COMPUTER ASSISTED MUSIC PRACTICE FOR GUITAR (CAMPGuitar)

BY

TUNG PHIEN PHIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2013

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: **COMPUTER ASSISTED MUSIC PRACTICE FOR GUITAR (CAMPGuitar)**

Academic Session: **JAN 2013**

I **TUNG PHIEN PHIN**

declare that I allow this Final Year Project Report to be kept in

Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

_____ (Author's signature)

Address:

31, Solok Nakhoda,

Taman Teluk Air Tawar,

13050, Butterworth, Penang

Date: _____

_____ (Supervisor's signature)

SOONG HOONG CHENG

Supervisor's name

Date: _____

COMPUTER ASSISTED MUSIC PRACTICE FOR GUITAR (CAMPGuitar)

BY

TUNG PHIEN PHIN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfilment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

JAN 2013

DECLARATION OF ORIGINALITY

I declare that this report entitled "**COMPUTER ASSISTED MUSIC PRACTICE FOR GUITAR (CAMPGuitar)** is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____

Name : _____

Date : _____

ACKNOWLEDGEMENTS

I would like to thank my project supervisor, Mr. Soong Hoong Cheng for giving guidance on this project. I am very appreciating Mr. Soong in helping me with his patience and enthusiasm throughout this project. I would also like to thank to my moderator, Mr Sun Teik Heng @ San Teik Heng who pay attend on this project progression. I have exposed many new knowledge throughout this project, including programming languages, techniques in develop a new system and creating a new project personally. I have gain a lot of knowledge which improve me and this will help me in my future career, and throughout rest of my life.

Other than that, I would also thank to my parents for their support, encouragement and love. They always taught me to be a positive thinker, which bring hope to me and my future. In addition, I would like to thank them giving me the opportunity for my further studies in Universiti Tunku Abdul Rahman.

Last but not least, I would like to thank God for giving me hope and energy to complete this project on time. I do face problems throughout this project but I believe this is the enemy of knowledge that given by God. Thank God.

ABSTRACT

This project proposes a method for game-based learning using an acoustic guitar. The game-based learning systems have been considered very effective in learning and gaining knowledge. This project introduces a platform for acoustic guitar lessons that has the ability to lead beginner guitarists to achieve guitar playing accuracy. This project studies the Fast Fourier Transform which used in the creation of guitar tuner. This project is using computer audio input for the guitar input, the guitar must be a pluggable guitar.

Based on the research, found that currently there is no combination of game-based learning and qualified examinations. To be honest, this might be an awesome combination to achieve the learning results. Existing game-based learning is simply attractive and able to gain knowledge, but it does not fully utilize the game-based learning system as in no further improvement other than just entertaining the gamers.

In order to solve the problems, few solutions have been discovered to further improve the game-based learning system in guitar. The combination of game-based learning for the exam-related guitar songs has a bright potential to benefit guitar players throughout the world.

TABLE OF CONTENTS

DECLARATION OF ORIGINALITY ii

ACKNOWLEDGEMENTS iii

ABSTRACT iv

TABLE OF CONTENTS v

LIST OF FIGURES viii

LIST OF TABLES x

LIST OF ABBREVIATIONS xi

CHAPTER 1 INTRODUCTION 1

1.1 Problem Statement and Motivation 2

1.1.1 Existing Rhythm Games Do Not Contain Any Exam-Related Lessons 2

1.1.2 Existing Rhythm Game Is Inconvenient In Setting Up 2

1.1.3 Existing Rhythm Games Do Not Tell the Weaknesses Of The Players In Terms of Examination 4

1.1.4 Fails In Learning Guitar With The Incorrect Steps 4

1.2 Objective 5

1.3 Project Scope 7

1.4 Contribution 8

1.5 Background Information 9

CHAPTER 2 LITERATURE REVIEW 11

2.1 Rhythm Game : Rocksmith 11
2.2 MIDI (Musical Instrument Digital Interface) 12
2.3 Game-based Learning 13
2.4 Fast Fourier Transform (FFT) 14
CHAPTER 3 METHODOLOGIES AND TOOLS INVOLVED 17
3.1 Methodologies 17
3.2 Skills/Technologies involved 20
3.2.1 XNA Game Studio 20
3.2.2 C# Programming Language 20
3.2.3 Sonuus G2M Guitar to MIDI Converter 21
3.2.4 Stereo Jack Adaptor 3.5mm Male to 6.5mm Female 22
3.3 Timeline 23
CHAPTER 4 SYSTEM REQUIREMENTS 24
4.1 Requirement Specification 24
4.1.1 User Requirements 24
4.1.2 System Requirements 24
4.1.3 Functional Requirements 25
4.1.4 Non-functional Requirements 25
CHAPTER 5 SYSTEM MODELING 26
5.1 STRUCTURAL DIAGRAM 26
5.1.1 Component Diagram 26
5.2 Behavioural Diagram 29

5.2.1 Use Case Diagram....29
CHAPTER 6 SYSTEM DESIGN....30
6.1 Graphic User Interface Design....30
CHAPTER 7 SYSTEM IMPLEMENTATION AND TESTING....34
7.1 Introduction....34
7.2 System Equipment Setup....34
7.3 System Implementation....35
7.3.1 Pitch Detection Implementation....36
7.3.2 Beats Movement Implementation....38
7.3.3 FFT Algorithm Implementation....39
7.4 Implementation Status....41
7.5 Problem Encountered During Implementation....44
7.6 Test Plan....47
7.6.1 Test Organization....47
7.5.2 Test Schedule....48
7.6 Test Strategy....49
7.6.1 Classes of Tests....51
CHAPTER 8 CONCLUSION....56
REFERENCE....58
BIBLIOGRAPHY....60

LIST OF FIGURES

Figure Number	Title	Page
Figure 1.1-F1	Xbox Full Setup	3
Figure 2.1-F1	Rocksmith Gaming Interface	11
Figure 3.1-F1	The Iterative Model	17
Figure 3.2-F1	Sonuus G2M Guitar To MIDI Converter	21
Figure 3.2-F2	Sonuus G2M Guitar To MIDI Converter Ports	22
Figure 3.2-F3	Stereo Jack Adaptor 3.5mm Male to 6.5mm Female	22
Figure 3.3-F1	Project 1 Gantt Chart	23
Figure 3.3-F2	Project 2 Gantt Chart	23
Figure 5.1-F1	Component Diagram for MainMenu	26
Figure 5.1-F2	Component Diagram for LastReplay	27
Figure 5.1-F3	Component Diagram for PlayGame	27
Figure 5.1-F4	Component Diagram for BearDance	28
Figure 5.1-F5	Component Diagram for Utilities	28
Figure 5.2-F1	Use Case Diagram for CAMPGuitar	29
Figure 6.1-F1	The design layout for the menu page	30

Figure 6.1-F2	The rhythm game playing GUI	31
Figure 6.1-F3	Guitar Tuner Screen	32
Figure 6.1-F4	Last Replay Screen	33
Figure 7.2-F1	Equipment needed for the system	35
Figure 7.3-F1	CAMPGuitar audio input element	36
Figure 7.3-F2	CAMPGuitar load music information element	37
Figure 7.3-F3	CAMPGuitar beats movement element	38
Figure 7.3-F4	FFT Buffer Coding Element	39
Figure 7.3-F5	Hamming Window Element	40
Figure 7.3-F6	Maximum Intensity Calculation Element	41
Figure 7.5-F1	“BadImageFormatException” Error Screen	44
Figure 7.5-F2	Change Platform Target Screen	45
Figure 7.5-F3	LoaderLock Error Screen	46
Figure 7.5-F4	Uncheck LoaderLock Screen	47

LIST OF TABLES

Table Number	Title	Page
Table 1.2-T1	Objectives and Problem Solved Relationship	6
Table 1.3-T1	Grade Calculation for Scoring System.	7
Table 2.4-T1	Notes and Fundamental Frequency values.	15
Table 7.4-T1	Implementation Steps for CAMPGuitar	42
Table 7.4-T2	Implementation status for the modules	43
Table 7.6-T1	Test Organization	48
Table 7.6-T2	Test Schedule for CAMPGuitar	49
Table 7.6-T1	Test Function in Unit Testing	52
Table 7.6-T2	Test Component in Integrated Testing	53
Table 7.6 -T3	Test Case in System Testing	55

LIST OF ABBREVIATIONS

MIDI	Musical Instrument Digital Interface
ROM	Read-Only Memory
FM	Frequency Modulation
IDE	Integrated Development Environment
OOP	Object-Oriented Programming
GUI	Graphic User Interface
RGT	Registry of Guitar Tutor
FFT	Fast Fourier Transform

CHAPTER 1 INTRODUCTION

This project Computer Assisted Music Practice for Guitar (CAMPGuitar) is specially designed for beginner guitarists. The aim of this project is to motivate the beginner guitarists to achieve a fun and correct procedure of practicing acoustic guitar.

In order to increase their interest of playing guitar on foundation, CAMPGuitar is designed. CAMPGuitar is a rhythm game based guitar learning system that uses a real acoustic guitar to play, users can just simply plug in their own guitar and enjoy the game.

The main purpose of designing this system is to allow users to play guitar with more fun, and yet able to achieve the correct guitar learning procedure. Practice makes perfect, with this proposed system, users can learn and practice guitar at anytime and anywhere by just preparing their own guitar and computer.

This system is providing simulation of qualified guitar lessons for beginner guitarists who are under grade 3. Other than just playing rhythm games, this system is able to give judgments, this is to allow users to understand their mistakes. Furthermore, score will also be given based on the users performance, once the users reach a scoring standard, the new stages or levels will get unlock and users are allowed to play the next level songs, this is to make sure that the users' current ability has reaches certain level, then only allow them to try harder songs.

In actual rhythm games, gamers are only allowed to play the whole fixed songs, this system allows users to play guitar music with different accompaniments, which they are interested to play with. Last but not least, this system also has some useful built-in guitar tools such as guitar tuner and guitar chords reference.

1.1 Problem Statement and Motivation

1.1.1 Existing Rhythm Games Do Not Contain Any Exam-Related Lessons

Rhythm game is a very entertaining music game, it actually attracted many fans who have interested on music to enjoy the game. However, most of the rhythm games are for entertainment purpose only, they do not contain any exam-related lessons. It is good to know that there are some rhythm games such as Rocksmith is providing a real-time music playing which directly interact with real guitar, this is a very fantastic concept as well. Rocksmith even introduced advanced guitar playing techniques, detailed video explanations, practice and tutorials, which actually brings a lot of benefits to the beginner guitarists.

The worst part is, this game does not contain any exam-related materials. Since the rhythm game has provided so many useful teaching materials to beginner guitarists, therefore developers should give a try on developing the combination of rhythm game and exam-related lessons.

1.1.2 Existing Rhythm Game Is Inconvenient In Setting Up

There are many rhythm games that are available in Xbox and Playstation 3, Rocksmith is one of the examples of Xbox rhythm game. Whenever a newbie wants to play this kind of games, they need to get a full set up of the Xbox.

