



**Budapesti Műszaki és Gazdaságtudományi Egyetem**  
Villamosmérnöki és Informatikai Kar  
Automatizálási és Alkalmazott Informatikai Tanszék

Simon Endre András

# Innovatív mobil zenei oktatóalkalmazás Brain-Computer interfész használatával

---

*Konzulens: Dr. Forstner Bertalan*

*2013. október 22.*

## Kivonat

Kutatások bizonyították, hogy a gépi tanulás során használt visszacsatolásos modell a való életben is jól használható. Az interaktív mobil oktatóalkalmazás ezen elven kívánja a felhasználókat hangszeres zenére oktatni. Az alkalmazáshoz MIDI interfészen keresztül számos hangszer csatlakoztatható: fúvós, billentyűs és húros hangszerek egyaránt támogatottak. A visszacsatolás a játékos agyi aktivitását mérő külső eszköz segítségével valósul meg. A felhasználó különböző leckéket old meg, miközben a teljesítménye és agyi visszajelzése alapján az alkalmazás a feladatok nehézségét a felhasználóhoz igazítja és az eredményeket folyamatosan jutalmazza. A testreszabott tanulási folyamat és a sikerélmény hatására termelődő dopamin hozzájárul a hatékony tanuláshoz. A projekt célja, hogy a különböző intézményekkel együttműködve történő, párhuzamosan folyó kutatás számára alkalmazásokkal bizonyítsa az oktatás hatékonyságának növelését.

## Abstract

Based on several research activities, the well-known model of reinforcement learning is also successful in real-life learning situations. Our project is an innovative approach to use this fact in educating musical instruments. The application can be connected to a variety of MIDI instruments: wind-, keyboard- and string instruments. The feedback is ensured by an EEG device, which monitors the brain activity of the user. The user has to solve different lectures, and meanwhile, based on the measured brain activity and actual performance the application manipulates the difficulty of the current lecture. The customized experience and the sense of achievement during the game causes dopamine production in the brain, and learning becomes a balanced and fun activity. The goal of the project is to prove the assumptions of our related research about effective learning.

## Tartalomjegyzék

Kivonat.....	2
Abstract .....	3
1 Bevezetés.....	6
1.1 Motiváció .....	6
1.2 A dolgozat felépítése .....	7
2 Az elektronikus eszközökkel segített tanulás jellemzői.....	9
2.1 Az elektronikus tanulásról általában .....	9
2.2 Példák a módszer alkalmazására.....	10
2.2.1 Wu's castle.....	10
2.2.2 edX .....	11
2.2.3 Duolingo.....	12
2.3 A projekt során használt megközelítés .....	13
3 MIDI eszközök bemutatása .....	14
3.1 A MIDI rövid története.....	14
3.2 MIDI eszközök képességei.....	15
3.2.1 Channel Mode Messages .....	16
3.2.2 Channel Voice Messages .....	16
3.2.3 System Real Time Messages .....	17
3.3 Elérhető eszközök .....	17
4 Brain-Computer interfész használata Android alapú rendszereken.....	19
4.1 A Brain-Computer interfész bemutatása.....	19
4.2 A Brain-Computer interfészek története .....	19
4.3 A Brain-Computer interfész működése.....	20
4.4 Használatuk a tanulás segítéséhez .....	21
4.5 A projekt során használt BCI .....	21
5 Az alkalmazás tervezése és megvalósítása .....	22
5.1 Általános ismertető .....	22
5.1.1 Célok.....	22
5.2 Specifikáció .....	22
5.2.1 Általános áttekintés.....	22
5.2.2 Funkciók.....	22
5.2.3 Felhasználók.....	23

5.2.4	Elvárások .....	23
5.2.5	Szótár .....	24
5.3	Képernyőtervek .....	24
5.3.1	Főképernyő .....	24
5.3.2	Profil.....	25
5.3.3	Leckék felsorolása .....	25
5.3.4	Gyakorló képernyő.....	26
5.4	Architektúrális terv .....	28
5.5	Adatmodell tervezése .....	29
5.6	Az alkalmazás megvalósítása .....	30
5.6.1	Az alkalmazás struktúrája .....	30
5.6.2	A SoundManager komponens.....	31
5.6.3	A GameLib komponens .....	32
5.6.4	Egyéb, implementáció során felmerült problémák .....	33
5.7	Az alkalmazás tesztelése .....	35
5.7.1	Általános tesztelés .....	35
5.7.2	A program hatékonyságának tesztelése .....	36
6	Összefoglalás .....	37
6.1	Eredmények .....	37
6.2	Továbbfejlesztési lehetőségek.....	38
	Irodalomjegyzék .....	39
A.	Függelék.....	41

# 1 Bevezetés

## 1.1 Motiváció

Az elektronikai eszközök elterjedésével napjainkban az emberek számos apró segítséget kaphatnak mindennapi teendők elvégzéséhez. Szinte mindenkinek a zsebében ott lapul egy okostelefon, és a legtöbb háztartásban található számítógép, internetkapcsolattal. Naptárunk immár online elérhető, bárhol megtekinthető és szerkeszthető, megosztható ismerősökkel, találkozóink könnyen, akár útközben is szervezhetőek. Egy film kikölcsönzése ma már nem igényli, hogy kimozduljunk otthonunkból, mindez elintézhető nappalinkból pár gombnyomással.

A bevezető gondolatokból látszik, hogy a technika rohamos fejlődése jelentősen könnyíti életünket, számos kényelmi funkció érhető el. A rohamos fejlődés másik velejárója, hogy napjaink embere kénytelen igazodni az információáramlás gyorsaságához és a technika fejlődéséhez. Végtélhetetlen mennyiségű irodalom érhető el egyetlen kattintással, így a tanulni vágyók számtalan lehetőségből választhatnak. Míg azonban az elektronikai eszközök rengeteg kényelmi funkciót nyújtanak, az ezekkel támogatott tanulás éppen csak felvirágzóban van [1].

Számos próbálkozás született már de ezek még magukon hordozzák a hagyományos tanulás vonásait: egy felületen, leckéről leckére haladva kell elsajátítani az elméletet, majd ezeket egy gyakorló felületen átismételni. Ebben a megközelítésben a programok lényegében a hagyományos tankönyv-munkafüzet modell digitális manifesztációi. Természetesen ez a módszer jól bevált, és már sokszor bizonyított az iskolapadokban, ám ezt módosítás nélkül átvenni mégsem olyan hatékony, mint ahogy azt elsőre gondolnánk. A tanárral végzett tanulás több szempontból hatásosabb:

- a tanuló direkt visszacsatolást kap teljesítményéről,
- a társak és tanár előtti teljesítménykényszer motiváló erőként hat,
- a rendszeres tanórák rutinszerűen beépülnek a mindennapjainkba.

Ezzel szemben egy hagyományos modellre épülő számítógépes programban végzett tanulás során a tanuló saját maga értékeli az eredményességét, és az egyetlen motiváló erő a saját eltökéltsége lehet. Mivel a program bármikor használható, a rutinszerű használat jellemzően nem valósul meg. A fentiekből könnyen látszik, hogy annak, aki elektronikus tanulásra adja a fejét, fel kell készülnie arra, hogy a tanulás hatékonysága a kezdeti lendület alábbhagyása után rendszerint csökken, ami a tanulási folyamat megszakadásához vezet.

A fenti problémákra természetesen születettek megoldási kísérletek (ld. [6][5][4][3]), melyek igyekeznek segíteni a hatékony elektronikus tanulást. Jelen dolgozat is egy ilyen próbálkozás, mely célja maximálisan kihasználni napjaink eszközválasztékát és ezek képességeit. A projekt nem mást igyekszik megvalósítani, mint egy szoftverrel támogatott oktatóprogramot, amely különböző hangszeren való játék tanítására szolgál. Mivel a tanulás témája meglehetősen szokatlan, a projekt számos

lehetőséget biztosít hatékony módszerek, és innovatív megoldások fejlesztésére, melyek a korábban említett problémákat igyekeznek teljes egészében áthidalni. Ehhez segítségül hívja a napjainkban elterjedt elektronikus zenei eszközöket és egy ún. Brain-Computer interfészt, azaz BCI-t is. Előbbi szerepe a tanultak begyakorlását és ellenőrizhetőségét segíti elő, míg utóbbi a tanulási folyamat beható megfigyelhetőségét biztosítja.

## 1.2 A dolgozat felépítése

A dolgozat második fejezetében a Motiváció részben megkezdett gondolatokat folytatom. Az első rész egy áttekintést ad az emberi tanulási folyamatról és az ismert nehézségekről. A második részben a mindenki által ismert klasszikus tanulási modellt ismertetem röviden, kitérve a környezet és egyén interakciójának sajátosságaira. Ezek után ismertetésre kerül több, napjainkban használt elektronikus tanulást segítő alkalmazás és weboldal, melyek mind valamilyen újszerű megközelítéssel igyekeznek elősegíteni a hatékony tanulást. Az innen merített ötleteket felhasználva, ill. tovább gondolva, a fejezet utolsó részében lefektetem a projekt keretein belül készülő alkalmazás főbb aspektusait. A készülő program ezen alapelveket használva lesz képes a későbbiekben a felhasználók igényeit kielégítő, hatékony eszközként együttműködni a tanulási folyamat minden fázisában.

A harmadik fejezet már az alkalmazás megvalósításához kapcsolódó technikai részletekkel foglalkozik. A hangszeres tanulás segítéséhez elengedhetetlen, hogy a felhasználó által használt hangszer és az alkalmazás kommunikálni tudjon, így ellenőrizhető ugyanis a felhasználó sikeressége. Ebben a fejezetben ismertetésre kerül az elektromos hangszerek és számítógépek összekötésének de facto szabványává vált MIDI technológia. Rövid történeti bevezető után a MIDI eszközök tulajdonságait és képességeit fogom bemutatni. Ez a rész a teljesség igénye nélkül bemutatja, hogy az alkalmazásban a csatlakoztatott MIDI eszközök mely tulajdonságaira és képességeire hagyatkozhatunk a megvalósítás során. A technikai alapok után ismertetem, hogy jelenleg milyen eszközök érhetőek el a piacon, így képet kaphatunk arról, hogy a megvalósított alkalmazással milyen hangszerek tanulása segíthető elő.

