

Audio Editing with a Game Controller for Blind Users

Björn Kaiser, Matthias Lindner, Sascha Reinhold, Michael Teistler

Faculty of Information and Communication, Flensburg University of Applied Sciences

Abstract

Blind people can be regarded as predestined for professions in which audio material is processed. In such professions, editing audio files can be an everyday activity that can be difficult for blind people to manage with established audio editing software that provides user interfaces that are optimized for visual perception. We have developed an operating concept that allows a blind person without prior knowledge of the application to quickly complete basic tasks in audio editing, utilizing a standard game controller. The operating concept is based on a menu for selecting actions and controller buttons that can be pressed gradually to control the speed during fast-forward and rewind. Our results show that people with and without experience with game controllers were able to solve the given test tasks blindfolded.

1 Introduction

Blind people are naturally more dependent on their hearing and therefore hear more consciously (Starlinger & Niemeyer, 1981). This is why one could regard blind people as predestined for professions in which audio material is processed, e.g. at radio stations or publishing houses. In such professions, editing audio files is often an everyday activity that can be difficult for blind people to manage with established audio editing software such as *Audacity* (Audacity, 2018). Even a simple audio editor like that, compared to full-fledged digital audio workstations, has about four dozen controls in the main window. The developer's efforts for good accessibility are not sufficient to allow a blind person to use the application effortlessly. Accessibility markups allow assistive technologies such as screen readers to present the content of an application window to the user. With complex user interfaces, however, it can be difficult for the user to gain an overview. The use of keyboard shortcuts and macros can partly help to reduce this problem. Professional music production software solutions such as *Steinberg Cubase*, *Avid Pro Tools* or *Apple Logic Pro X* offer extensive functions for this purpose (Avid Technology, 2011). Keyboard shortcuts must first be learned and internalized by the user in order to make work easier. However, these measures do not change the fact that the user interface has been optimized for visual perception. Troy Allman et al. developed a modification for

the music game *Rock Band* (Allman et al., 2009), which allows blind people to rudimentarily play drums. Information and instructions are given by vibration on the upper and lower arm as well as on the ankle and by supplementary speech synthesis. Game developer Gavin Andersen's goal in his creations is to enable blind and sighted people to play at eye level with and against each other (Andersen, 2002). To make this possible, he develops acoustical user interfaces in addition to graphical user interfaces right from the start. Similarly, we concentrated on the development of an operating concept that allows a blind person without prior knowledge of the application to quickly complete basic tasks in audio editing by means of a standard game controller and an acoustical user interface.

2 Methods and Material

The prototype was implemented in C++ with the cross-platform framework *Qt*. On the hardware side, we chose a low-cost *Xbox 360* controller. This controller has a total of 11 buttons, a digital control pad and two joysticks. The controller features two shoulder buttons, L2 and R2, which can be pressed gradually, allowing for sensitive input. In addition, the prototype uses text-to-speech synthesis (TTS) to communicate information to the user. The application provides the basic audio editing functions of copying, pasting and deleting marked audio sections. After starting the application, an audio file is loaded, on which the test subjects are to perform tasks. As test material we chose an excerpt from an interview (Schäfer, 2017). By pressing the A button, playback can be started, paused or resumed. The L2 and R2 trigger buttons can be pressed gradually, allowing the user to fast-forward and rewind at the desired speed. During this process the audio material is audible (so-called scrubbing). If playback has been started before the L2 or R2 button is pressed, playback will resume seamlessly after scrubbing. Whenever the user presses an unassigned button, an error tone is played. To copy or delete a section of the audio material, it has to be selected. A section can be selected by marking its beginning and end with markers. To perform an action such as setting the start marker, the user must first press the control pad to open a menu with several available actions (see figure 1.1). By pressing the up or down button of the control pad an action can be selected. By means of TTS, the currently selected action is verbalized to the user. By pressing the A button, the user can initiate the execution of the selected action. The result of an action is acknowledged with a verbalized success or error message. If the user reaches the last item while navigating through the menu and advances further, the first item is selected again. The same applies vice versa if the first element is reached. One could imagine the menu as a wheel. To reduce the navigation steps in the menu, we let the application predict which action would be selected next by the user. For instance, it is likely that the user will select the paste action after activating the copy action. Figure 2 shows the sequence of actions necessary to find, mark, copy and paste an audio section at a target position. Figure 1.2 shows the general activity diagram for this application.

Five users with and without game controller experience tested our prototype. In the absence of actual blind test subjects, the testers were blindfolded. Afterwards they were instructed about the button assignment and asked to familiarize themselves with the controls. The first task was to remove both of the interviewer's questions from the audio material. The second task was to

cut out the opening melody and paste it back at the end of the file. Finally, the testers were asked to save the audio material to a file. Afterwards the testers were interviewed to learn about their impressions, difficulties and suggestions for improvement. We also asked them to complete the User Experience Questionnaire (UEQ). In order to gain further insight for the analysis of the tests, we added a log function to the application.

