2013 IEEE 37th Annual Computer Software and Applications Conference

Implementation Considerations in Enabling Visually Impaired Musicians to Read Sheet Music Using a Tablet

Laura Housley, Thomas Lynch, Rajiv Ramnath, Jayashree Ramanathan
Computer Science and Engineering
The Ohio State University
Columbus, Ohio USA
{housley.7, lynch.268, ramnath.6, ramanathan.2}@osu.edu

Peter F Rogers
Engineering Education Innovation Center
The Ohio State University
Columbus, Ohio USA
rogers.693@osu.edu

***Abstract*— In this paper, we present the issues to be addressed and the practical solutions to these issues in a mobile application framework for reading and displaying musical scores enhanced to assist the visually impaired in reading and perform the pieces. This framework, currently operating on MusicXML input files, provides the structures and methods for developers to adapt for other music encoding file formats. It also provides the flexible user-settable colors and enlargement parameters to meet the needs of users with various visual impairments. The development challenges fall into three categories: Variable visual impairment driven requirements; Musical notation complexity, and screen real-estate limitations of a 10-inch tablet. The framework's practical solutions to each of these challenges are presented and contrasted with traditional solutions and competing solutions.**

Keywords-accessibilit; mobile computin)

I. INTRODUCTION

Most musicians read sheet music to perform and enjoy the complex works of current and historic composers. Though there are great musicians throughout the ages that were blind, losing the visual ability to read sheet music through age or accident can be devastating to musicians. Unlike losing the visual ability to read written words, screen readers and physical notation (similar to Braille) cannot provide the need assistance. The screen reader would be impractical given the amount verbalization required to communicate the music score and thus would disrupt the tempo. Braille-like notation on the other hand would require the musician to perform one handed thus interfering with the playing of the music.

Currently sheet music is printed with black ink on 9x12 inch white paper with staff lines 1/16th of an inch apart. This format fails the needs of readers with low vision and other visual impairments. Simply enlarging the score provides some relief for some music readers; however it fails to fulfill the needs for many users for several reasons including becoming too cumbersome as the paper size increases and/or requiring too many and too frequent page turns disrupting the musician's performance.

Cataracts and other visual impairments may also require different color schemes according to Elliott. [1] Printing the scores in a multitude of color schemes is impractical due to the limited demand compared to the cost.

There are mechanical devices dedicated to aid visually impaired musicians but they tend to be expensive and heavy. For example, the Lime Lighter is a standalone tablet-like device that displays enlarged music that retails around $3,995. This price does not include the software that costs several hundred more dollars (Lime).

These issues motivate a solution that dynamically generates sheet music in the desired size and color set by the user. An application that runs on a tablet device (that fits easily on a standard music stand) would allow the user to immediately see the music score resulting from the setting they choose. Furthermore, such an application can automatically scroll the score across the screen at the desired tempo eliminating page turns and allowing the musician to concentrate on performing the music. Creating an application that runs on a multi-functional tablet provides accessibility to people in a larger range of economic circumstance than an expensive, single-use dedicated device.

This research focuses on the needed affordances and framework required to generate musical scores on an Android tablet from a text-based input (MusicXML) and displaying the resulting score at various magnifications and color schemes. The score generated must correctly position and space musical symbols found in complex compositions. The output must be aesthetically pleasing and resemble regular sheet music. The score must scroll across the screen at the tempo specified by the user. Ideally the framework would be readily adaptable to other musical formats besides MusicXML. Most importantly, this system must allow the visually impaired to continue to read, perform and enjoy musical scores.

II. BACKGROUND INFORMATION

Before we can address the challenges of this research, the reader needs general knowledge about musical notation and the challenges it presents. The following section discusses the basic elements of music notation, and how they are structured and interrelated to form sheet music.

Also the reader will need insight to the different obstacles faced by the visually impaired and their relationship to reading sheet music. Simply making the notes bigger does

0730-3157/13 $26.00 © 2013 IEEE
DOI 10.1109/COMPSAC.2013.107

not solve the problem and introduces new problems. The subsequent section discusses the issues and consideration of addressing the needs of the visually impaired.

