

RESEARCH DOCUMENT

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
About this project	3
Musical Notation	3
What is Musical Notation?	3
Sheet Music	4
Treble Clef	4
Bass Clef	5
Musical Notes	5
Tablature	6
Symbols	7
Preferred Notation for this Project	7
Applications	8
Existing Applications	8
Transcribe!	8
Anytune	9
Muscores	10
ScoreCloud	11
Technology	13
Languages	13
Language feature descriptions	14
Python	14
PyAudio	14
Wave	14
Java	14
Java Sound API	14
C#	15
Native Libraries	15
CSCore	15
C++	15
Marsyas	15
Other features	16
PortAudio	16
Csound	16
Conclusion	16

GUI Frameworks	16
PyQT	16
PyGUI	17
Libavg	17
Kivy	17
File Formats	18
Types of File Formats	18
Uncompressed	18
Lossless compression	19
Lossy compressed audio format	19
Common File Types	19
File Descriptions	20
Wav	20
M4a	21
Mp3	22
Aiff	22
Pulse-code modulation	23
Mathematical equations	24
Methods required	24
Fast Fourier Transformation	24
Window function	25
Quadratic Interpolation	25
Normalizing	25
Frequency to note	25
Reference List	26

About this project

This is a music application that will help a user to record and transcribe music. The purpose of this application is to easily record a clip of music on a guitar and have the application transcribe this clip into tablature notation. This will allow the user to create a playable audio clip they can listen to and be able to view how they play it.

Musical Notation

What is Musical Notation?

“Music notation or musical notation is any system used to visually represent aurally perceived music played with instruments or sung by the human voice through the use of written, printed, or otherwise-produced symbols.” ([Wikipedia, 2017](#)).

Put simply, musical notation is a way a musician can play a song without learning the song off by heart. It would be equivalent to reading the lyrics to a song and singing along with it. In some forms of musical notation, the musician can read the tempo, the key and the notes that are being played. This allows musicians who can read musical notation to be able to play along without knowing the song. This is how an orchestra with a large amount of people all playing the same song can stay in key, tempo, and all be playing the right notes together. There are many different types of musical notation. These different types all vary for a number of reasons which include history, culture and purpose. The Catholic church created the way the western world notates music. They did this by notating plainchant melodies so that the same chants could be used throughout the church. This allowed the same chants to be sang in every church to keep uniformity.

There are many different types of musical notation, some more popular and more relevant than others. The two main types that you would come across are:

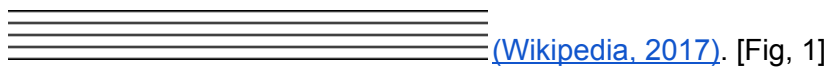
1. Sheet Music
2. Tablature

These two forms of notation are the most common you will come across as a musician. They are both useful in their own way. Although sheet music gives the reader a lot more information on the music they are playing, it is a lot harder to read and understand. This means people who have not studied music or sheet music can easily read tablature as it is a graphical representation of a fretted stringed instrument.

Sheet Music

“Sheet music is a handwritten or printed form of music notation that uses modern musical symbols to indicate the pitches (melodies), rhythms or chords of a song or instrumental musical piece.” ([Wikipedia, 2017](#)).

Sheet music is a useful way to display music. It can give the reader a lot of information about the song they are playing without listening to it. It is composed of a set of five horizontal lines and four spaces that each represent a different musical pitch. This is called a staff if you are in the US, or a stave if you are in the UK. These 5 lines are represented like this:



At the beginning of each bit of sheet music, there will be a clef. This clef, often a bass clef or treble clef, is a key signature indicating the key. This will also have the time signature, which typically has two numbers. This is read by the top number indicating how many beats are in a bar and the bottom number indicating the unit which the bars are composed of. For example 4/4 indicates there are four beats per bar and that the quarter note is the unit that the bar is made up from. This allows the musician to keep in time with how the music is being played, but also allows them to know what key they are in. There are a many more different types of clefs, including even more types of treble and bass clefs. The list goes on and on but these two are the most common. They are a visual representation of what note you start on in the bar.

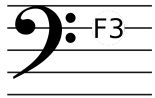
Treble Clef



([Wikipedia, 2017](#)) [Fig, 2]

“Where the G-clef is placed on the second line of the stave, it is called the treble clef. This is the most common clef used today, and the only G-clef still in use. For this reason, the terms G-clef and treble clef are often considered synonymous. The treble clef was historically used to mark a treble, or pre-pubescent, voice part.” ([Wikipedia, 2017](#)).

Bass Clef



([Wikipedia, 2017](#)) [Fig, 3]

“When the F-clef is placed on the fourth line, it is called the bass clef. This is the only F-clef used today so that the terms "F-clef" and "bass clef" are often regarded as synonymous.”

([Wikipedia, 2017](#)).

Musical Notes

After the clef follows notes; The musical notes are a visual representation of the pitch and duration of sound. Depending on which line they are on or in between on the staff, tells you which note is being played. For example if the note is on the bottom line on a bar with a treble clef, the note is an E. There are different symbols that represent different musical notes. They all have their own duration of sound which essentially allows you to know the beat of the song. Depending on which symbol is present, you know how long you need to hold on the note for.



