

Ear Trainer for Guitar Chords – An Android Application for Recognition of Guitar Chords

Matei-Alexandru Cioată, Adrian Iftene

“Alexandru Ioan Cuza” University, Faculty of Computer Science

General Berthelot, 16, Iasi, Romania

E-mail: matei.cioata1@info.uaic.ro, adiftene@info.uaic.ro

Abstract. Mastering a musical instrument is a task that requires very much time, patience, and motivation. Many people discover their passion for music late. Even though they begin to follow a path to become great musicians or they just search for the perfect hobby, it is hard to maintain the desire to learn this skill due to the low amount of free time. Interactivity is a very important element when learning music and therefore, this paper presents a method to help new guitarists who don't have the possibility to search for music teachers. In this paper, we will see details about *Ear Trainer for Guitar Chords* an Android application created to help train the ear of those who learn to play the guitar. The experiments we did showed that the application was appreciated by beginner, intermediate or advanced musicians.

Keywords: chord recognition, note recognition, discrete Fourier transform, frequency, hamming window

DOI: 10.37789/ijusi.2020.13.2.3

1. Introduction

Technology is evolving very fast and the number of people who use it as a helping hand is increasing. These days, many different domains use the software in order to automate tasks and music is not an exception. Tools for beginner, intermediate and advanced musicians are getting developed every day: learning applications, games, virtual tuners, metronomes, etc. The ease of use, the saved time, and the low price are just some of the advantages obtained while using virtual tools.

A simple application cannot replace a music teacher, but the goal is to become as close as possible, and therefore, interactivity is a must. It is very important to have it integrated into music projects because the users can receive feedback which they can use in order to improve. The learning process is entertaining and the motivation is maintained. Skills like *dexterity*,

music theory, and recognition by ear can be trained by this type of application.

2. Guitar Note and Chord Recognition

An interesting way to bring interactivity into music projects is to implement a method of note recognition. Thus, the device which runs the app has the capacity to listen and process the sound produced by the user's instrument (Pilhofer and Day, 2007).

There are multiple ways to develop such functionality, but the first one that comes to mind is a method that involves artificial intelligence and neural networks. However, it is not practical because of the difficulty to find enough good quality data. There are many types of guitars in the world, each one with its specific tone. Also, the sound is propagated differently depending on the environment (room size, guitarist position, microphone quality, background noise, etc.). It would be almost impossible for a person to record a single note thousands of times, with hundreds of different guitars in hundreds of different environments. At the same time, a huge amount of computational power is necessary in order to train a neural network of this complexity. Therefore, the Fourier Transform is a possible solution to this problem.

3. Similar Work

A considerable amount of popular applications that use a method of signal processing can be found in papers (Pilhofer and Day, 2007), (Suits, 1998), (Simion and Trausan-Matu, 2012), (Stere and Trausan-Matu, 2017), (Deerga Rao, 2018), (Potinteu and Stefanut, 2018) and (Santo Pietro, 2020). At the moment, a note recognition algorithm that ensures 100% accuracy doesn't exist because, most of the time, background noise is present or the quality of the microphone is weak. One way to counter this problem is to capture unchanged signals directly from the instrument (for example, by connecting an electric guitar to a computer, no microphone is involved).

*Rocksmith*⁴ is a video game developed by Ubisoft (available only on PC and consoles) which can teach the users to play a vast collection of rock,

⁴ <https://rocksmith.ubisoft.com/rocksmith/en-us/home/>

reggae, and metal songs and it uses the method described above. Even though the accuracy is still not 100%, it comes close. The only problem with this signal capturing method is that it is not practical on mobile devices.

*Yousician*⁵ is one of the most popular music learning platforms. It uses note recognition algorithms for guitar, piano, bass, ukulele, and even voice. The signal is captured through the microphone. Therefore, Yousician is available to multiple platforms (Windows, Android, iOS, etc.), but the user experience can be highly affected by the quality of the input sound device. Both *Yousician* and *Rocksmith* offer real-time feedback to the player and a corresponding score.

*Perfect Ear*⁶ is a mobile application that focuses on music theory, ear training, and rhythm. It uses a note recognition algorithm only for interval singing using the voice. The signal capturing is done with the microphone.

By analyzing the above applications, we can see that the use of the microphone assures compatibility with mobile devices while sacrificing accuracy. Also, the method used by *Rocksmith* for signal capturing requires a special cable (the higher the accuracy, the higher the cost). Therefore, the option of microphone usage is present in the video game for those who prefer not to purchase the cable.

