

Summer 2010

Improving Accessibility for the Blind on the Android Platform

Nicole Torcolini

Abstract

Over the last few years, touch screens have become more prevalent not only in household appliances but also in smartphones, PDAs, and computers. These devices, particularly smartphones, have many features and applications at a reasonable price that would be useful to the blind; however, the touch screens on early versions of such devices rendered them unusable by the blind. Touch screens, without any supplemental software or hardware, are inaccessible to the blind because they do not provide verbal output to convey where controls are located on the screen or what control the user has selected. Those touch screens that do have verbal feedback often do not allow the user to explore the screen without activating any of the controls. The Talking Tap Twice Technique addresses this problem on the Android smartphone by providing a self-voicing interface upon which programmers can build their applications. The Talking Tap Twice also defines an input method which allows the user to explore the screen and allows the programmer to control the exact output.

1. Introduction

Over the last few years, touch screens have become more prevalent not only in household appliances but also in smartphones, PDAs, and computers. These devices, particularly smartphones, have many features and applications at a reasonable price that would be useful to the blind, such as GPS, as well as having the capability of developing applications specifically for the blind, such as a color namer, a barcode scanner, or an OCR application. One important feature is that the user can install applications, not just use those that come with the device. However, the touch screens on early versions of such devices rendered them unusable by the blind. Touch screens, without any supplemental software or hardware, are inaccessible to the blind because they do not provide verbal output to convey where controls are located on the screen or what control the user has selected. Those touch screens that do have verbal feedback often do not allow the user to explore the screen without activating any of the controls.

The Talking Tap Twice addresses this problem on the Android smartphone by providing a self voicing interface upon which programmers can build their applications. The Talking Tap Twice also defines an input method which allows the user to explore the screen and allows the programmer to control the exact output. The Talking Tap Twice speaks the label for a control when it is tapped. However, it does not activate a control until the user taps twice on any of the controls, at which the last control that was tapped once is activated. The Talking Tap Twice only

works with applications that were developed using its interface; currently, these are the Android Talking Calculator, Android Talking Level, and Talking Tap Twice Demo.

2. Background

a. Touch Screen Accessibility in General

There are already several systems that work to improve the accessibility of touch screen devices. Two such systems that focus on addressing touch screen accessibility are Vanderheiden's Talking Fingertip Technique (Vanderheiden 2010) and Kane's, Bigham's, and Wobbrock's Slide Rule (Kane et alia 2008). The Talking Fingertip Technique was a touch screen that spoke the names of the controls as the user scanned the screen with his or her finger. When the desired control was located, the user activated it by pressing a physical button at the bottom of the screen. While this technique made the screen accessible, it had the problem that it could only work with touch screens that had a separate activation button. In contrast, Kane et alia's Slide Rule could theoretically be made to work with any touch screen that had the software to support it, because it does not require a physical activation button. The Slide Rule arranges controls in a specific way, provides an accessible input method, and provides verbal feedback. In the Slide Rule, controls are arranged in a vertical list, which the user scans and hears spoken by running his or her finger down the screen. The user activates the selected control by tapping anywhere on the screen with a second finger. The Slide Rule touch screen also defines a flicking gesture and an L-shaped gesture for other functions.

b. Other Smartphone Touch Screen Accessibility Systems

A few accessibility systems for touch screens have been designed for smart phones. Apple's VoiceOver (Apple 2010) is a screen reader for Apple devices, including their touch screen devices. VoiceOver takes two separate approaches of, first, leaving the screen in its original arrangement, or, second, leaving the screen as is but allowing the user to scan through items as though they were in a list by using a flick gesture to move to the next or previous item. VoiceOver also speaks the selected control as the user scans the screen, but it does not activate it. To activate a control, the user removes his or her finger from the desired control and then taps twice rapidly anywhere on the screen. VoiceOver also defines several other gestures, some of which are multi-finger, to allow navigation of the screen and adjustment of settings. Another screen reader for smart phones is Mobile Speak Pocket (Code Factory 2010), which takes a rather different approach. Mobile Speak Pocket divides the screen into four quadrants, which the user taps and sometimes holds to execute different commands. However, this requires the user to memorize commands which may not be intuitive. Mobile Speak pocket also allows flicking instead of using the navigation keys in some cases.

