

MFGP audió szoftver 2.0

Zabos Péter Roland
Debreceni Egyetem, Informatikai
Kar
Debrecen, Magyarország
zabosrolcsi@gmail.com

Erdei Timotei István
Mechatronikai Tanszék
Debreceni Egyetem, Műszaki Kar
Debrecen, Magyarország
timoteierdei@eng.unideb.hu

Absztrakt— Mára már teljesen elengedhetlenné vált bármely számítógépes szoftver használata LP-k, (long play) de egyetlen zeneszám elkészítésekor is. Az is fontos része lehet az alkotói munkának, hogy szoftverekkel már az effekt pedálok is helyettesíthetők. Az MFGP (Mostly for Guitar Players) alkalmazás a számítógép által nyújtott ezen előnyök ötvözésére készült. Egyaránt használható zeneszerkesztőként, zene lejátszóként továbbá különböző effekteket és hangolót is nyújt a feltörekvő gitárosoknak, akik saját maguk szeretnék hangzásukat kialakítani és felvételeiket kedvükre szerkeszteni. A ritmusérzék fejlesztése egy zenész életében rendkívül fontos, ezért a szoftver dob gép funkciót is tartalmaz, mely remek alapütemet szolgáltathat egy új zenei szerzeményhez. Emellett a modellezett looper pedál is egy jól bevált gyakorlati lehetőséget nyújt, vagy többsávú kíséret létrehozására is alkalmas lehet. Ez a dokumentáció az audio programozás által nyújtott azon eszközöket és módszereket fogja részletesebben taglálni, melyek a szoftver elkészítését lehetővé tették.

Kulcsszavak—C#; gitár; effekt; zeneszerkesztő; looper pedál; waveform

I. BEVEZETŐ

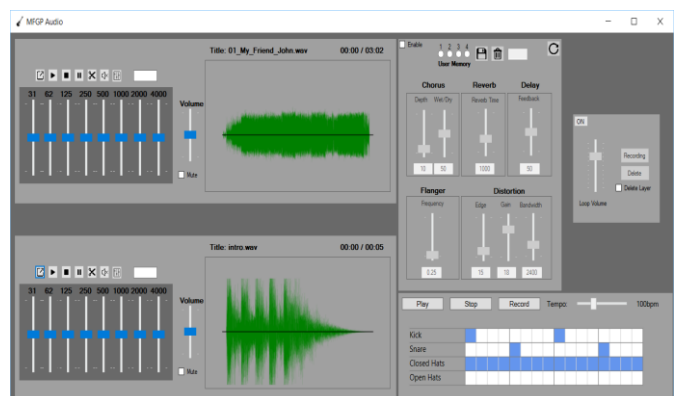
A zene és a számítógép első kapcsolata az 1950-es évekre tehető. Számítógépes zene felvételére már 1951-ben képes volt a Ferranti Mark 1 nevű számítógép, igaz ez szóbanyi mérete miatt még nem volt igazán praktikus. Ezen felvételeken olyan populáris dallamok is megtalálhatók voltak, mint a „God Save the Queen”, vagy Glenn Miller örökzöldje az „In the Mood” [1]. Néhány évtizeddel később, 1983-ban megszületett a MIDI (Musical Instrument Digital Interface) szabvány. A MIDI lehetővé tette a számítógép és más hardverek kommunikációját. Ezek után óriási ugrást jelentett a Sound Blaster Pro [2] hangkártya, amely már CD minőségben (44.1 kHz sztereó) is képes volt felvenni, így lehetőség nyílt arra, hogy otthoni számítógéppel is kiváló minőségben lehessen hangot rögzíteni. 1997-re már egyszerre több sávot is kezelni lehetett és megjelentek a plug-in lehetőségek is. A zenei szoftverfejlesztők még ennél is tovább feszegették a határokat, így mára már lehetővé vált egyetlen hang módosítása is [3]. A kutatás/fejlesztésnek a Debreceni Egyetem adott otthont [14].