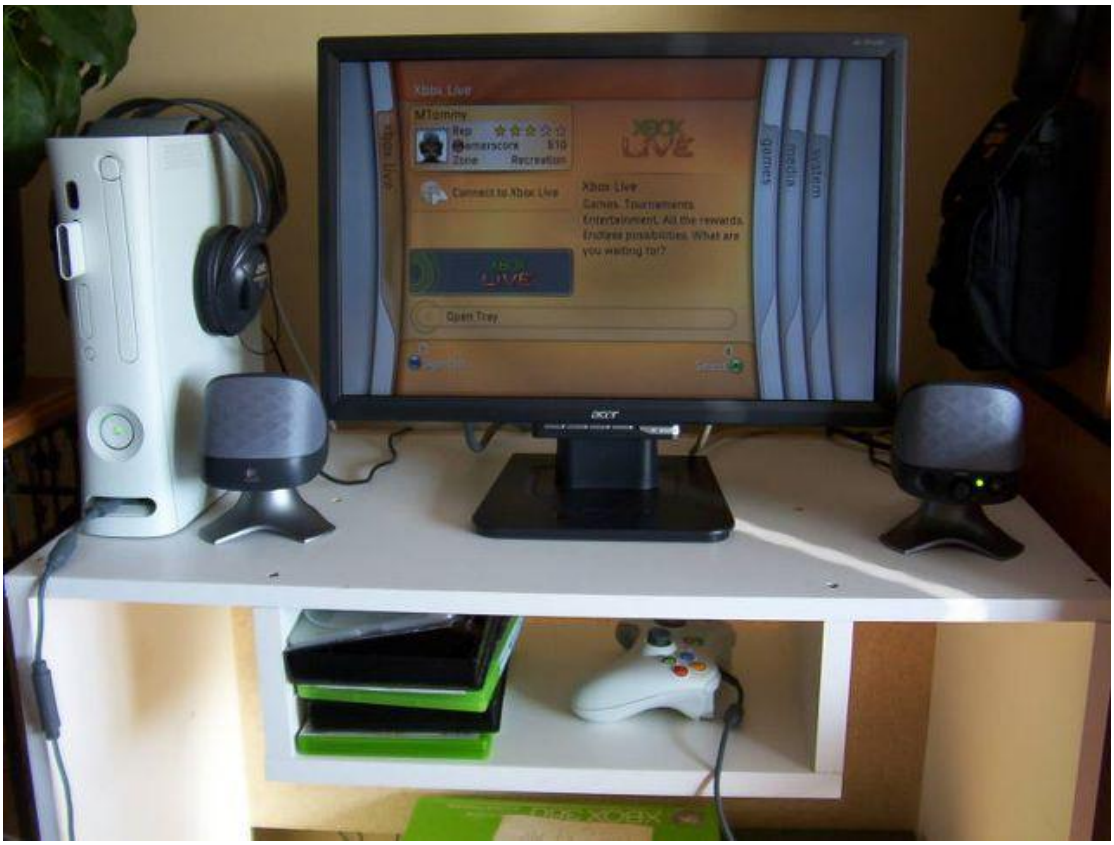


Figure 1.1-F1: Xbox Full Setup

Figure 1.1-F1 shows the full setup of Xbox. In order to set up an Xbox, a monitor and a set of speakers are required. Normally after the Xbox is setup, people will not move it to other places, should say that the Xbox will be fixed at somewhere in their house. Imagine when the beginner guitarists want to practice guitar, they are not allowed to move around or practice at other places, this is so inconvenient. Xbox is too heavy to carry around, therefore beginner guitarists are only able to practice at their house, in a fixed area, which might bring boringness to them.

1.1.3 Existing Rhythm Games Do Not Tell the Weaknesses Of The Players In Terms of Examination

Games are always introducing very entertaining features to attract more players. Most of the time the players just simply play around with the rhythm games, but they actually do not know what is happening on their current ability. The same problem goes to Rocksmith, most of the guitar beginners are so enjoy with the game, but they do not know what their current abilities are, what their mistakes are and what to improve. Although Rocksmith has provided so many useful features to the guitar players, it is still very disappointing that they do not give any judgments, most probably because the game developers know that humans do not like to be judged while having entertainment, therefore they have ignored the judging part.

1.1.4 Fails In Learning Guitar With The Incorrect Steps

Most of the guitar beginners do like to play fast and hurry for a result. They try to rush their practices as fast as they can, but when they face difficulties such as unable to change chords quickly, smoothly or not clean enough, they will start feeling lack of confidence, lack of motivation, and therefore results on loss of interest. On the other hand, guitar beginners also might fail to improve in time due to insufficient practice. Guitar beginners must have enough commitment in practicing guitar, if they do not practice it regularly, they might fail to learn guitar.

1.2 Objective

- **To provide exam-related graded guitar lessons platform by using the computer software.**
    - By doing this, guitar beginners can have an examination standard of practice at home. The simulation of guitar examination songs will be introduced in this system. This might help guitar beginners to increase their self-confidence in their guitar skills.
- **To allow beginner guitarists to learn step by step based on their current abilities.**
    - This allows beginner guitarists to learn according to their levels. This is to avoid beginner guitarists to practice advanced guitar skills without correct skills foundation. Other than that, there is a feature that allows beginner guitarists to choose and play songs with different accompaniments.
- **To improve guitarists' guitar playing skills, sight reading and increase the accuracy of tempo, rhythm and pitch playing by providing judgment.**
    - The actual music sheet will be shown in the user interface, the scoring system and judging system will give response based on the accuracy of tempo of the beginner guitarists.
- **To provide a more convenient method for learning guitar compared to traditional printed materials by providing personal learning software.**
    - This system provides a simple plug-and-play function, guitar beginners may just plug in their personal acoustic guitar and start practicing. In addition, it is more environment friendly compared to the traditional printed materials.

- **To provide related guitar tools such as guitar tuner and guitar chords reference.**
    - These are the tools that all the guitarists might wish to have one. This system will combine all these useful features. They can tune their guitar by just plugging in their guitar into computer.

Objective	Problem Solved
To provide exam-related graded guitar lessons platform by using the computer software.	Existing Rhythm Games Do Not Contain Any Exam-Related Lessons
To allow beginner guitarists to learn step by step based on their current abilities.	Fails In Learning Guitar With The Incorrect Steps
To improve guitarists' guitar playing skills, sight reading and increase the accuracy of tempo, rhythm and pitch playing by providing judgment.	Existing Rhythm Games Do Not Tell The Weaknesses Of The Players In Terms of Examination
To provide a more convenient method for learning guitar compared to traditional printed materials by providing personal learning software.	Existing Rhythm Game Is Inconvenient In Setting Up

Table 1.2-T1: Objectives and Problem Solved Relationship

1.3 Project Scope

Computer Assisted Music Practice for Guitar (CAMPGuitar) is a rhythm game based system which able to lead beginner guitarists to success. The reason the software is introduced in rhythm game based it is because rhythm game is a very interesting and entertaining music game. When learning comes together with entertaining, the effect of learning is a ton better.

This system will introduces the simulation of Registry of Guitar Tutor (RGT) grade examination standard acoustic guitar lessons under grade 3. The reason to limit the lessons under grade 3 is because the acoustic guitar grade 3 examinations consists of strumming patterns and multiple strings playing, which involves few strings to be played together, this might confuse the conversion of audio-to-midi.

Scoring system will also introduce to allow beginner guitarists to practice based on their current abilities, they can only proceed to the next stage when their score reached certain amount.

Score (Percentage)	**Grade**	**Standard**
Score ≤ 85%	A	Distinction
75% ≤ Score < 85%	B	Merit
65% ≤ Score < 75%	C	Pass
55% ≤ Score < 65%	D	Below Pass – Upper Level
50% ≤ Score < 55%	E	Below Pass – Lower Level
Score < 50%	F	Failed

Table 1.3-T1: Grade Calculation of the Scoring System

Table 1.3-T1 shows the method of calculation of the score and grade. This system will tell the beginner guitarists what are the grade they currently have. The guitarists are allowed to play next level of songs after they achieved Grade B in the current stage, which is Merit.

Song accompaniments system allows users to play the song with different accompaniments, users can choose whatever background tracks they which to practice with. They can also adjust the speed of the music notes in the game to increase their sight reading speed.

Last but not least, there will be some built-in tools such as guitar tuner and guitar chords reference. This guitar related tools are very useful to all guitarists.

1.4 Contribution

Computer Assisted Music Practice for Guitar (CAMPGuitar) is a game-based learning system that leads the beginner guitarists into graded guitarists, it introduces acoustic guitar lessons and the simulation of graded songs that allow users to practice repeatedly until they get a good result. In addition, this system is able to build self-confidence for the guitar beginners in order to achieve the standard of guitar playing.

Basically, CAMPGuitar is designed to achieve the correct procedure in learning acoustic guitar, this system is targeting the people who interested in learning guitar, especially for the people who have no knowledge in learning guitar. By providing game-based learning system, guitar beginners are able to learn guitar with more fun and yet achieve the grading standard.

Other than that, CAMPGuitar introduces scoring system to tell about the current standard of the beginner guitarists. Scoring system allow beginner guitarists to know what are the grades they are having for certain songs, what are the results when they try harder, and also when they can proceed to the next grade. There is also an additional feature tells the accuracy of playing of the users.

Last but not least, this CAMPGuitar includes many useful guitar related tools such as guitar tuner and guitar chords reference, these built in features are useful to all the guitarists in the way of success.

1.5 Background Information

Guitar is a very popular music instrument in this world. However, beginner guitarists are always face difficulty in learning guitar. In summary, they can be divided into three types of beginner guitarists, the first type has the habits of repeating the songs which they already know well, which makes them stuck in their current guitar skills and makes no improvement. The second type of beginner guitarists always try hard on the new stuffs and techniques, this type of guitarists do play guitar beyond their current abilities, which makes them improve in a slower way and also might bring them to failure. The third type is the one who able to find the balance point between the first type and second type. This type of beginner guitarists is able to divide their practice time well in mastering current abilities and also challenging the harder ones.

To be honest, no matter which type of beginner guitarists, many of them will start to feel bored when they meet difficulties. Sometimes the problems are not come

from the beginner guitarists, but the guitar tutors in music school. Most of the guitar tutors strongly believe that the lessons listed in the lesson book is the best procedure to learn guitar, but it is actually not.

The traditional way of learning guitar, which means completely follow the procedures as listed in the traditional guitar lesson book, is extremely boring, which is the main reason why the beginner guitarists loss of interest. Although most of the lessons introduced in the lesson book such as learn to read music notation, practice on finger and music theory are very useful for beginner guitarists, they will just started to feel reluctant when they felt hard to catch up, followed by lack of confidence and end up with loss of interest.

Rhythm game defined as music game which uses controller with appropriate buttons, analog or digital sticks to play. The game will guide players on when to press which buttons according to the music beats, the score will get higher when players react in accurate timing. The examples of rhythm games are Guitar Hero and Rock Band, which purposely designed for playing the plastic guitars, drums or microphones to play along with the licensed music. (Ben Kuchera,2008).

CHAPTER 2 LITERATURE REVIEW

2.1 Rhythm Game : Rocksmith

Rhythm games are in a state of decline, but this does not happen to a rhythm game known as “Rocksmith”. Rocksmith is an Xbox 360 game which designed and developed by Ubisoft, this rhythm game successfully combined a music game to real electric guitar. Gamers can just plug in their own guitar to the system by using the provided ¼-inch adapter cable, it takes some time on setting to get a better response between visual and audio lag. Rocksmith leads players starting from basic guitar playing, simple licks, single note and slow pacing until you familiar with it. (Nathan Meunier, 2011)

Figure 2.1-F1: Rocksmith Gaming Interface

Figure 2.1-F1 shows the gaming interface of Rocksmith, each guitar string is represented in different colors, the number of frets also displayed accordingly. Players need to achieve in certain minimum score in order to proceed to the next stage, this system is specially designed and it is reasonably lenient, therefore this game can still proceed even if players perform poorly. In addition, this game also introduces advanced guitar playing techniques, detailed video explanations, practice challenges, tutorial and built-in tuner.

