Audio Aids in Source Code

Dominic Roberts
Karlton Weaver

Rochester Institute of Technology Rochester, New York

ABSTRACT

Source code makes use of a number of visual cues to convey information to the reader. Things such as capitalization, multicolor highlighting, and white space all have tremendous value when navigating code. These cues are not accessible to the visually impaired however, and most blind or visually impaired programmers have had to make do with simple text to speech software translations of the files they wish to read. This is not ideal, as it is difficult to ascertain certain things (present level of nesting for example) about a large source file without these visual delimiters. This research seeks to determine how established auditory cues can be used to augment the default output of text to speech software to aid the blind and visually impaired with navigation and orientation within source files. Our preliminary findings suggest that spearcons and white noise could be effective tools for indicating and distinguishing between discrete blocks of code.

KEYWORDS

accessibility, visually impaired, blind, auditory cues, spearcons, programs, source code, synthetic speech

1. INTRODUCTION

There are many coding conventions concerning readability which make source code easier to navigate and maintain. Many of these conventions are reliant on visual cues, such as specific use of capitalization or white space. A great deal of these visual cues do not translate to the spoken representations of text on which the blind and visually impaired rely.

There is very little research involving accessibility as it relates to source code for the visually impaired in the literature. There is, however, a more robust repository of similar research in other areas (particularly in mathematics and software interfaces). In this research, we examined how audio cues used in these areas could be used to enhance visually impaired individuals' comprehension of code. We were particularly interested in denoting distinct blocks of code (for example functions and control structures), and the levels of nesting within these blocks.

We developed a number of audio files which altered the default output of the open source screen reader NVDA, and then presented these files to two blind users in survey format for evaluation. We used their feedback to re-evaluate the files, resulting in three iterations of increasingly refined audio. This paper details the design processes used to generate the files, and the results obtained from the surveys.

2. BACKGROUND

This project seeks to aid the blind and visually impaired with the problem of navigation and orientation within source code. While this specific problem has been scarcely addressed previously (though some attempts have been made **[2]**), there is a great deal of existing research concerning accessibility. We were thus able to identify the most commonly used auditory cues and techniques used to enhance accessibility.

2.1 Common Auditory Cues

In general, there are four audio mechanisms used to enhance text-to-speech representations of data. They are: auditory icons, prosody, earcons, and spearcons.

2.1.1 *Auditory Icons*

Auditory icons are audio representations which directly reflect what they represent (for example a creaking sound to represent a door icon). The benefit of auditory icons is that they are intuitive and simple to learn if designed properly. However, they tend to be time consuming to design, and it is often impossible to find meaningful mappings for concepts and ideas represented via text (what "sound" is copy? or save?).

2.1.2 *Prosody*

Prosody is the use of changing voice patterns (pitch, volume, rate of speech) to convey information. Several studies have reported that it can be extremely effective when used properly. One such study found that the insertion of pauses greatly improved comprehension of spoken algebraic equations **[2]**.

2.1.3 *Earcons*

Earcons are arbitrary sounds (often musical) which are mapped to some idea or concept that they are meant to represent. An example would be using a tone or beep to represent brackets. The advantage of earcons is that they are incredibly simple to design. Their weakness is that they must be learnt by the user, and excessive use of them can result in extreme cognitive strain on the user as they try to recall the meanings of all the different cues. There is research which suggests both earcons and auditory icons are actually counter-productive in their attempts to improve user performance and accuracy **[1]**, however we believed that they warranted testing.

2.1.4 *Spearcons*

Not to be confused with earcons, are spearcons. These are words, or parts of words, which are sped up significantly to produce an audio cue. Spearcons have several advantages in that they are simple to make, intrinsically unique (except in the case of homonyms), and more intuitive than earcons. Along with prosody, these seem to be the most effective cues for enhancing user performance and accuracy **[1][2][3]**.

2.2 Related Work

A good deal of research in accessibility pertains to the development and use of audio cues to convey information in various contexts (mostly math and different software interfaces). Many of the techniques described in these projects may be leveraged in our own.