([Dreamstime, 2017](#)) [Fig, 4]

Putting all these factors and rules together you create sheet music. Knowing what each symbol does allows the musician to be able to read the music and get an idea how the piece will sound. Although each symbol has a definitive meaning, sheet music takes time to be able to learn and read fluently. It is much more complicated than tablature, but it gives you more information. Saying this, if a musician finds it hard to understand sheet music, they can get by with only

knowing tablature. Usually a musician only needs to know sheet music if they are being classically trained or have a career as a session musician.



([Violin Sheet Music, 2017](#)) [Fig, 5]

Tablature

When being compared to sheet music, tablature is a lot easier to read and understand. Even if a person has never seen tablature before, there is a chance they could figure it out. This is not the case for sheet music which is why many new guitarists learn tablature and not sheet. Tablature is used for fretted stringed instruments as it makes them easy to read. Unlike sheet music, tablature is written indicating instrument fingering rather than musical pitches. Depending on the instrument, there is a number of lines that represent each string. For example a standard guitar tab has 6 lines whereas a standard bass tab only has 4. This makes it very easy to figure out what is being played. On each string, there are numbers which represent the fret that the note is being played.



([Learn and Master, 2011](#)) [Fig, 6]

If there is a straight line of numbers going vertically, this is considered a chord.

```
e |-----
B |-----
G | 5--4--
D | 5--4--
A | 3--2--
E |-----
```

([Tablature de Guitare, 2017](#)) [Fig, 7]

Here we can see in “Fig, 7” how each of the lines represent the 6 strings of a guitar in standard E tuning. In this instance the type of chord being played is a bar chord. The two chords being played are C and a B respectively. Although we can see what chords/notes are being played in tablature, there is also symbols just like sheet music, that tell us what is being done to the string

other than just playing the notes. For example, if you were to slide from an A note up to a C note, it would be represented using the symbol “ / ”.

Symbols

Symbol	Meaning
H	Hammer-On
P	Pull-Off
B	Bend String Up
R	Release Bend
/	Slide Up
\	Slide Down
V	Vibrato
T	Write Hand Tap
X	Play note with heavy damping

([Google, 2002](#)) [Fig, 8]

Preferred Notation for this Project

As one of the main functionalities of this project is to make life easier for a musician to record and read music they have written, I have found the best option is to display their music in tablature. From my findings, tablature is the easier form of notation to understand. This means there is more of a user base as tablature is known by a considerably larger number of people due to its simplicity.

Applications

Existing Applications

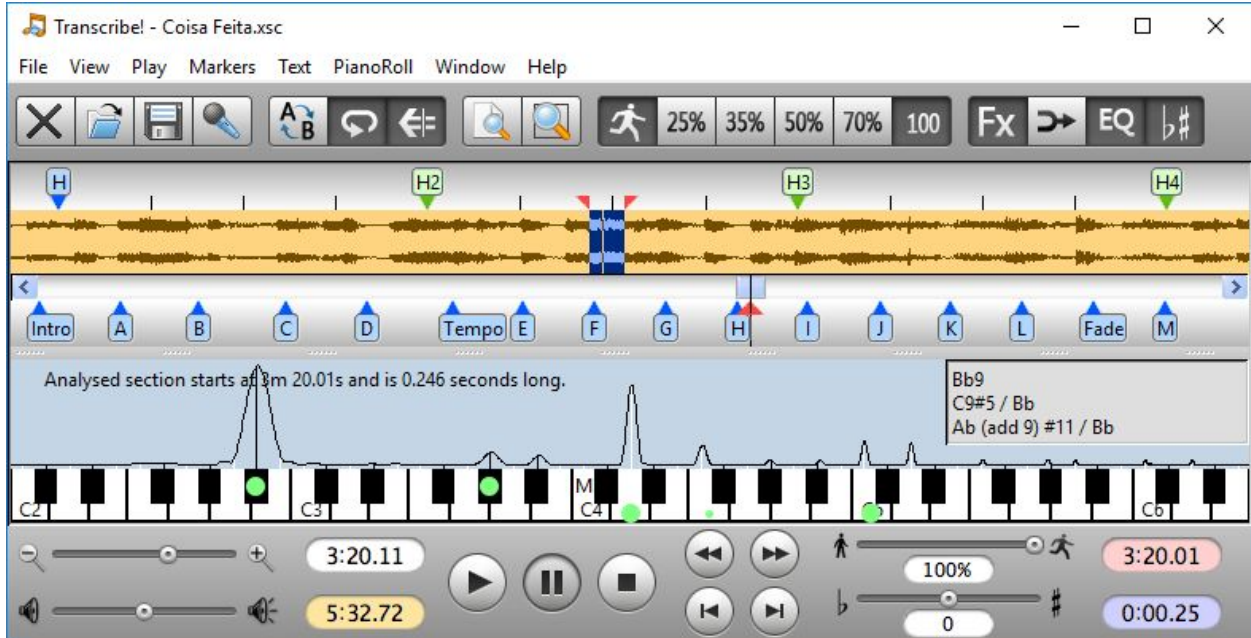
Currently on the market, there are many different musical applications that have various features. Some are for recording music, editing music, helping write sheet music, or helping transcribe music. Out of all the applications that I have found in my research, there were no applications that could transcribe music directly. Some of the applications that are being used as transcribing applications did not do the transcribing for the user. Instead, they would help the user transcribe the song by ear by slowing down the songs. Some popular applications that are relevant to my project are:

Transcribe!

