Mobile Soundscape Mixer – Ready for Action

Kari Salo[1(✉)], Merja Bauters[1], and Tommi Mikkonen[2]

[1] Helsinki Metropolia University of Applied Sciences, Helsinki, Finland
{kari.salo,merja.bauters}@metropolia.fi
[2] Tampere University of Technology, Tampere, Finland
tommi.mikkonen@tut.fi

Abstract. Today, cultural organizations such as museums are seeking new ways to attract and engage audience. Augmented reality based applications are seen very promising. The target is to provide more interactive experiences for an audience with high familiarity of digital interaction. So far, visual presentation has been dominant in augmented reality systems. In contrast to this trend, we have chosen to concentrate on audio augmentation as user generated soundscapes. This paper discusses our approach, focusing on how to design and develop an easy-to-use and smoothly working Android application, which increases user interaction by developing soundscapes from building blocks stored in audio digital asset management system. We have successfully implemented applications for Android platform and evaluated their performance.

Keywords: Soundscape · Android · Audio Augmented Reality · Research-based design · Participatory design · User centered design · Performance profiling

1 Introduction

Interaction with environment increases and enhances each day. Augmented reality (AR) is one way to provide increasingly interactive experiences. Audio Augmented Reality systems have been used for navigation [1] and interacting with virtual objects [2]. Most previous projects have used extra devices such as a headphone-mounted digital compass or Kinect depth camera to track user's (head) location to produce sound that is modified according to user's movement.

As we believe that interaction and experiences are holistic using all senses, we have decided to concentrate to auditory presentation, focusing on the acoustic environment. To this end, a soundscape can be a musical composition, a radio program or an acoustic environment [3]. A soundscape is created out of multiple, time-varying sound sources [4]. Many of the soundscape systems – such as Klang.Reise [5] and the Sound Design Accelerator (SoDA) [6] – are either targeted to sound designers and need a lot of knowledge to operate, or require a dedicated space. We have combined these two concepts - soundscapes and audio AR. Our approach aims at ease-of-use and interaction without previous knowledge on sounds and soundscapes. Thus, the user is the active party and technology is in the supporting role either for searching relevant sounds with the help of mobile applications or producing the acoustic environment using her creativity and imagination. The user is not expected to be familiar with

M. Younas et al. (Eds.): MobiWIS 2016, LNCS 9847, pp. 18–30, 2016.
DOI: 10.1007/978-3-319-44215-0_2

acoustic terms or dependent on extra devices for tracking her head and hand movements, when creating soundscapes.

Our project, The Neighborhood Living Room, studies different methods how to create a more dynamic, participatory audience relationship with area residents (especially youth) and the Museum of Technology in Helsinki, Finland.

When developing audio AR and soundscape systems ease-of-use of the application and the backend service supporting these applications are two key aspects. Our target is to design and implement smoothly working mobile soundscape mixing application to increase user interaction by developing soundscapes from building blocks stored in audio digital asset management system.

The paper presents our results so far regarding the design of mobile applications. It is organized as follows. First we describe the overall system, design process, application development, and performance evaluation of the mobile applications. In the discussion we ponder on the outcomes we have achieved, and in conclusion we sum up the process and outline the need of further research.

2 System Overview

The overall system consists of an audio digital asset management system (ADAM), a management application, and mobile applications (Fig. 1). ADAM provides functionalities to manage assets and offers interfaces for both for management application and mobile applications over the Internet. The management application is more or less an administration console to manage assets and users. Mobile applications are for example audio augmented reality, soundscape design, audio story recording and listening, or audio memory sharing applications. This paper concentrates on a specific mobile application, soundscape mixer, represented as "the Mobile Apps" in the Fig. 1.