There are many other similar applications: Fender Play, GarageBand, AmpliTube, Songsterr Guitar Tabs & Chords, BandLab, Ultimate Guitar, Guitar Tuna, and Chordify.

4. Signal Processing

4.1 Short Introduction to Musical Notes

A musical note defines the pitch (and duration) of a single sound and has a corresponding frequency (Santo Pietro, 2020). In music theory (Pilhofer and Day, 2007), there are seven different notes: A, B, C, D, E, F, and G. Together, they form the diatonic scale.

In Figure 1, the diatonic scale having the root in C can be observed. As the scale progresses to the right, the pitch of the sound raises. The first C note is

⁵ <https://yousician.com/>

⁶ <https://play.google.com/store/apps/details?id=com.evilduck.musiciankit&hl=ro>

one octave lower than the last. The pitch difference between two notes is called an interval. There are many types of intervals, but the most important ones are the semitone (S) and the tone (T - two times bigger than the semitone). If we break the tones in the diatonic scale into semitones, we obtain the chromatic scale.

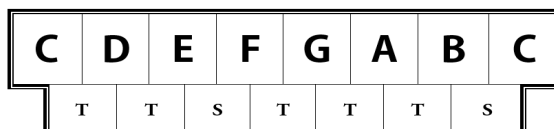


Figure 1. C diatonic scale

In Figure 2, the chromatic scale having the root in C can be observed. The ‘#’ symbol raises a note with one semitone.

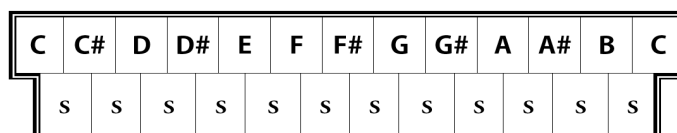


Figure 2. C chromatic scale

A standard guitar with 20 frets can produce a total of 45 different notes. Every one of these is actually part of the chromatic scale, only from different octaves. For example, a guitar can produce a C# in the third octave and a C# in the fourth octave. So, the notes C, C#, D, D#, E, F, F#, G, G#, A, A#, and B are all the notes that can be found on a normal guitar.

A chord is basically a multitude of notes played simultaneously. There are multiple types of chords, but there is no need to detail them in this paper.

4.2 Computing the Frequencies for Musical Notes

There are mathematical formulas that define relationships between notes (Smith, 2007), (Harris, 1978). As mentioned before, each note has a corresponding frequency (Suits, 1998). An octave is a complete cycle of notes from the chromatic scale. It is usually noted with an index (for example, A_4). Let $Fr(n)$ be the frequency of the note n . The following equation is known:

$$Fr(n_x) = 2 \times Fr(n_{x-1}) \quad (1)$$

This means that the frequency of a note is twice the frequency of that same note from the previous octave. The standard frequency of A_4 has been established to be 440 Hz (but can be slightly different for music variation). Using the formula from above, we also know that:

$$Fr(A_5) = 2 \times Fr(A_4) = 4 \times Fr(A_3) = \dots \quad (2)$$

However, a relation between different notes has to be established (not just A). The chromatic scale has 12 different notes, each pair being separated by one semitone. Therefore, they are located at equal distances on a logarithmic scale:

$$Fr(n) = Fr(N) \times (2^{\frac{1}{12}})^d \quad (3)$$

In this equation, n is the note with the unknown frequency, N is the note with the known frequency and d is the number of semitones between n and N (can be negative when n is lower than N).

4.3 Short Introduction to Sound Physics

Sound is a vibration, a periodic oscillation of the air that propagates and is interpreted by the human and animal ears. Graphically, we can represent it as a sinusoid. The frequency (speed of sound) measures the number of complete cycles of vibration executed in the time unit.

In the Figure 3, we can see a sound vibration with the frequency of 3 Hz (3 complete cycles in a second).

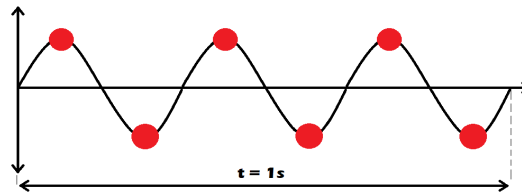


Figure 3. Sound sinusoid example

When we communicate through computers or record sound, the device needs a microphone that takes the information and forwards it. Given the fact that sound is a continuous signal (Deergha Rao, 2018), a method of discretizing it is necessary. The transformation of a continuous signal into a

discrete signal is called **sampling**. The **sampling rate** (in Hz) measures the number of times information is extracted per second. For example, in Figure 3, the sampling rate is 6 Hz (information is extracted in the red dots, 6 times per second).