Transcribe! Is a software that helps a user transcribe a piece of music in various ways. Some of the functionalities that this application has to help transcribe music are:

- Slowing down the audio without pitch change
- Analysis chords
- Displays notes present
- Ability to navigate audio with the use of markers

This software is available on each platform. Although the software does not transcribe music for the user, it helps make it easier for the user to be able to transcribe the music by ear. This could be useful to someone who would like to learn how to play a song but there is no sheet music or tablature available.



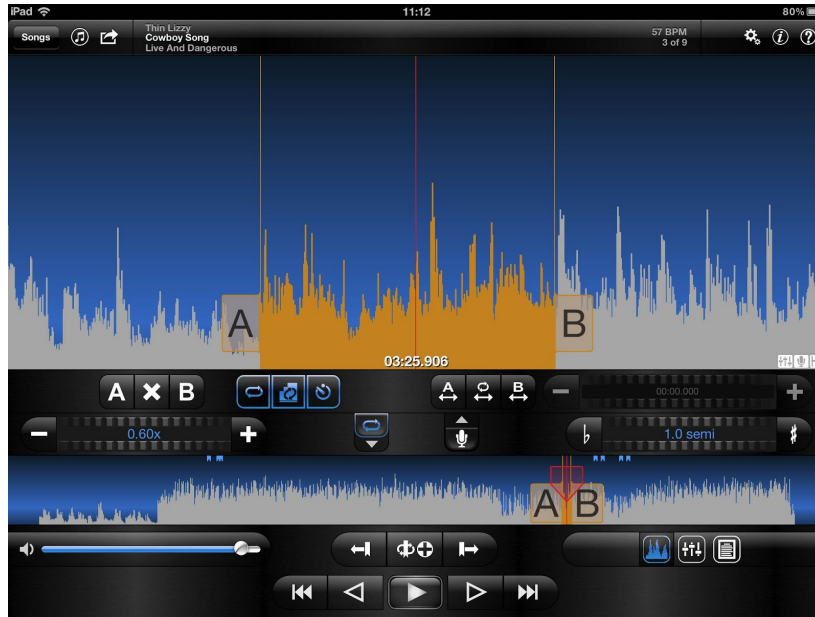
(Transcribe!, 2017) [Fig. 9]

Anytune

Anytune is a music software mainly available on Mac and IOS devices but also can be used on Windows and Android devices. The software has received the best Mac music application award of 2013. It is similar to Transcribe! but it has some extra features that could make it more useful. Some main features include:

- Slowing down the audio
- Ability to pick the pitch
- Ability to navigate audio with the use of markers
- Ability to pinpoint an instrument
- Ability to mute instruments

With these added features, Anytune appears to be the more useful software for transcribing music by ear.



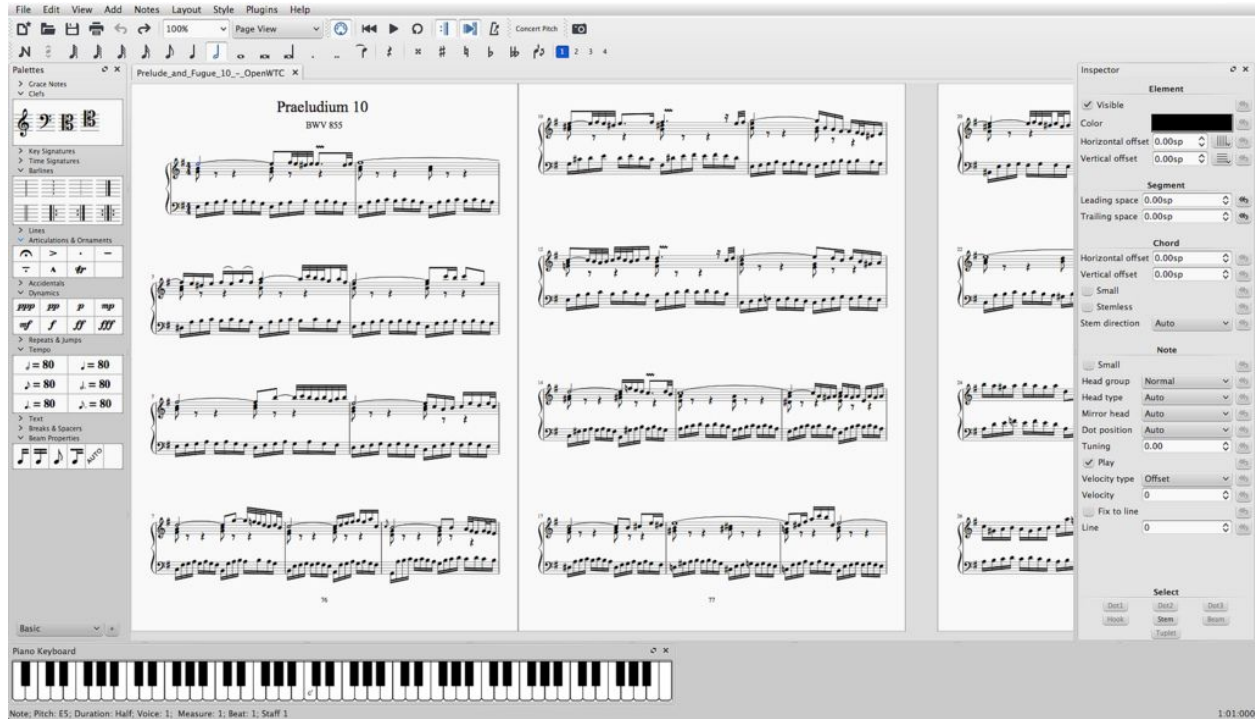
([Music App Blog, 2012](#)) [Fig, 10]

