

Analyzing App Inventor for Building Usable Touch Screen Courseware for Blind Users

Mohammed Fakrudeen¹, Sufian Yousef¹ and Abdelrahman H. Hussein²

¹Dept. of Computing and Technology, Anglia Ruskin University, Chelmsford, United Kingdom
mohammed.abdul@student.anglia.ac.uk, sufian.yousef@anglia.ac.uk

²College of Computer Sciences & Software Eng., University of Ha'il, Ha'il, Kingdom of Saudi Arabia
ar.hussein@uoh.edu.sa

Abstract- This paper focuses on how to design and develop a usable Touch Screen Mobile Courseware Application (TMCA) by using usability features recommended in the literature. For our study, we consider usability features that are imperative for blind users such as widget usability, validation, feedback and navigation. We explain the usability features needed to address the relevant usability issues in design and development of TMCA for blind users. We also suggest the appropriate location for control items or widgets in a flat screen. Furthermore, we make recommendations for ways to satisfy usability features through existing features. The study shows that prospective educators cum developers can implement usability features as a base for addressing usability for blind users to a significant extent. Although the usability features are guided by the literature, said features are simply defined in the literature. There is no guide for designers and developers demonstrating how exactly to achieve such features. The proposed approach in this paper makes the TMCA more usable for blind users and it is more likely to achieve its objectives.

I. INTRODUCTION

With more than five billion mobile devices in service worldwide, and exponential growth of mobile applications in all domains of daily life, mobile phone accessibility and assistive applications for persons with disabilities are at the forefront of the agenda of all countries that have signed onto the United Nations Convention on the Rights of Persons with Disabilities. In the United States, 54 million people live with disabilities, while that number reaches more than 650 million worldwide [1].

App Inventor is a free, open source application that permits people with any level of programming background to create software applications for the Android operating system. App Inventor uses a graphical user interface that allows users to drag and drop blocks (puzzle-shaped objects) to build applications without ever having to write traditional code.

App Inventor is currently being used in a variety of educational settings, including classrooms ranging from elementary school to college and after-school programs. While many programs teach computer science using App Inventor, there are also a number of educators using it as a tool to engage students while teaching other subjects. Many after school programs are being developed around App Inventor. These programs often focus on encouraging groups that are typically underrepresented in the field of technology to feel empowered to be creators of technology.

W3C discusses creating accessible apps on websites and in mobiles. However, it does not focus on mobile application development in touch screen. The development of applications in smart phones (mobile) with physical buttons varies with application development through touch screen. Website design and mobile applications using physical buttons are similar in many aspects. Website accessibility suits mobile application development through the same principles as suggested by W3C. Thus the developer is only concerned with the layout size and content that fits into the smart phone layout.

In touch screen, it is difficult for visually impaired users to identify widget location on a flat surface. Therefore, WCAG principles cannot be implemented precisely for TMCA development. However, it is imperative to have a user centered and inclusive design from the beginning of TMCA development. As a result, our study primarily focuses on how to implement WCAG 2.0 guidelines on touch screen mobile apps. We note that we have avoided using MWBP 1.0 for implementing in touch screen mobile apps as it is currently in the infant stage.

II. RESEARCH METHODOLOGY

The Software Engineering and Human Computer Interaction (HCI) disciplines consider usability as a nonfunctional requirement. Usability literature has provided extensive guidelines to build usable applications for prospective educators cum developers. Despite agreement between various works as to the nature of usability, each author has named the respective guidelines differently. Additionally, the exact nature of the guidelines also differs between works. In order to categorize a list of usability features, usability features with relevant benefits and strong design implications were identified. Based on the literature, we identified *widget usability, validation, feedback* and *navigation* as usability features with the required characteristics.

For each usability feature, we explain what is meant for App Inventor code segment and which App Inventor code blocks are available for Android mobile application developers to address the relevant usability issues. We also recommend ways to satisfy the usability features through Scratch blocks. Certain usability features are not addressed due to page constraints. Our study is based on App Inventor classic version of 30-Jun-2009.

III. RESULT AND DISCUSSION

The overall level of usability features can be implemented by designing courseware through the App Inventor tool. We have summarized our findings based on techniques that are feasible for touch screen smart phone applications.