For good quality recordings, the sampling rate needs to be high enough so as not to lose information, but not too high (in this case, the files are too big and filled with useless information). The **Nyquist-Shannon theorem** (Pintilie, 2020) states that the sampling rate needs to be at least equal to the double of the maximum frequency from the recording.

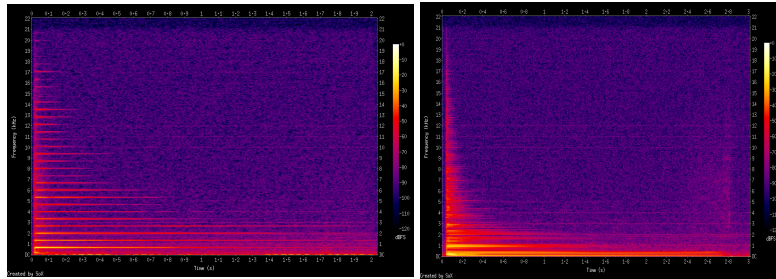


Figure 4. Spectrograms of notes E5 (left) and G2 (right)

The notion of harmonic also needs to be defined in this paper. When a note propagates in the atmosphere, its frequency is not the only one present. Vibrations with frequencies equal to multiples of the fundamental one also propagate and they are called harmonics. For example, when playing the note A_4 on the guitar, we produce more sounds (440 Hz – the main one, highest energy -, 880 Hz, 1320 Hz, etc.). The higher the order of the harmonic, the lower its energy is.

4.4 The Discrete Fourier Transform

Note recognition is equivalent to identifying the present frequencies. We saw in Figure 3 that sound is represented as vibration with time on the OX axis, amplitude, and pressure on the OY axis. Therefore, in order to detect frequencies, it would be much easier if the original sinusoid would be transformed in such a way that the time domain would become a frequency domain.

In Figure 5, we can see that the frequency of 1 kHz is present and dominant (has the biggest amplitude). Also, its harmonics (2 kHz and 3 kHz) are also

detected, but with a lower amplitude like described in section 4.3. The existence of these harmonics ensures that the fundamental 1 kHz vibration is indeed present. This transformation of the domain from time to frequency is called the Fourier Transform.

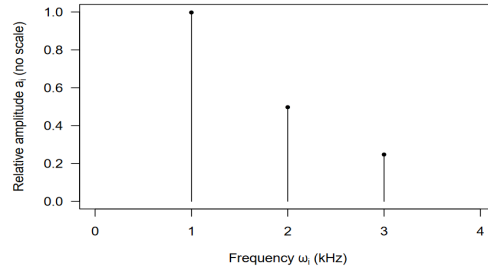


Figure 5. Frequency spectrum example (Sueur, 2020)

In the previous section, the fact that the signal has to be discretized has been mentioned. Therefore the method that needs to be used is called **Discrete Fourier Transform**.

Definition 1 (Smith, 2007). Let x be a signal (sound signal in the case of this paper). Its Discrete Fourier Transform is defined as such:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n) \times e^{-i\omega_k t_n}, k = 0, 1, 2, \dots, N - 1 \quad (4)$$

where:

N - total number of samples;

$x(t_n)$ - amplitude of the initial signal at time t_n ;

t_n - the time when the n -th sample is found in the recording (equal with $n \times T$);

T - the time interval at which sampling takes place;

f_s - the sampling rate of the recording (equal to $1/T$);

i - the imaginary unit ($i^2 = -1$);

ω_k - the frequency (angular, measured in rad/second instead of Hz) for which the amplitude is calculated (equal to $2\pi \times F_k$, where F_k is the frequency in Hz);

$X(\omega_k)$ - the spectrum (Wolfe, 2005) of signal x at frequency ω_k ; it is a complex number. To find the amplitude, the complex number module has to be calculated ($\sqrt{a^2 + b^2}$).

Given the fact that the recording has N samples, a maximum of N frequencies can be detected. This is the reason why k has values between 0 and $N-1$. If the frequencies for which amplitudes need to be calculated are not known, they must be sampled too. However, in this paper, a simpler formula for the Discrete Fourier Transform will be used because at every moment, the frequencies, the time, and the total number of samples are known:

$$X(\omega_k) = \sum_{n=0}^{N-1} x(t_n) \times [\cos(2\pi \times F_k \times t_n) - i \times \sin(2\pi \times F_k \times t_n)], k = 0, 1, 2, \dots, N-1 \quad (5)$$

