

FINAL REPORT

by

Aaron Ennis

C00190504

Supervisor: Paul Barry

Music Application Project

Date: 18/04/2018

Institiúid Teicneolaíochta Cheatharlach



INSTITUTE *of*
TECHNOLOGY

CARLOW

At the heart of South Leinster

Table Of Contents

Table Of Contents	1
Introduction	2
Description Of Project	3
Description of Conformance to Specification and Design	4
Does the project match the specification?	4
Does the project meet the initial design?	4
What could have been improved?	4
Learning Process	5
Personal	5
Technical	5
Fast Fourier Transformation (FFT)	5
Window Functions	5
.WAV Files	5
Linear pulse-code modulation (LPCM)	6
Personal achievements	6
Review Of The Project	7
What went right?	7
What went wrong?	7
Challenges I encountered	8
What is there left to do?	9
What would I have changed?	9
Advice for someone starting a project like this	9
Which technology choices were right or wrong?	9
What went right?	9
What went wrong?	10
Conclusion	10
Acknowledgements	12

Introduction

This document is an overview on my experience with my final year project. In this document I will go into detail about how I feel about the end result of my project and other personal and technical topics. These topics involve personal experiences with working on this project and also personal experiences with the technology I used, what I have learned, what I think I could have done better, what went right or wrong, how I feel about the project meeting its aim, and what I think I could have changed to make the project better.

After reading this document, you should have a keen understanding of my views when I first came up with this idea for a project. I talk about whether I find the end result to be a success and if it fixes the problem it was created to fix.

Description Of Project

This project is a music application that allows a user to record a piece of music on a guitar. When a user creates a new recording, the application transcribes the audio by calculating the notes that were played in the recording, and displaying the notes in tablature form.

The application was created to make life easier for musicians that write and record their own music. The user will be able to run the application on their computer or laptop. All a user needs is a working microphone for the laptop or computer, and a guitar to use for the recording.

When the application is run, the user will have the option to record a piece of music. The user has the option to either record a 5, 10, 15, 20, 25, or 30 second clip. The audio saved by the user is saved locally and can be accessed at any time as it is stored in a "Recordings" folder in the applications folder. The audio file is saved in .WAV form so it can be played outside of the application. When the user records and saves a clip of music, the application will then transcribe the piece of music to tablature form. This is how the application makes life easier for musicians as it visualises their music.

The design of the User Interface (UI) should be as simple as possible. This is because the application was designed to make life easier for a user. The less button clicks a user will have to do to achieve the task they want to do the better. The UI should be clearly readable and easy to navigate even for first time users.

The idea for this project was created to stop a common problem musicians have when saving recordings of their music. The problem is when a recording is saved on a machine and forgotten about, it can be hard to remember how to play that piece of music. The projects aim is for guitarists to easily write, record, read, and playback music they have written.

Description of Conformance to Specification and Design

The overall specification and design for this project was to make sure that the application is as simple as possible in both design and use. The application was not about being flashy or pretty, but being effective in what it was designed to do. The main goal was eliminating the problem of a musician recording and saving a piece of music on a machine and forgetting about it. It can be hard to remember how to play that piece of music after coming back to it. This application was designed to show the user how they played the piece of music fixing this problem. As this was the main goal for the project, the more accurate the transcribing process is the better. This is why I was not concerned about how many features it has, but rather I was concerned about the project meeting its goal.

Does the project match the specification?

After completing this project, I believe it has met the original specification. I believe this because it achieves the goal of eliminating the problem of forgetting how to play a piece of music. Since this application was created with an original idea of my own, I think I have achieved what I had set out to do.

Does the project meet the initial design?

From looking at other music applications, I found that most of them were designed to have a simplistic look. The applications are aimed towards people who tend to prefer the performance of the application over how good it looks. I believe I have achieved the initial design as the application has a very simplistic look. It follows “The less button clicks a user will have to do to achieve the task they want to do the better” mentality I talk about in my software design document. The application does what it is supposed to do and is not drowned out by additional unnecessary features that could potentially clutter the screens.

What could have been improved?

Although I believe that this project has met the original specification, I also believe that it can be improved even more. In the timeframe given to complete this project, I was able to successfully create an application that can transcribe music. This was a huge task as there is no other application on the market that does this. If there was more time, I could have achieved more accurate transcriptions and improved the applications performance. Although the application does transcribe the correct notes most of the time, it does not do it 100%. This is because there is so many variables when recording (background noise, low quality mic, etc..). These improvements could have been made if there was more time.

Learning Process

Throughout the project I feel that I have learned a lot about developing an application and also a lot about myself as a developer. Some of these lessons came with trial and error and others came from reading articles and browsing online. At the end of this project I feel that what I have learned I can use in future projects and grow as a developer.

Personal

Personally I have learned a lot about myself as a developer. I now know that if I set my mind to a project, I can see it to the end and accomplish what I set out to do. I feel I have the ability and confidence to create applications in the future, even if it seems like it will be a hard thing to accomplish.