A negyedik fejezet a Brain-Computer interfészeket mutatja be. Ez az eszköz a megvalósítandó alkalmazás két alappillére közül a második. Az agyi aktivitást mérő eszközök alapjainak áttekintése után, röviden ismertetésre kerül, hogy ezek miként használhatóak a jelenleg piacon lévő okostelefonokkal és tabletekkel együttműködve. Ez után bemutatásra kerül az általunk is használt konkrét eszköz rövid technikai specifikációja, ill. egy kitekintés arról, hogy ez milyen lehetőségeket nyit a tanulás segítésében.

Az ötödik fejezet első részében rögzítem az eddig tárgyalt megvalósítandó alkalmazás konkrét technikai specifikációját. Az eddig alapelveként megfogalmazott gondolatok és a csatolt eszközök képességei alapján pontos elvárásokat fogalmazok meg a készülő szoftverrel szemben. A specifikáció alapján az alkalmazás terveit a

következő rész ismerteti. Kiemelten megemlítésre kerülnek az alkalmazáshoz készült képernyőtervek, és az alkalmazás magasszintű architektúrális terve. A fejezetet az architektúrális tervben rögzített főbb alrendszerek megvalósításának ismertetése zárja.

A hatodik fejezet összefoglalja az elvégzett munka során elért eredményeket, egyúttal felfedi az újonnan felismert problémákat. A fejezet utolsó része ezek alapján a továbbfejlesztés lehetőségeit taglalja majd.



## 2 Az elektronikus eszközökkel segített tanulás jellemzői

### 2.1 Az elektronikus tanulásról általában

A klasszikus tanulás folyamata jól ismert mindenki számára. Az iskolában az órán ülve a tanár elmagyarázza a tananyagot, esetleg az erre a célra írt tankönyvek egyikében ábrák és képek segítségével ezt illusztrálja is. Egy, a tankönyvhöz készített munkafüzetben a diák gyakorolhatja a megtanultakat. Később a megszerzett tudásról számonkérések formájában ad tanúbizonyságot. Noha a módszer mindenkinek ismerős, a köznyelv a tanulást inkább nem kívánatosnak, fárasztónak titulálja. Ennek ellenére az iskolát mindenki elvégzi, de ha az ember egyéni tanulásra adja a fejét, az csak a legritkább esetekben vezet sikerhez. Ennek oka, hogy az előbbi esetben motivációt jelent a tanulásra az, hogy ne bukjunk meg, esetleg az iskolatársakkal versengjünk. Továbbá az sem elhanyagolható, hogy az iskola kötelező, így évekig mindennapi rutinunk része. Napjainkra azonban számos szoftver és weboldal áll rendelkezésre, amik a legkülönbözőbb témákat tanítják az arra vágyóknak. Összefoglaló néven ezekkel a felhasználó elektronikus eszközökkel segített tanulást végezhet. Az alábbiakban számba veszem, hogy ezek milyen technikákkal tudják felkelteni és fenntartani a felhasználók figyelmét.

Először definiálom, hogy mit is értek elektronikus eszközökkel segített tanulás alatt. [1] szerint az e-tanulás a következőképp definiálható: instrukciók, melyek egy elektronikus eszközön jutnak el a felhasználóhoz, és céljuk a tanulás segítése, támogatása. Ebben a dolgozatban főképp a szoftverekkel támogatott tanulást vizsgálom. Egy ilyen szoftver a következő tulajdonságokkal rendelkezhet:

- valamilyen elektronikus eszközön vagy adathordozón található meg
- nagy mennyiségű, az elsajátítani szándékozott témával kapcsolatos anyagot tartalmaz
- interaktív elemekkel igyekszik segíteni a tanulást
- instrukciókkal, gyakorlatokkal és visszacsatolással segíti a tananyag elsajátítását
- távoli instruktorként felügyelt tanulási folyamatot valósíthat meg (szinkron) vagy egyedi tanulásra tervezett
- segíti a felhasználókat a tananyag mély elsajátításában

Minden tanulást segítő szoftverre igaz néhány a fenti tulajdonságok közül. Ezek a szoftverek azonban a bevezetőben említett módon az alábbi hátrányokkal rendelkezhetnek a hagyományos tantermi tanulással szemben:

- motiváció hiánya
- a tanártól kapott direkt visszacsatolás hiánya
- rutinszerű tanulás hiánya

Az [1], [3] és [4] forrás számos módszert megemlít, melyekkel a motiváció, a versenyszellem és a tanulás iránti elkötelezettség növelhető. Ezek hatására a

tanulást segítő szoftverek jelenős fejlődésen mentek keresztül az elmúlt években, és fejlődésük napjainkban is folytatódik. A közelmúltban több alkalmazás is feltűnt, amelyek a játékok szórakoztató hatását, a közösségi élményt és a teljesítmény alapján személyre szabott tartalmat kihasználva innovatív módon igyekeznek növelni a felhasználók tanulási kedvét.

A játékos elemeket használó eszközök hatására a felhasználóban megnövekszik a dopamintermelődés, és [2]. kutatásai alapján ez egy egészséges emberben segíti a tanulást, a tanulásra fordított időt pedig kevesebbnek érezteti. A felhasználók az ilyen programokat nem csak szívesebben használják, de nagyobb eséllyel lesznek visszatérő felhasználók [3].

Az alkalmazások közösségi integrációja a felhasználók versenyszellemére hat. A hasonló funkciókkal rendelkező alkalmazásokat a felhasználók előszeretettel használják, majd osztják meg eredményeiket barátaikkal, ismerőseikkel. Ilyen funkció lehet egy toplista, vagy saját tartalom készítésének és megosztásának lehetősége is [4]. Az ismerősöktől kapott pozitív visszajelzés, vagy éppen a barátok megelőzésének a lehetősége a toplistán elég ok arra, hogy a felhasználók az egyes feladatok újra és újra ismételve egyre jobb eredményt próbáljanak elérni. Ez persze egy tanulásközpontú játéknál a tananyag mélyebb elsajátítását is magával hozza.

A fenti megoldásokat manapság már számos alkalmazásban és weboldalon megfigyelhetjük. A következő részben ezeket ismertetem, hogy majd ötletet merítve a saját zenei oktató alkalmazás főbb irányelveit is le tudjam fektetni.

## **2.2 Példák a módszer alkalmazására**

Ahogy a bevezető fejezetben említettem, az elektronikus eszközökkel segített tanulás számos kihívást állít az ilyen szoftverek készítői elé. A fő cél, hogy a fenntartsák a felhasználók figyelmét és kedvét a tanuláshoz. Az általános elveket az előző részben már láthattuk, azonban ezeket különféle módokon lehet a gyakorlatba ültetni. Éppen ezért fontosnak éreztem, hogy megvizsgáljak számos, való életből vett, valóban működő példát.

### **2.2.1 Wu's castle**

Az első példa a Wu's castle, amely egy 2-dimenziós szerepjáték. A játékot a North Carolina Egyetemen készítették, és célja nem más, mint a Computer Science nevű tárgy oktatásának segítése. Bővebben információ az [5] forrásban található. A játékban a felhasználók két tevékenységre képesek: ciklusváltozók manipulálására, és a karakterük irányítására, amivel ciklusokat kell bejárniuk. Ezekkel az elemekkel az egyszerű vezérlési szerkezetektől a bonyolultabb algoritmusokig sok küldetés modellezhető, és a játékon keresztül megtanítható.

A játék jelentős hangsúlyt fektet a visszacsatolásokra: hibás és helyes válaszok különböző vizuális ingereket váltanak ki. Ha a karakter végtelen ciklusba kerül, a

karakter életereje csökken, ha a felhasználó kiindexel egy tömbből, az robbanást okoz.

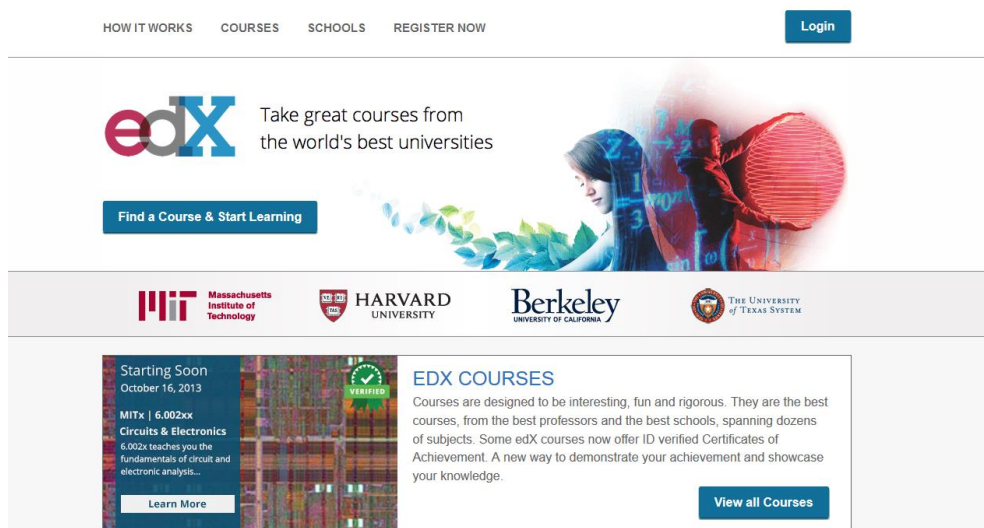


1. ábra: A játékos éppen két egymásba ágyazott ciklust jár be.<sup>1</sup>

Az [5] tanulmány rámutat, hogy a játékot használók összességében könnyebben sajátították el a tananyagot, és jobban teljesítettek az erre a célra készített teszteken.

### 2.2.2 edX

Második példaként a <http://www.edx.org> hozható, amely a világ vezető egyetemein hallgatható kurzusokat teszi elérhetővé mindenki számára. A különböző kurzusokat valódi oktató tartja, akinek az előadásait online követhetik az érdeklődők. Az előadások később visszanezhetők, a tananyag mellékelt fájlokban, vagy az ajánlott irodalomjegyzékben szintén elérhető.



