A Mobile Application to Train Chord-playing on Keyboard Instrument

By

Timmy Ong

12886

A project dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information & Communication Technology)

DECEMBER 2012

Universiti Teknologi PETRONAS
Bandar Seri Iskandar,
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

A Mobile Application to Train Chord-playing on Keyboard Instrument

by

Timmy Ong

12886

A project dissertation submitted to the
Information & Communication Technology Programme
Universiti Teknologi PETRONAS
in partial fulfillment of the requirements for the
BACHELOR OF TECHNOLOGY (Hons)
(INFORMATION & COMMUNICATION TECHNOLOGY)

Approved by,

____

(Assoc. Prof. Dr. Dayang Rohaya Bt. Awang Rambli)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK
November 2012

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this report, that the original work is my own except as specified in the references and acknowledgments, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

(TIMMY ONG)

ABSTRACT

Music played by a band is mostly annotated in chord charts, unlike complete notations for classical music. Beginner keyboardists playing in a band often struggle to catch up with only chord charts as reference because a single chord can be played in multiple ways on a keyboard instrument. To improve, practice is needed. However, the problems with practice chord-playing are the lack of training material, swayed practice goals as well as the immobility of instrument. Hence, the project has the objectives to: explore the multi-touch capabilities of tablet devices for music training, develop a structured and goal-oriented virtual keyboard mobile application to achieve chord-playing training; conduct a user perception study for the developed application on potential users; and conduct a usability study on the developed application. The application is targeted at beginner keyboardists to improve their existing skills in playing basic chords single-handedly, through a two-octave keyboard interface. The application was successfully developed with 4 prototype iterations. The user perception study showed that beginner keyboardists generally accept using mobile technology to train their chord-playing skills. The usability study showed that the developed application is quite a usable application.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor for her flexibility and openness in helping me complete this project. Without her trust and relaxed supervision, this project would have been a big burden on top of academic and extra-curricular workload.

Many thanks to my dear coursemates doing their projects on a similar platform for the mutual support and encouragement. The support, technically, academically and morally, was essential towards the completion of the project.

Special thanks go to Chris Cernoch, the creator of the ChordTrainer.com, for inspiring the idea of the project on the Android platform, and also for the permission to use the chord calculation algorithm, which saved me invaluable time on the prototype development for other aspects of the project.

Credits to my fellow orchestra members, for being cooperative survey respondents and helping me complete the surveys in an unimaginably short time. Jason Ngu from the orchestra gets my extra appreciation for lending me his personal tablet device for the prototype development and testing.

Last but not least, thanks to my family for being understanding and giving me complete freedom to pursue my academics.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS AND NOMENCLATURES

FYP1	Final Year Project 1
FYP2	Final Year Project 2
SDK	Software Development Kit
IDE	Integrated Development Environment
ADT	Android Developer Tools
OS	Operating system
App	Application
IT	Information technology
AVD	Android Virtual Device
API	Application Programming Interface
UTP	Universiti Teknologi PETRONAS

CHAPTER 1
INTRODUCTION

1.1 Background Study

Music played by a contemporary musical group (band) is mostly annotated in chord charts, unlike complete notations on classical music score sheets. A chord chart is a musical notation sheet containing a series of chord names that make up the structure of a song. A chord chart does not specify the exact notes or rhythm to play at a specific time in a song; it is only a general guide, following which the player can improvise.

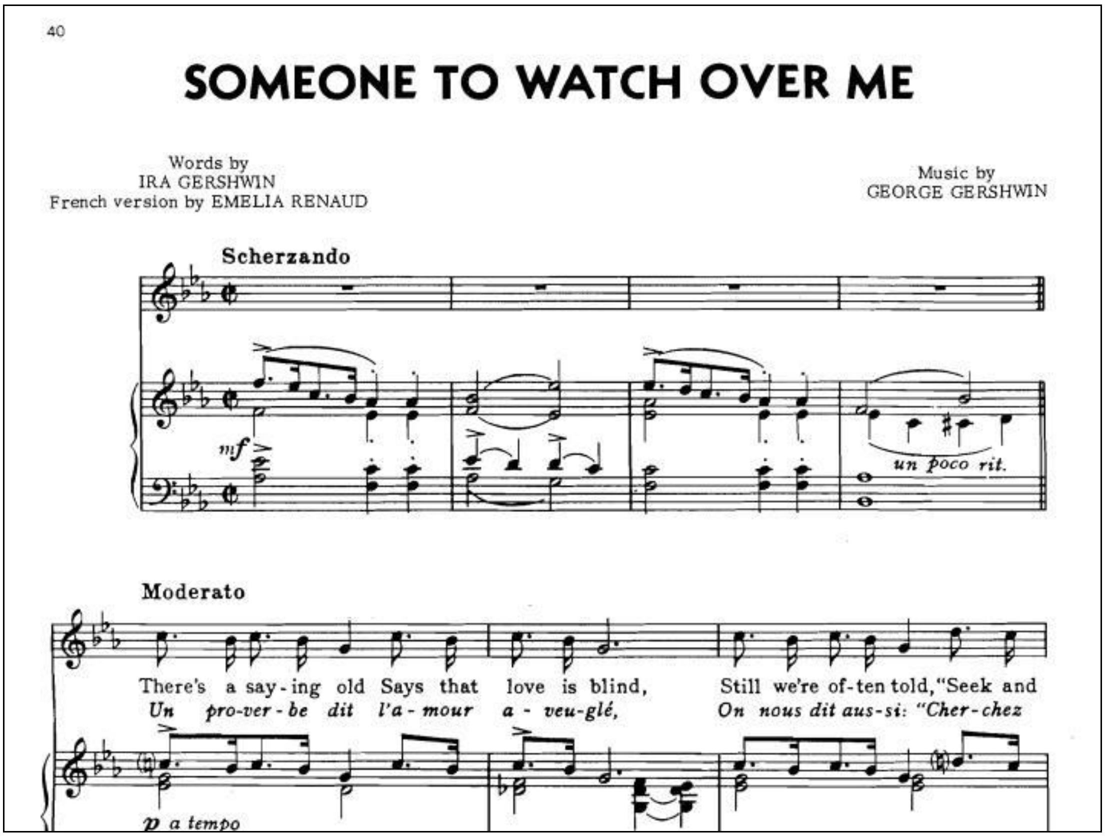

40
SOMEONE TO WATCH OVER ME
Words by
IRA GERSHWIN
French version by EMELIA RENAUD
Music by
GEORGE GERSHWIN
Scherzando
mf
un poco rit.
Moderato
There's a say-ing old Says that love is blind, Still we're of-ten told, "Seek and
Un pro-ver-be dit l'a-mour a - veu-glé, On nous dit aus-si: "Cher-chez
p a tempo

Figure 1 A piece of contemporary music in notation sheet with lyrics

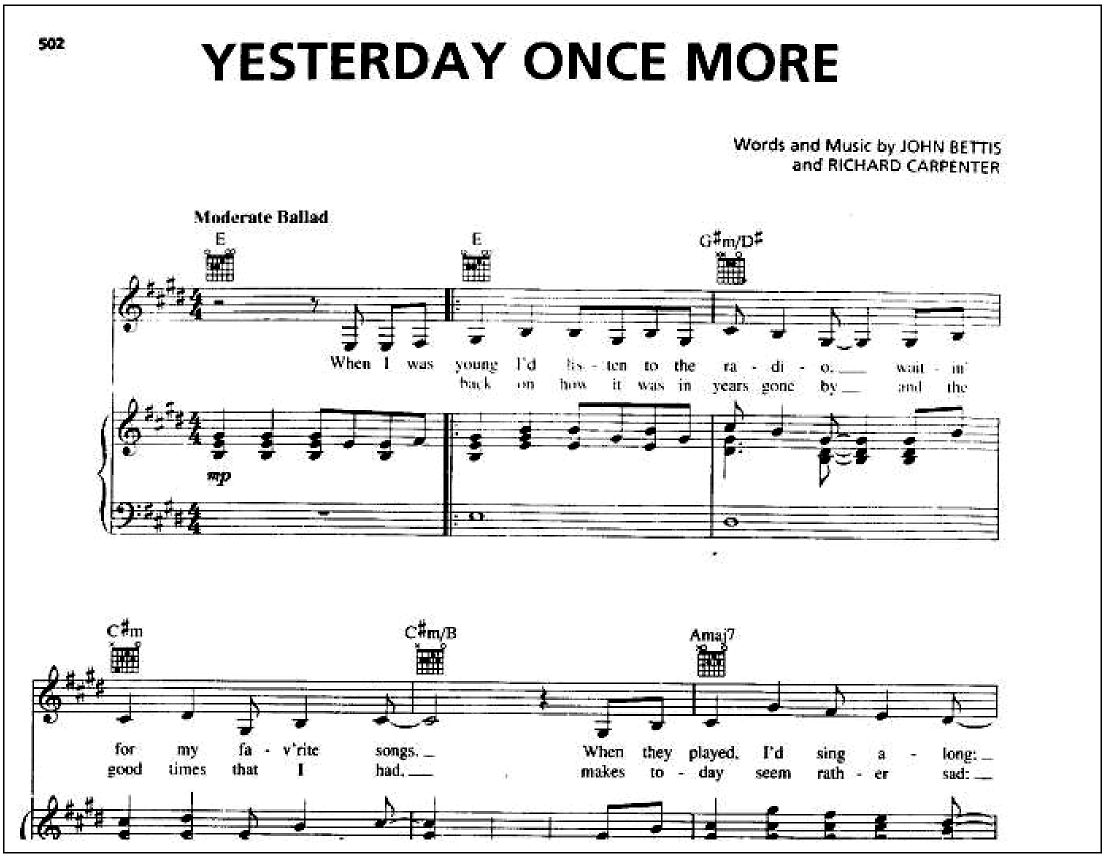
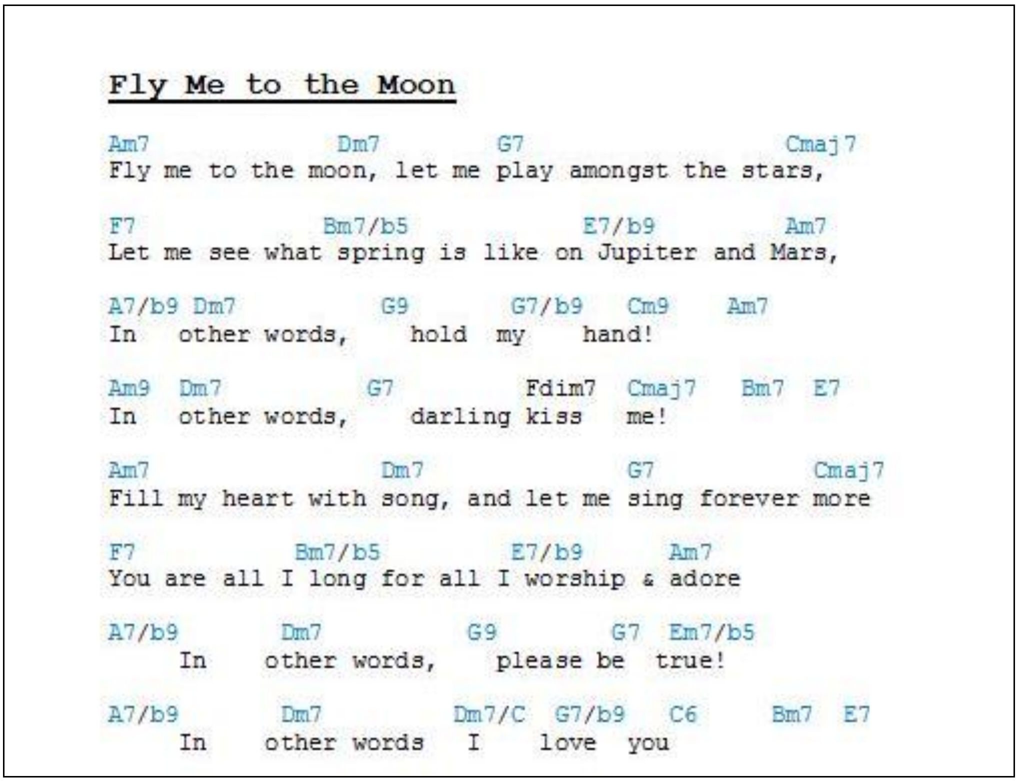

502
YESTERDAY ONCE MORE
Words and Music by JOHN BETTIS
and RICHARD CARPENTER
Moderate Ballad
E
E
G♯m/D♯
When I was young I'd lis - ten to the ra - di - o, wait - in'
back on how it was in years gone by and the
mp
C♯m
C♯m/B
Amaj7
for my fa - v'rite songs.
good times that I had.
When they played, I'd sing a - long:
makes to - day seem rath - er sad:

Figure 2 A piece of contemporary music with musical notation and chord chart



Fly Me to the Moon

```
Am7             Dm7        G7                  Cmaj7
Fly me to the moon, let me play amongst the stars,

F7             Bm7/b5            E7/b9         Am7
Let me see what spring is like on Jupiter and Mars,

A7/b9 Dm7          G9       G7/b9   Cm9    Am7
In   other words,    hold  my    hand!

Am9  Dm7          G7         Fdim7  Cmaj7   Bm7  E7
In   other words,    darling kiss   me!

Am7                Dm7              G7           Cmaj7
Fill my heart with song, and let me sing forever more

F7           Bm7/b5         E7/b9      Am7
You are all I long for all I worship & adore

A7/b9       Dm7          G9        G7  Em7/b5
     In    other words,    please be  true!

A7/b9       Dm7         Dm7/C  G7/b9   C6     Bm7  E7
     In    other words   I    love  you
```

Figure 3 A piece of contemporary music with lyrics and chord chart only

Figure 1 through 3 show several formats that a contemporary music piece may appear in. However, often times, for efficiency purposes, only the lyrics and chord chart are given (see Figure 3), and the musician is to improvise according to the chord chart given as the vocalist sings through the song.

A chord is a set of harmonious sounds containing three or more musical notes. When presented on a chord chart for keyboard, the only ‘rule’ is that the set of three or more notes that make up the chord should be played. As a single chord can be played in multiple formations (known as chord inversions) on a keyboard instrument, the position and hence combination of the notes that makes up the chord is entirely up to the comfort and creativity of the player.

To play a chord, keyboardists have to identify the individual notes that make up the chord, wherever on a keyboard, before striking it; unlike chords on the guitar, which usually works by memorizing placements of hand patterns on the guitar fret board.

To play a song, keyboardists have to correctly play all the chords presented on the chord chart of the song, transitioning from one chord to the next, following a specific tempo.

1.2 Problem Statement

1.2.1 Problem Identification

Beginner keyboardists playing in a band often find it difficult to catch up with only chord charts as reference because a single chord can be played in multiple formations on a keyboard instrument. To improve, practice is needed. However, there are a few problems when it comes to training chord-playing.

1) Training Material

To facilitate practice, training material such as fake books is needed. A fake book is actually a loose collection of musical lead sheets containing minimal information such as the lyrics and chord chart, and perhaps the melody line, meant to help a musician quickly learn new songs (“Fake Book”, n.d., para. 1). Formal and published material for a fake book is hard to find; and although with the advances of technology, informal material can be easily found online, fake books are not meant for novices. A musician generally needs to have high familiarity with chords and sheet music to work with fake books (“Fake Book”, para. 3). A beginner keyboardist may not have thorough familiarity with chords, and hence may face problems while practicing using fake books.

2) Practice Goal

One problem with practicing using fake books and other songbook-like materials is that the goal of the practice session often becomes learning up new songs instead of attacking the root of familiarizing oneself with chords. Even though the ultimate goal of training on chord-playing is to actually play new songs, the fundamental target of the training is very important. A beginner keyboardist should be focusing on knowing the chords first, before starting work on songs using fake books.

3) Portability of Instrument

Due to the size and portability of a keyboard instrument, keyboardists are unable to practise when they are on the go.

1.2.2 Significance of Project

The project is significant in that its product, a mobile application, addresses the problems outlined above.

1) Training Material

The application provides material for the user to practise by outputting chords. It is not in the format of fake book material, but a simpler and more adapted format for beginner keyboardists.

2) Practice Goal

The application has a specific practice target over existing virtual keyboard applications that only gives the user a keyboard interface; the application is only aimed at improving the chord-playing skills of beginner keyboardists.

3) Portability and Accessibility

The application can provide significant practice time for beginner keyboardists when they are on the go. Being a mobile application, it is more accessible than a physical keyboard, as mobile devices are easier to carry around than an actual keyboard.

1.3 Objectives and Scope of Study

1.3.1 Objectives of Study

This project has the following objectives:

- To explore the multi-touch capabilities of tablet devices for music training
- To develop a structured and goal-oriented virtual keyboard mobile application to achieve chord-playing training for beginner keyboardists
- To conduct a user perception study for the developed application on potential users
- To conduct a usability study on the developed application

1.3.2 Scope of Study and Limitations

The product of the project is a mobile application meant for landscape orientation on tablet devices only, as the screen sizes of smartphones are too small to simulate playing on a real keyboard. The tablet device should be at least 10.1 inches in size, and running on the Android operating system (OS). The reason for this shall be elaborated in Chapter 2: Literature Review, Section 2.3.

The application is aimed at beginner keyboardists. The application is not meant to teach keyboard skills and techniques to in-training keyboardists. It is to be used as an interactive self-learning tool to train the existing skills of beginner keyboardists who already have basics in keyboard, including the skill to identify chords and read chord charts.

The application is not meant to be a replacement for an actual keyboard. Practice on an actual keyboard is still important; the application only provides goal-oriented practice time for beginner keyboardists when the real keyboard is not available.

Translation of skill improvement gained from the application into an actual keyboard is encouraged.

The application includes only 4 basic chord types, which are major, minor, augmented and diminished chords, each of which comprises only 3 notes. Hence, the minimum and maximum number of keys to be pressed is 3 keys, on a two-octave (25 keys) keyboard interface, which gives a comfortable range for beginner keyboardists to work out chords. The application only trains a single hand at a time.

1.4 Relevancy of Project

The project is relevant in the current mobile application market as there are no existing applications that provide goal-oriented music training. Currently available applications only provide a virtual instrument for users to play and practise, and do not have specific training objectives. Furthermore, applications that do work on improving users' chord skills are passive, and do not allow for hands-on practice.

The project is also relevant in the music field because the way to improve one's music-playing skill is to perform a lot of focused practice. The application allows users to be able to practise while they are on the go.