II. PROGRAMOZÁSI ESZKÖZÖK KIVÁLASZTÁSA

A legfőbb szempont olyan programozási nyelvek kiválasztása volt, amely képes megfelelő minőségű hang felvételére és lejátszására. Erre tökéletes volt a C#, amely ugyan nem tartalmaz komolyabb beépített lehetőségeket, de az NAudio [4] nyílt forráskódú könyvtár segítségével ez könnyedén kiküszöbölhető. Különböző API-k (Application Programming Interface) használatával éri el a felhasználónak legmegfelelőbb minőséget és teljesítményt. Az NAudio eszközei felelősek a dob gépért. Emellett egy másik fontos technikai része is van a szoftvernek, mégpedig a C#Core .Net könyvtár [5]. A C#Core segítségével jöttek létre a real-time effektek, felvevő funkciók, lejátszók. A szoftver tartalmaz még egy fontos funkciót, ez pedig nem más, mint a hangolót. Ennek megvalósításáért a Java beépített javax.sound.sampled [6] csomagja felelős. A beépített eszközök megfelelőek voltak a beérkezett hangok felvételére és elemzésére.

III. KINÉZET ÉS FŐ EGYSÉGEK

A kinézet alapvetően a Visual Studio [7] Windows Form által nyújtott lehetőségeken alapul.



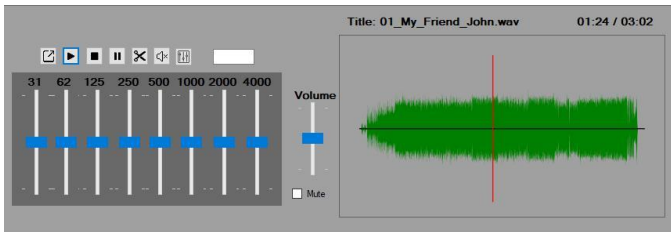
1. ábra: A szoftver kezdőképe

A szoftver tartalmaz egy fő menüt, mely több egységből tevődik össze. Ezek az egységek: a lejátszók, az effektek, a dob gép és looper szekció. Az egységek valamennyi eleme egy-egy User Controlon van elhelyezve. Kivételt képez

azonban a dobgép UI megvalósítása, mivel ez a .NET keretrendszer XAML jelölőnyelvére épül [8].

IV. LEJÁTSZÓK

A lejátszók egység valamennyi eleme tartalmaz megnyitás/tallózás lehetőséget .wav formátumban, továbbá az alapvető lejátszás, megállítás, leállítás, némítás, hangerő beállító funkciókat.



2. ábra: Lejátszó

A megnyitott hangfájlok EQ beállítások is végezhetőek. Az ekvlizeres szoftveres, vagy hardveres filterek, amelyek képesek növelni vagy csökkenteni a hangerejét bizonyos frekvenciáknak. Egy hangforrás különböző formájú terekben különbözően hangozhat. Olyan helyszínek esetén, amelyek jobban reagálnak a basszus frekvenciákra, a low end csökkenthető a gerjedés, illetve a rosszabb hangzás elkerülése érdekében. Az ekvalizerek távokban (band) operálnak. Egy személygépkösi két band-os EQ-ja esetén, az alacsony, illetve a magas frekvenciák állíthatók (treble, bass) [9]. Az MFGP alkalmazás egy 8 band EQ panellel rendelkezik. A band-ek száma egyenesen arányos a változatosabb beállítások elérhetőségével, mivel minél több frekvenciát lehet hangosítani vagy halkítani, annál behatóbban lehet a hangzást kontrollálni. Az alkalmazás mindegyik EQ potméteréhez hozzá van rendelve a trackBar_ValueChanged eljárás, amely a változtatott potméterhez tartozó frekvencia hangerejét változtatja.

```
private void trackBar_ValueChanged(object sender, EventArgs e)
{
    var trackbar = sender as TrackBar;
    if (_equalizer != null && trackbar != null)
    {
        double perc = (trackbar.Value / (double)trackbar.Maximum);
        var value = (float)(perc * MaxDB);
        int filterIndex = Int32.Parse((string)trackbar.Tag);
        EqualizerFilter filter = _equalizer.SampleFilters[filterIndex];
        filter.AverageGainDB = value;
    }
}
```

