Accessible Question Types on a Touch-Screen Device: the Case of a Mobile Game App for Blind People

Barbara Leporini and Eleonora Palmucci

ISTI - CNR, Via G. Moruzzi, 1 - 56124, Pisa (Italy)
barbara.leporini@isti.cnr.it, palmuccieleonora@gmail.com

Abstract. This study investigates accessibility and usability via screen reader and gestures on touch-screen mobile devices. We specifically focus on interactive tasks performed to complete exercises, answer questionnaires or quizzes. These tools are frequently exploited for evaluation tests or in serious games. Single-choice, multiple-choice and matching questions may create difficulties when using gestures and screen readers to interact on a mobile device.
The aim of our study is (1) to gather information on interaction difficulties faced by blind people when answering questions on a mobile touch-screen device, and (2) to investigate possible solutions to overcome the detected accessibility and usability issues. For this purpose, a mobile app delivering an educational game has been developed in order to apply the proposed approach. The game includes the typical question types and exercises used in evaluation tests. Herein we first describe the main accessibility and usability issues reported by a group of visually-impaired people. Next, the game and its different exercises are introduced in order to illustrate the proposed solutions.

Keywords: accessibility, mobile interaction, visually-impaired users, mobile games.

1 Introduction

Several mobile and web applications exploit exercises and questions for practice and evaluation purposes. Activities like pairing words or ordering items are also widely used exercises in digital games. Thus, potential application fields may be learning systems (e.g., for foreign languages) and educational entertainment (e.g. serious games). Although several guidelines have been proposed in the literature [7, 8, 12], digital games and learning systems need to be further investigated in terms of accessibility and usability for visually-impaired people [1, 10]. This should especially be done for interaction with touch-screen devices via a gesture-based screen reader.
In our study we investigated the accessibility and usability of mobile user interfaces for screen reading users. We specifically focused on the interaction with questions and exercises widely used for quizzes, games or evaluation tests. Exercises such as pairing words or ordering elements may create a number of issues when interacting via screen reader and gestures on a touch-screen. Our purpose is to investigate how to overcome those issues. First, a group of visually-impaired users were asked to report

their main accessibility and usability problems. Next, a mobile educational game was developed to test potential solutions proposed in order to enhance the interaction via screen reader and gestures with a case study.

This work focuses on the following goals:

(1) Gathering evidence of accessibility and usability issues encountered by the visually-impaired when interacting with exercises and questions on a touch-screen device;

(2) Designing various types of test typology that are commonly used for exercises and questions in the education context to practice specific topics, especially for an accessible gesture-based interaction on touch-screen devices.

To this aim, we developed a mobile app for touch-screen devices (tablet or smartphone) of an educational game composed of various exercises, which allowed us to test some technical solutions applied to different case studies on a touch-screen, especially for screen reading interaction.

In the previous work [5] the app was introduced to describe the design of the themes and layout in the user interface. This was done with the aim of providing equal auditory and visual content perception for sighted and non-sighted players. This paper describes the main mobile device interaction problems encountered with existing games and learning exercises. In addition, it illustrates how puzzles and exercises can be developed by describing the proposed solutions.

This paper is organized as follows: the next section introduces the methodology used; section 3 reports the issues encountered by the visually-impaired when interacting with questions and specific tasks; section 4 describes the proposed solutions applied to the various exercises on the mobile game. Conclusions ends the paper.

2 Method

Firstly the main accessibility and usability issues related to interaction with quizzes and exercises via a gesture-based screen reader were collected. This was done by interviewing a small group of visually-impaired people recruited by the local Italian Association for the Blind. The users were asked to report problems encountered when interacting with puzzles and exercises, in particular when carrying out specific actions. The common use cases in which the issues are encountered were also identified.

