Tiny App Of Technique

Developing a Digital Tool For Violin-Based Music Education

By
Liam Wolozny
Ethan Chang

9/9/20

Submitted to:
Dr. VJ Manzo

An Interactive Qualifying Project submitted to the Faculty of Worcester Polytechnic Institute in partial fulfillment of the requirements for the degree of Bachelor of Science

Abstract

Modern technology allows for the diversification and expansion of music technologies. What was once a purely auditory art form is now one that can be translated into many different sensory languages. This project seeks not to correct the traditional model of learning an instrument, but to supplement it with a new, easily accessible tool that takes advantage of the most important human sense for the digital age, our sense of sight [1]. Its end goal is to create an easily reusable tool that provides clear results in order to consistently train its user's ability to produce highly faithful musical renditions. A tool like this would allow students to practice independently with a reliable and accessible source of feedback and would circumvent overreliance on teacher input and thus allow for better learning of music fundamentals and playing fidelity. This IQP report outlines preliminary research and prototype work for a tool commissioned by IQP Sponsor and Violin Instructor Wendy Case. It considers current development challenges, previous attempts at similar tools, and design challenges moving forward.

[1] "Sense of sight is the most important for humans, right? Your ...." 11 Nov. 2018, https://www.technology.org/2018/11/11/sense-of-sight-is-the-most-important-for-humans-right-your-hierarchy-of-senses-is-not-objective/. Accessed 3 Sep. 2020.

Acknowledgments

We would like to thank Dr. V.J. Manzo for his advice and support throughout the project. His knowledge of Max and the lessons from his book he shared with us made our program possible.

We would also like to thank Wendy Case, our project sponsor, for her comments and feedback at each stage in development. She provided us with excellent ideas and features for the project as well as for future development.

Contents

Abstract I

Acknowledgments II

Contents III

1. Introduction 1
    1.1. Inspiration
2. Background and Viability 1
    2.1. 3D Particle System 1
    2.2. VisualMelody 3
    2.3. ImproViz 5
    2.4. Visualization For Education 6
        2.4.I. 3D Particle System 7
        2.4.II. VisualMelody 8
            2.4.II.1. Preliminary Survey 8
            2.4.II.2. Training Phase 8
            2.4.II.3. Testing Phase 9
            2.4.II.4. Summary Survey 10
            2.4.II.5. Key Takeaways 10
        2.4.III. ImproViz 10
            2.4.III.1. Composites: Melodic Landscapes and Harmonic Palettes 11
            2.4.III.2. Personal Use Tool 11
3. Methodology 12
    3.1. Paper Prototype 12
    3.2. Preliminary Interviews 13
    3.3. Prototype 13
        3.3.I. Software 13
        3.3.II. Modules 14
            3.3.II.1. Melody Contour 14
            3.3.II.2. Audio Management 16
            3.3.II.3. Waveform With Moving Playhead 17
            3.3.II.4. Pitch To Color 18
4. Concluding Thoughts 19
    4.1. Future Work/Implementation 19
    4.2. Conclusion 19

1 Introduction

1.1 Tiny App of Technique

The Tiny App of Technique project was commissioned as an expansion and translation of Wendy Case's "Tiny Book of Violin Technique," a guidebook intended to help Violin players understand and learn how to play their instrument through the lens of physics. Named by its developers so as an homage to their sponsor's work, research and development of the Tiny app was accomplished with two functions in mind: it was to primarily serve as an effective and easy to use tool capable of aiding in the compositional process, and secondarily as an additional point of focus for audience engagement during musical performances.

The Music Visualizer that we are envisioning will attempt to make visual the loudness of a sound, its timbre, its duration, and its pitch. This visualizer would be different from others previously made by its focus on leveraging human-pattern recognition. Simply put, we plan on making the visualized portions easily understandable and repetitive so that it may be used for consistent violin practice. This idea, when compared to others available to the public, would endeavor to avoid one of the major problems visualizers seem to encounter: lack of simplicity. The end goal for this project would be to have a simple and easily usable visualizer that can be tested and shown to actually improve violin playing.

2 Background and Viability

In order to create a tool that could fulfill the goals stated above, the Tiny App IQP team researched into the viability of music visualization as a teaching aid as well as into several existing music visualizers to study how they differ, how they err, and how they could be improved.

2.1 3D Particle System[2]

This as of yet unnamed music visualization tool, which we will refer to as the 3D Particle System, takes advantage of classical music's tendency to have a well-defined, time-varying structure. By focusing on specific structural musical aspects such as Tonal Pitch (matched to particle color), Volume (matched to varying particle size), and Timbre (matched to an additional set of particle emitters) and using MIDI files (files with pre-established musical parameters that make data processing less expensive), the 3D Particle System Group created an effective music visualization tool that showcases timbre to a satisfactory degree. Each note was associated with a particular color based on Louis Bertrand Castel's model for color mapping. This model establishes different colors for each of twelve musical notes, differing from those of other researchers such as Isaac Newton by also considering accidentals (semitones) and giving them appropriate color designations.

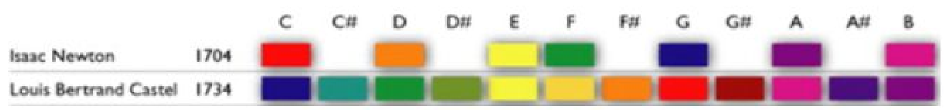


Fig. 1 Image of Louis Bertrand Castell's Model for Color Mapping (of note are the semitones, or those notes above with the hashtag; sharp notes).

Below lies a snapshot of the 3D Particle System in action. There are several notable characteristics visible:

- On the leftmost box, there are two emitters in the middle with equivalent colors but in reverse positions indicating reverse melodic movements played by both Violin II and Viola.
- The middle box showcases several emitters with higher variations in colors emitted indicating a higher number of notes that were played more quickly in comparison to the other diagrams.
- On the rightmost diagram, the largest particle sizes are visible, indicating notes that were played at a higher volume than the other two sets.