4.5 Optimizations for Note Recognition

The algorithm that starts from equation (5) has the complexity of $O(N^2)$ because for every frequency sample, the total amplitude is calculated based on every time sample. A faster transformation exists and it is called the Fast Fourier Transform ($O(\log N \times N)$), but in the proposed application, an even more efficient method was implemented. On a standard guitar, only 45 different notes can be produced, and therefore, the frequencies that need to be identified are known. The time complexity of the transformation becomes $O(45 \times N)$. Not only this method is very fast, but it also eliminates some of the background noises. It is unlikely that a noise frequency is the same as a note frequency. Other methods of optimizing the note recognition are: recognizing harmonics in such a way that they can confirm the existence of the fundamental frequency, applying a Hamming Window (Harris, 1978) on the sampled signal instead of a Rectangle Window in order to avoid spectral leaks (noises produced by the computer):

$$W(n) = 0.54 - 0.46 \times \cos(2\pi n/N - 1) \quad (6)$$

where:

n is the index of the current time sample;

$W(n)$ is the Hamming Window value for the current time sample.

5. Proposed Application

5.1 The Interface and the Architectural Models

Ear Trainer for Guitar Chords is an Android application that starts with the main menu from which the three main modules can be accessed: (1) the learning component represented by diagrams, practice, info&tips, (2) the evaluation module represented by a chord recognition game and (3) the settings. The application contains a single **activity**⁷ displaying the UI components of the main menu. Each option in the menu has a corresponding **fragment** (a portion of the graphical interface) (Sirișteanu and Iftene, 2013).

In the **Ear Trainer for Guitar Chords**, the fragments play the role of sub-activities because they have their own behaviors and interfaces occupying the entire screen. Even though they have their own lifecycle, they are part of an activity scope and they can communicate with other fragments and activities. For this project, the use of 5 fragments was chosen instead of 5 activities because of performance and good practices.

When selecting an option in the main menu, a fragment transaction is created (the replacement of the menu interface with the option's interface) and added to the top of a stack. This stack is part of a class named **FragmentTransaction** containing a series of operations for manipulating fragments. After the end of the fragment lifecycle, the transaction will be eliminated from the stack. Therefore, **Ear Trainer for Guitar Chords** will have a maximum of 1 transaction(s) in the stack at any time because the return to the main menu is necessary for selecting another option. The interface that interacts with the fragments of an activity is called **FragmentManager**. Until now, the separation of the graphical interface elements was discussed, but defining the behavior only in fragments and activities is a bad practice because it does not assure the separation of concerns. Therefore, the usage of architectural models is required. For this project, MVVM (Model-View-ViewModel - names after its components) was used.

⁷ <https://developer.android.com/reference/android/app/Activity#Fragments>

5.2 Database

The database is simple, being composed of a single table named “chords”, with the following structure: cid (chord ID), root (root note of the chord), type (the type of the chord - major, minor, etc.), sound_file_name (the name of the sound file found in the **res/raw** folder), diagram_file_name (the name of the chord diagram image found in the **res/drawable** folder), notes (a string with the component notes of the chord).

5.3 Application Modules

The Play, Practice, Diagrams, and Settings modules have their own fragment representing the **View** component of the MVVM architecture. Also, each of them has a **ViewModel** and a corresponding **ViewModel Factory**. The Info&Tips menu only has a fragment, because its only job is to display messages from the strings.xml file without operating with chords or other data.



Figure 6. Practice menu (left), Diagrams menu (middle), the G chord selected (right)

5.3.1 Practice Module

Allows the user to listen to all the chords in the database. The graphical interface is composed of a **GridView** with 4 rows and 2 columns, the cell from down-right representing the selected chord type (major, minor or 7th). The rest of the buttons (A, B, C, D, E, F, G) are the chord names. This menu

helps train the ear and prepares the user for the chord recognition game. The differences between the different types of chords can be identified (see Figure 6 - left).



Figure 7. Instructions scrollable list (left), selected instruction (middle), Settings menu (right)

5.3.2 Diagrams Module

Allows the user to see all the chord diagrams from the database (see Figure 6 - middle). The elements are placed in a scrollable list. When selecting an element, the corresponding diagram will appear on the screen (see Figure 6 right). For placing elements in the list, the **RecyclerView** class was used.

The users can use diagrams in combination with the Practice module to check if they reproduce the chords correctly and to exercise until their tries sound like the notes produced by the phone.

5.3.3 Info & Tips Module

Contains instructions:

- application role explanation;
- the utility of each module;
- hardware and usage recommendations;
- restrictions and requirements (guitar tuning);
- diagram reading tutorial;

- instructions for the game and for the rest of the modules;
- ideas for future updates.