1.5 Feasibility of the Project within the Scope and Time Frame

The project was feasible within the scope and time frame. The Android platform is open-source, and development for the Android platform utilizes skills whose basics the student possesses. The student has basics in mobile development, gained from industrial internship, as well as basics in the Java programming language and object-oriented programming. Covering only four basic chord types in the application, which was developed within a time frame of four months, with two months for research on existing applications as well as development techniques, the project was feasible.

CHAPTER 2
LITERATURE REVIEW

To play structured music takes more than just laying hands on a musical instrument and making uncoordinated sounds out of it. Instead, it is to make specific and intended sounds using a musical instrument. It takes practice to be familiar with the musical instrument and to have good hand-eye, and even ear, coordination with it. Then, taking it a step further, to make structured music, it takes some understanding of music theory.

An understanding of music theory, with high familiarity to their musical instrument, combined with physical and mental agility ("Sight-Reading Techniques", n.d., para. 2), allows a musician to easily keep up with an ensemble, even though having only minimal contact with a music piece or song prior to an ensemble rehearsal or performance.

2.1 Music Education Methods

As the project targets training chord-playing for beginner keyboardists, some music education methods were examineds.

The Suzuki method is a music education method pioneered by Japanese violinist Shin'inchi Suzuki ("Suzuki Method", n.d., Philosophy section). Its philosophy is that given the right learning and nurturing environment, all children can pick up skills on a musical instrument as naturally as they pick up their mother tongue language ("The Suzuki Method", n.d., para. 1). The Suzuki method encourages music learning for children to be tailored closely related to language learning, which include techniques such as saturating a child in a musical environment (as compared to being immersed in a community speaking their mother tongue), starting a child's music education early (as compared to speech learning for babies), and learning music by ear before notations (as compared to a child learning to speak before learning to read) ("The

Suzuki Method", para. 3) ("Suzuki Method", Philosophy section). The Suzuki method is already a widely acclaimed method used worldwide. However, its focus lies mainly in music education and training for children. Also, the Suzuki method does not have a particular method in training chord-playing.

The Dalcroze method was developed by Swiss musician and educator Emile Jacques-Dalcroze. This method has three focuses: aural training and the use of solfege (musical tones in a given musical scale or "key"), eurhythmics (the teaching of rhythm and musical structure and expressions through movements) ("Eurhythmics", n.d., para. 1), and improvisation (Farber & Thomsen, 2011, paras. 1 & 8)**.** The part of the method that is relevant to the project would be the improvisation focus, to be elaborated in the next section.

2.2 Music Training Aspects

Also a part of other music education methods such as the Orff Schulwerk method and Kodaly method, as well as jazz music training, improvisation is a music training aspect closely related to chord-playing training. Improvisation is the creation of musical compositions spontaneously (Hamlin, n.d., para. 1) following chord changes in a song ("Musical Improvisation", n.d., para. 1), during a rehearsal or performance. Chord-playing, especially on a keyboard, is highly similar to improvisation in the sense that it is entirely up to the keyboardists as to how a chord, or a series of chords, should be played or varied in a song. While improvisation creatively play to chord changes in a song (coming up with impromptu tunes), chord-playing creatively play the chord changes (using different chord variations).

Another music training aspect to be looked into for training chord-playing would be sight-reading. Sight-reading is to "play unfamiliar music from scores" ("Sight-Reading Techniques", n.d., para. 1). All forms of playing at sight are a subset of sight-reading. Hence, even though chord names denoted in chord charts do not specify notes like music scores, reading and playing chords require the same physical and mental agility that sight-reading requires. Rapid reading—and hence playing—requires familiarity of various musical structures including intervals, keys, scales, chords, melodic structure, as well as on-sight analysis ("Sight-Reading Techniques", para. 4). Practice is needed to achieve the aforementioned familiarity of musical structures that ultimately leads to immediate recognition of chords.

Both training on improvisation and sight-reading ultimately increase the responsiveness of the musician; and this responsiveness is a music skill required in chord-playing. The project capitalizes on responsiveness training for chord-playing.

2.3 Android Platform

The application runs on the Android operating system (OS) based on the following clarification. Android is a free and open-source OS developed by Google Inc., meant for mobile devices such as smartphones and tablets. This OS is used by a number of mobile device manufacturers from HTC to Samsung. Andy Rubin, Senior Vice President of Mobile at Google stated that mobile devices running on Android has reached over 300 million devices as of February 2012, with over 850 000 activations per day (Rubin, 2012). There are a few Android versions and the most current one is Android 4.1, which is also known as Jelly Bean.

Version	Codename	API Level	Distribution
1.5	Cupcake	3	0.1%
1.6	Donut	4	0.3%
2.1	Éclair	7	3.1%
2.2	Froyo	8	12%
2.3 - 2.3.2	Gingerbread	9	0.3%
2.3.3 - 2.3.7		10	53.9%
3.1	Honeycomb	12	0.4%
3.2		13	1.4%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.8%
4.1	Jelly Bean	16	2.7%

Table 1 Android versions usage distribution

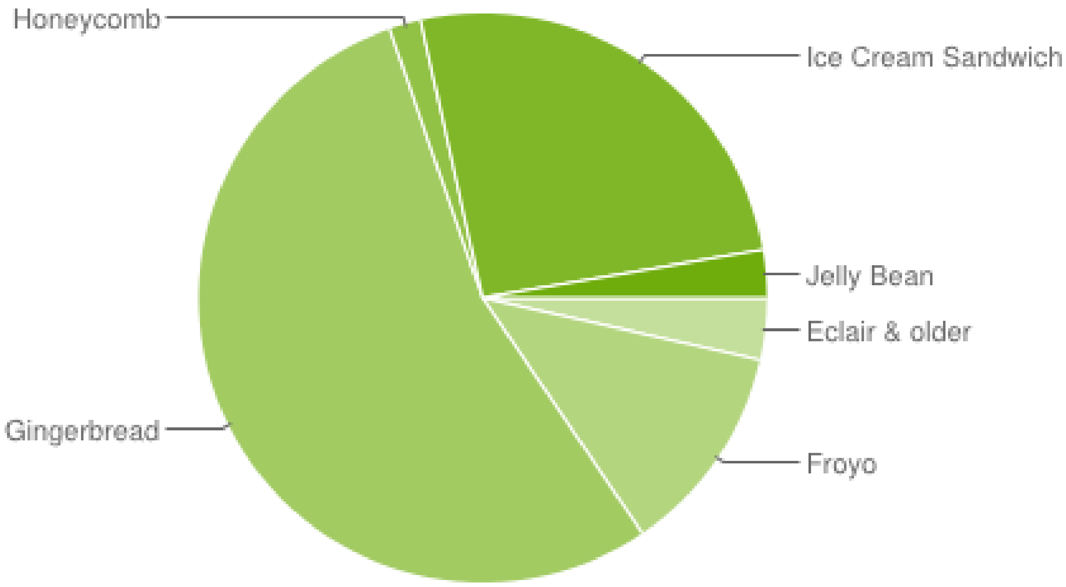


Figure 4 Android versions usage distribution

Table 1 and the pie chart in Figure 4 show the distribution of Android version usage based on the number of Android devices that have accessed Google Play, a marketplace for Android applications, within a 14-day timeframe ended on 1 November 2012 ("Platform Versions", n.d., Current Distribution section). From the chart, it is clear that the application should be compatible from at least Android 2.3.x (Gingerbread) upwards, because it has the highest share of usage distribution. It is predicted that there will be an increase in users for Android 4.0.x (Ice Cream Sandwich).

As to tablet device sizes, according to a compilation table on Wikipedia ("Comparison of Android Devices", n.d., Officially Released section, Tablet Computers subsection), device sizes range from 7.0 inches to 10.1 inches, with the two most common sizes being 7.0 inches and 10.1 inches. However, it is found that a 7.0inch screen is too small to host a two-octave keyboard interface to emulate real playing. A 7.75inch iPad screen only comfortably hosts a one-octave keyboard interface, shown in Figure 5.

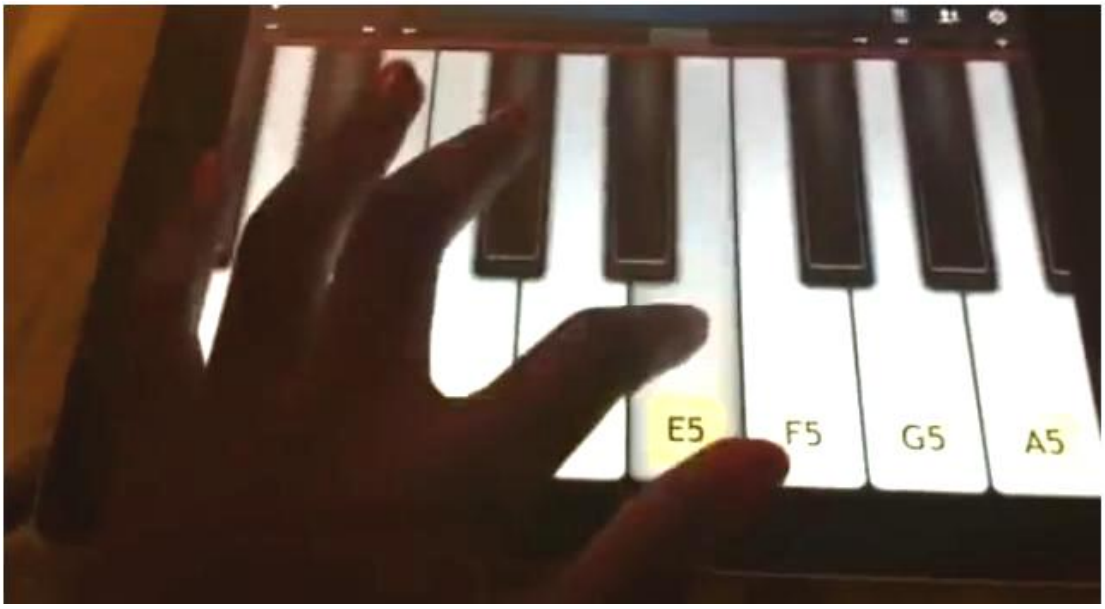


Figure 5 Keyboard interface on an iPad

Taken from a Youtube video (lange681, 2011), Figure 5 shows an iPad hosting a one-octave keyboard interface on landscape orientation. It can be seen that the key size is comfortable to emulate real-playing; but the keys look a little big. The iPad has a 7.75×5.82inch (197×148mm) width-length pair of screen dimensions ("iPad Dimensions and Features", n.d., iPad Screen Dimensions section), and to evenly divide the width for 8 white keys disregarding the spaces in between keys, one white key is about 24.6mm in width, which is actually bigger than a standard piano white key width of about 23.5mm, as noted from Wikipedia ("Musical Keyboard", n.d., Size and Historial Variation section, para. 3) as well as keyboard size researcher and keyboard maker Steinbuhler & Company's website (165.1mm octave span divided by 7 keys) (Steinbuhler, 2012, The Results section, para. 5).

Inversing the calculations, dividing a 10.1inch screen width with 15 white keys (the number of white keys in two octaves), one white key is about 17mm, which is only 1mm smaller than Steinbuhler & Company's ¾ DS Standard™ Keyboard size of about 18mm per white key (129.9mm octave span divided by 7 keys) (Steinbuhler, The Results section, para. 5). However the keyboard interface should only be hosted on landscape orientation.

2.4 Musical Interface Design for Mobile Devices

Many mobile devices today support multi-touch gestures on the screen. This allows for the mimicking of real music playing, as in on a real musical instrument. Interaction on the screen of a mobile device can be made intuitive, so that the user can play music on a mobile device just as how the user handles a real musical instrument. However, there are still some necessary constraints when it comes to designing musical interfaces for mobile devices because after all, the screen is only a flat surface without forms and shapes that can be felt by the user.

Schulz, Geiger and Reckter (2009) pointed out that because there is no haptic feedback when using a touch screen, visual feedback is necessary. Also, an optimum amount of allowable interaction, such as the number of detectable touches, should be decided depending on if the interface is meant for a single hand or double hands.

New techniques of interaction are arising for mobile devices. A new interaction technique mimics the real flute instrument by displaying "holes" on the screen that can be tapped simultaneously, and by blowing into the device’s microphone to produce a sound and tilting the device to create a vibrato (Schulz, Geiger and Reckter).

Another technique, by Brian Eno and Peter Chilvers, generates bell-like sounds that vary in pitch based on the Y-position of the user's touch point on the screen surface (Eno, n.d.). Toshio Iwai created a musical application for the Nintendo DS, called Electroplankton, that uses the position of game objects for sound generation (Schulz, Geiger and Reckter).

Chiang, Chiu, Dharma, and Tomimatsu (2012) proposed a technique that involves sketching drawings on paper and using the drawings to get sound responses from a mobile device connected to a bird-shaped sensor.

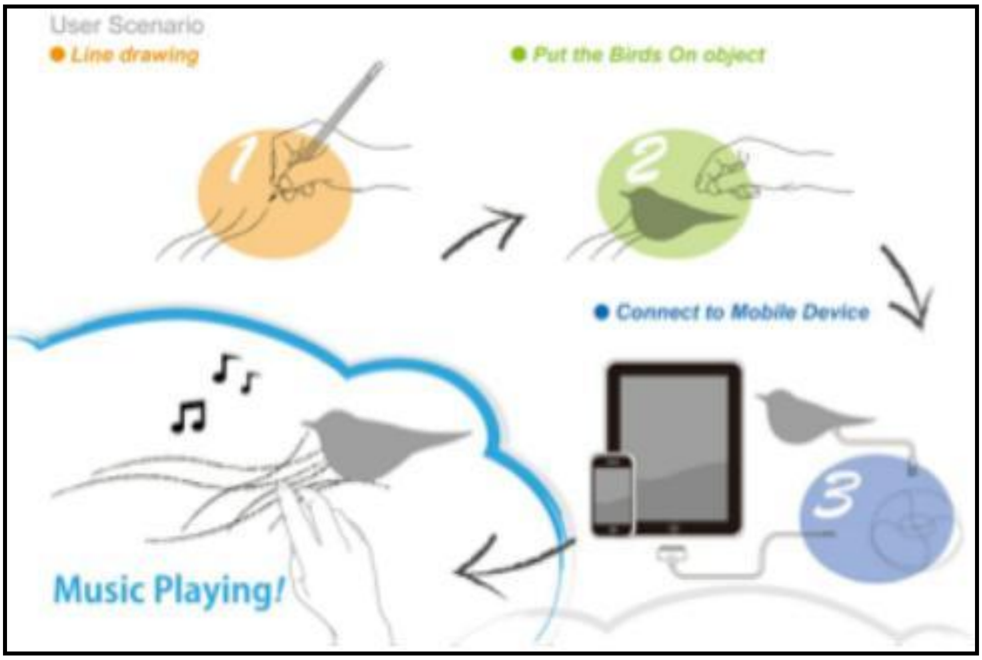
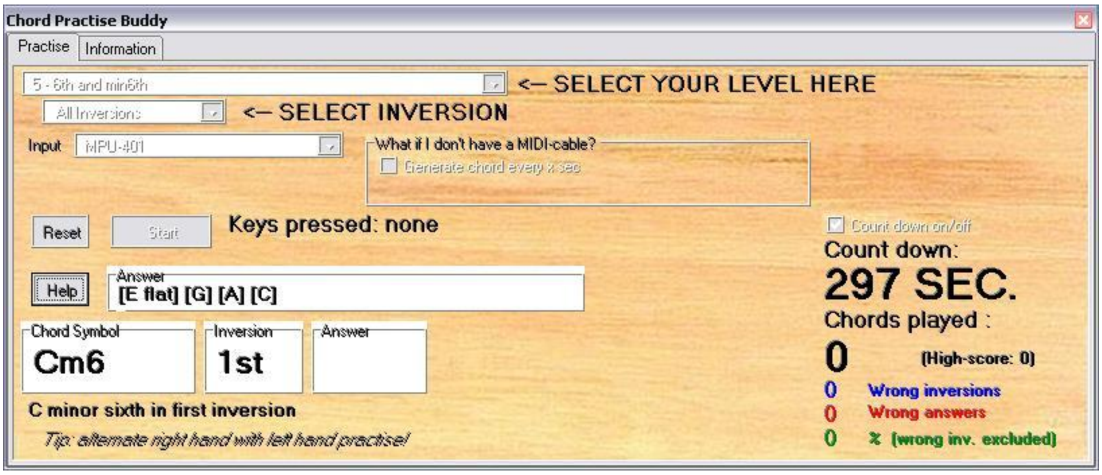


Figure 6 User scenario for the sketch interaction technique by Chiang, Chiu, Dharma and Tomimatsu

However, as the application to be developed for this project displays a keyboard or piano interface, conventional interaction techniques and constraints on a keyboard interface would be most suited. The technique would be to touch the keyboard keys on the screen as if hitting keys on a real keyboard. The constraints include: multiple touches allowed to mimic playing multiple keys on a piano; visual and audio feedback when keys are 'pressed'; and also overlapping audio responses when multiple keys are pressed at the same time.

2.5 Application Review

As the product of the project shall be a mobile application, existing applications are looked into in order to understand what has been done and what could be improved.



5 - 6th and min6th

All Inversions

Input: MPU-401

What if I don't have a MIDI-cable?
□ Generate chord every x sec

Keys pressed: none

Answer: [E flat] [G] [A] [C]

Chord Symbol: Cm6

Inversion: 1st

Answer:

C minor sixth in first inversion

Tip: alternate right hand with left hand practise!

