AudioSavant: Exploring Musical Structure Through Visual Interaction

Mark Daly John Alexis Guerra Gómez John Locke Megan Monroe

ABSTRACT

In music education, inexperienced students have great difficulty making a connection between the sounds they are hearing and the actions taken to produce those sounds. This is the primary hurdle that must be overcome before these students can move on to more advanced concepts such as music notation, creation, and composition. To this end, we introduce AudioSavant, a visualization tool that allows users to explore some of the hidden, complicated, or otherwise difficult to explain characteristics of a musical piece. Created in conjunction with Carnegie Hall's "Music for Everyone" initiative, the tool provides valuable information about musical organization and structure without requiring prior musical knowledge. An initial user study revealed the AudioSavant interaction model to be both intuitive and engaging to inexperienced music students across a wide range of ages.

1. INTRODUCTION

Research has shown on numerous occasions that quality music education can provide benefits far beyond the enjoyment of music itself [16, 20]. Students who learn to read and play music have shown improvements in other areas of their schooling such as math, science, and reading. Before students can play music, however, they must understand the sounds that their instrument produces and how those sounds relate to the overall composition. To many students, this serves as a significant challenge [19]. Music education is particularly difficult to teach in the classroom, and requires a more individual and hands-on approach to learning.

Music is rarely presented as a single instrument performance. Even the most elementary pop songs involve at least three instruments working together to produce the overall effect. To the untrained ear, listening to such a composition often presents as a knot of sound. Inexperienced students make no connection between what they are hearing and what they may eventually play as a musician. They cannot aurally distinguish between the instruments, or identify sections and patterns within the piece.

To assist such students (specifically high school students with little to no music experience), Carnegie Hall's Weill Music Institute developed the idea of a music map. Shown in Figure 1, this paper-based map charts the course of each instrument through a single composition. Students can listen to the composition and follow along with their music map, identifying key elements as they arise. To extend this idea, The Weill Music Institute intends to use the information in these music maps to inform the design of some sort of animation or lighting display that can play concurrently with the music, enhancing the music map experience for students. They are currently working with members of The University of Maryland's HCI Lab to explore these options.

In this paper, we present AudioSavant, a tool that specifically targets the listening skills of inexperienced music students. In the context of the music map project, AudioSavant has three primary goals.

1. Reduce the work required of expert music instructors in creating these music maps.

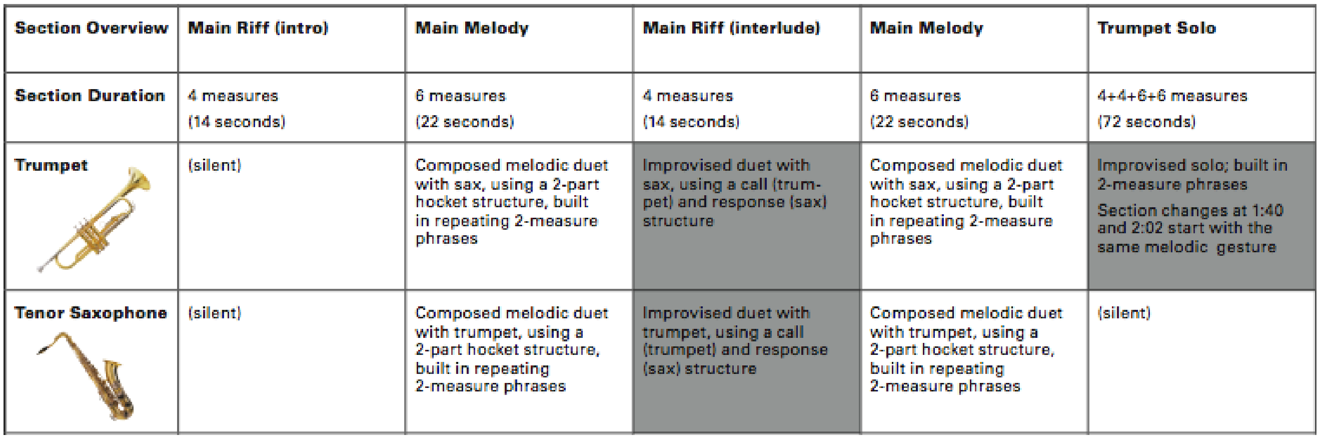
2. Recreate the paper-based music map in a more engaging, interactive form.

3. Provide an output stream of data based on the user's interactions that can be used to power subsequent animations or lighting displays.

AudioSavant, much like paper-based music map, intends to serve as a set of listening "training wheels". The application presents a composition as students hear it: as a knot of sound, and provides the simple and intuitive tools required to untangle it. After using this application, students should be able to perform this untangling process on their own, simply by listening to a piece of music. AudioSavant, in essence, teaches them what to listen for in a piece of music, from simple to complex.

Because this project was conducted over the course of a single semester, we were not able to observe the use of AudioSavant in a live, educational setting. However, an initial user test of our interaction model showed the interface to be intuitive, and easy to navigate. The feedback we have received from our sponsors, and other sources in music education has also been positive thus far.

This paper is organized as follows: we present relevant work in Section 2, the AudioSavant interface in Section 3, and the back-end infrastructure in Section 4. We then describe a brief pilot study in Section 5, future work considerations in Section 6, and our conclusions in Section 7.



Section Overview	Main Riff (intro)	Main Melody	Main Riff (interlude)	Main Melody	Trumpet Solo
Section Duration	4 measures (14 seconds)	6 measures (22 seconds)	4 measures (14 seconds)	6 measures (22 seconds)	4+4+6+6 measures (72 seconds)
Trumpet	(silent)	Composed melodic duet with sax, using a 2-part hocket structure, built in repeating 2-measure phrases	Improvised duet with sax, using a call (trumpet) and response (sax) structure	Composed melodic duet with sax, using a 2-part hocket structure, built in repeating 2-measure phrases	Improvised solo; built in 2-measure phrases Section changes at 1:40 and 2:02 start with the same melodic gesture
Tenor Saxophone	(silent)	Composed melodic duet with trumpet, using a 2-part hocket structure, built in repeating 2-measure phrases	Improvised duet with trumpet, using a call (trumpet) and response (sax) structure	Composed melodic duet with trumpet, using a 2-part hocket structure, built in repeating 2-measure phrases	(silent)

Figure 1: The paper-based music map presented a composition in a grid structure. Instruments are featured on the y-axis, and sections of the piece are listed across the x-axis. Each cell provides a brief description of what a single instrument is doing in a specific section.