3. ábra: EQ potméterek

Hangfájl megnyitása után, a fájl hullámformája rajzolódik ki, amely egy interaktív panelen helyezkedik el. Egy bal kattintással pozícionálható a számon belül. Bal klikkel és „drag and drop” módszerrel egy nagyobb részletet kijelölve,

az ollógombbal kivágható a részlet. A kivágás nem a tényleges hangfájl módosítását jelenti, hanem egy új fájl létrehozását a mellőzni kívánt rész nélkül. A rajzterületen kijelölt rész két oldala és a terület maximális méretének arányaiból adódik, hogy mely részletet kell kihagyni a hangfájlból. Azaz a kijelölt terület bal oldala úgy aránylik a maximális rajzterület hosszához, mint a hangfájlból való kihagyás kezdete a hangfájl hosszához. Ennek megfelelően a jobb oldal, a kihagyás vége. A GetRawElements beépített függvény segítségével, kiderül, hogy mennyi raw byte data van a hangfájl kezdete és a kihagyás kezdőpontja között. Ezt a mennyiségű adatot kell beolvasni először, majd a kihagyás vége időpontjától a hangfájl végig minden adatot.

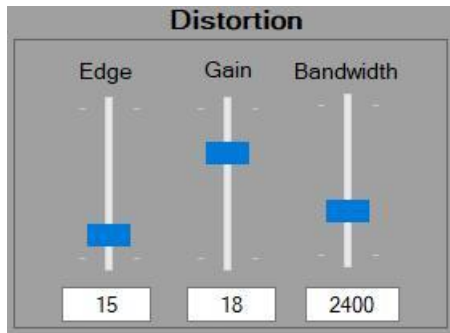
```
private static void AddTimeSpan
(IWaveSource source, WaveWriter writer, TimeSpan startTimeSpan, TimeSpan endTimeSpan)
{
    int read = 0;
    long bytesToEncode = source.GetRawElements(startTimeSpan);
    var buffer = new byte[source.WaveFormat.BytesPerSecond];
    while ((read = source.Read(buffer, 0, buffer.Length)) > 0)
    {
        int bytesToWrite = (int)Math.Min(read, bytesToEncode);
        writer.Write(buffer, 0, bytesToWrite);
        bytesToEncode -= bytesToWrite;
    }
    source.SetPosition(endTimeSpan);
    read = 0;
    while ((read = source.Read(buffer, 0, buffer.Length)) > 0)
    {
        writer.Write(buffer, 0, read);
    }
    writer.Dispose();
    writer = null;
}
```

4. ábra: Kivágás eljárás

A középső egérgombbal némítás helyezhető el, amelynek mértéke tetszőleges és másodpercben adható meg, a mute gombbal véglegesíthető. Hasonlóan a kivágásnál, itt is az arányokat használja fel az alkalmazás. Kattintás után egy piros csík jelzi ahova a némítás lesz beszúrva. A piros csík pozíciója és a rajzterület maximális hossza megadja a hangfájlon belüli időpontot. Az új hangfájl elkészítésének menete is hasonló a kivágásnál alkalmazotthoz. A GetRawElements függvénnyel megkapjuk a szükséges beolvasnivaló adatmennyiséget. Ezután egy csendet tartalmazó .wav fájlból (5minsilence.wav) annyi másodpercnnyi adat kerül az új fájlba, amennyit a felhasználó megadott a kijelölt textBoxban. Végezetül, az eredeti fájl további része kerül írásra az új fájlba.

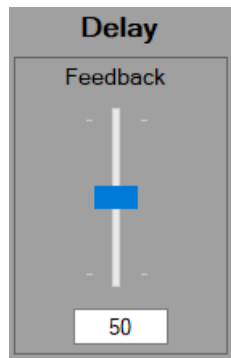
V. EFFEKTEK

A következő egység az effektek. Dave Davies nevéhez fűződik az első torzított hangzás, amit úgy tudott elérni, hogy borotvájával megvagdosta erősítőjének hangszóróját. Így szólhatott 1964-ben a „You Really Got Me” határozottabbnak, daraszerűnek [10]. Mára már bármelyik hangkártya képes ilyen hangzásra, a megfelelő programozási eszközök segítségével. Torzításra természetesen az MFGP alkalmazás is képes, ezzel együtt összesen öt darab effekt található rajta: torzító visszhang, chorus, flanger, zengető.