Rocksmith providing many good ways to improve player guitar playing abilities, but it still comes with pros and cons. The technology of Rocksmith works very well in Xbox 360. Furthermore, there are many useful built-in tools for guitar beginners, which bring a lot benefits to them.

Although Rocksmith is easy to set up, the game will lag when players connect Xbox 360 in high definition display via HDMI cable, this is very disappointing when a real-time gaming comes with delay. Other than that, the presentation of Rocksmith interface is weak, they tried to introduce interesting interface, but there is too many miscalculations and mistakes which causes the effect goes the other way round.

2.2 MIDI (Musical Instrument Digital Interface)

MIDI (Musical Instrument Digital Interface) is a protocol that is specially designed for capturing and playing recorded music on digital synthesizer which is supported by many makes of computer sound cards. MIDI Transmits information by using command sets, it represents the note-ons, note-offs, key velocity, pitch bend and all other related methods to control the synthesizer. The sound card or any receiving

instrument has already stored sound waves in a wavetable, when the command comes, the sound waves is produced and retrieved from wavetable. (Marganet Rouse, 2005)

Wavetable is normally stored in read-only memory (ROM) of a sound card chip, but it may also come together with some software. Computer sounds were originally generated by frequency modulation (FM). Prestoring sound wavesforms in a lookup table will improve the quality and throughput. In addition, the sound of wavetable can be improved by using some other effects before it is actually stored. (Ron K. Good, 2005)

Users can easily create music by using standard keyboard and some other input device with the MIDI program. The MIDI program provides graphical user interface to allow users to control easily, which looks almost like a sound studio control room. MIDI has the advantage of small in size, and yet still able to perform well. Unfortunately, MIDI has the disadvantage of less specific in sound control.

2.3 Game-based Learning

The main purpose of game-based learning is to provide a more fun way of learning to the learners, most of the game-based learning has its own challenges and objectives. Game-based learning allows learners to have fun while learning, experience and express their feelings by taking different challenges. Other than that, the scoring system also brings motivation to the learners. They will try to achieve better score every time they play the games, their skills might improve indirectly by continue playing the games.

Some games actually introduces an environment which simulate the worst situation of the real world, learners are able to learn problem solving skills, communication skills, leadership skills and social skills. Most of the game-based learning games consist of few different levels, the levels are arranged from low to high depends on the difficulty of the skills. Game-based learning is used for certain formal education with a very good result, such as in military, medicine, business, training, etc. (Maja Pivec, 2005)

There are actually many people who would like to learn guitar. Unfortunately, there are quite number of them do not continue learning due to fail in encountering obstacles. Once they failed to overcome obstacles, they might feel boring and result in lack of interest. By introducing game-based learning, strongly believes that those people might able to revive their motivation by taking the game challenges, and continue learning by playing the game-based learning games.

2.4 Fast Fourier Transform (FFT)

Fast Fourier Transform (FFT) is a mathematical technique used to process streams of data that underlies the operation of things, for example, WiFi routers and 4G cellular networks. FFT is presenting the signal wave as in sound recording, which is the combination of sine and cosine waves with different frequencies and amplitudes.

FFT is a very old but still powerful algorithm, it was developed in the mid-1960s. There is a new algorithm known as Sparse Fourier Transform (SFT), it is 10 to 100 times faster in streaming data compared to FFT. Although SFT is faster than FFT,

SFT is not absolutely better than FFT, it is because SFT algorithm does not work with all possible streams of data, it introduces much more limitation compared to FFT.

FFT algorithm can be used to build a simple guitar tuner. It can generate the spectrogram of the signal of short periods of time, and also determine the fundamental frequency by finding the index of the maximum value of the magnitude squared after generating the spectrogram.

	Traditional Octave Name (Scientific), Hz			
Note Name	Great (2)	Small (3)	One-lined(4)	Two-lined(5)
C	65.4064	130.8128	261.6256	523.2511
C#	69.2957	138.5913	277.1826	554.3653
D	73.4162	**146.8324**	293.6648	587.3295
D#	77.7817	155.5635	311.1270	622.2540
E	**82.4069**	164.8138	**329.6276**	659.2551
F	87.3071	174.6141	349.2282	698.4565
F#	92.4986	184.9972	369.9944	739.9888
G	97.9989	**195.9977**	391.9954	783.9909
G#	103.8262	207.6523	415.3047	830.6094
A	**110.0000**	220.0000	*440.0000*	880.0000
A#	116.5409	233.0819	466.1638	932.3275
B	123.4708	**246.9417**	493.8833	987.7666

Table 2.4-T1: Notes and Fundamental Frequency Values

Table 2.4-T1 shows the exact fundamental frequencies of the music notes. The guitar tuner used these values to compare and determine the pitch generated by the instruments, and shows the current notes based on the fundamental frequencies. (notmasteryet, 2010)

CHAPTER 3 METHODOLOGIES AND TOOLS INVOLVED

In order to develop Computer Assisted Music Practice for Guitar (CAMPGuitar) in a good way, many related software are involved in the process, same goes to the methodologies.

3.1 Methodologies

Developing this Computer Assisted Music Practice for Guitar (CAMPGuitar) requires many loops of analysis, design and evaluation for the functionality changes to achieve the quality of the system. Therefore, iterative development model is selected to be the backbone of this software development project.

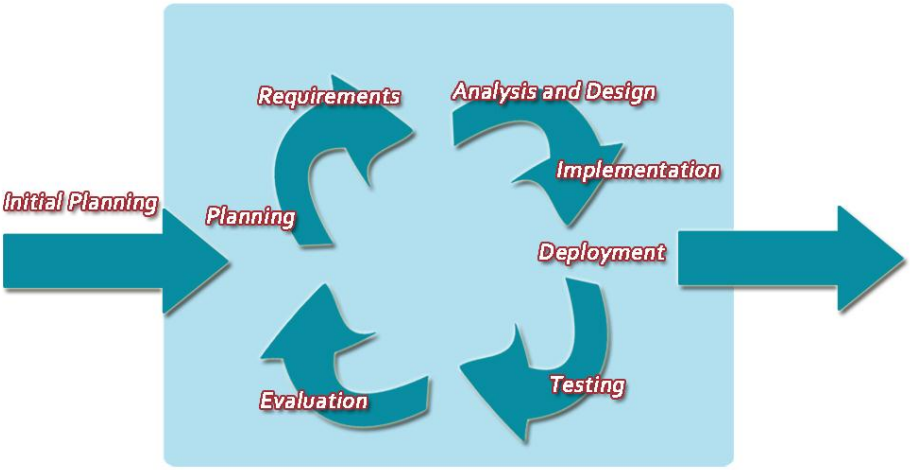


Figure 3.1-F1: The Iterative Model

Iterative development introduces a better process in developing a quality system, this method allows developers to determine the strengths and weaknesses of the system. Other than that, bugs and errors will get detected when developers perform testing, and so the system problems will be reduced.

Initial Planning

- Starts planning with low information of related field, most of the idea features are generated with no detail knowledge.
- Determine the related information that needs to be gathered.
- Determine the direction of the system.

Gather Requirements

- Gather detail requirements through related parties and internet.
- Determine the usefulness of the idea features and remove the less efficient features.
- Determine other system requirements which related to the system.

Analysis and Design

- Obtain enough information for the system requirements through internet and other possible places.
- Perform high-level analysis to ensure the information obtained is enough.
- Starts design the system functionality based on analyzed information.
- Design the graphic user interface of the system.

Implementation

- Implement the designed system requirements by coding, and then integrate to the system.

Testing

- Tests the added system features from the previous phases.
- Determine the acceptance of the system by using acceptance test.
- Functionality testing, requirements validation and verification testing will also be used in this phase.
- Users will perform the tests and determine the problems.

Evaluation

- Gather feedback from tests, bring the feedback to the next iterative cycle for further improvement.

Deployment

- After all testing is done, both users and developers are agreed and the system will be generated together with the module for integration testing.

3.2 Skills/Technologies involved

3.2.1 XNA Game Studio

XNA Game Studio is an integrated development environment (IDE), it provides C# library for creating computer games. It is a free of charge software. The games that developed by XNA Game Studio is able to play in Microsoft Windows and X-Box 360. In order to use XNA Game Studio, Microsoft Visual C# must be installed into the computer together with the development tools and runtime environment, so that the XNA Game Studio environment can be loaded in the development environment. (Margaret Rouse, 2008)

3.2.2 C# Programming Language

In order to use XNA Game Studio, C# Programming Language is the only workable programming language. C# is an extremely powerful object-oriented programming (OOP) language that developed by Microsoft. In addition, C# is the enhanced programming language of C++, it has the foundation of C and C++, and also has the similar features as Java. Other than that, C# has simplified the syntax and makes it more logical and consistent. Some of the complex features of C++ are also removed.

Furthermore, C# is now compatible with the introduction of XML comments. The XML format comments can be used in documenting C# codes.

Last but not least, C# is able to interoperate with .NET platform programming language, it can inherit or extend the project with other languages from C# itself.

3.2.3 Sonuus G2M Guitar to MIDI Converter

Sonuus G2M Guitar to MIDI Converter allows users to plug in their guitar and convert the input into MIDI format without any extra hardware and software. It helps to reduce the heavy work of converting the guitar audio input into MIDI via audio input of the computer sound card.

In addition, it has the ability to reduce the latency of the guitar input and the conversion of audio-to-midi. It tracks the guitar strings separately through the guitar pickups, this could be used to control the keyboard synthesizers and synth modules.

Figure 3.2-F1: Sonuus G2M Guitar To MIDI Converter

Figure 3.2-F2: Sonuus G2M Guitar To MIDI Converter Ports

Sonuus G2M Guitar To MIDI Converter is a very easy to use hardware, users may just plug in any electric guitar into the guitar input, and then connect the MIDI output to computer, it does not require any hexaphonic pickup. There is a boost switch beside the guitar input port, the function of the switch is to boost the low-output pickups by adding extra power. This hardware requires battery to work, there is a red LED light on the converter to tell users the battery status. By using this hardware, users can just convert the guitar input into different kind of music, for example, guitar to piano or saxophone.

3.2.4 Stereo Jack Adaptor 3.5mm Male to 6.5mm Female

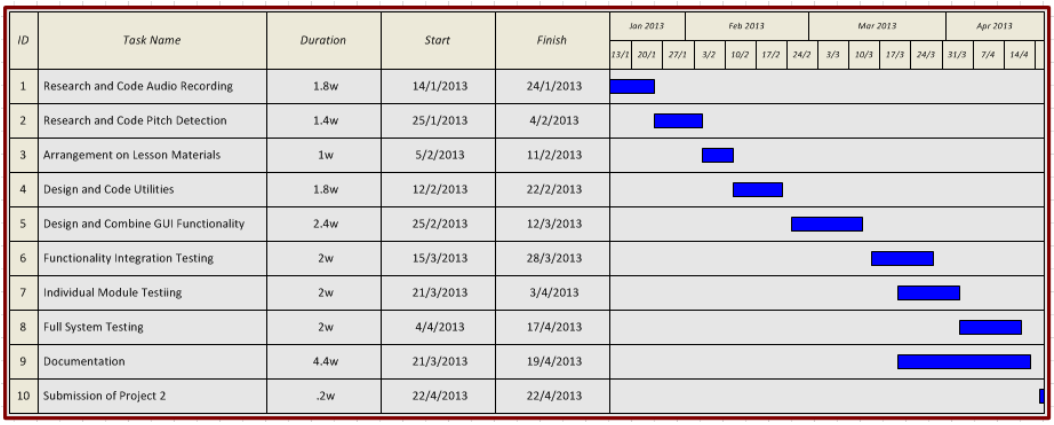
Figure 3.2-F3: Stereo Jack Adaptor 3.5mm Male to 6.5mm Female

This is one of the ways to directly bring the audio signal into computer input. In order to make normal guitar cable able to plug into computer, stereo jack adaptor has the ability to make it works. The guitar cable will plug into 6.5mm female side, and the male side of stereo jack adaptor will plug into the audio input of computer.

