L A S T M A R T E L É

A VIRTUAL REALITY APPROACH TO ANALOGIC VIOLIN PERFORMANCE

by

Sheng (Raymond) Liao

A Thesis Presented to the
FACULTY OF THE USC IMGD PROGRAM
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
MASTER OF FINE ARTS
(INTERACTIVE MEDIA AND GAMES)

May 2019

Table of Contents

Figure Page

Acknowledgements

Andreas Kratky, Jeff Watson, Carl Schnurr, Jane Pinckard

Alexis (Ming) Chen, Jose Kropp, James Maa, Joanna Lyu

USC IMGD MFA Cohort Class of 2019

At the beginning I want to thank all the people who have been devoting their precious time and giving meaningful advices and feedbacks to the Virtual Reality Project *Last Martelé*.

Professor Andreas Kratky approved of the impressive gesture implementation of the violin bow strokes and warned about the impossibility of using *Last Martelé* as a violin training application. Agreed by him, doing violin bowing gestures in Virtual Reality is an analogy of performing true violin. He also discussed with me about the potential strength of the narrative behind *Last Martelé* which lacks resources to be unfolded and implemented. Professor Jeff Watson mentioned the alternative physical controller used as the violin fretboard, discussed with me the principles about violin instrument and music theory as the cornerstones of *Last Martelé* as well as gave me much encouragement during the production phrase. Carl Schnurr played the prototype of *Last Martelé*, compared it with the celebrated music rhythm series *Guitar Hero* and emphasized the significance of User Interface for music interactive products. Professor Jane Pinckard gave similar feedbacks about the infeasibility of instrument-training design goal as achieved by the educational music game *Rocksmith* for *Last Martelé*.

Alexis (Ming) Chen from Engineering School of Viterbi contributed to the design scheme of User Interface and offered alternative plans of the *Last Martelé* shooting combat system waiting to be implemented. She is also supportive of the narrative untold in this project and introduced two friends as teammates. Jose Kropp from Thornton School of Music as a professional violinist showed his passion and fantastic violin bowing techniques he obtained through years of training as inspiration of potential design for *Last Martelé*. James Maa from Engineering School of Viterbi participated the planning for transplant of the project from Unity3D to Unreal Engine 4 (which has more mature Game Artificial Intelligence system and built-in MIDI support). Joanna Lyu from Interactive Media and Games Division of USC School of Cinematic Arts spent her time reading through the three scripts planned to be combined and reorganized for the narrative and left useful comments.

Thank all of my MFA cohort members of IMGD 2019. I have spent unforgettable three years with you during which I learnt so much and grew into a better designer and a person in general.

Abstract

This document expounds on the design and production process of *Last Martelé*, an on-going Oculus Virtual Reality project which aims at providing players a novel experience of performing analogic violin bowing to play songs in an enclosed, safe virtual space. Four sections regarding the ideation, preproduction and production phrases of the project will be presented for the readers. Section One will state three previous arts from the fields of commercial advertising, Virtual Reality use-case and culture-specific video game content as inspirations for *Last Martelé*. Section Two will clarify the personal motivation of research on the project and the study limitations. Section Three focuses on the technical details that are essential to make the project happen. Readers who are not interested in technicalities can skip the words. Nevertheless, quintessential information about the methodology for dealing with the design feasibility for *Last Martelé* will be rephrased in a legible way. Section Four, on the other hand, describes the reasons for the relatively scarcity of playtesting already being performed and leads to the next developing step.

Keywords

Interactive Virtual Reality, Violin Performance Analogy, Digital Music, MIDI Visualization

Introduction

Lived in Sichuan Conservatory of Music, one of the well-known interdisciplinary music institutions in China, I was surrounded by pianists, violinists, musicians of various ethnic musical instruments, vocalists and musicologists but never got a chance to formally learn music, the most efficient form of art and language to convey emotions, connotations and intentions. Even though accepted several years of training about instrument techniques, I still remain as an illiterate music amateur and never stepped into the sacred territory of this powerful means of communication.

To acquire language of music, one must first master its specific carrier. Though it usually takes years of fatigue training and requires big amount of financial support to obtain music education in ear-training, musical instrument or even composing, designers and technicians who buried seeds of rhythm, melody and harmony in their hearts are always trying to lower the threshold for novices and convince them to use economical but efficient educational tools. Thus, there are hardware synthesizer to substitute giant piano along with software and apps such as *Earpeggio* to guide listening.

As an interactive designer and software engineer, I want to emulate those pioneers, to use the knowledge and skills I obtained to explore something, something that established on the solid foundations of others' wisdom and creativity but makes a little more sense. This thing should give a chance, a reason and instant feedbacks to those who are curious but feel self-doubted, embarrassed or lacking of entry to start trying to practice music. It should be a music game.

Section I. Prior Art Review

Neither are there "meta" ideas, nor can a designer create their work totally based on their own imagination. A design principle of mine is seeking for underlying connections among objective things or others' impressive thoughts and rephrase them through the logic formed within my limitation of cognition. Three critical design choices are borrowed from a commercial advertisement, a VR interactive product and a classic screen-space video game from the ocean of innovative contents and transformed into a specific viewpoint of Virtual Reality technology musical educational use case.

A Pantene Commercial with Narrative about a Violinist

Simple character arc, bright contrast of good and evil, dramatic plot, pure motivation, classic music piece, Thai Commercials are good at giving warmness to advertisements through storytelling. *PANTENE: A Very Touching Deaf Violinist Commercial* (Vergara) shows a miracle of a Thai girl with hearing disability proving her violin genius against her haughty pianist partner. Even though the foreshadowing in the *Commercial* there is for showing the deaf protagonist's shining long hair that only has a loose connection with her shining violin contest fightback, the lucid gesture of the girl's mentor when he is trying to encourage her also lighted a bulb in my mind. It is the gestures the embodiment of violinists' steadfastness in obtaining the formidable violin techniques through persistent practice. It might also explain why *Last Martelé* has to be about violin performance: the

elegant gestures of bowing beat all those of string instruments that can generate similar euphonic tones.

Figure 1. Screenshots of the Pantene Commercial on YouTube.

A VR Guitar Simulator Exploiting Alternative Controller

"You can use alternative controller for your personal show in an art gallery." [1]"If I want to use your product to gain useful knowledge for playing true violin, I want to see strings, rather than buttons."[2] But to be honest, I did not even recognize the importance of physical props for projects like *Last Martelé* before I tried *Rockband VR* by myself. The game contents themselves are not suitable as references for this thesis project. All of the loud rock and roll, restless virtual fans and dazzling particle visual effects are fine, but it becomes some-how boring after a while as one flips the switch on the instrument prop body like a kid and clumsily presses chunks of button standing for guitar fretboard. However, the exclusive hardware device offers an intuitional interaction for its users who may never play true guitar. This brand uses Oculus VR left controller as a tracker mounted on the plastic guitar controller's headstock. Rather than provide

[1] A rephrased comment from Jeff Watson, Ph.D. Assistant Professor in the Interactive Media and Games Division of USC School of Cinematic Arts when I was showing him the prototype version of switching violin strings in *Last Martelé*.

