1. Introduction

The purpose of this sub-project is to extract the sound envelopes that compose any sound and categorize the dominant frequencies in specific bands. Sound envelopes are composed of Attack, Decay, Sustain, Release. The input is extracted either from given .wav files or live from microphone that captures sound from the surrounding above a certain threshold. This threshold is altered in real-time, affected by **standard deviation** of previous and current values of amplitude (**mean value**). After recording specific number of samples, with a max length, each sample is processed so as to extract the above characteristics. Frequencies are categorized in bands using Fourier Analysis and simple Algorithmic thought. The output are the envelope values in msec, average amplitude and dominant frequency values in certain frequency bands.

2. Tools

a. Pure Data

The whole sub-project has been build using Pure Data. Pure Data (or just Pd) is an open source visual programming language for multimedia. Algorithmic functions are represented in Pd by visual boxes called objects placed within a patching window called a canvas. Data flow between objects are achieved through visual connections called patch cords. Each object performs a specific task, which can vary in complexity from very low-level mathematical operations to complicated audio or video functions such as reverberation, FFT transformations, or video decoding.

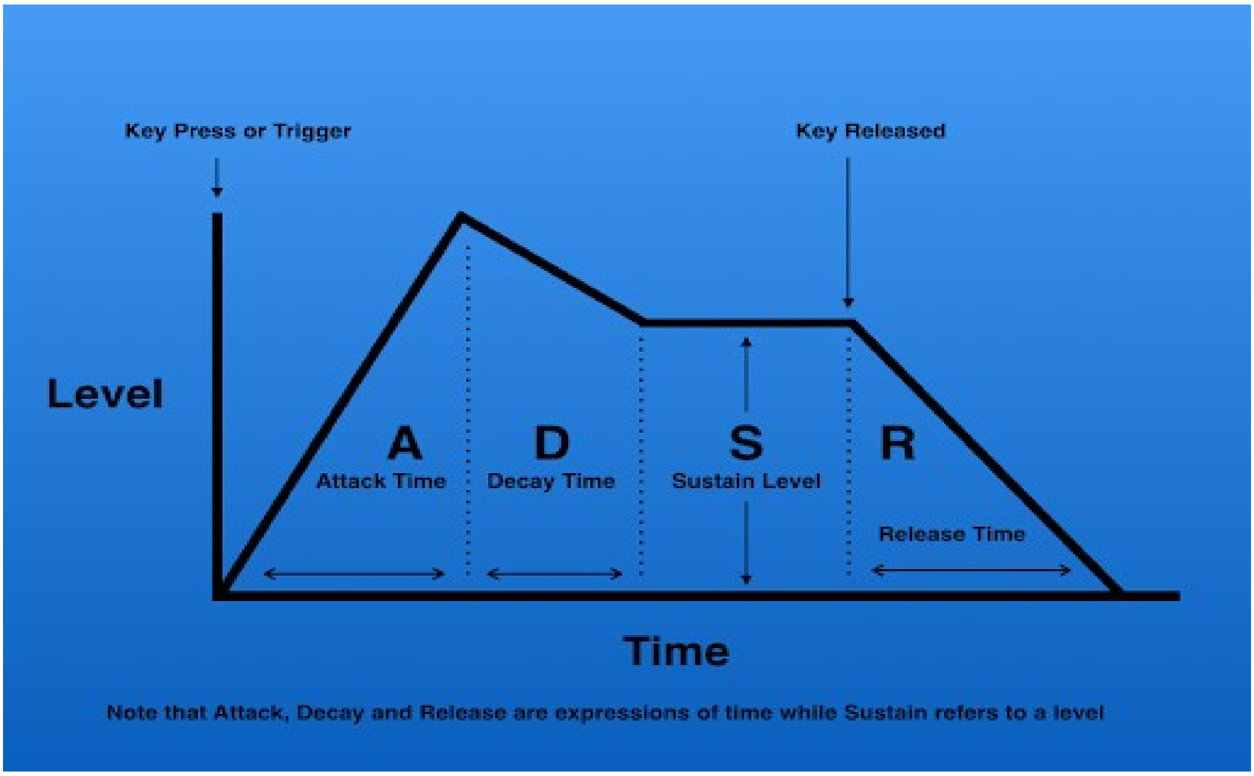
b. Shell Scripting

A shell script is a computer program designed to be run by the Unix shell, a command-line interpreter. The various dialects of shell scripts are considered to be scripting languages. Typical operations performed by shell scripts include file manipulation, program execution, and printing text. In the project only one script is used to create a text file which consists of .wav names we want our pd patch to analyze.