For our purposes, the main common types [6] used when preparing quizzes, tests for practicing and evaluating specific educational topics were considered in the users participatory study. The typologies of exercises and questions considered in the study are summarized as: Single-choice (just one answer is right); Multiple-choice (more than one answer may be right), Matching items (one item of a first list is paired with another one of the second list); True / False, Ordering items, Gap-filling (completing the sentence with an item).

On the basis of the reported accessibility issues, possible solutions have been proposed and applied to the exercises developed for a mobile game in order to make them more accessible for the visually impaired. In designing the different types of exercises, we considered the gestures available on a touch-screen which are compatible with the main screen readers for a smartphone / tablet. Examples of these are

VoiceOver[1] for IOS systems and TalkBack[2] for Android platforms. This is important to ensure the App gestures are compatible with those available in the screen reader for touch-screen devices. We relied on simple gestures and VoiceOver-like interaction techniques with menus (single tap to hear an option and double tap to select it) in line with earlier accessibility work [4].

The mobile game prototype was developed using the Cordova Framework to design a cross-platform app. Since the user interface can be developed via X/HTML, CSS and JSs, the WAI-Aria suite [11] was used to enhance accessibility features. Studies like [2, 9] already applied WAI-Aria to ensure accessibility for the Web, in our work we test its efficacy for a mobile interface.

3 Exercise Types: Accessibility and Usability Issues

3.1 Participants

Five visually-impaired users were involved in our study. Their ages ranged from 25 to 70 years old. All of them had experience in using ICT and mobile devices with screen reading software: 2 people out of the 5 had experience of the Android-based devices and 3 people of the IOS operating system. All the users showed interest in a mobile digital game based on puzzles for a number of reasons. One important aspect is related to the lack of digital games for people with vision impairments. In addition, performing exercises or answering questions can be difficult, even impossible to accomplish due to the actions required, such as drag-and-drop. The issues described in this paper are essentially referred to mobile phone interaction. However, the users reported that they widely encountered these problems also with Web pages and applications. This implies a significant limitation in access to a number of online educational games or learning apps (e.g. for learning a foreign language).

3.2 Interaction Issues

Most interaction problems are related to the exploration mode used to detect text and items on the touch-screen. The exploration may be either (1) touch (i.e. slowly dragging one finger around the screen), or (2) swipe linearly to hear the items in order [3]. The main issues (from I1 to I6) reported by users can be summarized as:

I1 - Reading question content. The text is not always automatically read by the screen reader when a question is selected and displayed on the screen. Therefore, the user has to explore the screen with his/her finger in order to be able to read the text. For complex interfaces, especially in games, being able to explore and locate the right text can require a considerable effort for a non-sighted person.

I2 - Finding items. All the users stated that they often encounter problems in properly identifying the items related to possible answers. This occurs especially

[1] https://www.apple.com/accessibility/iphone/vision/

[2] https://www.androidcentral.com/what-google-talk-back

when the items are randomly or horizontally arranged. Even when they are listed in two columns, it may be very difficult to clearly identify the correct column (left or right) for an item. In this case, it is necessary to explore the screen by touch (exploration mode (1)) rather than with a swipe gesture (preferred by 4 out of 5 users interviewed). As a result, performing tasks such as matching items, ordering items or selecting answers might be unfeasible. This can occur also for True/False questions, i.e. even when there are only two answers. This may be because the position of the answers is not consistent for all the similar questions or the items are located very far from each other. Another reason could be that some elements might not be 'touched' when exploring by touch. For the users 'Left' and 'right swipe' exploration is considered a more reliable way to reach all the elements on the screen (4 out of 5 users expressed this preference).

I3 - Moving items (drag-and-drop). Many exercises / questions require users to move or drag the items around the screen. This happens, for example, for ordering or pairing the items (e.g. matching exercises). When performing drag-and-drop, even if the user is able to identify and select the first item, it is practically impossible to, find and select the second one (the target element or position) while dragging the element. In practice, both exploration modes (touch or swipe) do not work with drag-and-drop exercises.