c. Accessibility on the Android Smartphone

Like other devices of its kind, when the Android Smartphone was released, it was inaccessible to the blind (except for possibly keyboard input on models with keyboards). In Android version 1.6, Google added TalkBack, SoundBack, and KickBack, which provide spoken, sound, and haptic feedback, respectively. However, all of these applications have to be installed and

activated. In addition, TalkBack has the problem of mis-pronouncing some words and names, particularly those that are not pronounced phonetically. Although this is a problem with all speech synthesizers, some synthesizers use a pronunciation dictionary, where the correct spelling of a word or phrase is associated with a spelling that is such that the synthesizer will pronounce the phrase correctly. TalkBack does not have this feature.

The Talking Tap Twice addresses these issues, not requiring that any additional software (besides the application that extends the Talking Tap Twice) be installed or activated. The Talking Tap Twice also has a pronunciation dictionary. Furthermore, the Talking Tap Twice does not require the use of the keyboard; some applications on the Android are only accessible if the keyboard navigation keys are used, as the Android does not provide a means of exploration without activation for these applications. Google also added the EyesFree Shell for the home screen. When using the EyesFree Shell, wherever the user touches the screen is set to home or center. The user can then slide around the screen to hear the available options until the desired option is located. However, EyesFree Shell has one main problem. Unless the user releases back at the original point of home, some item is selected. This requires remembering exactly where that point is or where it is located in comparison to other options.

d. Adding Accessibility Independent of System Features

Programmers of touch screen devices can also add accessibility to their applications without relying on system features or overlying systems. For example, the Android application programming interface (API) provides a TextToSpeech class that allows the programmer to integrate speech output into his or her application. However, using such resources does force the developer to give certain feedback. The Talking Tap Twice provides guidelines and easy implementation for giving accessible feedback.

3. Design Principles

The Talking Tap Twice was designed based on the following principles and objectives:

1. **Built in:** The accessibility features of a program should be built in and, if the program is specifically for blind users, should not have to be activated or rely on a system feature being activated or installed.
2. **Modification:** The device should not have to be modified or have any additional hardware added to it in order to work with the accessibility system.
3. **Exploration without activation:** The user should be able to explore the screen without activating any of the controls.
4. **Ease of use:** The user should not have to learn any new gestures or commands or worry about accuracy.
5. **Audio feedback:** The user should receive audio feedback that adequately describes what action had occurred.
6. **Intuitiveness:** Different controls should respond to different actions in the way that makes most sense to a typical user.

7. Understandability: The user should not have to adjust to phrases being mis-pronounced.
8. Versatility: Accessibility features should not render a system frustrating or hard to use by a sighted person.
9. Customization: The developer should be able to control how the accessibility system interacts with the application.

4. Design

The Talking Tap Twice was designed specifically for the Android smart phone, with the intension of being put into use shortly after its development. The Talking Tap Twice was written in Java using the Android API. It is an abstract class that extends the Java class, Activity, which developers will extend in order to include its functionalities in their applications. The Talking Tap Twice uses the TextToSpeech class of the Android API to provide verbal feedback. The Talking Tap Twice uses XML attributes of custom classes that extend the standard widgets to allow the developer to pass information about each widget, such as what should be spoken when the widget is tapped, what should be spoken when the widget is selected, and what action should be performed when the widget is selected. The Talking Tap Twice also has a pronunciation dictionary to allow for the correction of pronunciations. If the developer knows that the TextToSpeech will mis-pronounce something or wishes to have an abbreviation displayed but the expansion spoken, he or she can enter a set of values into the pronunciation dictionary, the first being the phrase that the TextToSpeech will receive and the second being the way in which it should be pronounced.

5. User Interface

The Talking Tap Twice does not designate how controls should be laid out. This design feature is intended so that a sighted user does not become frustrated by a layout that is accessible to a blind user, but visually confusing or annoying. However, the developer should still consider characteristics, such as size and spacing, that affect usability by the blind.