The modules presented until this point compose the learning module. They contain information that prepares the user not only for the chord recognition game but for becoming a good musician. Like in the Diagrams module, the instructions in Info & Tips are placed in a list with RecyclerView and they appear on the screen when pressed (see Figure 7 - left and middle).

5.3.4 Settings Module

The settings menu (Figure 7 - right) allows the user to choose the chord type used in the game (a single type or all of them for a bigger challenge) and in the Practice module. A game controller can be selected (guitar or phone). In the first case, the user records his answer and the device analyzes the sound, offering a point if the correct frequencies are identified. If the phone is the selected controller, the answers are going to be offered through buttons. This setting is useful in the following cases: the user does not have a guitar yet, he/she uses the application only for improving the ear, the device has limited resources and the recognition algorithm does not work at its full potential.

The settings are saved in **SharedPreferences** and can be accessed anytime, anywhere from the application. These preferences resemble a dictionary (*key-value* pairs) that can be modified anytime. When the application is run, the settings are loaded from **SharedPreferences**.

5.3.5 The Chord Recognition Game

Represents the evaluation module. It is composed of more rounds and it lasts until the user makes a mistake. At the beginning of each round, a random chord is chosen (with the type selected in settings) (see Figure 8 - left). The player has the right to listen to that chord 3 times per round, to see 7 hints generated on the screen, and to use 3 helpers (once used, they are not available in the corresponding game).

Button roles are explained in the Info & Tips menu. If the phone is selected as a controller, the 7 hints are pressable (used to answer). The right answer will always be one of those. If the guitar is selected, the answer is recorded by pressing the red button from the up-right. At the start of the recording, 3 threads begin their execution besides the main one (see Figure 8 - right):

- the first one handles the sound recording in bytes form; it transfers the

information further;

- the second one is responsible for analyzing the information received from the previous thread. The chosen chord for the current round is known, therefore the identified notes will be compared with the ones from the database (table “chords”, column “notes”). When the chord is identified, a notification is sent to the main thread and the player receives a point. The round ends instantly;
- the third one is a timer of 5 seconds which stops the first two threads at the end and notifies the main thread that the game is over. At that moment, the score and the right answer are displayed;
- the main thread displays elements to the screen, listens to buttons and responds to the notifications correspondingly.

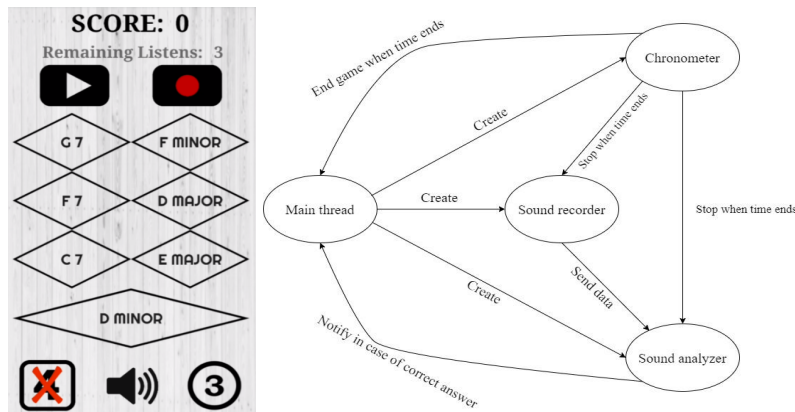


Figure 8. Chord recognition game (left), Thread communication for sound recording and analysis (right)

A score is saved in **SharedPreferences**. Although the classes that handle sound recording and analysis are used just by the chord recognition game, they are part of the **Model** component of the MVVM architecture because they contain business logic. When the application is run, an initialization algorithm that calculates all guitar note frequencies executes. A **MutableMap** is obtained (key-value map, with the key being the name of the note, and the value is the frequency of the note). This algorithm helps with the identification of the chords. For this project, royalty-free images from

Pixabay⁸ (backgrounds for menu, settings, practice, diagrams, info & tips and for the game) were used. The diagrams for chords were downloaded from this page⁹. The used fonts in the application are Indie Flower¹⁰ and Righteous¹¹ from Google Fonts.

6 Feedback and Usability Tests

The best method for testing such an application is to offer it to users in order to receive feedback (Pinzariu and Iftene, 2016), (Chițaniuc and Iftene, 2018). This way, the potential impact on the music world can be observed. In order to write this chapter, a Google Form was made for 9 volunteers to answer some questions about their personal opinion on **Ear Trainer for Guitar Chords**. The guitarists that offered to help with the experiment (ages between 18 and 23 years old) have different experiences in music and different opinions about the usefulness of this application. Two volunteers have no guitar experience while the others have between 10 months and 10 years of experience. A summary with all the opinions of the volunteers and statistics regarding different application components can be found below. The form begins by asking some questions about the learning modules, especially helpful for beginners.