2. RELATED WORK

A full, extensive report could be dedicated to the myriad software tools that are currently available for music education. For our purposes, we will describe these tools in the context of five general characteristics that make them unsuitable for our users and goals.

Targeting Young Beginners and Old Experts

While young beginners and old experts likely represent the primary markets for music education tools, they are not representative of the entire spectrum of music education needs. Tools such as Adventures in Musicland [5], Cloud 9 Music [6], and Ricochet [10] provide many of the necessary introductory principles, but have a childish design that would likely be off-putting to older users. Conversely, tools such as inForm [7], Musique [9], and Music Skill Builder [8] dive straight into notation and instrument techniques. These principles are beyond the scope of our target users and their skill level. An example of these two types of tools is presented in Figure 2.

Automatic Instrument Separation

Music tools such as TamTam [17], GarageBand [3], and Rosegarden [18] present a composition as a list of instruments, each separated into its own track. This model, however, does not match the model that inexperienced users have of a musical composition, which we have already likened to a knot of sound. In the case of these tools, the software is unraveling the music for the user, potentially depriving them of the knowledge gained from doing it themselves.

No Identification of Structural Elements

We were not able to find any tools that could identify the structural elements between and within instruments. Instrument patterns such as repeating a sequence of notes, playing a call and response with another instrument, and harmonizing are critical to understanding the role that each instrument plays within the composition. While other tools provide excellent capabilities for composing and analyzing music, they do not supply users with this structural element identification capability.

Editing and Creation Capabilities

The ability to edit and compose musical arrangements is present in a wide range tools at every skill level. This capability, however, is beyond our scope. We are only interested in teaching users how to understand music, not compose it. Editing capabilities would allow inexperienced users to make changes that they might not fully understand. Not only would it be difficult to reverse these actions, but it may create a sense of anxiety while using the tool due to the lack of control. Thus the complexity that editing and creation tools add to a music application is unnecessary and unwanted for our purposes.

Insufficient Preparation for Future Tools

Perhaps the most intriguing and relevant tool we came across was TimeSketch (Figure 3) [11]. This tool employed a unique visual display that captured some of the basic structural elements that are of interest to this project. However, the drawback to this tool, and others we saw such as Tone Matrix [1], is that its representations are not consistent with more complex tools, and thus may not serve as adequate preparation for continued music education.

3. AUDIOSAVANT USER EXPERIENCE

The three goals of AudioSavant are to simplify the process of creating a music map, provide an interactive interface that makes exploring a composition easy and enjoyable, and support external uses of the manipulated data. In this section we describe these goals in greater detail, and present our solutions. The following is a complete walkthrough of the AudioSavant user experience.

3.1 Input

A critical aspect of any educational initiative is that the instructor's time is used wisely and efficiently. In creating the aforementioned paper-based music map, a music expert is required to identify elements of the composition that

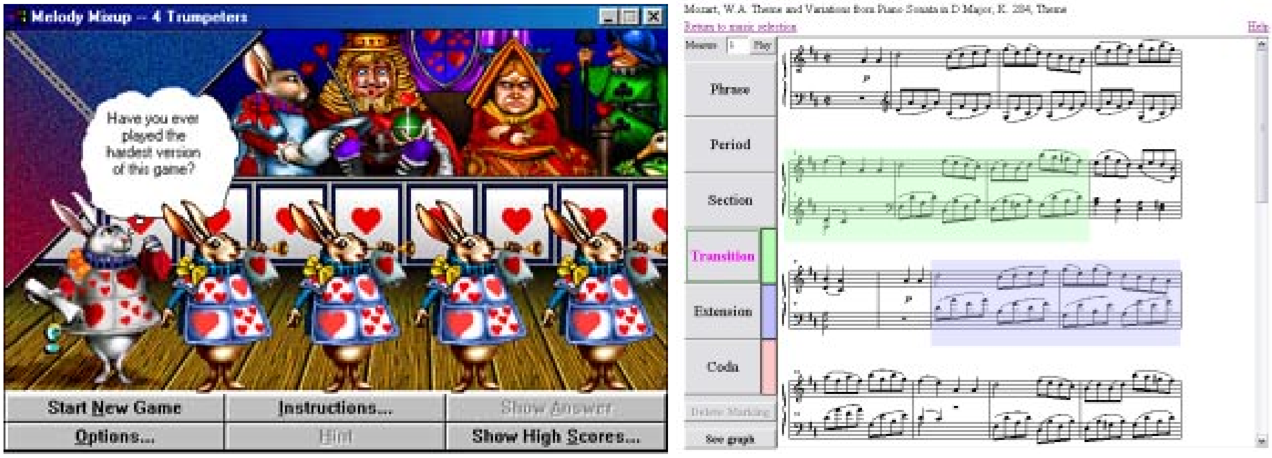
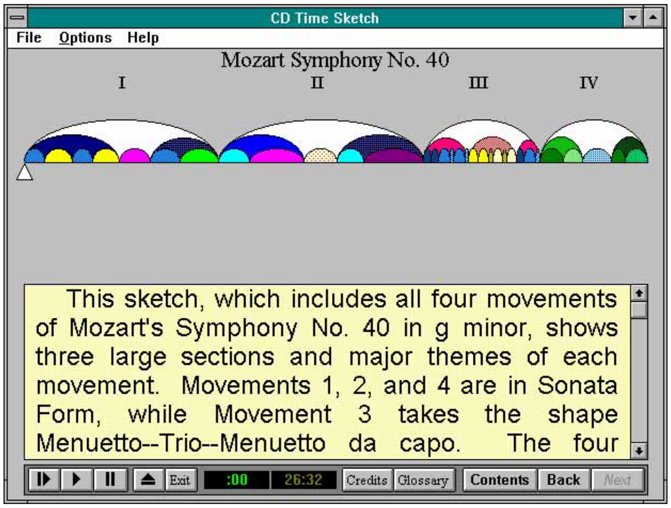