[2] A rephrased comment from Andreas Kratky, Ph.D. Associate Professor of Cinematic Arts and Chair of Media Arts and Practice Division during one of our thesis meetings.

players experience of genuine guitar performing, the alternative controller still makes me feel authentic owing to the consistent presentations of the guitar toy in virtual and real worlds.

Figure 2. Photos of Oculus controller(left) and the alternative controller used in Rockband VR (right).

A Japanese Survival Horror Series Combating through Picture Taking

Did you ever experience a game in which you acting as the protagonist need to defeat antagonistic evil apparitions through taking photos of them? There are a number of cultural elements serving for storytelling and world-building in the Japanese *Fatal Frame, titled Zero* horror video game series: sisters as protagonists to reflect the eastern tale of "telepathy" phenomenon between twins, "Spirit Stone Radio" that reflects a supernatural phenomenon of receiving voices from the nether world and so on. Leave the narrative behind, *Fatal Frame* series contains connotations about fear and taboo by selecting an unusual designing choice of a camera weapon and the matching User Interface mode of "ViewFinder". Taken the series as a reference to gamify the general use of the virtual violin instrument, a potential development path of *Last*

Martelé is to establish the mechanics of performing violin bowing and generating soundwaves to fight with illusions imagined by the protagonist. To further rationalize this magic effect, narrative designers can put the protagonist into a situation that she suffers stage fight and is extremely afraid of the judges of her instrumental performance, taking them as an evil power she needs to conquer.

Figure 3. Screenshots of the "Spirt Camera" in Fatal Frame series (left) and the "ViewFinder" mode.

Figure 4. Previous poster to illustrate the concept of using violin as a weapon in Last Martelé.

Section II. Purpose of Study

Aside from gamifying the unique violin-performance mechanics using system design and narrative, this MFA thesis paper will focus on discussing the methodology of dealing with the tortuous design and implementation issues about the bowing gestures in VR, digital audio integration in Unity3D game engine and User Interface for violin fingering guidance that I met during the preproduction and production phrases in developing *Last Martelé*.

"Your goal should be for each page or screen to be self-evident, so that just by looking at it the average user will know what it is and how to use it." (Krug) [3]Indeed, a bunch of products not limited to Websites on the markets fuse Steve's User Experience principles and offer their customers comfortable User Interfaces to interact with. Nevertheless, one may not understand the complexity behind the conciseness unless they start to create an innovative mechanics and its UI system. It is the situation I face throughout creating *Last Martelé*. Not a violinist or a professional musician, I tried my best to research on some basic relevant theoretical knowledge such as "the strokes of the bow must describe a straight line"[4] or "Circle of Fifths shows the relationship of one key to another"[5] in order to seek for obvious (not for a non-violinist) principles and design potential. In a nutshell, I am seeking a balance of simplifying the violin bowing and fingering but still keeping their professionalism. At the same time, try to make players of this project believe they are performing the string instrument.

[3] See *Don't Make Me Think*.
[4] See point 2 on page 3 of *Practical Violin Method*. (Hohmann)
[5] See the details on *Alfred's Essentials of Music Theory*, p53. (Andrew Surmani, Karen Farnum Surmani, Morton Manus)

In order to concretize the design goals, two essential design tasks must be discussed from the beginning: the gestures of bowing recognition and a User Interface for this project that is acceptable for guiding novices who have no previous experience on the instrument.

Innovative VR Interactivity in Analogy with Musical Performance

Some foundational violin performance knowledge should be declared for the readers of this document before arguing about the methodology of research and experiment for *Last Martelé*.

If looking at the physical violin headstock and fretboard parts on the lower left illustration, one is able to see four strings without “frets” like those on a guitar. The four strings of my violin used for the study from left(thickest) to right(slimmest) in the photo vibrating separately to generate Low G(Sol), D(Re), A(La) and High E(Mi) under open strings. Aesthetically, violinists swing their bow arms as their hands draw graceful curves while playing mellifluous notes. In addition, the other hands of them keep busy at pressing the string the bow is attached on to shorten its section able to vibrate. The hardest skill for them is pressing at a precise point on the string for generating an exact pitch. On the other hand, they must move their bow hand in an arc path up and down to switch the vibrating string if the previous string does not contain the pitch range the current note belonging to.

"Can you let a player interested in your *Last Martelé* play a scale or *Twinkle Twinkle Little Star*?"[6] Indeed, playing an elementary sheet music can be daunting using violin, considering all those bowings and fingerings combinations discussed above. It was also indistinct what the whole system would be like that supports the intuitive gameplay of performing the project is aimed at when I got start. If one analyzes the performance of a violinist carefully, they can notice that the movement of the bow can be quantified along with two "axes" in the air. When moving perpendicularly to a string, the instrumentalist makes a sound; moving up and down, the player can switch strings. This statement seems to be self-evident, but our computers that try to simulate the continuous motion do not think so. Check the (Liao, Raymond's MFA Thesis Last Martelé - String Switching) explaining the quantitative analysis and technical details behind the high-level gesture design.

Figure 5. A photo and Screenshots of the violin instrument research(left) and animation for string-switching breakdowns(right).

[6] A rephrased comment from Carl Schnurr, Adjunct Assistant Professor in the IMGD Department of School of Cinematic Arts, while we are discussing about the project progress.

The other mechanics of fingering that truly complicated in real world is effortless for Oculus Virtual Reality, in the developer's perspective. If we glance at Oculus Left Controller, the two buttons (X, Y), the Trigger and the Grip can be mapped as the four points on a string representing natural notes finger positions (such as F, G, A, B on the rightest E string)[7]. The Thumb-stick is potential for the vibrato [8]effect, as integrated on some brands of Sound Synthesizers or MIDI Keyboards. Plus, if we added in flats and sharps, another button on the right controller could be combined in the input mapper. Nonetheless, the input combinations can be rather complicated for users and increase the difficulty of User Interface design. Furthermore, "the button layout of Oculus Controller is quite arbitrary"[9]. Without a tutorial in the executable, even violinist play-tester can get confused and frustrated in that those buttons are not arranged on a plane as those finger positions on a string.

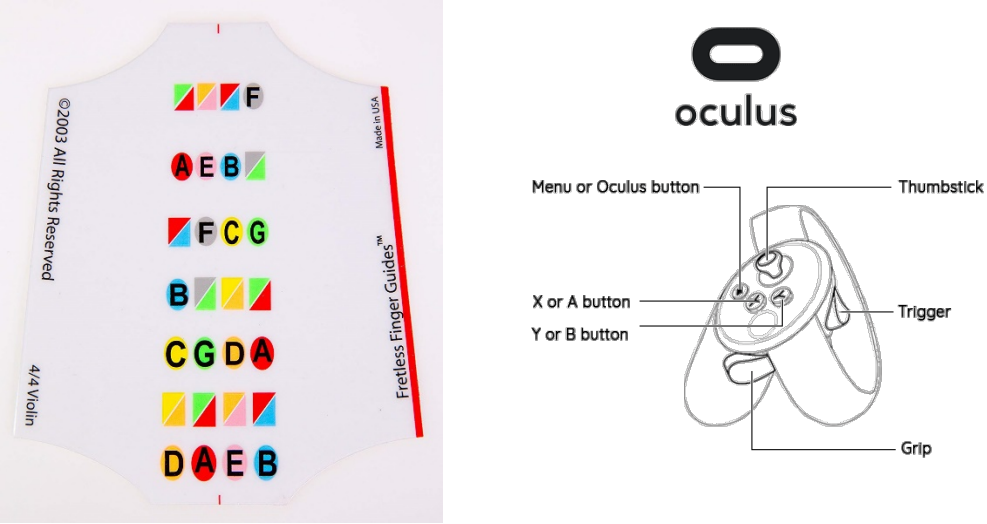


Figure 6. Key Mapping from Fretboard to Oculus Controller.

