

EFEKTY GITAROWE

Autor: Karol Witusik
Akademia Górniczo-Hutnicza

Spis treści

1. WSTĘP	4
3. FUNKcjONALNOŚĆ (FUNCTIONALITY)	6
4. ANALIZA PROBLEMU (PROBLEM ANALYSIS).....	7
4.1 DELAY	7
4.2 DISTORTION	8
4.3 FILTR DOLNOPRZEPUSTOWY I GÓRNOPRZEPUSTOWY	9
5. PROJEKT TECHNICZNY (TECHNICAL DESIGN)	10
5.1 PARĘ UWAG WSTĘPNYCH	10
5.2 PLIKI NAGŁÓWKOWE	11
5.3 DIAGRAM KLAS.....	13
6. OPIS REALIZACJI (IMPLEMENTATION REPORT)	14
7. PODRĘCZNIK UŻYTKOWNIKA (USER'S MANUAL)	15
7.1 USTAWIENIA EFEKTÓW	16
7.2 KOLEJNOŚĆ EFEKTÓW	17
7.3 ZARZĄDZANIE PLIKIEM ORAZ WYŁĄCZANIE PROGRAMU	17

Lista oznaczeń

DSP	Digital Signal Processing
FIR	Finite Impulse Response
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory

1. Wstęp

Dokument dotyczy programu umożliwiającego wgrywania własnych ścieżek dźwiękowych w formacie wav i nakładanie na nich wybranych efektów gitarowych w dowolnej kolejności.

2. Wymagania systemowe (*requirements*)

Aby uruchomić program z pliku o rozszerzeniu exe wystarczy komputer z systemem operacyjnym Windows (32 lub 64 bitowy).

Aby uruchomić program w środowisku programistycznym Qt Creator należy:

1. Zainstalować środowisko Qt Creator (przykładowy tutorial:
<https://www.youtube.com/watch?v=eS7ank-qFjg>)
2. Dodać biblioteki SFML do środowiska (przykładowy tutorial:
<https://www.youtube.com/watch?v=HO1yhzTVzDo&>)

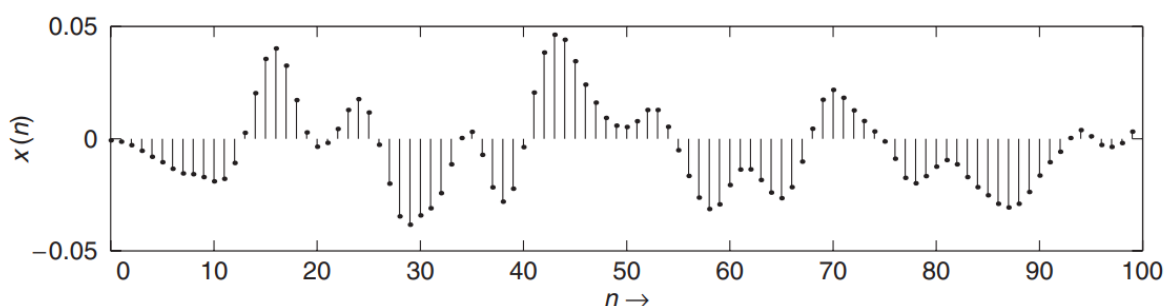
3. Funkcjonalność (*functionality*)

- Wgrywanie pliku dźwiękowego (format wav)
- Nakładanie na ścieżkę dźwiękową efektu Delay
- Nakładanie na ścieżkę dźwiękową efektu Distortion
- Nakładanie na ścieżkę dźwiękową filtra dolnoprzepustowego
- Nakładanie na ścieżkę dźwiękową filtra górnoprzepustowego
- Odtwarzanie przetworzonej oraz oryginalnej ścieżki
- Zapis edytowanej ścieżki do nowego pliku

4. Analiza problemu (*problem analysis*)

Pliki w formacie wav, prócz informacji takich jak bitrate czy ilość kanałów, przechowują amplitudę sygnału w konkretnym czasie. Oznacza to, że pliki tego typu zawierają spróbkowany sygnał audio. Dzięki temu stosując DSP w łatwy sposób można uzyskać różne efekty dźwiękowe.