I4 – Awareness of selected status. In some exercises or questions the user is asked to select more than one answer (e.g. multi-choice questions). Once an item has been picked up, it is selected and the user cannot pick it up again. Other tasks can lead to a similar issue: difficulty for the user to understand if an item has already selected or not. In these cases, the screen reader is typically unable to perceive the elements already picked up (i.e. the questions already completed or the answers already selected). Accordingly, the user is constrained to proceed by trial and error (e.g. by clicking on the element again) or by remembering the elements previously chosen. This interaction is consequently inaccessible, and more importantly unusable.

I5 – Detecting gaps to fill in. All participants reported difficulties in detecting which blanks are to be filled in and how to edit them. Such an issue is related to the Gap-filling task (complete the sentence by filling in the blanks). This happens especially on mobile interfaces since the screen reader often interprets the entire phrase/paragraph as a single chunk. Consequently, the selection of gap fields becomes impossible.

I6 – Receiving feedback and confirmation. Task success/failure is often provided to the user in a visual manner. Therefore, a blind user can encounter several problems in detecting it. For instance, sometimes items are marked or just graphically rendered in a different way (e.g. changes to image, dimension, color, or position). So, even if the element is textual, getting information on status changes can be very difficult or impossible.

4 The Proposed Solutions

The game has been designed with eight different exercises. These exercises represent the various types of questions, which are usually used in a test environment. While the work [5] describes the exercises s in terms of layout and themes to achieve an auditory perception equal to the visual rendering, this section explains the solutions proposed to enhance the interaction with specific tasks on a touch-screen. Specifically, through the development of the exercises we proposed potential technical solutions to solve the accessibility and usability issues summarized in the section 3.

Features common to all exercises

All the exercises address issue I1 with regard to properly reading the question content. The question text has been arranged according to the graphical (and visual) layout in order to make it as attractive as possible. This is an example of a graphical interface which is difficult for a blind person to use. Since the layout may require a scattered arrangement of the elements for graphics purposes, detecting some items by touch (such as the question text) can become a very difficult task for a visually-impaired user.

We resolve this reading issue (I1) by using HTML tags and WAI-Aria attributes (e.g. “role”, “aria-live”, “aria-atomic”, etc.). Interaction accessibility has been further improved in terms of touch accuracy. We noticed that the block size (i.e. the <div> width) affects the dimension of the tap-sensitive region: a larger <div> block requires a smaller touch accuracy to select the items. Consequently, some interface components have been enclosed in <div> blocks with a greater width. While this has a positive impact on screen reader interaction via gestures, it does not interfere with the visual rendering. As a result, the screen reader can detect an item regardless of its size and position within the block. As a result, detecting items is simpler for a visually-impaired user. See the green block in the Fig. 1 for an example: the ‘True’ is in a larger block (the color green has been used only to show the example).

Another issue common to the all exercises is related to feedback messages. As stated in issue I6, many difficulties are often connected to the graphical ways used for giving a feedback. To overcome this inconvenience, hidden <div> blocks have been added to the interface to provide information detectable only by the screen reader. Exercise 1 and exercise 3 are two examples.

Single-choice (Exercise 1). This case has been implemented with a list of possible choices to be selected (only one right answer). The items are displayed in a column in the middle of the screen, thus facilitating exploration by touch (I2), as described above. Just one of them can be selected with a double tap. When tapping on an item, the answer is given and the game moves on to the next question. A graphical feedback is provided to the sighted user. Instead a feedback message is available to the screen reader to indicate the success/failure of the answer.

To make the feedback truly accessible (issue I6), a specific <div> block displaying the message has been added to the interface. The block is hidden so that only the screen reader is able to detect it. Moreover, to allow the screen reader to read it automatically, the added bloc needs to receive the focus.