Melody Mixup -- 4 Trumpeters
Have you ever played the hardest version of this game?
Start New Game
Instructions...
Show Answer
Options...
Hint
Show High Scores...
Mozart, W.A. Theme and Variations from Piano Sonata in D Major, K. 284, Theme
Return to music selection
Help
Measure
Play
Phrase
Period
Section
Transition
Extension
Coda
Delete Marking
See graph

Figure 2: Adventures in Musicland uses pictures and animations to appeal to children (left). inForm uses complex music notation that may be confusing to novices (right).



This sketch, which includes all four movements of Mozart's Symphony No. 40 in g minor, shows three large sections and major themes of each movement. Movements 1, 2, and 4 are in Sonata Form, while Movement 3 takes the shape Menuetto--Trio--Menuetto da capo. The four

Figure 3: The TimeSketch interface uses arches to signify the different sections of a composition.

could have just as easily been identified automatically. Additionally, time is required to describe these elements textually since the paper-based map provides no visual interface. Through AudioSavant we hope to reduce this workload, freeing the instructor's time through simplified music map creation, while aiding the students directly through an intuitive educational user interface.

The input for AudioSavant can be created in Microsoft Excel, or by constructing a tab-delimited text file. This file has essentially the same structure as the paper based map, with the instruments listed down the left column and the sections listed across the top. However, significantly less information is required in each cell. Table 1 shows an example of such a streamlined music map input file.

Table 1: Music map input format.

Instr. Name	Instr. Num.	Measure: 1-12	13-24	25-36
Time	0	4/4	4/4	4/4
Piano	1	melody()	melody()	melody()
Guitar	2	tacet()	duet(3)	solo()
Bass	3	tacet()	duet(2)	mainriff()

AudioSavant automatically combines this metadata provided by the instructor with data taken directly from the music files (MIDI files). The tool is able to identify the mathematical aspects of the music such as repetitions of a given sequence, instruments playing in unison, and instruments playing in harmony. Thus, the instructor must only indicate higher level structural elements such as sections of the piece, timing, and interactions.

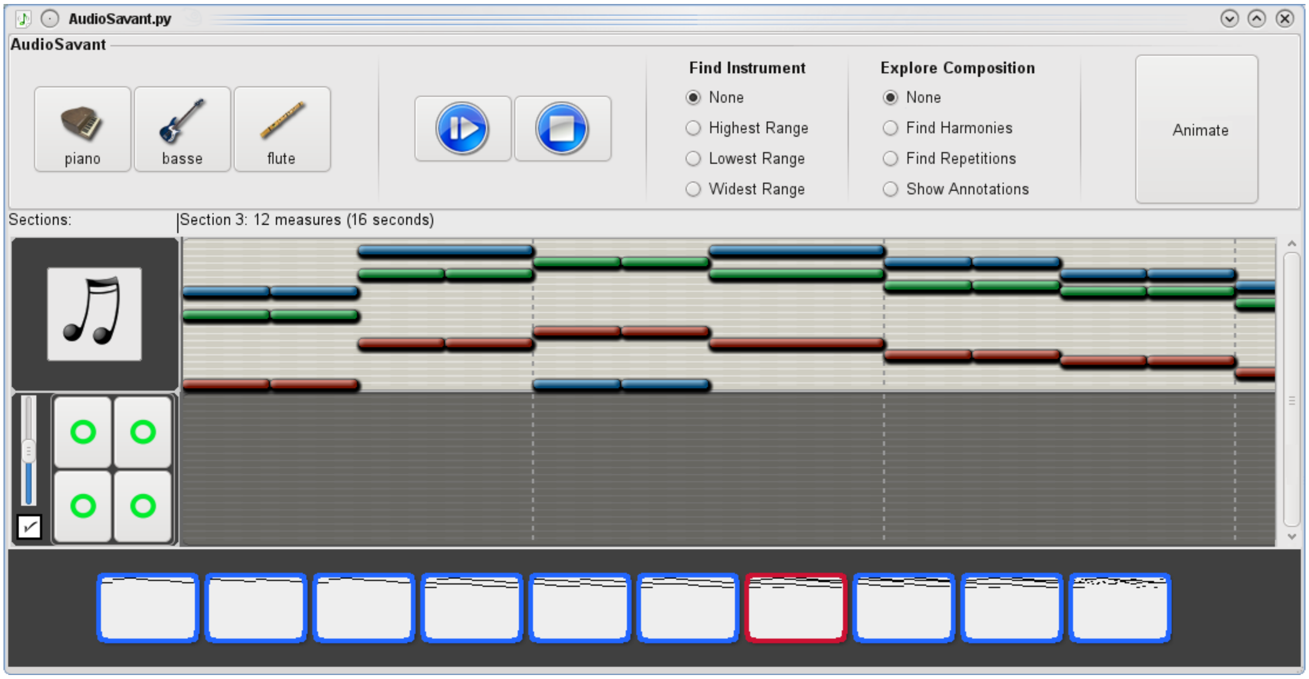
Though our initial design of these input files was structured to mimic the information in the paper-based music map, they can essentially be tailored to fit any need. Presumably with more time and more feedback from the actual music instructors, a universal and ideal input format could be developed. The purpose here is to demonstrate one example of a faster alternative to the paper-based format.

3.2 The AudioSavant Interface

Once the music and meta-data files are provided, AudioSavant loads the entire composition into a single track (Figure 4). Visually, this presents as a jumble of colored notes with no apparent structure or organization. While this may seem counterintuitive, to novices this model reflects what they are hearing aurally. By allowing students to visually untangle the composition (perhaps a less daunting prospect), they will also improve their ability to do so aurally.