As pointed out above, the management application is used both for user management and asset management. The museum staff is able add new users, add audio file collections to users, and define which audio files belong to these collections. Users can be persons or devices. Sound designers as users can add new audio files, which are used as sound-scape building components. On the other hand a mobile device can be

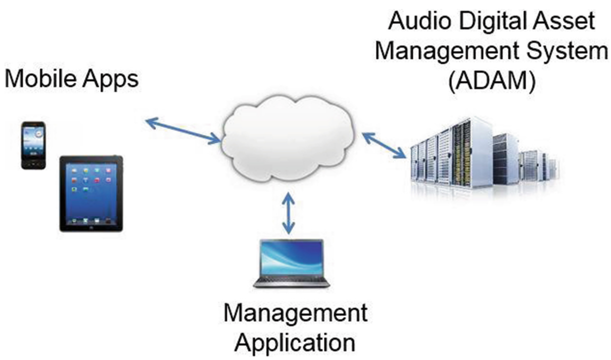

Fig. 1. System overview.

seen from the ADAM perspective as a user who would like to access audio files. To enable communication between ADAM and mobile applications three APIs were required: an authentication API, a search API, and an upload API. The authentication API is needed by the mobile users to receive a token, which in turn will be used with search and upload of APIs. The authentication is a security feature and ensures that only authorized users have access to token. The search API is a HTTP GET request containing token and predefined search parameters. The response given in JSON format contains audio files' metadata based on search parameters that are set along with the search request. Thus, the search API also enables downloading, as the link to audio file is a part of the metadata. The upload API lets users with a valid token upload their audio files along with metadata they choose to transmit to ADAM as a multi-part form using HTTP post. Metadata and token will be encoded in a part of the URL, and the audio file in the body of HTTP post (see Fig. 2).

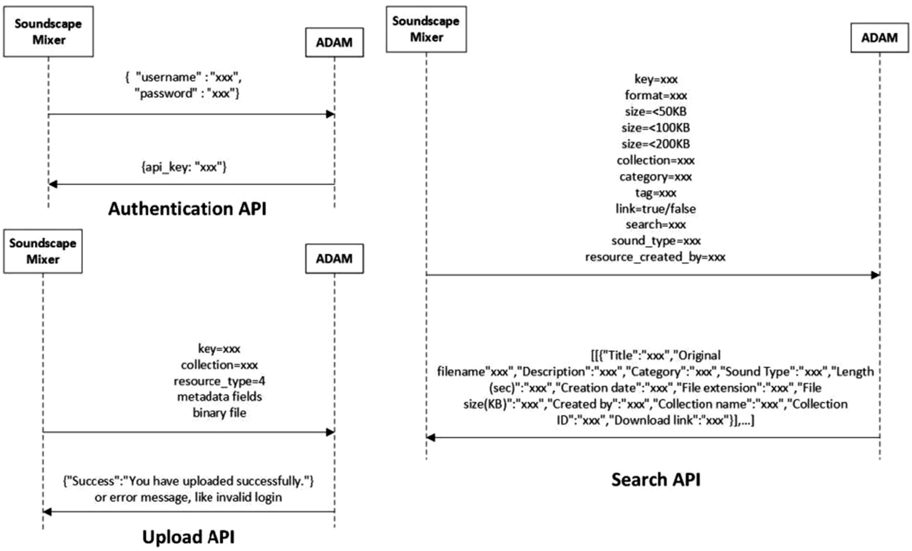

Fig. 2. APIs between soundscape mixer and ADAM

When searching, finding, and utilizing relevant audio files, it essential to use metadata. There are several metadata standards for different purposes, such as metadata exchange between systems, general metadata for broad range of domains, and audio specific structural and administrative metadata. We chose metadata set, which in the future enables the exchange of assets by supporting Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH). Most of the metadata will be input manually during the storage of audio files, although some of the metadata will be extracted automatically from the audio file properties [7].

3 Design Process of the Mobile Soundscape Mixer Application

For designing the mobile soundscape mixer applications, the research-based design approach [8] was used as a design process. It is an iterative process consisting of the following phases: contextual inquiry, participatory design, product design and prototype as hypothesis. The phases were executed in parallel. The emphasis of a phase changes during the process. The aim of the design was to design and implement a soundscape mixer application for mobile devices. Third year students from Helsinki Metropolia University of Applied Sciences, Helsinki, Finland carried out the design process. The students came from two courses (both of which lasted one semester, or three and half months): design-oriented course called Usability and Interface (28 students) and Android programming course called Android Advanced Application Development (19 students). Four sound design students created the sounds to be used in soundscape creations. We formed seven teams out of the design and programming courses. The teams were composed of 3–4 design students and 1–3 programmers. The design and programmer students had 5 organized meetings to present, organize, and test their intermediate outcomes. Between these joint meetings the sub-teams of designers and programmers were working on their own fields. In the end of the fifth iteration round a workshop was organized in the Museum of Technology. Each iteration included design in different levels of prototype granularity, ranging from low fidelity prototypes (see Fig. 3) to a running prototype (see Fig. 4).