A. Basic Sheet Music

Sheet Music as shown in Fig. 1 is a representation of a piece of music using symbols and accepted musical notation so that two users can read the same piece of sheet music and output the same song. There is some room for interpretation by the musician, but the timing, pitch and duration that each note should be played is explicitly stated. This allows many musicians to play the same music simultaneously in a group setting such as an orchestra with positive results. It should be noted that music is always read left to right, in a top to bottom manner.

Sheet Music is separated into multiple staffs or staves as shown in Fig. 2, to provide a convenient way to specify that multiple notes are played simultaneously by one or more performers. As shown in Fig. 1 (a single staff) and Fig. 2, each staff has five staff lines enclosing four spaces. These lines and spaces are used to specify the pitches of the notes. If two notes are aligned vertically, they should be played simultaneously. These staffs are representative of one or more parts. For example, a musical piece with three staffs can represent two parts: the first part containing one staff could represent the melody and lyrics, and the second containing two staffs, the notes for the piano accompaniment. In this instance, someone could sing the song while someone else plays the piano from the second part.

The length of time a note should be played is indicated by its shading, stem, and flags (Fig. 3). Whole notes are not shaded and have no stem. Half notes are played for half as long as whole notes, and quarter notes are played for a quarter of the time of a whole note. Half notes are not shaded but have a stem. Quarter notes are shaded and also have a stem. Eighth, sixteenth, and thirty-second notes follow the same pattern. For each type of note, there is a corresponding type of rest. Rests occur when no note is being played for the specified duration.

Notes whose lengths are in between the simple durations are marked with a trailing dot indicating its duration is 1.5 times as long as it would be without a dot. For example, a dotted quarter note has the length of one quarter note plus one eighth note.

The vertical position of a note in relation to the staff lines indicates the pitch of the note. Within a single staff, a higher note represents a higher pitch. Sometimes notes need flat, natural, or sharp signs next to them to slightly alter the pitch. An understanding of the specific pitches in a piece of music is not necessary for the discussion of how to display sheet music.

Left to right on each staff, the music is divided into logical groups called measures symbolized by a solid vertical line from the top of line to the bottom of the staff. Measures help the reader keep track of the beat. Each piece of music has a time signature which describes how many notes of each type are included in each measure. Time signatures consist of two numbers: the bottom represents the type of note the signature is based on, and the top number represents how many of this base note (beats) are in each measure. For instance, a time signature of 3/4 indicates that there will be three evenly spaced beats per measure and the quarter note or its equivalent represents a single beat. A 3/2 time signature means there are three beats per measure and the half note gets a beat. This is just the length of each measure, so these three quarter notes could be written as six eighth notes or one half note and one quarter note. All that matters is that the total duration of the notes and rests adds to three quarter notes for a 3/4 measure.

Beams are used to connect the flags of notes and prevent a piece of music from being filled up with flags off the stems. When several notes are to be played in a row, they are connected together to show they are played as a group as show in Fig. 4. Eighth notes use single beams, sixteenth notes use double beams and thirty-second notes use triple beams.

There are many other symbols found in musical notation, but an understanding of how notes and rests are displayed is enough to follow the general procedure for displaying music. In Fig. 2 some of the discussed musical symbols are labeled in a sample piece of music. This piece has two parts - one of which has one staff and the other has two staves. Three and a half measures are at least partially visible in this screen capture.

B. Visual Impairments and Music

According to the American Foundation for the Blind an estimated 10 million people in the United States are blind or visually impaired. [2] They estimated that 80% of the 10 million have some useful vision. Those with some useful vision have different type of impairments that impact their ability to read regular sheet music. These impairments can cause low vision, small field of vision, and lack of focus.

Low vision takes many forms, and does not always mean that the person's vision is faded equally. For example, some people lose their central vision, peripheral vision, or night vision. Other examples of issues causing low vision are tunnel vision, distortion, and multiple field loss. Multiple field loss means that there are multiple spots missing from a person's field of vision.