3.2.1 Track Filtering

Users can begin to untangle the composition by filtering instruments into additional tracks or “filter tracks.” This is done by dragging and dropping up to four instruments into


AudioSavant.py

AudioSavant

Find Instrument
● None
○ Highest Range
○ Lowest Range
○ Widest Range

Explore Composition
● None
○ Find Harmonies
○ Find Repetitions
○ Show Annotations

Sections: |Section 3: 12 measures (16 seconds)

Figure 4: Above is the AudioSavant interface upon opening a composition. Notes are displayed knotted together in a single track. The "pages" along the bottom of the interface essentially serve as a scroll bar, allowing the user to jump to other areas of the composition. Each page icon provides a preview of what that page contains.

a filter track using the instrument icons in the control panel. Regardless of the new tracks created, the top track always represents the complete composition. When an instrument is moved into a filter track, its representation in the main track is grayed out, but remains visible. Based on feedback from our sponsors, we want to continually reinforce the idea that each individual instrument, though it has individual properties, is still a component of the overall composition.

To simplify the musical presentation, and to prevent users from getting lost in their analysis, each instrument can only be represented in one filter track apart from the main track. When an instrument is moved into a filter track, its corresponding button is grayed out in the top control panel. If the user wants to see this instrument in a different filter track, the instrument must be removed from the filter track in which it is currently displayed. This is done by simply clicking that instrument icon in its current filter track. Figure 5 shows the relationship between the main track and the filter tracks.

Filter tracks can also be used to control playback. Each filter track can have its volume increased, decreased, or muted. Thus the user is able to listen to any single instrument on its own, as well any grouping of instruments. This allows the user to understand the individual and interaction properties of every instrument.

3.2.2 *Selection and Highlighting*

AudioSavant allows users to identify structural elements of the composition, much like they were able to do with the paper-based music maps. The radio buttons in the control panel are used to highlight a feature of the user's choice within the composition. For example, by clicking the "Harmonies" button, all of the notes within the composition that are involved in a harmony will be highlighted. Students can use this information in order to make informed filtering decisions. If they see that two instruments are frequently playing in harmony, they can filter those two instruments into a separate track to explore this interaction more closely, or hear it in isolation.

Users can narrow this exploration further by using selection. Instead of highlighting a feature throughout the entire piece, users can select a specific area of the piece in which they are interested in seeing relevant interactions. This is done using either a double click to select the section they are in, a triple click to select a single track, or a click and drag to highlight any custom region. Selection uses color coded highlights to show how the selected sequence of notes relates to the entire composition. The user can hover over any highlight (either selection highlights or radio button initiated highlights) to see a tool tip that explicitly describes the type of interaction. Finally, any selection made in one of the filter tracks is mirrored in the main track to again remind users how their selection relates to the composition as a whole. A demonstration of selection and highlighting can be seen in Figure 6.

3.2.3 *Preparation*

Though AudioSavant is designed specifically for listening practice, its representations of notes and tracks are similar

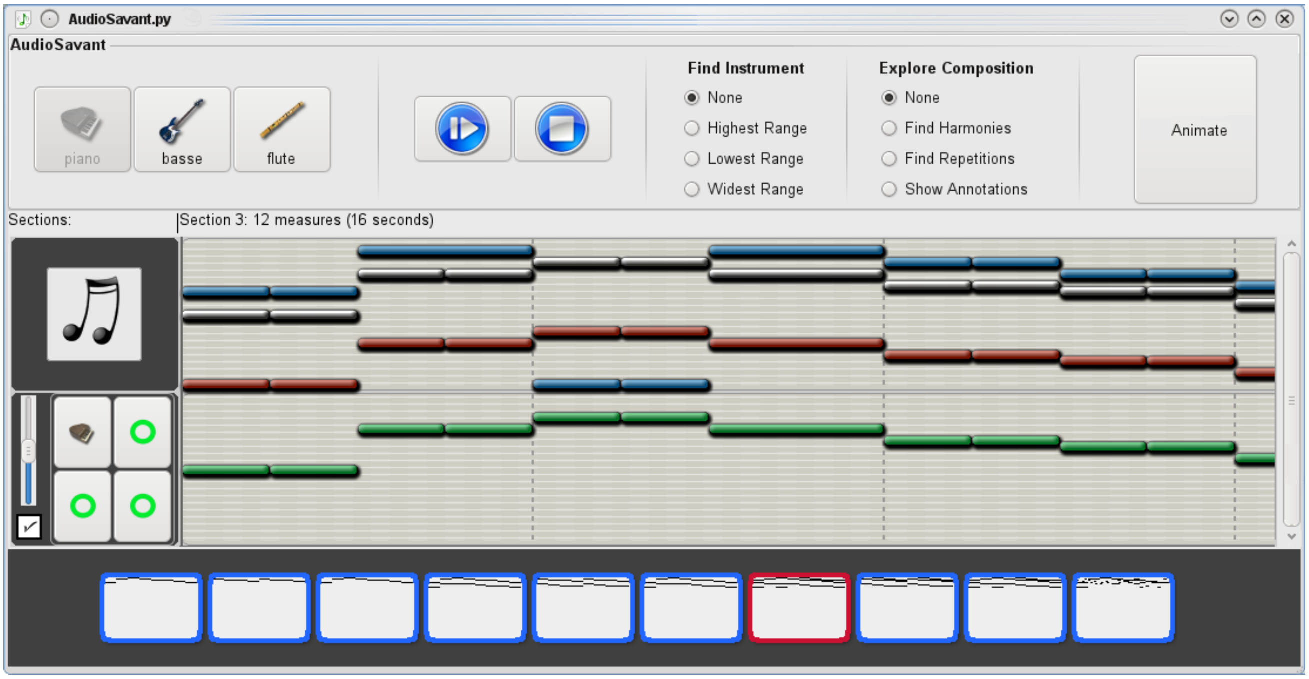


Figure 5: Filter tracks are activated by dragging instruments into them. When an instrument is moved into a filter track, it is disabled in the control panel and cannot be used in an additional filter track. The notes of the filtered instrument are grayed out in the main track.