Musescore

MuseScore is open source music notation software that runs on Windows, MacOS, and Linux, and is available in over forty different languages. It features an easy to use WYSIWYG editor with audio score playback for results that look and sound beautiful with features that help create, playback, and print sheet music for free. It has feature like:

- Create sheet music, fast and easy
- WYSIWYG, notes are entered on virtual page
- Unlimited number of staves & up to four voices per staff
- Easy and fast note entry with mouse, keyboard or MIDI
- Playback with integrated sequencer and FluidSynth software synthesizer
- Import of xml, mxl, md, bww, ove, mgu, sgu, cap, kar
- Export to midi, wav, flac, ogg, pdf, png, ps, svg, ly

The interface for this application is designed to make it easy to navigate and write music.



([Flickr, 2015](#)) [Fig, 11]

At the bottom of the image “Fig, 11” you can see a small keyboard. This is a MIDI keyboard that a user can directly play notes written to the sheet music, instead of dragging and dropping the symbols to the staff.

ScoreCloud

ScoreCloud is a software that can instantly turn your songs into sheet music. It is music notation software ideal for musicians, students, teachers, choirs and bands as well as composers and arrangers. It is available for both Windows and Mac devices and has software available for mobile devices. It can be used as free software or commercial.

“ScoreCloud is marketed as the only service offering the ability to record audio, transcribe it to MIDI, transform it to sheet music, and share the sheet music online. Score Cloud's ability to transcribe real-time MIDI input into sheet music has been praised in the British music technology magazine Sound on Sound.” ([Wikipedia, 2017](#))

As this software has the ability to convert real time audio into sheet music, this makes this software the closest and most relevant to my project. This software contains features like:

- Instant Score from MIDI or Audio
- Ability to record directly from MIDI keyboard or microphone

- Arrange and Edit your Score
- Print, Export and Share files
- Audio and MIDI transcription
- Fully featured score editor
- Synchronization and sharing through the cloud
- Major export formats: Midi, MP3, PDF, MusicXML

The software allows the user to record audio which the software will then analyse and produce sheet music that can then be edited by hand.



(Wikipedia, 2017)

[Fig, 12]

Technology

Languages

In my findings, it seems C#, C++, Python and Java all have something to offer when it comes to coding with audio. Each language has their own features to offer and have either libraries or frameworks that assist with this type of programming. For my project I am looking for a language or framework that has great documentation and online support. A language with various features to choose from that can work well with an application designed for recording and playing audio files. As the application will be using different files, the language will need to have good path finding abilities. A bonus to all the languages but not a necessity, would be how comfortable I am with the language to reduce the learning curve and save time.

Language	Features	Pros	Cons
Python	PyAudio Wave	Great documentation Many libraries to choose from Easy code to read and understand Easy to read compiling errors Great for visualising data I'm comfortable with the language Great for file pathing	Non-native audio libraries Non-semantic errors Managed language
Java	Java Sound API	Great documentation In build GUI system Deployable on any platform Great for try/catch and encapsulation	A lot of boilerplate code I'm not comfortable with the language Managed language
C#	Native audio libraries	Easy to read code	Managed language

	CSCore	Native audio library Good documentation	I'm not comfortable with the language Only deployable on Windows
C++	Marsyas	Non-managed language I'm comfortable with the language A lot of libraries and online support Pointers Deployable on any platform Great for threading Customisable makefiles	Hard to read documentation Memory leaks A lot of boilerplate code

Language feature descriptions

Python

PyAudio

PyAudio provides Python bindings for PortAudio, the cross-platform audio I/O library. With PyAudio, you can easily use Python to play and record audio on various platforms. Works well with Wave which allows you to read, record and edit audio.

Wave

The wave module provides a convenient interface to the .WAV sound format. It does not support compression/decompression, but it does support mono/stereo.

Java

Java Sound API

The Java Sound API can be used for controlling audio playback, audio capture, MIDI synthesis, and basic MIDI sequencing. It includes source code, and is designed to work with the Java 2 SDK. This API is supported by an efficient sound engine which guarantees high-quality audio mixing and MIDI synthesis capabilities for the platform.

C#

Native Libraries

As C# was created for windows devices, it comes with many features that allow you to access in built features of windows including easy access to the device's microphone or speakers.