There are many possible causes of low vision. Different diseases may lead to different impairments. For instance, macular degeneration causes blurred vision and possibly

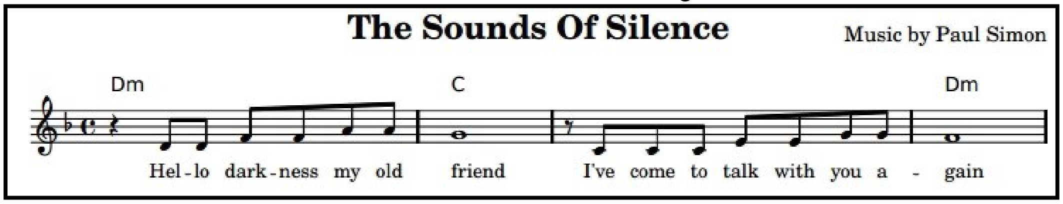

Figure 1. Sample Sheet Music

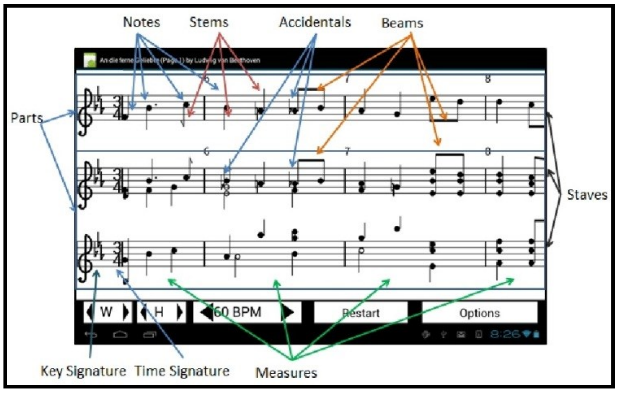

Figure 2. Parts of a musical score.

symptoms of night blindness and loss of peripheral vision. Others with low vision have issues seeing different color combinations. For example, yellow on black might be preferable to some people, but other people would be unable to distinguish between these colors enough to read the music. These visual impairments require changes to sheet music.

Based on visual acuity alone, low vision requires that sheet music be magnified. The color issues require customizability of the colors of the sheet music as some would prefer a light background with dark text, and others would prefer a dark background with light text.

Issues with field of vision present challenges for scrolling the music. Musicians read ahead of the point they are playing so they need as many future measures available as possible. By scrolling the music instead of paging allows continuously displaying future measures timely. Because a person with blind spots cannot see the entire screen at once, it is important that the music be scrolled in a way that people can continuously read the music. Continuous smooth-scrolling achieves this effect, since the music that is being played is always in the same relative position. Shifting the music one or more measures at a time, causing the music to jump would not be acceptable for people with a low field of vision. They could possibly encounter difficulties tracking their position in the score. However, shifting instead of scrolling could be better for some people who find scrolling/moving music difficult to read. Thus affordances for both features are required.

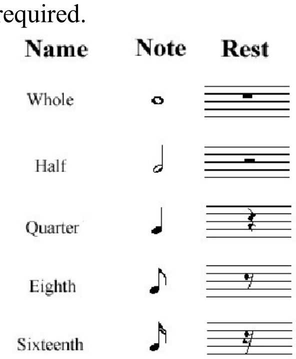

Figure 3. Notes and Rests.

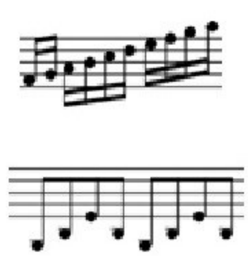

Figure 4. Beamed Notes.

III. Related Work

Many researches have explored usability challenges posed by current mobile devices for the visually impaired. [2] [3] [4] [5] They have also explored adaptive technology and affordances to improve the experience for the visually impaired. Markus, et al has researched and developed various apps to assist the visually impaired use the current devices. [5] Thus the target user group is able to use the tablet device despite of their impairments.

No research was found that specifically addressed assisting the visually impaired to read music using a mobile tablet computer. However there is a standalone dedicated tablet-like device that displays enlarged music called Lime Lighter and requires a piece of software called Lime. Together they retail for over $4,000. The price alone limits the access to this solution. The tablet solution is more affordable and Kane found that people with special needs such as the visually impaired would rather have non-dedicated devices. [4] The non-dedicated device means not having to carry around another device and it gives them a larger variety to choose from.

Finally, using a laptop or a desktop computer to display musical scores in place of paper scores would require new music stands at least and possibly expensive touch screen computers since mice and keyboards would be impractical. Though feasible, this is an expensive solution.