Wizualizacji przykładowego sygnału spróbkowanego:



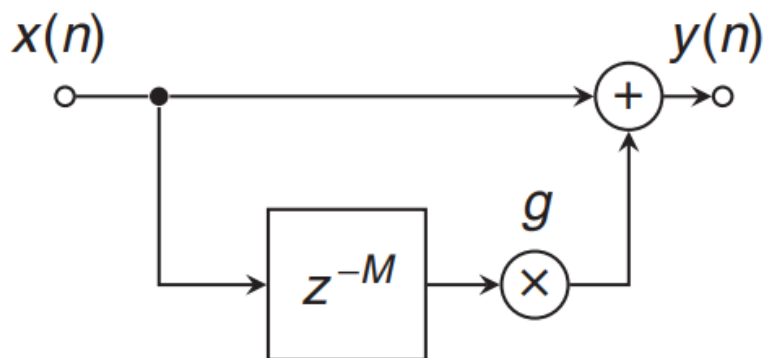
Rys.1. Z książki DAFX: Digital Audio Effects 2 wydanie

Program, którego dotyczy ta dokumentacja, ma zaimplementowane cztery efekty dźwiękowe. Poniżej znajduje się krótki opis każdego z nich.

4.1 Delay

Pierwszym efektem jest delay. Imituje on echo, tym samym dodając wrażenie przestrzenności.

Aby go uzyskać, należy opóźnić ścieżkę dźwiękową o zadany czas względem oryginału, a następnie przemnożyć przez dany współczynnik, by zmniejszyć głośność "echa". Kolejnym krokiem jest dodanie nowo utworzonego sygnału do pierwotnej ścieżki. Czynność tą wykonuje się zadaną ilość razy.



Rys.2. Przykładowy sposób, w jakim można uzyskać efekt Delay (z jednokrotnym powtórzeniem). Parametr M wpływa na opóźnienie, a g na amplitudę. Źródło: DAFX: Digital Audio Effects 2 wydanie

4.2 Distortion

Dzięki temu efektowi można uzyskać specyficzny metaliczny dźwięk gitar.

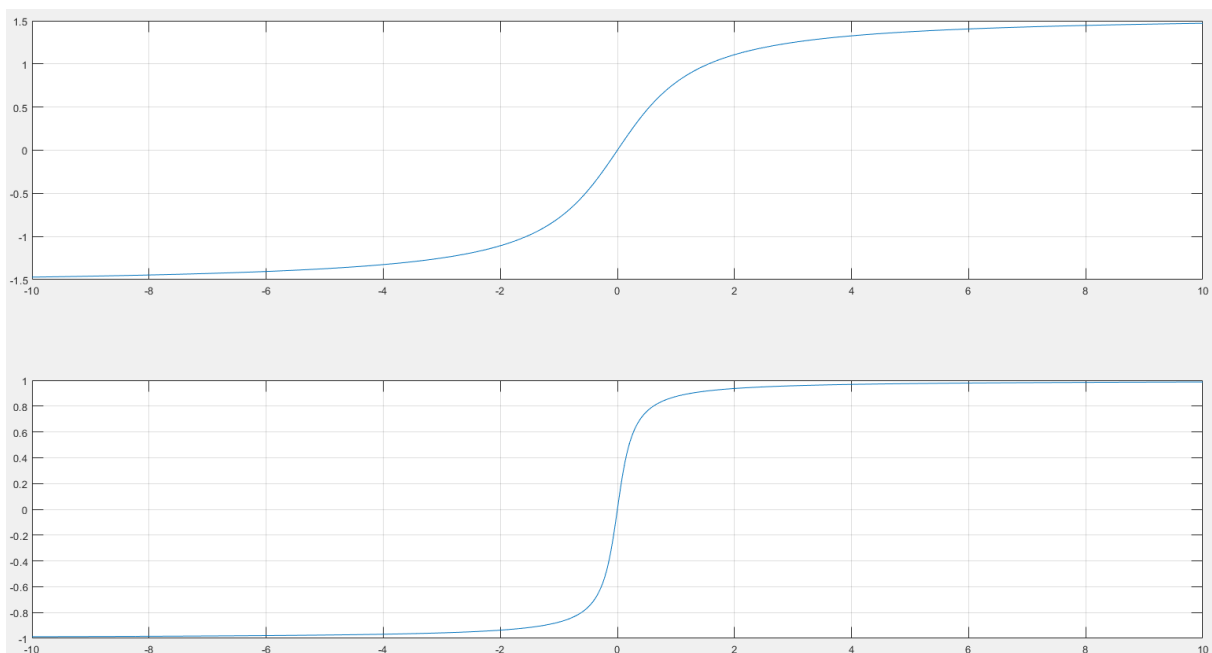
Jego najprostsza realizacja wygląda w ten sposób, że obcina się amplitudę sygnału do określonej wartości. W programie zamiast prostego przycinania wykorzystuje się funkcję arcus tangens, dzięki czemu przycięcia są wygładzone.