[2] "Creating and Evaluating a Particle System for Music Visualization". Dec. 2013, https://www-sciencedirect-com.ezpxy-web-p-u01.wpi.edu/science/article/pii/S1045926X13000566. Accessed 3 Sep. 2020.

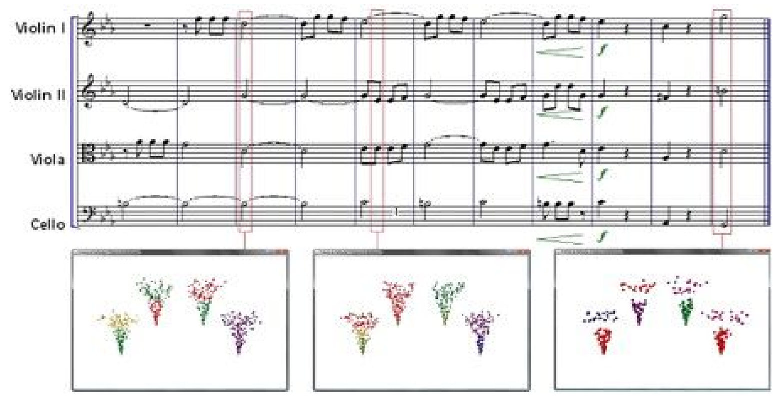


Fig. 2 An image showcasing the 3D Particle System's Emitter visualization functionality. Emitters are each tied to their respective red rectangles and visualize only the area enclosed therein.

The 3D Particle System, or a variation on the technology, is something we believe would make an ideal aid to live performances. Though perhaps not the best system for actual musical training given its flashy and quickly changing nature, it would be interesting to provide audiences something like a key, that they might understand what the changes in the particles mean during a performance. More work could be done in this regard in order to make the particle fountains more visually attractive; the team thought this system was worth mentioning regardless.

2.2 VisualMelody[3]

VisualMelody was designed to aid in the resolution of real, documented problems that occur to students in music conservatories. By studying the learning process in three separate conservatories in three distinct Italian cities, they settled their foci on two notable situations:
More time taken to complete the compositional process as a result of checking and rechecking for possible errors, as well as the considerable time spent on recovery from mistakes. VisualMelody seeks to overcome and streamline these situations by simplifying the compositional process to ameliorate the time consumed in understanding the structure of a composition. This in turn allows for a quicker transition into the error-checking stage, and as such allows for a great reduction in time taken.

'Four-Part Music,' or 'Four-Part Harmony,' is a term used to describe music written for four voices/instruments. VisualMelody was devised as a tool to define the notes of each of these

[3] "Understanding the Structure of Musical Compositions: Is Visualization An Effective Approach" 30 June, 2016, https://journals-sagepub-com.ezpxy-web-p-u01.wpi.edu/doi/full/10.1177/1473871616655468. Accessed 3 Sep. 2020.

melodic voices and simultaneously analyze the relationships between them in order to shorten the time required to complete compositional processes. VisualMelody distinctly differentiates between visualizations by showing a colored trajectory for each of the four represented voices. Musical notation, set below the line visualization, has colors matching to each of their respective lines. Melodic errors are represented by two intersecting lines highlighted in red as seen below. (Both Fig. 3 and Fig. 4 are Bach's BWV 606)

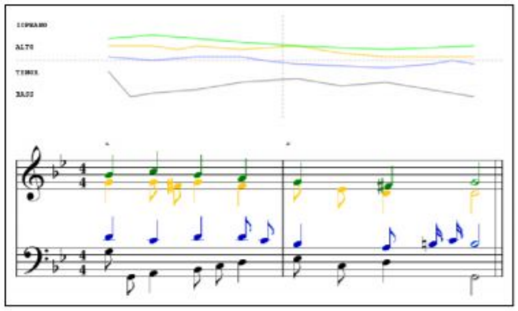


Fig. 3 VisualMelody's visualization of melodic trajectories

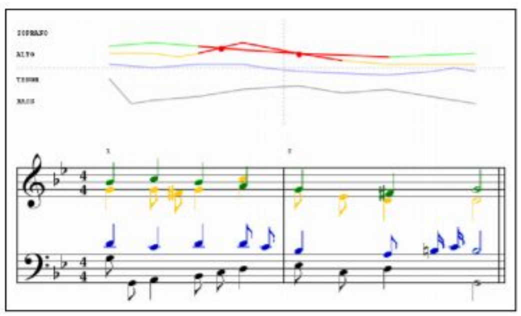


Fig. 4 VisualMelody's 'intersection error' highlighting system

VisualMelody also boasts interval (the 'distance' between notes) representation. These are shown as an ellipse with a length approximating the interval "length" and colored to match type (green for unisons, blue for fifths, yellow for octaves). Intervals violating composition rules have ellipses outlined in red. Notably, an overlap of ellipses can occur whenever there is an intersection between lines. (See Fig. 5 for an example of overlaps and Fig. 6 for an example of both overlaps and composition rule violations)

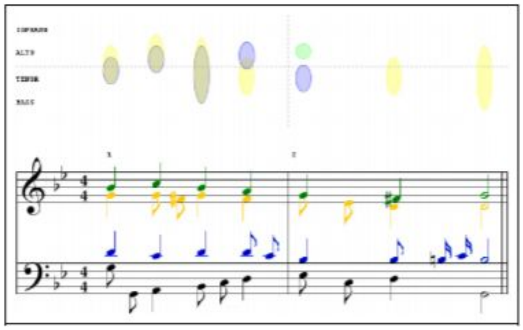


Fig. 5 Visualization of intervals in Bach's BWV 606's last two measures

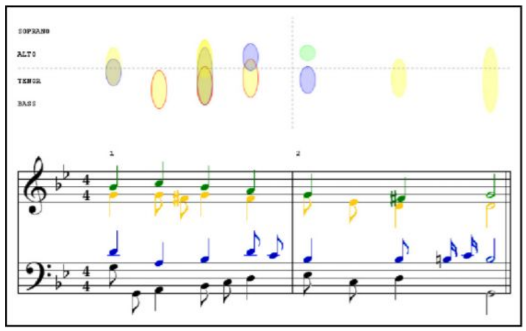


Fig. 6 Visualization of the unison, fifth, and octave intervals in Bach's BWV 606's last two measures, modified with parallel octaves

VisualMelody's simplistic and elegant design, as well as its error detection system are objects of inspiration for our own visualizer. The visualizer's functionality for music editing, melodic visualization, music playing, and saving and loading of both the score and visuals leaves very little to be desired, making it a very strong competitor in the market.