3.3 Timeline

ID	Task Name	Duration	Start	Finish
1	Research on Desire Project	10d	28/5/2012	8/6/2012
2	Project Title Registration	7d	7/6/2012	15/6/2012
3	Requirements Planning	13d	14/6/2012	2/7/2012
4	Timeline Planning	7d	2/7/2012	10/7/2012
5	Information Gathering	15d	11/7/2012	31/7/2012
6	Determine Problem Statement, Objective and Project Scope	15d	27/7/2012	16/8/2012
7	Determine Development Tools and Technologies Involved	17d	3/8/2012	27/8/2012
8	Proposal Preparation	31d	16/7/2012	27/8/2012
9	Submission of Project 1 Final Report	3d	29/8/2012	31/8/2012

Figure 3.3-F1: Project 1 Gantt Chart



ID	Task Name	Duration	Start	Finish
1	Research and Code Audio Recording	1.8w	14/1/2013	24/1/2013
2	Research and Code Pitch Detection	1.4w	25/1/2013	4/2/2013
3	Arrangement on Lesson Materials	1w	5/2/2013	11/2/2013
4	Design and Code Utilities	1.8w	12/2/2013	22/2/2013
5	Design and Combine GUI Functionality	2.4w	25/2/2013	12/3/2013
6	Functionality Integration Testing	2w	15/3/2013	28/3/2013
7	Individual Module Testiing	2w	21/3/2013	3/4/2013
8	Full System Testing	2w	4/4/2013	17/4/2013
9	Documentation	4.4w	21/3/2013	19/4/2013
10	Submission of Project 2	.2w	22/4/2013	22/4/2013

Figure 3.3-F2: Project 2 Gantt Chart

CHAPTER 4 SYSTEM REQUIREMENTS

4.1 Requirement Specification

4.1.1 User Requirements

1. The system should able to present guitar notes well in GUI.
2. The system should able to connect the real acoustic guitar.
3. The system should able to give score according to the user guitar playing standard.
4. The system should able to select song with different accompaniments for practice purpose.
5. The system should able to lead users to achieve acoustic guitar examination standard.
6. The system should able to proceed stage by stage depends on the score played by the users.
7. The system should able to tune guitar.
8. The system should have guitar chord reference.

4.1.2 System Requirements

1. The system should run in Windows-based operating system.

2. 6.5mm to 3.5mm Audio Jack Converter is required to convert the jack cable and make the guitar able to plug into computer audio input port.
3. The system should able to receive the guitar input.
4. The system should able to match the input with the running notes in order to determine the score.
5. Laptop with audio input port was required for this system for receiving the guitar input signal.

4.1.3 Functional Requirements

1. The system should able to determine and calculate the overall score of playing by users.
2. The system will display the actual guitar sheet of the playing songs.
3. The system should provide guitar tuner and guitar chords reference.

4.1.4 Non-functional Requirements

1. Users must have their own acoustic guitar and plug in cable.
2. The system should able to achieve the correct guitar learning procedure.
3. Users should spend more time with the system in order to achieve a good result.

CHAPTER 5 SYSTEM MODELING

5.1 STRUCTURAL DIAGRAM

5.1.1 Component Diagram

Component diagrams illustrate the overall physical structure of the program code, it maps and shows the logical view of the project classes. This diagram is representing the locations of all the files and classes. By designing component diagram, the structure of the system become more solid and also helps in guiding the source code location.

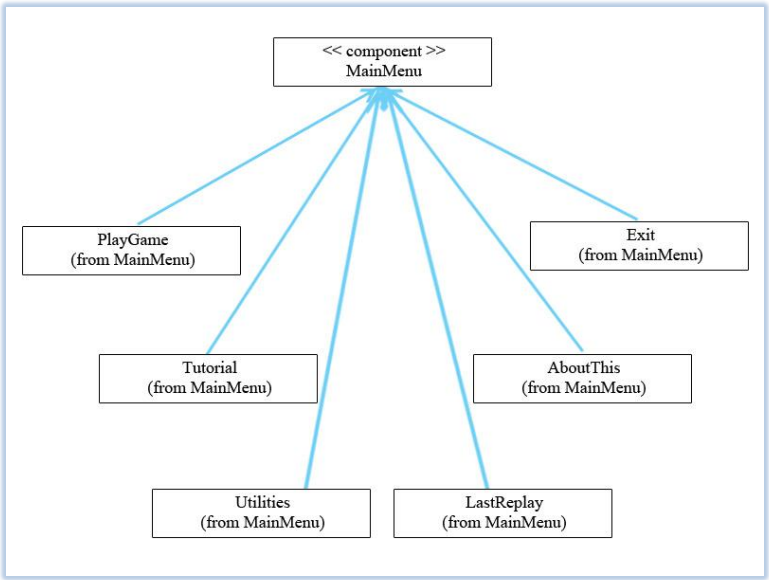


Figure 5.1-F1: Component Diagram for MainMenu

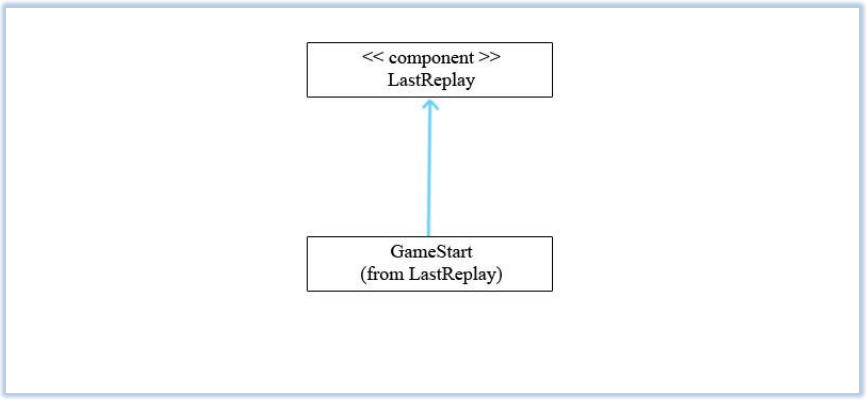


Figure 5.1-F2: Component Diagram for LastReplay

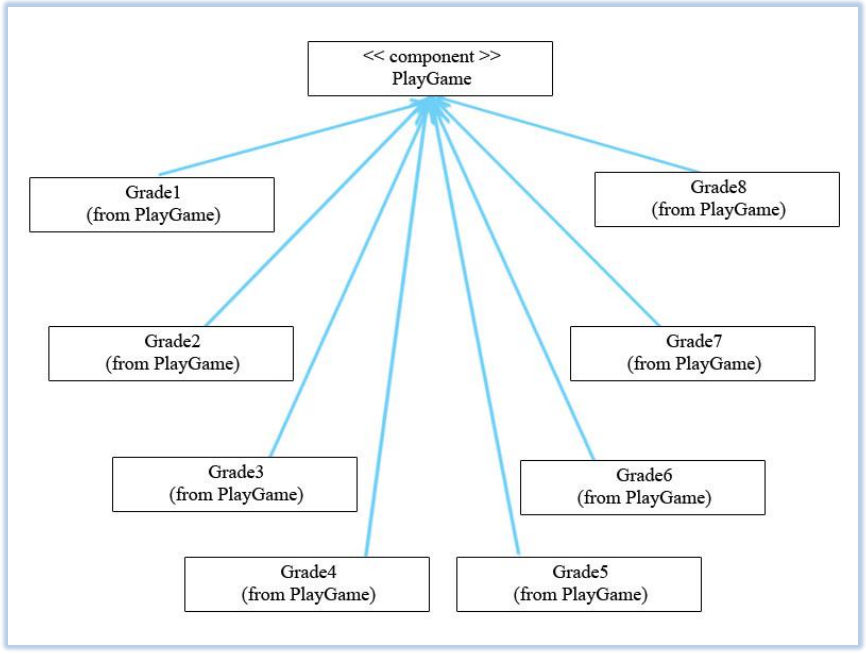


Figure 5.1-F3: Component Diagram for PlayGame

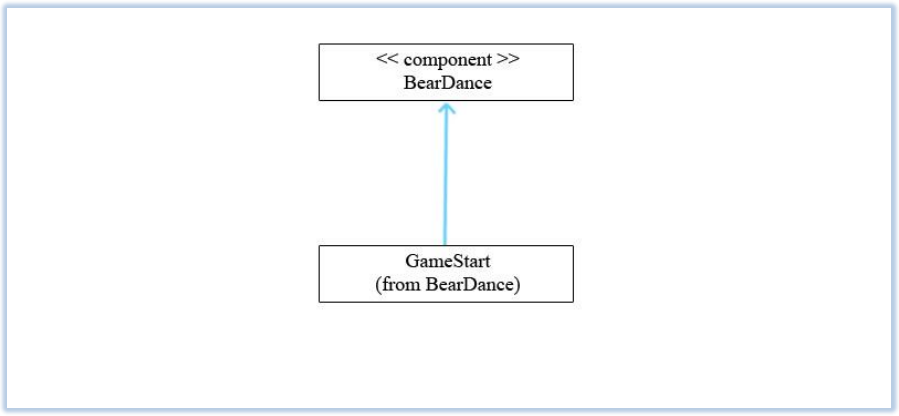


Figure 5.1-F4: Component Diagram for BearDance

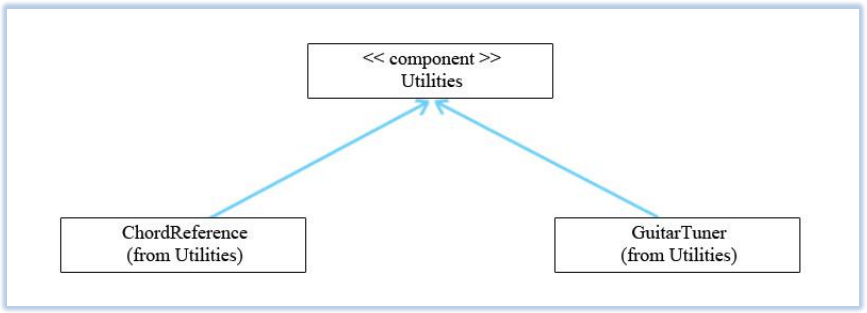


Figure 5.1-F5: Component Diagram for Utilities

5.2 Behavioural Diagram

5.2.1 Use Case Diagram

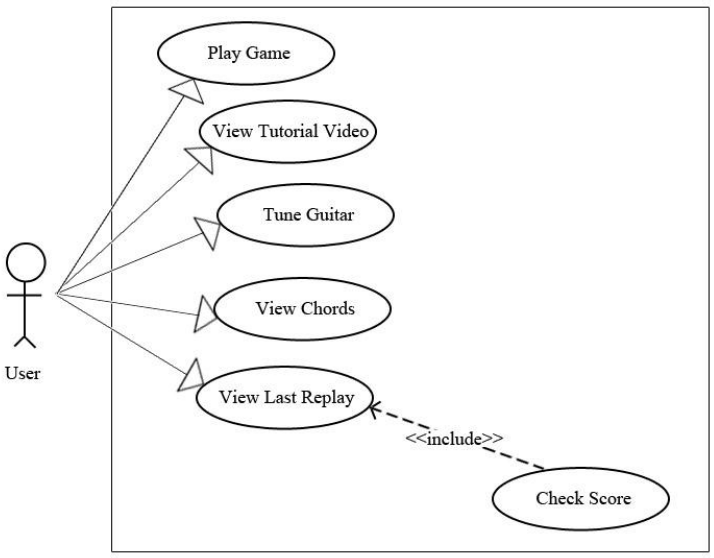


Figure 5.2-F1: Use Case Diagram for CAMPGuitar

The use case diagram above shows the overall communication between the user and the system. Other than just playing the examination song rhythm game, the user is allowed to check their grade and score of play. When they are playing, there is an accuracy judgment in real time that shows the correctness of the tempo of playing. Other than that, they can also refer back to the last replay that they have played.