2.2.1 *In Mathematics*

In mathematics, the enhancement of text to speech software to convey additional information has been a topic of interest for some time. The problem driving this research is identical in nature to our own: written mathematical expressions convey information to the reader which is not adequately communicated by spoken representations of those expressions. Indeed, much of the ambiguity of spoken mathematics arises from the listener's inability to identify discrete blocks within the spoken expressions, which mirrors one of our primary concerns regarding code. Though the "blocks" of code with which we will be dealing are significantly larger than the blocks within mathematical expressions, we can still learn a great deal from the techniques used to denote those blocks.

It has been noted that the the two most common approaches to disambiguating spoken mathematic equations are the use of prosodic and lexical cues **[1]**. Regarding the former, two studies have been conducted which suggest that the use of pauses has tremendous value in clarifying spoken mathematical expressions **[1],[5]**. A third study made use of non-verbal lexical cues (notably earcons and spearcons) and found that spearcons dramatically improved comprehension **[12]**. This study did note however, that a problem with comprehension arose at the level of speed necessitated by spearcons (for example with "subscript" and "superscript").

2.2.2 *In Software Interfaces*

There have been two research papers of note in the field of software interfaces, the first and more relevant of which concerns the design and effectiveness of a sonified debugger called SODBeans **[2]**. SODbeans is a "NetBeans module suite designed to enhance accessibility for the blind in modern programming environments". One of the design decisions they made which had a great impact was using what they called "semantic as opposed to syntactic cues". The example they gave in the paperwas:

> Whereas pressing the arrow keys produces semantic cues in SOD, it produces syntactic cues in Visual Studio + JAWS# For example, when visiting the statement m[3] = 14; a SOD user would hear "m sub 3 equals 14." In contrast, a Visual Studio + JAWS user would hear "m left bracket three right bracket equals fourteen semicolon."

Overall it was found that the use of SOD audio cues resulted in greater comprehension than Visual Studio+JAWS audio cues.

Additionally, Pavani Yalla et al **[11]** conducted research into the design of auditory scrollbars which found that the use of a short alphabetic spearcons increased users' ability to scan large menus effectively.

3. METHODOLOGY

In order test the merit of various cues, we first generated samples of these cues being used with code snippets and then presented these samples to a blind expert programmer (and subsequently a blind user with little programming experience) whose feedback we used to refine the samples.

3.1 Audio Creation and Examples

Audacity (a free audio editor) was used to generate multiple audio files, which were then presented to a blind expert programmer to determine the merits of each. A total of seven audio files were created, sixof which uniquely made use of tones, supplementary text, semantic cues , and white noise to enhance the screen reader, and one of which was simply the default behavior of the screen reader and provided a baseline. After three iterations of surveys, improvements were made to each of the audio files. The improvements to the files hail from the responses and feedback that we received.

3.1.1 *File Creation*

The audio files were made to represent a snippet of java code comprising an if statement within a for loop (see figure 1). The screen reader was then used to read this snippet and its output was recorded. Finally, the various audio cues were recorded, and then merged with the output of the screen reader to create the individual audio files. The one exception was the example wherein semantic cues were used; here the code snippet was modified to include the semantic cues (for example "==" was replaced with "is equivalent to").

```
for( int i = 0; i <= 10; i++ )
{
    system.out.println( x );
    if( i == 5 )
    {
        system.out.println( "Half Way");
    }
}
```

Figure 1

3.1.2 *Audio File 1- Baseline*

This audio file was simply a recording of the default behavior of the screen reader. The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/example2_1_Default.mp3

3.1.3 *Audio File 2- Tones + Semantic Cues*

In this audio file, short tones of various pitches were played just before the opening and closing braces. This approach used four distinct pitches to correspond to the four braces in the code; two each for both levels of nesting. In addition, a twang was placed in the middle of the string literal to denote it as quoted text. Finally, "==" was replaced with "is equivalent to", and "<=" with "less than or equal to". The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/Tones2-1.mp3.
At the end of three survey trials we edited this tone file to make the pitches match at the start and end of each nested level. The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/ok.mp3

3.1.4 *Audio File 3 – Tones Version 2*

This audio file utilized tones to correspond to the different braces. The tones were played on top of the speech for the braces rather than before it, and only two tones were used; one for each set of braces. The complete audio file can be found here: http://www.se.rit.edu/~sal/audiosurvey/example2_1_Tones.mp3.