In spite of this, the Talking Tap Twice allows the user to interact with the interface in an intuitive way. A user can tap anywhere on the screen to hear the label for a control, which has been specified by the developer, without activating any of the controls. Once the user finds the desired control, he or she can tap twice on any of the controls to activate the last selection. As well as providing spoken feedback, the Talking Tap Twice also plays a clicking sound and vibrates when the screen is tapped.

The Talking Tap Twice does not use flicking because some users have trouble mastering the flicking motion. Also, the input method was designed to be as close to that which a sighted user would use, except that it is accessible.

6. Future Work

- Currently, the Talking Tap Twice only has Buttons, CheckBoxes, and TextView widgets that are designed to be used with it; that is, these are the only widgets which support the

- There is the possibility that the Talking Tap Twice could be designed to be available anywhere on the Android. At this time, the Talking Tap Twice is only available in those applications which include it.
- The Talking Tap Twice could be designed to respond to actions initiated by the Android itself or indirectly by the user. Currently, the Talking Tap Twice only responds to actions initiated by the user. For example, if pressing a button causes the text to change, the Talking Tap Twice will speak the phrase associated with pressing that button, but it will not read the text that has been changed unless that was included as part of the action to be performed when the button was pressed.
- The pronunciation dictionary of the Talking Tap Twice could be expanded to allow the user to input entries.

7. Conclusion

The Talking Tap Twice is a self voicing interface for the Android smartphone upon which programmers can build their applications. The Talking Tap Twice defines an input method which allows the user to explore the screen and allows the programmer to control the exact output. Hopefully, it will help developers add accessibility to their applications in a way that is straight forward for the developer and intuitive for the user.

8. Acknowledgements

I wish to thank Dr. Richard E. Ladner, Boeing Professor in Computer Science and Engineering, Department of Computer Science & Engineering, University of Washington; Shaun K Kane, PhD Candidate, The Information School, University of Washington; and Chandrika Jayant, PhD Candidate, Department of Computer Science & Engineering, University of Washington for their assistance and advice regarding accessibility and adaptation of Talking Tap Twice.

9. References

"Accessibility - iPhone - Vision." Apple. Visited 11 Sep 2010

<<http://www.apple.com/accessibility/iphone/vision.html>>.

"Android Version Guide - Android Accessibility." Eyes-free - Project Hosting on Google Code.

Visited 11 Sep 2010 <[\[free.googlecode.com/svn/trunk/documentation/android_access/versions.html\]\(http://eyes-free.googlecode.com/svn/trunk/documentation/android_access/versions.html\)>.](http://eyes-</p>
</div>
<div data-bbox=)

Kane, Shaun K., Jeffrey P. Bigham, and Jacob O. Wobbrock. Slide Rule: Making Mobile Touch Screens Accessible to Blind People Using Multi-Touch Interaction Techniques.

Rep. 2008. University of Washington. Visited 1 Sep 2010

<<http://students.washington.edu/skane/pubs/assets08.pdf>>.

Millsap, Chris. "SeroTalk Tech Chat 68 – Accessibility and Usability of Android Phones | SeroTalk." Interview by Michael Lauf and Joe Steinkamp. SeroTalk | A podcast and interactive blog on the accessible digital lifestyle, produced by Serotek, the Accessibility Anywhere people. 10 Sep 2010. Visited 11 Sep 2010

<<http://serotalk.com/2010/09/10/serotalk-tech-chat-68-accessibility-and-usability-of-android-phones/>>.

"Mobile Speak and Mobile Magnifier for Windows Mobile Phones." Code Factory: Making mobile phones and PDAs accessible to the blind and visually impaired. Code Factory, S.L. Visited 11 Sep 2010

<http://www.codefactory.es/descargas/family_4/ms4_userguide_wm.html#_Toc254946119>.

"TalkBack: An Open Source Screenreader For Android." Google Open Source Blog. 20 Oct 2009. Visited 11 Sep 2010 <<http://google-opensource.blogspot.com/2009/10/talkback-open-source-screenreader-for.html>>.

Vanderheiden, Gregg C. Use of audio-haptic interface techniques to allow non-visual access to touch screen appliances. Rep. Trace Research and Development Center - Trace Center, University of Wisconsin-Madison. Visited 11 Sep 2010

<http://trace.wisc.edu/docs/touchscreen/chi_conf.htm>.