There are some tutorial videos for the user to understand more about guitar. The user can also use all other utilities including guitar tuner, metronomes and guitar chord reference.

CHAPTER 6 SYSTEM DESIGN

6.1 Graphic User Interface Design

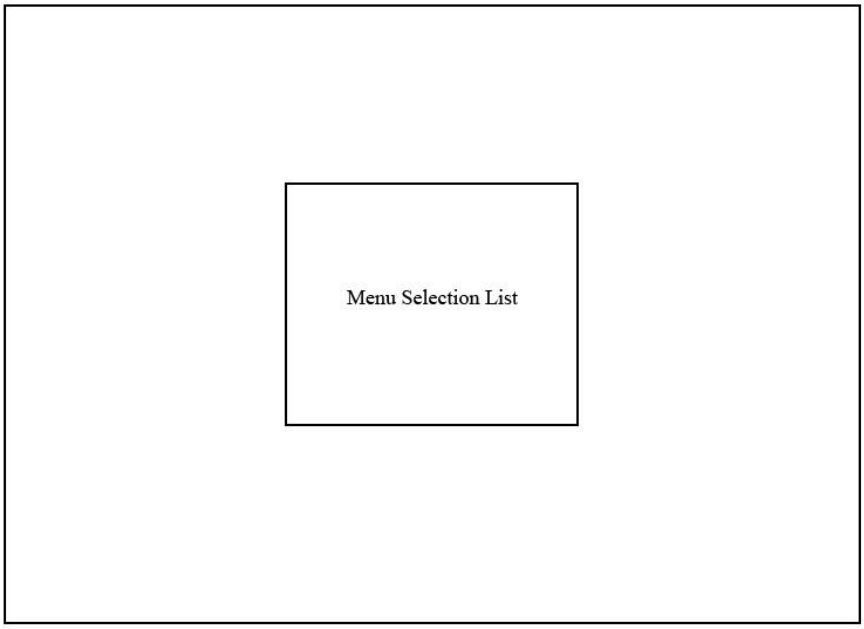


Figure 6.1-F1: The design layout for the menu page

Figure 6.1-F1 shows the GUI design layout for CAMPGuitar. The design is simple yet easy to use, a list of choices is shown on the page, and user is allowed to choose the input by pressing up and down buttons on keyboard. This design is also

can be used on X-box, it is because there is no keyboard for X-box, user may select their choice from the list using the controller.

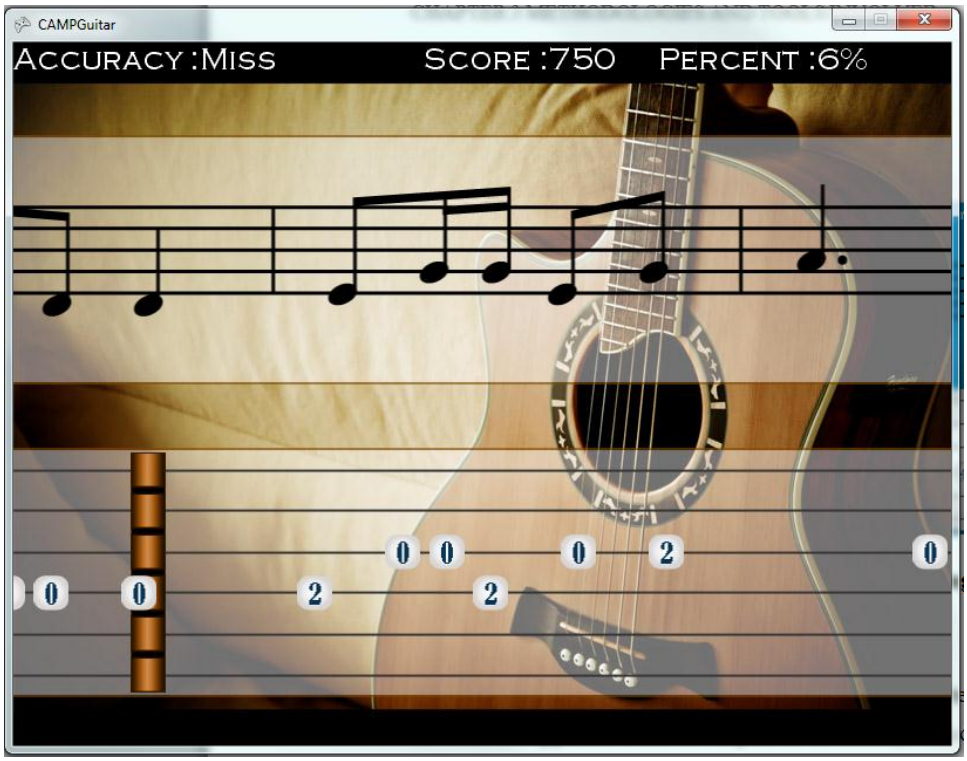


Figure 6.1-F2: The rhythm game playing GUI

Figure 6.1-F2 shows the playing screen of this system. The music notes and the numbers appeared at the lower part of the screen, both of them moving leftward accordingly followed by the music beats. When the number has reached the neck, the user needs to pluck the guitar strings accordingly. There are also score, percentage and accuracy appearing on the screen. Score tells the current score of the user, and the score will increase according to the accuracy of user's playing beats. Percentage

shows the actual percentage that the user has successfully achieved in the particular song. Accuracy can tell the users how accurate they actually played.

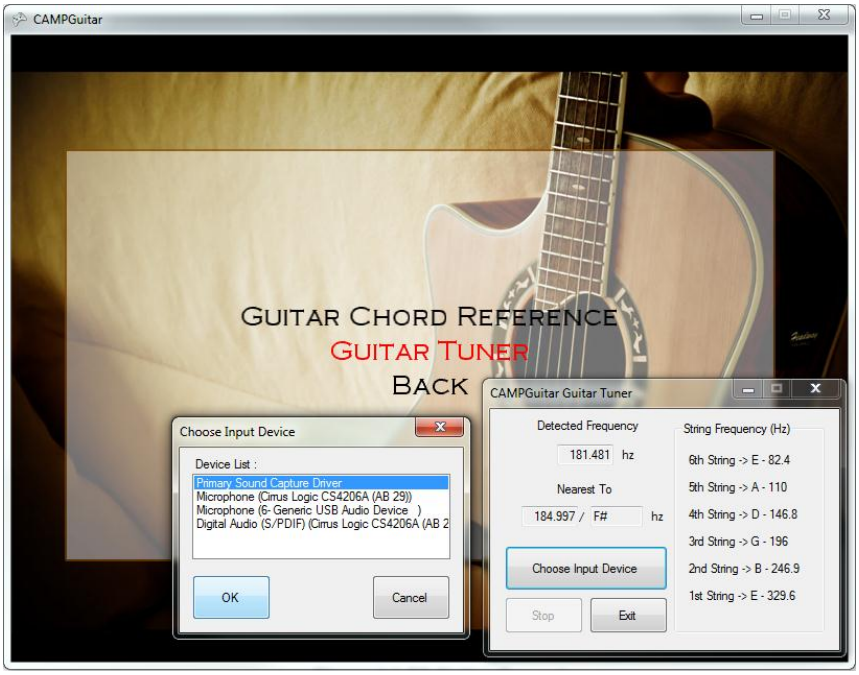


Figure 6.1-F3 Guitar Tuner Screen

There are two guitar utilities that are available in this system. The figure above shows that when users select Guitar Tuner, a small guitar tuner program will pop out.

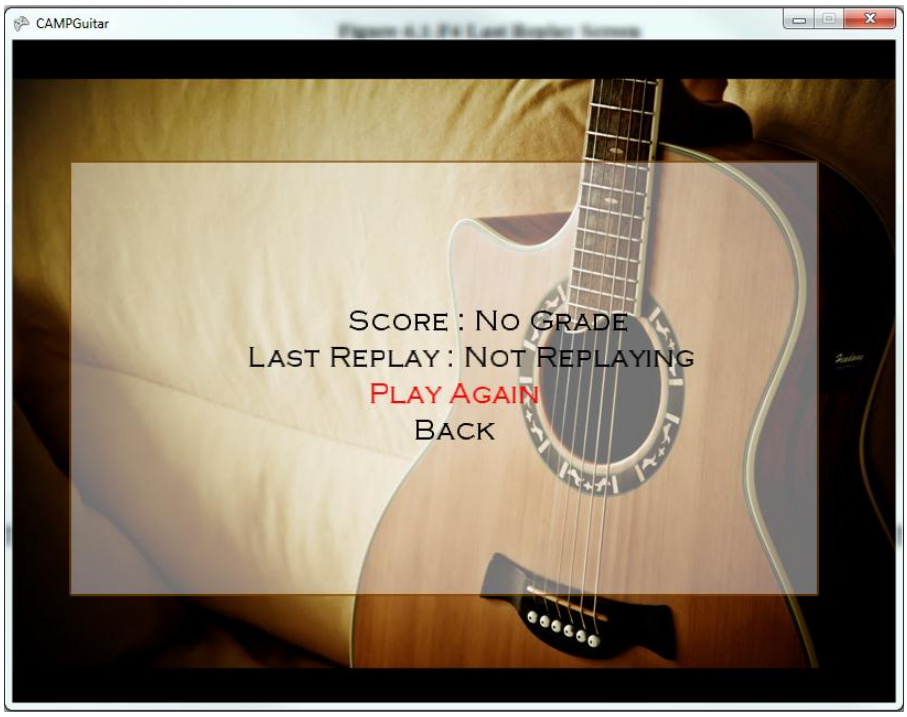


Figure 6.1-F4 Last Replay Screen

After finished a particular song, users are allowed to check their Score in grade as shown above. In addition, they can also listen to the last replay, so they can know what their obvious mistakes are.

CHAPTER 7 SYSTEM IMPLEMENTATION AND TESTING

7.1 Introduction

In order to achieve the best result, implementation and testing actions need to be taken seriously. As soon the implementation is perfectly done, heavy testing is conducted on all the implemented modules. The purpose of testing is to determine the bugs and errors of the system, and also perform checking on the user requirements. At the end of testing, the test result will be analyzed and reviewed in order to achieve the final system quality. Furthermore, updates and further maintenance will also put into consideration on the final stages of the system development.