3. Terminology Explained

-Attack, Decay, Sustain, Release - A.D.S.R. Envelope
-Decibel Threshold
-Fourier Transform (Used for frequency analysis)

3.1 ADSR Envelope (ATTACK, DECAY,SUSTAIN,RELEASE)



Attack time: Attack is the duration or time that it takes for a signal to reach the highest point of amplitude after being triggered

Decay time: Decay, like attack, is a parameter that sets a duration – But rather than the time it takes to increase the volume level, decay represents the time it takes to drop down to the sustain level after reaching the initial peak of the attack phase.

Sustain time: Sustain is the only factor in ADSR that doesn’t represent time. Although it doesn't represent time, in our project we measure the duration of sustain level. The sustain is a level of amplitude that the signal remains on for as long as the sample/synth/sound is being triggered. In regards to a keyboard, the duration of the sustain is determined by how long you hold down a key.

Release time: Release, as with attack and decay, represents a change over time. The release phase begins as soon as the sample stops being triggered (for example when you stop holding the keyboard's key down).
The release parameter determines how long it takes for the sound to fade out completely (or eventually fall below a threshold) from the sustain level. You can think of it as the opposite of the attack in a sense.

3.2 Db Threshold

In order to capture sounds worth of processing we record only if the current incident is above a predefined db level which is also altered live during the recording by measuring the standard deviation of the sound level.

3.3 Fourier Transform

In mathematics, a Fourier transform (FT) is a mathematical transform which decomposes a function (often a function of time, or a signal) into its constituent frequencies. Pure data offers the Fast Fourier Transform (FFT) which is used in order to extract from the incoming signal the dominant frequency in each of the predefined frequency bands.

4. Explanation

4.1 ADSR Envelope

A. Attack

The algorithm used to extract the attack from a wav file is the following:
A timer starts counting in milliseconds, starting at the same time with wav file.
Each time we find a sample max amplitude a bang is sent to the timer which outputs the current time and the decay operation restarts. When we reach the end of the wav file the atack time will be that of the sample with max amplitude.

B. Decay

The algorithm used to extract the decay from a wav file is the following:
A timer starts counting in milliseconds, starting/restarting every time a max amplitude value is found from the above attack operation. The timer stops when the current amplitude stops falling . Every time this happens the sustain operation is triggered.

C. Sustain

The algorithm used to extract the sustain from a wav file is the following:
A timer starts counting in milliseconds, every time the above decay operation stops (the amplitude value stops falling). The timer stops when the current amplitude stops being stable (**with the term stable we mean that the amplitude value is not moving above/below more than a specified constant**). This **constant** is determined by **analyzing the mean value of relative increase and decrease between current amplitude value and next amplitude value for the whole .wav file** (this happens while the .wav file is being recorded, before we start analyzing audio for attack, decay, sustain, release). The above is needed, because we must have a better understanding of what means ‘increase’ and ‘decrease’ in each sound file, before we process it. (Having the same constant for every wav file is wrong, because different .wav files have different increase/decrease values→ **Increase/decrease is relative**). **This constant is used for all the above operations (attack→ increase, decay→ decrease, sustain→ stable).**

D. Release

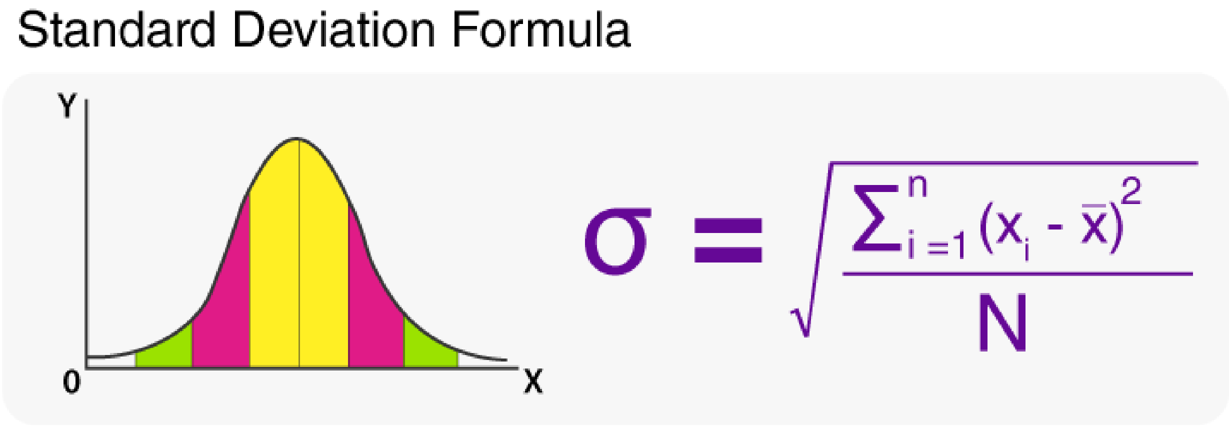
The formula used to extract the sustain from a wav file is the following:

