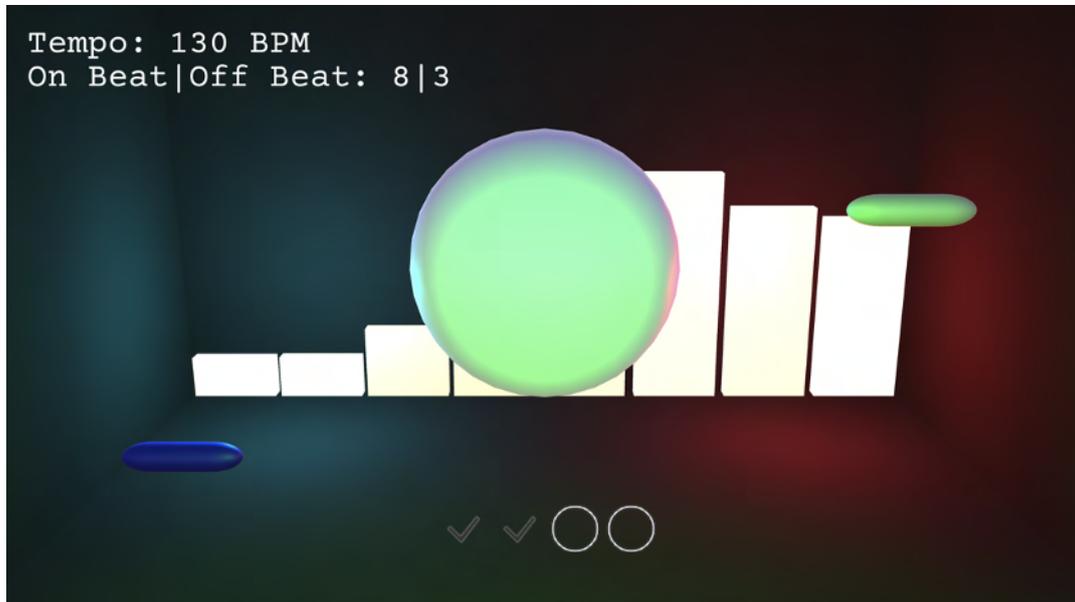


Figure 7. Integrated interface with visual metronome and music, spectral analysis, and rigged controller projection

4 References

References

- [1] Audio Visualization — AudioPeer, <https://www.youtube.com/c/PeerPlay>.
- [2] Beat Saber — Beat Games, https://en.wikipedia.org/wiki/Beat_Saber.
- [3] Virtuoso — Really Interactive Studio, <https://store.steampowered.com/app/1213710/Virtuoso/>.
- [4] Guitar Hero — Activision, https://guitarhero.fandom.com/wiki/Guitar_Hero_Wiki.
- [5] Just Dance — Ubisoft, <https://justdance.fandom.com/wiki/Home>.
- [6] Looking-Glass/JoyconLib — Github, <https://github.com/Looking-Glass/JoyconLib>.
- [7] “Pretty solid metronome for Unity,” <https://gist.github.com/filipruisl/eba3ffff4adde59889c0f836729db458>.
- [8] MIDI Player Tool Kit — Bachmann, <https://paxstellar.fr>.
- [9] Nintendo Switch — Wikipedia, https://en.wikipedia.org/wiki/Nintendo_Switch.
- [10] Nintendo Switch Reverse Engineering, https://github.com/dekuNukem/Nintendo_Switch_Reverse_Engineering.
- [11] Unity — Unity Technologies, <https://unity.com>.