☑ Count down on/off

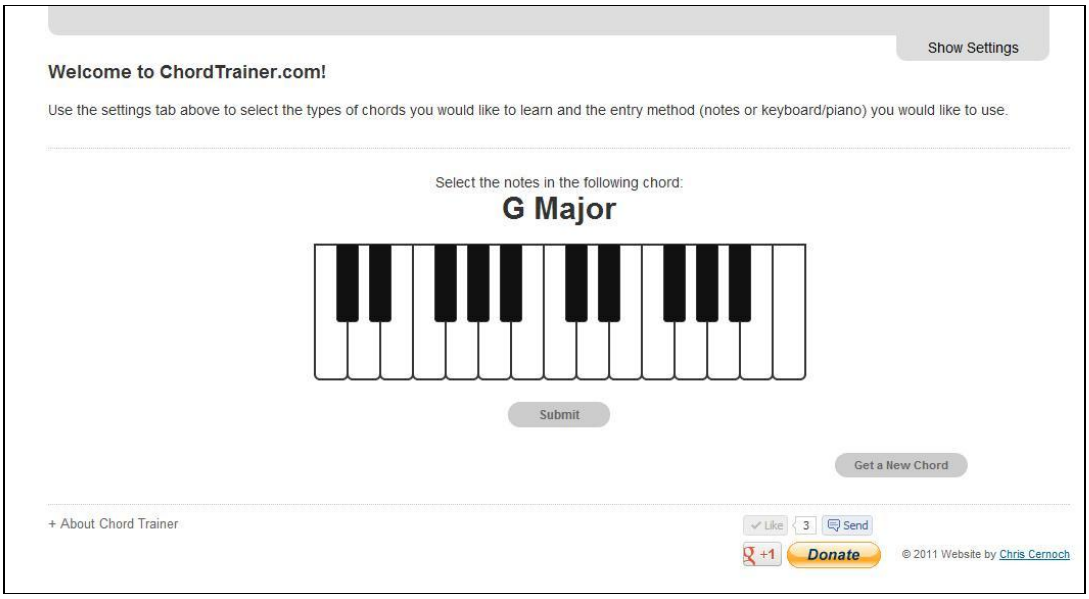
Count down:
297 SEC.

Chords played :
0 (High-score: 0)
0 Wrong inversions
0 Wrong answers
0 % (wrong inv. excluded)

Figure 7 Chord Practice Buddy screenshot

Chord Practice Buddy is a product of KeyPiano.com, a program meant to train sight-reading piano scores. Chord Practice Buddy is a chord-training program in the form of a game (“Products – Chord Practice Buddy”, n.d., para. 4). It works by giving the user a chord name, and the user is to play all the correct notes, in the specified sequence or chord inversion. The user may or may not be given feedback, depending on if they are running the program with a MIDI keyboard input.

This program is good in that it is hands-on, and performance statistics is recorded for the user. However, the limitations are that it requires a MIDI input, and it is not portable.



Welcome to ChordTrainer.com!

Use the settings tab above to select the types of chords you would like to learn and the entry method (notes or keyboard/piano) you would like to use.

Select the notes in the following chord:

G Major

Figure 8 ChordTrainer.com screenshot

ChordTrainer.com (Cernoch, n.d.) is a website with a two-octave keyboard interface that trains chord-playing by presenting the user with a major, minor, augmented or diminished chord, and having the user to click on three keys on the keyboard interface that make up the chord. It is inversion-invariant, meaning that there are no specific chord inversions taken into account of determining right or wrong chords; this is good for beginner keyboardists as they will have more freedom to explore different chord formations.

However, being a website, the keyboard interface can only be accessed using a mouse. This does not provide hands-on practice for the user. Also, the user can only click one key at a time, as opposed to real chord-playing of playing three keys at a time.

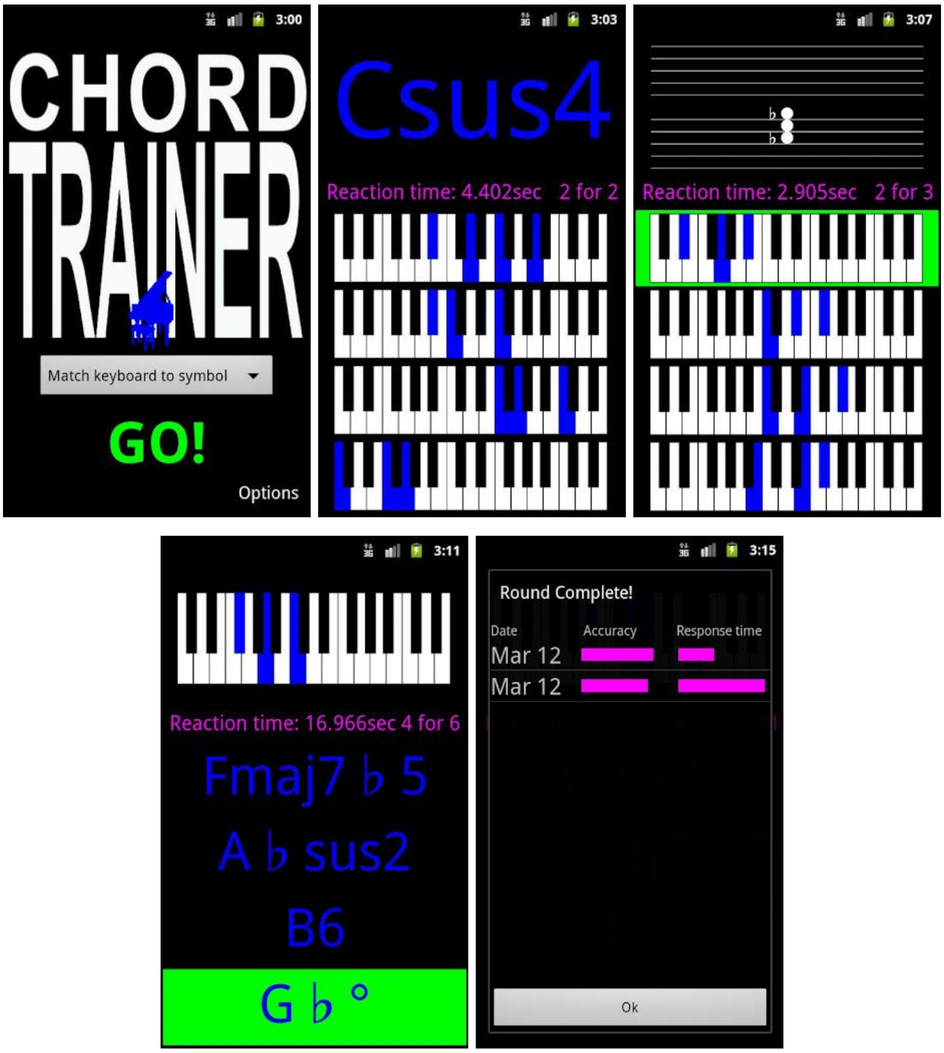


Figure 9 Chord Trainer Android application screenshots

Chord Trainer, by automatSoftware, is an Android application intended for beginner to intermediate piano players to practise visual recognition of piano chords, in the form of quizzes (automatSoftware, 2012, Description section, para. 1). It presents the user with a chord name and multiple-choice answers of highlighted keys on a keyboard, or vice versa. The user needs to select the correct answer in the fastest time. It also has quizzes in musical notations versus keyboard or chord name answers. Chord Trainer tracks user performance statistics like accuracy and reaction time.

Chord Trainer is a good application in that it supports training in multiple musical formats from notations to chord names to visual keyboards. However, the fact that the answers are presented in a multiple-choice format does not allow the user to have hands-on practice.

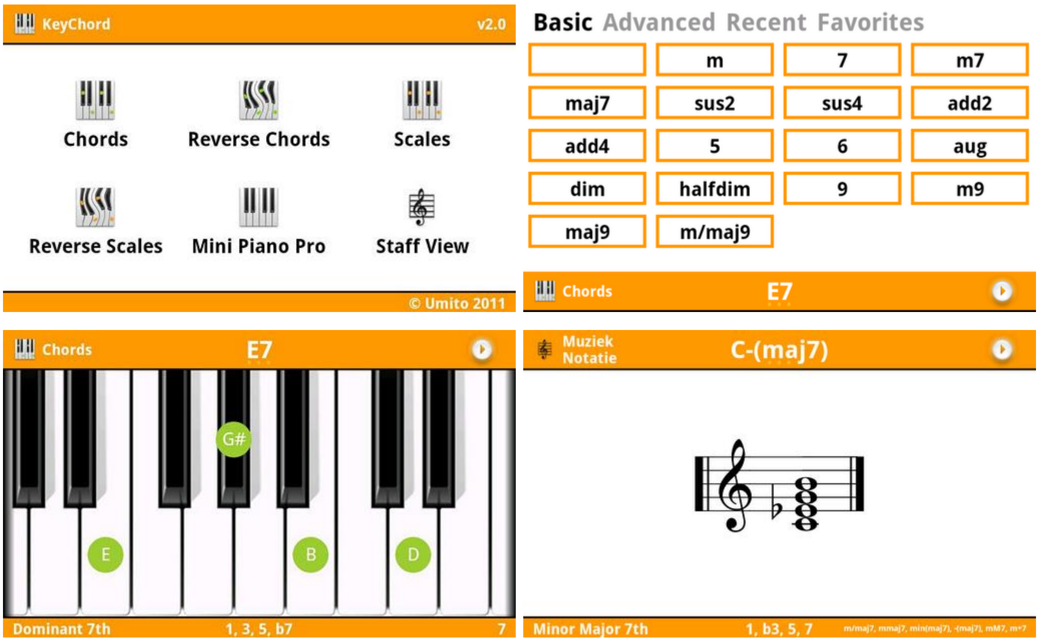


Figure 10 KeyChord Android application screenshots

KeyChord is an Android application that also trains chords, but more towards visual recognition only, as the application is a dynamic dictionary or reference application for keyboard chords and scales (Umito, 2012, Description section, para. 2). The user picks a chord name and the relevant keys on the keyboard are highlighted, vice versa.

This application, and many other similar applications, supports many chord types as it has backend algorithms to calculate chords. It allows novice keyboardists to learn new chords or to refresh on existing chord knowledge, but does not allow for hands-on practice to train responsiveness and finger agility.

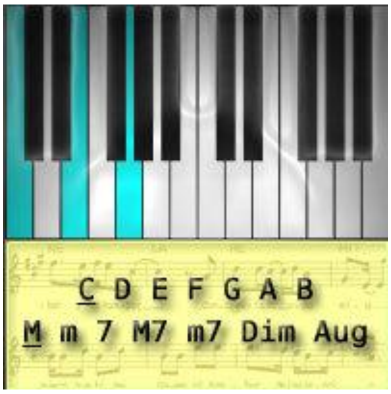


Figure 11 IQ Piano Chords 1.0 screenshot

IQ Piano Chords 1.0 is similar to KeyChord, a dynamic dictionary or reference application for keyboard chords in the form of an Android application. The user picks a note for the chord and the chord type, and the relevant keys are highlighted on the keyboard (IQ-mobile, n.d., Description of IQ Piano Chords 1.0).

Again, applications as such does not actually provide hands-on training and learning through practice, but only visual recognition of chords. This application would be useful for novice keyboardists looking to learn or refresh on chords, but not for beginner keyboardists who can already play.

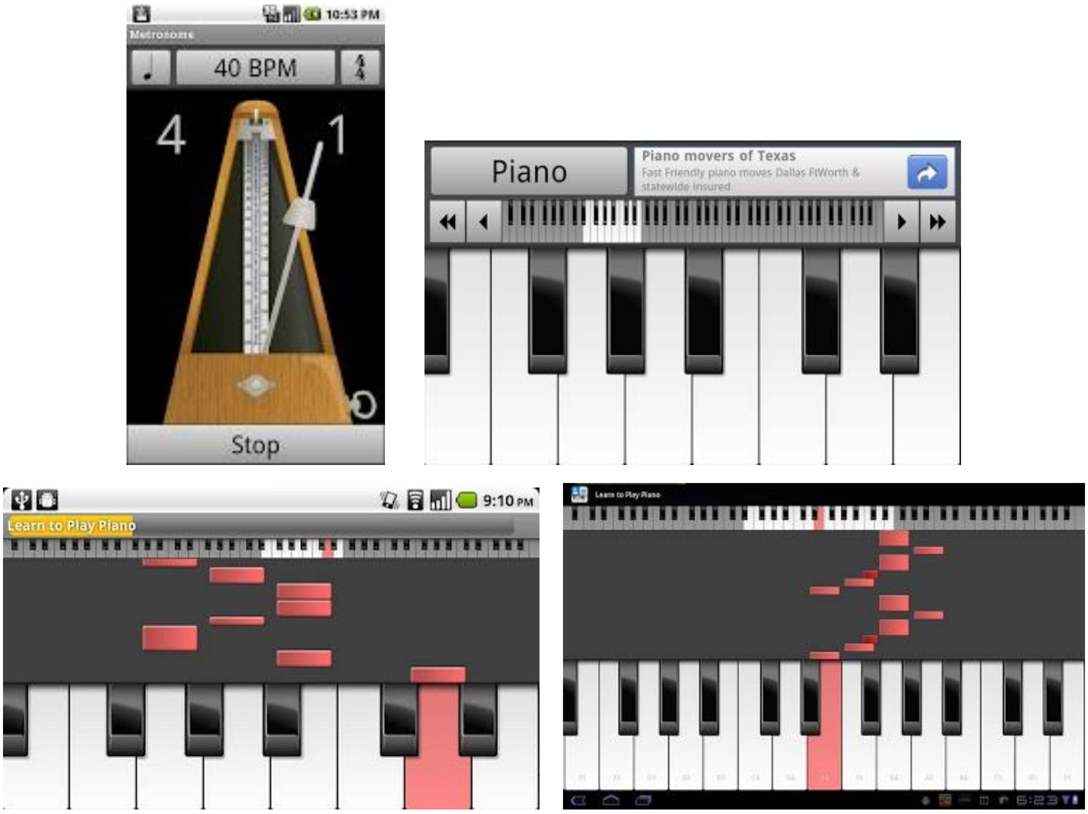


Figure 12 Musical Piano screenshots

Musical Piano is an Android application developed by Christopher Souvey. It is essentially a piano application with many side features including a metronome, pitch pipe, drumkit, MIDI instruments played on a piano interface, and a piano practice mode that allows the user to learn to play the piano via falling blocks directly above the key to be pressed (Souvey, 2012, Description section).

This application is a good music application, given all the features that all levels of musicians can make use of. However, it teaches the user to play individual notes instead of chords. Also, its uses a follow-along teaching method, in which users play the piano following the sequence of the falling blocks without understanding the theory behind it. Users who are good at playing the piano in the application may not be able to emulate the skills in real-playing without the falling blocks.

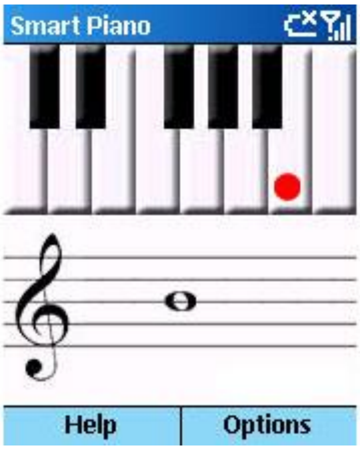


Figure 13 Smart Piano screenshot

Smart Piano is a piano application designed for Windows Smartphone, and supposedly the keys on the piano are controllable through the phone's keys, with each key mapping to a note on the piano (Thornton, n.d.). Smart Piano also includes four chords (C Major, G Major, F Major, and Middle C) accessible through the directional keys on the phone. The relevant keys on the piano will be highlighted, and the relevant notes appearing on the music stave.

This application also does not provide hands-on practice. Furthermore, it only teaches four chords within one chord type. Hence this is not very practical for chord-training.

There exist many other chord-training related applications aside from the above reviewed applications. However it is found that many related applications work on similar concepts than the above reviewed applications. Others are simply virtual piano with no specific learning targets for the user.

Hence, below is the summary for the abovementioned applications, pinpointing the features that other related applications emulate.

App Name	Type	Pragmatism	Provision of Performance Statistics	Portability
Chord Practice Buddy	Multi-level game	Hands-on; however a MIDI keyboard input is needed	Time, number of chords played, wrong answers and the percentage of wrong answers are recorded	Not portable as it is meant to be used on a computer with a MIDI keyboard input
ChordTrai-ner.com	Website	Semi hands-on; however only one key can be selected at a time	No statistics provided; only indicates right or wrong	Semi portable; accessible from a mobile browser
Chord Trainer	Mobile app; multiple-choice quiz	Not hands-on; user picks answer from multiple choices	Accuracy of answers and response time	Portable as it is a mobile app
KeyChord	Mobile app; chord dictionary	Passive; user only sees the highlighted keys of a chord	Not applicable	Portable as it is a mobile app
IQ Piano Chords 1.0	Mobile app; chord dictionary	Passive; user only sees the highlighted keys of a chord	Not applicable	Portable as it is a mobile app
Musical Piano	Mobile app; virtual piano	Hands-on; user plays keys according to falling blocks; but it does not teach chords	Unknown	Portable as it is a mobile app
Smart Piano	Mobile app; virtual piano	Not hands-on; user plays the piano or chords through phone keys	Not applicable	Portable as it is a mobile app

Table 2 Summary of application review

With the above analysis of works on related topics as well as existing applications, the concepts, requirements and scope of the application for the project is known. In terms of interaction design, the application uses conventional touch interactions on a virtual keyboard or piano interface. The interface provides audio feedback upon touching the keys, as well as visual feedback showing touched and untouched keys to compensate for the lack of haptic feedback.

Functionally, the application has a two-octave keyboard interface as of ChordTrainer.com. It displays chord names like all the reviewed applications, and the user is to select the three notes involved in the chords. The chords are randomly generated to avoid fixations on chord patterns in songs. However, the chords are inversion-invariant to give freedom to beginner keyboardists, as well as to exercise their creativity. Timing the user's progress to drill on responsiveness, the application records user performance statistics like the Chord Trainer Android application, so that the user can keep track of their improvement.

CHAPTER 3
METHODOLOGY

3.1 Research Methodology

The research was accomplished through surveys. Two surveys were conducted using questionnaires with closed-ended and open-ended questions targeted at potential users of the application. Potential users of the application are the people having a high possibility to voluntarily use the application out of the research environment, and these people are beginner keyboardists, or beginner musicians who could also play the keyboard or piano. The number of target users surveyed was 30 beginner musicians.

The surveys were carried out in the orchestra room in UTP, where all levels of student musicians gather in the student orchestra. Amongst the student musicians, the beginner ones were targeted for the surveys. The surveys, respectively, were meant to find out about the user perception to training chord-playing using a mobile application; and to measure the usability of the application, or in other words, the experience and effectiveness of training chord-playing using the application.

But of course, the application was first developed before the surveys were conducted.