The prototypes were tested in each of the iterations with users after which the students negotiated within their teams how the programming should continue, and how to prioritize features and functions of their applications [9, 10]. The aim was at creating a working mobile application for young museum visitors ready for the workshop.

The first phase of the design process was to familiarize with the context. The design students visited the Museum of Technology, discussed with the museum personnel, investigated the physical place and brainstormed on design [11]. The visit lasted three hours, and the students collected contextual material in the form of images and notes. The outcomes of the contextual inquiry were paper prototypes (see Fig. 3). The paper

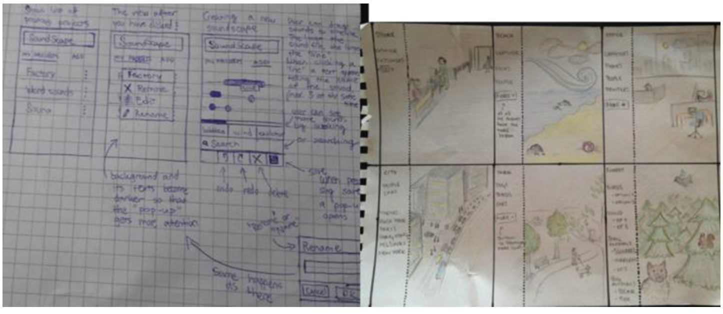

Fig. 3. Two different kinds of paper prototypes which were created after the visit to the Museum of Technology.

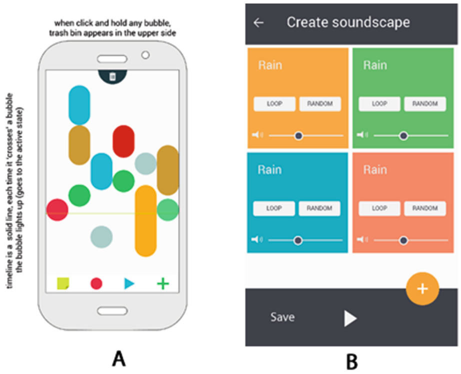

Fig. 4. (A) The left side screenshot displays an innovative idea on how to mix the sounds in a visuospatial manner. Each sound is represented as a colored bubble. The length of the bubble indicated the duration of the sound and the position of the bubble indicated when the sound is played in the timeline. (B) The right side displays an interface to mix the sounds in a more conventional orientation and interaction flow. Still, however, it achieves a clear interface for creating a soundscape. (Color figure online)

prototypes were discussed with the programmer students to construct a solution that is implementable with the teams' resources. The students, museum personnel, lecturers, and potential users listed out the first set of functional requirements [12], which were: listen, record, search files, save, delete and mix sounds for soundscape, login and out as well as sound file categories.

The created prototypes were tested with users belonging to the defined target group of 15–24 years old. Since the students themselves also belonged to the defined target group, they could pre-test their ideas with each other. The tests were video recorded and analyzed. Improvements were executed according to the test results for the next prototype (see Fig. 4).

The screenshot provides a glimpse over the multiple iterations that the students' teams performed. The screenshots are revealing since the idea of how to mix a soundscape is very different, but both are still easy and pleasant to use. In addition, one group implemented a QR code reader, through which sample sets of sounds can be retrieved in the museum. The technical implementation of the design ideas will be presented next.

4 Guidelines and Implementation for the Mobile Soundscape Mixer Application

When starting soundscape creation application (soundscape mixer) development we had to make a decision what smart phone platforms to support. As the Android platform is dominant at the moment, it was an obvious choice [13, 14]. Also an iOS version as well as designing a hybrid solution that would work on both platforms were considered, but we did not have any native iOS developers in the teams, and based on our previous experience on hybrid development [15], we ended up to selecting native Android application as web based audio APIs are not mature enough.