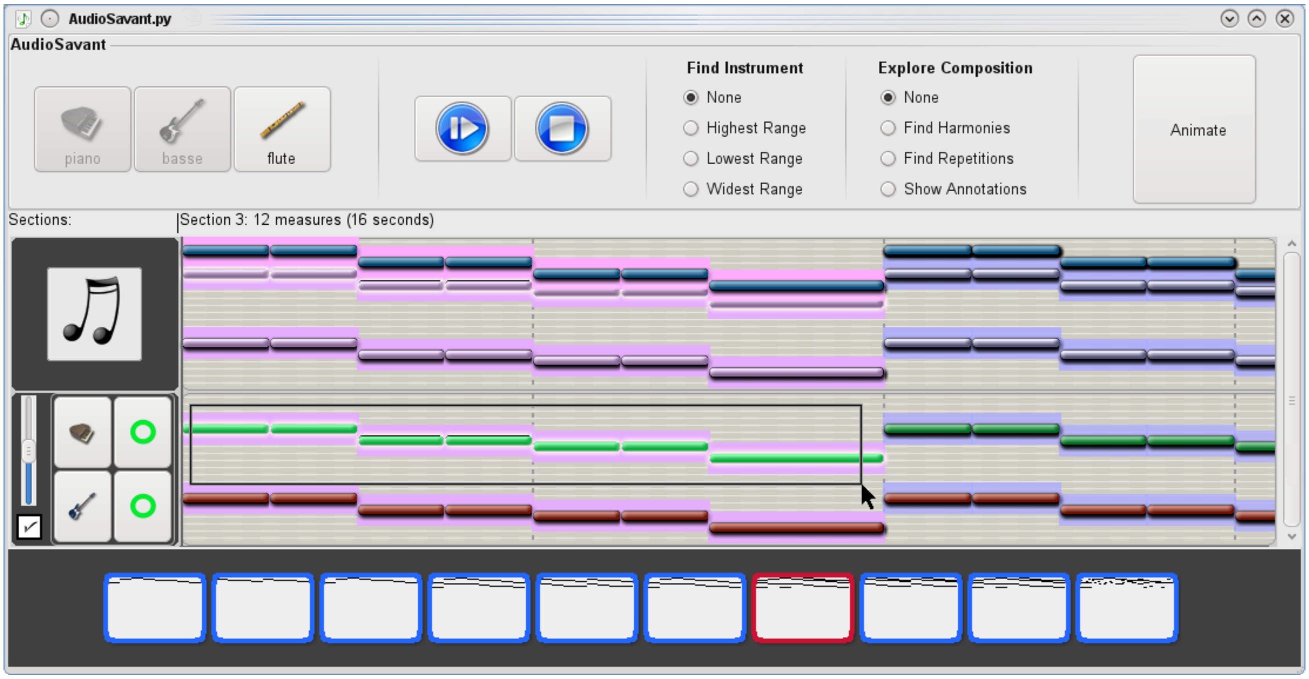


Figure 6: Selection shows how a specified sequence of notes relates to the rest of the composition. In this case, the selected sequence of piano notes plays in harmony with the flute and bass (highlighted in pink), and is repeated by both the piano and bass in other areas of the composition (highlighted in purple).

to those of more advanced applications that are designed for music playing and writing. Its representations are also similar to the standard bar/staff notation. Thus, users will not have to adjust to an entirely new set of schematics when they move on to other applications.

3.3 Output

The final feature of the AudioSavant user interface is its data output, which is controlled by a single button in the top right corner of the control panel. Pressing this button locks the state of the control panel (including tracks, volume, selection, and highlighted features) and generates a data file of the composition as it appears in that snapshot.

Using this output feature, external applications or a future version of AudioSavant itself can create different representations of the composition based on the user's data manipulations. The two most probable manifestations of this concept, as they have been discussed within this project, are:

- The creation of a lighting display around a live performance of the composition. For example, an AudioSavant user could specify that he is interested in harmonies involving the tuba and the trumpet between the 4th and 16th measures. Using the AudioSavant output stream, a lighting display could be implemented such that these two performers were the only ones lit during this section.

- The design of an animation to further visualize the composition. Considerable research in music education has shown that a metaphorical representation of musical structure can be very helpful [2, 13, 4]. For example, birds flying across the screen to represent high notes and fish swimming underwater to represent low notes. The AudioSavant output data could be used to power such an animation.

Though implementing these extensions was beyond the scope of our project, AudioSavant is designed to support a wide range of different options for visualizing musical structure. AudioSavant can be viewed not only as a music education tool, but also as a building application for a yet to be determined product.

4. INFRASTRUCTURE

Due to the short time frame in which this project had to be implemented, and considering our user needs and application goals, we used the music application TamTam as the model for AudioSavant. TamTam, a Python based application, was developed within the One Laptop Per Child program (OLPC), and is designed to provide music creation and instruction for young children. We chose TamTam, shown in Figure 7, because although the application is intended for younger children, it did not have a prohibitively childish interface. Additionally, the basic framework of TamTam allowed for an easy transition to some of the selection and tracking capabilties that AudioSavant required. This section provides a detailed description of the TamTam infrastructure, and its evolution into AudioSavant.

4.1 Extracting TamTam from Sugar

While TamTam was originally designed as a standalone GTK application, the authors eventually decided to include it into the OLPC project. The OLPC is a low cost laptop specially designed for children which runs a modified version of the Fedora Linux operating system which replaces a traditional desktop manager for an specially designed one called Sugar. Sugar is the base for the OLPC Human Interface and manages applications completely differently than a standard window manager in an attempt to make the experience easier and more entertaining for children. Unfortunately, Sugar is quite restrictive in many ways, particularly in its ability to run only one application at a time and scarce package availability for popular Linux distributions. Due to these shortcomings, we decided to remove TamTam's dependency on the Sugar platform, effectively "desugarizing" it, and rebuilt the project as a typical standalone GTK application.