3.2 Project Activities

3.2.1 Development of Application

The chosen development methodology was the prototyping model. As the product of the project is a mobile application, there could be many ways to achieve the same user interface and functionalities, such as the algorithms to calculate the correct notes in a chord and the way the keyboard interface is animated. Hence the prototyping model was useful to identify, through quick iterations of prototypes, the methods that worked and those that did not, as well as the method that was best suited to be used for a function on the application. Changes could be made fast to eliminate the techniques that held back the application. Also, prototypes allowed for early user evaluation and hence participation in the development of the application, as well as a faster grasp of the intended final product. This could be seen in Iteration 3 of the application development (see Results and Discussions chapter, Section 4.3.3), where the app was put out for user testing before proceeding with Iteration 4. This ensured that the final product suits the target users.

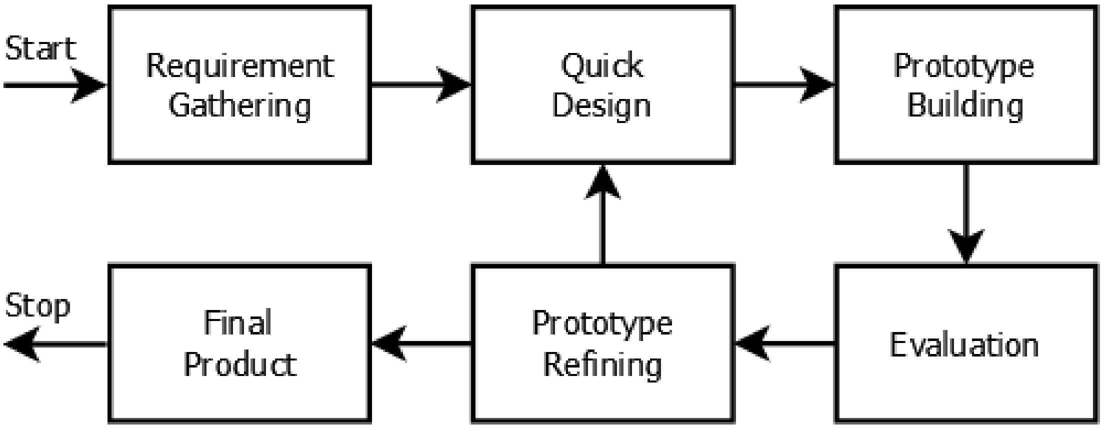


Figure 14 Prototyping development model

3.2.2 Development Activities

Based on the above development methodology, the activities in the project were divided into a few phases.

In the requirement gathering phase, functionalities needed for the application were identified; this included features such as the backend chord calculation and the data to gather to measure a user's performance and improvement using the application. Also gathered were the best suited methods or algorithms to implement the identified features for the application, such as the algorithms to calculate the correct notes in a chord, and the algorithms to calculate a user's performance using the application as a measurement of their learning and improvement.

In the quick design phase, the phase where the prototyping cycle starts, a prototype was rapidly drafted out based on the requirements gathered and the necessary algorithms, found from the requirement gathering phase (for the first prototyping cycle) or gathered from the previous iteration (for the second and following cycles). Following the model produced from the quick design phase, the prototype building phase saw the actual development of the prototype.

The prototype was then evaluated in the evaluation phase. In this phase, various testings were be carried out, to see if the prototype met the requirements or exceeded the scope, and if any new requirements arose, or if there were any problems that needed to be addressed. At a later iteration, the prototype was also put out for testing by target users. All the feedback and results gathered in this phase were noted. Then, in the prototype refining phase, the feedback received was ordered and addressed accordingly, by tweaking the prototype, or redefining some requirements, before looping back into the quick design phase.

The combined phases of quick design, development, testing and refining the prototype are collectively called the prototyping phase. The prototyping phase was iterated in 4 cycles in this project, before a final prototype that met all the requirements was produced. The final draft of prototype is the final product of the project development.

Of course, before embarking on all the activities in the development of prototypes, there was a development research phase. Activities in the development research phase (an overlapping activity with literature review) included analyzing existing applications to determine the features that could be adapted into the application prototype, as well as the features that could be improved. Techniques in Android application development were also researched to find the best way to develop the application prototype. Also, the methods in music education and music training were researched to understand the factors that are important in creating a conducive and effective training environment, such as providing two octaves of piano interface to give beginner keyboardists enough freedom to explore chords.

The processes in each prototyping iteration are documented in the Results and Discussion chapter, Section 4.3.

3.2.3 User Perception Study

A questionnaire that consists of 9 close-ended questions and 1 open-ended question was given out to 30 target users, along with the Chord Trainer application prototype. The respondents are beginner musicians playing in a student orchestra. Aside from their orchestral instrument, they all play the piano or keyboard, whether formally trained or otherwise. The survey was focused at finding out the users' music background and abilities, as well as their acceptance towards music training using a mobile application. The questionnaire also aimed to find out if the respondents find it practical to train their chord-reading skills using a mobile application, and if they generally find the Chord Trainer application to be helpful. The open-ended question asked for other comments that the respondents might have. The questionnaire could be found at Appendix B.

Results and analysis of the survey are documented in the Results and Discussion chapter, Section 4.4.

3.2.4 Usability Study

Another questionnaire was given out to the respondents after they have spent an average of 30 minutes using the application, to find out about the general usability of the Chord Trainer application.

The questionnaire used was the System Usability Scale (SUS), a simple, ten-question questionnaire assessing the general usability of a system (Brooke, n.d., SUS – the System Usability Scale). The SUS is meant to be used when users have spent some time on a system, but before they could be debriefed or have a discussion about it. The SUS records the users' rather immediate response towards the system. Users are advised to give quick responses for each question, rather than dwelling on the questions for a long time.

The questionnaire has ten fixed questions, and all the questions use a Likert scale for the answers, with 1 being “strongly disagree” and 5 being “strongly agree”. It could be found at Appendix C.

Results and calculation for this usability study could be found in the Results and Discussion chapter, Section 4.5.

3.3 Key Milestones

Key milestones are outlined in the Gantt charts. They required a deliverable in the form of either a document submission or oral presentation, or a combination of both. Past milestones are coloured in green.

3.4 Gantt Charts

3.4.1 Final Year Project 1

Week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Title Selection/Proposal	🟩											
Submission of Project Proposal		▨	🟩									
Literature Review				▨	▨							
Extended Proposal Submission						🟩						
Proposal Defense							▨	▨	🟩			
Interim Report										▨	🟩	

Figure 15 Gantt chart for FYP1

3.4.2 Final Year Project 2

Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Programming Research	▨													
Prototype Development		▨	▨	▨	▨	▨	▨	▨	▨					
Submission of Progress Report							🟩							
Surveys to Target Users									▨	▨				
Pre-SEDEX										🟩				
Dissertation											🟩			
Viva												🟩		
Technical Paper													🟩	
Final Dissertation														🟩

Figure 16 Gantt chart for FYP2

3.5 Tools Required

In terms of hardware, as mentioned in an earlier section, the screen size of smartphones is not suitable to simulate real playing. Hence, even though development for smartphones and tablet devices are highly similar, the targeted medium for development was Android tablet devices with a screen width of 10.1inch or above. A Samsung Galaxy Note 10.1 was used for testing.

As to software, the development was performed using an integrated development environment (IDE) that supports the Android software development kit (SDK). Eclipse 4.2 was used, as Eclipse is the officially supported IDE by Android, with the Android Development Tools (ADT) plugin that allows for easy integration with the Android SDK ("Building Your First App", n.d., para. 3) ("Developer Tools", n.d., para. 1). For initial virtual testing, the Android emulator, which comes with the ADT, was used. However, as there are limitations in testing using the emulator, soon testing switched to solely using hardware.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Data Gathering and Analysis

From a brief programming research prior to the beginning development, it was found that there are various resources on Android programming available on the Internet. However, it was gathered that the most helpful resource to getting started was the official Android Developers website, as it has step-by-step guides from setting up the integrated development environment (Eclipse, the recommended and hence chosen one) for Android development to developing a beginner app and getting it displayed on the Android device emulator. The guides were enough to get the student acquainted to Android development.

Given the student’s brief experience in mobile development from internship, the student could quickly run through some steps in learning and move onto scouting out for tutorials and resources in development.

From the development process, it was found that the most helpful resources included: the aforementioned official Android Developers website, the Android Development Video Series by thenewboston on Youtube, and, usually for diagnostic cases, links to StackOverflow.com from search engine results, as well as the discussion forum on Google Code.

The official Android Developers website provides the full reference to the Android API, as well as conceptual guides on the running of Android, such as the Android activity lifecycle, and a range of information on other aspects from training to tools. The most used component was the API reference to check on classes and their functionalities.

The Android Development Video Series by user “thenewboston” on Youtube are made up of easy-to-understand tutorial videos covering various techniques on Android development, up to publishing an app to the Android market. There are 200 videos in the series, but each on average is only about 5 minutes, and the videos could be watched separately for a developer with Java and mobile development exposure. They were useful when coming across a new class or technique on Android programming.

Discussions on StackOverflow.com and the Android project page on Google Code were usually helpful when debugging. Users post unsolved errors or even programming conceptual questions on these websites, and the solutions given by other users were adapted to similar problems faced during programming.

Other useful Android development resources included tutorial blogs such as the Android Developers Blog (http://android-developers.blogspot.com/), Android 4 Mobile (http://android-developing.blogspot.com/), helloandroid.com (http://www.helloandroid.com/tutorials/), Vogella.com (http://www.vogella.com/articles/), and even information technology (IT) blogs such as ZDNet (http://www.zdnet.com/blog/).

Android development resources aside, one resource specific to the application in this project was the algorithm to calculate the correct notes for an inversion-invariant chord given the chord root note and the chord type. This algorithm was needed to check against user input notes on the virtual keyboard, to determine if the entered chord is correct. Fortunately, this algorithm was available from the jQuery script of ChordTrainer.com, and the owner Chris Cernoch, gave his permission to use the algorithm.

4.2 Experimentation / Modelling

As a first step to Android development after the setup of Eclipse and the necessary plug-ins, the Building Your First App on the official Android Developers website was followed through. A simple app named FirstApp that displays “Hello World!”, as well as its extension of displaying user input text, was made and run on the Android device emulator (also known as Android Virtual Device).

Following that, instead of starting afresh to build a prototype, the prototype was expanded from FirstApp, to allow for experimentations without concern on the core structure the app such as defining its “main function” (known as main activity in Android development), and at the same time to further understand the core structure of an Android app including how the Android main activity and delegates work. Android user interaction aspects were also studied along the way, to determine what views and interaction or gesture events would work the best for the app.

4.3 Prototype

4.3.1 Iteration 1

From the experimentations, and with some help from the videos by thenewboston, the prototype was successfully modelled from FirstApp, and run on the Android Virtual Device (AVD).

The chord calculation algorithm by Chris Cernoch was also successfully adapted into a java class to be used with the prototype.

As bigger screen sizes on the AVD could not be supported, the prototype had to be run on a ‘wrong’ screen size. That aside, the AVD also did not support multiple touches (multi-touch events). Hence the prototype fell back to single touch only.

As seen in the following figures (Figures 17 and 18), the prototype was locked on landscape orientation, which is one of the requirements for the app. If the keyboard interface was to be displayed on portrait mode, it would not be possible to display two octaves of keys.

Figures 19, 20 and 21 show that selection of one, two and three keys respectively. Upon detecting three keys, a method to calculate the chord notes, adapted from Chris Cernoch’s algorithm on ChordTrainer.com, would be invoked. Then the selected keys would return to be unselected, as in Figure 17 or 18.