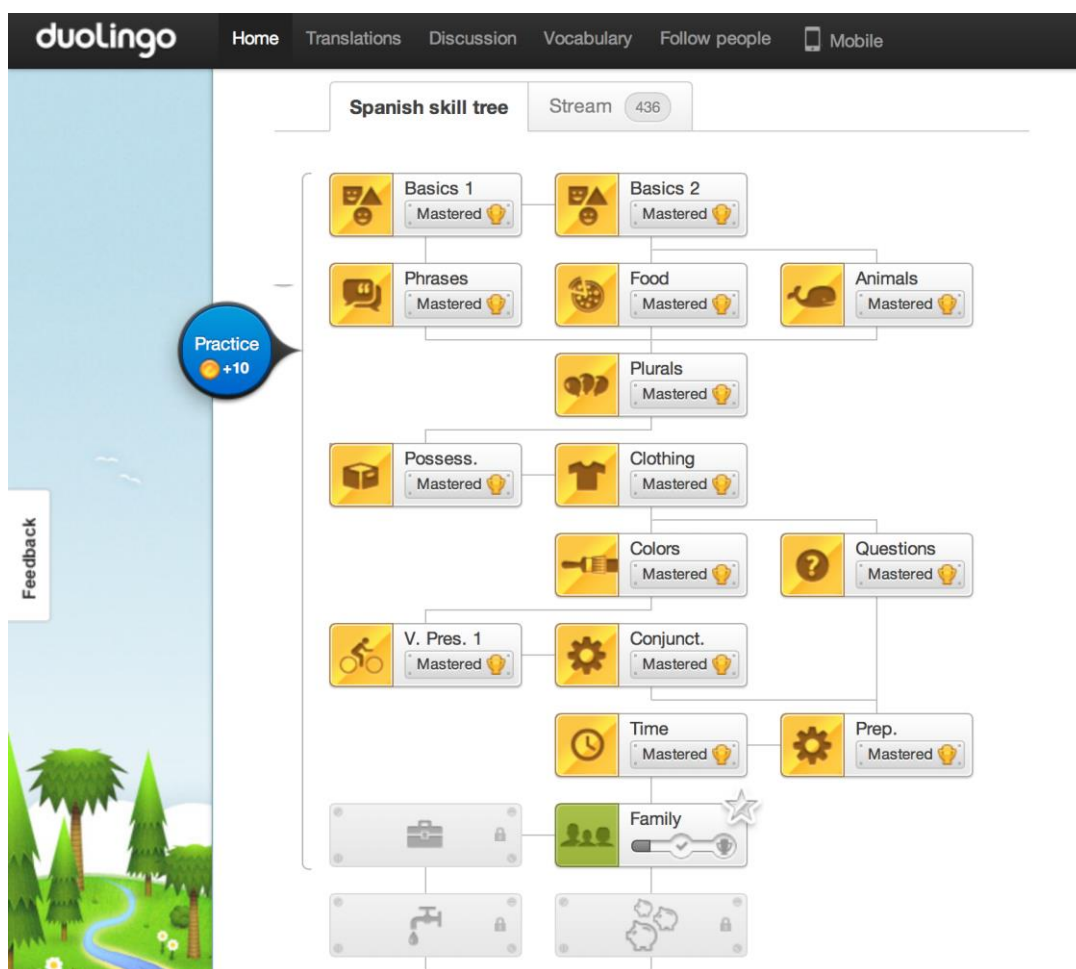
2. ábra: A képen a <http://www.edx.org> látható.

<sup>1</sup> Forrás: <http://src.acm.org/Eagle/eagle/wuNumberFigure.jpg>

A kurzusok hallgatói házi feladatokat és félévközi, ill. félév végi vizsgákat is teljesítenek, melyek eredménye mind beleszámít a kurzus végén kapott jegybe. A kurzus közben az ismerősök eredményei láthatóak, ez a kompetitív szellemiségű tanulóknak jelentős motiváció. A kurzus elvégzéséért oklevél kapható az elért eredménnyel, ez megosztható különböző közösségi oldalakon.

### 2.2.3 Duolingo

Végezetül egy nyelvtanulásra használható oldalt mutatok be, amit használva könnyen és játékosan sajátíthatunk el nyelveket. A weboldal iskolapéldája lehetne a motivációteremtésnek és az érdeklődés felkeltésének. A Duolingo projekt Luis von Ahn nevéhez fűződik és nem kisebb célt tűz ki maga elé, mint az internet lefordítása [6].



3. ábra: A Duolingo leckék és egymástól függésük.

A felhasználók egy nyelvi kurzuson vesznek részt ahol különböző témájú modulok oldanak meg. Egy modul megoldásának feltétele, hogy az előkövetelmények már teljesítettek legyenek. A felhasználók motiválására számos eszközt használ az oldal: a modulok megoldásáért pontokat gyűjtenek, és bizonyos pontok hatására magasabb szinteket érhetnek el a felhasználók. Az oldalon szereplő ismerősök bejelölhetőek, és azok azonnal bekerülnek egy személyes, barátokból álló toplistába.

Ha valaki, a saját toplistánkon megelőz minket az oldal e-mailben figyelmeztet, csakúgy, mint a napi gyakorlás jelentőségére. Ha a felhasználó rendszeresen gyakorol, bónusz pontokat gyűjt, ill. az oldal számon tartja a legtöbb egymás után gyakorlással töltött napok számát. A nyelvi modulok teljesítése mellett, a modulokhoz kötődő szövegrészletek lefordítására is lehetőség van, amelyek az internetről származó (pl. Wikipedia) szövegfoszlányok. Amennyiben több felhasználó hasonló fordítást ad egy szövegre, az oldal lefordítottak könyveli el a mondatot.

### 2.3 A projekt során használt megközelítés

A saját alkalmazás tanuló modulja a fenti példák alapján már könnyen meghatározható. Az alapelvek lefektetése közben szem előtt tartottam, hogy az oktatás mellett a szoftver szórakoztató is legyen. A játékmenet ebből eredően meglehetősen laza és interaktív, de a játék által szolgáltatott tartalom minden esetben pontos és kimerítő.

Az alkalmazás leckékre bontott egységekre tördeli a zenei elmélethez és hangszeres játékhoz kapcsolódó elméleti anyagokat, melyek különböző gyakorlatok során sajátíthatók el. A leckék egymás után következnek, egy új lecke csak minden előző anyag teljesítése után érhető el. A leckéken belüli gyakorlatok már szabadabban, bár még mindig némi megkötéssel választhatóak. A végső verzió pedagógiailag megfelelő felépítésében a Zeneakadémia Zenepedagógiai Tanszéke van segítségemre.

A gyakorlatok értékelése a teljesítés sikerességével arányosan történik: 4 csillagos skálán értékelhető a felhasználó teljesítménye. A lecke a benne lévő gyakorlatok eredménye alapján hasonlóképpen értékelődik. Az elért pontok alapján a felhasználó különböző achievementek (kitüntetések) megszerzésére képes. Ezek a kis kitüntetések jelzik, hogy a felhasználó elért egy szintet (pl. 500 lecke teljesítése stb.), vagy valamilyen különleges eredménnyel büszkélkedhet (100% eredmény egy gyakorlat során).

Ezek az eredmények a felhasználói profilon tekinthetőek meg, amelyeket más felhasználók, barátok és ismerősök egyaránt látnak. A felhasználók teljesítményeik alapján toplistán is megtalálhatóak.

Maga a gyakorló modul jelentősen eltér az eddig megszokott alkalmazásoktól. Mivel a gyakorlás során egy hangszeret kell használni, elengedhetetlen, hogy ezt össze lehessen kötni a szoftvert futtató géppel. Erre a célra az elterjedt MIDI szabványt használó hangszerek kiválóan alkalmasak. A gyakorlás során a lejátszandó kotta megadott ütemben megjelenik a kijelzőn, melyet a felhasználónak helyesen kell lejátszania. A gyakorlás így az egyszerű hangoktól a bonyolultabb darabokig bármit tartalmazhat.

Azért, hogy a megtanult gyakorlatok valóban jól rögzüljenek a gyakorlás során a felhasználót egy Brain-Computer interfész vizsgálja, mely eldönti, hogy szükséges-e további gyakorlás, esetleg vissza kell-e lépni, hogy átismételje a korábban tanultakat.



Ebben a részben két új eszköz is előkerült, melyek a megvalósítandó alkalmazással együttműködve adják a teljes játékélményt, és a személyre szabott tanulási folyamatot. Ezen eszközök megismerése az alkalmazás megvalósítása előtt elengedhetetlen, így a következő két fejezetet ezen két eszköznek szentelem.

### 3 MIDI eszközök bemutatása

#### 3.1 A MIDI rövid története

A zenélés hőskorában számos kísérleti elektromos hangszer látott napvilágot. Ezek a többnyire kísérleti célú hangszerek azonban csak egyetemeken és különböző kutatóközpontokban voltak elérhetőek, így a nagyközönség kevéssé ismerhette őket. Ahogy azonban az árak elkezdtek rohamosan csökkenni, több zenész számára megvásárolhatóak lettek ezek az eszközök. Sorra jelentek meg a különböző, általában billentyűs hangszerek, és a zenészek igyekeztek ezekből a hangszerekből minél többet beszerezni. Ugyanis a kezdeti hangszerek zenei jellemzői, a billentyűk száma és a képességeik jelentősen különböztek. A zenészek természetesen minél sokszínűbb hangzást szerettek volna elérni, így hangszerek tucatjait halmozták fel. Ennek kapcsán hamar felmerült az igény arra, hogy a sok eszközt összekössük, és egy vezérlőn keresztül szólaltassuk meg a többit. A kezdeti analóg technikának azonban számos hátránya volt: mivel analóg jelről beszélünk, az összekötő kábelek feszültsége a hangmagasságtól függően változott. Így ezek a megoldások nagyon pontos hangolást igényeltek. A másik probléma, hogy a kábeleken egyszerre egy feszültségintet lehetett kiadni, így a többszólamúság kezelése is problémás volt.

A megoldást természetesen a digitális kapcsolatok jelentették. Ez már csak azért is volt logikus lépés, mert abban az időben kezdtek elterjedni a digitális vezérlést használó szintetizátorok. Több nagy cég, a Roland, az Oberheim Electronics és a Sequential Circuits is gyártott ilyen eszközöket. Ezek a gyártók azonban mind saját interfészt használtak az adatátvitelre, ami hamarosan újabb problémát szült: nem lehetett összekötni két különböző gyártó készülékeit [7].