CSCore

CSCore is a free .NET audio library which is completely written in C#. Although it is still a rather young project, it offers many features like playing or capturing audio, encoding or decoding many different codecs and effects.

C++

Marsyas

“Marsyas is an open source software framework for rapid prototyping and experimentation with audio analysis and synthesis with specific emphasis to music signals and Music Information Retrieval. The basic goal is to provide a general, extensible and flexible architecture that allows easy experimentation with algorithms and provides fast performance that is useful in developing real time audio analysis and synthesis tools.”([Marsyas, 2015](#))

Other features

Although there are many libraries and frameworks for each language, there are libraries and frameworks available that are not language specific.

PortAudio

“PortAudio is a free, cross-platform, open-source, audio I/O library. It lets you write simple audio programs in 'C' or C++ that will compile and run on many platforms including Windows, Macintosh OS X, and Unix (OSS/ALSA). It is intended to promote the exchange of audio software between developers on different platforms. Many applications use PortAudio for Audio I/O. PortAudio provides a very simple API for recording and/or playing sound using a simple callback function or a blocking read/write interface. Example programs are included that play sine waves, process audio input (guitar fuzz), record and playback audio, list available audio devices, etc.”([PortAudio, 2017](#))

Csound

Csound is a computer programming language for sound, also known as a sound compiler or an audio programming language, or more precisely, an audio DSL. It is free software, available under the LGPL. One of the main principles in Csound development is to guarantee backwards compatibility.

Conclusion

Although each language has their own benefits to bring, Python seems to be the best option. This is because it has great and easy to use mathematical libraries that are required for the project. You can easily install any module you need for free and there seems to be a module for almost everything. It also has a lot of GUI frameworks to offer that are free. On top of all of this, Python has a lot of help online through online documents or help forums.

GUI Frameworks

PyQT

Qt is a multi-licensed cross-platform framework written in C++ for Python. If your application is completely open source, you can use Qt for free under the community license; otherwise you'll

need a commercial license. Qt has been around for a long time and was owned by Nokia for a while; it's a very comprehensive library of tools and APIs, widely used in many industries, and covers many platforms including mobile. If a gadget such as a SatNav has a GUI, there's a good chance it'll be Qt based.

PyGUI

Compared to Kivy and PyQt, PyGUI is considerably simpler and just for Unix, Macintosh and Windows platforms. Developed by Dr. Greg Ewing at the University of Canterbury in New Zealand, the MVC framework focuses on fitting into the Python ecosystem as easily as possible.

One of the platform's aims is to interpose as little code as possible between the Python application and the platform's underlying GUI so the application's display always reflects the native GUI of the platform. If you're after a simple and quick way to learn GUI, start with this one.

Libavg

This is another third-party library, written in C++ and scripted from Python, with properties of display elements as Python variables, a full-featured event handling system, timers (setTimeout, setInterval), support for logging and more. Like Kivy, libavg uses OpenGL and makes use of hardware acceleration.

Libavg runs on Linux, Mac OS X and Windows, and is open source and licensed under the LGPL. It's been used extensively for artistic exhibitions and has a wide range of features such as a layout engine that can deal with thousands of objects (images, text, videos and camera output), fast video output, and a markup system for displaying text, as well as GPU shader effects such as blur, Chromakery and more. Plugins written in C++ have access to all libavg internals.

Kivy

One of the more interesting projects, the liberal MIT-licensed Kivy is based on OpenGL ES 2 and includes native multi-touch for each platform and Android/iOS. It's an event-driven framework for Python, based around a main loop, and is thus very suitable for game development. Your application adds callbacks from the main loop at a scheduled frequency, or by one-off trigger. The Kivy framework is very powerful for handling everything from widgets to animation, and includes its own language for describing user interface and interactions.

If you want to create cross-platform graphical applications, or just need a very powerful cross-platform GUI, Kivy is highly recommended.

File Formats

“An audio file format is a file format for storing digital audio data on a computer system. The bit layout of the audio data (excluding metadata) is called the audio coding format and can be uncompressed, or compressed to reduce the file size, often using lossy compression. The data can be a raw bitstream in an audio coding format, but it is usually embedded in a container format or an audio data format with defined storage layer.”([Wikipedia, 2017](#))

Types of File Formats

Audio file formats store digital audio which is a technology that can be used for sound recording and reproduction using audio signals that have been encoded in digital form. This is how the computer can read the sound waves and produce the sound. There are three major groups of audio file formats.

1. Uncompressed
2. Lossless compression
3. Lossy compression

Uncompressed

One major uncompressed audio format, LPCM(Linear pulse-code modulation), is the same variety of PCM(Pulse-code modulation) as used in Compact Disc Digital Audio and is the format most commonly accepted by low level audio APIs and D/A converter hardware. Pulse-code modulation (PCM) is a method used to digitally represent sampled analog signals. In a PCM stream, the amplitude of the analog signal is sampled regularly at uniform intervals, and each sample is quantized to the nearest value within a range of digital steps. Linear pulse-code modulation (LPCM) is a specific type of PCM where the quantization levels are linearly uniform.