Ogólny wzór efektu wygląda następująco:

$$f(x) = c_1 * \operatorname{atan}(c_2 * x)$$

c_1 – kontroluje amplitudę by ta nie wyszła poza zakres

c_2 – wpływa na ostrość obcięć

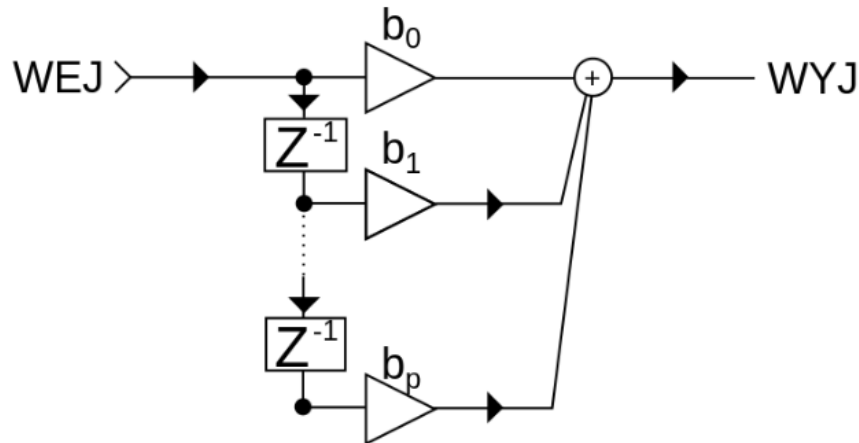


Rys.2. Pierwszy wykres przedstawia funkcję arcus tangens ze współczynnikami $c_1 = 1$, $c_2 = 1$. Na drugim wykresie funkcja ma współczynniki równe $c_1 = 2/\pi$, $c_2 = 5$.

4.3 Filtr dolnoprzepustowy i górnoprzepustowy

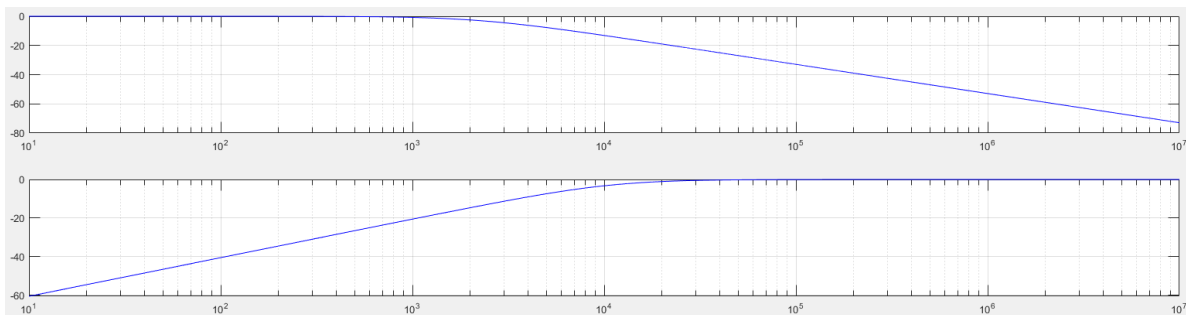
Ostatnimi dwoma efektami są filtry. Pozwalają one obciąć zbędne częstotliwości sygnału, tym samym zmieniając brzmienie ścieżki.

Aby zrealizować te filtry, zostały zastosowane filtry FIR.



Rys.3. Implementacji filtru FIR. b_i to współczynniki filtru, a Z^{-1} to opóźnienia o jedną próbkę.

Źródło: [https://pl.wikipedia.org/wiki/Filtr_o_skończonej_odpowiedzi_impulsowej](https://pl.wikipedia.org/wiki/Filtr_o_sko%C5%84czonej_odpowiedzi_impulsowej)



Rys.4. Przykładowe charakterystyki filtrów górno oraz dolnoprzepustowych.