3.1.5 *Audio File 4 – Supplementary Text*

In this audio file, supplementary text (such as "start for") was added before the opening and closing braces. The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/Spearcons2-1.mp3

3.1.6 *Audio File 5 – Supplementary Text Version 2*

This audio file made also used supplementary text, but placed it almost on top of the speech for each brace, near the end of the "brace". The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/example2_1_Spearcon.mp3 At the end of three survey trials we found that playing the supplementary text before the brace allowed the user to anticipate what was happening. This idea was well liked and supported by the users who tested the audio files. The complete file can be found here:

http://www.se.rit.edu/~sal/audiosurvey/ok2.mp3

3.1.7 *Audio File 6 – White Noise*

In this audio file, white noise was played on top of the speech at different levels of amplitude. The intention was to symbolize the level of nesting using white noise, with different amplitudes symbolizing different levels of indentation. This file was created as a another option to depict different levels of nesting. The complete audio file can be found here: http://www.se.rit.edu/~sal/audiosurvey/example6_1_WhitenoiseV1.wav

3.1.8 *Audio File 7 – White Noise Version 2*

In this audio file the white noise was played on top of the speech at different levels of amplitude. The intention was to symbolize the level of nesting using white noise, with different amplitudes symbolizing different blocks. Note that audio file 6 changes amplitude with indentation, whereas this file changes amplitude as the reader enters and leaves different logical blocks of code. The complete file can be found here: http://www.se.rit.edu/~sal/audiosurvey/example2_1_WhitenoiseV2.wav

3.2 Survey Creation

3.2.1 *Setup*

In order to create the survey we utilized the survey tool on http://www.surveygizmo.com/. We registered for a student account which was a free account but had limited features. Based on the amount of questions we had, we associated each audio file with two questions short answer style. In order to place the audio file on the survey, we utilized HTML coding to embed the file on the survey. Here is an example:

```
<a href = 'http://www.se.rit.edu/~sal/audiosurvey/ok.mp3'>Audio file 1</a>
<br /><br />
```

Only changing the link to the audio file and the Audio file # (both highlighted) let us do the same thing for all of the audio files.

3.2.2 *Questions*

Both of these questions were asked with each audio file given.

Question 1: Please listen to the audio file and describe what you hear.

This question was posed in order to get raw feedback from the user. We wanted to know what significant traits were in the audio example; what did they understand it to mean? how did this example improve on the default behaviour, if at all?

Question 2: Please describe what issues, if any, you noted about the audio quality.

This question was simply posed to ask the user about the audio file's sound quality. Was it clear? Was it hard to tell what was being said? Was the volume to high or to low? Also were the cues used able to be heard?

3.2.3 *Survey Links*

- http://edu.surveygizmo.com/s3/585959/NVDA-Code-Reading-Variations-Test
- http://edu.surveygizmo.com/s3/601999/NVDA-Code-Reading-Variations-Test-2

4. DISCUSSION OF RESULTS

The purpose of creating these surveys was to get feedback from those who are in the field of study with usability. We took the answers and criticism so that we could enhance the audio files output to increase usability and understanding.

With the completion of the analyzing the first survey, Dr Fitzpatrick gave very helpful feed back not only to enhance the audio quality but enhance the way of using cues in the files. At first we wanted to specialize a certain pitch for the tones on the start and end of all nested levels; however; he suggested that we match the pitches so that one can put together what nested level is starting and ending. He also liked that we put the cues on top of the speech instead of stopping and starting. Also with the spearcon files, he didn't necessarily agree with them being spearcons; he just considered them alternate speech. So we listened to what spearcons actually sounded like and adjusted the rate at which they were being heard. We adjusted the tones file so that the start and end of the different nested levels matched, and we kept the cues on top of the speech.