[7] This mechanics was implemented in the second version (Liao, Raymond's MFA Thesis Last Martelé Bowing and Fingering) of the prototype.
[8] Check *How to do Vibrato*. (Howcast)
[9] Cited as a comment from professor Andreas Kratky.

Improvement of UI System for Music Interactive Products

General music digital products such as the virtual MIDI keyboard of online software *MuseScore* or video game *Deemo* all utilize similar mechanics of "falling blocks" to indicate the upcoming notes in a song. Once those chunks touch the bottom, music notes will be generated. The mechanics is so abstract that it has nothing to do with physical keyboard instruments. One guess for this phenomenon is that it may originate from the time indicator scanning through MIDI tracks as the interface of some traditional Digital Audio Workstation like Ableton does. "If we find something that works, we stick to it."[10] Product customers are tend to use what they are already familiar with no matter how badly it is. The Interface solution is okay for virtual piano, but by no means can it be transplanted directly to *Last Martelé*. On the one hand, Acoustic piano is inherently percussion instrument, which is rather distinct from string instruments like violin. Although a note can still be represented as an elongated block for a bow stroke, the wide difference caused by one-dimensional layout of keyboard comparing with two-dimensional finger positions on the four strings still make the matter complicated. On the other, no interface of a music video game precisely indicates musical notes. They are either arbitrary falling abstracts like *Deemo* and *Guitar Hero*, or icons with symbolized meanings such as *Taiko no Tatsujin* and *Nisan Shaman*. Above all, all the rhythm games contain only the rhythm aspect of a music piece and have no concern with other basic musical elements (pitch, interval, etc). *Last Martelé* aims at pushing the situation on the market a little further so it needs a system to display UI elements for approaching to music notation.

[10] See *Don't Make Me Think* (Krug)

Section III. Design and Technical Solutions

This section discusses various solutions tried out during *Last Martelé* production phrase intent on addressing the design questions raised in the previous section: how to analogize violin bowing and fingering using Oculus Rift controllers, how to generate audio for the virtual instrument, what system is needed to be implemented in order to support the User Interface as well as what an appropriate UI system would be like. Some topics like MIDI file interpretation are considerable but rather complex and heavily-engineering-oriented[11]. So that no relevant code snippets will be presented. It is not the purpose of this paper. Readers who are interested can check details on *Sanford MIDI Toolkit* (Sanford), *Melanchall DryWetMIDI* (Dobroselsky, DryWetMIDI: High-Level Processing of MIDI Files) and other cited website pages.

Gesture Recognition of Violin Bowing

Not mentioned in the Prior Art section, the *Virtual Reality Longbow (The Lab)* experience contributes both conceptually and technically to the formation of this project. The "arrowing" mechanics is akin to the bow stroking on single violin string: one can hear the virtual wooden body creaks while drawing the bow, feel the authentic haptic feedback generated from sampling of the recorded sound effect audio clips, and visually see the arrow procedurally nocked on the bow through position and rotation interpolation. Let us put it this way, the violin bowing mechanics implemented in *Last Martelé* is inherited from *Longbow*.

There is a SteamVR Unity3D package containing the prefabs and source codes that demonstrate a programming example of Virtual Reality Archery. See the left part of the figure below, three

[11] That is why I spent almost two months on figuring it out.

transforms of “empty game-objects” marked in yellow texts are used for denoting the arrow its tail and nock docking points on the arch and the string. Similarly, there is an object rendered as cyan spherical gizmos near the center of the strings that is used for measuring the distance between the nearest point on the virtual violin bow and this dummy helper. Once the distance shortening below a threshold it stands for the bow’s eligibility of being attached on certain string. Then the program calculates the nearest one of those rendered as yellow spherical gizmos from the bow to determine which will be the current attached string.

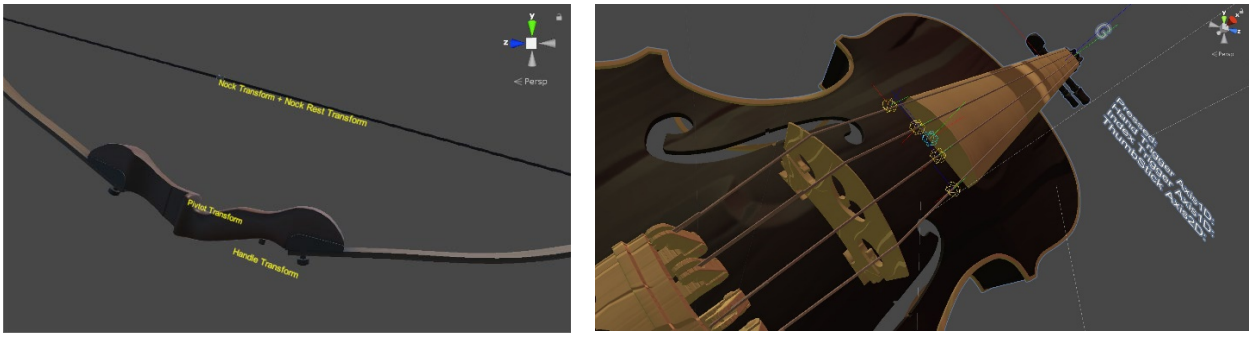


Figure 7. Stroking Mechanics Implementation Similarity between LongBow and Last Martelé.

As readers can see in the sequence of screenshots below, the distance and angle quantities are displayed for debugging, as well the “current string” highlighted in red. To make it obvious, there is a giant circular UI (circle of fifth) in the air indicating the note that will be played if the bow moves. In this prototype the algorithm gets the relative position and orientation of the two Oculus controllers for calculation and conditional judgement.