In order to coach programming teams towards the target, we provided some guidelines. From the development point of view these were the guidelines:

- Iterative/agile development process;
- Support Android 5.0 and newer version;
- Follow Google's material design guidelines;
- Support MP3 and/or PCM/Wave-format;
- Utilize Soundpool or Audiotrack classes for playing audio files;
- Utilize AudioRecord for recording.

Programming teams were given the following rough functional requirement specification:

- Login into Audio Digital Asset Management System (ADAM);
- Search content (audio files) in ADAM utilizing metadata;
- Download, save and play selected files either in MP3 or raw (PCM) format;
- If needed convert audio file format;
- Mixing, i.e. define combination of saved files that will played, possibility to loop, change volume, etc. of each audio file separately;
- Record audio file, convert the audio format and upload together with metadata into ADAM.

Following the guidelines and requirements, the design and programmer teams were able to implement mobile soundscape mixer applications according to iterative process described above. Four of the development teams were able to provide a fully working and tested application within the given timeframe. All of these four applications were in some respects different from each other. This was expected, as we were hoping to see each team use their imagination and creativity when designing and implementing the application. To test these applications with real users, we organized a workshop. For the workshop three applications were chosen for testing. These applications (Sound Bubbles, SoundSpace and SoundScape), were the most advanced applications. The UniChord application was left out because the Museum of Technology did not have the QR codes in their museum items or exhibition spaces.

The three remaining applications were implemented according to Android technical guidelines and best practices. As can be seen from the high level class diagrams (Fig. 5) object oriented approach was followed, multithreading was applied where required to avoid UI thread blocking, required functionalities were implemented, and recommended audio classes were used.

For security reasons, user identity and password were required when starting the application. In addition, also a collection identity was asked from the user. User identity and password are sent to backend service (ADAM), where they are checked and only authorized application will receive as a response security token, which will be used when sending search, download or upload requests to backend service. Each user identity could have several collections of audio files. Thus collection identity is used to limit which particular collection application is able to use. Implementation of this login