2.3 ImproViz[4][5]

ImproViz is an example of a more specialized kind of visualizer meant to specifically analyze the patterns of jazz musicians when they improvise. It is made up of two parts: a melodic landscape representing the contour of melody the musician played and the harmonic palette showing a distribution of the notes played in each measure. The melodic landscape shows the rise and fall of pitch played, but it does not display specific notes. Multiple melodic landscapes from different choruses of a performance are overlaid to show patterns in the solo. The harmonic palette's distribution covers the chord played during each measure in the

[4] "A Report on Musical Structure Visualization" August, 2007, http://www.cse.ust.hk/~wallacem/winchan/research/music-report-winnie.pdf. Accessed 3 Sep. 2020.

[5] "ImproViz: Visual Explorations of Jazz Improvizations" April, 2005, https://www.ischool.berkeley.edu/research/publications/2005/improviz-visual-explorations-jazz-improvizations. Accessed 3 Sep. 2020.

performance, showing which fundamental notes, such as the first, third, fifth, and seventh intervals, were or were not played. This palette reveals the musician's tendency to use or not use certain groups of notes throughout the performance. With melodic landscapes and harmonic palettes of multiple soloists, ImproViz can compare the different styles of soloists on the same song. The melody contour system in ImproViz directly influenced our decision to use a similar system. The different performances lined up vertically or overlaid on top of each other allows for direct comparison.

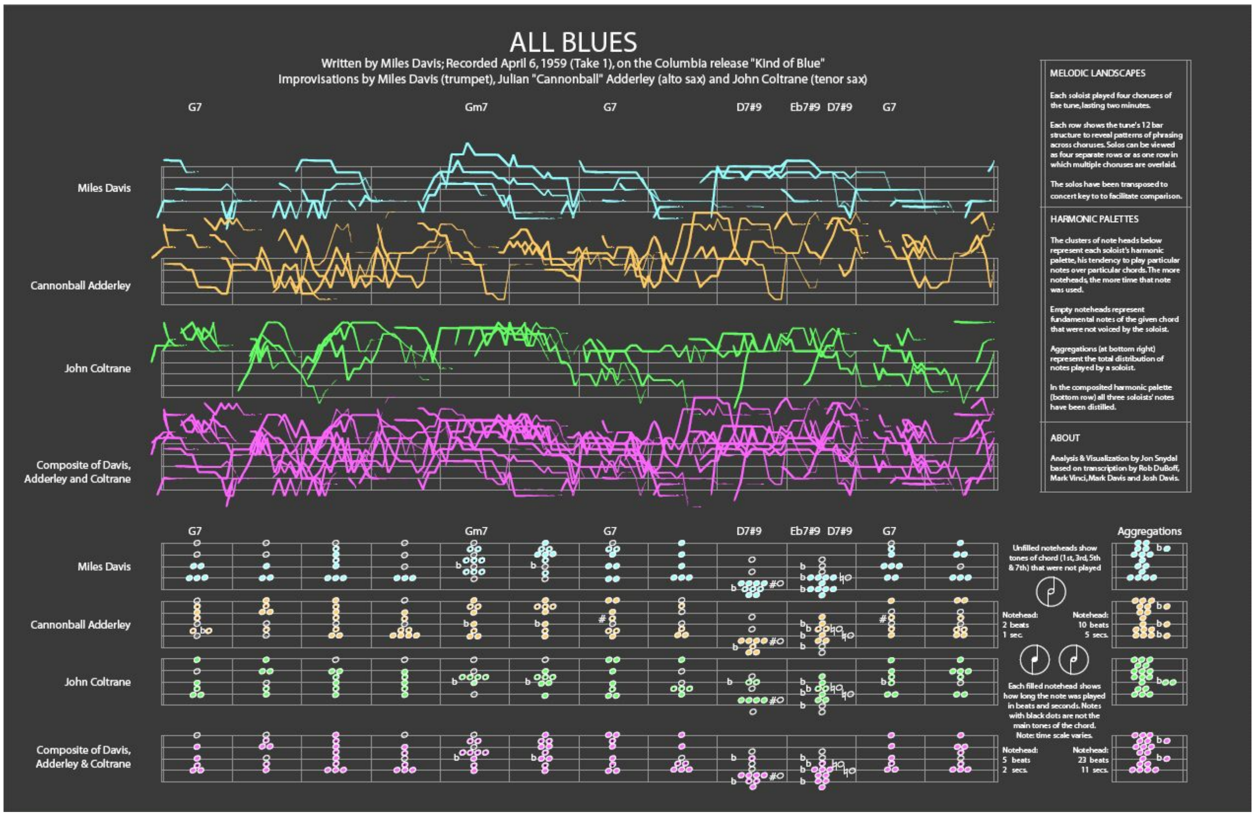


Fig. 7 ImproViz melodic landscapes and harmonic palettes for three musician's performances of All Blues, as well as a composite melodic landscape and harmonic palette

2.4 Visualization For Education

In order to understand the educational potential of such an idea as the 'Tiny App,' we resorted to studying not only the test results produced by the different visualizers we previously discussed, but also the data and testimonies produced by those of other more generalized studies.