Question 1: *How do you think that the learning module helps beginners? If you are part of this category, do you think that it can support your evolution?*

- “It’s very useful to have all the chords in the same place when you want to learn and to monitor the progress. If you learn a chord wrongly you can remember it like this (for example, an E7 instead of an E), and the Practice module helps you notice.” (10 years’ experience);
- “I consider that this module helps a lot, especially beginners: they can learn chords by having the diagrams in front of them and comparing their sounds with the ones in the database.” (1 year experience);
- “Beginners can learn fundamental chords, but the application does not

⁸ <https://pixabay.com/ro/>

⁹ <https://midnightmusic.com.au/2013/08/free-guitar-chord-image-library/>

¹⁰ <https://fonts.google.com/specimen/Indie+Flower>

¹¹ <https://fonts.google.com/specimen/Righteous>

offer real photos or videos of each chord. They can be easily misled.” (7 years’ experience);

- “I like that the application helps me to learn to read a diagram. Also, I found out how correct chords sound. The Info&Tips module helps me understand how the application works and how to use the other modules to my advantage.” (No experience).

Question 2: *Just for intermediates and advanced: do you consider that the learning module can help you at this level? If yes, how?*

- “Personally, it helps me with the listening part and to advance in a chord type that I don’t know very well yet.” (2 years’ experience);
- “Just the diagrams, because I don’t know all the chords by heart.” (10 years’ experience);
- “I consider that it is useful even for advanced guitarists because it helps to maintain the musical ear. The constant exercise is mandatory, no matter the level of experience.” (6 years’ experience).

Question 3: *In your opinion, what is the most helpful learning module?*

There were different opinions at this question between the “Diagrams” and the “Practice” modules. The volunteers that already know the chords tend to choose “Practice” because it helps them with ear training, while the others prefer the diagrams.

Question 4: *In your opinion, are there useless functionalities in the learning module?*

The only problem reported here was that some instructions from “Info & Tips” are hard to follow. The following questions will refer to the chord recognition game.

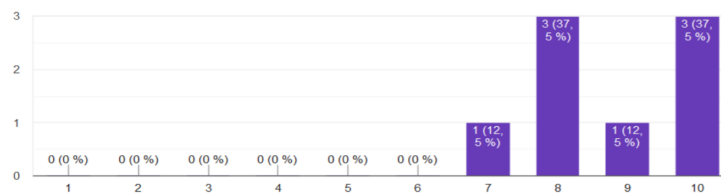


Figure 9. The grades for the chord recognition algorithm

Question 5: *Please offer a grade for the chord recognition algorithm if you played the game with the guitar on a phone with a 4+ cores processor (Figure 9).*

Question 6: *How fun do you think the game is?*

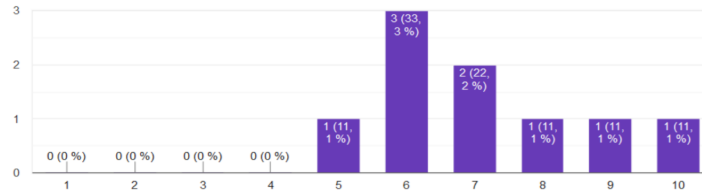


Figure 10. Grades for entertainment

Question 7: *How do you think that this game helps you with your experience?*

- “It helps me to recognize a lot of chords by ear. Also, I learned some chord shapes.” (1 year experience);
- “For a beginner, the game can be a little hard, but after I learn the chords using the other modules, the game is a very good way to test myself.” (no experience);
- “It offers me a better understanding of musical chords; also, since I use this application, I can say that I noticed an improvement of my musical ear.” (10 months experience);

Question 8: *How difficult is the game in your opinion (from very easy to very hard)?*

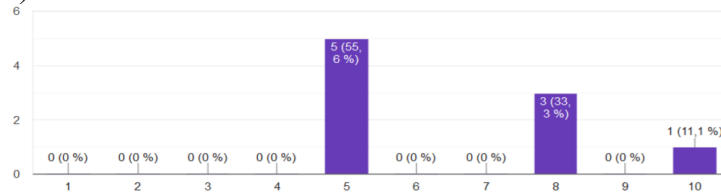


Figure 11. Grades for game difficulty

Question 9: *What is the most captivating element of the game?*

- “The fact that you have only 5 seconds to reproduce the given chord. The idea that I have a short time to recognize and reproduce the sound puts me in alert and I feel like I am in a competition with myself.” (10 months experience);

- “The chord recognition and the recording of the guitar.” (7 years’ experience);
- “The High Score determines you to obtain more points, therefore you are motivated to learn the chords.” (no experience);

The following questions are about the application as a whole.