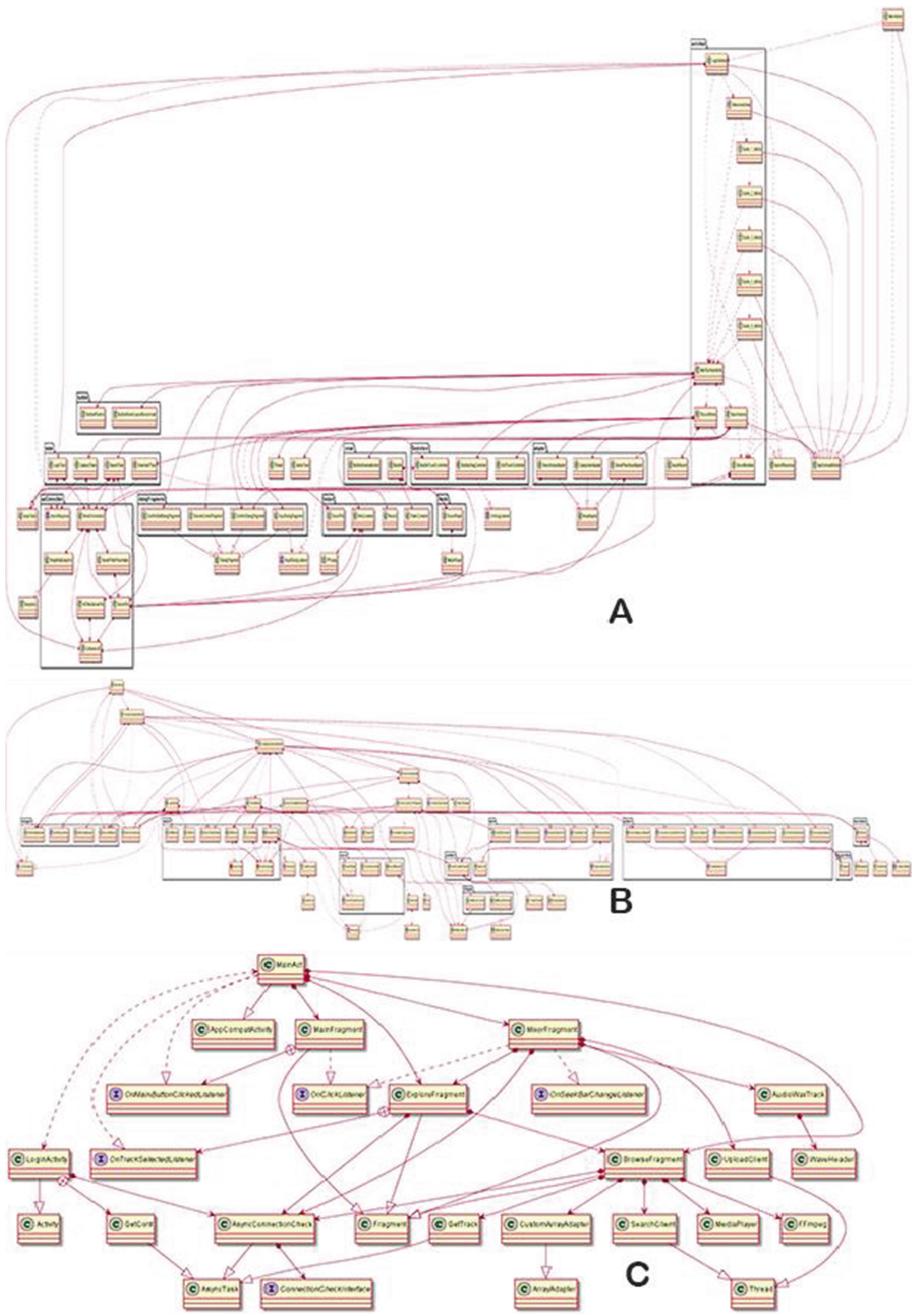

Fig. 5. Class diagrams describing the structure of applications. From top down there are A Sound Bubbles, B SoundSpace and C SoundScape applications.

functionality was similar in all applications, the only difference originating from the possibility to save credentials locally into phone's memory.

Searching audio files from ADAM was typically implemented either by providing a free search based on title or by asking user to select one of the four predefined categories (nature, human, machine and story) and then displaying as a scrollable list or grid all the titles. List and grid then provide a possibility to listen sounds before selecting them as a part of soundscape.

The possibility to record own audio files for using them as soundscape building components was found in all the applications. After recording these files, the files can be used locally or uploaded with metadata into backend service and thus shared with other users. Implementation of this recording and uploading functionality followed tightly the material guidelines being similar in all applications.

Finally, when all the soundscape components are available, the main functionality – mixing or creating the soundscape – can be described. Implementing mixing functionality differs from application to application. The applications implemented either time limitation or component limitation into their sound mixer. Sound Bubbles application is based on the idea of one minute soundscape and it can be divided into six parallel audio tracks. Each track can have zero or more audio files. SoundSpace and SoundScape applications, however, have limited the amount of audio files (Fig. 6). Looping either the whole soundscape or separately each audio file produces a longer soundscape than is possible with Sound Bubbles application.

Fig. 6. Soundscape mixing functionality. From left to right there are A Sound Bubbles, B SoundSpace and C SoundScape applications

When the user is satisfied with her soundscape, she is able to save it either using digital audio recorder connected to Android phone's audio line-out, or in case of SoundScape (C) application, upload the soundscape file into ADAM. Testing the performance of these applications is described next.

5 Performance Evaluation

Application's performance has a vital impact on user experience. Testing the presence and effects of poor responsiveness is challenging due to non-existing testing strategies for exposing causes of poor responsiveness in Android applications. Some research has been done and approaches proposed [16, 17]. We decided to use tools available as a part of Android Studio, in particular the Android lint tool that checks Android project source files for potential bugs and optimization improvements. We run lint code analysis for all three applications to get an overall picture of potential problems. The following table (Table 1) describes the relevant findings. In addition, based on empirical study [17] we decided to mostly concentrate on GUI lagging type of performance bugs. Based on findings, it was necessary to check if there are problems with list scrolling (solution View Holder design pattern). As we are not heavily using long strings then potential StringBuffer problems do not have a major impact of performance.