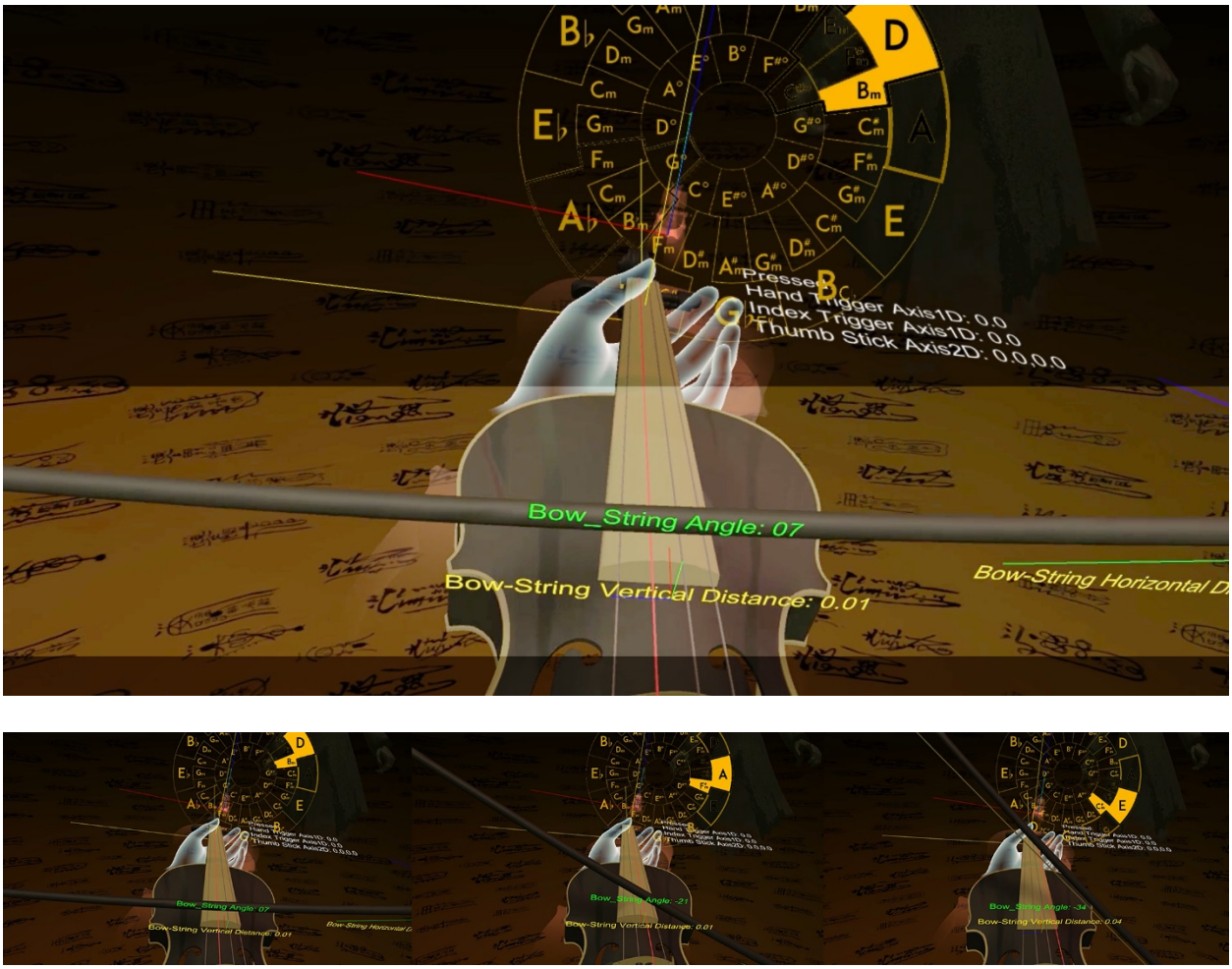
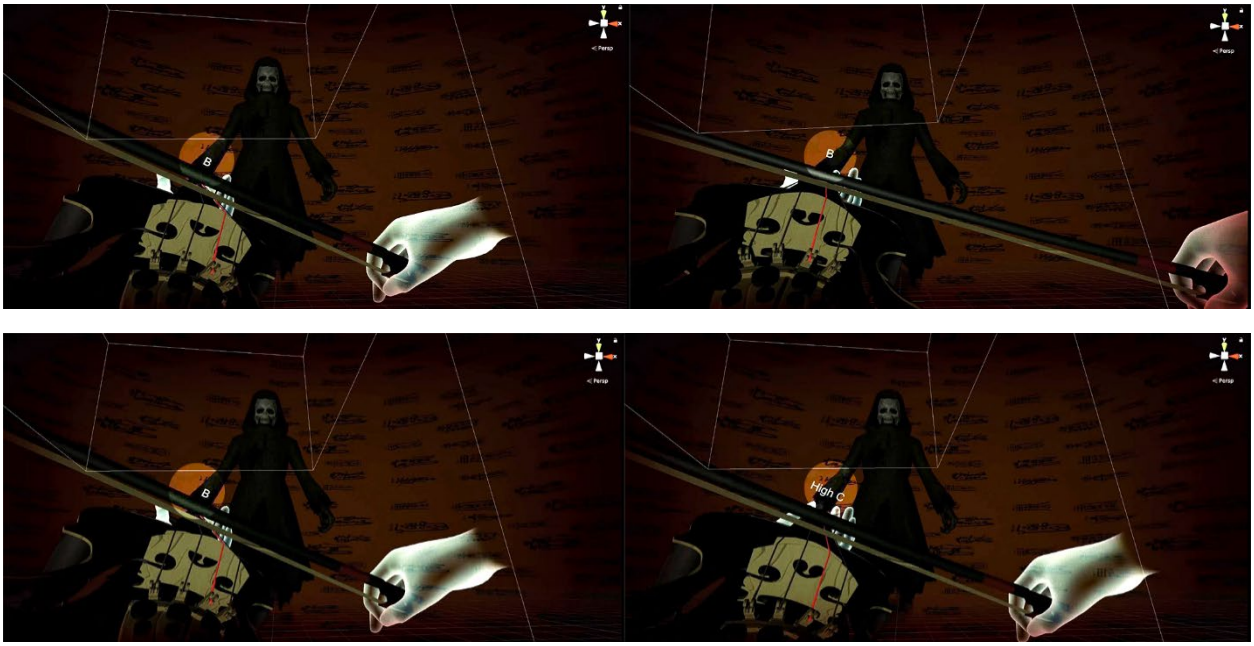


Figure 8. String-Switching Mechanics in Last Martelé.

The next step is to implement the bow stroking for generating note sounds. Mathematical interpolation is a good friend of smoothly moving and rotating game objects, and it is also a helper for violin bowing simulation. This version of the mechanics prototype includes features listed as follows:

- When the violin bow is near certain string it will be automatically attached to the string. No matter how the player moves the bow the contact point between the bow and the string will stay the same.

- Only when the bow is attached on certain string and the player is pressing the Index Trigger of the right Oculus controller will they be able to play notes through moving the bow.[12]
- While the player is playing notes the violin body will rotate aligning with the direction from the bow root to the contact point. Thus, the bow will never "slip" away from the string unless the player releases the right Index Trigger.[13]
- The player needs to press the Index Trigger, the Y button, the X button or the Hand Grip to play different notes belonging to certain string (simulating pressing the string at different finger positions).



[12] During the playtesting, this feature is quite annoying for players and will be discarded or altered.
[13] Same as (12), this feature is also borrowed from *LongBow VR* and renders the violin body rocky so it will be altered later.

Figure 9. Stroking and Fingering Mechanics in Last Martelé. The first row of screenshots shows a down-bow movement of a stroke, whereas in the second the "B" and "High C" UI Texts visualize the "Ti" and High "Do" musical notes played by pressing the Trigger and Y button of the left Oculus controller separately while the virtual bow is attached on the third(A) string. Finally, the third row shows the bad feature of aligning the violin body with the rotation of the bow that renders the violin body rocky.