Question 10: *Have you encountered bugs while testing the application?*

- An error signaled by a single volunteer: after a period of 15-20 minutes, the sound did not work anymore. There was indeed a problem caused by all the MediaPlayer instances. They must be erased manually from memory. Freeing them at the fragment closure was omitted. This bug has been fixed;
- Another error reported by a single volunteer is the fact that some texts from “Info & Tips” went out of the screen. After some repositioning, the bug has been fixed;
- 3 out of 9 volunteers reported small problems for the chord recognition algorithm in the presence of background noises (some correct chords were not recognized);
- The other 4 volunteers did not report any errors.

Question 11: *How intuitive is the interface in your opinion?*

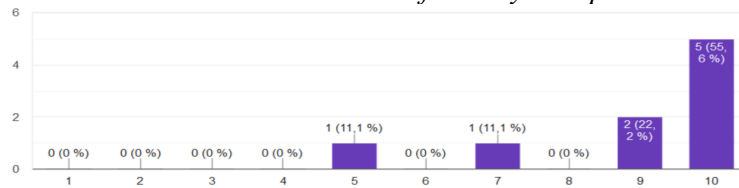


Figure 12. Grades for interface intuitiveness

Question 12: *How accessible is the application in your opinion, having in mind that it can be used totally offline?*

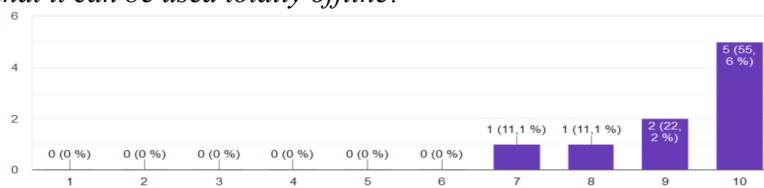


Figure 13. Grades for application accessibility

Question 13: *Would you use the application for yourself? Would you recommend it to family and friends? Why?*

- “Yes. I would recommend it because it has all the necessary learning instruments for beginners and intermediates.” (2 years’ experience);
- “I intend to use it when I begin to learn how to play guitar because it seems like a very good helper. I will recommend it every time I have the chance.” (no experience);
- “Yes, I used the app and I admit: it helped me very much. Yes, I would recommend it to all of those who want to start playing guitar because it is easy to use, the explanations are accessible to everybody and it is interactive.” (10 months experience);
- “I wouldn’t use it very much because I already know the chords that I need. However, the game is interesting and it can help me even at this level. I recommended it to some beginner friends.” (10 years’ experience);

Question 14: *Finally, how many stars would you give the application on Google Play?*

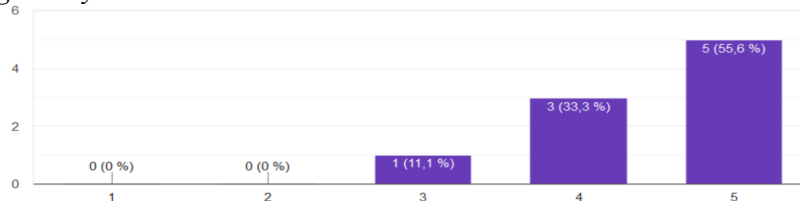


Figure 14. Stars on Google Play

After the usability tests, the problems reported at question 10 were remediated. The volunteers were asked to give a detailed description of the situations that presented errors of the chord recognition algorithm. Based on this information, possible optimizations were researched. Some of them have been applied. The beginner volunteers asked for more instructions (for example, tutorials for reading diagrams). This request was satisfied in the “*Info & Tips*” module.

In conclusion, Ear Trainer for Guitar Chords had some good reviews from the volunteers. Differences between the opinions of beginners, intermediates and advanced users can be observed. However, everybody found the application to be useful. The learning module is destined especially for beginners. This may be the reason that caused those volunteers to declare that

they use it intensively for chord understanding, while the game can be a bit difficult for them. Intermediate and advanced users think that the game has medium difficulty. It helps them more because the majority already know the information offered by the learning modules. The volunteers were pleased by the functionality of the chord recognition algorithm. Only small problems that appeared in the presence of background noises were reported.