7.2 System Equipment Setup

There are requirements needed to be fulfilling in order to perform the system faultlessly. CAMPGuitar is fully corporate with the plug in real guitar, the input of the system must be purely guitar sound, and any other sound interruption from the input might cause the system to compute the input frequency incorrectly. Figure 7.2-F1 shows the basic equipment requirement of CAMPGuitar.

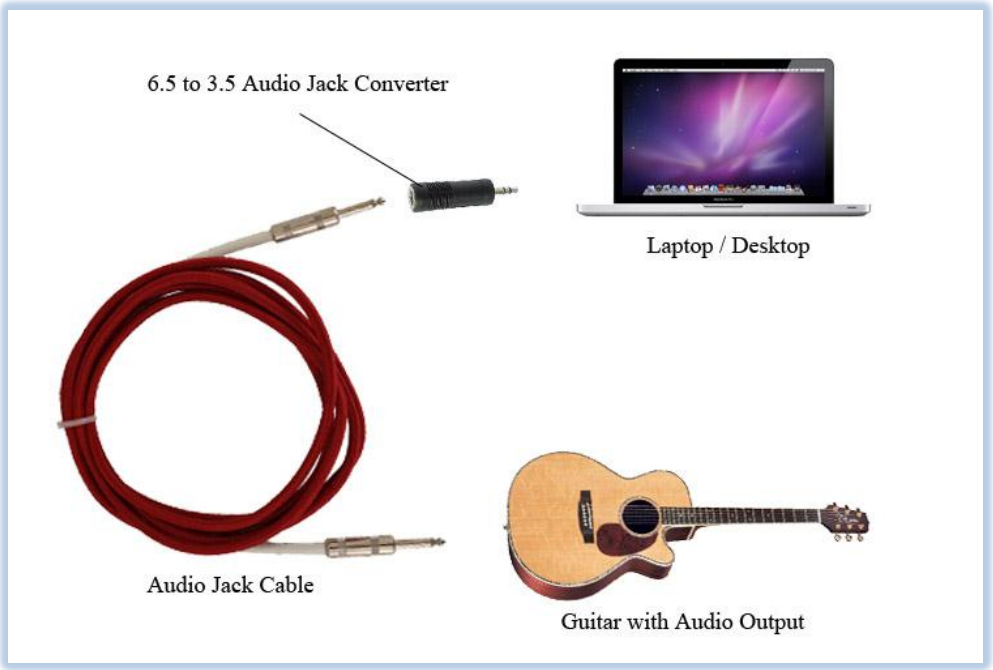


Figure 7.2-F1: Equipment needed for the system

7.3 System Implementation

CAMPGuitar is a rhythm game-based system, it is purely emphasized on music and audio. This system gets the guitar input from the audio input, and calculates the frequency of the input in real time, and matches the current playing music note as well as to calculate the score and show it to the user immediately. The difficult part is not only the pitch detection but also the concept of matching the appearing beats on screen synchronize with the song, the calculation of beats per minute (BPM), which increases the difficulty on completing the project.

7.3.1 Pitch Detection Implementation

Pitch detection consists of several stages, the stages include getting the input from the audio input, fetch the input immediately to FFT algorithm to calculate the frequency in real time, and also match the frequency with current appearing notes on screen.

```
FileStream FS_Write = File.OpenWrite(lastReplayFileName);

int deviceNumber = 0;
sourceStream = new NAudio.Wave.WaveIn();
sourceStream.DeviceNumber = deviceNumber;

sourceStream.WaveFormat = new NAudio.Wave.WaveFormat(44100, NAudio.Wave.WaveIn.GetCapabilities(deviceNumber).Channels);

sourceStream.DataAvailable += new EventHandler<NAudio.Wave.WaveInEventArgs>(sourceStream_DataAvailable);

waveWriter = new NAudio.Wave.WaveFileWriter(lastReplayFileName, sourceStream.WaveFormat);

LoadNewSong();
MediaPlayer.Play(song);
sourceStream.StartRecording();
```

Figure 7.3-F1: CAMPGuitar audio input element

Figure 7.3-F1 shows the implementation code for recording the input from audio input. An audio file is being opened and writes the audio input immediately to the 'lastReplayFileName' location in real time. A microphone wave in channel is declared and used for record and analyze the audio input signal, the signal is then being fetched to the FFT algorithm for computing the frequencies. The recorded audio is allowed to play back for reference.

```
StreamReader noteSR = new StreamReader(fileName + ".txt");
StreamReader noteBarSR = new StreamReader(fileName + "Bar.txt");

string noteRead = noteSR.ReadToEnd();
string noteBarRead = noteBarSR.ReadToEnd();
string[] noteNumbers = noteRead.Split(',');
string[] noteBarNumbers = noteBarRead.Split(',');

noteList.Clear();
noteBarList.Clear();

for (int i = 0; i < noteNumbers.Length; i++)
{
    if (i == 0)
        songOffset = double.Parse(noteNumbers[i]);
    else if (i == 1)
    {
        BPM = int.Parse(noteNumbers[i]);
    }
    else
        noteList.Add(int.Parse(noteNumbers[i]));
}


for(int j = 0; j<noteBarNumbers.Length;j++)
    noteBarList.Add(int.Parse(noteBarNumbers[j]));

for (int k = 0; k < noteList.Count; k++)
    if (noteList[k] != 0)
        noteCountForScore++;
```

Figure 7.3-F2: CAMPGuitar load music information element

All the information regarding the music notes are represented by digits and stored in text file. The digits are retrieved after the users chosen their songs, and the digits are stored in the arrays for later use. This function is loaded frequently every time when the users have chosen the song they wish to play.

7.3.2 Beats Movement Implementation

Figure 7.3-F3 shows how the notes are being assigned with the appearing time and how it actually located when the time is running. The first for loop is assigning the actual appear time of the notes, every single note is assigned by time, the song offset time plus second per beat, which appeared as 'sixteenth' at figure above. The second for loop is executed in the Update section of XNA Game Studio, which rapidly repeated until the song ended. The note position will move to the left accordingly.

```
for (int i = 0; i < noteList.Count; i++)
{
    noteActualTime[i].X = (float)((songOffset + (i * sixteenth)));

}
```

```
for (int i = 0; i < noteList.Count; i++)
{
    if (noteList[i] == 0)
        notePosition[i].X = -999;
    else
    {
        dt = (float)(noteActualTime[i].X - currentSecond);
        notePosition[i].X = (float)(110 + dt * noteSpeed);


    }

}
```

Figure 7.3-F3: CAMPGuitar beats movement element

7.3.3 FFT Algorithm Implementation

There are various types of algorithms that can function on pitch detection and one of the best algorithms that chosen for the implementation known as Fast Fourier Transform(FFT). FFT gather the "time domain" signal and convert it into "frequency domain" signal. The audio signals are interpreted as a branch of complex numbers, and then the numbers are computed by FFT.

```
private float[] fftBuffer;
private float[] prevBuffer;
public float DetectPitch(float[] buffer, int inFrames)
{
    Func<int, int, float> window = HammingWindow;
    if (prevBuffer == null)
    {
        prevBuffer = new float[inFrames];
    }

    int frames = inFrames * 2;
    if (fftBuffer == null)
    {
        fftBuffer = new float[frames * 2];
    }

    for (int n = 0; n < frames; n++)
    {
        if (n < inFrames)
        {
            fftBuffer[n * 2] = prevBuffer[n] * window(n, frames);
            fftBuffer[n * 2 + 1] = 0;
        }
        else
        {
            fftBuffer[n * 2] = buffer[n-inFrames] * window(n, frames);
            fftBuffer[n * 2 + 1] = 0;
        }
    }
```

Figure 7.3-F4: FFT Buffer Coding Element

Figure 7.3-F4 shows how the FFT buffer samples being interleaved, the previous buffer is crossing with the current buffer. This is a common way to increase the accuracy of computation of an FFT. After receiving the buffers, it is then being passed to windowing function. Windowing function captures the waveform and computes the amplitude based on the chosen window. There are various types of windows with its own strengths and weaknesses, in this case, Hamming window is chosen for the FFT computation. Figure 7-1-4 shows the hamming window formula, this window function is commonly used for frequency detection.

```
private float HammingWindow(int n,  int N)
{
    return 0.54f - 0.46f * (float)Math.Cos((2 * Math.PI * n) / (N - 1));
}
```

Figure 7.3 –F5: Hamming Window Element

Once the interleaved buffer is ready, it is passed to the FFT algorithm. The chosen FFT algorithm is implemented by Stephan M. Bernsee. After the computation of FFT, the output complex numbers is representing in “bins”, and it might need to recalculate again to get the exact frequency in Hz. The range of “bins” is then being calculated, and then proceed to calculate the magnitude, which also known as “intensity” for each frequency by performing the sum of squares. Figure 7.3-F6 shows the code for computing the maximum intensity, the function is then returned the detected frequency value back to the calling location. (Mark Heath, 2011)

```
float binSize = sampleRate / frames;
int minBin = (int)(85 / binSize);
int maxBin = (int)(300 / binSize);

float maxIntensity = 0f;
int maxBinIndex = 0;
for (int bin = minBin; bin <= maxBin; bin++)
{
    float real = fftBuffer[bin * 2];
    float imaginary = fftBuffer[bin * 2 + 1];
    float intensity = real * real + imaginary * imaginary;
    if (intensity > maxIntensity)
    {
        maxIntensity = intensity;
        maxBinIndex = bin;
    }
}
return binSize * maxBinIndex;
```

Figure 7.3-F6: Maximum Intensity Calculation Element

7.4 Implementation Status

This stage is where the implementation of the system is almost done. Although it is about to complete, there are several upgrades can be done. This system still has the possibility to further improve and enhance the performance.

The Table 7.4-T1 shows the implementation steps of every stages for the CAMPGuitar system. It further describes what has done during the steps of implementation.

No.	Implementation Steps	Description
1	Gathering Information	To gather related information as well as the requirements needed for CAMPGuitar
2	Audio Input Analysis	To understand, analyze and structure the audio input for further implementation
3	System GUI Designs	The structure of user interface of the system
4	Application and Equipment Connection Setup	To implement the connection between CAMPGuitar and its required equipment
5	Coding for the System	Programming the structure of CAMPGuitar
6	Lessons Media Data Input	Insert the songs and music score to the system
7	Coding for Related Utilities	Insert and code for the related materials
8	System Testing	Testing the overall system

Table 7.4-T1: Implementation Steps for CAMPGuitar

As for the Table 7.4-T2, the use module implementation status of the system is shown. The progress and the completion of the system are also shown in the table in

order to make the progress traceable. Although there are necessary updates available, the implementation of the system is considered complete for current stage.

No.	Modules (Use Case)	Description	Duration (Days)	Completion (Date)	Percentage Completion
1	Play Game	Play the guitar lesson rhythm game	5	4-2-2013	95%
2	View Tutorial Videos	Play guitar related videos	2	6-2-2013	100%
3	Tune Guitar	Tune their plugged in guitar	4	10-2-2013	100%
4	View Chord Reference	Search for desired chords	2	19-2-2013	100%
5	View Last Replay	Listen to the previous played song	2	22-2-2013	98%
6	Check Score	Check the score and result of playing	1	23-2-2013	100%

Table 7.4-T2: Implementation status for the modules

7.5 Problem Encountered During Implementation

The implementation phases do not go smoothly as expected, there are several problems encountered during the implementation which really takes time to figure the problems behind as well as the solution to encounter the problems.

