

Music Learning Tools For Android Devices

M. Luna Herruzo, Ana M. Barbancho, Isabel Barbancho, Lorenzo J. Tardón
mlunaht@ic.uma.es, abp@ic.uma.es,ibp@ic.uma.es, lorenzo@ic.uma.es
Universidad de Málaga, ATIC Research Group, ETSI Telecomunicación,
Dpt. Ingeniería de Comunicaciones, Campus Teatinos, 29071 Málaga, Spain

Abstract—In this paper, a musical learning application for mobile devices is presented. The main objective is to design and develop an application capable of offering exercises to practice and improve a selection of music skills, to users interested in music learning and training. The selected music skills are rhythm, melodic dictation and singing. The application includes an audio signal analysis system implemented making use of the Goertzel algorithm which is employed in singing exercises to check if the user sings the right musical note. This application also includes a graphical interface to represent musical symbols. A set of tests were conducted to check the usefulness of the application as musical learning tool. A group of users with different music knowledge have tested the system and reported to have found it effective, easy and accessible.

I. INTRODUCTION

Nowadays, the ways of teaching and learning are changing. There is a vast number of learning games, from the ones used to teach children basic geometric forms to the ones for learning a new language. On the other hand, mobile devices are no longer just wireless mobile phones. Now, they are rather tiny portable computers which provide entertainment and information [3]. The application market that only existed for computers a few years ago has avised for portable devices. Inside this new large market, we can find learning applications specifically, this paper focuses on music learning applications. Music has always been a topic that has attracted the attention of the human being. That is why developing an application about music learning seems a good choice. A music learning application can deal with several music topics depending on the objective and level of musical knowledge of the user. Likewise, an mobile application can implement different exercises, like learning musical notes or the timbre of different instruments, rhythm exercises, music dictation, singing exercises or musical theory exercises.

In this paper, we present the application *Music Learning* focused on user's with low musical knowledge. This kind of users wants to improve their music skills by straightforward exercises. The description of the application includes the processing algorithms for audio signal analysis, the database employed, and the type of exercises and tests implemented.

Music Learning has been implemented for Android devices which is an operating system designed primarily for touch-screen mobile devices such as smart-phones and tablets [3]. The App can be used by users of different ages and it has included three kinds of musical exercises: rhythm exercises, dictation exercises and singing exercises.

The structure of the application *Music Learning* is explained in Section II. In Section III, the three auxiliary systems developed for the application are explained, namely: the musical symbol display system, the audio signal analysis system and the exercises database. Next, in Section IV, the

three exercises of the application are explained. The different tests that were made to improve the application and reach for possible errors are presented in Section V Finally, in section VI, the conclusions about the development of this application are drawn.

II. SYSTEM DESCRIPTION

The structure of the *Music Learning* application is shown in Fig. 1. In this figure it can be seen that the developed application offers three exercises, making use of the auxiliary systems: Musical Symbols, Exercises Database and Audio Signal Analysis. The audio signal analysis system is used in the singing exercises.

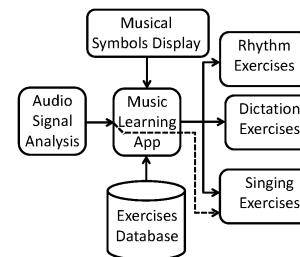


Fig. 1. Structure of the application *Music Learning*.

III. AUXILIARY SYSTEMS

Besides the programming and development of the main application structure and the three main types of exercise of the App, it was necessary to implement a specific system to display musical symbols, to be used by the three kind of exercises. On the other hand, as the App has a singing exercise, it required a system of analysis of the audio signal, to allow the application show the differences between correct and incorrect sounds. The system also needed a database in which all the exercises are stored.