5. ábra: Torzító panel

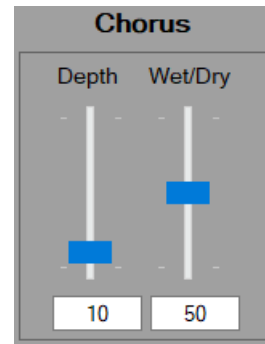
A torzító főleg rock/métál műfajokban elterjedt erőteljesebb hangzásért felelősek. Ha a bemeneti jel átlépi az adott eszköz (erősítő) határát, a jel összenyomódik (compress) és létrejön a torzított hangzás. A túlzérelt jel csúcsai levágásra (clipping) kerülnek. Többféle esete van a vágásnak. Hard clipping, amikor a végeredmény négyszög jel, ez főleg a heavy metálban használatos, mivel a torzítás meglehetősen nagy mértékű. A soft clipping kisebb mértékű torzítást eredményez (dirty), a rock és blues műfajban elterjedt. Az alkalmazás torzítója három beállító potméterrel rendelkezik. Az edge a torzítás intenzitását állítja 0 és 100 között, alapértelmezetten 15. A gain azt szabályozza, hogy a bemeneti jel mennyire változzon meg az effekt alkalmazásával, -60 dB és 0 dB között, alapértelmezetten -18 dB.



6. ábra: Visszhang panel

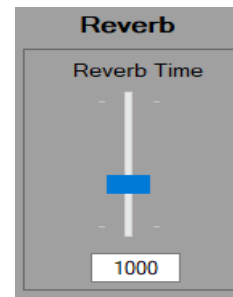
A visszhang (delay/echo) effekt a nevében hordozza hatását, amikor a hang visszapattan a felszínről. Régebben ezt szalagos magnókkal oldották meg, majd analóg módszerekkel, de manapság digitálisan érik el a hangzást. Lényegében a lejátszott hang kerül ismétlésre. Egyetlen potmétere a feedback szabályozza a lejátszott hang ismétléseinek számát, 0 és 100 között, alapértelmezetten 50.

A chorus effektel azt a hatást érjük el, mintha egyszerre több hangszer szólalna meg. Az eredeti hang megduplázása után, a két hang közzé késleltetést helyez és a másolt hangmagasságát megemeli. Mintha egy kórus énekelné ugyanazt a hangot, valójában innen kapta a nevét. Az alkalmazás chorus panelje két potméterrel rendelkezik.



7. ábra: Chorus panel

A wet/dry a feldolgozott, illetve feldolgozatlan jelek arányát befolyásolja, 0 és 100 között, alapértelmezetten 50. A depth határozza meg, hogy milyen mértékben modulálja az alacsony-frekvencia oszcillátor a késleltetett másolt hangot, 0 és 100 között, alapértelmezetten 10.



8. ábra: Reverb panel

A zengető azért a rezonáns hangért felelős, amelyet nagyobb teremben figyelhetünk meg, például tornateremben, templomban. A korai felvételeknél ezt egy nagy teremben lévő mikrofonnal és hangszórával oldották meg. A mikrofon a terem egyik végén rögzítette a terem másik végében lévő hangszórá hangját. Gitárerősítőkben rugópárokkal van megvalósítva. Az alkalmazás Reverb Time potmétere beállítja, hogy mennyi ideig tartson a zengés, 0.001 és 3000 között miliszekundumban, alapértelmezetten 1000.