Éppen ezért a nagy cégek 1981-ben közös szabvány kifejlesztését kezdték meg. A szabvány célja az volt, hogy a különböző gyártótól származó szintetizátorok könnyen összekapcsolhatók legyenek. A szabvánnyal kapcsolatban az alapvető elvárások a következők voltak: az egyszerű vezérlés minden eszközön ugyanúgy működjön, és tetszőleges számú eszköz csatlakoztatható legyen. Ezeket a szempontokat figyelembe véve fejlesztette ki a Sequential Circuits az USI (Universal Synthesizer Interface) szabványt, ami történetileg a MIDI elődjének tekinthető. Az USI szabvány csak a használt csatlakozó típusában és átviteli sebességében különbözött a MIDI-től [7].

Egy évvel később, a nagy gyártók a fennmaradó kérdéseket megvitatva, pár változást beiktatva létrehozták a MIDI (Musical Instrument Digital Interface)

szabványt. Hamarosan számos gyártó kezdett el MIDI kompatibilis eszközöket gyártani, és az eszközök száma évről-évre gyarapodott.

A szoftvergyártók hamar felismerték a MIDI-ben rejlő lehetőségeket, és szoftveres hangkeltő eszközöket kezdtek gyártani. Ezek a programok rugalmasságuk és sokoldalúságuk miatt hamar elterjedté váltak. A MIDI történetének részletesebb ismertetése a [7] forrásban megtalálható.

A MIDI szabvány fejlődése nem állt meg, keletkezése óta számos változáson esett át, és napjainkban is számos eszköz és szoftver használja. Sőt, az USB-csatlakozóknak hála rengeteg eszközhöz, pl. okostelefonhoz és tablethez is csatlakoztatható.

### 3.2 MIDI eszközök képességei

A MIDI szabványt eredetileg zenei eszközök összekötésére és vezérlésére fejlesztették ki. A szabvány mára azonban jelentősen kibővült, és sok olyan eszközt is támogat, amely nem kapcsolódik szorosan a zenéléshez, pl. koncert közben használt világítástechnika.

Ennek megfelelően számos különböző adatot továbbíthatunk MIDI-n keresztül. Ezeket az adatokat MIDI-üzenetnek hívjuk. Egy üzenet legalább 1 bájt hosszú, és státusz, ill. adatbájtokból áll. A státuszbájtok az üzenet fejlécei, meghatározzák az adat típusát, így a fogadó hangszer tudni fogja, hogy hogyan értelmezze a kapott adatokat. Az adatbájtok a státuszbájtokban meghatározott típusú adatokat tartalmazzák. Azért, hogy a státusz- és adatbájtok könnyen megkülönböztethetők legyenek a státuszbájtok MSB-je (Most Significant Bit) mindig 1, ugyanígy az adatbájtok MSB-je mindig 0. Az üzenetek 5 alaptípussal rendelkeznek:

- **Channel Mode Messages:** MIDI üzemmódok között válthatunk segítségével.
- **Channel Voice Messages:** Zenélés közben előforduló események továbbítódnak ilyen üzenetek formájában, pl. billentyű lenyomás és felengedés.
- **System Common Messages:** Valamely eszköznek szóló üzenet.
- **System Real Time Messages:** Szinkronizációs célokat szolgáló üzenetek.
- **System Exclusive Messages:** Hangszer specifikus vezérlési parancsok.

A fenti üzenettípusok nagy részét persze nem fogom használni, ezért a három legjelentősebb típust kiemelem, és ismertetem a részletes szerepüket, ill. a típusba tartozó lényeges üzeneteket. A három lényeges csoport a készítendő szoftver szempontjából: Channel Mode Message típusú üzenetek az üzemmódválasztáshoz elengedhetetlenek, a Channel Voice Message típusú üzenetek fogják kódolni a lejátszott zenei hangokat, a System Real Time Messages típusú üzenetekkel pedig a szinkronizációt valósítom meg. A következő részekben ezen típusok ismertetése következik.

### 3.2.1 Channel Mode Messages

A MIDI készülékeknek beállíthatunk egy négy bites csatornaazonosítót. A vezérlőjeleket egy megadott csatornán adhatjuk ki, és csak azok az eszközök reagálnak rá, amelyeknek a csatornaazonosítója megegyezik az üzenet csatornaazonosítójával. Ily módon egyszerre több eszközt kezelhetünk egyetlen vezérlővel. Persze felmerül a kérdés, hogy mi van akkor, ha szeretnénk, hogy egy eszköz több csatornát is figyeljen. Erre valók a MIDI üzemmódok:

- **OMNI ON, POLY:** A hangszer minden üzenetre reagál, csatornakódtól függetlenül. A hangszer ilyenkor polifón módban üzemel, azaz képes egyszerre több hang megszólaltatására is.
- **OMNI ON, MONO:** Hasonló az előzőhöz, de a hangszer egyszerre csak egy hangot szólaltat meg.
- **OMNI OFF, POLY:** Az eszköz csak a saját csatornáját figyeli, polifón módon.
- **OMNI OFF, MONO:** Az eszköz csak a saját csatornáját figyeli, de egyszerre csak egy hangot játszik. Ez a gitárjátékhoz hasonló: a húrok egymástól függetlenül szólnak, de egy húr csak egy hangot játszik [7].

### 3.2.2 Channel Voice Messages

A hangszerek a hangokat ilyen üzeneteken keresztül szólaltatják meg. Nem csak tiszta zenei hangok, hanem hajlítások, a megszólalás ereje is ilyen üzenetek formájában kerül továbbításra. A legfontosabb üzenetek:

- **Note On:** Ezt az üzenetet a hangszer akkor küldi, ha egy billentyűjét lenyomták. A megszólaltatott hangot egy 0-127 közötti szám kódolja. Oktávonként 12 hanggal számolva így több, mint 10 oktávot le lehet fedni. A hang addig fog szólni, amíg az adó egy következő üzenettel le nem állítja. Az üzenet tartalmaz egy további értéket: egy 0-127 közti szám kódolja, hogy a billentyűt milyen gyorsan nyomták le. A 0 sebesség speciális értelmezéssel bír a MIDI specifikáció szerint: ez egyenértékű az adott hangra kiadott Note Off üzenettel.
- **Note Off:** Ez az üzenet az előző fordítottja: a felengedés sebessége és a felengedett hangot tartalmazza. Hatására a hangszer a hang lecsengési fázisát kezdi játszani, mely fokozatosan elhalkul.
- **Pitch Bend:** Ez a 2 adatbájtot tartalmazó üzenet kódolja a hajlítás mértékét. Mivel 2 adatbájtos,  $2^{14}$  (mindkét adatbájt 0-val kezdődik, a maradék 7-7 bit a felső ill. alsó 7 bitet adja) különböző érték kódolható így. A 8192 érték kódolja a hajlítás nélküli állapotot, az ennél kisebb számok lefele hajlítást, az ennél nagyobbak felfele hajlítást kódolnak.
- **Polyphonic Pressure:** A billentyűkre eső nyomás erőssége, billentyűnként kódolva.
- **Channel Pressure:** Ez a billentyűkre eső nyomás, de nem billentyűnként, hanem a teljes csatornára átlagolva.





A másik támogatott eszköztípus a MIDI gitár hangszedő és a MIDI fúvós hangszer. Előbbi eszköz gitárra szerelhető, és a lejátszott hangok MIDI vezérlőjellé alakítva eljutnak a szoftverhez, amelyeket az felhasználó tud dolgozni.



5. ábra: Roland GK-2A MIDI gitár hangszedő.<sup>3</sup>

Utóbbi pedig fúvós eszközként használható, és a megszólaltatott hangok MIDI jelként jutnak a számítógépre. A támogatott képességek itt is hasonlóak a MIDI billentyűzet esetén támogatottakhoz, azaz feldolgozható a játszott hang és a pengetés intenzitása is.



6. ábra: Az Akai EWI fúvós MIDI hangszere.<sup>4</sup>

<sup>3</sup> Forrás: <http://home.epix.net/~joelc/gk-2a.jpg>

<sup>4</sup> Forrás: <http://www.akaipro.com/product/ewiusb>

## 4 Brain-Computer interfész használata Android alapú rendszereken

### 4.1 A Brain-Computer interfész bemutatása

A Brain-Computer interfész (BCI) egy külső eszköz, mely a felhasználó agyi aktivitását mérve képes a BCI-hez csatlakoztatott eszközök vezérlésére [9]. A BCI célja nem a felhasználó akaratának előrejelzése, hanem egy új kommunikációs csatorna nyitása az emberi agy és az elektromos eszközök között [8]. Az említett agyi aktivitás különböző módszerekkel mérhető, ezek két csoportba sorolhatók [8]:

- **invazív technikák:** A mérő eszközt az emberi testen belül kell elhelyezni, pl. mélyagyi elektróda.
- **non-invazív technikák:** A mérő eszköz az emberi testen kívül van, kívülről mérhető jellemzőket érzékel, pl. pulzust, izommozgást.

Természeténél fogva a dolgozat szempontjából a non-invazív technológiák érdekesebbek, hiszen ezek teszik lehetővé a mindennapi és kényelmes használatot, ami követelmény projektünk szempontjából. A non-invazív BCI technológiák is sokfélék lehetnek, minden olyan technikát értünk ez alatt, ami valamely módon az agyban lejátszódó folyamatok alapján mér, majd számítógépes feldolgozásra továbbítja. A dolgozatban a továbbiakban BCI alatt az agyi aktivitást mérő, non-invazív eszközöket értem.

### 4.2 A Brain-Computer interfészek története

Hans Berger az 1920-as években sikeresen vezetett el agyi bioelektromos jeleket emberről, melyre a fejen mérhető elektromos aktivitásból következtetett [10]. A technika később elektroencefalográfia (EEG) néven vált ismertté. Már abban az időben kialakult egy feltevés, hogy elektromos eszközöket is lehetne irányítani a technikával. Az első eredményekre az 1970-es évekig kellett várni, amikor is kifejlesztették az első primitív vezérlőt, amely EEG-vel leolvasott adatokra támaszkodott [9].