Readers can check the recorded video (Liao, Raymond's MFA Thesis Last Martelé Bowing and Fingering) to get an intuitive feeling of the implemented mechanics. The violin sounds are play-backed from sampled audio clips, which is not good enough for this project. It is okay to use audio clips only for the virtual violin, considering the limited size of memory spaces those audio samples occupy while the game is running. Nonetheless, one design goal is to make users to play songs using the instrument. It would be the best if the player can practice any song in different skill levels as they would throughout practice of an instrument. Never can it be achieved if we only use audio samples.

Synchronization of Music Notes and Visuals

If one wants to practice an instrument, they must know what note to play. Taking research on how to play-back any song and synchronizing UI elements to guide users the precise notes duration and pitch is ciritcal for this project. Several existed solutions are inquired during the Audio-Visual synchronization development phase from January to March, 2019.

Method I. Event-Based Beat Matching

We can guess that most of music video games used this way to generate UI elements. As can be seen in the left part of the screenshot below, several markers highlighted in green are placed aligning with audio beats in the editor panel of the popular Unity plug-in *Koreographer*. Whenever the time indicator reaches a marker, an event will be called to run certain part of the codes. Apparently in this way the program gets nothing of the information about what the note is in the audio track. *Last Martelé* is looking for a similar pillar supported by another plug-in called *RhythmosEngine*[14], as can be glanced on the lower right. At least the program needs to know what the musical element is at this moment.

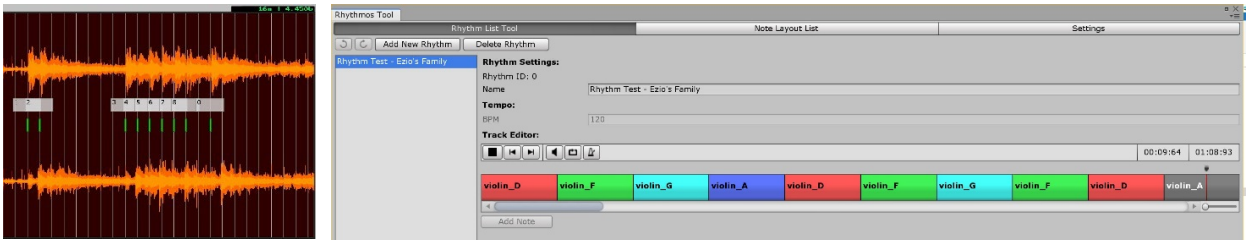


Figure 10. Koreographer and RhythmosEngine Plug-ins in U3D.

Method II. Music eXtensible Markup Language

This file format contains much more information about the note itself: its interval, octave, pitch, duration, etc. Besides, it occupies less hardware resource. However, it is very lengthy and time-consuming to write a parser for any xml file, as described by an experienced music software developer Dominique Vandenneucker: "To draw the content of a sheet of music directly with the Windows and Mac graphic API requires quite a lot of programming work."[15]. More importantly, it can only be used for drawing visual elements like sheet music or blocks of notes but have no control of digital audio. In another word, it cannot play-back song. Readers can get a sense of

[14] Unfortunately, the plug-in is compiled and not open-source, and the features offered are rather limited.
[15] See the comment through accessing his website either on page (Dobroselsky, DryWetMIDI: High-Level Processing of MIDI Files) or (Dobroselsky, DryWetMIDI: Working with MIDI Devices)

what node sections are like in a MusicXML file on behalf of the half notes D(Re) and F(Fa) in the following screenshots.

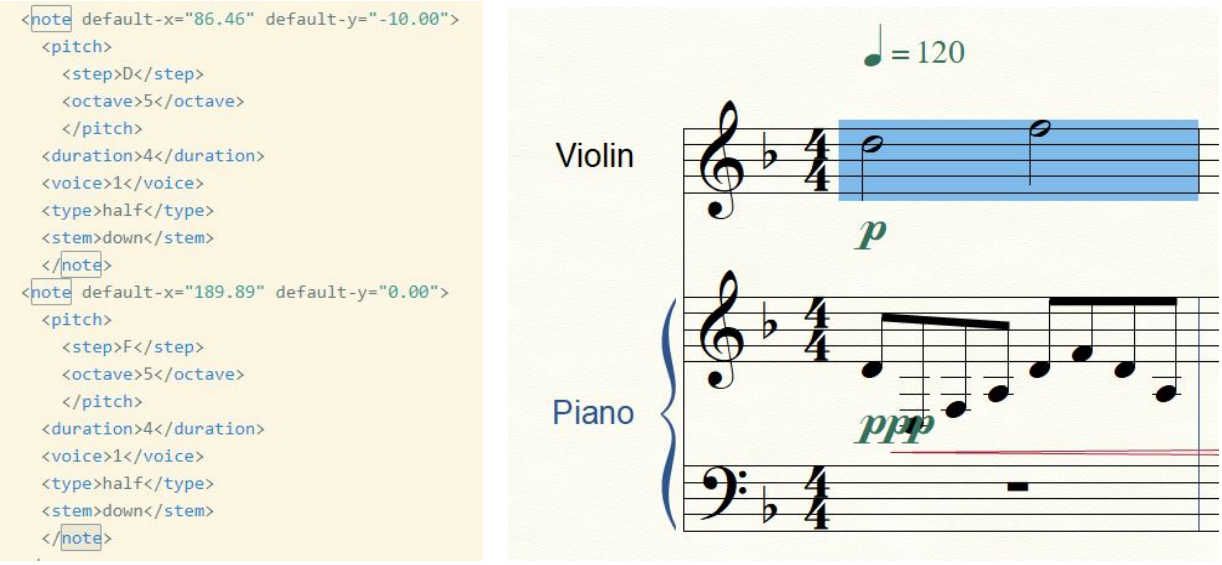


Figure 11. MusicXML file (left) exported from Finale 2014 (right). The two "note" nodes containing step nodes of "D" and "F" from the xml file are used by the software to draw the "Ri" and "Fa" half notes on the violin part treble clef.

Method III. Musical Instrument Digital Interface

There can be a whole another thesis paper (not for MFA degree though) talking about MIDI. Generally, it is a technical protocol for the communication between different digital instruments. A form of file storing the MIDI words, with the suffix ".mid", contains the information of the communication which is called MIDI message. Those messages are transmitted from a MIDI input device like a MIDI keyboard, to an output device such as a software synthesizer integrated into a Digital Audio Workstation (DAW). One intimidating fact about MIDI file, though, is that it is in binary mode, as a message represented in hexadecimal format. Readers can glance at what the messages that stand for the beginning and the end of certain note look like:

Note-On message: (1001 - 0x90 for Channel 0)

- Status byte: 1001 CCCC - CCCC=MIDI channel (0-15)
- Data byte 1: 0PPP PPPP - PPPPPPP=pitch (0-127, Middle C = 60/0x3C)
- Data byte 2: 0VVV VVVV - VVVVVVV=velocity (0-127)

Note-Off message: (1000 = 0x80 for Channel 0)

- Status byte: 1000 CCCC
- Data byte 1: 0PPP PPPP
- Data byte 2: 0VVV VVVV - VVVVVVV=release velocity

We need some interpretations of these obscure machine words. As one can see, a "Note On" message has three parts, named the status byte, the data byte1 and the data byte2. The last four digits of the status part tell the machine which channel within the message is transmitted, as messages in the same channel control one virtual instrument. The other data parts, meanwhile, carry the information of how high (pitch) and loud (velocity) the note will be played. Except for the "Note On" and "Note Off" MIDI messages listed above, messages such as "Program Change (to select instrument) or "Pitch Bend (to create a cartoon-spring-like sound effect) are differentiated by the first four digits of the status part and may contain one or more data parts.