This software will dynamically generate the musical score based on the user's preferences. Work has been done in Adaptive Document Layout (ADL) to address the challenges in dynamically laying out documents on tablet-like devices. Most of what has been done has focused on laying out text. However, other items in documents such as images can be dynamically placed based on screen size as well. [6] [7] The research used a nesting approach to group the related items in parent child relationships to maintain the required item groupings essential to the documents.

Our application implements the techniques of ADL to develop the sizing and layout methods for the dynamic generation of the scores. As explained in the preceding section, the physical relationships of the components of the score must be maintained when generating the sheet music to maintain the integrity of the music. Like the adaptive document layout algorithm, the score is kept in a tree like structure to maintain the relationships between the components.

IV. Implementation

The application runs in four stages: parsing the XML into the music data structure, building the component hierarchy from the music data structure, generating the image from the component hierarchy, and finally scrolling the bitmap across the screen at the set tempo.

A. Music Data Structure

The MusicXML file's top element is the Score. The Score consists of one or more Parts, and each Part consists of one or more Measures. The Measure contains one or more Notes (which include rests) which can be beamed or tided together. The measures contain a width which usually varies measure to measure. The notes contain an X location specifying the notes horizontal location relative to the left edge of the measure. The vertical location of the note is specified as a pitch and octave for mapping onto the staff. The sample score shown in Fig. 1 contains a quarter rest and six eighth notes joined by two beams. The data for the first measure of the sample score is shown in Table 1. Fig. 5 shows a sample of a portion of the MusicXML file for this score.

B. Musical Score Framework

The structure of the framework is similar to the structure of the MusicXML file. The Score is composed of Parts which in turn are composed of Measures which contain Staves containing Notes, which have various attributes. However we deviate from MusicXML in that beams are members of the Measure object since they span notes, while ties and slurs are members of the Part object since they can span measures.

The Note object contains the parts of the note including the Head, Stem, Dot, and Accidental. The Dot and Accidental are enhancements to the note. The accidental is placed before the note and the dot immediately follows the note. The X location specified in the MusicXML file is specified with considerate of the space the accidental and dot require avoiding collisions between notes and the note enhancements.

Due to the relationships between the parts of the musical score, measures of the various parts align and the placement of the notes on each staff must align correctly in the vertical plane to properly encode the score to insure that the performance of the music is correct. The Android SDK provides a flexible layout method called RelativeLayout that allows the object's child views to properly align. This application exploits this feature to align the views from each of the children. Fig. 6 shows the RelativeLayout hierarchy of this application.

The Score is "printed" into a layout hierarchy by calling each its immediate children's print methods with that child's width and height allocation. The child divides its allocated space for its immediate children and calls each child's print method with their height and width allocation. This continues on down to the lowest level node children. Each node child generates an image of its self and then returns the image to its parent which assembles the children's images into RelativeLayouts, adds any additional graphics and returns the resulting RelativeLayout to the calling parent.

This tree method simplifies the sizing of the score. Each object only needs to know their allocation and how to allocate for their children.

TABLE I. NOTE INFORMATION FOR THE FIRST MEASURE OF "THE SOUNDS OF SILENCE" AS ENCODED IN MUSICXML.

Measure width="413" number="1"							
Note	**Pitch**		**Dur.**	**Type**	**Beam**	**Lyric**	
default-x	***step***	***octave***				***text***	***syllabic***
108	rest		4	quarter			
171	D	4	2	eighth	Begin	Hel	begin
201	D	4	2	eighth	End	lo	end
251	F	4	2	eighth	Begin	dark	begin
291	F	4	2	eighth	Continue	ness	end
313	A	4	2	eighth	Continue	my	single
371	A	4	2	eighth	End	old	single

Figure 5 .Sample MusicXML

C. Measure Width

MusicXML files give the width of each measure as shown in Fig. 5. This usually differs from measure to measure since the MusicXML exactly reproduces the original printed musical score. In the sample score in Fig.1 measures one and three are wide while two and four are narrow. Having variable width measures is fine when the music is stationary and scrolling is done with the eyes, but this poses a problem when scrolling is done automatically.