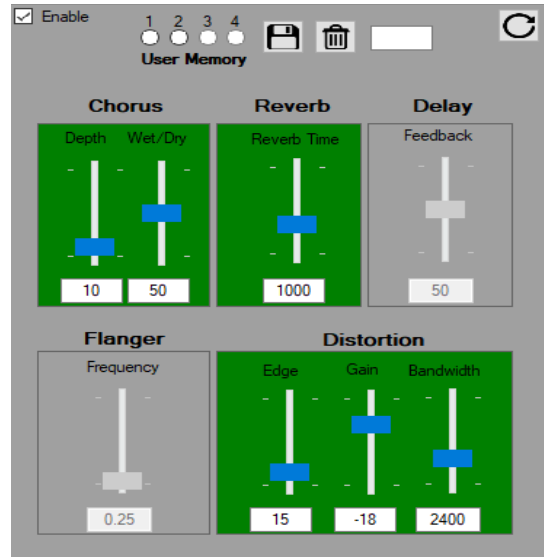
Minden effekt paneljének van egy engedélyezésére szolgáló eljárása. A panel_Click eljárás a chorus effekt be és kikapcsolásáért felelős. Bekapcsoláskor a „sorrend” String-eket tároló adatszerkezetbe kerül a „chorus” szó, ezzel párhuzamosan a „lista” IWaveSource interfészt használó objektumokat tároló adatszerkezetbe egy új DmoChorusEffect példány kerül. Kikapcsoláskor a „chorus” szót eltávolítjuk a „sorrend” adatszerkezetből és a DmoChorusEffect objektumát a „lista” szerkezetből. Be és kikapcsoláskor, a „lista” tartalma törlődik, majd a „sorrend” szerint épül újra az effekt chain, így tartva mindig a felhasználó által diktált sorrendet. Egy effekt a láncba való bekerülésekor a „lista” utolsó elemét használja paramétereként, azaz ezen tag által tárolt raw byte data-val fog a következőkben dolgozni. A lánc első tagja mindig a „source” objektum, ami tisztán a gitár bemenetét tartalmazza, mindenféle változtatás nélkül. Az első effekt a „source” által

tárolt adatot fogja módosítani, a második az elsőét és így tovább. A „Play” függvény feladata, hogy paraméterét egy WasapiOut objektumnak szolgáltatssa. A WasapiOut a Windows API-t használja hangfájlok lejátszására. Minimum támogatott operációsrendszer: Windows Vista [11].

```
private void panel1_MouseClick(object sender, MouseEventArgs e)
{
    if(panel1.BackColor==Color.Green)
    {
        depthText.Text= DmoChorusEffect.DepthDefault.ToString();
        wetdryText.Text = DmoChorusEffect.WetDryMixDefault.ToString();
        Stop();
        sorrend.Remove(sorrend[index("chorus")]);
        soundIn = new WasapiCapture(false, AudioClientShareMode.Shared, 0);
        soundIn.Initialize();
        panel1.BackColor = SystemColors.ButtonShadow;
        soundIn.Start();
        effects = lista();
        depth.Enabled = false;
        wetdry.Enabled = false;
        depthText.Enabled = false;
        wetdryText.Enabled = false;
        Play(effects[effects.Count - 1]);
    }
    else
    {
        Stop();
        depth.Enabled = true;
        wetdry.Enabled = true;
        depthText.Enabled = true;
        wetdryText.Enabled = true;
        soundIn = new WasapiCapture(false, AudioClientShareMode.Shared, 0);
        soundIn.Initialize();
        soundIn.Start();
        panel1.BackColor = Color.Green;
        effects = lista();
        sorrend.Add("chorus");
        effects.Add(new DmoChorusEffect(effects[effects.Count - 1]));
        depthText.Text = DmoChorusEffect.DepthDefault.ToString();
        wetdryText.Text = DmoChorusEffect.WetDryMixDefault.ToString();
        Play(effects[effects.Count - 1]);
    }
}
```