A DARPA ebben az időben kezdte támogatni a hasonló témájú kutatásokat, ugyanis a technológiában nagy potenciált látott, mint katonákat segítő berendezés. A nagy mentális stressznek kitett katonák segítésére az első próbálkozások kudarcba fulladtak, így a DARPA kénytelen volt általánosabb célokat kitűzni: olyan eszközt szerettek volna fejleszteni, amely valós idejű adatok alapján berendezéseket irányít. Az UCLA egyetem Brain-Computer Interface Laboratory falai között megszületettek az első eredmények [11].

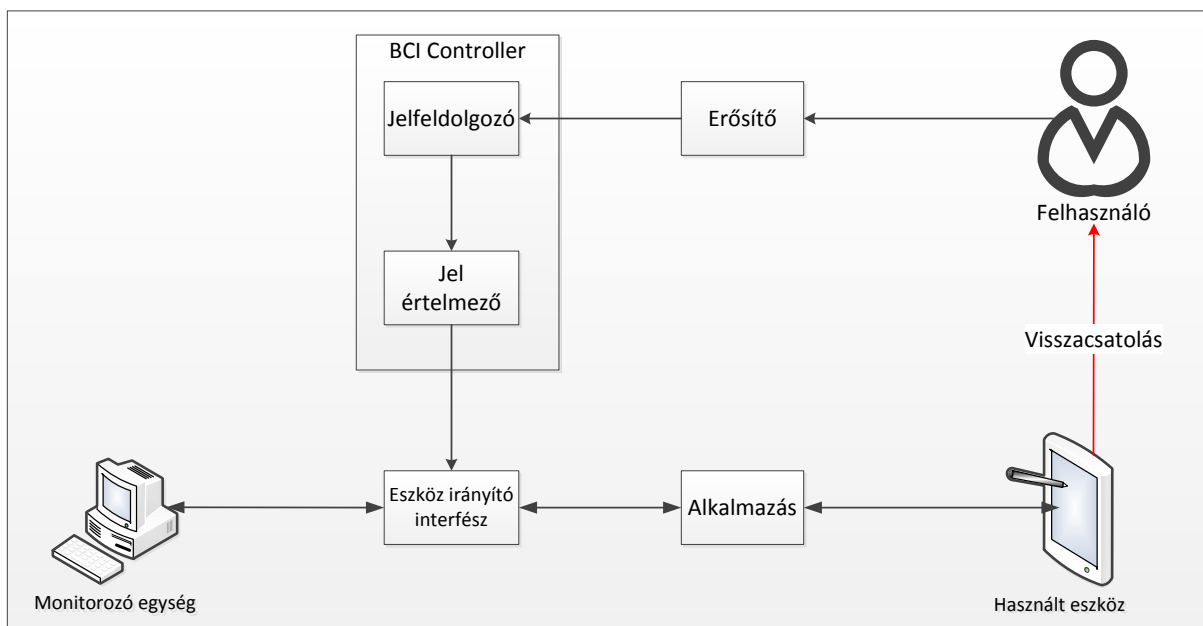
Napjainkra az EEG-technológia olyan szintű fejlődésen ment keresztül, hogy a BCI-k a mindennapi élet kisebb dolgaiban hatékony segítségnek bizonyulnak, bár ezek nem széles körben elterjedt termékek, inkább laboratóriumi körülmények közt használtak. A kutatások másik érdekes iránya, az okostelefonokhoz és tabletekhez csatlakoztatott egyszerű BCI-k vizsgálata. A technika gyors fejlődése lehetővé tette a hagyományos, sokszor dollármilliókba kerülő eszközök helyett egyszerűbb, átlag

felhasználó számára is elérhető termékek gyártását. Ezek pontossága azonban nyilvánvalóan nem vetekedhet az előbb említett, laboratóriumi gépekkel, de hétköznapi felhasználásra megfelelőek.

### 4.3 A Brain-Computer interfész működése

A BCI használatához ismerni kell a BCI-t használó rendszerek felépítését. Jelen projekt során a BCI mindig az alább ismertetett kontextusban működik. Ennek a kontextusnak az ismerete nélkülözhetetlen a visszacsatolási folyamat megértéséhez, éppen ezért az alábbiakban ismertetem egy BCI-t használó rendszer fő építőelemeit és ezek feladatát. Jelen rendszert [12] munkája alapján készítettem. A fő építőelemek:

- **felhasználó:** a BCI eszközt viselő, és a szoftvert használó ember.
- **erősítő:** a mért jelek erősítésére szolgál, opcionális.
- **BCI controller:** feladata a bejövő, mért adatok alapján a megfelelő logikai utasítások kiadása az eszköz felé, két részből áll:
  - **jelfeldolgozó:** értelmezi a kapott jelek közül a számára érdekeseket.
  - **jel értelmező:** a feldolgozott jeleket logikai utasítássá fordítja
- **Eszköz irányító:** az eszközön futó alkalmazás, ami a logikai utasításokat közli az alkalmazás megfelelő interfészével
- **Monitorozó egység:** Az eszköz irányító tevékenysége figyelhető meg ebből az alkalmazásból.
- **Alkalmazás:** az eszköz irányítóval egy interfészen keresztül kommunikál, a számára eljuttatott parancsokat feldolgozza, és cselekszik, ami a felhasználó számára egy visszacsatolásként jelenik meg.



7. ábra: Az ábra a BCI rendszerek egy általános felépítését mutatja.

A dolgozat fő témája egy rendszerbeli alkalmazás készítése.

#### 4.4 Használatuk a tanulás segítéséhez

A BCI-k használata tanulás segítéséhez az eszköz alkalmazásának igen újszerű megközelítése. Mivel a BCI képes a felhasználó agyi aktivitását érzékelni, a tanulás közben keletkező heves vagy kevésbé heves ingerek detektálásra is lehetőség van. Az ötlet ezen értékek felhasználása, és ez alapján a gyakorlás bizonyos paramétereinek változtatása. Így a gyakorlás a felhasználó igényeihez igazítható, és a tanulás folyamat hatékonysága tovább növelhető. Az alábbi lista pár lehetséges alkalmazást sorol fel:

- gyakorlás sebességének módosítása
- a gyakorlat cseréje, ha a felhasználó feltehetőleg unatkozik gyakorlás közben
- továbblépés, ha a felhasználó a jelenlegi gyakorlatot már jól ismeri
- szünet javasolása, ha a felhasználó fárad

Ezek a visszajelzések már az előző részben látott Eszköz irányító interfészen keresztül jutnak el az alkalmazásba, és többek között az alábbi paraméterek mérésével határozhatók meg:

- a P300-as komponens időbeli eltolódása
- béta frekvenciatartományú agyi aktivitás

Bővebb információ a BCI által mért értékek és a belőle levonható következtetések kapcsolatáról Nagy Richárd TDK dolgozatában olvasható. A jeleket feldolgozó és értelmező keretszoftver Angeli Róbert és Fekete András munkája, részletekért ld. az általuk készített TDK dolgozatot.

#### 4.5 A projekt során használt BCI

A projekt során használt eszköz az Emotiv EPOC Neuroheadset, amely 14 érzékelővel és beépített giroszkóppal rendelkezik, vezeték nélkül csatlakoztatható az őt használó feldolgozóeszközhöz, így használata kényelmes és egyszerű. A BCI illesztését Nagy Richárd végezte, bővebb ismertető az ő TDK dolgozatában található.



8. ábra: Az Emotiv EPOC Neuroheadset.<sup>5</sup>

<sup>5</sup> A kép forrás: <http://www.emotiv.com/store/hardware/epoc-bci/epoc-neuroheadset/>

## 5 Az alkalmazás tervezése és megvalósítása

### 5.1 Általános ismertető

A projekt alapvetően a megerősítéses tanulási modellt igyekszik felhasználni interaktív környezetben, a cél különböző hangszereken való játék elsajátítása. Ehhez két speciális hardvert is felhasznál: az egyik egy MIDI-hangszer, ami eljuttatja a lejátszott hangot a programnak és ezáltal lehetővé teszi a megszólaltatott hang minőségének kiértékelését, a másik a felhasználó agyi aktivitását méri (EEG), ez alapján von le következtetéseket a tanulás hatékonyságáról. A cél, hogy interaktív, játékos tanulási folyamat során a felhasználók könnyedén, a hagyományos tanulás során tapasztalt „unalom” és „kedvtelenség” érzések mellőzésével gyorsan elsajátíthassák valamely hangszereken a játék alapjait.

#### 5.1.1 Célok

A bemutatott program célja, hogy a modern technika vívmányai segítségével mindenki számára elérhető, könnyű tanulási keretrendszert nyújtson hangszerekhez, így elősegítve a zenei kultúra szélesebb elterjedését.

Ehhez elsősorban egy tablet készülékre és a rajta futó oktatóprogramra, másodsorban az agyhullámokat pásztázó készülékre és egy MIDI szabványt támogató hangszerre van szükség.

### 5.2 Specifikáció

#### 5.2.1 Általános áttekintés

Az oktatórendszer három jól elkülöníthető egységből épül fel: az oktatószoftver, a külső EEG eszköz, a külső MIDI eszköz. Maga a megerősítéses tanulás a szoftver interaktív felületén történik, amelyet használva a két csatlakoztatott eszköz visszacsatolást biztosít a felhasználónak.

#### 5.2.2 Funkciók

Az oktatószoftver tehát két visszacsatolással is rendelkezik a környezetéről: figyeli a felhasználó agyi aktivitását és az általa végzett tevékenység minőségét (zenei hangok tisztasága). A szoftver használatához ki kell választanunk, hogy mely **hangszert** szeretnénk a használatával gyakorolni. Ekkor különböző, egymásra épülő **leckék** közül választhatunk, melyek egymás utáni elvégzése alkalmas a kiválasztott hangszereken való játék elsajátítására. Minden leckére négy különböző **értékelést** lehet kapni, attól függően, hogy a visszacsatolások alapján milyen minőségben sikerült azt elvégezni (lásd később). A leckék az egészen egyszerűtől (pl.: akkordok pengetése gitáron) a nehézig (komplett szám eljátszása) váltakoznak, és lehetnek olyan leckék melyek előírják más, korábbi anyagok teljesítését. **Szabad gyakorlás** is lehetséges a megadott témakör(ök)ből, amivel a felhasználó újabb tapasztalatokat szerezhet. A felhasználó a leckék ill. szabad gyakorlás teljesítéséért **pontokat** is kap, melyek kifejezik, hogy mennyi a tapasztalata az adott hangszerekből.