Sugar offers a standard set of functionality to all the OLPC Activities. These functionalities vary from GUI Widgets (Windows, Buttons, Text Areas), to file storage capabilities and even connectivity with other children and their laptops.

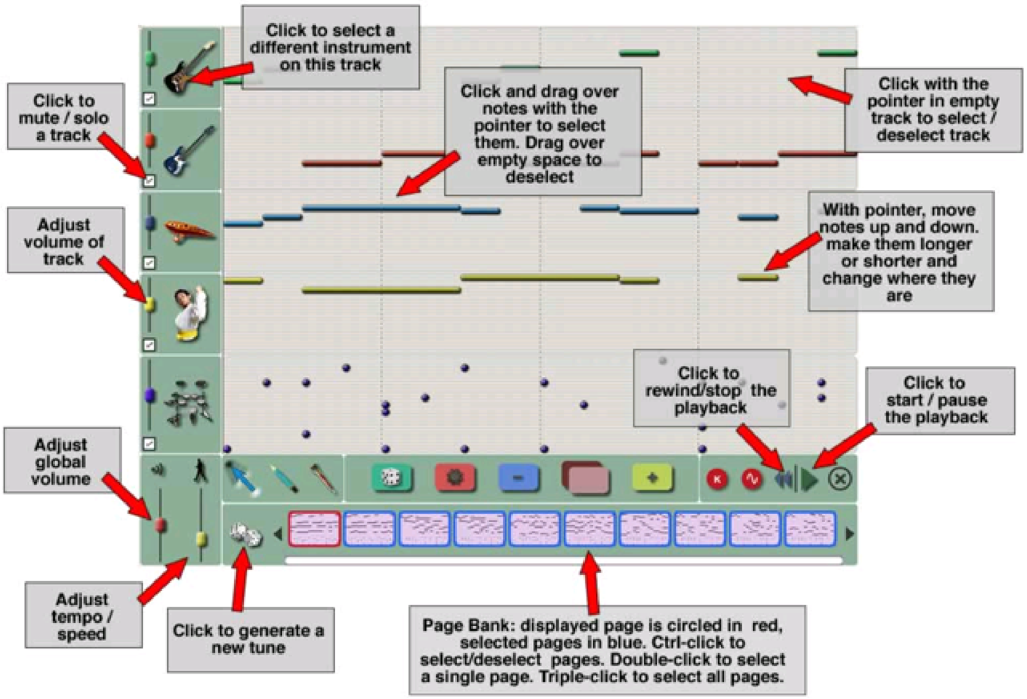
Our goal was to desugarize TamTam while preserving the majority of its original code. To this end, instead of replacing all the references to the Sugar library, we decided to create a dummy transition library called *Desugarize.py* that provides the same functions and classes as Sugar, while using only GTK primitives in its implementation. For example, a Sugar *Activity* (the OLPC term for "application") must create an instance of the class *Sugar.Activity*, which represents the main windows of the application. The Desugarize.Activity class offers the same class prototype but a different implementation, so instead of generating a Sugar window, it just generates a traditional GTK.Frame.

The final result of this process was a TamTam version without major modifications to the core code base that could run on nearly any Linux desktop as a GTK standalone application. This is the code base which we used to start developing AudioSavant.

4.2 Stripping Down TamTam

While TamTam was lacking many features required for an educational listening tool, it also contained features that were unnecessary and potentially confusing to music students. The first feature we stripped from TamTam was the editing capability. AudioSavant is a listening tool, not a musical composition application. In TamTam, a user could create new notes, slide them up and down in pitch, and left to right in time. As we previously mentioned, AudioSavant requires notes to be pre-specified with an input file, and they may not be modified during usage. This allows users to concentrate on the learning and visualization aspects of the tool, and protects them from accidental modification.

Additionally, TamTam users are able to choose instruments on-the-fly for each of the playing tracks. TamTam also provides a "secondary" instrument in a split view contained in the same track. Since the musical pieces we are visualizing are based on specific instruments, the ability to modify the playing tracks (and to specify such secondary tracks) was useless, and once again, potentially confusing to the



Click to select a different instrument on this track

Click to mute / solo a track

Click and drag over notes with the pointer to select them. Drag over empty space to deselect

Click with the pointer in empty track to select / deselect track

Adjust volume of track

With pointer, move notes up and down. make them longer or shorter and change where they are

Click to rewind/stop the playback

Click to start / pause the playback

Adjust global volume

Adjust tempo / speed

Click to generate a new tune

Page Bank: displayed page is circled in red, selected pages in blue. Ctrl-click to select/deselect pages. Double-click to select a single page. Triple-click to select all pages.

Figure 7: AudioSavant was created by stripping TamTam down to its basic functionality and building it back up around our users and goals.

users. Therefore, the instrument selection feature has been removed for AudioSavant.

Finally, in the process of desugarizing TamTam, we encountered myriad calls to miscellaneous functionality which is not useful for our effort, such as network connectivity and storage access. Such functionality may be useful in the future, for example by expanding AudioSavant to load files at runtime, but it was not appropriate for the goals we have set for this project. During the desugarization process, the method calls for such functionality were stripped from the code base, and the corresponding implementations were not recreated in the *Desugarize* library.

4.3 Audio File Manipulations

As discussed in Section 3, one of the primary goals of AudioSavant is to allow skilled musicians and music instructors to create textual encodings of high-level compositional elements in an intuitive and efficient manner. However, these annotations are of little use by themselves: users must be shown the *connection* between abstract musical concepts and the notes in a given piece that express those concepts. To this end, the music map data must be unified with a representation of audio/note data for the piece charted by that map before it can be represented by AudioSavant.