A. *Widget Accessibility*

VoiceOver is a gesture based screen reader on the Mac and it is available in iPhone 4S, iPhone 4 and iPhone 3GS. It enables speaking into a touch screen to perform commands such as sending text messages or emails, getting directions, and listening to music [2]. In Android phones, Google's voice action was built for voice to text to issue commands through voice. However, relevant literature is not available regarding the accuracy of commands when issued in different environments such as in busy places, in bad weather and so forth.

Voice-based commands are primarily suited for invoking applications. They cannot be applied to invoke controls such as radio buttons or check boxes. In such cases, almost all input controls are not accessible for blind users. Input control such as the button are accessible for blind users if they are able to identify the position of the button. Input through textbox using virtual keyboard is also not accessible for blind users.

An alternative method can be adopted to use input control as a replacement for traditional methods. For instance, the use of radio button and check box controls can be replaced by simple command buttons for different options. For every toggling, the audio can inform the user what option the user should select. Another alternative is simply through textbox using voice synthesizer enabled.

Assuming that the user has a lot of information from which to choose, the drop down item (list picker in App Inventor) is used in the traditional method. However, the list picker is not accessible for blind users in touch screen. It is difficult for a blind user to identify the location of the item. In order to overcome this, we adopted the flung technique to iterate through the items using canvas widget. The aforementioned technique has been implemented by the first author of this work for e-assessment tools for blind users in TMCA [3].

Another important interaction control is the textbox control where a blind user has to enter the text using a virtual keyboard. Since the button sizes are small with difficult to identify locations for each keyboard button, blind users are not able to use the virtual keyboard. We developed a keyboard interface that adopts the finger technique used by the first author in a phone dialer application [4]. Using this technique, it is easy for a blind user to identify the keyboard buttons.

Findings 1

1. Toggle buttons such as radio button and check box control can be replaced by a simple button.
2. List picker control is replaced by iterating items through flung event of the canvas control.
3. Keyboard interface can be developed using finger technique to replace virtual keyboards.

B. Validation

To check for plausibility of user input, a procedure with result code block is used which returns 'true' in the case that everything is correct or 'false' in the case that there is an error. The choose code block can be used in nested form to keep the source code as small as possible. Fig. 1 demonstrates the validation performed for registration page used in the courseware. It validates null entry in student name and grade entry when the submit button is clicked.

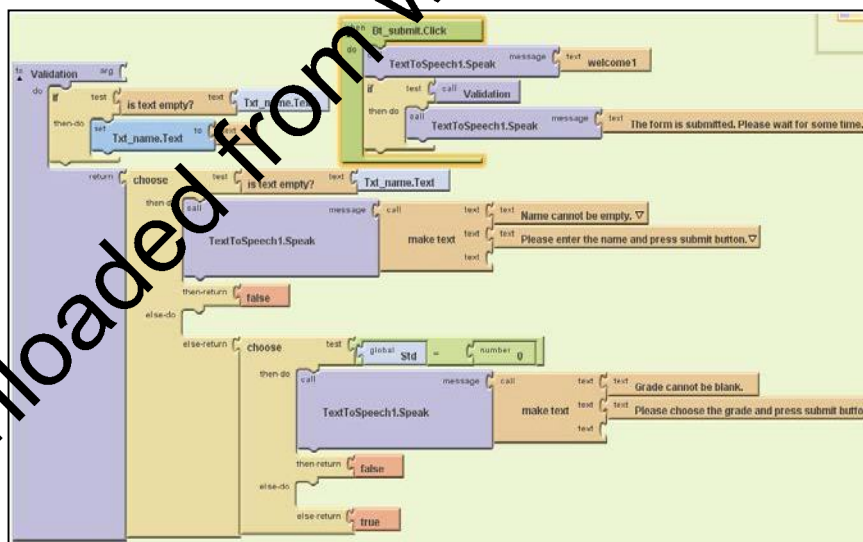


Figure 1. Code block for validation.

Findings 2

1. A procedure with result code block is used for calling validation procedure.
2. Choose code block is used to validate the item.
3. The appropriate message has to be delivered using text-to-speech to guide the blind user to correct mistakes.

C Feedback

Feedback is a technique used to inform the user about what is going on inside the system. The HCI literature proposes four types of feedback: 1) Interaction feedback – to inform the user that the system has heard their request; 2) Progress feedback – to inform the user about tasks that will take time to finish; 3) System Status – to inform the user about any changes in the system status; and 4) Warning – to warn the user about irreversible actions.