During the implementation, there is a class of code is imported for guitar tuning features. When the first time of import the piece code, it comes out with an error "BadImageFormatException". The error is shown in Figure 7.5-F1.

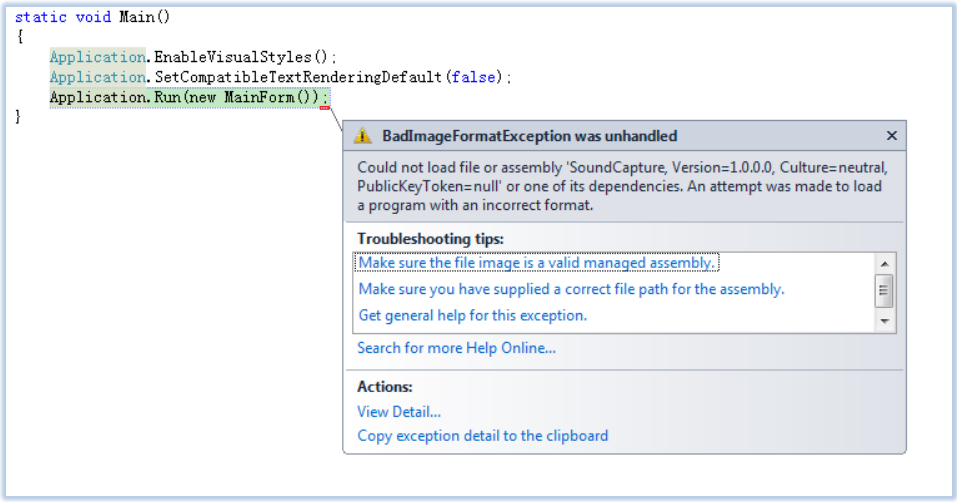


Figure 7.5-F1: "BadImageFormatException" Error Screen

This is an error that not related to the code, but related to the internal software of the computer. After few days of research, found that this problem will occur is because of the OS System type. The imported piece of code is originally created in

32-bit Operating System, and the current system developing machine has 64-bit Operating System, which is the main problem of this error.

The problem is easier to solve when the root is found. Visual Studio allows the developer to change the platform target, which can solve problem of different OS System type. By right clicking the project in the Solution Explorer, properties of the project can be accessed for changes. Figure 7.5-F2 shows how the platform target is being changed.

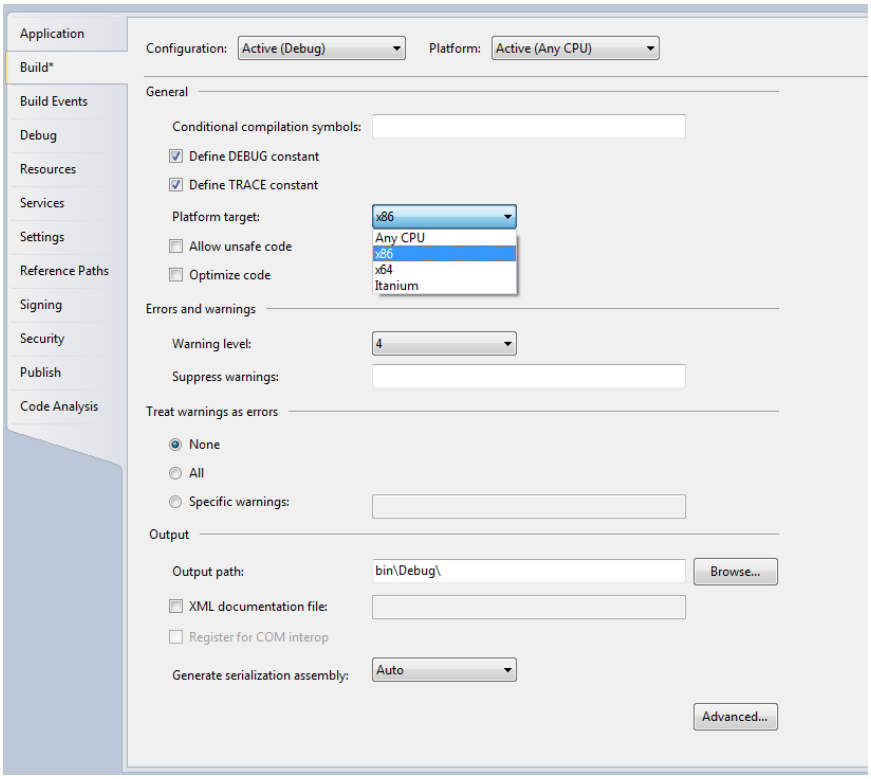


Figure 7.5-F2: Change Platform Target Screen

Unfortunately, this problem has not come to the end yet, there is another problem faced after changing the platform target.

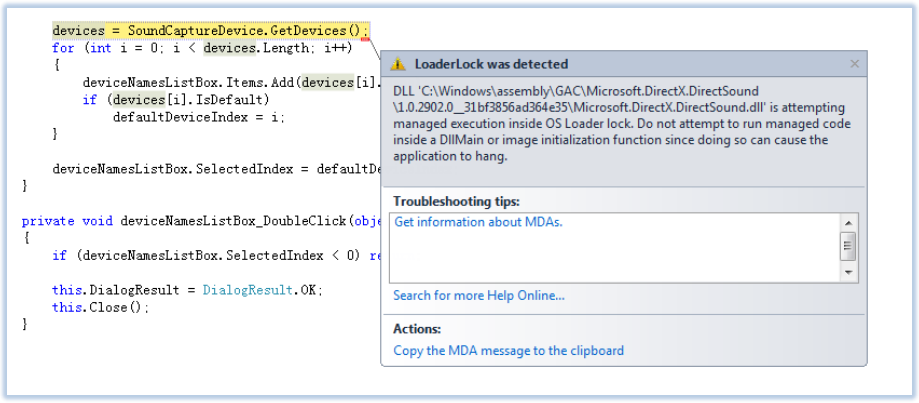


Figure 7.5-F3: LoaderLock Error Screen

Figure 7.5-F3 shows the problem faced after changing the platform target. Loader lock is a managed debugging assistant (MDA) which helps in avoiding deadlocks when the written code is being executed in different Operating System. Visual Studio has enable loader lock by default, therefore it has to be manually disable in order to execute the project. To disable, go to exceptions under Debugs tab, extend Managed Debugging Assistant list and uncheck Loader Lock. After all the steps are done, the project can be executed smoothly.

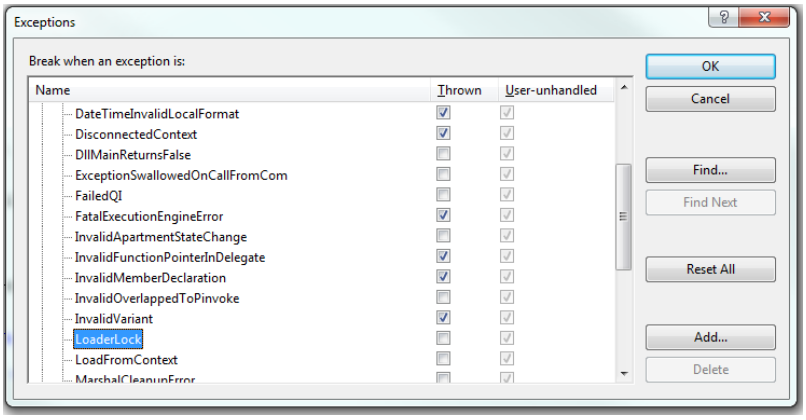


Figure 7.5-F4: Uncheck LoaderLock Screen

7.6 Test Plan

As the testing involves different users and materials, the information and methods need to be well-organized in order to achieve the best result. Thus, the following sub-chapters are as follows:

7.6.1 Test Organization

As for the test organization, the involved testers will be listed down specifically, including their roles and responsibilities. There are two people involve in the system testing, they are the system developer as well as the supervisor of this project.

There are experienced guitarists and also beginner guitarists involved in the end user testing, the supervisor can substitute the position of tester as the supervisor has the advance knowledge on the related field.

No	**Tester**	**Roles and Responsibilities**
1	System Developer	Involved in planning, designing and handle the testing as well as the result documents
2	End Users or Supervisor	Experienced guitarists involved in the testing in order to determine the accuracy of the system. Beginner guitarists, which are the target users of the system to test the usefulness of the system.

Table 7.6-T1: Test Organization

7.5.2 Test Schedule

The test schedule is planned to make sure that all the testing is operated and finished in time. From the estimated time of testing, it takes about two weeks' time to finish as listed in Table 7.6-T2.

Test	Duration (Day)	Date
1. Unit Test	3	11-3-2013 to 13-3-2013
2. Integration Test	3	14-3-2013 to 16-3-2013
3. User Acceptance Test	3	17-3-2013 to 19-3-2013
i. System Developer	1	17-3-2013
ii. End Users	2	18-3-2013 to 19-3-2013

Table 7.6-T2: Test Schedule for CAMPGuitar

7.6 Test Strategy

There are many types of tests for software testing, including bottom-up testing, top-down testing, black box testing and white box testing approach. As for CAMPGuitar, black box testing and white box testing are chosen as the test methods to ensure the entire system is tested faultlessly. In addition, there are both end users and system developers involved in the testing to ensure the system has meet the overall requirements.

1. White Box Strategy

White box testing is named based on the concept of ‘clear box’ or ‘glass box’, which means the internal things can be seen through. The function of white box testing is to ensure the workability of internal coding and infrastructure of the system. This test focuses primarily on the input and output flows of the system, and also improves the design of the system together with the usability. This test is applicable at the unit, integration and levels, but it is typically applied on the unit only.

2. Black Box Strategy

Black box testing focuses on checking inputs and outputs of the system without looking at the internal code structure, implementation details and also any knowledge of the internal coding. This test is basically focused on testing the external part of the system and also tests the requirements, as well as the specification of the system. The main focus of this testing is to validate the functional requirements. This test is definitely widely used on user testing.

7.6.1 Classes of Tests

There are several types of testing which can be conducted in software testing. From the various types of tests, there are four types of testing is suitable to be conducted for a new rhythm game.

The classes of tests are unit testing, component testing, system testing and user testing.

1. Unit Testing

Unit testing is one of the most important tests among all. By operating the unit testing, all the small testable functions are tested as well as the modules of source code. This test is conducted by the system developer and end users.

The purpose of this test is to make sure all the smallest testable functions, or also known as units, to be tested and make sure they are functioning appropriately. The tested functions are listed in Table 7.6-T1.

Test Function	Result
getFileName	Passed
checkKeyboard	Passed
getSpeedSelected	Passed
getAccompanimentSelected	Passed
LoadNewSong	Passed
loadGameInfo	Passed
supposedNote	Passed
PlayingNote	Passed
bytesToFloats	Passed
sourceStream_DataAvailable	Passed
getFullScorePercent	Passed
replayStatus	Passed
scoreInGrade	Passed

Table 7.6-T1 : Test Function in Unit Testing

The result of the first testing is good as expected. The user input is limited to a few choices only, therefore the functions are easier to handle. Most of the functions are used for multiple times as there are several actions needed to have similar information.

2. Integrated Testing

Most of the functions implemented in the system are dependent to the other functions, and therefore integrated testing is a must for this testing phase. Integrated testing is basically combining the unit functions and to determine their workability and to determine the bugs. It is also helps in determine the correctness of the response when there is any state change or any relevant reaction in the component.