Music map and note data are merged by preprocessing prior to being loaded into AudioSavant. This preprocessing pipeline takes as input a standard MIDI (Musical Instrument Digital Interface) [15] file and a corresponding textual music map file, and outputs a score file that is consumed by AudioSavant. Standard MIDI files were selected as the input format because they use representation of note data that can be both read and written by a wide variety of popular music composition and sequencing tools, making AudioSavant input data generation relative easy; additionally, a large number of MIDI files can be readily found on the Internet, further simplifying the instructor's task. [1]

While many composition tools can process MIDI files, TamTam does not have support for file-encoded MIDI data; instead, it uses a representation based on the CSound audio language [12]. Rather than integrating MIDI file parsing capabilities into AudioSavant, a separate translation step is carried out in the first stages of audio processing, converting MIDI note data into a textual form that is largely similar to that utilized by TamTam's own internal music model. This is done using the free `midi2sco` utility by Pete Goodeve [14] that outputs notes as CSound data points. While CSound's textual representation is easier to manipulate than MIDI files (which are stored as collections of byte streams), tool support for CSound data is limited, making MIDI a superior format for initial note data input.

The resulting CSound file is then processed by a custom

[1] Raw audio files (such as CD audio or MP3 files) were not considered a viable option for AudioSavant input, as the implementation and application of the sort of information extraction and quantification techniques required to apply the pattern recognizers used by AudioSavant to raw audio would have resulted in a truly immense increase in overall program complexity without significantly contributing to the visual aspects of the tool.

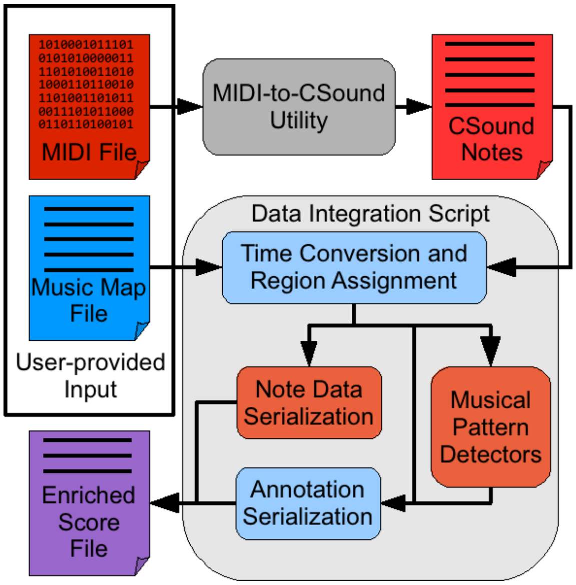


Figure 8: Music map and note data are merged before being loaded by AudioSavant. Components in blue are driven mainly by annotation and timing data, which are provided primarily by the music map; red components are related to note data, which are provided by the MIDI file.

merging program that reconciles the various representations of time in the input files into one compatible with TamTam's (and thus AudioSavant's) internal model of note data. Similar annotations in the music map are consolidated and linked to the regions of musical time to which they correspond.

The pattern detectors described in Section 3.1 are applied to note data on a per-measure basis, creating an inexact but complete characterization of certain common musical devices within the piece. This is done not only to provide an initial analytical basis for AudioSavant users, but also to avoid an all-to-all measure comparison at run-time. The detectors use a flexible, sample-based approach to note sequence alignment so as to allow for comparison across disparate time signatures and tempos. They employ heuristic scoring techniques to characterize the relationship between two sequences of notes, and can perform simple sliding-window comparisons when searching for atemporal musical devices (like repetition). The detector data model and base API are relatively straightforward, and could easily be extended to included pattern recognizers for compositional elements beyond harmony, unison, and repetition (for which detectors already exist).

The combined converted note, annotation, and pattern data is serialized, and written to a single "enriched" score file that AudioSavant can accept as input. This file format represents assinged and detected attributes as a graph (where regions of the score are nodes and shared properties are edges), allowing for easy addition of new musical patterns and properties to AudioSavant's internal music model.

4.4 Building Up AudioSavant

AudioSavant was built up using TamTam's basic track and note drawing functionality. TamTam draws each instrument using a separate color and separate track. The notes are drawn using raised lines that clearly show the start and end of each sound. Our modifications allow the notes from different instruments to be displayed within the same track, and moved between tracks. We also modified the color scheme such that instruments in filter tracks appear less prominently in the main track.

TamTam also had some basic selecting and highlighting functionality built in to its editing interface. We rewired this functionality into the AudioSavant control panel such that selecting and highlighting did not modify the music. We also conceptually separated these functions such that notes that were selected were not automatically highlighted. The AudioSavant selection function changes the track background behind the selected notes, and the highlighting function is controlled by the radio buttons in the control panel.

Finally, we completely rebuilt the TamTam control panel to support the critical features for our users and tasks. We eliminated TamTam's concept of multiple activities, and implemented only a single activity, audio exploration. The result was a simple, interactive control panel where the user can see and evaluate their full set of options at all times.

5. USER STUDY

In the time provided for this project, we were not able to conduct user tests on a working version of the software. However, in order to test our basic interaction model, we conducted a user study using lo-fi mock-ups of the AudioSavant interface. We were able to mimic the functionality of our interface surprisingly well using stacked transparencies and external audio playback. User interactions were simulated by adding, moving, and removing the layered transparencies. We recruited five participants, ranging in age from ten years old to fifty six years old, and asked them to perform the following three tasks.

1. Play the drum and bass parts in the first two sections of the piece.
2. Describe the interactions taking place in the third section of the piece.
3. Play the section of the piece in which the piano has a solo part.

Since this application is intended to be introduced in a classroom setting, we gave the participants a brief overview of the control panel before they began. All of the participants were familiar with playing music using applications such as iTunes and Windows Media Player, but none of them had any extensive music training involving notation or composition.

All of the participants found the track creation and selection capabilities to be very intuitive. There were no problems observed while navigating the control panel and completing the first task. In the second task, users had difficulty when there was more than one interaction taking place at the same time (in this case, a call and response interaction and a unison interaction). We believe that this could be solved by adding tooltips when the user hovers over a highlighted item, explaining the type of interaction that is occurring.

Finally, the third task was completed with little incident. Most users scanned through the piece, looking for the other instruments to drop off. Using this strategy, they were able to find the piano solo quickly. One user moved the piano into its own track and attempted to find the solo this way. This strategy worked, but took significantly longer. Once the solo was found, none of the participants had difficulty selecting and playing that section.