I. 3D Particle System[2]

The 3D Particle System team engaged in a multi-group study in order to evaluate the usefulness of their program. This trial used the finalized emitter visualization system made by synchronizing a computer animation with the ticks of a MIDI event- each point during which an action can be performed in a given MIDI file. Through this, each musical note's timestamps in the provided file are synchronized with each frame of the emitter animation allowing for an accurately represented visualization.

The groups that formed part of these trials were Group 1 (Seven expert musicians from the University Camerata with 10-20 years musical experience), Group 2 (Thirteen undergraduate music students), and Group 3 (Thirteen generic, non-music undergraduates). The trial, though not extensive in terms of quantity of people, provided answers to interesting questions, some notable examples of which can be found below:

Group 1		Group 2		Group 3	
Yes	No	Yes	No	Yes	No
7	0	11	2	10	3

Fig. 8 Table for Question 3: "Can you identify frames in which all the emitters have the same colors, indicating that the instruments played a note in unison or octave?"

	Group 1	Group 2	Group 3
Yes, by rhythmical patterns only	2	1	4
Yes, by melodic patterns only	0	0	0
Yes, by both patterns	4	12	4
Neither of them	1	0	5

Fig. 9 Table for Question 5: "Can you identify the motif (musical unit) in the animation? By rhythmical patterns? By melodic patterns? Both?"

	Group 1	Group 2	Group 3
Yes	6	13	13
No	1	0	0

Fig. 10 Table for Question 9: “Do you think that this type of visual experience can become an additional and interesting support for music learning? To conclude, please feel free to include in the user survey form any comments and suggestions you would like to make.”

The data gathered from this general user study seems to support the idea of a music visualizer as a musical teaching aid and musical recital enhancer. The volunteers “demonstrated a strong perceptual relation between music terms (tonal pitch, volume, and timbre) and visual attributes (colors, size, and emitter). The users also expressed positive opinions about the animation’s ability to “express the mood of the music (happy, sad, lyric, aggressive, etc…),” functionally acting as form of abstract musical teaching. Further, thirty-two of the examinees agreed “this type of visual experience can become an additional and interesting support, particularly for music theory learning.” This first study’s results were promising and encourages the Tiny App team’s belief in the usefulness of a tool such as this.

II. VisualMelody[3]

Forty users were tested for VisualMelody’s study. Its aims were to: analyze whether visualization can “facilitate the study of the rules used in classical music,” simplify the compositional process, collect user opinions about the usefulness of such an application, as well as gather further data on its long-term potential.

The VisualMelody team envisioned and carried out a study split into four distinct phases:

1. Preliminary Survey

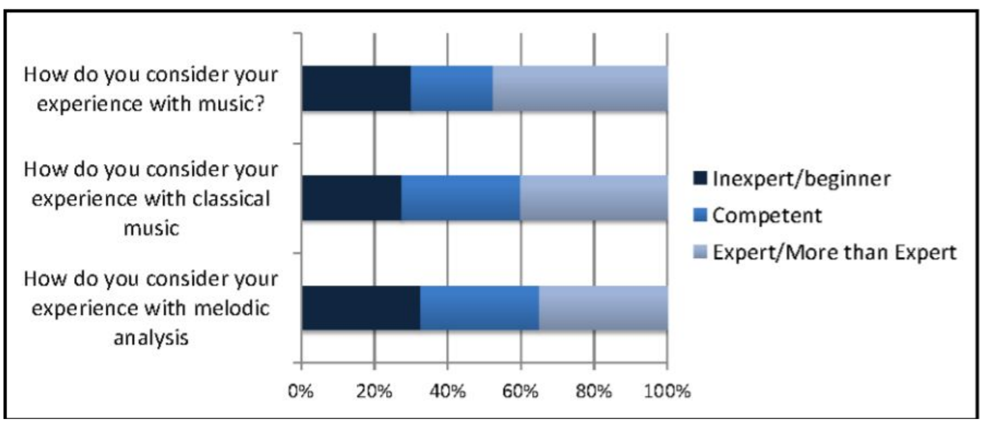
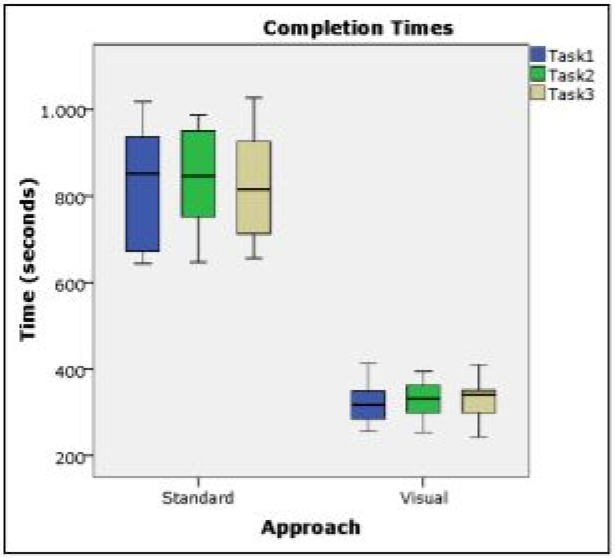
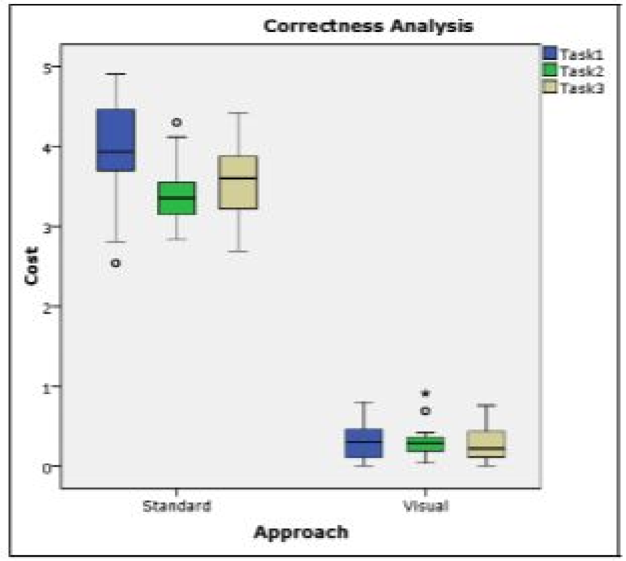


Fig. 11 Image of study’s Phase I Preliminary Survey

2. Training Phase

During the training phase, users were given both a 5-min period during which they would become familiar with the tool as well as some basic information on how to operate it.