5. Projekt techniczny (*technical design*)

5.1 Parę Uwag Wstępnych

Dane autora: Karol Witusik, karol.witusik@gmail.com

Data rozpoczęcia prac nad programem: 20-12-2019 (kolejne modyfikacje są dostępne pod adresem: <https://github.com/ferre111/Guitar-effects>)

Prawa autorskie klasy wykorzystanej do generowania filtrów FIR:

FIR filter class, by Mike Perkins

a simple C++ class for linear phase FIR filtering

For background, see the post <http://www.cardinalpeak.com/blog?p=1841>

Copyright (c) 2013, Cardinal Peak, LLC. <http://www.cardinalpeak.com>

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- 1) Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.*
- 2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.*
- 3) Neither the name of Cardinal Peak nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.*

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL CARDINAL PEAK, LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

5.2 Pliki nagłówkowe

Pliki nagłówkowe powiązane z konkretnymi klasami.

```
class Guitar_effects {
private:
    //buffer object for samples from track file
    sf::SoundBuffer buffer;
    //sound object for playing track from buffer object
    sf::Sound sound;
    //buffer pointer
    const sf::Int16* samples_p;
    //vector for new samples values
    vector<sf::Int16> samples_vec;
    //size of buffer
    size_t count;

public:

    Guitar_effects();
    ~Guitar_effects();

    //loading samples from file to buffor
    //file - absolute path to the file
    //return true if method completed correctly
    bool load_from_file(string file);

    //loading samples from vector
    //return true if method completed correctly
    bool load_buffer_from_sampels();

    //loading samples from vector who have samples that was process by delay effect method
    //d - delay
    //return true if method completed correctly
    bool load_buffer_from_sampels(int d);

    //saving samples from buffet to file
    //file - absolute path to the file
    //return true if method completed correctly
    bool save_buffer_to_file(string file);

    //before call method play we must set buffer
    void set_buffer(void);

    //return samplerate of track
    unsigned int get_sample_rate(void);

    //playing track
    void play(void);

    //effects methods
    void delay_effect(int delay, double factor, double volume);
    void distortion_effect(double blend, double volume, double range);
    void filter_LPF_effect(double freq);
    void filter_HPF_effect(double freq);
};
```

```
class MainWindow : public QMainWindow
{
    Q_OBJECT

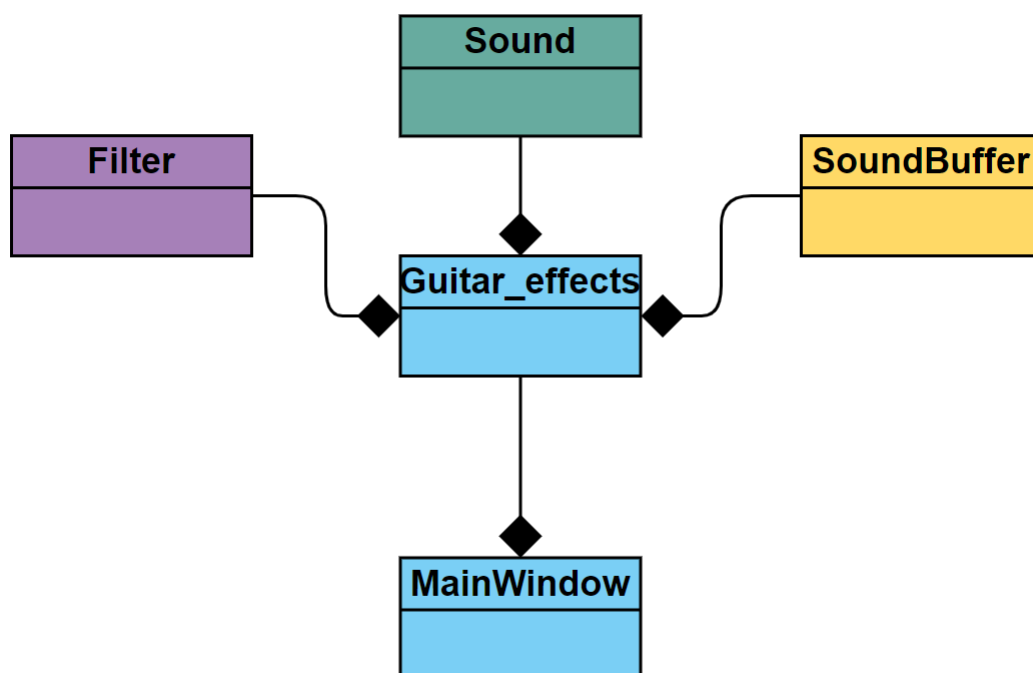
public:
    MainWindow(QWidget *parent = nullptr);
    ~MainWindow();

private slots:
    //methods call when user click on particular button
    void on_pushButton_Open_clicked();
    void on_pushButton_Play_clicked();
    void on_pushButton_Save_clicked();
    void on_pushButton_Play_Original_Track_clicked();

private:
    Ui::MainWindow *ui;
    Guitar_effects *guitar_effects;
    std::string file_name;

    //methods whose get value from knobs and call particular effect method
    void delay(void);
    void distortion(void);
    void filter_LPF(void);
    void filter_HPF(void);
};
```