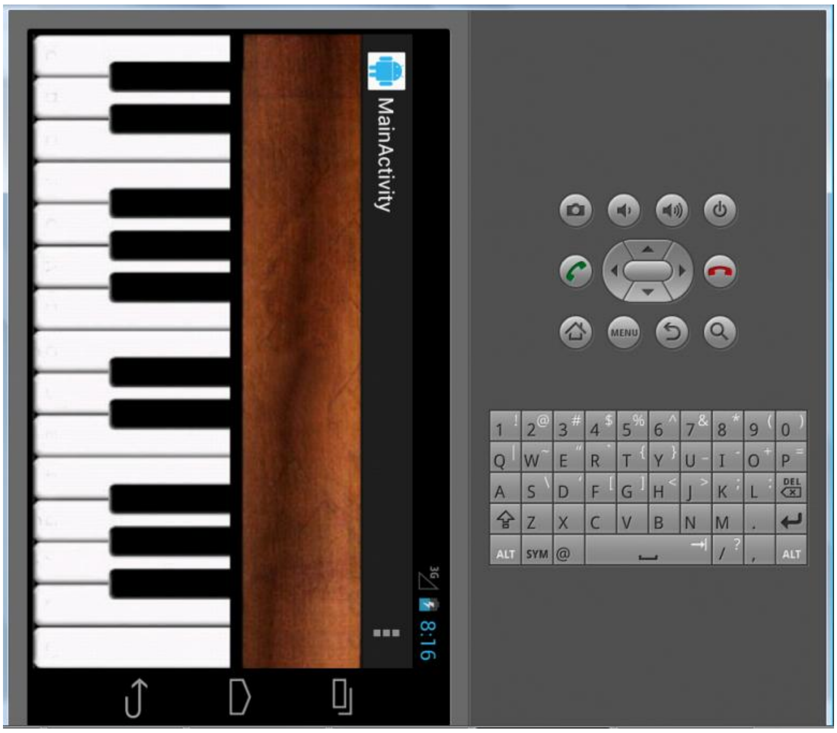


Figure 17 Prototype on portrait orientation on the emulator

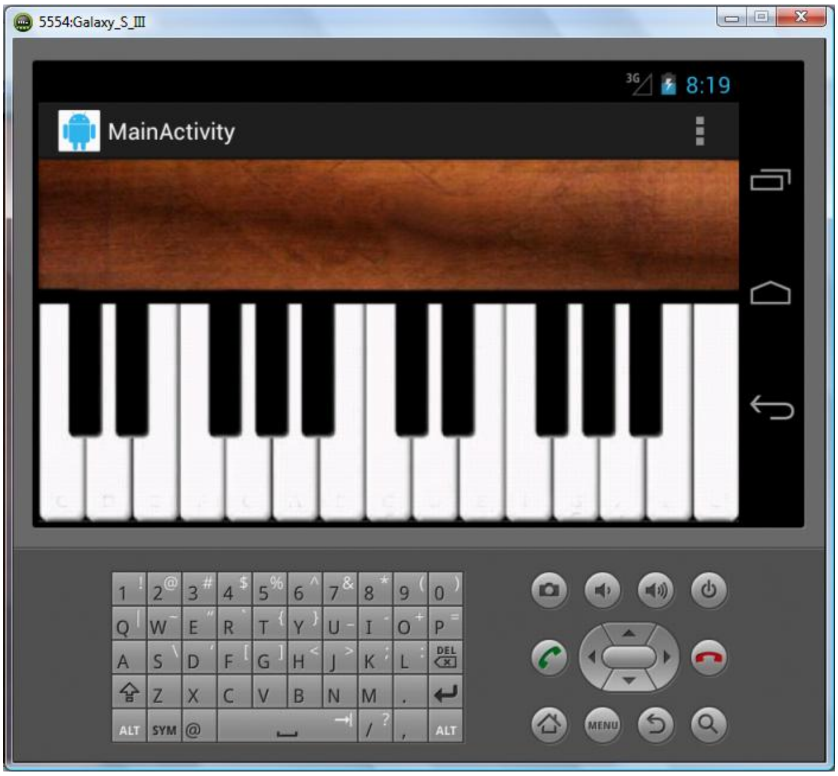


Figure 18 Prototype on landscape orientation on the emulator

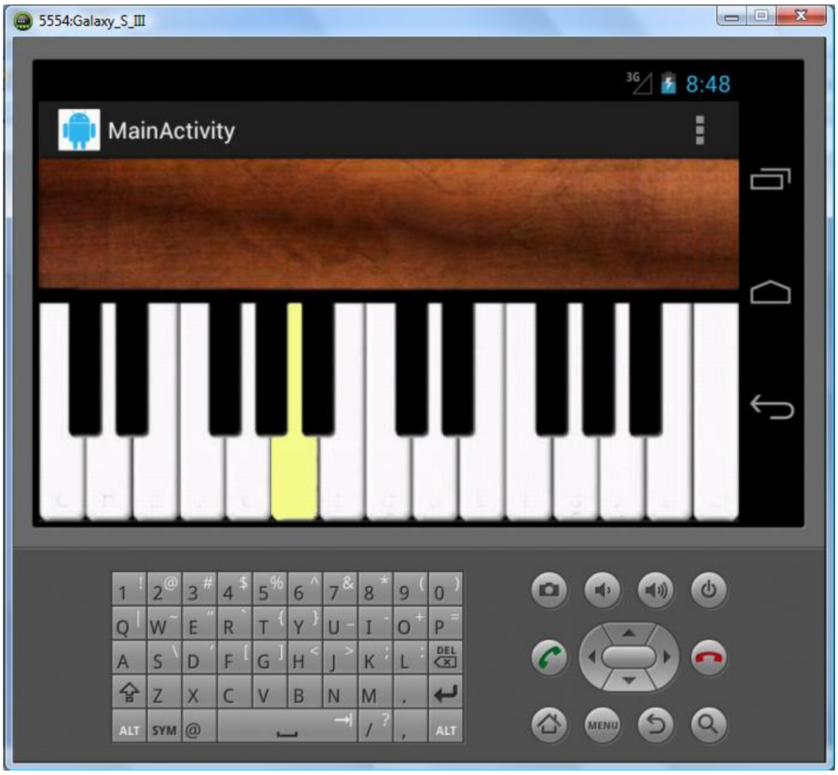


Figure 19 Selection of one key

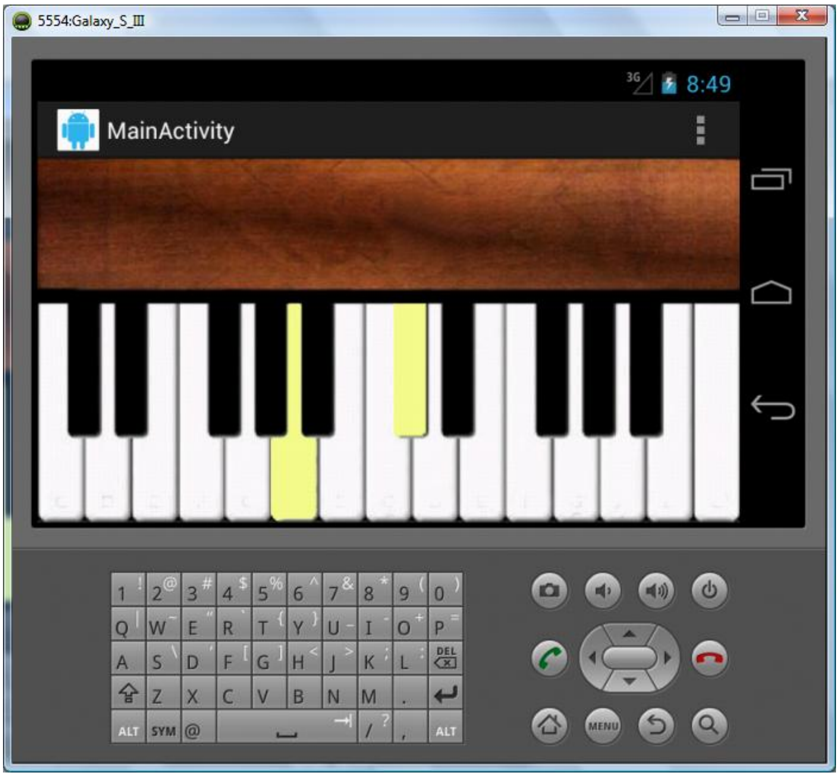


Figure 20 Selection of two keys

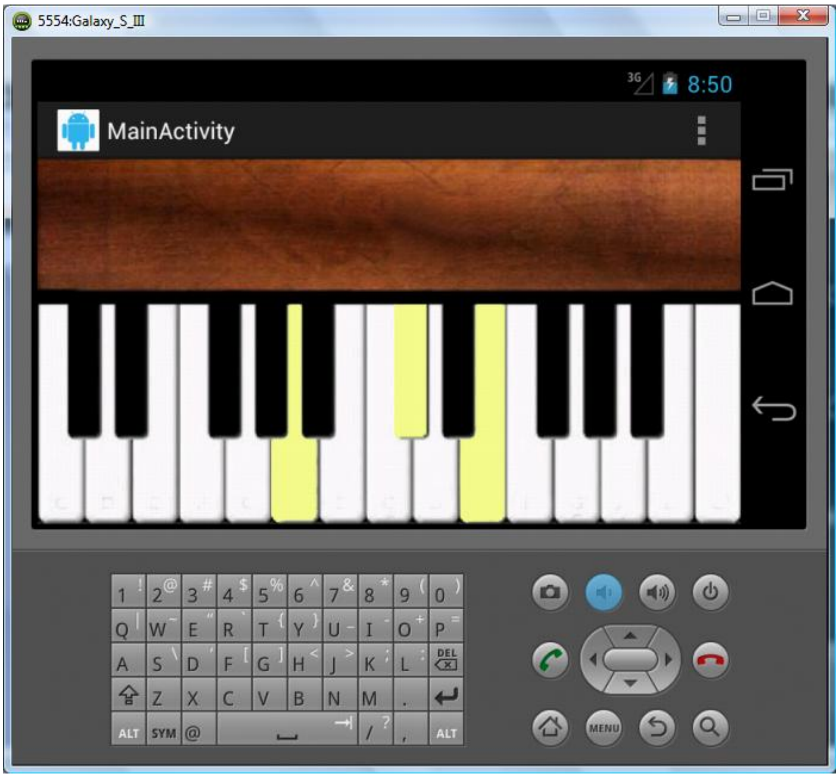


Figure 21 Selection of three keys

Ideally, a key should only be highlighted when it was touched/clicked, and unhighlighted when the touch/click was removed. However, since multi-touch was not supported on the AVD, the prototype was made to fall back to three single touches to make up a chord.

Also, once the chord calculation method was invoked, the user should be informed if the entered chord was correct or wrong, but text display was not available on the prototype yet. On the Eclipse console log, it could be seen that the chord calculation method worked.

Aside from multi-touch and text display, other features that had not been implemented in the prototype included: elapsed time display, performance statistics such as how many chords attempted and how many correct chords, random generation of chords for the user to attempt, and sounds when keys were touched.

The prototype was still an expansion of FirstApp, the “Hello World!” app. Now that the core workings of an Android app were roughly understood, the prototype could be transferred to a new Android application project on Eclipse. Also, a real tablet device was needed to work on the multi-touch feature, which is an important part of the app.

The following was the planned flowchart for the prototype in Iteration 2.

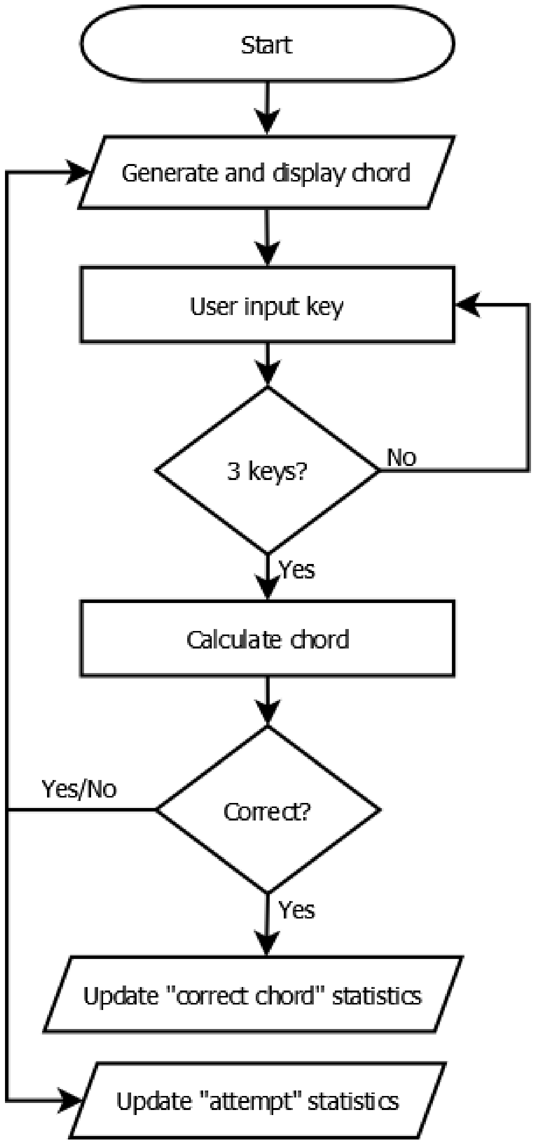

Start
Generate and display chord
User input key
3 keys?
No
Yes
Calculate chord
Yes/No
Correct?
Yes
Update "correct chord" statistics
Update "attempt" statistics

Figure 22 Planned flowchart for prototype in Iteration 2

Note that there is no explicit end to the flowchart. This is because the app could be paused/stopped at any point in the flowchart. Also, the app is meant to continuously run until the user puts it to the background. Android's app activity has its own lifecycle to pausing/stopping it when the app is pushed to the background. Once in the background, the Android activity lifecycle states that the activity will be put on-hold and will not execute any code ("Starting an Activity", n.d., Understanding the Lifecycle Callbacks) ("Pausing and Resuming an Activity", n.d.). Hence the app does not need to be explicitly ended; although, of course, there would be variables to be saved before the app retreats to the background.

4.3.2 Iteration 2

After ensuring that the backend chord calculation algorithm was working, it was time to add textual messages above the piano interface to show the user the chords to play, whether they were correct or wrong, and a record of attempts made and correct chords played, as well as the time they had spent on the application prototype.

The layout and colours of the text displays were chosen based on the prototype's preliminary design (see Appendix A). However, the layout took awhile to be implemented due to Android's flexibility in layout design.

There are a number of layouts available for Android, including linear layout, relative layout, frame layout, and list and grid layout. And the layout could also be built programmatically or using XML. As the text displays were to overlay the piano interface, relative layout was chosen. A relative layout for the group of text displays were built using XML. The text displays managed to overlay the piano interface, however they also overlapped with each other.

After a period of diagnosis, the student decided that his exploration on XML was not yielding effective results, and hence switched to building the layout programmatically. Also, in the end, the text display layout finally turned out right after nesting a few layers of linear layouts into the relative layout.

After that, the student was finally able to obtain an Android tablet device. It was not the targeted Samsung Galaxy Tab 10.1, but a Samsung Galaxy Note 10.1. But it sufficed as a testing device as its screen size is also 10.1inch.

With the device at hand, the multi-touch feature was worked on, modifying from the single touch fallback. The prototype was able to support up to 10 key touches, although a chord would only be calculated when there were 3 key touches. Thereafter, the prototype was ported away as an extension of FirstApp, and over as a new Android app project on Eclipse, where it was named Chord Trainer.

As a new Android project, the random generation of chords was implemented, to test if the text displays worked dynamically. The chord name display was able to show dynamically generated chords. The backend chord calculation could also take in the chords for computation. And the message display worked fine to indicate if a chord was played correctly or wrongly.

Then the other text displays, such as the chord attempt counter and the correct chord counter, were quickly implemented using integer variables that update themselves with each chord generation (chord attempt counter) and with each correct chord played (correct chord counter). The timer counter was easily implemented using Android's Chronometer class. The font size for all the text displays was upped to fill the tablet screen fuller and better.

Next, sound was implemented. Piano sound samples of small file sizes were downloaded. As a complete range of piano sound samples was not available free, only sound samples of three C notes were downloaded. Then they were pitch-tuned using audio-editing software, to obtain two octaves of piano sounds. After that the sound samples were included into the prototype, matching their trigger with the multi-touch detection codes.

When built and run on the device, it was found that the prototype took about 10 seconds to load, due to the loading of the sound samples into memory. Previously the prototype's startup was almost instantly. In any case, since with the sound samples, it was expected that the prototype took time to load (as checked with other piano apps), a splash screen was built to show the user a loading spinner while the sound samples were being loaded.

The following diagram depicts the prototype architecture at the end of Iteration 2.

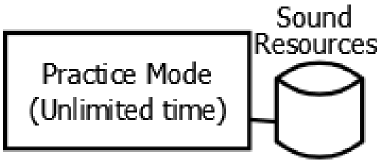


Figure 23 Prototype architecture at Iteration 2

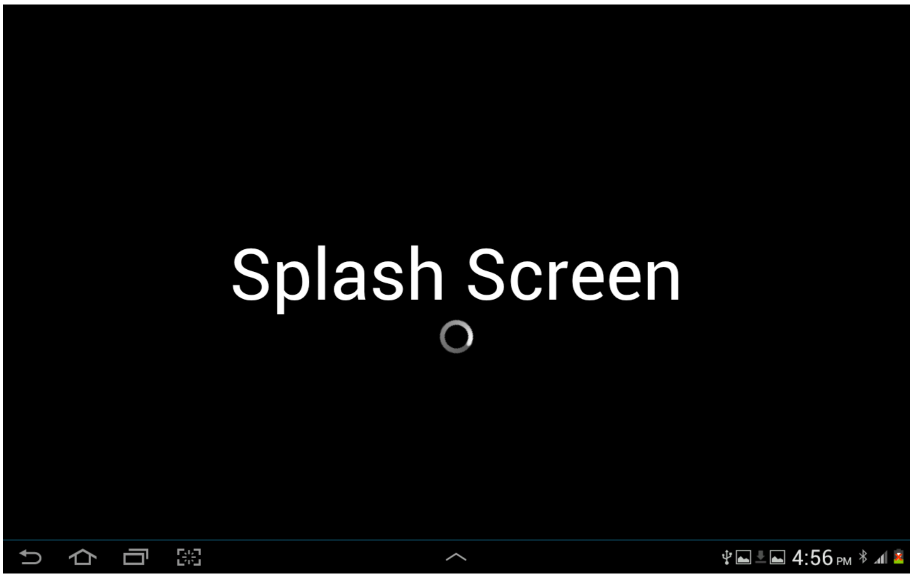


Figure 24 Splash screen with a loading spinner

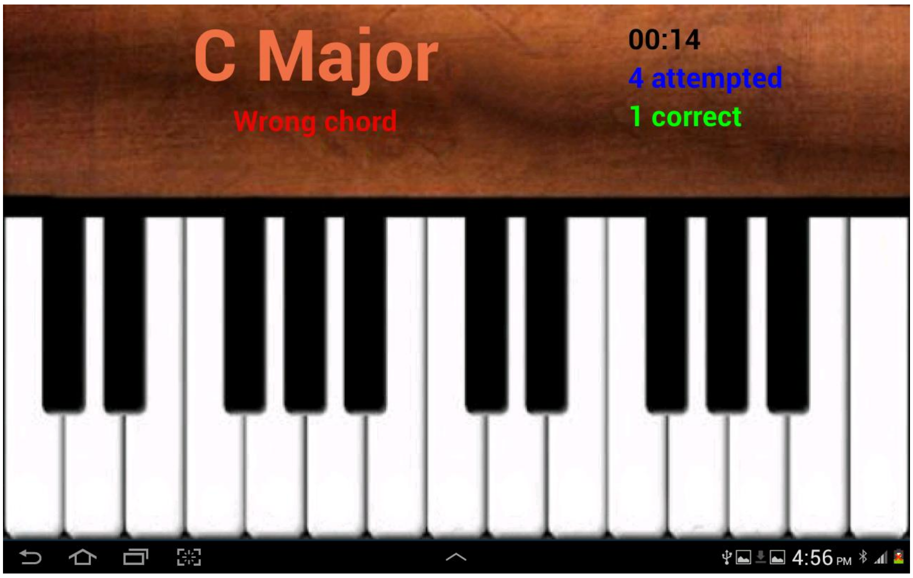


Figure 25 The piano interface with all the implemented text display

4.3.3 Iteration 3

The supervisor felt that the prototype did not have enough substance for the user, and suggested that a quiz mode to be added, in which the user could test their own progress after they felt like they had had enough practice.

The supervisor also asked as to the lack of a main menu screen. It was explained that a main menu screen was not practical for the application prototype because the user would have to wait twice: once during the splash screen (the splash screen should remain to provide information about the application to the user) before the main menu screen; and once more when they were moving to the piano interface as the sound samples were being loaded.

Looking into other piano apps, it was found that they also did not have a main menu screen, but simply a splash screen while the loading the sound samples, and then straight into the piano interface.

Also, it was read from the web that it was not recommended to load and hold an object set of sound samples during the launching of the application, and then pass the sound sample object over to the piano interface when it was called. This was because the sound sample object was considered heavy media, and holding them in memory and passing them around the application was not a good practice for memory management on Android. Hence the prototype would not have a main menu screen.

However, the supervisor's suggestion of a quiz mode was taken into consideration. The quiz mode was implemented in the form of a touch button, where, on toggle, would start or stop the 1-minute quiz. The results of the quiz would be saved in a simple database, so that the user could review their progress. This statistics would be viewable through another touch button.

A third touch button, for exiting the app, was added below the two abovementioned buttons. This was added to give user the comfort of explicitly exiting the app, even though in Android, applications exit automatically when pushed to the background. However, its functionality was not added yet, just to see if the user was comfortable without it.

The buttons aside, it was found that the chord calculation had some glitches, in that it could not calculate augmented chords, as well as chords ranged in the upper octave of the piano interface. It turned out to be that there was an inconsistency in the chord displayed and the chord plugged into the chord calculation algorithm. Also, checking with Chris Cernoch's algorithm, a missing condition was found in the adapted chord calculation algorithm.

After adding the buttons and fixing the chord calculation glitches, as well as putting up a proper splash screen, the app was put out for users to test.

The architecture of the prototype, with the added quiz mode, at the end of Iteration 3, is as the following diagram.

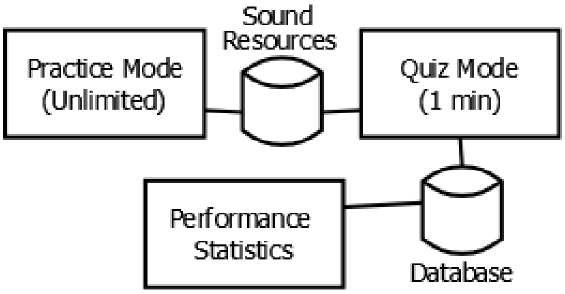


Figure 26   Prototype architecture

There are two modes in the application prototype: the default practice mode that is started when the application is started; and the 1-minute quiz mode that the user starts on their own. There is also a viewable performance statistics. The practice and quiz modes share the same screen and make use of the same set of piano sound sample resources.

The quiz mode also inputs user performance data into a simple database. The performance statistics view retrieves information from the database to be presented to the user on demand.

Figure 21 is the flowchart for the default practiced mode. Below is the flowchart for the quiz mode.