3. Testing Phase

Users were then asked to compose new *tenor*, *alto*, and *soprano* lines for a pre-established bass line. The participants were divided into two equal groups, those who would use VisualMelody's graphical aids and representations (Visual), and those who would not (Standard). VisualMelody collected performance data for each individual such as performed task, completion time, as well as the number of inserted or deleted notes. Below are the results of these tests:

	Approach	Tasks			U, p-value
		Task 1	Task 2	Task 3	
Time	Visual	318	330	329	−5.41, p < .001
	Standard	818	840	831	
Cost	Visual	0.32	0.31	0.27	−5.41, p < .001
	Standard	3.96	3.39	3.56	

Figure 12 Left table shows a comparison between the amount of time required to finish each task by using the Standard (left) and Visual (right) approaches. Right table shows a comparison for the "melodic cost," or the errors committed, between the Standard and Visual approaches. The bottom table shows a numerized collection of data from which the first two graphs were derived.

Pictured above, the use of VisualMelody's graphical features made more efficient the composition of chorales as well as reduced the amount of errors and mistakes by a considerable margin. To specify, users encountered an almost 2/3s reduction in time taken to complete their tasks as well as a 92% reduction in errors made.

4. Summary Survey

Question	Visual	Standard	U, p-value
It was simple to use VisualMelody	4.4	3.9	126, **.046**
It was simple to add notes	4.2	3.8	177, .552
It was simple to delete notes	4.1	3.6	148, .163
It was simple to modify the height of a note	4.6	3.7	78, **.001**
It was simple to modify the duration of a note	3.6	3.2	126, **.046**
Do you will continue to use the system in the future?	4.8	4.2	131, .063

Fig. 13 Summary Survey, Questions submitted to both Visual and Standard groups

	Question	Mean (standard deviation)
Q_S	Could you find helpful a tool that shows a graphical representation of the melodic lines during a music composition process?	4.7 (0.7)
Q_V	Do you think it was useful the idea of graphically show melodic lines during a music composition process?	4.9 (0.4)

Fig. 14 Summary Survey, Q_S submitted to the Standard Group, Q_V submitted to the Visual Group

The final part of the trial encountered uplifting results with an average score of 86% on the desirability and usability survey (Fig. 13). A follow-up targeted survey (Fig. 14) showed positive results when presented with questions that inquired about the helpfulness/usefulness of a musical compositional aid in the form of a visualizer.

5. Key Takeaways

A tool such as this benefits immensely in public opinion from its accessibility. The app and its separate menus must be both engaging and informative without being overwhelming. The user interface must allow for enough functionality that the tool remains useful, but not be so cluttered as to be detrimental to the user experience. The Tiny App seeks to draw upon and be influenced by these wealths of preexisting knowledge such as VisualMelody's segmented trials in order to avoid common hurdles and entrapments that come with developing visualizing software.

III. ImproViz[5]

The Improviz system seems to be a tool that was not used in any form of trials at all, however information can be drawn from its source file about its potential as an educational tool. As described by its creators, ImproViz is a tool intended to increase students' "understanding of jazz theory through visual solo analysis," as well as improve improvisation skills. ImproViz' contribution to music education was defined by its creators in the following two ways:

1. Composites: Melodic Landscapes and Harmonic Palettes

Following the previously discussed idea of melodic landscapes which show the general 'idea' behind a soloist's playing of a piece of music, Composite Melodic Landscapes show just that: a composite of Melodic Landscapes taken from several different artists and displayed at once in order to showcase and highlight the differences and similarities between the musicians' improvisations.

Composite Harmonic Palettes on the other hand shows similar images differentiated by their focus on the distribution and range of notes used in an improvisation as well as in the time/spaces that each artist would uniquely add.

2. Personal Use Tool

ImproViz's creators sought to have this tool be of personal use in multiple ways. They first sought to create a high level educational approach to teaching jazz playing improvisation from multiple different sources (soloists) simultaneously. They also intended ImproViz as a tool that a soloist could use to translate their own improvisations into visualizations in real time.

ImproViz' competitive edge comes from its ability to analyze improvisations in real time. This ability to read functionally any random input data from a MIDI enabled instrument and be able to visualize it in real time while simultaneously showing note distribution shows real potential as a teaching aid. Having gathered data from these three successful, potential-rich visualizers, we set off to create our own tool for our sponsor Wendy Case: The Tiny App of Technique.

3 Methodology

3.1 Paper Prototype

As part of the development process, before finalizing our functional prototype in Max, we created a mockup to summarize our plans for the App. This mockup would later be presented to Wendy, our sponsor, and through her feedback we would ensure that our work would be aligned with her goals. Using our research as summarized above, we created a mockup composed of several elements inspired by our research into visualizers that could be mixed together on screen: the playback and recording controls, melody contour, particle fountain, waveform, visuals selector, and lesson text. After our initial interview with Wendy as well as after being influenced by our research, we settled upon a visual selector in order to make modifiable the controls and allow for selective simplification of the menus. Our design was predominantly focused towards modularity and flexibility, so that any one or combination of visuals could be on screen at any time without affecting or relying on the others.