Multiple-choice (Exercise 2). The question implemented through this exercise displays all possible answers in a vertical column (Fig. 1) as in the 'single-choice puzzle'. The difference lies with the need to select more than one item and so how to proceed the next question. In the 'single-choice question' after tapping on a choice, the game moves directly to the next question. This approach cannot be applied to the multiple-choice question since more than one choice needs to be selected before proceeding. In addition, for this question type issue I4 had to be considered.

To implement a multiple-choice question, we made it possible to pick up the items one by one via a double tap. Once the item is chosen, it is displayed in bold (for visual rendering) and the screen reader announces "selected" to inform the user of their choice This result was obtained via WAI-Aria and a hidden-label detected only by the screen reader. The user can select the other items in the same way. On completion, a 'Confirm' button allows the player to proceed to the next question, this need to confirm the action makes this exercise different from the 'single-choice question'.

True or false (Exercise 3). For this question type (Fig. 1), selection is very simple since only two choices are available (true/false | yes/no | etc.). When selecting True or False, the screen reader announces 'right/wrong' (to overcome issue I6) and the next question is immediately displayed (as for the single-choice).

Some previously reported issues were related to how easy it is to locate the two choices by touch (I2). To avoid this potential problem we firstly introduced consistency in the location of the two answers (i.e. always in the same position). In this way the user can learn the given position for the two answers. Then, we worked on the tap-sensitive areas for the two choices, i.e. the <div> blocks enclosing the answers have been developed with a larger dimension.

In this exercise we also considered issue I4 related to more than one question being available. A predetermined number of questions is available: the user can select them one by one. Once a question has been answered, it can no longer be selected. When 'touch' locates an element related to a question already completed, the screen reader makes an announcement, for example, "Question already answered".

Matching items (Exercise 4). This exercise consists in matching the items in two different sets. The items are displayed in two lists in two columns: one on the left and one on the right. It is common to complete the exercise using drag-and-drop to pair two items, however, this cannot be performed via screen reader and gestures.

To compensate for this, we designed a step by step selection process so as to match an item from one set with the corresponding one in the other set. The procedure is as follows: the first item in the left column is selected by a tap; the corresponding item on the opposite side can be picked up by another tap. At this point, the second item moves from the right to the left, pairing to the first one. Consequently, the two matching items are both shown on the left, marked as paired (Fig. 2). When finished to match all elements, on the right side there are no more items. The user confirms completion of the exercise via a specific button "End and Score". Through this approach the I3 issues related to drag-and-drop tasks are resolved. In addition, I2 (locating items) and I4 (awareness of selected status) issues are also considered in this exercise. In order to facilitate column detection (left and right), two hidden labels have been added before the two lists in order to inform the screen reader that the following elements are related to the left or right column.

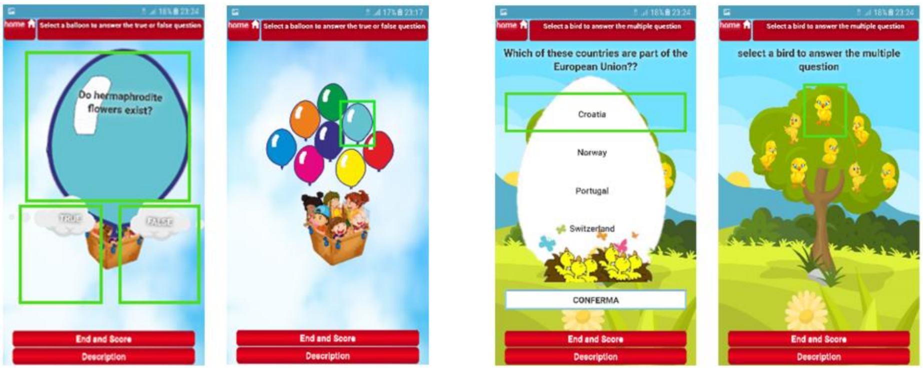

Fig. 1 Multiple-choice (on the left) and true or false (on the right)