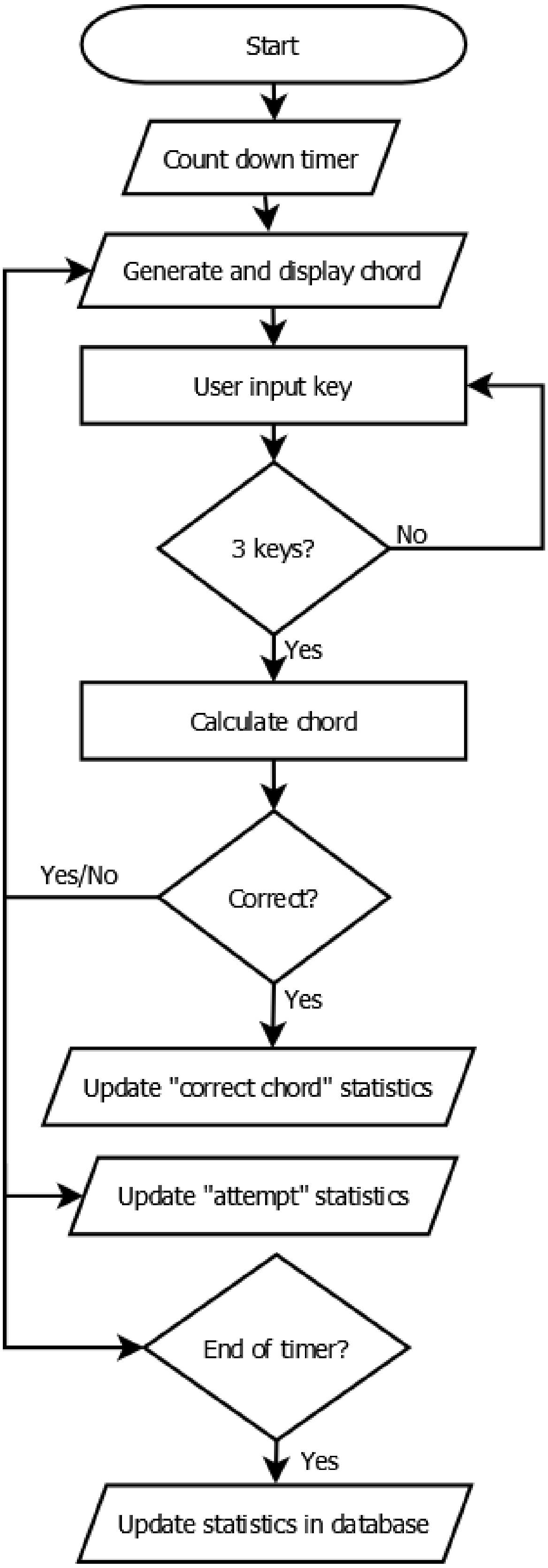
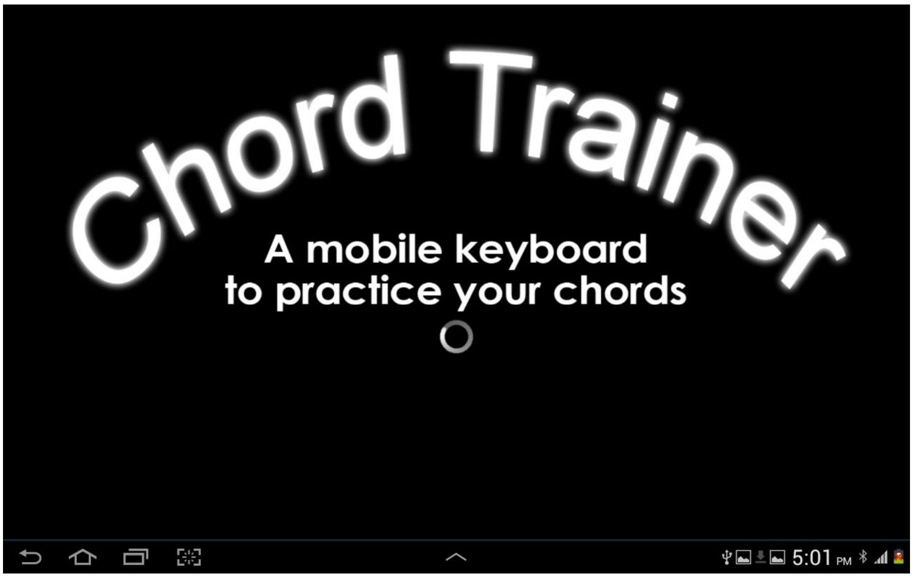

Start
Count down timer
Generate and display chord
User input key
3 keys?
No
Yes
Calculate chord
Correct?
Yes/No
Yes
Update "correct chord" statistics
Update "attempt" statistics
End of timer?
Yes
Update statistics in database

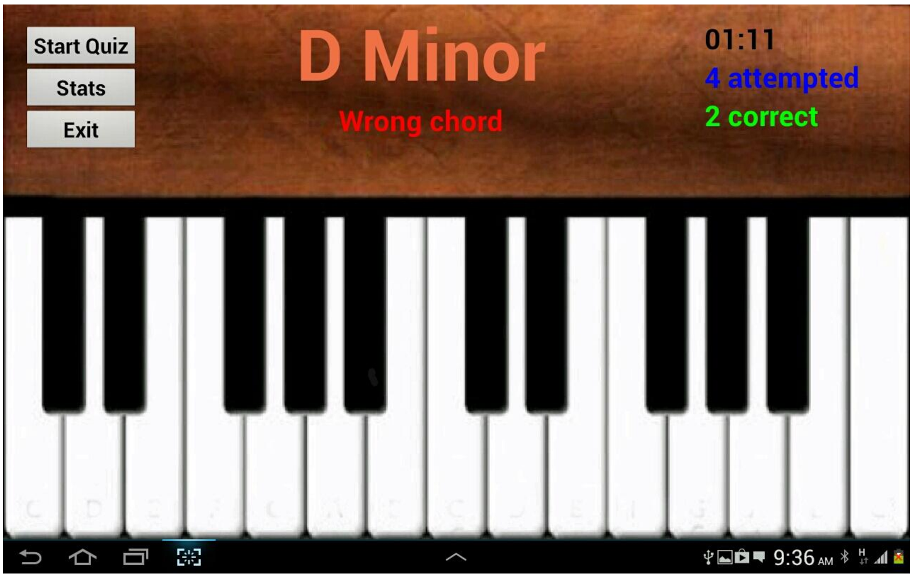
Figure 27 Flowchart for quiz mode



Chord Trainer

A mobile keyboard
to practice your chords

Figure 28 The proper splash screen



D Minor

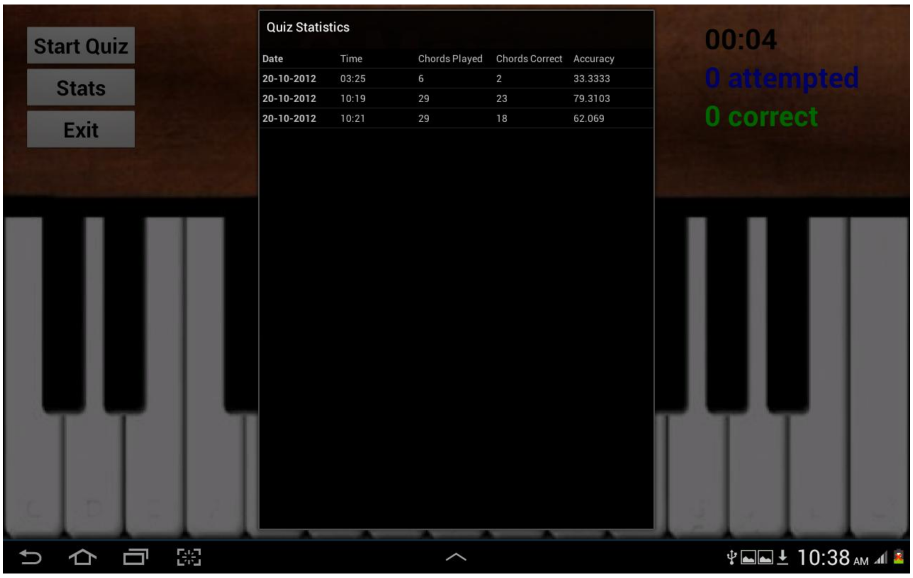
Wrong chord

01:11

4 attempted

2 correct

Figure 29 Buttons added to the existing piano interface



Quiz Statistics

Date	Time	Chords Played	Chords Correct	Accuracy
20-10-2012	03:25	6	2	33.3333
20-10-2012	10:19	29	23	79.3103
20-10-2012	10:21	29	18	62.069

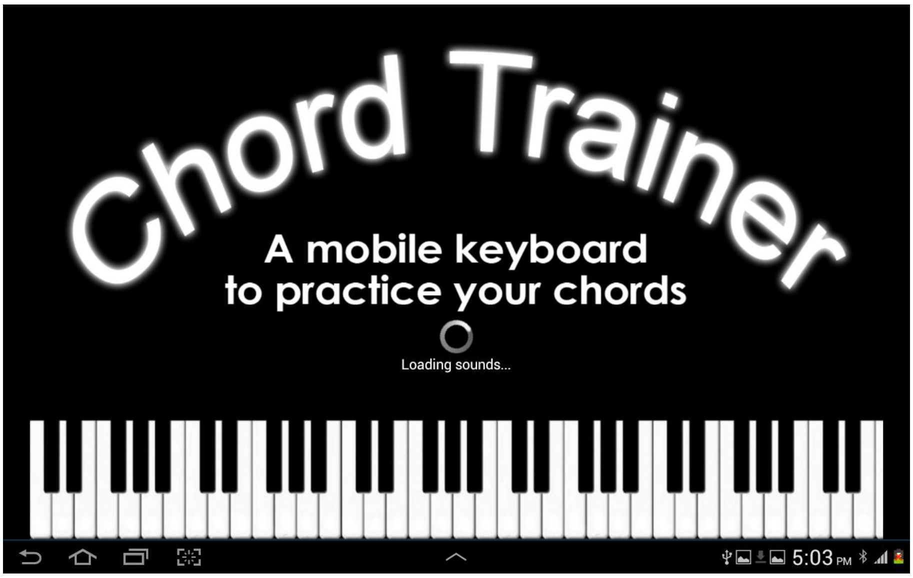
Figure 30 Pop-up statistics view

4.3.4 Iteration 4

After the user testing, as well as another meeting with the supervisor, the prototype was tweaked, more for aesthetical enhancement than functional development

The splash screen was changed into a proper image overlaid under the loading spinner. The text display indicating to the user if they played a chord correctly or wrongly was animated to fade away after 1 second’s message display. The font colour of the chord name display was changed into a lighter shade for more contrast as the supervisor thought the original colour looked a little blended into the background. The pop-up statistics view was also widened, with enlarged font size.

One functional development was the exit button. It seems that the user wants an explicit confirmation that they are exiting the application.

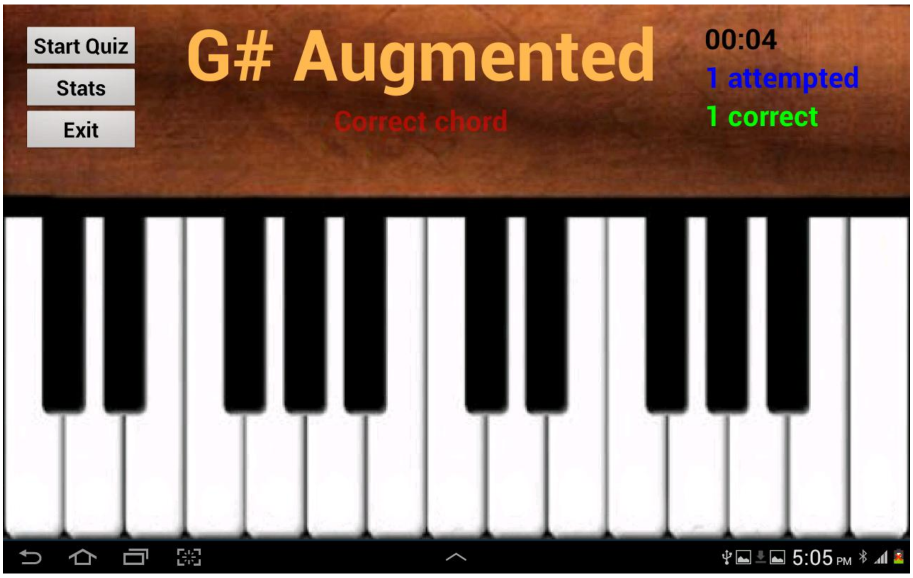


Chord Trainer

A mobile keyboard to practice your chords

Loading sounds...

Figure 31 Revised splash screen



G# Augmented

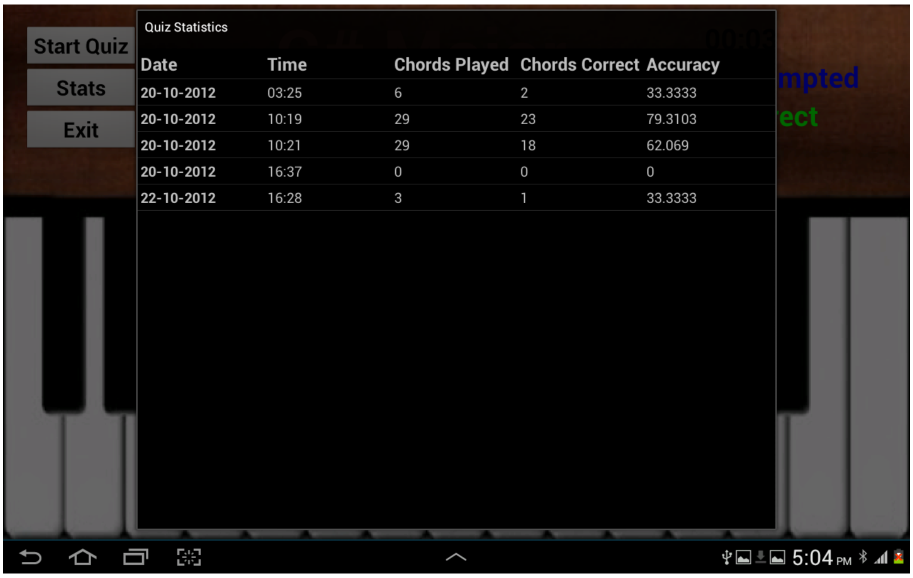
Correct chord

00:04

1 attempted

1 correct

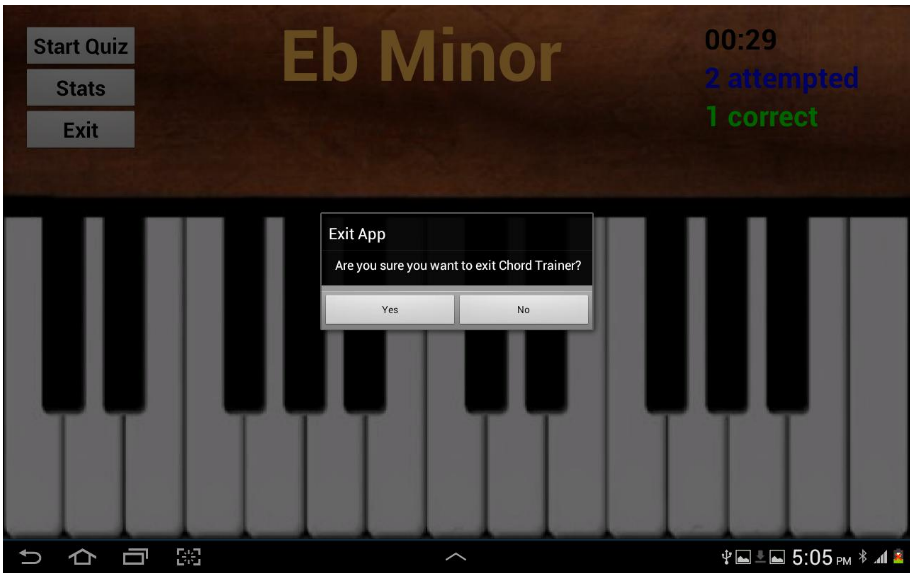
Figure 32 A lighter shade of colour for chord text display; correct or wrong chord message display animated to fade out



Quiz Statistics

Date	Time	Chords Played	Chords Correct	Accuracy
20-10-2012	03:25	6	2	33.3333
20-10-2012	10:19	29	23	79.3103
20-10-2012	10:21	29	18	62.069
20-10-2012	16:37	0	0	0
22-10-2012	16:28	3	1	33.3333

Figure 33 Widened pop-up statistics view with larger font sizes



Exit App

Are you sure you want to exit Chord Trainer?

Figure 34 Exit confirmation on click of the exit button

4.3.5 Prototype Iterations Summary

Iteration	**Features added**	**Remarks**
1	- piano interface - selection of piano keys - chord calculation algorithm - orientation lock: landscape	- prototype was FirstApp's extension - prototype was run on AVD, and hence only single touch could be implemented
2	- text displays - multiple piano key touches - sound on piano key touch - random generation of chords - splash screen	- splash screen was added because sound resources take time to load - prototype was ported over as a new Android app project
3	- layout of buttons - quiz mode button - quiz statistics button - revision on chord calculation algorithm - addition of splash screen image	- quiz mode was added to let users assess their own progress - prototype was put out for user testing
4	- revision on app graphics and layouts - animation of message display - exit button and exit confirmation	

Table 3 Summary of prototype iterations

4.4 User Perception Study

The results of the user perception questionnaire given out to 30 beginner musicians were compiled and analysed. The first question was asked to determine the popularity of Android devices amongst the respondents. The next 2 questions were asked to find out about the respondents music background, the next 3 to find out their music abilities, and the other 4 questions to find out their acceptance towards training sight-reading or chord-reading skills using mobile technology.

4.4.1 Do you own an Android device?

This question was asked to see how many respondents use Android devices, and hence are potential users of the Chord Trainer application.

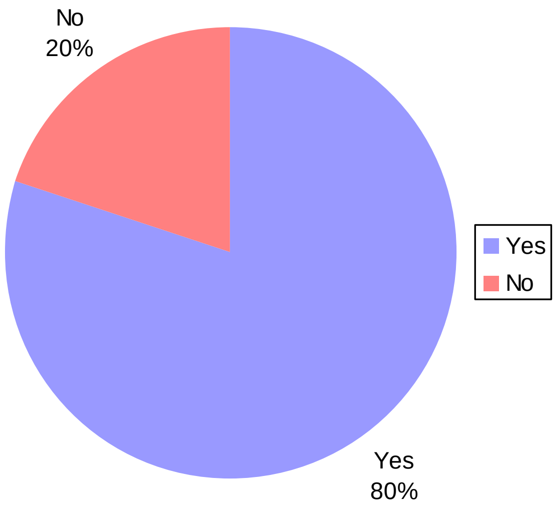


Figure 35 Pie chart for “Do you own an Android device?”

Figure 35 shows that 24 people or 80% of the respondents own an Android device. This means the application is quite marketable amongst beginner musicians.

4.4.2 For how long have you played the piano/keyboard?

This question aimed to find out the level of music-playing skills of the respondents based on their years of experience. Although this number may not accurately determine the level of music-playing skills, as different people have different learning speed, generally a player with accumulated playing years of about 3 to 4 years can be considered at the intermediate level.

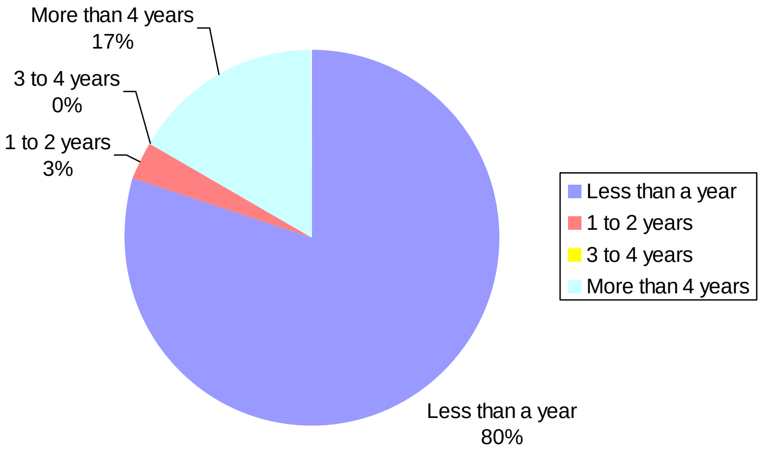


Figure 36 Pie chart for “For how long have you played the piano/keyboard?”