5.3 Diagram klas



Rys.5.

6. Opis realizacji (*implementation report*)

Komputer, na którym było testowane oprogramowanie:

CPU – Intel Core i7 6700HQ

RAM – 16GB, DDR4

GPU – NVIDIA GeForce GTX 960M

System operacyjny – Microsoft Windows 10 Home

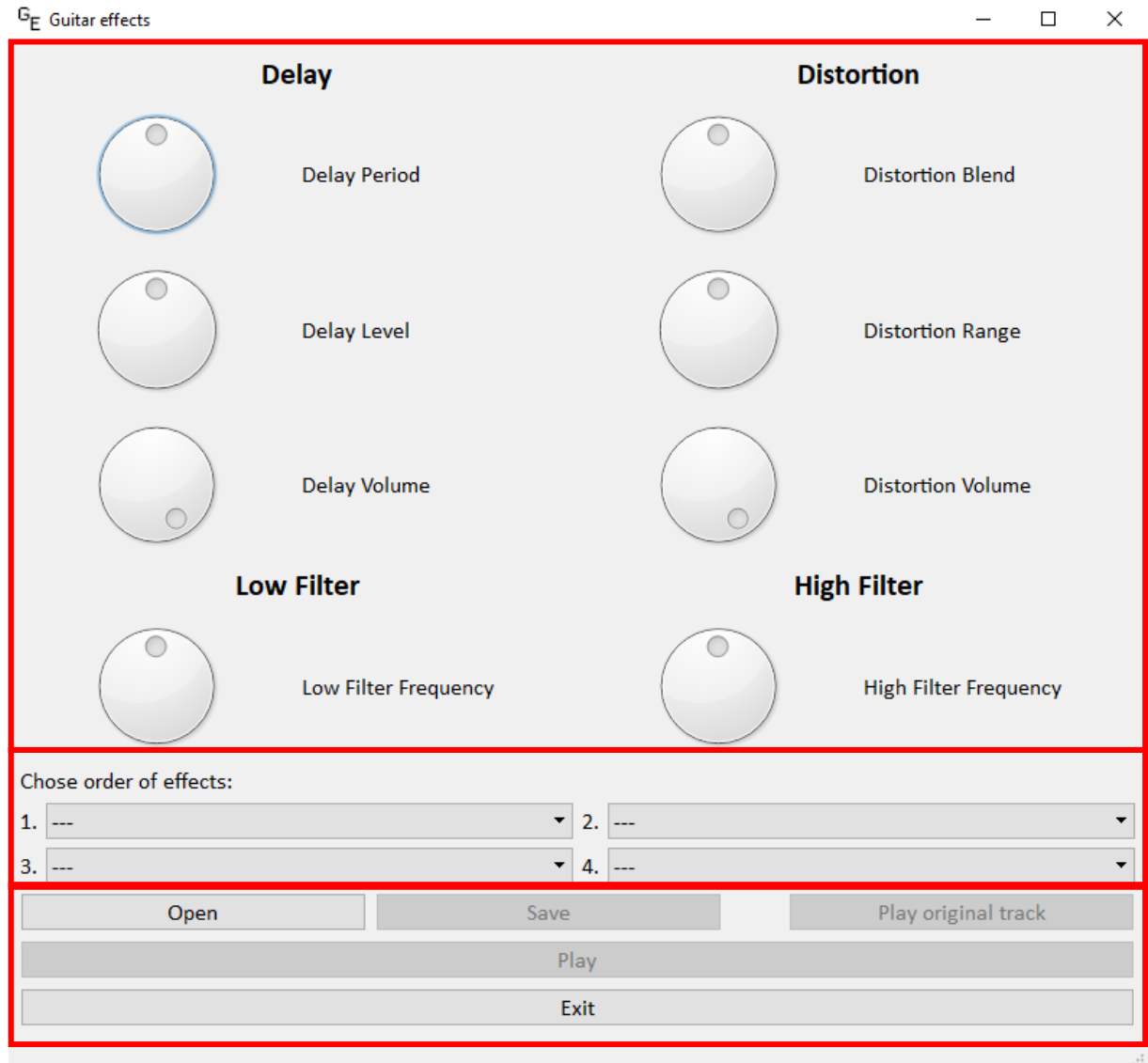
Środowisko:

Qt Creator Based on Qt 5.14.0 (MSVC 2017, 32 bit)

Kompilator:

Windows gcc 7.3.0 mingw 32-bit

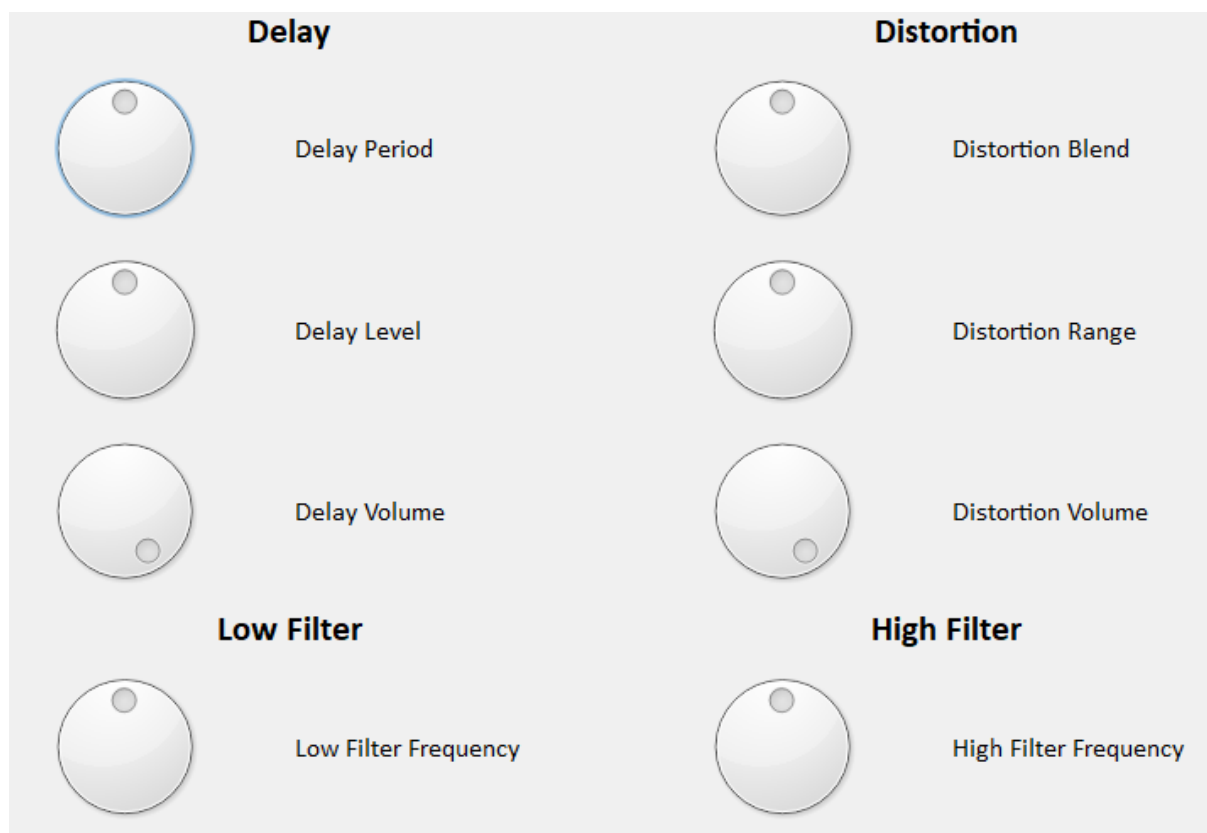
7. Podręcznik użytkownika (*user's manual*)



Rys.6.