9. ábra: Chorus panel ki/be

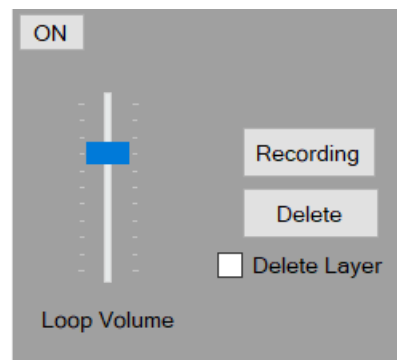
Az effekteket bal egérgombbal aktiválhatóak, ekkor zöldre változnak. Egyszerre több effekt alkalmazható, a megfelelő hangzás érdekében sorrendjük nagyon fontos, például nem érdemes torzító után zengetőt használni. A felhasználó, miután megtalálta a számára tökéletes hangzást, a sorrendet és a potméterek értékeit elmentheti a felhasználói memória egyikébe. Ezt a save gombbal teheti meg, a memória számát pedig a megadott 10. ábrán látható textBox-ban kell megadni. A rádiógombok csak akkor aktívak, ha van mentve beállítás a hozzájuk tartozó szöveges állományában. Ezeket az alkalmazás több helyen is ellenőrzi. Törölni a delete gombbal lehet, ez az adott memóriához tartozó szöveges állomány felülírását jelenti.



10. ábra: Effekt panel

VI. LOOPER

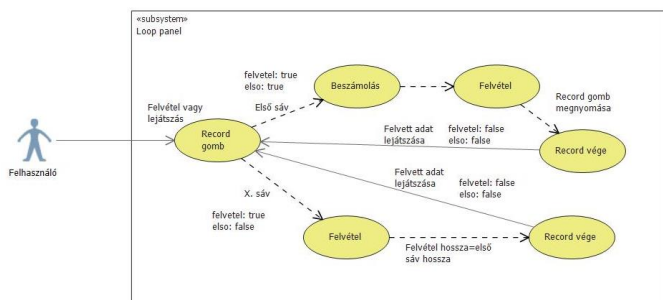
A 20. században a komoly zene mellett új zenei formák jelentek meg, amelyek inkább a tonalitásra összpontosítottak, azaz olyan dallamokra melyeknek hangjai egy alaphanghoz tartoznak és szoros kapcsolatban vannak vele [12]. Új előadók, más megközelítést alkalmaztak ritmus, ütem és ismétlések szempontjából. Az első loop magnetofon segítségével jött létre. Két eszköz segítségével a kazetta szalagját folyamatosan játszani lehetett. Majd a nyolcvanas években megjelentek a digitális megoldások a Roland által. Később piacra került az első dedikált looper eszköz [13]. A loop technika sokszínűségét az is bizonyítja, hogy manapság looping előadók is jelen vannak a zeneiparban, mint például Ed Sheeran, vagy Grace McLean. Az első loop pedálok a kétezres években jelentek meg, a cél az élő zene színesítése volt, zene élvezete, improvizáció képességének fejlesztése, gyakorlás, komponálás.



11. ábra: Looper panel

A panel lényege a looper pedál modellezése és a fent említett célok valamennyiének elérése. Fontos szerepet játszik a CSCore könyvtár LoopStream osztálya, amely implementálja az IWeaveSource interfész metódusait. Ezáltal saját Read metódussal is rendelkezik. Ezen függvény a

bemenetül kapott raw data bájtkokat addig olvassa, amíg nulla nem lesz a beolvasottak darabszáma, ekkor ugyanis a hangfájl végére ért. Az EnableLoop logikai adattag igaz/hamis értékétől függ, hogy ismétlődni fog-e a hangfájl. Recording gomb lenyomásával elindul az első réteg felvétele. A felvétel kezdését egy beszámolás jelzi. A count_in.wav hangfájl tartalmaz négy darab dob leütést, négy negyedés ütemmutatóval. Az alap sáv végét a felhasználónak kell jelezni a gomb ismételt megnyomásával. Ekkor a fájl eltárolódik és lejátszásra kerül. Miközben az első réteg lejátszása folyik a felhasználó új réteget vehet fel a Record gombbal. A felvétel nem egyből kezdődik, hanem a meglévő sáv vége után, hogy fel lehessen készülni és hossza az alapsáv hosszával fog megegyezni. A rétegek száma tetszőleges. Törlésre két lehetősége van az alkalmazás használójának. Az első egy teljes törlést jelent, amitől újra kezdődik a loop folyamat. A második csak az utolsó sáv törlését jelenti, ehhez a Delete Layer kapcsolót kell használni.