Overall, the interface proved to be intuitive and engaging. Participants liked the reversibility of the track creation feature. Many of them created and deleted multiple tracks as they explored the interface and worked through their tasks. They felt comfortable trying a potentially incorrect path, knowing that they could quickly pursue another option if it did not work out. AudioSavant, therefore, appeared to encourage self-motivated exploration, and would not require much additional instruction after the control panel is introduced.

6. FUTURE WORK

As AudioSavant was developed as part of a semester project, we were only able to implement a limited number of features. However, through our work we developed ideas for four possible extentions that could be implemented in future versions.

Increased Automated Analysis

Though we were able to reduce the amount of work required of expert users (music instructors) to create an AudioSavant music map, we believe that this reduction could be taken even further. Using more advanced audio analysis tools and further feedback from music instructors, we believe that the process of identifying important structural elements within a composition could be almost fully automated. It may also be reasonable to develop a standardized data format for future music map audio files that encorporates both audio and structural data from the onset.

Composition Comparisons

AudioSavant allows users to explore how each individual instrument relates to the composition as a whole. It may be equally engaging and educational to explore the relationships between two different compositions. This would be especially interesting for songs that are quite similar. For example, *Boulevard of Broken Dreams* by Green Day has the exact same chord progression as *Wonderwall* by Oasis. Seeing these two songs side by side could produce valuable insight about how these two songs are still able to sound unique.

Internal Animation Support

We have already discussed AudioSavant's output capabilities. Ideally, this data stream would eventually be used by AudioSavant itself to create an animated representation of the composition based on the user's manipulations. Such an animation would serve primarily as an extension to the highlighting feature. Currently, highlighting an element in the piece (such as an instrument, or a section of notes) provides visual feedback in the track, but does not highlight that element in the audio playback. An animation could highlight these elements more clearly in unison with the audio playback.

Educational Exercises

Similar to other music education tools, we would like to eventually encorporate class-like excersises into the AudioSavant interface. For example, the application would identify one part of a harmony, and the user would have to identify the other instruments that take part in that harmony. Users could also be prompted to identify sections of the composition on their own, before the application reveals that information to them.

Custom Code Base

Though TamTam provided some useful selection and tracking capabilities, the application ultimately proved to be quite fragile. Due to its system requirements and dependencies, TamTam did not lend well towards building a universally usable application. Our recommendation for future work would be to build a more stable infrastructure from scratch.

7. CONCLUSIONS

AudioSavant was designed to provide a substantial amount of information to inexperienced users while requiring a minimal amount of input from expert users. To this end, we have automated the process of identifying basic structural information within a musical composition. We have also enabled the transfer of this information to external applications using a flexible and universal data format.

AudioSavant provides inexperienced music students with the opportunity to understand a composition aurally by exploring it visually. Through intuitive representations and interactive feedback, students have the freedom to unravel and examine a piece of music without the fear of altering it or taking steps that cannot be reversed. Our hope is that this freedom encourages students to use this application on their own, without requiring perpetual instruction. Furthermore, our hope is that students will gain a deeper understanding of musical structure and interaction that will enable a smooth transition into musical performance and composition.

8. ACKNOWLEDGEMENTS

We would like to thank Allison Druin and Greg Walsh of The University of Maryland's HCIL for allowing us to develop our own ideas and solutions within their ongoing work with Carnegie Hall and The Weill Music Institute. We would also like to thank our interface evaluation participants, paper reviewers, and members of The University of Maryland's Music Education Department for their helpful feedback and guidance.

9. REFERENCES

[1] André Michelle Laboratory. http://lab.andre-michelle.com/tonematrix.

[2] A. Antle, D. M. Milena, and G. Corness. Playing with the sound maker: Do embodied metaphors help children learn? In *Proceedings of ACM International Conference on Interaction Design*, 2008.

[3] Apple - GarageBand. http://www.apple.com/ilife/garageband/.

[4] T. Bergstrom, K. Karahalios, and J. C. Hart. Isochords: Visualizing structure in music. In *Proceedings of Graphics Interface 2007*, 2007.

[5] ESC Media - Adventures in Musicland. http://www.ecsmedia.com/indivprods/musaim.shtml.

[6] ESC Media - Cloud 9 Music. http://www.ecsmedia.com/indivprods/cloud9.shtml.

[7] ESC Media - inForm. http://www.ecsmedia.com/indivprods/inform.shtml.

[8] ESC Media - Music Skill Builder. http://www.ecsmedia.com/indivprods/MusicSkillBuilder.shtml.

[9] ESC Media - Musique. http://www.ecsmedia.com/indivprods/musmusique.shtml.

[10] ESC Media - Ricochet. http://www.ecsmedia.com/indivprods/musricochet.shtml.

[11] ESC Media - TimeSketch. http://www.ecsmedia.com/indivprods/tseditor.shtml.

[12] B. V. et al. The canonical csound reference manual. http://www.csounds.com/manual/html/index.html.

[13] T. Fischer and W. Lau. Marble track music sequencers for children. In *Proceedings of the 2006 conference on Interaction design and children*, 2006.

[14] P. Goodeve. midi2sco. http://www.goodeveca.net/midi2sco/, May 2008.

[15] P. Goodliffe. Midi documentation. http://prdownloads.sourceforge.net/tse3/midi.ps.gz, October 1999.

[16] I. McKinnon. Childrens music journey: the development of an interactive software solution for early childhood music education. *Comput. Entertain.*, 3(4), 2005.

[17] OLPC - TamTam. http://wiki.laptop.org/go/TamTam.

[18] Rosegarden. http://www.rosegardenmusic.com//.

[19] J. Sterling. A cognitive approach to the teaching of musical form using computer assisted instruction. In *Proceedings of the Eighth International Technological Directions in Music Learning Conference*, 2001.

[20] N. Weinberger. The music in our minds. *Educational Leadership*, 128.200.122.84, 1998.