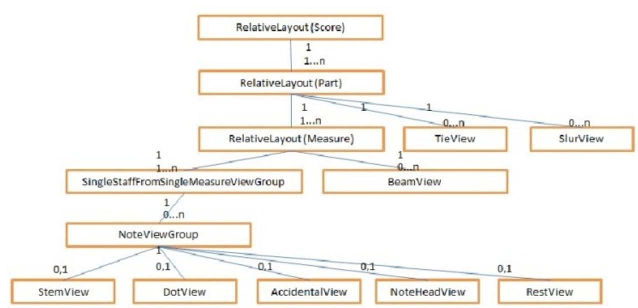

Figure 6. View and Layout Hierarchy

Because each measure should be on the screen for the same amount of time, a fixed measure width is required, otherwise the speed of scrolling would need to constantly change less the required measure arrives late or exits early. Also, the variable scrolling speed would be extremely burdensome on the musician and impair their ability to play as they chased the music with their eyes.

Due to the above, being able to change the width of the measures is a requirement of the application. The application defaults to displaying three equal width measures spanning the screen. The user may increase or decrease the number of measures. A whole number of measures need to be on the screen at any time to ensure that the notes currently being played are on the screen at the correct moment. When the user changes the number of measures displayed, the score is recreated with the new measure widths.

D. Multiple-Staff Layout and Overlap

If this application had no concerns about making the notes as large as possible, it would just divide the vertical space evenly between the different parts and staffs. Since size is important, screen area needed to be optimized. In music, there must be a significant amount of whitespace above and below the staff lines to accommodate off staff notes. Often, the whitespace is unused thus wasted. To optimize the legibility of the music score, the whitespace areas were allowed to overlap. As shown in Fig. 7, the usage of overlap allows for significantly larger note font. This is desirable in an application geared toward the visually impaired. However, it can lead to note collisions if a note from a lower staff lands in the same place as one from the staff above it. Instead of restricting the note size based on this possibility, the application allows the user to increase the height to the point of a collision if desired. Some users may decide that larger notes are worth one or two collisions.

E. Maximizing Tablet Screen Space

Additional efforts besides overlapping parts were made to maximize the screen space of the tablet. Primary trials of this application were done on a Motorola Xoom with a 10.1 inch diagonal display. It has a resolution of 1280x800, so maximizing vertical and horizontal space was important. First, the signature measure was fixed to the left of the music display area. Signature measures contain the time and key signature, needed so that the user can check which notes should be modified and how many beats are in a measure.

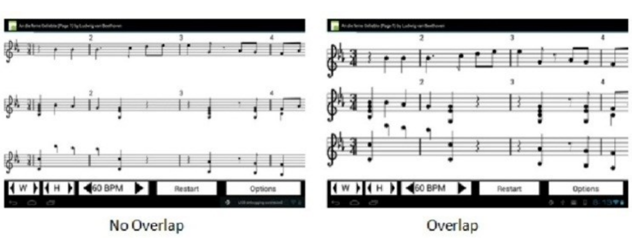

Figure 7. Overlapping White Space

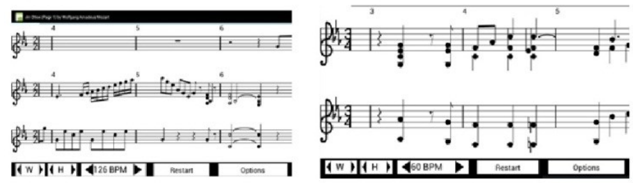

Figure 8. Two Parts Versus One Part

The signature measure was made to be a fraction of the width of a regular measure, so that it took up as little horizontal room as necessary. Music readers look ahead from the measure they are currently playing, so more measures on the screen are desirable. A smaller signature measure accomplishes this.

Often a user plays only one part of the song at a time, rendering the remaining staves unnecessary. Thus, the user can hide the scores of the parts they are not interested in as shown in Fig. 8. This significantly increases the size of the notes.

Horizontal space was also saved by being careful to include a minimum number of buttons on the screen. Width, height, tempo, restart, and a link to the options menu were determined to be the bare minimum of buttons needed on the score page by the informal focus group. This resulted in 85% of the screen height available for displaying the music score. The focus group felt that this was the minimum font for the buttons to be legible.