After round two of editing the survey, Dr. Fitzpatrick gave just a little feedback from the improved audio files. Regarding the tones and matching pitches, the only problem that seemed to occur was the pitch of the tones not being loud enough to hear over the voice. The problem was corrected by lowering the volume of the voice.. In light of the spearcons, he liked the file in which the spearcons came before the braces; he said it gives the user time to anticipate what is going to happen. As a new idea we tried to implement white noise in the background of the speech in order to indicate nesting. That did not really fit so well with him as he thought it was an error until asking him a question about the white noise. In summary, we received positive feedback with minor twitches to be made in the audio files.

With the completion of a last and alternate survey, the overall response from the user was good. The user explained correctly every time what was going on in each audio file; however, we don't believe that they realized that the tones played for the start and end of the nested levels matched (same issue with the white noise). They were not familiar with spearcons, but they did seem to like them. On the other hand they did suggest one interesting thing. Combining the spearcons along with the white noise would be very helpful. Something new to think about.

5. CONCLUSION AND FUTURE WORK

This research has verified the effectiveness of the use of spearcons to enhance comprehension of source code by visually impaired individuals. It has further suggested the direction that should be taken in the future to further refine the output of a custom screen-reader.

While this project focused on proof of concept by generating audio files and testing their effectiveness, future research will seek to enhance the dynamic output of the screen reader itself by implementing the audio cues presented here as a part of the screen reader. Additionally, the audio cues themselves may be further refined, and a larger test group used to optimize the experience of visually impaired users.

CITATIONS

1. Ed Gellenbeck , Andreas Stefik, Evaluating prosodic cues as a means to disambiguate algebraic expressions: an empirical study, Proceedings of the 11th international ACM SIGACCESS conference on Computers and accessibility, October 25-28, 2009, Pittsburgh, Pennsylvania, USA

2. A. Stefik, C. Hundhausen, R. Patterson. An Empirical Investigation into the Design of Auditory Cues to Enhance Computer Program Comprehension. The International Journal of Human-Computer Studies. Submitted 2010. Accepted 2011.

3. Elizabeth D. Mynatt, Transforming graphical interfaces into auditory interfaces for blind users, Human-Computer Interaction, v.12 n.1, p.7-45, March 1997

4. Robert D. Stevens , Alistair D. N. Edwards, An approach to the evaluation of assistive technology, Proceedings of the second annual ACM conference on Assistive technologies, p.64-71, April 11-12, 1996, Vancouver, British Columbia, Canada

5. Fitzpatrick, D. 2006. Mathematics: how and what to speak. In Proceedings of ICCHP, International Conference on Computers Helping People with Special Needs, Springer Verlag, pp. 1199--1206 .

6. Moos, A. and Trouvain, J. 2007. Comprehension of ultra-fast speech - blind vs. "normally hearing" persons. Proceedings of the 16th International Congress of Phonetic Sciences, pages 677—680.

7. Brewster, S. A., Wright, P. C. & Edwards, A. D. N. 1995. Experimentally derived guidelines for the creation of earcons. Adjunct Proceedings of HCI'95, Huddersfield, UK.

8. Edwards, A. D. N. 1997. Using sounds to convey complex information. in A. Schick and M. Klatte (ed.), Contributions to Psychological Acoustics: Results of the Seventh Oldenburg Symposium on Psychological Acoustics, Oldenburg, pp. 341—358.

9. Walker, B., Nance, A. and Lindsay, J.: Spearcons. 2006. Speech-based Earcons Improve Navigation Performance in Auditory Menus. In Proceedings of the 12th International Conference on Auditory Display, London, UK, June 20-23.

10. Stevens, R. D., Principles for the Design of Auditory Interfaces to Present Complex Information to Blind Computer Users. 1996. PhD Thesis, University of York, UK

11. Pavani Yalla , Bruce N. Walker, Advanced auditory menus: design and evaluation of auditory scroll bars, Proceedings of the 10th international ACM SIGACCESS conference on Computers and accessibility, October 13-15, 2008, Halifax, Nova Scotia, Canada[;pl

12. Emma Murphy, Enda Bates, Donal Fitzpatrick. 2010. Designing auditory cues to enhance spoken mathematics for visually impaired users. In *Proceedings of the 12th international ACM SIGACCESS conference on Computers and accessibility* (ASSETS '10). ACM, New York, NY, USA, 75-82.