A. Musical Symbol Display System

For the display system, several object classes have been implemented according to the kind of musical symbol to show. A main interface class named Symbols establishes the variables and methods which are common to the classes that represent different musical symbols [1]. These classes are: Note, Rest, Clef, Time Signature, Tick and Interval. The main characteristic of these classes are a variable named *gap*, that has been designed to scale all the symbols to the display size and resolution of different devices. First, the height of a staff is calculated, which should be the 20% of the *Displays Height* (1). Then, the variable *gap* represents the space between lines of a staff, which means it is the 25% of the staff's height (2).

$$Staff\ Height = 20\% \cdot Display\ Height \quad (1)$$

$$gap = 25\% \cdot Staff\ height \quad (2)$$

B. Audio Signal Analysis System

The audio signal analysis system is required for the Singing exercises. This system will be able to notify if a certain frequency component appears with significant power in the analysed audio signal. This system has been implemented by using the Goertzel Algorithm because of its computational simplicity in the detection of single tones. Actually, this is the algorithm used in the recognition of DTMF tones produced by the buttons pushed on a telephone keypad [7].

The Goertzel Algorithm can analyse selected frequency components from a discrete signal. For covering the full spectrum, the Goertzel algorithm has higher complexity than Fast Fourier Transform (FFT) algorithms, but for computing a small number of selected frequency components, it is more computationally efficient. The simple structure of the Goertzel algorithm makes it well suited to small processors and embedded applications, though not limited to these.

As the FFT, the frequency resolution of Goertzel Algorithm depends on the sampling rate and the number of samples selected [4]. In our case, the system must work with a frequency range from 196 Hz to 784 Hz, which corresponds to the notes G3 to G5. This note range has been selected because it is a normal singing range for amateurs. So, the minimum frequency resolution (Δf), the sampling rate (SR) and the number of samples (N) must be selected.

The sampling rate needed for this system comes from the Nyquist Theorem (3),

$$SR = 2 \cdot f_{max} = 2 \cdot 784Hz = 1568Hz \cong 1600Hz \quad (3)$$

The minimal frequency resolution must be 11Hz (4), that is the minimum frequency separation between the note range G3 - G5, that corresponds with the difference in frequency between the lowest notes, that is G3 and G#3:

$$\Delta f = f(G_{\#3}) - f(G_3) = 207.65Hz - 196Hz \cong 11Hz \quad (4)$$

The number of samples (N), depends on the sampling rate (3) and the frequency resolution (4). It is calculated as follows:

$$N = \frac{SR}{\Delta f} = 145.5 \cong 146 \text{ samples} \quad (5)$$

Due to Android characteristics, this theoretical configuration cannot be used. Android allows a specified number of sampling rates, and only guarantees a smooth running in every device of the sampling rate 44100Hz. After some running tests, the final configuration of this Audio Signal Analysis System is: $SR = 44100Hz$, $N = 4096 \text{ samples}$ and $\Delta f \cong 10Hz$.

C. Exercises Database

The Database of the application *Music Learning* has been implemented in SQLite. SQLite is a software library that implements a self-contained, server-less, zero-configuration, transactional SQL database engine. SQLite is in the public domain [2]. Android incorporates all the tools needed to create and manage a SQLite database.

This Database is made up of three tables, one for every kind of musical exercise the application has. That is, a table to save rhythm exercises, a table for dictation exercises and another table for singing exercises. The data is saved in the database in the format of a letter and a number. The letter reveals the kind

of symbol, namely note, rest, time signature and clef, while the number conveys the information to identify the specific symbol within each symbol type. For the dictation exercises, the notes are represented following the American Notation, which names musical notes with the letters A,B,C,D,E,F,G.

IV. MUSICAL LEARNING EXERCISES

In this section, the three types of exercises are detailed as well as the application user interface.

A. Rhythm Exercises

There are two main components of music: rhythm and melody. Rhythm is associated with time, and melody is, on the other hand, associated with the pitch or the key. Although rhythm and melody could be studied separately, in music, they generally interact together and influence each other in a complex way. Among these two properties, rhythm is considered, by many academics as the most important one. The ancient Greeks held that, without rhythm, melody lacked form and strength [6].