These formats can be stored on a computer as a raw audio format, but PCM is usually stored as .WAV (Windows) or .AIFF (macOS). The AIFF format is based on the Interchange File Format (IFF), and the WAV format is based on the similar Resource Interchange File Format (RIFF). Both formats are designed to store different types of audio formats like, lossless and lossy. They just add a small, metadata-containing header before the audio data to declare the format of the audio data, such as LPCM with a particular sample rate, bit depth, endianness and number of channels.

Lossless compression

“A lossless compressed format stores data in less space without losing any information. The original, uncompressed data can be recreated from the compressed version.

Uncompressed audio formats encode both sound and silence with the same number of bits per unit of time. Encoding an uncompressed minute of absolute silence produces a file of the same size as encoding an uncompressed minute of music. In a lossless compressed format, however, the music would occupy a smaller file than an uncompressed format and the silence would take up almost no space at all.” ([Wikipedia, 2017](#))

Lossy compressed audio format

“Lossy compression enables even greater reductions in file size by removing some of the audio information and simplifying the data. This of course results in a reduction in audio quality, but a variety of techniques are used, mainly by exploiting psychoacoustics, to remove the parts of the sound that have the least effect on perceived quality, and to minimise the amount of audible noise added during the process. The popular MP3 format is probably the best-known example, but the AAC format found on the iTunes Music Store is also common. Most formats offer a range of degrees of compression, generally measured in bit rate. The lower the rate, the smaller the file and the more significant the quality loss.” ([Wikipedia, 2017](#))

Common File Types

wav	Standard audio file container format used mainly in Windows PCs. Commonly used for storing uncompressed (PCM), CD-quality sound files, which means they can be large in size around 10 MB per minute. WAV files can also contain data encoded with a variety of (lossy) codecs to reduce the file size. Wav files use a RIFF structure and stores metadata in a chunk before the audio data.
m4a	An audio-only MPEG-4 file, used by Apple for unprotected music downloaded from their iTunes Music Store. Audio within the m4a file is typically encoded with AAC, although lossless ALAC may also be used. This is also the file format that sound recordings are saved as when using the voice recorder application that comes with a windows machine.

mp3	MPEG Layer III Audio. It is the most common sound file format used today. This uses a lossy compressed file format.
aiff	A standard audio file format used by Apple. It could be considered the Apple equivalent of wav.

File Descriptions

Wav

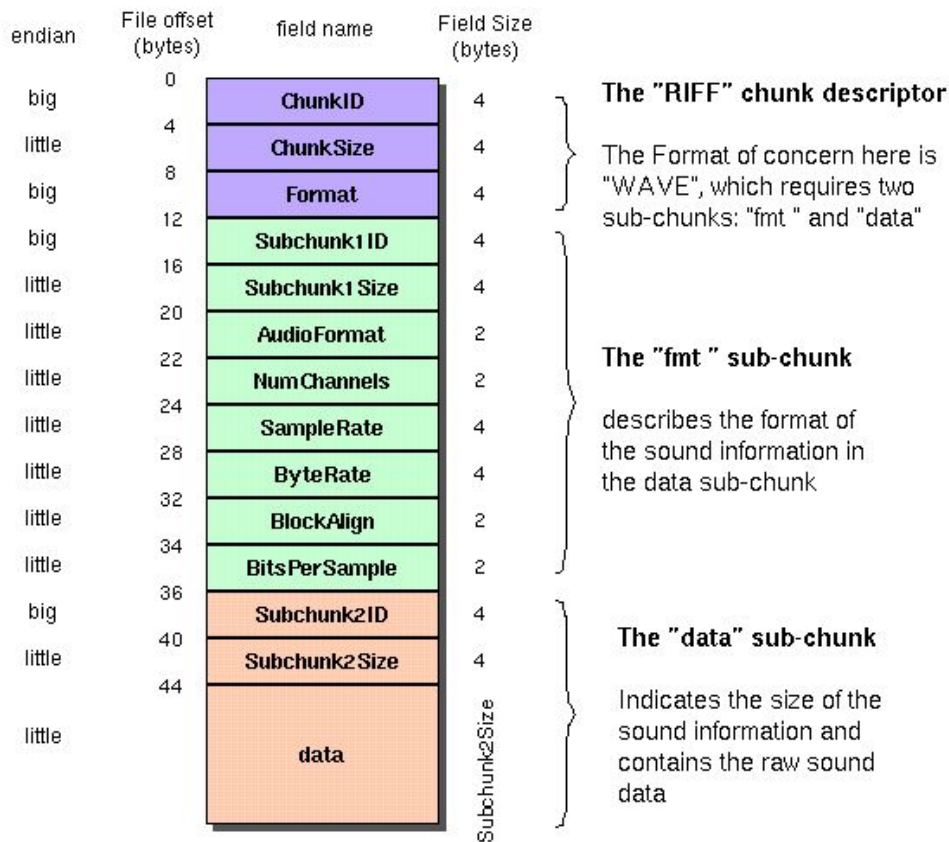
File name extension:

.wav .wave

Description:

Waveform Audio File Format (wav) is a Microsoft and IBM audio file format standard for storing an audio bitstream on PCs. It is an application of the Resource Interchange File Format (RIFF) bitstream format method for storing data in "chunks". It is the main format used on Windows systems for raw and typically uncompressed audio. The usual bitstream encoding is the linear pulse-code modulation (LPCM) format.