Gap-filling Exercise 5). This exercise consists of a question to be completed by filling in the gaps. The sentence contains the ellipsis to indicate the missing term. To handle issue I5 (detecting gaps), the hidden label '[blank]' has been put in the position of the ellipsis to indicate the gap to the player. In order to avoid making editing too multifaceted, a list of potential answers is provided rather than asking the user to insert the word using the virtual keyboard. In this way, the puzzle is very similar to the 'single-choice puzzle', but the 'gap issue' has been solved. The items are visually arranged in a scattered order (see Fig. 2), but a blind user can perceive them in a regular list when exploring the interface via finger touch (i.e. I2). The solution was implemented with WAI-Aria techniques and hidden labels.

Fig. 2 Matching items (on the left) and gap-filling (on the right)

5 Conclusions

This work investigates the main accessibility difficulties encountered by screen reading users when interacting with touch-screen devices through gestures. Five blind users have been consulted to collect information on the problems arising when answering questions on a mobile device. Six accessibility and usability issues have been identified and summarized. A proposed solution aimed at avoiding those issues has been implemented via a mobile serious-game app for multiplatform systems. A few exercises have been designed. The app shows how exercises can easily be performed also via simple gestures (swipes and double taps) on a touch-screen by a blind user. The Cordova environment used to develop the app allows us to confirm that HTML tags and WAI-Aria techniques can support accessibility also for mobile multiplatform device with screen reader. In future, we plan to investigate interaction modalities for other typologies of exercises e structured user testing will be conducted.

6 References

1. Calvo, R., Iglesias, A., & Moreno, L. (2014). Accessibility barriers for users of screen readers in the Moodle learning content management system. Universal access in the information society, 13(3), 315-327.
2. Carvalho, L. P., Ferreira, L. P., & Freire, A. P. (2016). Accessibility evaluation of rich internet applications interface components for mobile screen readers. In Proceedings of the 31st Annual ACM Symposium on Applied Computing (pp. 181-186). ACM.
3. Google accessibility. Exploration by touch with Talkback, At https://support.google.com/accessibility/android/answer/6006598?hl=en
4. Kane, S. Wobbrock, J. O. and Ladner, R.E. 2011. Usable gestures for blind people: understanding preference and performance. In Proc. CHI ’11. ACM, New York, NY, 413422.
5. Leporini, B., & Palmucci, E. (2017, October). A Mobile Educational Game Accessible to All, Including Screen Reading Users on a Touch-Screen Device. In Proceedings of the 16th World Conference on Mobile and Contextual Learning (p. 5). ACM.
6. Madsen, H.S., Techniques in Testing. Oxford University Press
7. Mi, N., Cavuoto, L. A., Benson, K., Smith-Jackson, T., & Nussbaum, M. A. (2014). A heuristic checklist for an accessible smartphone interface design. Universal access in the information society, 13(4), 351-365.
8. Park, K., Goh, T., & So, H. J. (2014, December). Toward accessible mobile application design: developing mobile application accessibility guidelines for people with visual impairment. In Proc. of HCI Korea (pp. 31-38). Hanbit Media, Inc..
9. Spiliotopoulos D., Dalianis A., Kouroupetroglou G. (2014) Accessibility Driven Design for Policy Argumentation Modelling. In: Stephanidis C., Antona M. (eds) Universal Access in Human-Computer Interaction. Design for All and Accessibility Practice. UAHCI 2014. Lecture Notes in Computer Science, vol 8516. Springer, Cham
10. Yuan, B., Folmer, E., & Harris, F. C. (2011). Game accessibility: a survey. Universal Access in the Information Society, 10(1), 81-100.
11. W3C, WAI, Accessible Rich Internet Applications (WAI-ARIA) 1.1. Recommendation 14 December 2017, https://www.w3.org/TR/wai-aria-1.1/
12. W3C, WAI, Web Content Accessibility Guidelines 2.0, WCAG 2.0. http://www.w3.org/TR/WCAG20/