Using App Inventor, interaction feedback is provided by using a player component which vibrates to indicate to the user that an action has been performed. Other feedback is provided through a text-to-speech component that informs with verbal messages. To differentiate the warning message with other feedback, the warning message is provided both through vibration and aurally.

Feedback provokes the blind user to perform the next course of action. In a physical keyboard attached device such as a desktop computer or in a laptop, the blind user presses the tab key to navigate through different controls or links. Then the user presses the enter key or another key to open the link or application. In comparison to sighted users this matter is not of much significance because a sighted user is able to click the control or widget on the touch screen by using vision. However, it is difficult for the blind user to interact with widgets directly in keyboard attached devices. To address this issue, when a blind user touches the widget, it provides feedback about the purpose of the widget. Then, the blind user will decide whether to press using a long click to open the application or using a normal click to navigate through to the next item.

Findings 3

1. Feedback is provided verbally through text-to-speech component and physically through player component.
2. Audio feedback can be provided when a blind user presses the button providing information about the application.
3. After initial audio feedback, the blind user should double tap (as on iPhone) or long click (as on Android) to be able to open the application.

D Navigation

When a sighted user uses a mobile application, the page is quickly scanned and the user is able to read for the desired information. However, the blind user has to depend on assistive technologies to read information for them. The blind user requires extreme patience to reach the required page. It requires the blind user to navigate from page to page or section to section until the required information is identified. As a consequence, navigation is classified as page-to-page navigation and in-page navigation.

Page-to-page navigation

The keyboard is the point of interaction for performing any functions using assistive technology. With the advancement of touch screen, all the functionalities such as navigation, opening an app and invoking an event are performed through gestures. App Inventor supports gestures in canvas such as flung (quick swipe), touched, touch-down and touch-up events. While flung event is used to navigate through pages, touch events are generally used to open an app or to pause a process. However, the touch event of a button such as Click or Long Click can also be handled to open a new page.

The *open another screen* code block is used to open a new page without passing information to the new page (Fig.

2). To pass the information to the new page, *open another screen with value* code block is used.

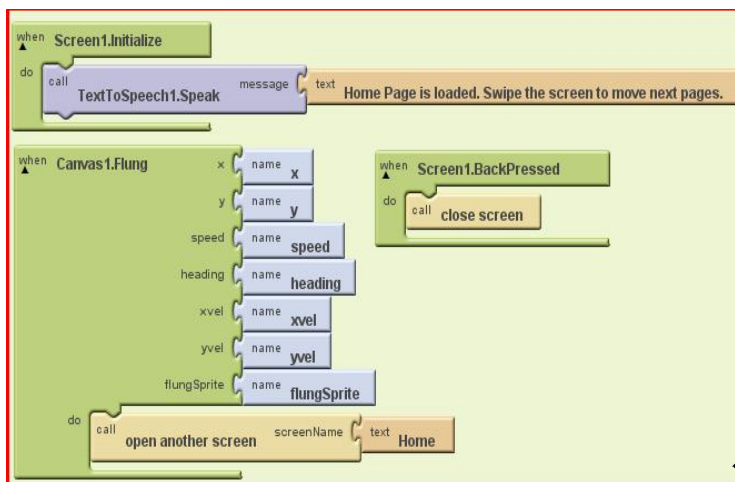


Figure 2. Code blocks for opening new page and closing the page.

When the page opens, the page title or page information can be recognized through TTS. The navigation in touch screen is linear on both sides (left or right) [5]. Hence, no trapping of events takes place during navigation. However, if the flung event is used for multiple simultaneous functionalities, such as navigating to the next page and navigating to the next control in the page, conflict occurs. The user can press the back button to exit the page or to go to the previous page.

Findings

1. *Open another screen* and *open another screen with values* code block are used to open a new page.
2. Flung event of the canvas or Button Click event is handled to navigate through the pages.
3. Back Pressed event of the screen is used to navigate to the previous page or home page.
4. When a page is open, a blind user has to be informed about the information on the page using TTS.

In-page Navigation

The blind user reads the information in pages through a screen reader. The screen reader will read the contents in a linear fashion sentence-by-sentence, automatically scrolling over the pages until it reaches the end. If the page is lengthy, the user has to spend a prolonged duration listening to the audio until it is complete. If the audio reading the content is programmatically controlled, it will read the whole page irrespective of scrolling of the page. It is tricky for the blind user to go to a desired section. In our courseware, we adopted a technique to demonstrate in-page navigation by reading the information sentence-by-sentence (see Fig. 3).