Test Component	Result
GameStart	Passed
MenuComponent	Passed
BearDanceComponent	Passed
Grade1Component	Passed
PlayGameComponent	Passed
Utilities	Passed
LastReplay	Passed
ChordReference	Passed

Table 7.6-T2: Test Component in Integrated Testing

It is ridiculously many unexpected errors occurred in the first testing. As the execution of the XNA Game Studio is slightly different from the normal programming environment, the codes are repeatedly performed on the Update sections. It takes time to understand to flow and the errors and bugs occurred when combining the functions. However, all the errors and bugs are eliminated and the functions are working at the end of testing, the outcome is good and slightly better than expected.

3. System Testing

System testing is a test which is aimed to determine the implemented modules whether they have achieve required result. It is also used to explore the errors from the system. There are errors and bugs occurred not because of the coding but the complexity of the system, as human has limited ability to imagine and manage the complexity of a moderate size system. By applying system testing, the overall system will be tested as well as the input and output of the system.

Test Case	Result
Connect Guitar	Passed
Read Guitar Frequencies	Passed
Write and Record Input Sound	Passed
Read and Play Recorded Sound	Passed
Connect to Guitar Tuner Software	Passed
Real-time Frequency Matching	Passed

Table 7.6 -T3: Test Case in System Testing

All test cases are done after system testing. The system requirements and functions worked properly, the outcome of the test cases are within expectation. The system developer managed to solve all errors and bugs immediately when the failures occurred.

4. USER ACCEPTANCE TESTING

User Acceptance Testing is done when the project is come to the final stages. This test is to acquire confirmation by the end users, as well as gather the information about the requirements to further enhance the system. The test can be based on User Requirements Specifications that the system should consist of.

CHAPTER 8 CONCLUSION

In a nut shell, the completion of the first phase of this project documentation is around the corner. The project introduction, literature review and project methodology are successfully accomplished before the deadline given. Most of the requirements of Project 1 are fulfilled and the objectives are well achieved.

Although faced many problems from different aspects, most of them have been encountered successfully but still, there are some minor problems. These problems including convert audio signal into MIDI form, and also the limits of understanding the actual acoustic guitar examination structure due to the author has no experience on it. Other than that, it is hard to develop a system by learning a new programming language within a short and limited time, therefore the timeline must be well planned for the next few phases of the project.

From the information gathered throughout the first phase, found that this system is worth to be developed as it might bring lots of benefits to the people throughout the world who interested in learning guitar. In addition, there is temporary no similar concept of this system throughout this world, it motivates the author in implementing the system.

Most of the objectives are able to achieve successfully and also able to design and implement into the CAMPguitar system positively. The existing rhythm games do not provide guitar practice for examination songs. Furthermore, there is no any rhythm game with the purpose of examination practices. Therefore CAMPGuitar is introduced to help people who wish to learn guitar and go for grades.

CAMPGuitar system implemented based on rhythm games concept. The purpose of using rhythm games concept is to increase enjoyment of the guitar learning process. By making the process of learning guitar more entertaining, the efficiency of the learning progress is definitely better. Other than that, CAMPGuitar is more convenient than any existing systems, the system does not need any big places or many materials, but just a computer and a plug in guitar, the user can start practice and play guitar at whatever places when they have the required materials.

Throughout the whole process of this project, there are many problems that have been encountered. Although the input frequency detection algorithm is widely used, it does not seem able to suit well in guitar pitch detection. There are limitations in detecting the guitar input frequencies, the algorithm is actually detecting the harmonic of the pitch instead of the exact pitch when it is at the higher range. Furthermore, the algorithm is unable to detect multiple frequencies at a time, which restricted the user playing patterns, some multiple strings playing patterns are unable to implement into the system.

There are still many improvement can be done in this system. In order to make it more convenient than the current version, the future work could be implementing this system into mobile version and make it become an apps. On the other hand, there is another future work can be done which is to solve the limitation of input frequency detection algorithm. The current implemented algorithm restricted the playing patterns, and therefore the higher grade songs are unable to operate due to the complexity of playing style. These improvements could enhance the usefulness of CAMPGuitar and also benefits more necessary people.

REFERENCE

Ben Kuchera, 2008. *Rhythm games*. [online] Available at: <http://arstechnica.com/technopaedia/2008/04/rhythm-games/> [Accessed 23 July 2012].

Marganet Rouse, 2005. *MIDI (Musical Instrument Digital Interface)*. [online] Available at: <http://searchcio-midmarket.techtarget.com/definition/MIDI> [Accessed 23 July 2012].

Margaret Rouse, 2008. *What is XNA Game Studio?*. [online] Available at: <http://whatis.techtarget.com/definition/XNA-Game-Studio> [Accessed 24 July 2012].

Maja Pivec, 2005. *The benefits of the game-based learning*. [online] Available at: <http://www.elearningeuropa.infor/it/node/2498> [Accessed 26 July 2012].

Mark Heath, 2011. *Autotune.NET*. [online] Available at: <http://channel9.msdn.com/coding4fun/articles/AutotuneNET> [Accessed 1 March 2013].

Nathan Meunier, 2011. *Rocksmith Review*. [online] Available at: <http://asia.gamespot.com/rocksmith/reviews/rocksmith-review-6341628/> [Accessed 23 July 2012].

notmasteryet, 2010. *FFT Guitar Tuner*. [online] Available at: <http://www.codeproject.com/Articles/32172/FFT-Guitar-Tuner#_articleTop> [Accessed 22 August 2012].

Ron K. Good, 2005. *wavetable*. [online] Available at: <http://whatis.techtarget.com/definition/wavetable> [Accessed 23 July 2012].

BIBLIOGRAPHY

Ben Kuchera, 2011. *The three reasons Rocksmith fails as a way to learn guitar.* [online] Available at: <http://arstechnica.com/gaming/2011/10/the-three-reasons-rocksmith-fails-as-a-way-to-learn-guitar/> [Accessed 23 July 2012].

Corey Palmer, 2010. *7 Mistakes Every Beginner Guitarist Makes*. [online] Available at: <http://truefire.com/blog/sevens/7-mistakes-every-beginner-guitarist-makes/> [Accessed 21 July 2012].

Craig Utley, 2002. *Why you should move to C#.* [online] Available at: <http://www.techrepublic.com/article/why-you-should-move-to-c/1050356> [Accessed 25 July 2012].

Ken adwell, 2007. *Why You'll Fail to Learn Guitar – And What You Can Do About It*. [online] Available at: <http://ezinearticles.com/?Why-Youll-Fail-to-Learn-Guitar---And-What-You-Can-Do-About-It&id=757196> [Accessed 24 July 2012].

Mark Anderson, 2012. *A Faster Fourier Transform*. [online] Available at: <http://www.technologyreview.com/article/427676/a-faster-fourier-transform/> [Accessed 22 August 2012].

Michael Ross, 2010. *Sonuus G2M MIDI Converter Review*. [online] Available at: <http://www.premierguitar.com/Magazine/Issue/2010/Oct/Sonuus_G2M_MIDI_Converter_Review.aspx?Page=2> [Accessed 22 August 2012].

Tom Hess, 2005. *5 Big Mistakes You Should Avoid When Teaching Beginning Guitar Students.* [online] Available at: <http://tomhess.net/TeachBeginnerGuitarStudents.aspx> [Accessed 21 July 2012].

APPENDICES

APPENDIX A: Final Year Project

Biweekly Report

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 1 & Week 2**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

1. WORK DONE *[Please write the details of the work done in the fortnight]*

- Download and import NAudio library
- Install XNA Game Studio 4.0
- Research and code for Audio Recording

2. WORK TO BE DONE

- Research and code for Pitch Detection

3. PROBLEM ENCOUNTERED

- Difficulties in understanding how the recording procedure works

4. SELF EVALUATION OF THE PROGRESS

- New to C# Programming and XNA Game Studio 4.0
- First time on implementing a system which needed to handle raw audio data input

_____ _____

Supervisor's Signature Student's Signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 3 & Week 4**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

1. WORK DONE *[Please write the details of the work done in the fortnight]*

- Successfully implemented Audio Recoding in the system
- Research and Code for Pitch Detection by using FFT algorithm
- Combine Audio Recording and Pitch Detection for the game playing purposes

2. WORK TO BE DONE

- Arrangement on Lesson Materials

3. PROBLEM ENCOUNTERED

- Difficulties in understanding how to implement FFT algorithm into the system

4. SELF EVALUATION OF THE PROGRESS

- Took too much time in understanding the use of FFT algorithm
- Need to speed up in order to follow up the schedule

_____ _____

Supervisor's Signature Student's Signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 5 & Week 6**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

2. WORK DONE *[Please write the details of the work done in the fortnight]*

- Successfully implemented the combination of Audio Recording and Pitch Detection
- Guitar materials and lessons are prepared, including the songs and the equipment
- Did testing for guitar input to determine the accuracy of pitch detection

2. WORK TO BE DONE

- Design and Code Utilities

3. PROBLEM ENCOUNTERED

- The input frequency pitch detection doesn't good as expected
- The guitar lessons are limited due to the knowledge and qualification of the author

4. SELF EVALUATION OF THE PROGRESS

- Need to start on other functions instead of just focus on Audio Recording and Pitch Detection

_____ _____

Supervisor's Signature Student's Signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 7 & Week 8**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

3. WORK DONE *[Please write the details of the work done in the fortnight]*

- Done all the proposed built-in utilities for guitar
- Stored music notes and able to move according to the beats
- Music sheet is able to move accordingly

2. WORK TO BE DONE

- Design and Combine GUI Functionality

3. PROBLEM ENCOUNTERED

- The accuracy of music notes matching is not accurate as expected
- Took too much time in implementing Guitar Tuner

4. SELF EVALUATION OF THE PROGRESS

- The whole picture of the system is almost complete, need to further improve the GUI and some other functions

_____ _____

Supervisor's Signature Student's Signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 9 & Week 10**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

4. WORK DONE *[Please write the details of the work done in the fortnight]*

- Completed the design of the system
- Recorded video for tutorial purpose
- Implemented the Last Replay function successfully
- Finished functionality integration testing

2. WORK TO BE DONE

- Individual module testing
- Full system testing

3. PROBLEM ENCOUNTERED

- GUI is user friendly but it is not attractive

4. SELF EVALUATION OF THE PROGRESS

- The overall system considered complete, need to speed up on doing the documentation

_____ _____

Supervisor's Signature Student's Signature

FINAL YEAR PROJECT BIWEEKLY REPORT

(Project II)

Trimester, Year: Year 3 Trimester 3	**Study week no.: Week 11 & Week 12**
Student Name: Tung Phien Phin	**Student ID: 09ACB03935**
Supervisor: Mr. Soong Hoong Cheng	
Project Title: Computer Assisted Music Practice For Guitar (CAMPGuitar)	

5. WORK DONE *[Please write the details of the work done in the fortnight]*

- Finished Individual Testing
- Finished Full System Testing
- Completed Chapter 6 and Chapter 7 of the full report

2. WORK TO BE DONE

- Documentation
- Create installer and poster

3. PROBLEM ENCOUNTERED

- Some minor problems and bugs determined from the system

4. SELF EVALUATION OF THE PROGRESS

- Able to complete the system in time, but there are some minor bugs to be hunt and fix.

_____ _____

Supervisor's Signature Student's Signature