Technical

There has been a lot that I have learned by doing this project. This ranges from programming, to the physics of how sound works, to how computers interpret sound, to the mathematics involved in audio processing. The things I learned by doing this project are:

Fast Fourier Transformation (FFT)

The FFT was used in the application to transcribe the audio data. I read a lot in to this to get an understanding on how it works to apply it in to my project. By learning how the FFT works, I was able to comfortably work with the results I was getting. This is because when I got certain results I would know why.

Window Functions

Up until this project, I never had to work with a window function. From my understanding, they work by zoning in to the relevant data as if you were looking at the data through a window, hence the name. The type of window function I used in this project was a Hamming window as I found that it was the best for the job, as it works well with the FFT from my research online. It was interesting to see how to apply a window function to code.

.WAV Files

As I chose to use .WAV format to save the audio files, I had to learn about their architecture. I needed to know how they store their data so I could access it and use it in the application. The audio data was used to transcribe the music.

Linear pulse-code modulation (LPCM)

This is a method used to digitally represent sampled analog signals. I learned about LPCM when reading about the architecture of a .WAV file. I needed to know about LPCM because it was involved with the audio data stored in the .WAV files. This audio data was what I used to find the notes being played when transcribing.

Personal achievements

The main personal achievement that I achieved throughout this project is finishing it. Since the application was my own idea and design, I feel that I have accomplished a personal goal.

From my research, I found that there is no application on the market that has the same functionality as this one. I feel I have achieved something different and new.

One of the personal achievements that I have achieved is finishing the project on time and that I got all of the core functionality working. As I got the core functionality working, I was successfully able to transcribe music. This was a challenge I set myself to see if I was able to complete it.

Review Of The Project

What went right?

During the planning of this project, I anticipated that there would be aspects that I would find challenging. Some of these aspects were challenging but then some were straight forward. One of these aspects was finding a way to transcribe audio data. From my first approach of reading about guitar tuners and how they work, led me straight to the Fast Fourier Transformation (FFT). It turned out that this is the main way audio data is transcribed in guitar tuners and also other things like radio transmissions or phone transmissions. This allowed me to start working straight away.

The second aspect that went right straight away was eliminating the silent noise from the recordings. I had the idea that by adding a small clip of a sound file to the beginning of a recording and passing it through a normalizing function it would fix this. This approach did fix the problem. I then removed the sound clip from the beginning so the user does not have to hear it.

What went wrong?

Although I believe this project was an overall success, there were a few aspects that did not go according to plan. Half way through the project I encountered a problem with the main algorithm when transcribing the audio data. This became the biggest challenge I had to face when doing this project. The algorithm was not accurate and would give the wrong notes. Where I went wrong with this problem was I spent too much time trying to make the way I had it to work, instead of trying a different approach. This wasted time that could have been used progressing further.

In physics, a sound wave consists of different frequencies, and because of this the application could not distinguish when a steady note is being played. For example, on the open E string, there is a B frequency present. This causes the wrong note to be displayed from time to time making the tablature incorrect.

The third aspect that went wrong is how the application calculates what that octaves are of the notes. When I first began to write the algorithms for transcribing the notes, I found that the octaves detected were never 100% accurate. After coming back to try fix this problem, I found that this task is a lot more complex than I originally thought. From my research in to the matter I found that this is a common problem that has not been fixed. There are a few ways to get around this problem but there is no real answer to it. The reason this is a complex problem is because it is to do with the frequency of the sound wave. A computer can not accurately find the exact frequency which makes it hard to find the correct octave. Because of this problem, the

tablature looks wrong even though it is the right notes being displayed, it is just in the wrong octave.

Challenges I encountered

One of the main challenges I encountered in doing this project was knowing where to begin. This was because this type of application is not on the market and I had to figure out what the best approach would be to begin. I got over this by looking in to how a guitar tuner can determine what note is being played. I figured that I could use the same technique in transcribing the audio data. This technique was the Fast Fourier Transformation (FFT).

The second challenge I faced was the biggest of the project. This was trying to get accurate readings from the FFT. I was able to get the correct readings every couple of tries but this was not enough. My approach involved recording the file and then split the data in to chunks and then apply the FFT to the chunks. I found out the reason it was not accurate was because the audio data needs to be used through a window function (hamming window) before applying the FFT. By using my approach, the chunks could not be used in a window function. How I got past this problem was recording and transcribing the audio data in real time together. This fixed the problem by passing the stream of data coming from the microphone through the window function and then applied the FFT to that.

The third challenge I faced was eliminating white noise from the recordings. When a new recording was created and no noise was present (the recording is silent), the silence is amplified as the audio data goes through a normalizing function which levels out the noises. In this case, it would amplify silence creating white noise. If any noise was present in the recording, the silence would act the way it should. How I fixed this problem was I added a sound clip of a clap to the beginning of a new recording. I then passed it through the normalizing function which would level it out correctly, even if the recording was silent. Before I saved the audio data to a .WAV file, I removed the sound clip of the clap so the user only hears their recording.