12. ábra: Looper használati eset

VII. ÖSZEGZÉS

A projekt során az MFGP szoftver továbbfejlesztésére került sor. A tervezett új funkciók bekerültek az alkalmazás eszköztárába. A zeneszerkesztő lehetőségeket bővítette a lejátszó egység megújulása. A hullámformák kirajzolásával vizuális eligazodásra is van lehetőség egy hangfájl esetében. A hullámforma panel interaktivitása rugalmasabbá tette a szoftvert és a hangfájlok módosítását (kivágás, némitás). A looper pedált modellező funkció a szoftver gyakorlásra, komponálásra alkalmas eszközök palettáját színesítette. A fejlesztés során az effektek egység is további fejlesztésen esett át. Ennek oka az volt, hogy a CScore könyvtár által nyújtott lehetőségek nem voltak teljesen kiaknázva az MFGP szoftver előző verziójában. Az effektek láncszerű alkalmazása még több lehetőséget nyújt a szoftver felhasználójának a saját hangzása kialakításában. Emellett felhasználhatósága is javult az egységnek azzal, hogy átláthatóbbá vált a panel, a felhasználó elmentheti a beállításait és kommentek lettek

elhelyezve a gombokon, amelyek az eligazodást segítik, ez utóbbi minden egység esetén alkalmazásra került.

VIII. KÖSZÖNETNYÍLVÁNÍTÁS

A publikáció elkészítését az EFOP-3.6.1-16-2016-00022 számú projekt támogatta. A projekt az Európai Unió támogatásával, az Európai Szociális Alap társfinanszírozásával valósult meg.

Köszönöm Erdei Timotei Istvánnak, a projekt elkészülése közben felmerült problémákkal kapcsolatos tanácsait. Továbbá köszönöm családomnak, akik a zenei tudásukkal javították a szoftver minőségét.

IX. HIVATKOZÁSOK

- [1] Georgi Dalakov, (2018.04.09.). [Online]. Available: <https://history-computer.com/ModernComputer/Electronic/Ferranti.html>
- [2] Sound Blaster, (2018.04.09.). [Online]. Available: <http://www.soundblaster.com/about/>
- [3] Computer Music Specials, (2018.04.09.). A brief history of computer music. [Online]. Available: <http://www.musicradar.com/news/tech/a-brief-history-of-computer-music-177299>
- [4] Mark Heath, (2018.04.10.). NAudio, [Online]. Available: <http://naudio.codeplex.com/documentation>
- [5] CScore, (2018.04.10.). [Online]. Available: <https://github.com/filoe/cscore>
- [6] Oracle, (2018.04.10.). Package javax.sound.sampled [Online]. Available: <https://docs.oracle.com/javase/7/docs/api/javax/sound/sample/package-summary.html>
- [7] Microsoft, (2018.04.10.). Visual Studio [Online]. Available: <https://www.visualstudio.com/>
- [8] Microsoft, (2018.04.10.). XAML Overview (WPF) [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/wpf/advanced/xaml-overview-wpf>
- [9] YatriTrivedi, (2016.09.28.). What Is an Equalizer, and How Does It Work? [Online]. Available: <https://www.howtogeek.com/59467/htg-explains-what-is-an-equalizer-and-how-does-it-work/>
- [10] Jeff Giles, (2014.12.01.). Dave Davies accuses brother Ray of lying about the Kinks „You really got me” guitar sound [Online]. Available: <http://ultimateclassicrock.com/kinks-you-really-got-me-guitar/>
- [11] Microsoft, (2017.12.10.). About WASAPI [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/desktop/dd371455\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd371455(v=vs.85).aspx)
- [12] Fábrián Zoltán, (2018.04.10). Torzító - distortion, overdrive, fuzz [Online]. Available: http://www.fzolee.hu/fw/distortion_overdrive_fuzz
- [13] Michael Peters, (1996.10.13). The Birth of Loop- A Short History of Looping Music [Online]. Available: <http://www.loopers-delight.com/history/Loophist.html>
- [14] T. I. Erdei, Zs. Molnár, N. C. Obinna, G. Husi, „Cyber physical systems in mechatronic research centre,” MATEC Web Conf. Volume 126, 2017.