Table 1. Static code analysis results.

Application	Lint category	Subcategory	Class name
Sound bubbles	Android lint	View holder candidates	• CategoriesAdapter • RecordingsAdapter • ServerFilesArrayAdapter
SoundSpace	–	–	–
SoundScape	Android lint	View holder candidates	• CustomArrayAdapter
	Performance issues	String Concatenation as argument to 'StringBuffer.append()'	• LoginActivity • SearchClient • WaveHeader
	Performance issues	'StringBuffer' can be replaced with 'String'	• WaveHeader

Before checking dynamic rendering of the frames of UI window we will use one more static tool, the Hierarchy Viewer. This tool visualizes application's view hierarchy and profiles the relative rendering speed for each view. We aimed to spot red dots in leaf nodes or view groups with only a few children. As an example, find the Hierarchy View tool's results for SoundScape (Fig. 7). As seen in the figure, there are no potential problem areas in leave nodes except in EditText, where the draw process could be slow. This EditText view refers to login screen's password field. When running on a device it seems to work smoothly. So far we have used static information. Next step was to use the GPU Monitor, which gives a quick visual representation of how much time it takes to render the frames of a UI window. It profiles the amount of time it takes for the render thread to prepare, process, and execute the draw commands.

As an example we have run the SoundSpace in Samsung Galaxy S5 and found some potential parts of the application where the user may see slower response than

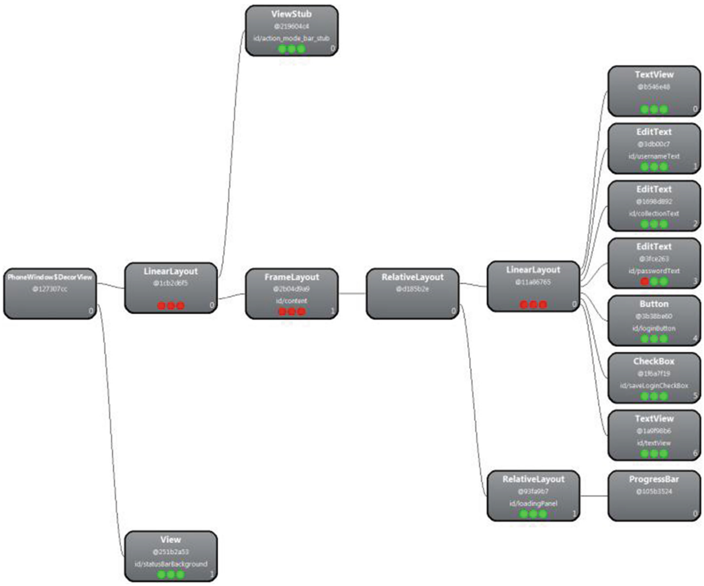

Fig. 7. SoundScape application's relative rendering speed for each view

expected (see Fig. 8). Adding audio components to soundscape is the main functionality of the application. The results are promising. Only few frames are exceeding 16 ms (green) line. This 16 ms frame duration is calculated from the recommended frame rate 60 frames per second, which ensures that user interactions with application are buttery smooth [18]. Saving soundscape project will happen very seldom, but it is pretty evident that user will see some slowness. We used GPU Monitor to test all three applications using Samsung Galaxy S5 and LG Nexus 5 phones. The results were similar to the above described SoundSpace results.