A **gyakorló felület** egy kotta sematikus képe, melyen a lejátszandó hangok be vannak jelölve. A gyakorlás indításával a kotta lassan mozogni kezd, a felhasználó feladata a hangszerrel követni a kottát. Eközben az egyik visszacsatoláson figyelemmel követjük, hogy a megszólaltatott hangok helyesek-e, míg a másik **visszacsatolás** a felhasználó agyi aktivitását méri. Az agyi visszajelzések alapján gyorsítható pl. a kotta mozgása (ha a felhasználónak érezhetően nem kerül nagy erőfeszítésbe a lejátszása), pontatlanság esetén bekapcsolhatók segédvonalak/metronóm, unalom vagy fáradtság esetén pedig a felhasználó elküldhető pihenni. Az így mért adatok alapján egy **százalékos teljesítmény** számolható, ami alapján a lecke négyfokozatú minőségi értékelése kialakítható. Ha a felhasználó nem felelt meg, újra kell próbálkoznia, ha megfelelt 1-4 „**csillaggal**” jutalmazandó, a százalékos eredmények alapján. A rendszer azonnali visszacsatolást is ad, aminek szerepe az elvárt és tényleges jutalom elemzése. Ebből számos dologra, pl. a játékos frusztráltságára lehet következtetni. Bővebben ld. Szegletes Luca doktorandusz kutatását.

A felhasználó az általa elért eredményekről rendelkezik egy **profil**al, ami összefoglalja mindazt, amit a program használatával elért. Ez elsősorban egy összefoglalás saját részre, de a felhasználó ezt minden további nélkül **megoszthatja** bármely közösségi médiumon (facebook, twitter). A profil tartalmazhat kitüntetések, melyek különböző **achievementek** elérését jelenti (pl. 100%-os lejátszott dal).

Az itt leírt szoftver a könnyű használat és a magas fokú hordozhatóság érdekében tablet alkalmazás formájában kerül megvalósításra, Android platformon.

### 5.2.3 Felhasználók

A szoftver felhasználói olyan zenekedvelő (többnyire fiatal) tablet felhasználók, akik célul tűzték ki valamely hangszer elsajátítását, ám ehhez nem kívánnak órákat ill. magántanárt igénybe venni. Ebben az esetben a készítendő szoftver az igényüket tökéletesen ki tudja elégíteni, és az alkalmazás kellően gyakori, és lelkiismeretes használata a megszokottnál gyorsabb és alaposabb egyéni tanulást segít elő.

### 5.2.4 Elvárások

Az alkalmazással szemben alapvető elvárás, hogy napjaink vezető mobilplatformján (Androidon) futtatható, könnyen és gyorsan használható, játékszerű élményt nyújtó alkalmazás legyen. Ehhez a körültekintő tervezésen kívül egyéb felhasználói szempontok is előtérbe kerülnek.

A szoftver gyors, rezponzív és dinamikus kell, hogy legyen, nem megengedhető, hogy az alkalmazás hosszú betöltődése, illetve egyéb várakozása a felhasználót „untassa”. Különösen kritikus az elindulás, hiszen ekkor a felhasználó még nem vette rá magát a tanulásra. Egy kellően gyorsan elkezdett játék azonban akár pozitív irányba is billentheti a mércét.

Alapvető elvárás, hogy szoftver lekösse felhasználóit, így a visszacsatolások alapján történő osztályozás nem lehet se túl engedékeny (ekkor hamar „végigjátsszák”) se



pedig túl szigorú (elveszi a lelkesedést). Utóbbi két feltétel természetesen a megerősítéses tanulást is elősegíti.

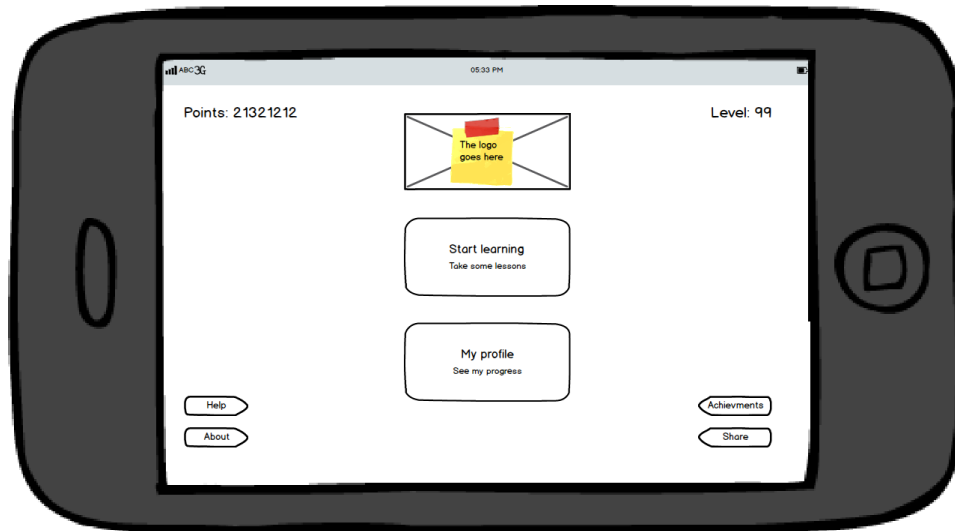
### 5.2.5 Szótár

A projekt során használt fogalmak rövid összefoglalása és magyarázata méretei miatt az A Függelékben kapott helyet.

## 5.3 Képernyőtervek

A következő szakaszban az alkalmazás képernyőtervei kerülnek megtervezésre. Ezek egyrészt segítenek megérteni az egyes funkciók közötti kapcsolatot, másrészt a végleges, alkalmazás által használt felhasználói felület ezen tervek alapján készül.

### 5.3.1 Főképernyő



9. ábra: Főképernyő

A főképernyő a program belépési pontja, indítás után ide jut a felhasználó. Itt két fő lehetőség közül választhat:

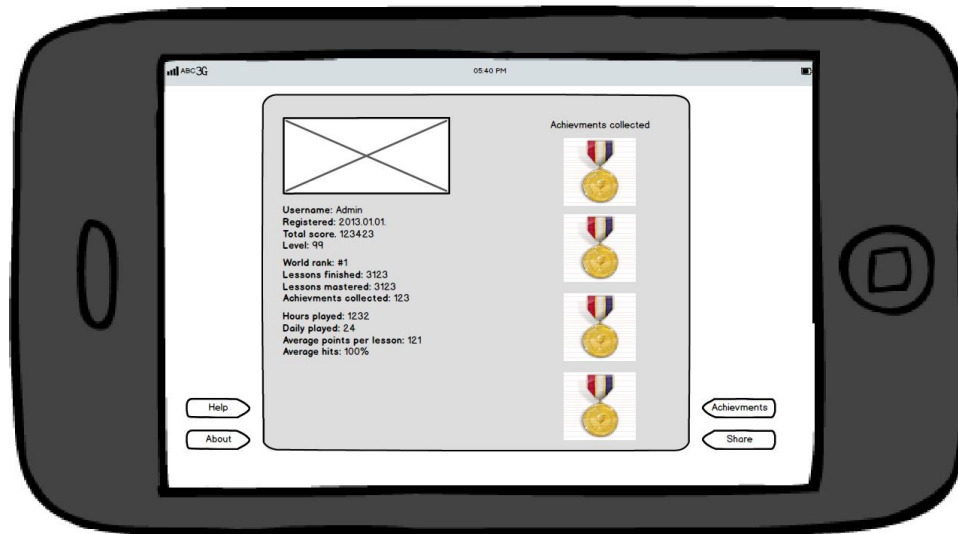
- Megtekintheti a leckéket
- Megtekintheti saját profilját

Az egyéb másodlagos funkciók a képernyő két szélén érhetőek el, ezek:

- Súly
- Információ a programról
- Kitüntetések (Achievements) megtekintése
- Megosztás valamely közösségi médiumon



### 5.3.2 Profil

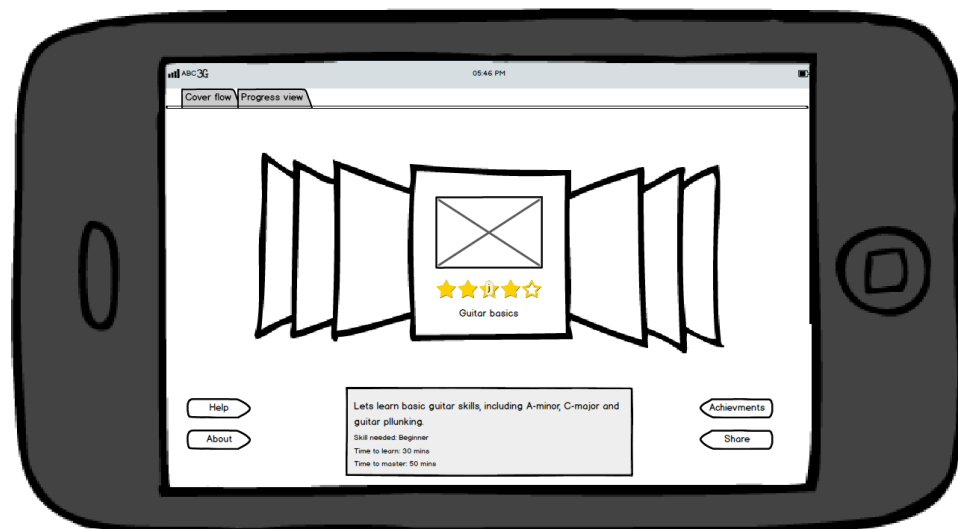


10. ábra: Profil

Az előző képernyőn a profilra kattintva juthatunk ide. Itt a felhasználó már elért eredményei láthatóak: megszerzett pontok, szint, kitüntetések, átlagos teljesítmény. Ez elsősorban a felhasználó számára egy összegzés, amellyel röviden áttekintheti saját eredményeit. Ezt a képernyő oldalán lévő gombokkal akár meg is oszthatja barátaival.