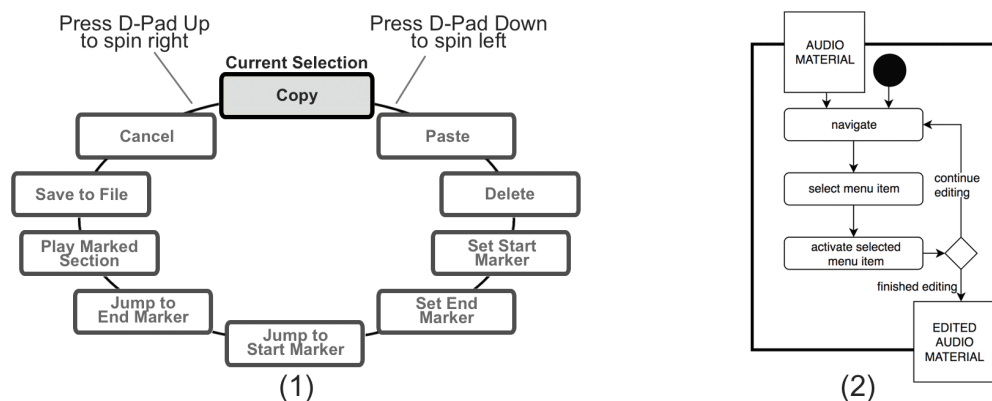


Figure 1: (1) Actions menu (2) Activity diagram: “navigate” refers to navigation within the audio material.

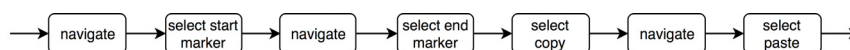


Figure 2: Action sequence for copy and paste

3 Results

All testers were able to solve the tasks in five to ten minutes (see figure 3.1). Experience with game controllers was not an advantage. It was noticeable that after a short time all testers developed routine in repeating the similar steps necessary to complete the tasks. The time measurements include the time between first setting a start marker and saving the audio.

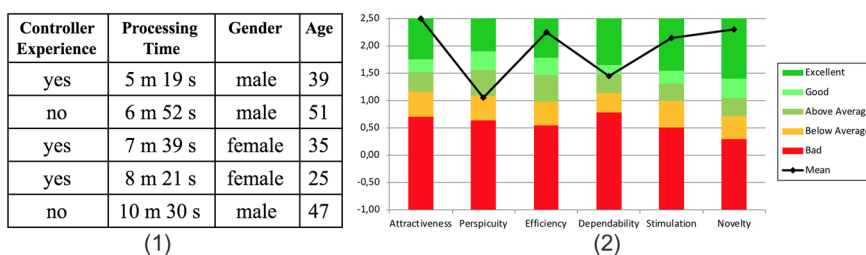


Figure 3: (1) The times the testers needed to solve the tasks. (2) UEQ results

The prototype achieved average to excellent results in the evaluation of the UEQ (see figure 3.2). However, two testers were confused using the menu. One tester assumed that she would always have to close the menu with the cancel action after activating an action. She also tried to close the menu by pressing left or right on the control pad. She imagined the menu as an expanding menu that would have to be closed before she could continue with the next action. In fact, however, the menu closes when an action is activated. Another tester was confused that the last activated action was not preselected when reopening the menu. As mentioned, the application tries to predict which action the user might want to activate next. This behavior caused the confusion. The tester said that he would prefer the application to always preselect the most recently activated action when opening the menu.

4 Discussion

The confusion of one of the testers could be traced back to the idea of an expanding menu. This idea may be rather unknown to an actually blind person, which is why this irritation might not occur with him or her. As an improvement the tester suggested to indicate the appearance and disappearance of the menu with suitable sounds. This would be a behavior known from screen readers, which usually tell the user when a menu is opened or closed. It is possible that blind people would imagine the menu differently from sighted people, who use visual models or metaphors for interpretation. Our results show that we have succeeded in implementing an operating concept that people with and without experience with game controllers describe as intuitive to use. It can be assumed that people who are actually blind can get used to such a control concept even faster than our test subjects, who needed time to concentrate solely on their hearing. Our testers with controller experience suggested assigning actions from the menu directly to previously unassigned buttons on the controller. However, assigning additional buttons can lead to overload for users without game controller experience. Further tests could be used to find out when the operating concept is becoming too complex for users who are inexperienced with game controllers or with which measures one could counteract emerging overloads. Another approach would be to introduce beginner and expert modes. Future tests should be conducted with actual blind people and include a comparison of the presented approach with an operating concept based on keyboard shortcuts. Interviews with the blind could reveal features they would like to see in such an application.

References

- Allman, T., Dhillon, R. K., Landau, M. A. E. & Kurniawan, S. H. (2009). *Rock Vibe: Rock Band® computer games for people with no or limited vision*. In Trewin, S. & McCoy, K. F. (ed.): *Proceedings of the 11th International ACM SIGACCESS*. Pittsburgh, PA, USA: ACM. p. 51-58.
- Andersen, G. (2002). *Playing by Ear: Using Audio to Create Blind-Accessible Games*. On Graft, K. & Wawro, A. (ed.): *Gamasutra*. San Francisco, CA, USA: UBM TechWeb. www.gamasutra.com/view/feature/3493/playing_by_ear_using_audio_to_.php (last accessed 18 May 2018)

- Audacity. (2018). *Accessibility*. In *Audacity Manual*. manual.audacityteam.org/man/accessibility.html (last accessed 18 May 2018)
- Avid Technology, Inc. (2011). *Pro Tools® Shortcut Guide Version 10.0*. Daly City, CA, USA: Avid
www.protocolshdx.de/SMM/pdfs_files/Pro%20Tools%20Shortcuts.pdf (last accessed 26 May 2018)
- Schäfer, S. (2017). *Alexa Skills für Amazon Echo mit PHP entwickeln | Interview mit Ralf Eggert*. Frankfurt, Germany: Software & Support Media GmbH. www.youtube.com/watch?v=qraw039D4lk (last accessed 30 May 2018)
- Starlinger, I. & Niemeier, W. (1981). *Do the Blind Hear Better? Investigations on Auditory Processing in Congenital or Early Acquired Blindness I. Peripheral Functions*. In Grune & Stratton (ed.): *Audiology*. USA: International Society of Audiology. p. 503-509.