Całe GUI programu można podzielić na trzy sekcje wyznaczone powyżej (Rys.6.). Każdy czerwony prostokąt odpowiada pojedynczej sekcji.

7.1 Ustawienia efektów



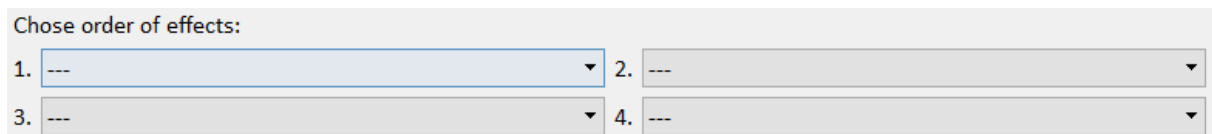
Rys.7.

W tej sekcji użytkownik może wybrać wartości poszczególnych współczynników.

- Delay
 1. Delay Period – określa opóźnienie, po którym będą dodawane kolejne kopie oryginalnego sygnału.
 2. Delay Level – określa, jak szybko echo będzie wyciszane tj. współczynniki, przez które będą przemnażane kolejne kopie sygnału.
 3. Delay Volume – zmienia głośność wyjściowego sygnału (zapobiega przesterom).
- Distortion
 1. Distortion Blend – określa proporcje między oryginalnym a przetworzonym sygnałem.
 2. Distortion Range – określa ostrość przycinania.
 3. Distortion Volume – zmienia głośność wyjściowego sygnału (zapobiega przesterom).

- Low Filter
 1. Low Filter Frequency – określa częstotliwość graniczną filtra.
- High Filter
 1. High Filter Frequency – określa częstotliwość graniczną filtra.

7.2 Kolejność efektów



Rys.8.

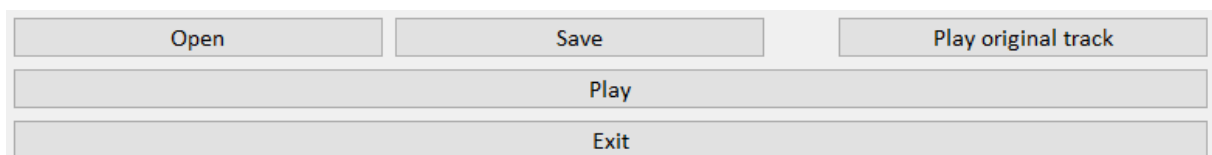
W tej części GUI użytkownik może ustalić których efektów chce użyć oraz jaka ma być ich kolejność.



Rys.9.

Po kliknięciu któregoś z pól wyboru użytkownik może wybrać czy i jaki efekt ma się wykonać. Aby żaden z efektów się nie wykonał należy wybrać opcję „---”.

7.3 Zarządzanie plikiem oraz wyłączanie programu



Rys.9.

W ostatniej sekcji znajduje się pięć przycisków:

- Open – włącza eksplorator plików i pozwala na załadowanie pliku dźwiękowego (wav).
- Save – włącza eksplorator plików i pozwala na zapisanie edytowanego pliku dźwiękowego .
- Play original track – odtwarza oryginalną ścieżkę dźwiękową.
- Play – odtwarza edytowaną przez efekty ścieżkę dźwiękową.
- Exit – wyłącza program.

Bibliografia

- [1] „DAFX - Digital Audio Effects” Udo Zolzer drugie wydanie
- [2] The C++ Programming Language Book Bjarne Stroustrup
- [3] <https://www.cardinalpeak.com/a-c-class-to-implement-low-pass-high-pass-and-band-pass-filters/>
- [4] <https://www.sfml-dev.org/tutorials/2.5/audio-sounds.php>Press W.H., Teukolsky S.A.,
- [5] <https://www.youtube.com/watch?v=EkjaiDsiMQ&list=PLS1QulWo1RIZiBcTr5urECberTITj7giA>
- [6] <https://www.youtube.com/watch?v=iNCR5fISuDs>
- [7] <https://kidpatel.wixsite.com/dspaudioeffects/distortion>