So, within the musical learning context, it is essential to develop the rhythm sense of the students. In a composition, the rhythm refers to the notes' length and the rhythmic pattern the notes are organized into. The listener perceives a temporal succession of elements arranged as a structure.

The rhythm exercises of this application consist of a staff with a number of musical figures, which can be notes or rests. Note that in this type of exercise only the figure's length matters. The user must beat the rhythm of the exercise by pressing the circular button on the screen.

These exercises have a metronome help available as well. A metronome is a system that produces regular beat, which help the user to keep a steady tempo as the user performs a musical piece.

This Rhythm block has a configuration screen and a main screen. The configuration screen allows the user to select an exercise from a list, the speed of the exercise and the use of the metronome help.

As shown in Fig. 2, the main screen of the rhythm exercise has a staff with some musical figures on it, and a button on the right side. As the user presses this button to beat the rhythm, the application records the time in which every keystroke is made. When the number of beats has reached the total musical notes of the exercise, every saved beat is compared against the reference. If a beat is close, within a certain temporal margin, to the reference beat, it is correct. Otherwise, the beat is wrong. The application shows the outcome to the user by means of a result window.



Fig. 2. Main screen of the Rhythm exercise.

B. Dictation Exercise

There exist two kinds of dictations: rhythmical dictation and melodic dictation. A melodic dictation exercise consists of listening to the reproduction of a melody and writing its

transcription down to score. Comprehending the combination of pitch, rhythm and meter that occur in any melody requires a basic understanding and practical application of music theory. So this kind of exercise helps to develop the student musical sense, since they have to determine which note they are hearing without any visual help.

For this application, a simple melodic dictation exercise has been chosen. The device plays a melody without instrumental accompaniment and a simple rhythm. The objective is to distinguish between different consecutive notes.

This Dictation block has a configuration screen and a main screen. The configuration screen allows the user to select an exercise from a list. The main screen of the Dictation Exercise is shown in Fig. 3. At the top of the screen, the user will find a row of figures and rests, an staff and the buttons to play the melody and check the dictation. The user must move and place each note in the right place on the staff. With the check button, the user can check if the figures are placed correctly. Every time the user touches and moves a figure, the application is saving its coordinates in the screen. Then, when the check button is pressed these saved coordinates are compared with a list of positions. These positions are the coordinates of every possible note in the staff. If a position is correct, a green tick appears over the corresponding figure. After pressing the check button three times, the application shows the final outcome to the user in a result window.



Fig. 3. Main Screen of the Dictation Exercise.

C. Singing Exercises

Singing properly implies that the main frequency components of the produced sound follow a musical composition.

Turning to music learning, singing intervals is an interesting singing exercise. An interval is the difference between two pitches [5]. This kind of singing exercise teaches the student to tune their own voice to a certain note, just by hearing another note which is separated a specific interval of the first one. This application allows the user to train with six different intervals. There are two major thirds, which correspond to the easiest exercises, two perfect fourths and two perfect fifths [8].

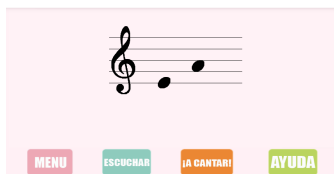


Fig. 4. Main Screen of the Singing Exercise.

This singing block, like the others, has a configuration screen and a main screen. The configuration screen allows the user to select an exercise from a list. In the main screen of the singing exercise, Fig. 4, the user can find a graphic

representation of the interval previously chosen from the list and two main buttons. The first button plays the first note of the interval giving an audible reference for the exercise. The second main button is pressed when the user is ready to sing the second note of the interval. This button starts the audio analysis process. The application records and saves the audio data in buffers of 4096 samples. When a whole buffer is filled, these 4096 samples are processed with the Goertzel algorithm explained in subsection IV.B. For every 4096 samples, this process returns one Goertzel magnitude. After completed and processed 40 consecutive buffers (3.71 recording seconds), the system determines by majority voting the singing note. This recording length has been selected to avoid mistakes due to the beginning or ending of the singing note.