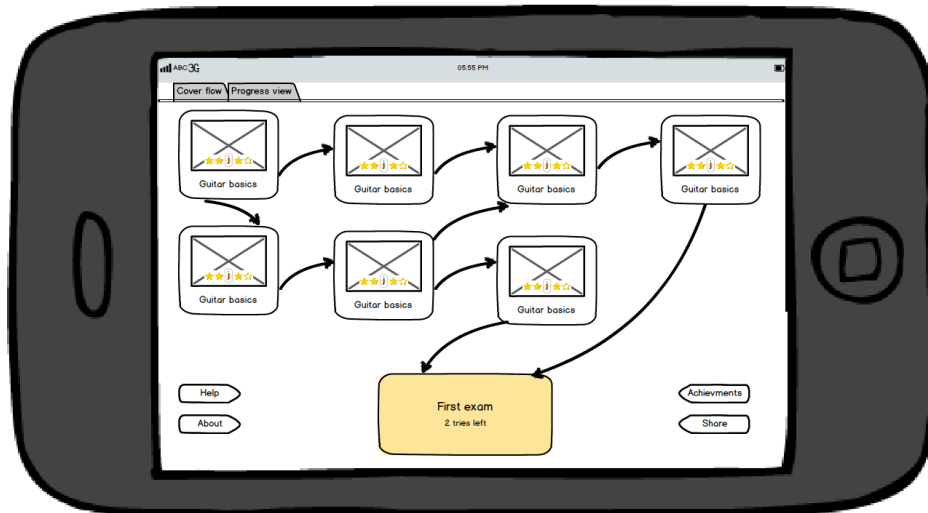
### 5.3.3 Leckék felsorolása

A program kétféle leckenézetet tartalmaz, az első nézet mintegy borítóként tartalmazza az elérhető leckéket, melyek között a felhasználó könnyen lapozhat. Lapozás közben feltűnik a lecke neve, az elért eredménye (ha a lecke teljesített) és egy rövid leírás.



11. ábra: Leckék listázása

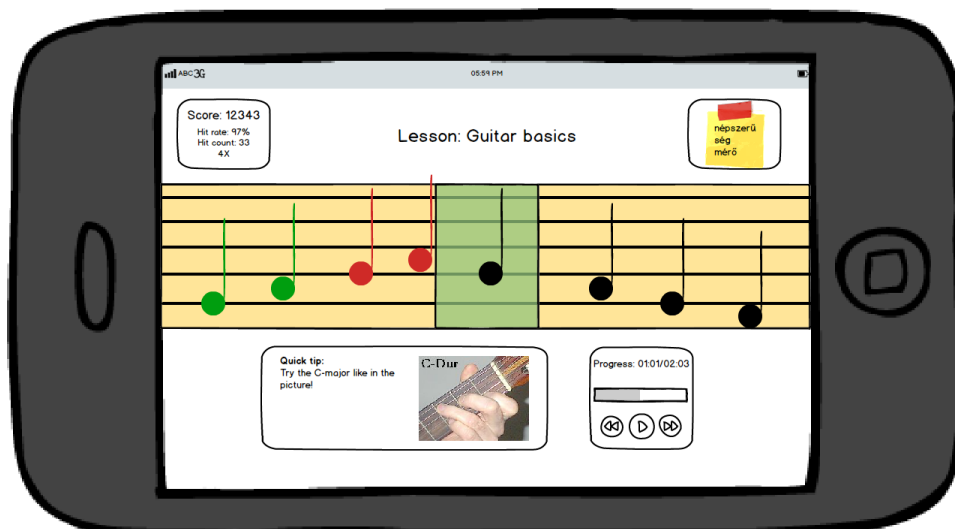
A másik nézet a leckéket tantervi hálóba rendezve jeleníti meg, így pontosan áttekinthető és megtervezhető az előrehaladás. Bizonyos számú lecke után, egy mérföldkőhöz érkezve a felhasználó számot adhat tudásáról, és csak ez után folytathatja a későbbi leckékkel. A tantervi hálóban egyszerre nagyjából egy mérföldkőhöz szükséges leckék látszanak, így ez mintegy modulokra osztja a teljes tanulási folyamatot.



12. ábra: Tantervi háló áttekintése

A két nézet között fülekkel válthatunk, egy leckét kiválasztva a gyakorló képernyőhöz jutunk.

### 5.3.4 Gyakorló képernyő



13. ábra: Gyakorló képernyő

A gyakorló képernyő a program lelke. A lecke elindítása után a kottán lévő hangjegyek jobbról-balra áthaladnak a zöld sávon. A cél az adott hangszeren a jelölt

hang megszólaltatása. A megszólaltatott hangot a MIDI-eszköz továbbítja a programnak, ami kiértékeli, és jelzi, hogy helyes volt-e az adott hang, jó volt-e az időzítés stb. A már lejátszott hangokat attól függően, hogy helyesen vagy helytelenül teljesítették a program zöldre ill. pirosra festi (elképzeltető még a rontás mértéke szerint változó színezés is). A képernyő bal felső sarka közben folyamatosan jegyzi a hangok lejátszásával szerzett pontokat, azt, hogy hány százalékban volt pontos a felhasználó, illetve a megszakítás nélkül hibátlanul lejátszott sorozat hosszát.

A képernyő jobb felső sarka egy mércét tartalmaz, ami jelzi, hogy a képességeinkhez (elért pontok, teljesített leckék) mérten mennyire jó az adott teljesítmény (pl. ha egy gyakorlott játékos egy alapgyakorlat során hibázik, az alacsony minősítést von maga után, ha egy kezdő viszont meglepően jól játszik egy nehéz darabot, akkor nagyon jó értéket mutat a mérce).

A képernyő alsó részén megállíthatjuk a gyakorlást, ill. láthatjuk, hogy hol tartunk benne. Mellette egy interaktív sűgő, ami a leckéhez tartozó elméleti ismereteket adja át „tippek” segítségével.

A gyakorlás végeztével a kotta „visszatekerhető” és kielemezhetőek például a gyakran elhibázott kombinációk. A gyakorlás újratekinthető, ill. a megfelelő eredmény elérése esetén a következő lecke elkezdhető.

Az agyi aktivitás vizsgálata a gyakorlás során szintén folyamatos. Így megoldható, hogy a gyakorlás sebessége a felhasználóhoz igazodjon. Ha a felhasználó különösebb probléma nélkül teljesíti adott hangsor lejátszását, a ritmus gyorsítható, viszont ha a felhasználó szenved, lassítható (vagy egyéb segítséget kaphat, pl. metronóm bekapcsolható).

A gyakorlás után a felhasználó nemcsak az lejátszott hangjait, de folyamatos agyi visszajelzéseit is visszanezheti, így akár azonosíthatók azon pontok a kottán, melyek lejátszása a legnagyobb megterhelést jelenti.

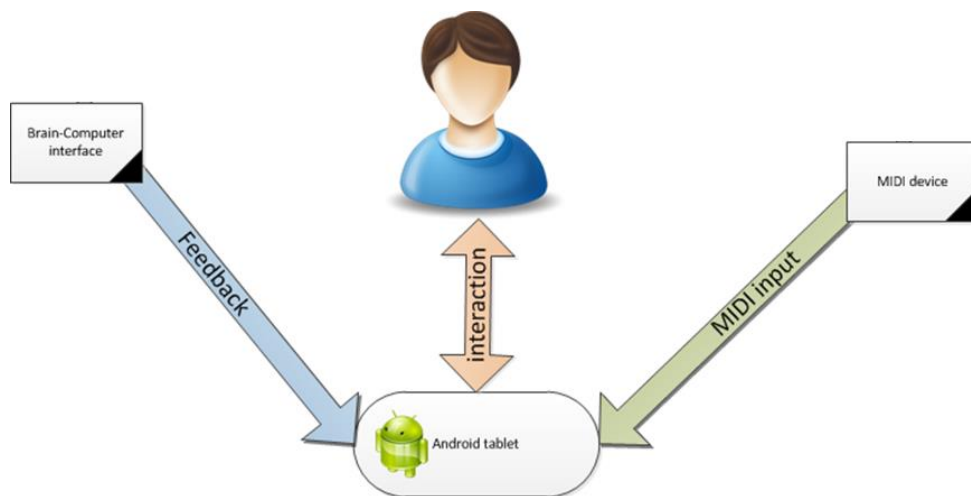
## 5.4 Architektúrális terv

A szoftver a 14. ábra: szerinti felépítést követi, lényegében ez definiálja az alkalmazás általános környezetét. Három fő komponenssel rendelkezik:

- A Brain-Computer interfész itt most az agyi aktivitást mérő és a szoftver számára értelmezhető adattá alakító komponenst jelöli, nem csupán magát a BCI eszközt.
- A MIDI eszköz a játékhoz csatlakoztatott MIDI vezérlő, mely a zenével kapcsolatos bemeneti jeleket szolgáltatja.
- Az Android tablet futtatja magát az alkalmazást

A negyedik szereplő a felhasználó: ő az, aki kapcsolatba lép a szoftverrel. A felhasználó ebben a kontextusban csak, mint a programot használó személy jelenik meg, de mindenképp megemlítenőd, hogy ő az, akit a BCI monitoroz és ő az, aki a MIDI eszközt is használja. Az alkalmazás szempontjából ezek egy komponens formájában jelennek majd meg, mely komponens az adott eszközzel való kommunikációt vezérli.

A továbbiakban a szoftver környezetén felül bemutatom az alkalmazás általános architektúráját. Ez a bemutatás még mellőzi az Android specifikus osztályok megjelenítését, csak egy általános képet nyújt a szoftver fejlesztésekor követendő fejlesztési irányelvekről.