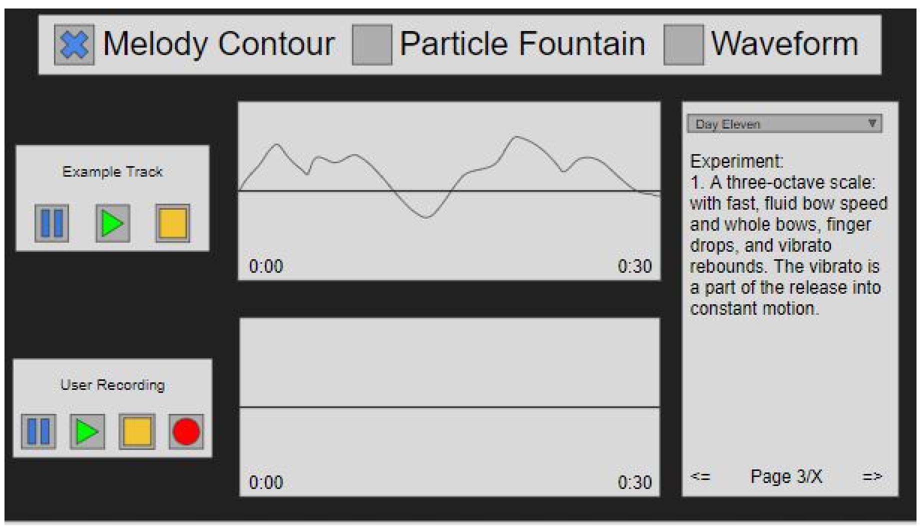


Fig. 15 Paper Prototype, Potential Function Prototype Layout 1. Notable features include the visual selector (top), Wendy's Lessons with dropdown menu (*From Tiny Book of Violin Technique*, Wendy Case), Playback and Recording Controls (left), and melody contour (center)

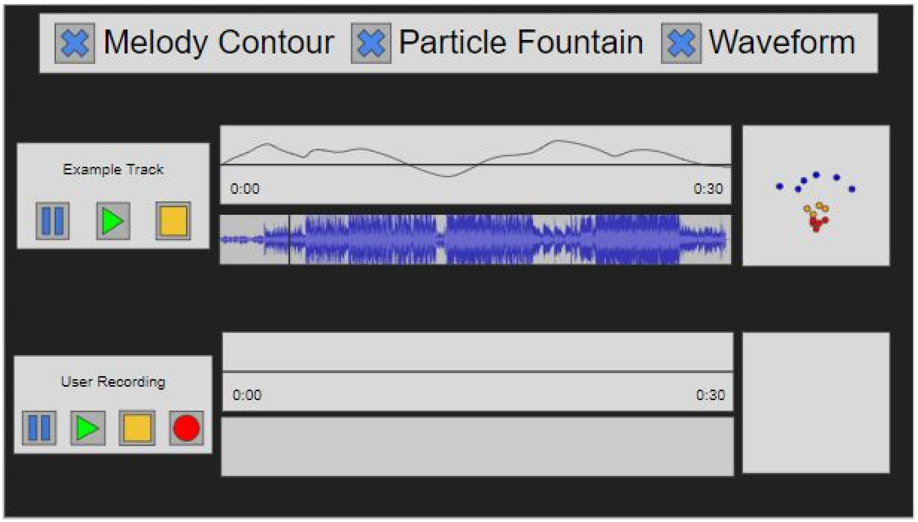


Fig. 16 Paper Prototype, Potential Functioning Prototype Layout 2. Differences include the double particle fountain on the right, as well as the addition of the sound waveform.

3.2 Preliminary Interview

The goal behind this project was to create a tool for Violinist Professor and Synesthete Wendy Case. As a result, she was consulted several times in order to understand what her original vision was, and what were the features that she desired in her tool. Originally the Tiny App was decided to be a tool that would accomplish the overarching goal of improving music education and performances. This tool would be used to bridge the knowledge gap between a student and a teacher allowing them to, in Wendy's words, "really hear the way they [actually] sound." Wendy envisioned this as a visual, appealing, and informative way of showing musical articulation. She seemed very receptive to the idea of particle fountains (discussed in section 2.1, 3D Particle System) as one of the main recital aids. As a synesthete, she also expressed interest in a pitch to color matching system that would be used for associating colors to specific notes, as well as for simulating the color associations she makes as a result of her condition.

3.3 Prototype

I. Software

The functional prototype for our tool was developed using the Max/MSP software developed by Cycling74, a visual programming language. Max is a language that uses objects (software in charge of actions), patchcords (software in charge of connecting one object to another), and controls (software in charge of connections between UI objects to provide control values or display results). Our app was developed modularly as separate patches that were later merged into a single patch in order to simplify coding milestones, compartmentalize bugs, and allow for modifiable parts in order to facilitate future work. The main milestones accomplished during the completion of this project were: melody contour, audio recording/playback/saving, waveform with moving playhead, and pitch to color matching.

II. Modules

This subsection deals with the modules that were developed as discussed directly before this. It includes work done upon them and the main challenges faced.

1. Melody Contour

The melody contour was the most difficult and time-consuming part to develop as a result of the team's unfamiliarity with the Max MSP platform's audio visualization features. As a result, this module required several iterations before the team achieved a satisfactory result.