The last challenging encounter that I faced was when I was creating the user interface (UI). This was that I could not have pop-up messages coming from different threads it had to be the main one. This is because when a pop-up message is used, that becomes the main window. If the pop-up message is on a separate thread from the main one, the UI would freeze. This was due to how the library I used (PyQt) works. The only way I could fix this problem was just letting the UI freeze. It would only freeze when the user is recording a piece of music. As the user would be playing the music to be recorded, they do not need access to the UI until they are finished recording. If I had the time I would have tried to fix this problem so the UI never freezes and always remains responsive.

What is there left to do?

One of the functionalities I would have liked to have in my project, would be the ability to export the tab and audio recordings as a package. I was not able to do this as I ran out of time.

I would have liked to have a way to stop a recording short. At the moment when a recording is being played back, there is no way to stop it you have to let it finish. This is because the way the threading is done in the project. The threadpool I use comes from a module named PThread from the PyQt library. To add this stop feature I would need to redo the threading in the application as the PThread threads could not be stopped once sent to complete a task.

When displaying the tablature, I would have liked to make it look neater. At the moment it can be aligned wrong because of the different sizes of the characters.

The last functionality I would have liked to have in my project, is for the user to be able to manually edit the tab if the application did not get it correct.

What would I have changed?

If I knew when I was starting to create the threading system for the project, that the module could not kill its threads once started, I would have taken a different approach. This lead in to the problem of not being able to stop a recording once it's being played.

Advice for someone starting a project like this

If I knew someone was to begin a music application like this I would tell them to make sure to read in to the physics of how sound works. The more I read in to this, the more I got an understanding of how to approach the project and make it possible to achieve.

Which technology choices were right or wrong?

What went right?

After looking in to the best languages and frameworks for audio programming, I found that Python had the most to offer. I chose to do the entire project in Python which turned out to be the right choice. I chose to use the Anaconda framework to develop the application in. This is because the framework comes pre-installed with a number of mathematical libraries that I needed for the transcription of audio data. An added benefit to using the Anaconda framework was the fact that it was easily installed and also the fact that installing new modules to it was straight forward.

At the beginning of the project I had to choose a file type to save the audio files in. After researching each of the possible file types, I chose .WAV files. I choose this file format because it seemed like the best option as the architecture was easy to work with. This turned out to be a good choice as there was a lot of modules built in to Python that allows you to access the audio data in .WAV files. They were also very easy to construct.

What went wrong?

When I was beginning to create the threading system for the project, I used a module named PThread from the PyQt library. This ended up being a mistake as the thread in this module could not be killed once sent to do a task. This lead in to problems that I had to compromise for because fixing this, I would have needed to redo the entire threading system which would have taken to much time.

A second thing that went wrong with the technology I chose was a conflict of the IDE (Spyder) I used and the PyQt library. They both are built with the same module. This lead to a problem where every time I compiled the PyQt code, I would have to restart the Kernel or the console would crash. After I decided to try and fix this problem, I found code that would stop the console from crashing that I needed to insert in to my main file. Although I fixed this problem, it wasted time having to restart the Kernel over and over. Since this was only a problem I would encounter while developing the application, I figured it was not a priority to fix which was a bad choice as it wasted time overall.

Conclusion

From the beginning to the end of this project there has been multiple different challenges I had to face. Although some of these challenges took a bit longer than I thought they would, I believe I overcame them with the best of my ability. I feel that this project was a success because it achieves the goal of eliminating the problem of forgetting how to play a piece of music. Since this application was created with an original idea of my own, I think I have achieved what I had set out to do.

After completing this project I feel that I have learned a lot about many different things. One of the things I have learned is that I have the skills to create an application of my own idea and design. Another thing I have learned that I believe is a valuable lesson, is the experience in how to approach a project as big as this one. The planning involved in staying on track and not fall behind played a big part in completing this project.

In conclusion to the project, I feel that it was an overall success as it achieves its original aim. The application was kept simple like the original design and the functionality works. I have

learned a lot about myself as a developer and also a lot about developing as a whole. I believe I can use this experience to help me when I work on future projects.

Acknowledgements

There are many people who I would like to thank, but I will keep this short and sweet. Firstly I would like to thank Paul Barry my supervisor for this project. Paul guided me through this project and gave me useful advice throughout which help in the final result. With this help, I learned of different approaches when tackling a project of this size.

Secondly I would like to thank all of my college friends that made this year a little less stressful as we were all going through the same thing.

I would like to thank all of the developers I talked to throughout this project. They helped me understand the fundamentals of the Mathematics involved in transcribing music.

Finally I would like to thank all of my friends, my family and my girlfriend. They helped make this year a less stressful experience and gave me the support I needed.