The application shows the final assessment of the user's performance in a new window. A help button in the main screen is provided that plays the second note of the interval, in case the user can not sing the note correctly without help.

V. TESTS AND RESULTS

In order to check the correctness and usability of the application system and user tests have been carried out.

A. System Tests

These tests are done to evaluate the performance and tune the behaviour of the system to provide the users with a valuable music learning experience. The tests carried out for the rhythm exercises were performed to establish the time margins in which a beat made by the user is considered correct. The duration of the time margin depends on the speed of the exercise. After several tests, the optimal configuration found established the margin as $\pm 20\%$ of the configured beat speed. That is, for a speed of 60 bpm, the margins, before and after the exact beat instant will be of 200 milliseconds.

The aim of the tests carried out for the dictation exercises was to check the performance of the system of the figure position in the staff. The coordinates of the five lines of the staff are known. For every figure, the lower Y coordinate is calculated and saved, this Y coordinate is compared against the position of the lines of the staff to establish the name of the note. For every possible note in the staff, graphic limits have been calculated.

The running tests for the singing exercises were the most complex, because many parameters needed to be tested. First, tests to confirm the setting of the configuration parameters of the recording Android class were made. From these tests, it was discovered that the best sampling rate option is 44100 Hz, and that the recording buffer of the recording Android class is of 4096 samples. Then, several tests were carried out to find an optimal threshold to decide if a target frequency is in the recorded audio data with sufficient energy, making use of the Goertzel magnitudes. The threshold was established in 10^{12} . Fig. 5 shows the Goertzel magnitudes of a test where the target frequency was 329 Hz (E4) and the sung frequency was 311 Hz (D#4). The difference between these two frequencies is 18 Hz, which is over the system frequency resolution, which is 10 Hz. As it can be seen in the image, most of the Goertzel magnitudes don't exceed the threshold, which means the target frequency is not present in the recorded audio data (the sung note is incorrect), as expected.

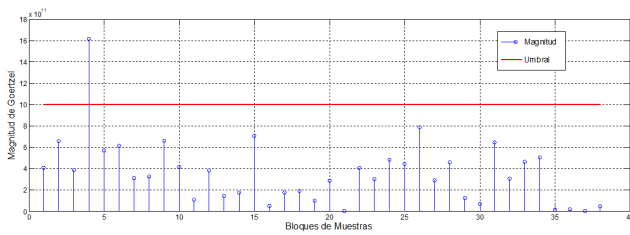


Fig. 5. Goertzel Magnitudes of E4 - D#4 test.

B. User Tests

In order to assess the usability and the user's experience provided by the application, user tests were carried out. Different devices, with different Android OS versions, and especially, different users were involved in the test.

To perform this task, every user was given a manual and a questionnaire. The questions asked in the questionnaire were:

- Q1: Is the application easy and accessible?
- Q2: Do you find the application useful?
- Q3: How many exercises have you completed correctly?
- Q4: Are the exercises easy or difficult?
- Q5: Which aspects of the application should be improved?
- Q6: Age
- Q7: Degree of musical knowledge

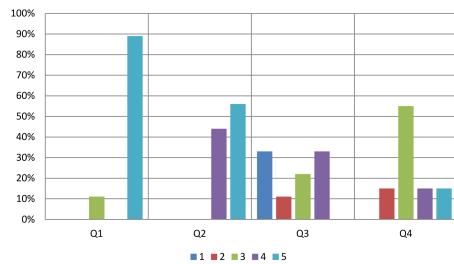


Fig. 6. Users answers to questions 1 to 4.