Release time = total_wav_time – (attack_time + decay_time + sustain_time)

Release time is the remaining time when all the above operations are done.

4.2 Db Threshold

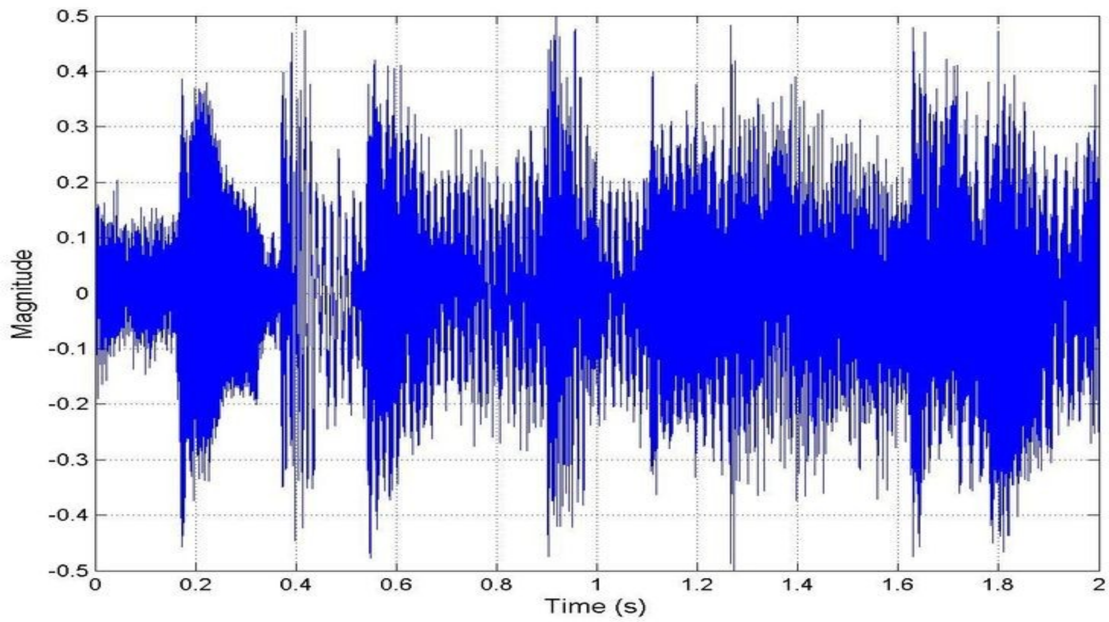
The db threshold is configured by the following way:
The predefined db level is given by the user using a "pure data vslider" object.
After that the threshold is configured by measuring the standard deviation of current amplitude . If the mean amplitude minus (-) standard deviation smaller than (<) the predefined db level, this predefined db threshold is used.



4.3 Frequency Analysis

5. Conclusion

Although we capture envelope characteristics (A.D.S.R.) for every signal, in reality there are signals that are incompatible with the definition.



If we observe the above signal, we can't obtain the envelope with absolute certainty. What's worth mentioning is that if we cropped it in smaller samples, it would be much easier to specify attack, decay, sustain, release. In other terms, we could say that **adsr** envelope depends on the defined threshold.

6. Sources

https://puredata.info/
https://theproaudiofiles.com/synthesis-101-envelope-parameters-uses/
https://en.wikipedia.org/wiki/Envelope_(music)
https://www.investopedia.com/terms/s/standarddeviation.asp
https://en.wikipedia.org/wiki/Shell_script