It can be seen from Figure 36 that 24 out of 30 respondents have experience of less than a year, which means that they are sufficiently beginner keyboardists and target users of the Chord Trainer application. 1 has played the piano/keyboard for 1 to 2 years; this duration can be considered the upper limit of the beginner keyboardist label. None has experience of 2 to 3 years, which are considerably intermediate players who may or may not have chord-reading skills beyond trainable using the application. 5 have played the piano/keyboard for more than 4 years.

4.4.3 How did you pick up the piano/keyboard?

This question was meant to find out how many of the respondents are formally trained pianists/keyboardists. This is because the piano/keyboard may not be their first or primary instrument, meaning they may not be the best at it compared to the instrument that they play in the orchestra. However, whether formally trained, and whether being their best instrument, do not guarantee that they can play well reading chord charts. Hence, in a way, this question is to find out how many formally trained pianists/keyboardists are able to play reading chord charts (to be complemented with the next question).

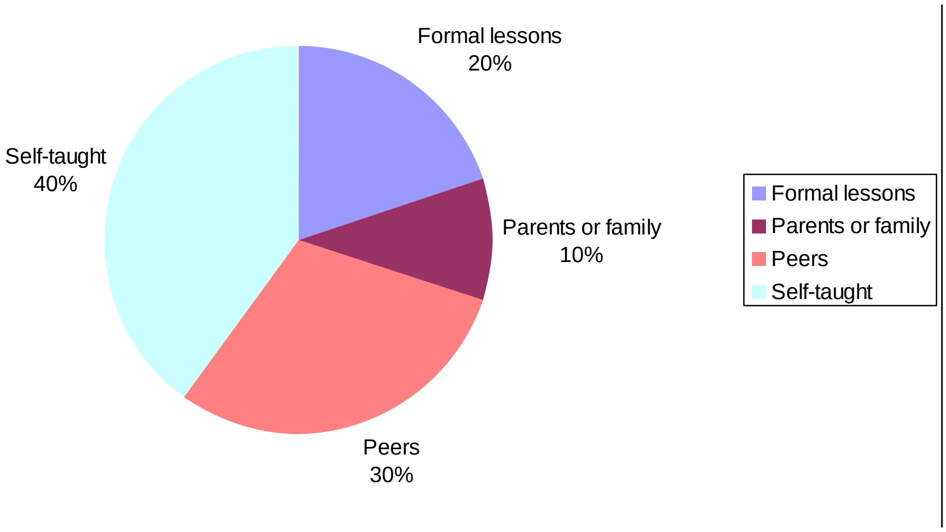


Figure 37 Pie chart for “How did you pick up the piano/keyboard?”

Figure 37 shows that 12 respondents are self-taught. This means that their orchestral instrument is probably their primary instrument, and they play the piano/keyboard out of interest or curiosity. 9 respondents are taught by peers, also a method that is not considered formal training, and possibly out of interest or curiosity as well. 3 are taught by parents or family members, who may or may not be professional music teachers; and only 6 out of 30 take formal lessons.

4.4.4 Could you play songs without formal music scores but with just chord charts?

This question complemented the previous questions: to find out how many self-taught pianists/keyboardists can actually play reading chord charts.

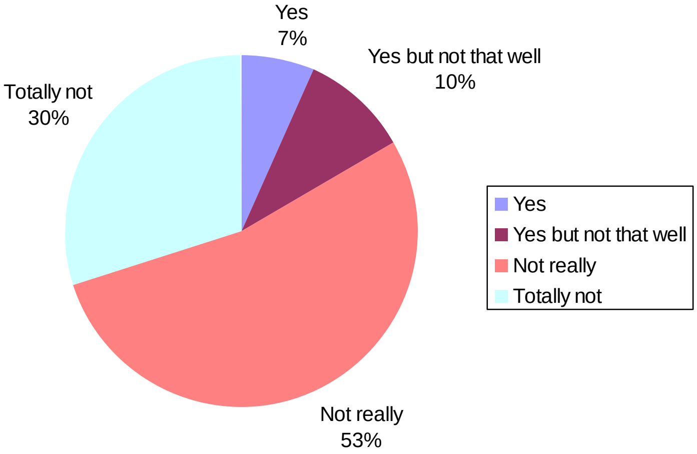


Figure 38 Pie chart for “Could you play songs without formal music scores but with just chord charts?”

Given the answers from the previous question, it is not surprising to see from Figure 38 that 16 people could not really play by reading just chord charts. This could be due to the lack of music theory knowledge to play chords, or the inadequacy of chord knowledge to play a full song. In any case, this could be improved through practice, and perhaps a brief music theory lesson. 3 people could play songs without formal music scores but not that well; they would be the perfect candidates to make use of the Chord Trainer application to sharpen their skills. 2 people said yes, they could play with just chord charts. 9 people totally could not play by reading just chord charts.

4.4.5 Rate your sight-reading skills.

This question gave the respondents a chance to rank themselves on their sight-reading skills regardless of their years of experience and method of learning the piano/keyboard. This sight-reading component encapsulates chord-reading skills.

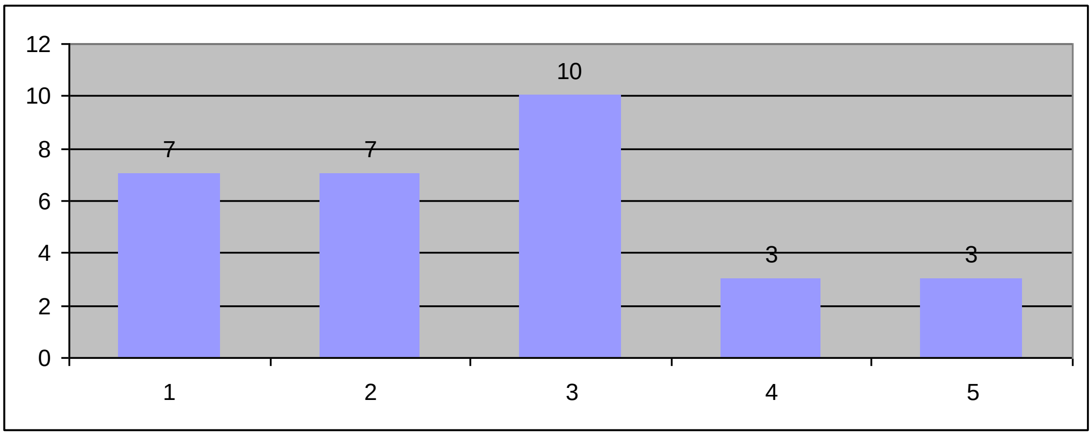


Figure 39 Bar chart for "Rate your sight-reading skills."

Based on Figure 39, on a scale of 1 to 5, with 1 being bad and 5 being excellent, only 3 people think they have excellent sight-reading skills. 3 people ranked themselves at 4, which means good. 10 people rated themselves 3, and 7 each ranked themselves 2 and 1, indicating that they are not very confident about their sight-reading skills. These are the people who could make use of the Chord Trainer application to improve their sight-reading (specifically chord-reading) skills.

4.4.6 Rate your improvisation skills.

Following their rating of their sight-reading skills, respondents are also given a chance to rank their own improvisation skills. This skill determines if they are able to "play out of the box" with a given chord, meaning to play different chord variations.

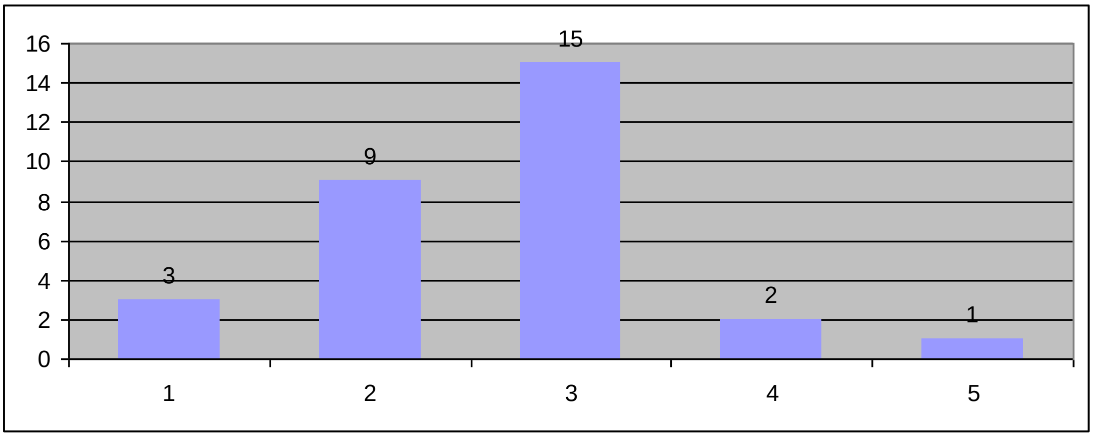


Figure 40 Bar chart for "Rate your chord-reading skills."

As seen from Figure 40, despite the ranking of their sight-reading skills, the respondents mainly think they could improvise averagely: 15 people ranked themselves 3 out of a scale from 1 to 5, with 5 being excellent and 1 being bad. 9 people ranked themselves 2. 3 people ranked themselves 1. 2 people rated themselves 4, and only one person thought their improvisation is excellent.

This can be interpreted as: the respondents do not think they have very good sight-reading skills, but when they can sight-read, they think they can improvise quite well.

4.4.7 Do you think sight-reading or chord-reading skills can be trained using a mobile application?

This question was to find out if beginner musicians accept mobile technology as a platform to train their sight-reading or chord-reading skills.

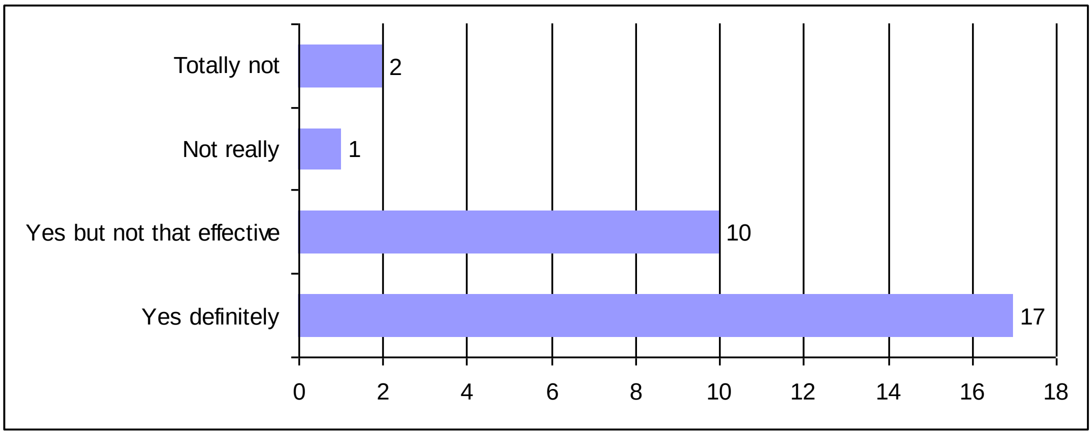


Figure 41 Bar chart for “Do you think sight-reading or chord-reading skills can be trained using a mobile application?”

From Figure 41, it can be seen that beginner musicians generally accept using a mobile application to train their sight-reading or chord-reading skills, with 17 people saying yes, definitely. 10 people thought that mobile application is useful to for training, but it would not be that effective. 1 person thought that training using a mobile application would not be helpful; 2 people thought it would not be helpful at all.

4.4.8 Would you voluntarily use a mobile application to train your sight-reading or chord-reading skills?

After knowing if they accept using mobile technology to training their music skills, this question was to find out if they would voluntarily use mobile technology for their training.

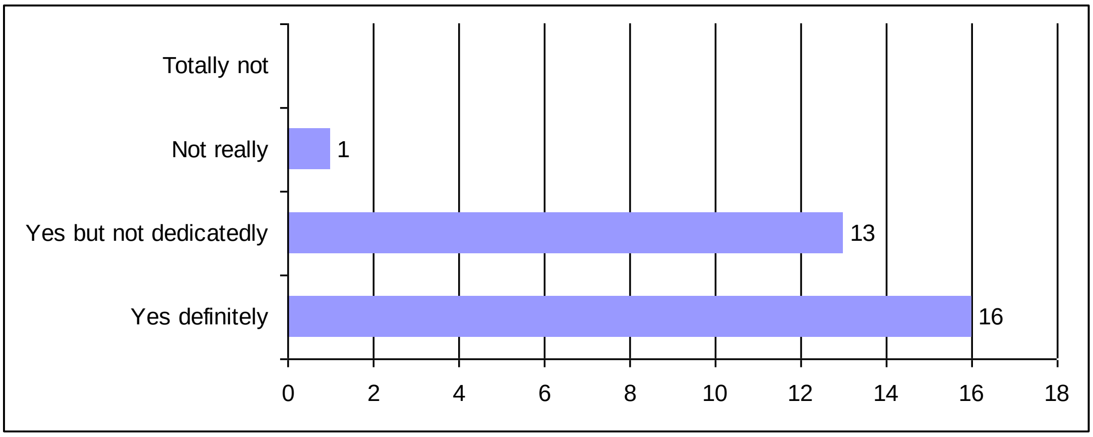


Figure 42 Bar chart for “Would you voluntarily use a mobile application to train your sight-reading or chord-reading skills?”

Figure 42 shows that 16 respondents would voluntarily make use of a mobile application to train their sight-reading or chord-reading skills. 13 people would use it, but not dedicatedly. This means that they would also incorporate other methods of training aside from using a mobile application. Only 1 person would not voluntarily use a mobile application for their training. No respondents totally rejected the usage of a mobile application for their training.

4.4.9 Would you specifically use Chord Trainer to train your sight-reading or chord-reading skills?

This question is asked ultimately to find out if the Chord Trainer application is a helpful application to beginner musicians.

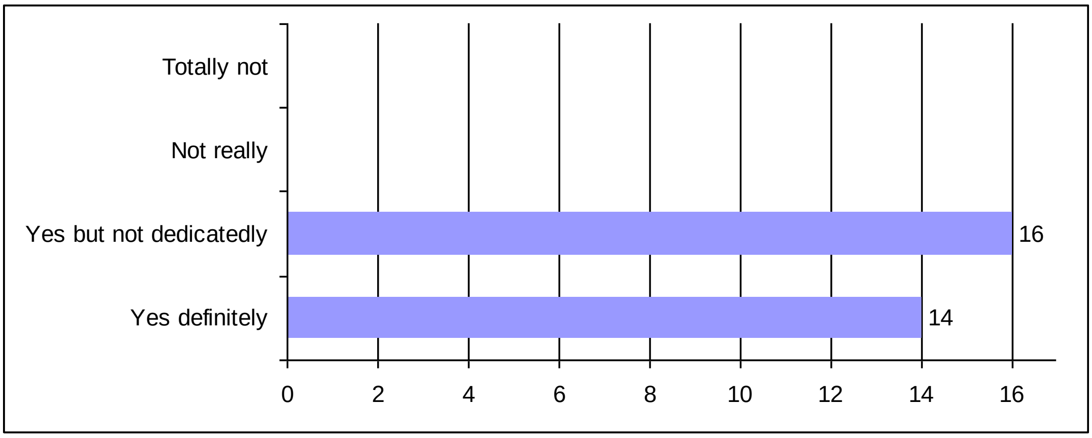


Figure 43 Bar chart for "Would you specifically use Chord Trainer to train your sight-reading or chord-reading skills?"

It can be seen from Figure 43 that all the respondents thought that the Chord Trainer application is useful to them. 16 people would use it in the training though not dedicatedly. The other 14 people thought yes, they would definitely use it to train themselves.

4.4.10 Other comments

This open-ended question let users voice out other suggestions that they might have. A user that suggested that the app could "…include Bluetooth connection to… [allow for] …'chord battle' with other users."

Some comments pointed out issues not covered in the project scope. These included: the inclusion of a tutorial as well as provision of corrections when the user plays a chord wrongly. These are good suggestions. However, it is stated in the project scope that this project does not teach beginner keyboardists to play, but only to let them practice. Hence features based on the suggestions are not inserted in Iteration 4 of the prototype.

Comments that are rather irrelevant to the application, but more targeted at the hardware included: "the tablet lagged", and, "cannot reset."

Other positive comments are: "Nice application! First in the market, I think this can help me a lot," and "Good app!"

4.5 Usability Study

The results of the usability study conducted using the System Usability Scale were compiled, and calculated according to the SUS article to make the final result meaningful.

According to the SUS article, the rating for individual questions alone is not meaningful. The scale gives a single number, ranging between 0 and 100, that measures the overall usability of the system being rated (Brooke, n.d., Scoring SUS).

However, to calculate the final single number representing the composite measurement of usability, a series of calculation first needs to be done on the individual questions.

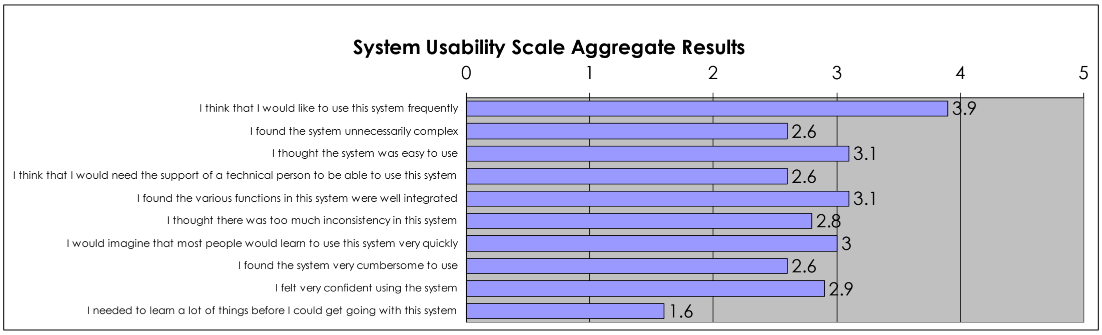


Figure 44 Averaged score for all 10 items from 30 questionnaires

Figure 44 shows the averaged score for each item on the questionnaire. The score for each item was calculated using the method given by John Brooke in the SUS article. For odd-number questions, the score was calculated by taking the given rating and subtracting one. For even-number questions, the score was obtained by subtracting the given rating from 5. After that, all the scores for each question from all 30 respondents were added together and divided by 30.