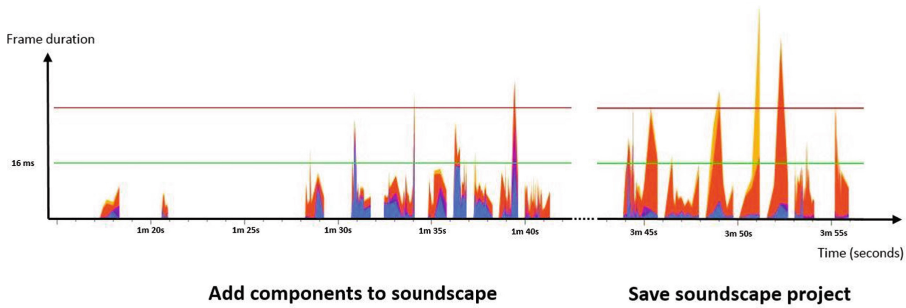

Fig. 8. SoundSpace GPU monitoring (Color figure online)

6 Discussion

Our overall system consists of pretty simple audio digital asset management system and smart clients. This type of architecture enables to utilize full power of mobile platforms when developing audio related applications, like soundscape mixer. This in turn results into innovative applications. Selecting auditory presentation instead of visual one enables faster communication between mobile clients and backend service as transmitted audio files are typically smaller than video or 3D model files. We believe that distributing main functionality to mobile platform and keeping traffic light between client and server will be the basis for smooth interaction.

We have seen that research-based design process and teams consisting of designers with user experience angle and programmers with understanding of Android platform capabilities and limitations will result into realistic and highly usable design. It was important that the target group (15–24 years old) for the application was defined already in the beginning. Thus we were able to test prototypes by the target group.

During the implementation phase Android best practices were followed, which ensured that the interaction with application follows Google's material design and avoids the common implementation pitfalls. This is needed for two reasons:

- Application should look like and behave like Android application so that Android phone owners will feel comfortable without any surprises;
- Most trivial performance bottlenecks will be avoided.

As application's performance has a vital impact on user experience it is important to evaluate performance before releasing applications. So far there are no (de-facto) standard testing strategies for exposing causes of poor responsiveness on Android applications. Thus we decided to utilize tools that are available as a part of the Android Studio. Static code analysis results confirmed that most of the common Android development pitfalls related to performance were avoided. As the GUI lagging is the most common performance bug we decided to check GUI performance. Hierarchy Viewer tool provides valuable information about potential rendering speed bottlenecks. In our case all three applications did not expose any major bottlenecks. Finally utilizing GPU monitor tool provided the realistic picture how the application behaves in real world environment. We run all three applications on Samsung Galaxy S5 and gathered GPU monitoring data. We did not find any major potential problems. On the other hand, none of the applications could stay all the time under 16 ms frame rate. We analyzed those parts where the frame rate exceeded recommendation and came into conclusion that responsiveness is most of the time at good level and only in some occasions slowness could be seen. So we were confident enough to put these applications into hands of real users.

It should be noted that all applications were tested by the students several times. However, the final test was the workshop with a school class. The school class was from secondary school in Helsinki, Finland and fitted well the target group. Based on the findings from the workshop we can state that the interaction with mobile soundscape mixer application was smooth and well appreciated [19]. This verified that our design and development process resulted into successful applications.

7 Conclusions

In this paper we have proposed how to develop an easy to use and smoothly working Android application to increase user interaction by developing soundscapes from building blocks stored in audio digital asset management system.

We have successfully designed and implemented four and tested three different sound mixer applications. Based on the performance testing we anticipated that the interaction with mobile mixer application is smooth. This was verified later from findings from the first workshop with real users. However, the outcomes that we draw are preliminary and require further testing. Next we will organize a similar workshop for adults in Museum of Technology. In addition, to prove that mobile mixer applications are versatile we will test applications in an outdoor city planning event together with our People's Smart Sculpture Project partner.

Acknowledgements. We thank media engineering and information technology students who participated into design and development of Android applications. In addition, we thank Aura Neuvonen and her sound design students for soundscape support. The work is co-funded by EU, Creative Europe Programme, The People's Smart Sculpture project (http://smartsculpture.eu). We thank the project partners for inspiring discussions.