Milestone/Goal	**Expansion**
How to detect pitch/fundamental frequency from audio using fzero~	Issue related to unfamiliarity, it was necessary first to understand how the fzero~ object works before it was employed
Sizing the contour so it is centered around the recorded frequencies	The program changes the maximum frequency displayed (the height of the Y-axis) in real time to fit the contour in the center of the graph. Initially at a low maximum frequency (50), the program keeps track of the median of all frequencies recorded and multiplies it by 1.5, resulting in the new maximum frequency. The reason this median is multiplied by 1.5 is to provide equal amounts of space above and below the melody contour, since if the median itself were used, the highest frequency recorded would be at the very top of the graph, making the graph difficult to see and vertically lopsided.
Recording and storing pitch data for a specific amount of time	When loading the example audio, the program transmits the length to the user recording melody contour module. This sets the amount of time the program records and how many samples to use in that time. When recording pitch data, the program samples once per millisecond for as long as the example audio, timed by a delay object. For example, if the example audio were 1 second, then recording would last 1 second and produce a melody contour of 1000 samples.

Dealing with the program slowing down when multiple modules were in the same patch, causing the sampling for the contour to be inaccurate	Visual elements such as the waveform are expensive for Max to compute, which in turn puts the audio recording out of sync with the melody contour because the program was too busy to consistently sample for contour data. This was solved using Overdrive mode in Max, allowing the intensive visuals to operate in the main thread while the timing objects used for the melody contour operate in a separate thread.
Saving and loading melody contour data	Since the melody contour is drawn while the audio plays or records, we needed a way to load in the melody contour of the example audio without needing to replay the example. The example contour is loaded from a file previously saved. The data is two parts: the frequencies sampled every 1 ms for the length of the recording, and the range of frequencies for displaying it in the middle of the graph. When the user saves the audio they record, the contour is automatically saved.

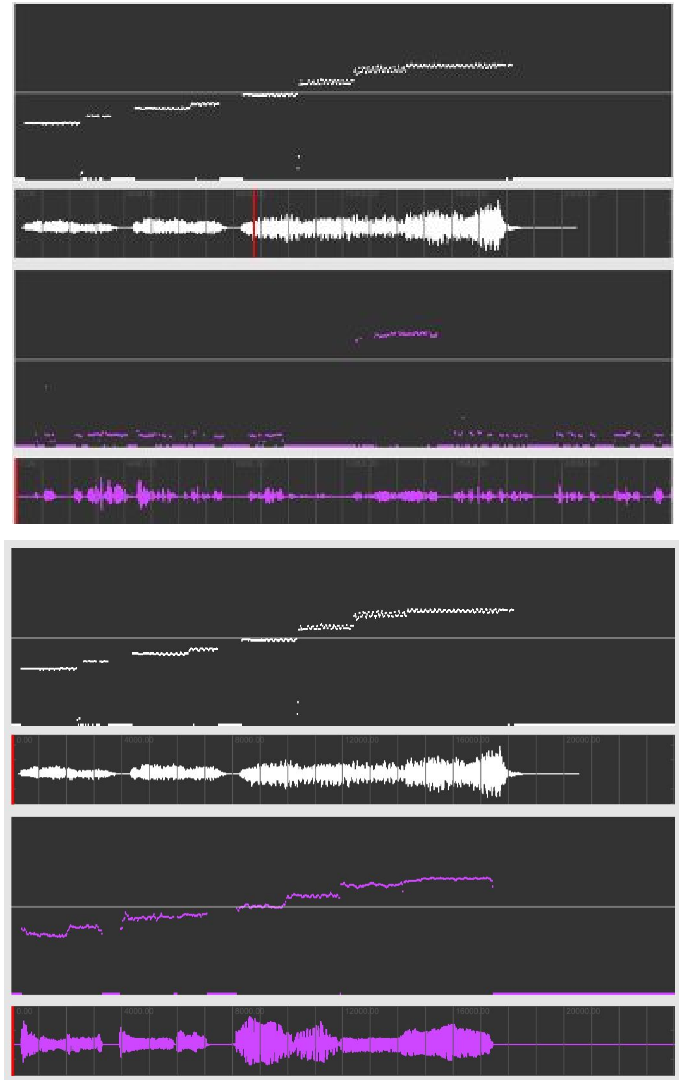


Fig. 17 Images of Tiny App of Technique. Image shows melody contours, white for sample files and purple for recordings

2. Audio Management

Audio management was somewhat complicated by the amount of subfeatures that were involved, all connected by a shared audio buffer object “buffer~.”

Milestone/Goal	**Expansion**
Recording	The user can record for the same amount of time as the example recording.
Playback	The user can play, pause, and restart the playback of both the example and their own

	recording.
Saving/Loading Audio Files	Loading example files is currently hardcoded to use a specific example audio file and melody contour file, but this can be easily changed to load different files by selection. When the user saves, they can choose the name and location of the audio file, and the melody contour data is automatically saved as a file “user_contour.”

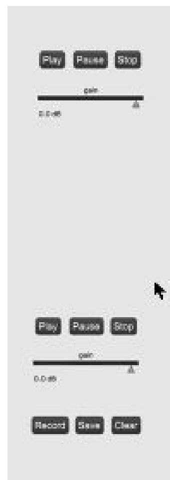


Fig. 18 Image of Tiny App of Technique’s Audio Controls

3. Waveform With Moving Playhead

Waveform with moving playhead was relatively simple as we were able to use knowledge and objects from a previous Max project and apply it into this one.

<u>Milestone/Goal</u>	**<u>Expansion</u>**
Waveform with Moving Playhead	The playhead is a slider controlled by the audio playback object, “groove~.” The groove~ object outputs the current location of playback as a decimal number following the range 0.0-1.0, where 0.0 is the beginning of the song and 1.0 is the end.

Fig. 19 Image of Tiny App of Technique in use. Image shows waveform with moving playhead

4. Pitch to Color

Milestone/Goal	**Expansion**
How to determine pitch/frequency?	Using the same previously mentioned “fzero~” object to determine pitch/frequency from audio as well as the “ftom” object to convert from frequency float to MIDI
Discover a way to map a note to color, as well as the setting of notes on a rising color gradient (lower notes = darker colors, and higher notes = lighter colors)	There was some initial difficulty with color mapping but ultimately we settled on a switch function. This way, the twelve possible notes would control the twelve possible, preset colors. The color gradient was developed by dividing the MIDI notes by twelve in order to find the octave and subsequently scaling the ten possible octaves to a multiplier (0.4-1.0). As the colors are RGB, the multiplier can be applied to the red, green, and blue values, where a lower multiplier makes the values lower and the color darker and vice versa. Colors used are currently randomly assigned.