Unintelligible as they are, the MIDI messages are compressed so densely that it can be a nightmare to restore them back to a melody. Fortunately, many materials talk about MIDI[16] online. Some talented engineers even developed MIDI Toolkits written in CSharp using .NET framework, which are exactly suitable for a Unity3D project. Two of them are precious for the development of *Last Martelé*, called *Sanford MIDI Toolkit* and *Melanchall DryWetMIDI*. The former one contains a demo with User Interface but is less powerful than the later, as it offers high-level abstraction and encapsulation of data as musical elements such as note, chord, scale, mode, etc. Thus, it makes no sense not to use this standard in the field of digital audio and music.

[16] As listed in the Bibliography of this document.

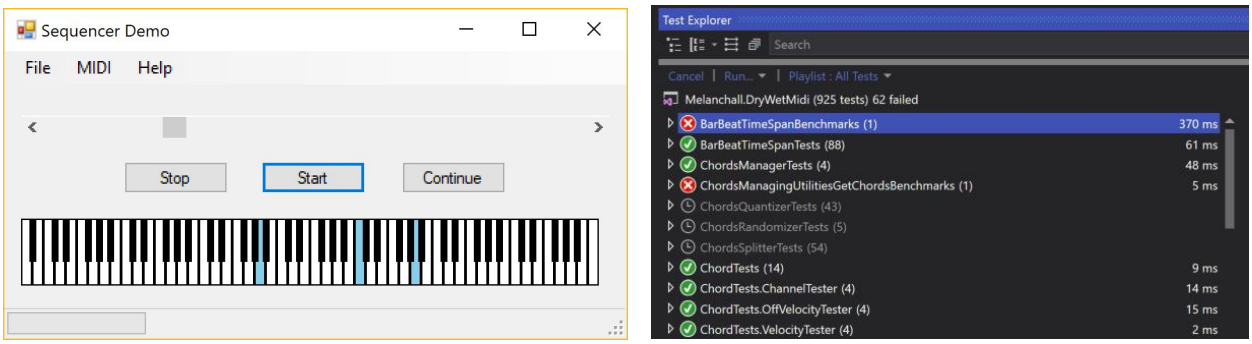


Figure 12. (left) Sanford's MIDI Software is playing-back a song; (right) NUnit Tests running on DryWetMIDI.

In a word, MIDI is the best choice for *Last Martelé*. It is worthwhile to recast a music software containing great features to utilize MIDI power as well as written in tens of thousands lines of codes. At the current development stage, the MIDI system has been successfully integrated into Unity3D for this project. As can be seen in the following screenshots and (Liao, Raymond's Last Martelé MIDI System Integration), the system can receive MIDI messages from input device such as a computer keyboard, a MIDI keyboard or the virtual violin. It can also send out messages to the software synthesizer installed on the operating systems of users' computers to play-back MIDI file, synchronizing audio and the User Interface elements that represent true musical notes[17]. More importantly, only through muting the sounds of the violin part from a song while displaying the synchronized, visualized violin notes can we achieve the goal of guiding the users playing the virtual violin by themselves. With the MIDI capability of filtering out the notes in the "violin" channel discussed before, it finally comes true.

Another note here is that the MIDI messages are controlling a software synthesizer and cannot generate audio by themselves. Besides, the built-in software synthesizers like *Microsoft GS*

[17] Its pitch, duration, velocity, etc. all can be visualized.

Wavetable synth in Windows operating system might create terrible sounds. Thus, users might want to install an alternative one such as *Coolsoft Virtual MIDI Synth*[18] for enhancing sound quality. Please get more information on Donya Quick's Website (Quick).

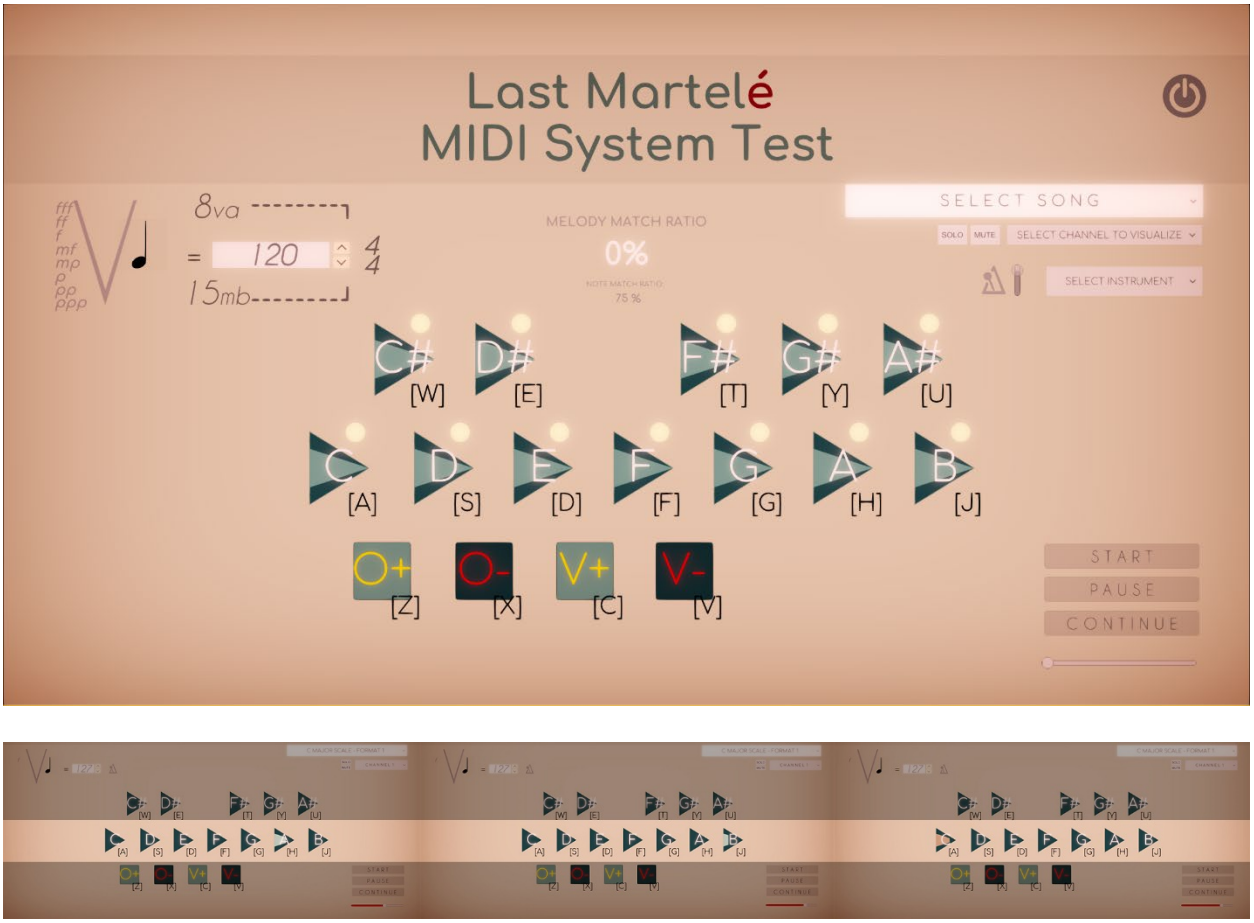


Figure 13. MIDI System integated in Unity3D for Last Martelé. The lower images show notes played in sequence.