F. Color Changes

An essential feature of the system is the ability to change the foreground and background colors. This is part of the Score object and used throughout the application. Currently the color set consists for a foreground color for the notes, staves, etc. and a background color. The colors are currently limited to white, black, yellow and blue as these were the colors requested by the focus group of visually impaired musicians. Other color options can be added as requested.

G. Smooth Scrolling

Android's built-in scrolling implementation did not afford sufficient granularity of scrolling or control over its timing. Thus, scrolling the music was accomplished by first creating a bitmap file of the score as discussed previously. This bit map is controlled by the MusicAnimationView object. This object interacts with a running thread that constantly updates a screen sprite. The sprite calculates the position of the score that should be visible on the screen given the elapse time. The current position is the tempo (in pixels per millisecond) times the difference between the current time and the starting time. The bitmap is then sifted to the left accordingly thus updating the visible portion of the score. This method provides a frame rate of approximately 30 frames per second, scrolling the music smoothly in the eyes of the user

H. Evaluation

This player was demonstrated to a small number of visually impaired users and has received promising reviews. A wider study is underway.

V. Conclusions and Future Work

This research defines a framework for use in drawing musical scores based on the MusicXML data format. This application was developed with the ongoing review, testing and feedback from a visually impaired group of musicians with an immediate need for a low cost device that would allow them to continue to read, perform and enjoy music. With the multitude of musical scores available on the web, this device gives them continual access to new music. As their vision changes, the affordances provided should allow them to perform for many years.

Future work will include quantitative and qualitative evaluation with a range of users. The pilot program used a limited group of representative users from the Columbus School for the Blind.

Acknowledgements

This work was partially supported by the National Science Foundation's Industry-University Cooperative Research (Award #0753710) at the Ohio State University and partial by the Tony Wells Foundation.

References

[1] D. B. Elliott, B. Patel and D. Whitaker, "Development of a reading speed test for potential-vision measurements," *Investigative Ophthalmology & Visual Science,* vol. 42, no. 8, pp. 1945-1949, 2001.

[2] T. Guerreiro, J. Oliveira, J. Benedito, H. Nicolau, J. Jorge and D. Gonçalves, "Blind People and Mobile Keypads: Accounting for Individual Differences," in *Human-Computer Interaction – INTERACT 2011*, Lisbon, Portugal, 2011.

[3] S. Chiti and B. Leporini, "Accessibility of Android-Based Mobile Devices: A Prototype to Investigate Interaction with Blind Users," in *Proceedings of the 13th international conference on Computers Helping People with Special Needs - Volume Part II*, Linz, Austria, 2012.

[4] S. K. Kane, C. Jayant, J. O. Wobbrock and R. E. Ladner, "Freedom to roam: a study of mobile device adoption and accessibility for people with visual and motor disabilities," in *Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility*, Pittsburgh, Pennsylvania, USA, 2009.

[5] N. Markus, S. Malik, Z. Juhasz and A. Arató, "Accessibility for the blind on an open-source mobile platform: mobile slate talker (MOST) for android," in *Proceedings of the 13th international conference on Computers Helping People with Special Needs*, Linz, Austria, 2012.

[6] C. Jacobs, W. Li and D. H. Salesin, "Adaptive document layout via manifold content," *Second International Workshop on Web Document Analysis (wda2003),* 2003.

[7] C. Jacobs, W. Li, E. Schrier, D. Bargeron and D. Salesin, "Adaptive document layout," *Communications of the ACM,* vol. 47, no. 8, pp. 60-66, 2004.

[8] T. Guerreiro, "Assessing mobile-wise individual differences in the blind," in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services (MobileHCI '10)*, Lisbon, Portugal, 2010.

[9] M. Connolly, C. Lutteroth and B. Plimmer, "Document resizing for visually impaired students," in *Proceedings of the 22nd Conference of the Human-Computer Interaction Special Interest Group of Australia on Computer-Human Interaction (OZCHI '10)*, Brisbane, Australia, 2010.

[10] D. B. Elliott, B. Patel and D. Whitaker, "Development of a reading speed test for potential-vision measurements," *Investigative Ophthalmology & Visual Science,* vol. 42, no. 8, pp. 1945-1949, 2001.