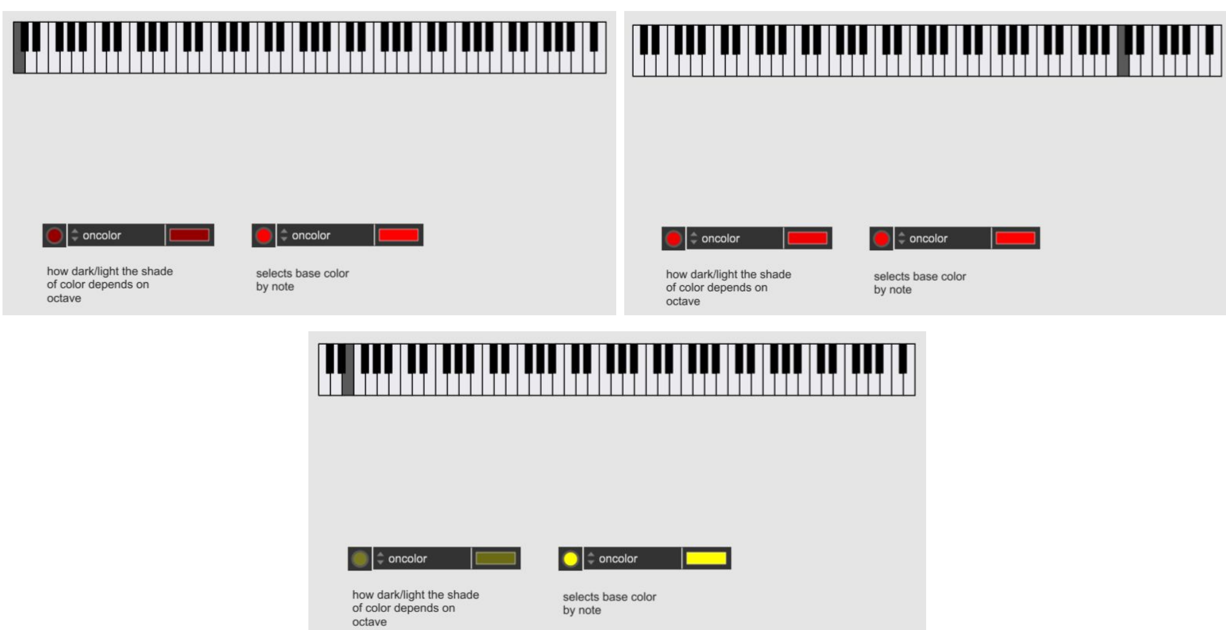


Fig. 20 Images of color mapping of twelve colors to twelve different notes. The color lightens or darkens depending on which octave is played (color lightening is matched to rising pitch and color darkening is matched to falling pitch)

4 Concluding Thoughts

4.1 Future Work/Implementation

Though we have a functioning prototype, there is much that can still be improved and added. Development on the Particle System is something that should be of primary focus to any future teams that work on this project. While we were able to figure out basic functionality for color mapping notes to colors, the challenge of creating an accurate particle fountain. An example as to how this was done can be seen in the 3D Particle System's paper

An improvement requested by Wendy is to have the lessons that can be found on her webpage embedded within the app itself. While this was received late in the development process, it is possible to add the files to Max as an embed within the patcher. A possible implementation of this work was drafted in the paper prototype in which a dropdown menu for lesson selection was prototyped. A possible downside is an increase in visual clutter to the App, however adding it as another tab might serve as an easy fix.

Wendy also requested additional forms of functionality. A feature that could overlap the melody contours of both the sample file and trainee in order to give the user an easier way to compare their progress with their goal. Additional functionality could be added by making the Tiny App capable of receiving user-inputted sample files. Currently the sample file in the Tiny App is hard coded into the patch, but changing the App so that it might accept user inputted samples would increase its user interactivity.

Beautification improvements could be made in the form of choosing non-random colors to map to the twelve different notes such as the Louis Bertrand Castell Model mentioned in section 2.1. Allowing for dynamic selection for which visualizer is on screen, an idea that could reduce visual clutter. Three specifically requested improvements by Wendy were to have the mapped colors appear on the piano keys themselves (piano is the one seen in Fig. 20), have the playhead move along the melody contour as well as the waveform, and simple beautification of the UI in order to make it more pleasing to the eye.

4.2 Conclusion

The Tiny App team believes that we were successful in developing a functional prototype for visualizing musical performances for violin students. Further testing still needs to be conducted in order to fully understand the benefits of visualization for music education. Regardless, we believe that this paper functions as a serviceable collection of information showcasing its benefits. Moreover, our prototype functions as a proof of concept and we are confident that we will receive positive results if/when subjected to general trials.We would like to point towards the modularity of our tool's design and its general assistive and supplementary nature as its focal points. Though predominantly intended for students, the Tiny App's modular design means that any music student capable of using Max is able to change the Tiny App's features and main functionality to tailor the Tiny App to their own specific needs.

The project had its limitations. The time we had to develop our tool was far smaller than a project of this scope deserves. As a result, some of our originally planned features are either incomplete or not present. The team was also composed of only two people which further limited the amount of work that could be completed. Truly though, one of the most important goals we had, student trials and testing, could not be completed which essentially means that we have no real data of how the Tiny App would fare in the 'outside world' so to speak. In the end though, despite what limitations we did have, the Tiny App team is satisfied with what we have accomplished. While much work remains to be done, the team believes that the project resulted in a good base and prototype from which a fully realized tool might arise.