7. Conclusions

Electrical devices are very important instruments used in learning processes. Music is a difficult and wide domain that requires a lot of time and devotion. This is why the advantages of electrical devices are very important, since they bring extra convenience, time, and diversity of learning resources. In our opinion, there are many quality resources, free and paid, that develop musical skills and can be found online. However, there were some shortcomings that made me come up with the idea of **Ear Trainer for Guitar Chords**. We consider that Ear Trainer for Guitar Chords offers this kind of help to beginners in an interactive and easy to understand way. Even if intermediate and advanced guitarists already know the chords, they can always test their musical ear with the chord recognition game, evolving even more. This is why we believe that the application could represent an impact for all the guitarists.

Ear Trainer for Guitar Chords is at the beginning of its life. Lots of improvements and functionalities of different complexities destined to all kinds of musicians can be implemented.

After some discussions with the volunteers who tested the Android application, we came up with a list of improvements: (1) the first priority is to add some theory elements. Beginners wish to understand the definition, structure of chords and of other musical elements (for example, scales); (2) there are ideas of improving some graphical elements: reorganizations of the “Diagrams” menu and putting the elements in a table instead of a list, deleting some instructions and adding others, etc.; (3) support for vocal commands, since the pressing of buttons with the guitar in the hand is uncomfortable. There were attempts to solve this problem with offline libraries, but without success. We consider that the lack of necessity to connect to the internet is a defining feature of Ear Trainer for Guitar Chords. Therefore, in a future update, users will be able to decide if they want to use an online Speech to

Text service or if they want to remain offline; (4) new types of chords; (5) images and videos containing instructions of hand positions for every chord; (6) other interactive games with other musical elements (for example, recognition of pentatonic scales).

Acknowledgments. This work was supported by project REVERT (taRgeted thErapy for adVanced colorEctal canceR paTients), Grant Agreement number: 848098, H2020-SC1-BHC-2018-2020/H2020-SC1-2019-Two-Stage-RTD.

References

- Chițaniuc, M., Iftene, A. (2018) GeoAR-An Augmented Reality Application to Learn Geography. In *Romanian Journal of Human-Computer Interaction*, 11(2), 93-108.
- Deergha Rao, K. (2018) Signals and Systems. In *Springer International Publishing*, 271-320.
- Harris, F. J. (1978) On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. In *Proceedings of the IEEE*, 66(1), <http://web.mit.edu/xiphmont/Public/windows.pdf>
- Pilhofer, M., Day, H. (2007) Music Theory for Dummies. In *Wiley Publishing, Inc.*, Part II & Part III.
- Pintilie, L. N. (2020) Teorema lui Nyquist – Shannon - Demonstratie; Evidentierea conceptului de ”timp de esantionare” sau ”frecventa de esantionare”. https://epe.utcluj.ro/SCTR/Indicatii/Teorema_lui_Nyquist_Shannon.pdf
- Pinzariu, M.N., Iftene, A. (2016) Sphero - Multiplayer Augmented Game (SMAUG). In *International Conference on Human-Computer Interaction*, 8-9 September 2016, Iasi, Romania, 46-49.
- Potinteu, I. C., Stefanut, T. (2018) Smart Watch-based Gesture Recognition to Control a Music Player. In *Proceedings of RoCHI 2018*, 84-90.
- Santo Pietro, D. (2020) Introduction to sound. *Khan Academy*. <https://www.khanacademy.org/science/apphysics-1/ap-mechanical-waves-and-sound/introduction-to-sound-waves-ap>
- Simion, A., Trausan-Matu, S. (2012) Automatic musical genre detection. In *Romanian Journal of Human-Computer Interaction 2012*, 5(1), 19-36.
- Sirițeanu, A., Iftene, A. (2013) MeetYou - Social Networking on Android. In *Proceedings of 11th International Conference RoEduNet IEEE: Networking in Education and Research*, 17-19 January 2013, Sinaia, Romania, 228-233.
- Smith, J.O. (2007) *Mathematics of the Discrete Fourier Transform (DFT) with Audio Applications*. Second Edition, <http://cirma.stanford.edu/jos/mdft/>
- Sueur, J. (2020) A very short introduction to sound analysis for those who like elephant trumpet calls or other wildlife sound. <https://cran.r-project.org/web/packages/seewave/vignettes/seewaveanalysis.pdf>

- Suits, B. H. (1998) Physics of music - notes. <https://pages.mtu.edu/suits/Physicsofmusic.html>
- Stere, C.C., Trăușan-Matu, S. (2017) Generation of musical accompaniment for a poem, using artificial intelligence techniques. In *Revista Romana de Interactiune Om-Calculator* 10 (3), 250-270.
- Wolfe, J. (2005) What is a Sound Spectrum? <https://newt.phys.unsw.edu.au/jw/sound.spectrum.htm>.