(PCM and LPCM explained below)



(soundfile.sapp.org, 2017) [Fig, 13]

M4a

File name extension:

.mp4 .m4a .m4p .m4b .m4r .m4v

Description:

M4A is a file extension for an audio file encoded with advanced audio coding (AAC) which is a lossy compression. M4A was generally intended as the successor to MP3, which had not been originally designed for audio only but was layer III in an MPEG 1 or 2 video files. M4A stands for MPEG 4 Audio. Audio-only MPEG 4 container files usually have an M4A file extension. M4A files are unprotected. Protected files usually have an M4P file extension. M4A files sound better than MP3 files when encoded at the same bit rate due to some enhancements made to the format like:

- Compression based on perception -- if it is not within human perception, that sound data may be lost without notable effect to quality.

- Smaller sample block size of 120 or 128 samples rather than 192 for changing (or transient) signals, allowing for more precise details where they are needed.
- Larger block sizes in stationary signals, 1024 or 960 vs MP3's 576 sample blocks, allowing for fewer data to represent a portion of the music that does not have the same complexity.

Mp3

File name extension:

.mp3

Description:

The MP3 format is a compression system for music. The goal of using MP3 is to compress a CD-quality song by a factor of 10 to 14 without noticeably affecting the CD-quality sound. With MP3, a 32-megabyte song on a CD compresses down to about 3 MB. This allows an mp3 file be downloaded faster. Both M4A and MP3 file extensions are used for audio-only files.

Aiff

File name extension:

.aiff .aif .aifc

Description:

Audio Interchange File Format (AIFF) is an audio file format standard used for storing sound data for personal computers and other electronic audio devices.

“The audio data in most AIFF files is uncompressed pulse-code modulation (PCM). This type of AIFF files uses much more disk space than lossy formats like MP3—about 10 MB for one minute of stereo audio at a sample rate of 44.1 kHz and a bit depth of 16 bits. There is also a compressed variant of AIFF known as AIFF-C or AIFC, with various defined compression codecs.

In addition to audio data, AIFF can include loop point data and the musical note of a sample, for use by hardware samplers and musical applications.”([Wikipedia, 2017](#))

An AIFF file is divided into many chunks just like wav files. Each chunk is identified by a chunk ID more broadly referred to as FourCC. The types of chunks found in AIFF files:

- Common Chunk (required)
- Sound Data Chunk (required)
- Marker Chunk
- Instrument Chunk
- Comment Chunk
- Name Chunk
- Author Chunk
- Copyright Chunk
- Annotation Chunk
- Audio Recording Chunk
- MIDI Data Chunk
- Application Chunk
- ID3 Chunk
- Metadata

The metadata stores information like name, author, comment, annotation, and copyright chunks.

Pulse-code modulation

“Pulse-code modulation (PCM) is a method used to digitally represent sampled analog signals. It is the standard form of digital audio in computers, compact discs, digital telephony and other digital audio applications. In a PCM stream, the amplitude of the analog signal is sampled regularly at uniform intervals, and each sample is quantized to the nearest value within a range of digital steps.

Linear pulse-code modulation (LPCM) is a specific type of PCM where the quantization levels are linearly uniform. This is in contrast to PCM encodings where quantization levels vary as a function of amplitude. Though PCM is a more general term, it is often used to describe data encoded as LPCM.” ([Wikipedia, 2018](#))

Mathematical equations

For this application to be achieved, a lot of mathematics and physics come in to play as we are working with sound. The sound the user produces when creating a recording needs to be analysed for application to work.

Methods required

There are a number of methods that are required when working with and analysing audio data. These methods are:

- Fast Fourier Transformation
- Window Function
- Quadratic Interpolation
- Normalizing
- Frequency To Note

Each one of these methods are required to record and transcribe audio data. These are the main methods required from my findings on how to achieve the goal of the application.

Fast Fourier Transformation

“A fast Fourier transform (FFT) is an algorithm that samples a signal over a period of time (or space) and divides it into its frequency components. These components are single sinusoidal oscillations at distinct frequencies each with their own amplitude and phase.”([Wikipedia, 2018](#))

The Fast Fourier Transformation (FFT) is a method used to find a frequency in audio data. The frequencies found are a direct link to the note being played at that time. This is because a frequency is unique to each note. The FFT was derived from the Discrete Fourier Transformation (DFT), but the DFT was too heavy on computer resources. Using Log algorithms, the FFT is a lot faster and lighter to use over the DFT.

The algorithm takes in a sequence (audio data) and splits it up into evenly spaced bins based on time. Each bin is used to find the frequency at that given time. By doing this, the FFT can show the different frequencies present in the audio signal and when the frequencies are present. This is the method a guitar tuner uses to find the notes being played.