From there, further calculation was done to acquire the ultimate single SUS score. The averaged score for all 10 items were totaled up and multiplied by 2.5. Hence the SUS score for the Chord Trainer application is 28.2 × 2.5 = 70.5 out of 100. This means that Chord Trainer is quite a usable application.

CHAPTER 5
CONCLUSIONS AND RECOMMENDATIONS

Music training via mobile technology is proving to be not just acceptable, but quite popular amongst beginner musicians. The Chord Trainer application seems to be quite a usable application, and helpful to train beginner keyboardists on their chord-playing.

The objectives of the project are achieved. Music training via the multi-touch capabilities of a tablet device was attained through the Chord Trainer application. The Chord Trainer application is the structured and goal-oriented virtual keyboard mobile application that allows beginner keyboardists to train their chord-playing skills, as listed as the second objective. A user perception study was conducted on potential users, who are beginner keyboardists owning Android devices. Through this study, it is found that using mobile technology for music training is generally acceptable to beginner musicians. A usability study was also conducted on the Chord Trainer application. The score of the study implies that the Chord Trainer application is quite a usable application, and helpful to train beginner keyboardists.

For future work on the application, the application could be expanded to include multiple users on the application, so that several users could use the same device to train their chord-playing and keep track of their own progress. Also, as suggested by the survey respondents, the application could use a simple tutorial to refresh the beginner keyboardists on their knowledge when they need to. The application could also provide corrections to strengthen the user’s chord knowledge.

As a continuation of the project research, the user’s chord-playing progress using the Chord Trainer application could be monitored to see if there is any significant improvement. This information would be useful to music education field. If results are positive, similar applications could even be developed for other instruments, such as strings, guitar and drums. However, it should be noted that the size of the application should be given attention to allow for realistic simulation of real playing.

Lastly, the Chord Trainer application could be a first step in incorporating mobile technology in the regular training of beginner musicians.

REFERENCES

automatSoftware. (2012, March 12). Chord Trainer: Overview. In *Google Play*. Retrieved June 27, 2012, from https://play.google.com/store/apps/details?id=com.john.pianoapp

Brooke, J. (n.d.). *SUS – A quick and dirty usability scale.* United Kingdom.

Building Your First App. (n.d.). Retrieved June 29, 2012, from Android Developers website: http://developer.android.com/training/basics/firstapp/index.html

Cernoch, C. (n.d.). *ChordTrainer.com*. Retrieved June 27, 2012, from http://chordtrainer.com/

Chiang, C. W., Chiu, S. C., Dharma, A. A. G., Tomimatsu, K.. (2012, February 19). *Birds on Paper: An Alternative Interface to Compose Music by Utilizing Sketch Drawing and Mobile Device.* Ontario, Canada.

Comparison of Android devices. (n.d.). In *Wikipedia.* Retrieved July 23, 2012 from http://en.wikipedia.org/wiki/Comparison_of_Android_devices

Delaney, R. (2010, March 22). APA Citation Style. *Publication Manual of the American Psychological Association,* (6th ed). Retrieved from Lond Island University Post - B. Davis Schwartz Memorial Library website: http://www2.liu.edu/cwis/cwp/library/workshop/citapa.htm

Developer Tools. (n.d.). Retrieved June 29, 2012, from Android Developers website: http://developer.android.com/tools/index.html

Eurhythmics. (n.d.). In *Wikipedia.* Retrieved June 25, 2012, from http://en.wikipedia.org/wiki/Eurhythmics

Eno, B. (n.d.). *Generative Music.* Retrieved December 1, 2012, from http://www.generativemusic.com/bloom.html

Fake book. (n.d.). In *Wikipedia.* Retrieved June 27, 2012 from http://en.wikipedia.org/wiki/Fake_book

Farber, A., & Thomsen, K. (2011). *The History of Dalcroze.* Retrieved June 26, 2012, from The Dalcroze Society of America website: http://www.dalcrozeusa.org/about-us/history

Hamlin, J. (n.d.). *Should My Child Learn to Improvise?* Retrieved June 26, 2012, from San Francisco Classical Voice website: http://www.sfcv.org/kids-family/tips-advice/teaching-methods

Internation Suzuki Association. (n.d.). *The Suzuki Method.* Retrieved June 24, 2012, from http://internationalsuzuki.org/method.htm

iPad Dimensions and Features. (n.d.). Retrieved July 23, 2012, from iPad Dimensions and Features website*:* http://www.ipaddimensions.org/

IQ-mobile. (n.d.). IQ Piano Chords 1.0. In *Brothersoft Mobile.* Retrieved October 28, 2012 from http://mobile.brothersoft.com/iq-piano-chords-9203.html

lange681. (2011, December 18). Adele some one like you piano tutorial on ipad [Video file]. Retrieved July 23, 2012, from http://www.youtube.com/watch?v=Hn1DUZ1Ttis

Musical improvisation. (n.d.). In *Wikipedia.* Retrieved June 25, 2012, from http://en.wikipedia.org/wiki/Musical_improvisation

Musical keyboard. (n.d.). In *Wikipedia*. Retrieved July 23, 2012 from http://en.wikipedia.org/wiki/Musical_keyboard

Pausing and Resuming an Activity. (n.d.). Retrieved October 27, 2012, from Android Developers website: http://developer.android.com/training/basics/activity-lifecycle/pausing.html

Platform Versions. (n.d.). Retrieved June 28, 2012, from Android Developers website: http://developer.android.com/about/dashboards/index.html

Products - Chord Practice Buddy. (n.d.). Retrieved June 30, 2012, from KeyPiano.com website: http://www.keypiano.com/products.php?ACT=showProduct&id=9

Rubin, A. (2012, February 27). Google+ post Andy Rubin. In *Google+*. Retrieved June 28, 2012 from https://plus.google.com/u/0/112599748506977857728/posts/Btey7rJBaLF

Schulz, F., Geiger, C., Reckter, H. (2009, October 29). *ANTracks – generative mobile music composition.* Athens, Greece.

Sight-Reading Techniques. (n.d.). Retrieved June 24, 2012, from Music Education Center website: http://music.thefxcode.com/sight.htm

Souvey, C. (2012, March 6). Musical Piano FREE: Overview. In *Google Play.* Retrieved October 28, 2012 from https://play.google.com/store/apps/details?id=souvey.musical&hl=en

Starting an Activity. (n.d.). Retrieved October 27, 2012, from Android Developers website: http://developer.android.com/training/basics/activity-lifecycle/starting.html

Steinbuhler, D. (2012). *Our Research.* Retrieved August 1, 2012, from Steinbuhler & Company website: http://www.steinbuhler.com/html/our_research.html

Suzuki method. (n.d.). In *Wikipedia*. Retrieved June 24, 2012, from http://en.wikipedia.org/wiki/Suzuki_method

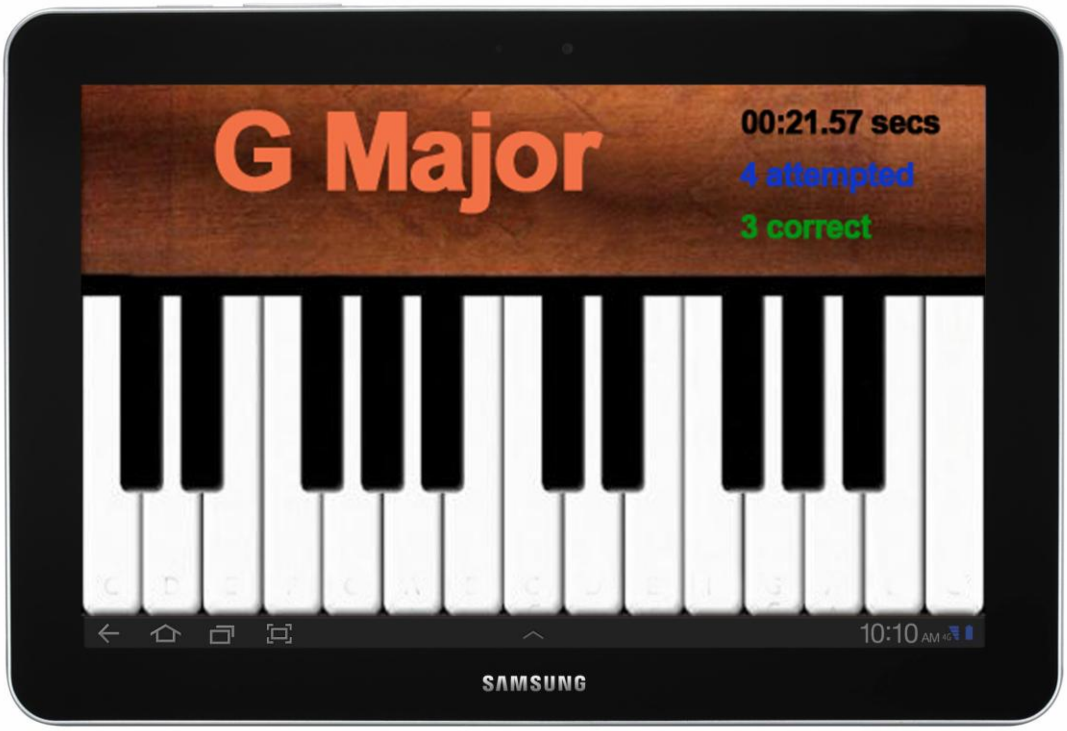
Thornton, J. (n.d.). *Play the piano on your mobile.* In *Softtonic: Smart Piano*. Retrieved October 28, 2012, from http://smart-piano.en.softonic.com/windowsmobile

Umito. (2012, July 04). KeyChord – Piano Chords/Scales: Overview. In *Google Play*. Retrieved June 30, 2012 from https://play.google.com/store/apps/details?id=umito.android.keychord

APPENDICES

APPENDIX A

PRELIMINARY WIREFRAMES OF APPLICATION

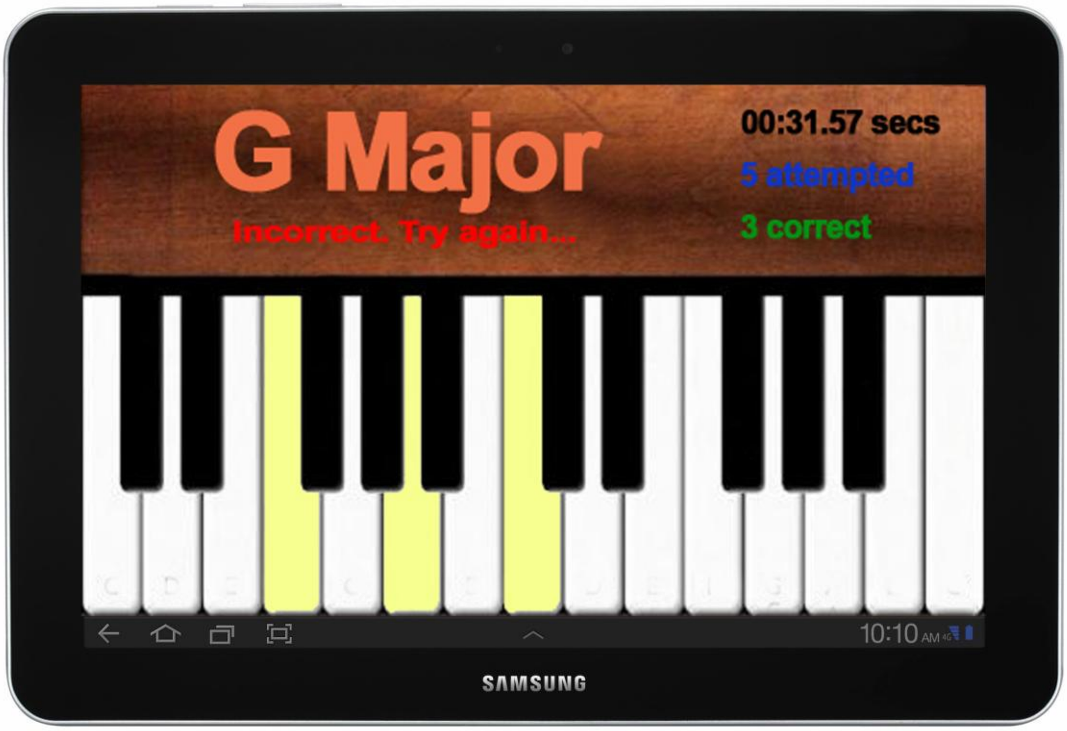


G Major

00:21.57 secs

4 attempted

3 correct



G Major

Incorrect. Try again...

00:31.57 secs

5 attempted

3 correct

APPENDIX B

USER PERCEPTION STUDY QUESTIONNAIRE

1. Do you own an Android device?

Yes No

2. For how long have you played the piano/keyboard?

a. less than a year

b. 1 to 2 years

c. 3 to 4 years

d. More than 4 years

3. How did you pick up the piano/keyboard?

a. formal lessons

b. parents or family

c. peers

d. self-taught

4. Could you play songs without formal music scores but with just chord charts?

a. yes

b. yes but not that well

c. not really

d. totally not

5. Rate your sight-reading skills.

Excellent 5 4 3 2 1 Bad

6. Rate your improvisation skills.

Excellent 5 4 3 2 1 Bad

7. Do you think sight-reading or chord-reading skills can be trained using a mobile application?

a. yes definitely

b. yes but not that effective

c. not really

d. totally not

8. Would you voluntarily use a mobile application to train your sight-reading or chord-reading skills?

a. yes definitely

b. yes but not dedicatedly

c. not really

d. totally not

9. Would you specifically use Chord Trainer to train your sight-reading or chord-reading skills?

a. yes definitely

b. yes but not dedicatedly

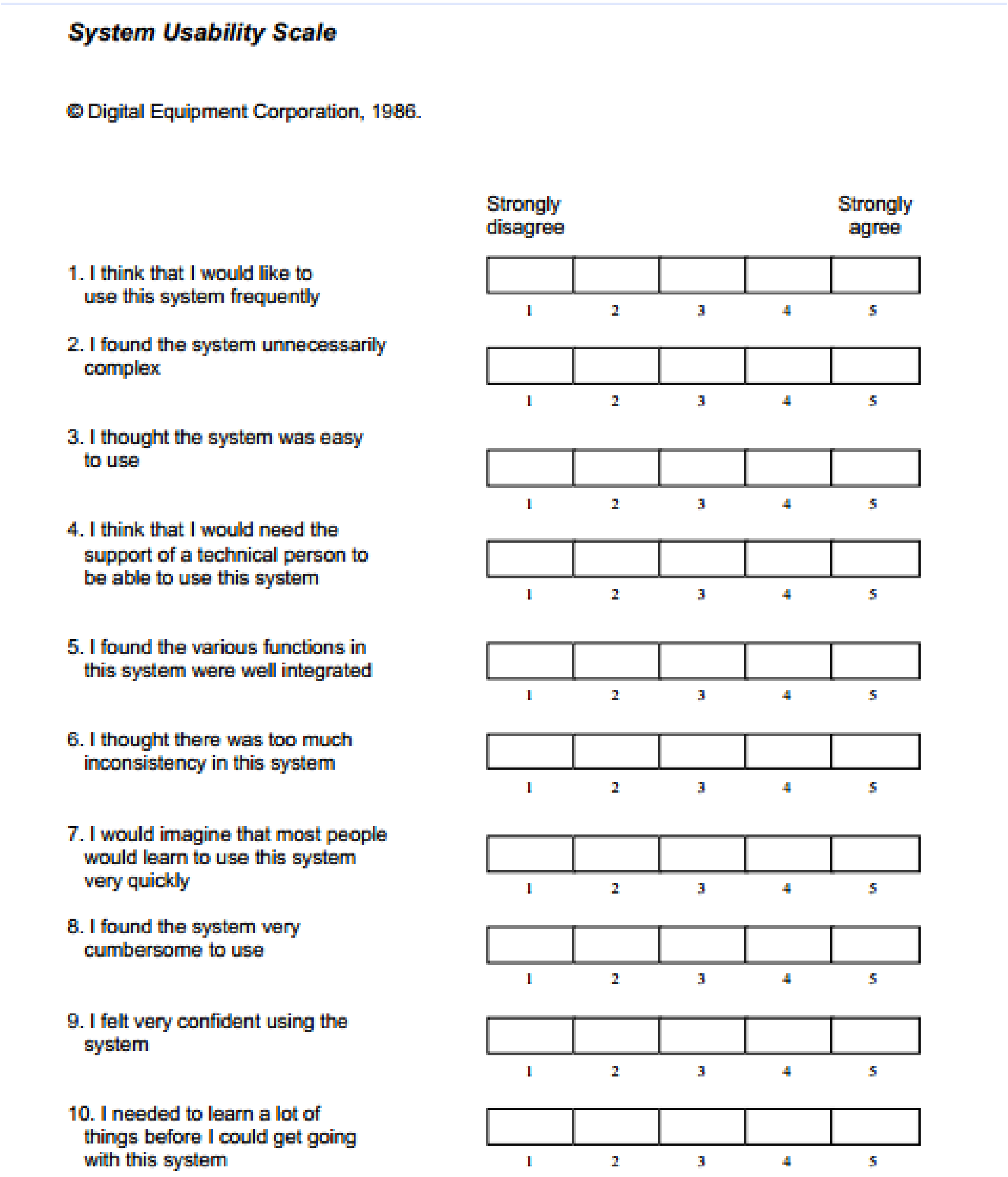
c. not really

d. totally not

10. Any other comments.

APPENDIX C

SYSTEM USABILITY SCALE QUESTIONNAIRE



System Usability Scale

Strongly disagree — Strongly agree

1. I think that I would like to use this system frequently
□ □ □ □ □
1 2 3 4 5

2. I found the system unnecessarily complex
□ □ □ □ □
1 2 3 4 5

3. I thought the system was easy to use
□ □ □ □ □
1 2 3 4 5

4. I think that I would need the support of a technical person to be able to use this system
□ □ □ □ □
1 2 3 4 5

5. I found the various functions in this system were well integrated
□ □ □ □ □
1 2 3 4 5

6. I thought there was too much inconsistency in this system
□ □ □ □ □
1 2 3 4 5

7. I would imagine that most people would learn to use this system very quickly
□ □ □ □ □
1 2 3 4 5

8. I found the system very cumbersome to use
□ □ □ □ □
1 2 3 4 5

9. I felt very confident using the system
□ □ □ □ □
1 2 3 4 5

10. I needed to learn a lot of things before I could get going with this system
□ □ □ □ □
1 2 3 4 5

APPENDIX D

TECHNICAL PAPER