A Compromised User Interface from Oculus Buttons Layout

Expected as usual, it takes amounts of efforts and time to set up an acceptable, if not effective and friendly yet, real-time User Interface based on the MIDI system of *Last Martelé*. Before shouldering the heavy burden of extra engineering and Human-Computer Interaction iterative

[18] on Windows only

designing tasks, we might utilize the paper prototyping method first. "(Through paper prototyping) It's much easier to justify spending the resources to realize a game in software after the game framework is developed and refined through more cost-effective means, such as analog prototypes." (Fullerton) If the potential users still fail to understand the designed UI drawn on paper after being explained, never can we expect it will guide them through real-time virtual musical instrumental performing. On the other side, we assumed it can cause lots of confusion to those who have not yet obtained music notation education[19], if we utilize musical symbols at the beginning. Thus, a compromised designing solution was drawn in Adobe Photoshop, combining abstract progress-bar like components mapped from the layout of Oculus controller buttons and note names.

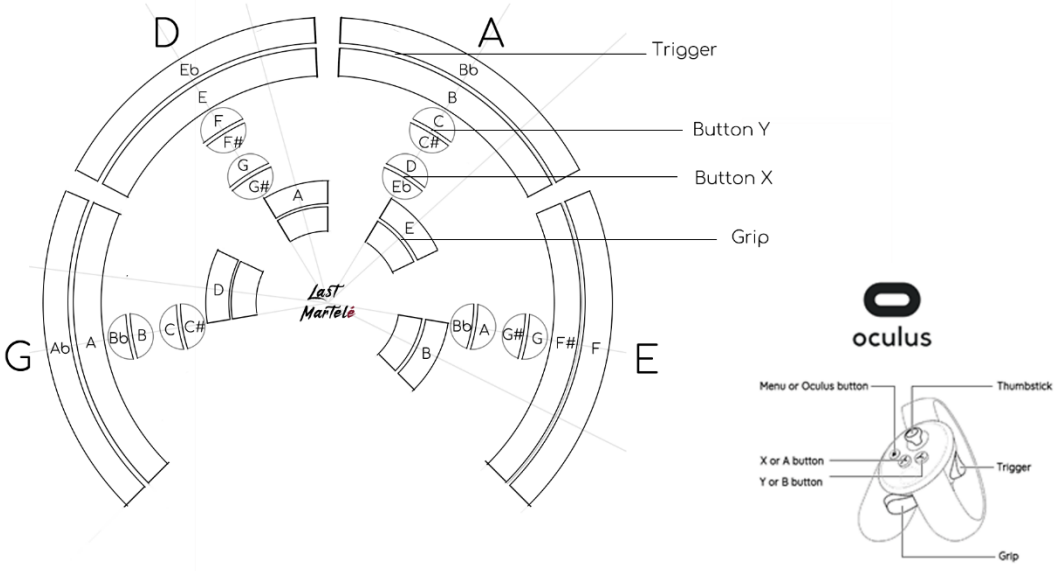


Figure 14. A UI designing paper prototyping solution for Last Martelé, using the combination of abstract diagram and note names.

[19] Some audience complained the abstruse symbols on the MIDI System interface presented in the previous section, after watching it on the screen while listening to my thesis defense.

As can be seen in the diagram, four fan-shaped sections stand for the four strings of a violin, within each eight abstract figures in various shapes are divided into four groups. The eight block figures mapped from the shapes and layout of the Trigger, the Y button, the X button and the Grip of a left Oculus controller are used for visualizing the eight musical notes during a MIDI playback or a player's bow stroke, whereas they being grouped is due to the limited buttons of the controller.

The following sequence of images as a flow diagram illustrate the process of this User Interface visualizing a B(Ti) musical note played-back in a MIDI song and certain user's subsequent bowing performing on the virtual A string along with pressing the Oculus Trigger analogy with fretboard fingering on a top position of the string, shortening it for generating a B note.

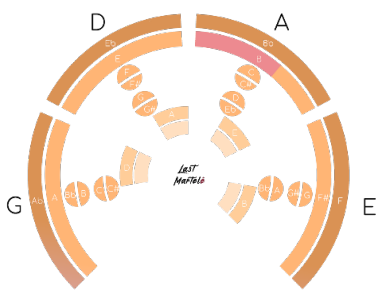


Figure 15. UI Paper Prototyping – Diagram 1. A B(Ti) MIDI note is on during certain MIDI playback, which is marked by a conspicuous pink progress bar, overlain on the orange block of B musical note. The lifetime of this filling UI component is the same as the length of the note.

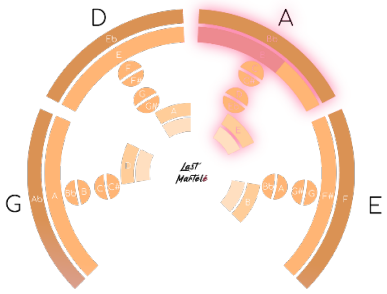


Figure 16. UI Paper Prototyping – Diagram 2. The third fan-shaped section is highlighted in pink, indicating that the user has being rotating the Oculus controller for bow into the angle range that allows the bow attached onto the virtual A string, preparing for stroking.

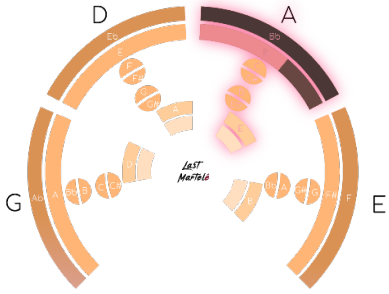


Figure 17. UI Paper Prototyping – Diagram 3. The top two note blocks turned into dark brown as the user/left-hander pressed the Trigger on the Oculus left controller. This emulates the A string being shorten through the corresponding point pressed down by a violinist's finger. Both are darkened to reflect the fact that they are controlled by the same controller button.