Window function

“In signal processing, a window function (also known as an apodization function or tapering function) is a mathematical function that is zero-valued outside of some chosen interval. For instance, a function that is constant inside the interval and zero elsewhere is called a rectangular window, which describes the shape of its graphical representation. When another function or waveform/data-sequence is multiplied by a window function, the product is also zero-valued outside the interval: all that is left is the part where they overlap, the "view through the window".”([Wikipedia,2018](#))

There are many different types of window functions. Based on the problem you are facing, you choose the best suited window function. In this application, it would seem that the Hamming Window. From my findings, this is the best suited window when working with the FFT.

Hamming window = $0.5(1 - \cos(2\pi n / N - 1))$ where N is the number of samples per window.

Quadratic Interpolation

Quadratic Interpolation is a method used to find the true peak of a sinusoid. It is required to achieve accurate results when find the frequency in a FFT bin. If the true peak is not found the results may be off and it will give back the wrong frequency.

Normalizing

“Audio normalization is the application of a constant amount of gain to an audio recording to bring the average or peak amplitude to a target level (the norm). Because the same amount of gain is applied across the entire recording, the signal-to-noise ratio and relative dynamics are unchanged.”([Wikipedia, 2018](#))

Frequency to note

From my findings, the best way to find the note of a frequency is to use the MIDI number. Each MIDI number corresponds directly to a note. When the frequency is found, the MIDI note can be determined using this algorithm:

$$69 + 12 * \text{np.log}_2(F/440.0)$$

The “69” in the algorithm is a reference to the A4 note as 69 is the MIDI number of that note. “F” is the entered frequency and “440.0” is the frequency of an A4. In this example the A4 is used as reference and a starting point to find the note.

Reference List

Wikipedia. (2017). *Musical Notation*. Available: https://en.wikipedia.org/wiki/Musical_notation. Last accessed 28/10/2017.

Wikipedia. (2017). *Sheet Music*. Available: https://en.wikipedia.org/wiki/Sheet_music. Last accessed 28/10/2017.

Wikipedia. (2017). *Staff (music)*. Available: [https://en.wikipedia.org/wiki/Staff_\(music\)](https://en.wikipedia.org/wiki/Staff_(music)). Last accessed 28/10/2017.

Wikipedia. (2017). *Clef*. Available: https://en.wikipedia.org/wiki/Clef#Treble_clef. Last accessed 28/10/2017.

Victor Kunz. (2017). *Musical Notes Symbols*. Available: <https://www.dreamstime.com/royalty-free-stock-photo-music-notes-symbols-image23843195>. Last accessed 28/10/2017.

Joseph Brackett. (2017). *Simple Gifts*. Available: <http://violinsheetmusic.org/title/s/simple-gifts/>. Last accessed 28/10/2017.

Learn and Master. (2011). *Guitar Tab vs. Standard Notation*. Available: <https://www.learnandmaster.com/guitar-blog/gibsons-learn-master-guitar/guitar-tab-standard-notation>. Last accessed 28/10/2017.

Tablature de Guitare. (2017). *Tablature*. Available: <https://apprendre-laguitare.com/tablature-guitare/>. Last accessed 28/10/2017.

Google. (2002). *Method and system for improved tablature notation in digital music scores*. Available: <https://www.google.com/patents/WO2000034936A9>. Last accessed 18/10/2017.

Transcribe!. (2017). *Transcribe! screen shots*. Available: <https://www.seventhstring.com/xscribe/screenshots.html>. Last accessed 29/10/2017.

Music App Blog. (2012). *Music app review*. Available: <http://musicappblog.com/anytune-pro-hq-app-review/>. Last accessed 29/10/2017.

Flickr. (2015). *Piano Score*. Available: <https://www.flickr.com/photos/musescore/16655673190>. Last accessed 29/10/2017.

Wikipedia. (2017). SoundCloud. Available: <https://en.wikipedia.org/wiki/ScoreCloud>. Last accessed 29/10/2017.

Marsyas. (2015). *Overview*. Available: <http://marsyas.info/about/overview.html>. Last accessed 30/10/2017.

PortAudio. (2017). *Overview*. Available: <http://www.portaudio.com>. Last accessed 30/10/2017.

Wikipedia. (2017). *Audio File Format*. Available: https://en.wikipedia.org/wiki/Audio_file_format. Last accessed 02/11/2017.

Soundfile.sapp.org. (2017). *WAVE PCM soundfile format*. Available: <http://soundfile.sapp.org/doc/WaveFormat/>. Last accessed 02/11/2017.

Wikipedia. (2017). *Audio Interchange File Format*. Available:
https://en.wikipedia.org/wiki/Audio_Interchange_File_Format. Last accessed
02/11/2017.

Wikipedia. (2018). Pulse-code modulation. Available:
https://en.wikipedia.org/wiki/Pulse-code_modulation. Last accessed
28/03/2018.

Wikipedia. (2018). Fast Fourier Transformation. Available:
https://en.wikipedia.org/wiki/Fast_Fourier_transform. Last accessed
27/03/2018.

Wikipedia. (2018). Window Function. Available:
https://en.wikipedia.org/wiki/Window_function. Last accessed 27/03/2018.

Wikipedia. (2018). Audio normalization. Available:
https://en.wikipedia.org/wiki/Audio_normalization. Last accessed 27/03/2018.