References

1. Väänänen-Vainio-Mattila, K., Suhonen, K., Laaksonen, J., Kildal, J., Tahiroğlu, K.: User experience and usage scenarios of audio-tactile interaction with virtual objects in a physical environment. In: Proceedings of the 6th International Conference on Designing Pleasurable Products and Interfaces, DPPI 2013, pp. 67–76. ACM, New York (2013)
2. Heller, F., Krämer, A., Borchers, J.: Simplifying orientation measurement for mobile audio augmented reality applications. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI 2014, pp. 615–624. ACM, New York (2014)
3. Schaefer, R.M.: The Soundscape: Our Sonic Environment and the Tuning of the World. Inner Traditions International/Destiny Books, Rochester (1993)
4. Jacucci, G., Oulasvirta, A., Salovaara, A.: Active construction of experience through mobile media: a field study with implications for recording and sharing. Pers. Ubiquit. Comput. **11** (4), 215–234 (2007). http://www.cs.helsinki.fi/u/oulasvir/scipubs/jacucci-etal_revised_print.pdf
5. Drechsler, A., Raffaseder, H., Rubisch, B.: Klang.Reise: new scientific approaches through an artistic soundscape installation? In: Proceedings of the 7th Audio Mostly Conference: A Conference on Interaction with Sound, AM 2012, pp. 44–46. New York, USA (2012)
6. Casu, M., Koutsomichalis, M., Valle, A.: Imaginary soundscapes: the SoDA project. In: Proceedings of the 9th Audio Mostly: A Conference on Interaction With Sound, AM 2014, Article No. 5. ACM, New York (2014)
7. Salo, K., Giova, D., Mikkonen, T.: Backend infrastructure supporting audio augmented reality and storytelling. In: Yamamoto, S. (ed.) HIMI 2016. LNCS, vol. 9735, pp. 325–335. Springer, Heidelberg (2016). doi:10.1007/978-3-319-40397-7_31
8. Leinonen, T.: Designing Learning Tools. Methodological Insights. Aalto University, Helsinki (2012). ISBN 978-952-60-0032-9

9. Snyder, C.: Paper Prototyping: The Fast and Easy Way to Design and Refine User Interfaces. Morgan Kaufmann, Burlington (2003). ISBN 9780080513508
10. Säde, S., Battarbee, K.: The third dimension in paper prototypes. In: Branaghan, R. (ed.) Design by People for People: Essays on Usability, pp. 203–212, Chicago, USA (2001)
11. Stappers, P.J.: Creative connections: user, designer, context, and tools. Pers. Ubiquit. Comput. **10**(2), 95–100 (2006)
12. Mullins, C.: Responsive, mobile app, mobile first: untangling the UX design web in practical experience. In: Proceedings of the 33rd Annual International Conference on the Design of Communication (SIGDOC 2015), Article 22, 6 p. ACM, New York (2015)
13. Smartphone OS Market Share, 2015 Q2. http://www.idc.com/prodserv/smartphone-os-market-share.jsp
14. Top 8 Mobile & Tablet Operating Systems in Finland from Oct 2014 to Oct 2015. http://gs.statcounter.com/#mobile+tablet-os-FI-monthly-201410-201510
15. Salo, K., Shakya, U., Damena, M.: Device agnostic CASS client. In: Marcus, A. (ed.) DUXU 2014, Part II. LNCS, vol. 8518, pp. 334–345. Springer, Heidelberg (2014)
16. Yang, S., Yan, D., Rountev, A.: Testing for poor responsiveness in android applications. In: Proceedings of 1st International Workshop on the Engineering of Mobile-Enabled Systems (MOBS), 25 May, pp. 1–6. IEEE, San Francisco (2013)
17. Liu, Y., Xu, C., Cheung, S.-C.: Characterizing and detecting performance bugs for smartphone applications. In: Proceedings of the 36th International Conference on Software Engineering, ICSE 2014, pp. 1013–1024. ACM, New York (2014)
18. Testing Display Performance. http://developer.android.com/training/testing/performance.html
19. Salo, K., Bauters, M., Mikkonen, T.: Mobile Soundscape Mixer Augmenting Reality in Museum. Manuscript submitted for publication

http://www.springer.com/978-3-319-44214-3

Mobile Web and Intelligent Information Systems
13th International Conference, MobiWIS 2016, Vienna, Austria, August 22-24, 2016, Proceedings
Younas, M.; Awan, I.; Krywinska, N.; Strauss, C.; Do, T. van (Eds.)
2016, XIV, 444 p. 147 illus., Softcover
ISBN: 978-3-319-44214-3