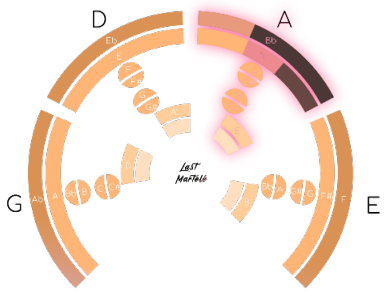


Figure 18. UI Paper Prototyping – Diagram 4. Two on-going UI progress bars, in the same colors as the note blocks covered in original status (corresponding button not pressed), responds the input from the user through performing a bow stroke. The direction of the filling for them should be the same as that of the stroke, which is clockwise for a down-bow in this case.

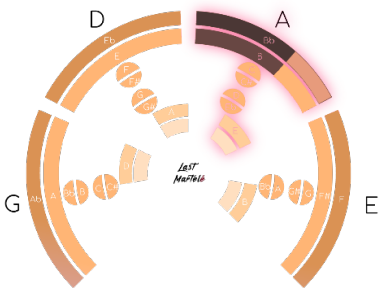


Figure 19. UI Paper Prototyping – Diagram 5. The mirror status of the counterclockwise filling UI progress bars compared with those represented in Figure 18, showing that the user is performing an up-bow stroke on the A string. A method needs to be come up with to avoid annoying fast-flipping aftermath caused by switching downbow and up bow frequently.

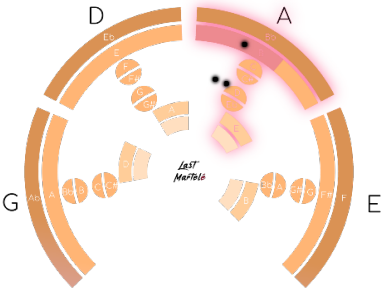


Figure 20. UI Paper Prototyping – Diagram 6. In real life, musicians perform their instruments either through reciting scores or looking at them while playing songs. In either case they can get information of a chunk of notes rather than only one. Three black dots in this diagram blinking at different speeds will tell the user what the next three notes being play-backed are.

The effect of utilizing this paper prototyping is rather limited. It will help users understand the MIDI visualization system but hardly can guide them play a euphonic melody. After all, music performance is such a game with strict time constraint, all about playing notes according to their pitches and lengths specially. There is no doubt that the iteration of the User Interface system driven by the underlying supporting MIDI system for this project will continue for a long time.

Section IV. Reflections and Next Step

Due to the complexity of the main two pillars of this project: violin bowing (fingering) analogy and music-UI synchronization, it cost much time to implement the system under the hood. Without the User Interface made out, several play-testers of *Last Martelé* did not know how to press buttons and triggers on Oculus controllers to create various notes, nor could they play a song by themselves. Though they all knew how to do the bowing gestures and were amazed about the mechanics.

Meanwhile, coming along with the development of *Last Martelé* is the underlying MIDI visualization system, which can be more meaningful in a sense. Music games and products developers can use it to create any musical instrument simulators as they want, as it was created within robust software engineering and encapsulates professional music concepts and practical knowledge. "The MIDI system can be extremely useful for children's music education; If combined with Machine Learning technologies it can also aid music compositions for those beginners."[20]

The next step is to implement and iterate on the extremely important system: User Interface. Once it gets done, we can test if the methodology used will work out or not.

[20] Cited as a commend from one of my friends who was graduated from School of Design, Jiangnan University in China as a master student in Interaction and Service Design.

Bibliography

Ableton. *Live MIDI Effect Reference*. 2019. Document. February 2019. <https://www.ableton.com/en/manual/live-midi-effect-reference/>.

Andrew Surmani, Karen Farnum Surmani, Morton Manus. "Circle of Fifths." Andrew Surmani, Karen Farnum Surmani, Morton Manus. *Alfred's Essentials of Music Theory: A Complete Self-Study Course for all Musicians*. Van Nuys: Alfred Publishing Co., Inc., 2004. 53. Document.

Association, MIDI. *GM 1 Sound Set*. n.d. Document. 2019 February. <https://www.midi.org/specifications/item/gm-level-1-sound-set>.

CCRMA. *WAVE PCM soundfile format*. n.d. Document. February 2019. <http://soundfile.sapp.org/doc/WaveFormat/>.

CSIE, NTU. *The MIDI File Format*. n.d. Document. February 2019. <https://www.csie.ntu.edu.tw/~r92092/ref/midi/>.

Dobroselsky, Maxim. *DryWetMIDI: High-Level Processing of MIDI Files*. 2 February 2019. Document. March 2019. <https://www.codeproject.com/Articles/1200014/DryWetMIDI-High-level-processing-of-MIDI-files>.

—. *DryWetMIDI: Working with MIDI Devices*. 31 January 2019. Document. March 2019. <https://www.codeproject.com/Articles/1275475/DryWetMIDI-Working-with-MIDI-Devices>.

Fullerton, Tracy. "Prototypes and Playtesting in the Industry." Fullerton, Tracy. *Game Design Workshop, A Playcentric Approach to Creating Innovative Games, Third Edition*. A K Peters/CRC Press, 2014. 19. Document.

Hohmann, C.H. *Practical Violin Method, Book 1*. Carl Fischer, LLC, 1910. Document.

Howcast. *How to Do Vibrato*. 1 August 2013. Video. September 2018.

Krug, Steve. *Don't Make Me Think: Revisited: A Common Sense Approach to Web Usability*. Peachpit, 2014. Document.

Liao, Sheng (Raymond). *Raymond's Last Martelé MIDI System Integration*. 1 April 2019. Video. 1 April 2019. <https://youtu.be/v6ZGymo16rQ>.

—. *Raymond's MFA Thesis Last Martelé - String Switching*. 1 April 2019. Video. 1 April 2019. <https://youtu.be/kbh5sRbuxQ0>.

—. *Raymond's MFA Thesis Last Martelé Bowing and Fingering*. 1 April 2019. Video. 1 April 2019. <https://youtu.be/bc-f9h8BGKI>.

Mellor, David. *The difference between channel aftertouch and polyphonic aftertouch*. 25 December 2005. Document. February 2019. <https://www.audiomasterclass.com/newsletter/the-difference-between-channel-aftertouch-and-polyphonic-aftertouch>.

Nicora, Claudio. *VirtualMIDISynth*. 2017. Document. March 2019. <https://coolsoft.altervista.org/en/virtualmidisynth>.

Phillips, Winifred. "Why Would You Want to Write Music for Games - Music Theory." Phillips, Winifred. *A Composer's Guide To Game Music*. 2014. 29. Document.

Quick, Donya. *Working with MIDI on Windows (Outside of a DAW)*. 14 September 2014. Document. March 2019. <http://donyaquick.com/midi-on-windows/#x1-70002.2.1>.

Sanford, Leslie. *C# MIDI Toolkit*. 18 April 2017. Document. February 2019. <https://www.codeproject.com/Articles/6228/C-MIDI-Toolkit>.

Vandenneucker, Dominique. *MIDI Tutorial*. 2012. Document. January 2019. <http://www.music-software-development.com/midi-tutorial.html>.

—. *Music Data Structures*. 2012. Document. February 2019. <http://www.music-software-development.com/music-data-structures.html>.

Vergara, Jhanraphael. *PANTENE : A Very Touching Deaf Violinist Commercial*. 26 January 2010. Commercial. Feburary 2018. <https://youtu.be/Ect56804xfA>.