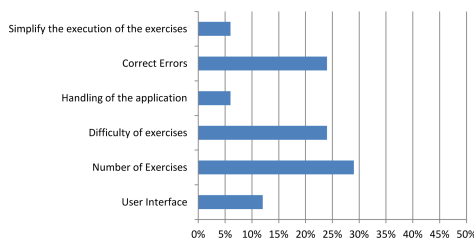


Fig. 7. Users answers to question 5.

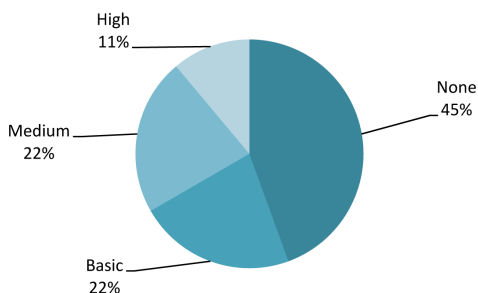


Fig. 8. Users musical knowledge.

In Fig. 6, the results of the users to questions 1 to 4 can be seen. These questions had to be answered with a number from

1 to 5, where 1 means difficult/lowest/worst punctuation and 5 means easy/highest/the best. From questions 1 and 2 we can conclude that the users find the application useful and easy to handle. The 56% of the users of this test had a basic or higher musical knowledge, as shown in Fig. 8 which illustrates the answers to question 7, and the 66% of them are between 11 and 40 years old.

On the other hand the results found for questions 3 and 4 seem reasonable, as the users have a well varied range of musical knowledge.

Regarding question 5, Fig. 7, a number of possible improvements were suggested to the users. They selected as the most important aspects to improve the number of exercises the application offers and their difficulty.

The conclusions of this user tests have been very positive, because the users find the application effective, easy and accessible. Due to the diversity of levels of musical knowledge of the users, the conclusion about the difficulty of the exercises is that it would be convenient to increase the number of exercises, on the one hand by adding easier exercises for users without musical training, and on the other hand by adding more difficult and challenging exercises for users with a superior musical knowledge. Also, the users, found the interface user-friendly and the procedure of the exercises simple and intuitive.

VI. CONCLUSIONS

An application for interactive music learning that can be used in Android devices has been developed.

The *Music Learning* application has been implemented with three different kinds of exercises that allow the users to develop and improve their musical skills and knowledge. An audio signal analysis system that can differentiate between frequencies with a maximum difference of 10 Hz has been developed. This application implements its own musical symbols display system, needed to show the exercises to the users. A database implemented with SQLite has been designed and integrated inside the application to store the exercises data.

In order to evaluate the correct performance of the application *Musical learning* different devices and different Android OS versions have been tested and external users have evaluated the usability of the application with positive results.

ACKNOWLEDGEMENTS

This work has been funded by the Spanish Ministerio de Economía y Competitividad, Project No. TIN2013-47276-C6-2-R and by the Junta de Andalucía under Project No. P11-TIC-7154. This work has been done at the Campus de Excelencia Internacional Andalucía Tech, Universidad de Málaga.

REFERENCES

- [1] Android Developers, <http://developer.android.com>, last retrieved 02/02/2015, Official web site, 2015.
- [2] SQLite, <http://www.sqlite.org>, last retrieved 02/02/2015, Official web site, 2015.
- [3] J. Ribas-Lequerica, *Desarrollo de aplicaciones para Android*, Anaya, 2011.
- [4] A.V. Oppenheim, A.S. Willsky, *Signals and Systems*, Prentice Hall, 1996.
- [5] H. Grabner, *Teoría General de la Música*, AKAL, 2001.
- [6] G. Toussaint, *The Geometry of Musical Rhythm*, CRC Press, 2013.
- [7] E. Jacobsen, R. Lyons, "The Sliding DFT," *IEEE Signal Processing Magazine*, vol. 20, March 2003, pp. 74 - 80.
- [8] D. De Pedro, *Teoría completa de la Música vol. 1 y vol. 2 (Edición revisada)*, Real Musical-Carisch España, 2014.