Findings

1. Navigating section-by-section or sentence-by-sentence in a page will make it easier for a blind user to grasp the information easily.

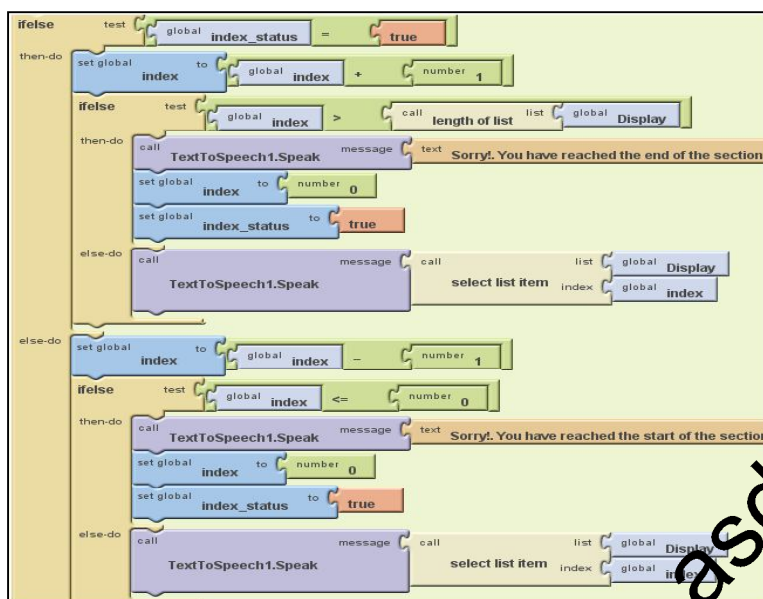


Figure 3. Code Blocks for In-page navigation.

IV. CONCLUSION

Although touch screen has many accessibility issues to address, this paper demonstrates how to adopt WCAG 2.0 as a basis for addressing accessibility to a significant extent for TMCA. Careful planning and designing is needed to achieve accessibility. Currently, touch screen accessibility largely depends on voice recognition. The researchers found that voice recognition failed in two main aspects: 1) Presence of background noise when using voice recognition and 2) Exact word or command not being extracted from the voice. Therefore, our paper suggests techniques that can be used to satisfy various success criteria of the WCAG.

Although W3C's Web accessibility guidelines provide an excellent framework for establishing the objectives of accessibility for blind users, more work needs to be done toward demonstrating methods for actualizing those objectives for educators who wish to further pursue the development of practical applications. TMCA accessibility can be realized by way of the methods proposed in this work. Additionally, our methods make the objectives established in the W3C guidelines a tangible reality while taking into account the rapid changes in technology and their deployment as well as advancements in users' knowledge and expectations.

REFERENCES

- [1] G. S. "Accessibility - New Frontier for Mobile Technologies," Disabled World, 15 03 2011. [Online]. Available: <http://www.disabled-world.com/communication/mobile-technologies.php#ixzz1dC2jQt6C>. [Accessed 19 07 2014].
- [2] Apple, "Accessibility," 2014. [Online]. Available: <http://www.apple.com/accessibility/ios/>. [Accessed 15 07 2014].
- [3] M. Fakrudeen, S. Yousef, A. Hussein and M. Ali, "Presentation Design of E-assessments for Blind Users Using Touch Screen Devices", Proc. of the International Conference on Electronics, Computers and Artificial Intelligence, IEEE ECAI 2013, 27-29 June, 2013, Pitesti
- [4] M. Fakrudeen, S. Yousef and A. Hussein, "Finger Based Technique(FBT): An Innovative System for Improved Usability for Blind Users Dynamic Interaction with Mobile Touch Screen Device", Lecture Notes in Engineering and Computer Science: Proc. of The World Congress on Engineering 2014, U.K., 2-4 July, 2014, London. URL:http://www.iaeng.org/WCE2014/doc/titles_E_N.html [Approved]
- [5] M. Fakrudeen and S. Yousef, "M-Learning: Development Of User-Friendly Navigational Structures For Blind Users Of Mobile Touch Screen Devices," International Journal of Engineering and Technology, p. [Accepted], 2014.