14. ábra: A szoftver és környezete

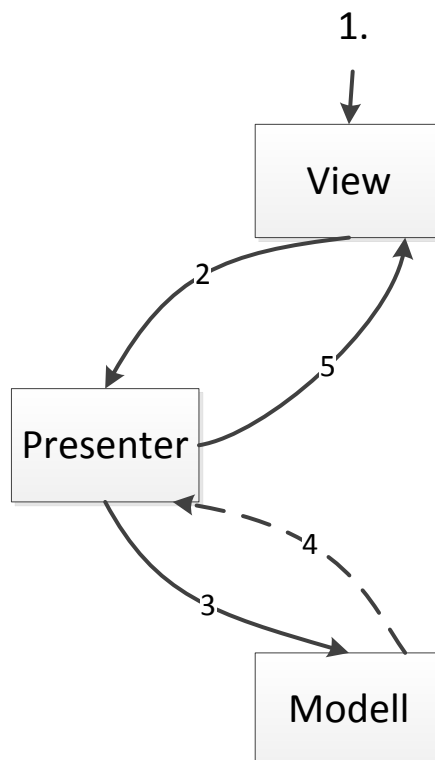
Az alkalmazás tehát az MVC mintát követi. Az MVC lényege, hogy az interaktív alkalmazást három részre tagolja [13]:

- A Modell tartalmazza a fő funkcionalitást és az alkalmazás által használt adatokat.
- A Controller kezeli a felhasználói inputot, és reagál rá a megfelelő akció kiváltásával.
- A View feladata a felhasználói interfész megjelenítése.

A mintát széles körben használják különböző szoftverek fejlesztése közben, és számos változata alakult ki az idők során. Az Android rendszereken megszokott konvencióknak megfelelően az alkalmazás az MVC egy változatát, a Model-View-Presenter (MVP) mintát használja [14]. Ez az MVC esetén már tárgyalt komponenseket használja, a különbség a komponensek kommunikációjában rejlik. A könnyebb érthetőség érdekében, lássuk, hogy egy MVP-t használó alkalmazás hogyan reagál a felhasználói interakciókra:

1. A felhasználó interakcióba lép az alkalmazással
2. A View továbbítja a hívást a Presenter számára
3. A Presenter elvégzi az esetleges módosításokat a Modellen
4. A Modell elvégzi a változtatásokat és egy callback metóduson keresztül, tehát aszinkron módon értesíti a Presenter komponenst
5. A Presenter frissíti a View állapotát

Az egyes lépéseket mutatja be a 15. ábra: .



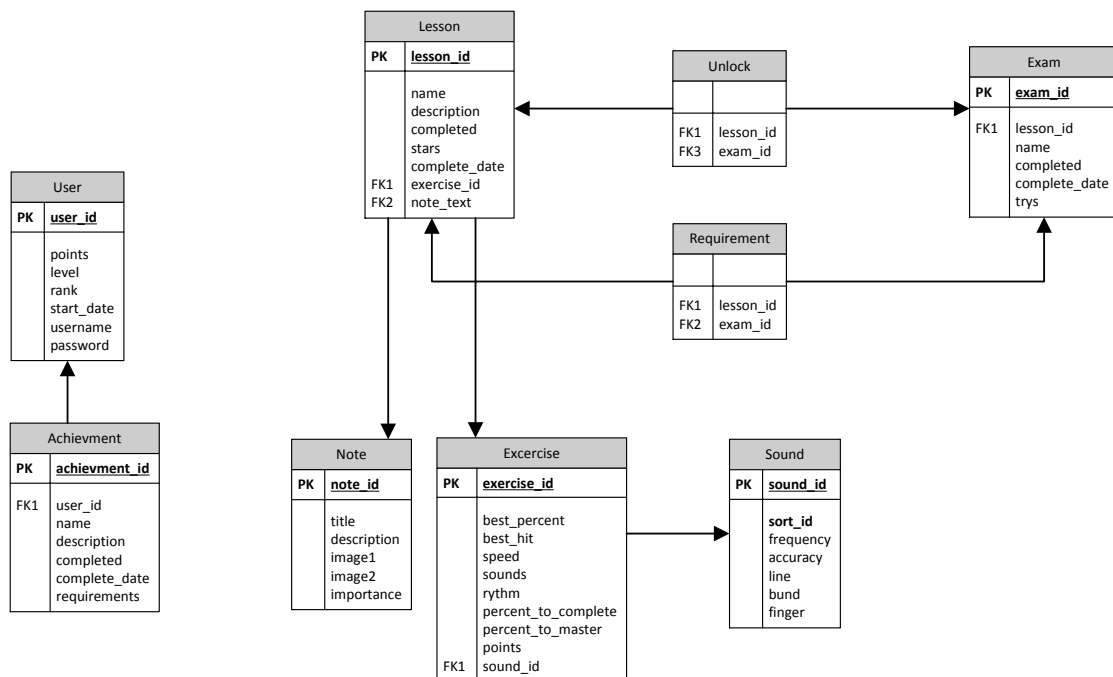
15. ábra: Az MVP minta működése

## 5.5 Adatmodell tervezése

Az alkalmazás a működéséhez szükséges adatokat egy adatbázisban tárolja. Mind az adatbázis, mind az adatbázis réteg felett húzódó absztrakciós réteg valamely adatmodell szerint van szervezve. Az adatmodellt a 16. ábra: szemlélteti egy entitás-relációs diagramon.

Az alábbiakban röviden bemutatom az egyes entitások szerepét:

- **User:** felhasználóspecifikus adatokat tárol.
- **Achievement:** A felhasználó által megszerezhető achievementeket tárolja.
- **Lesson:** Az egyes leckék anyagát tárolja.
- **Note:** A leckékhez fűzhető elméleti segítségek, tippek.
- **Excercise:** Egy gyakorlattal kapcsolatos adatokat tárol.
- **Sound:** A gyakorlat során lejátszandó hangok.
- **Exam:** vizsgák adatai.
- **Unlock:** Egy vizsga elvégzés milyen leckéket szabadít fel.
- **Requirement:** Mely vizsgának mik az előkövetelményei.



16. ábra: Az alkalmazás Entitás-Relációs diagramja

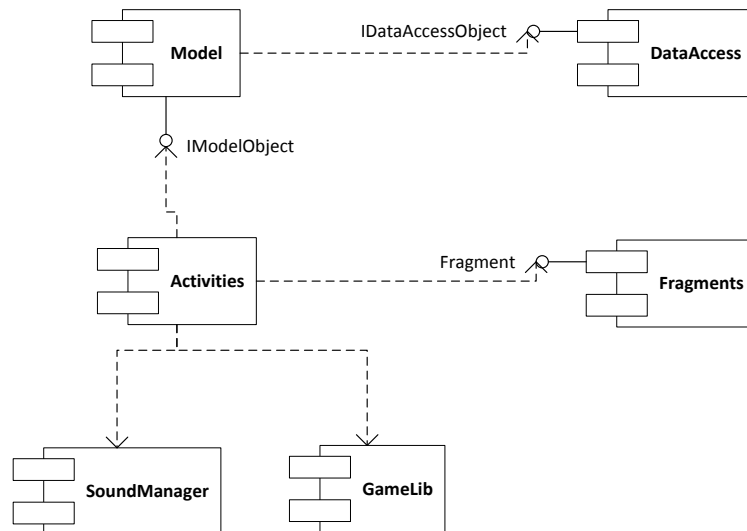
## 5.6 Az alkalmazás megvalósítása

Ebben a fejezetben az alkalmazás konkrét, Android platformon történő megvalósításáról lesz szó. Mivel ez a terület teljes mélységben való ismertetése meghaladná a dolgozat kereteit, csak a lényeges dolgok és az esetleges érdekes implementációs kérdések tárgyalására szorítkozik a fejezet.

### 5.6.1 Az alkalmazás struktúrája

Az előzőekben már említettem, hogy az alkalmazás az MVP architektúrát követi. A következőkben részletesebb képet kaphatunk az alkalmazás struktúrájáról. Ezen a részen már megjelennek az Android platform jellemzői, és bizonyos, a platformhoz köthető osztályok. Ilyenek pl. az Activity és a Fragment. Előbbi az Android alkalmazások építőköve, mely minden alkalmazásban megtalálható, utóbbi az alkalmazás felületének rugalmasabb tagolásában segít [17][14]. Ezen felül megjelennek olyan segédosztályok is, amelyek pl. az Android beépített relációs adatbázis megoldásának könnyebb használatát segíti elő. Ez a módszer szintén egy széles körben használt megoldás a platformon.

Az alkalmazás struktúrája jól láthatóan komponens alapú felépítést követ. Így az alapvetően azonos célt szolgáló osztályok azonos komponensbe kerülhetnek, a jelentősen eltérő szerepűek pedig külön komponensbe. Ez a megközelítés nagyban segíti a későbbi könnyű bővíthetőséget és hordozhatóságot, de átláthatóbbá is teszi a teljes struktúrát.



17. ábra: Az alkalmazás struktúrája

Az alábbi felsorolás a fenti ábra komponenseinek hatáskörét ismerteti, a későbbiekben az implementációs szempontból érdekes részek külön ismertetésre kerülnek:

- **Model:** Az MVP architektúrában a Model szerepét betöltő komponens.
- **DataAccess:** Absztrakciós réteg az Android beépített SQLite adatbázisa felett. Data Access Objecteket tartalmaz.
- **Activities:** Az MVP architektúra Presenter komponense. Az Android platformon használt Activity őosztályból leszármaztatott osztályok.
- **Fragments:** Szintén Android specifikus osztályok, a képernyő egy részének kezelésére.
- **SoundManager:** A MIDI eszközök kezelését végző osztályok.
- **GameLib:** A BCI kezelését végző osztályok.

### 5.6.2 A SoundManager komponens

A SoundManager komponens végzi a MIDI eszközök kezelését az alkalmazás használata során. Mivel a MIDI szabvány támogatásának teljes implementálása nem kis feladat, jelen projekten belül is egy meglévő implementációra támaszkodtam [15]. Ez lényegében a java világból jól ismert javax.sound.midi [16] osztályaihoz hasonló MIDI implementáció.

Képességei miatt tökéletesen segíti a projekt során elkészítendő alkalmazást:

- minden eszközön fut, nem szükséges rendszergazdai jog (root) a használatához
- támogatja a MIDI hangszereket
- nyílt forráskódú, így igény esetén testre szabható

Az alkalmazás szempontjából releváns információk a támogatott MIDI üzenetek és típusaik. Ezt szerencsére a [15] alatt található dokumentáció jól összefoglalja, és arról is információt nyújt, hogy az adott üzenettípus megfelelően tesztelve volt-e.

MIDI üzenettípus	Jelentés	Tesztelve?
MidiMiscellaneousFunctionCodes	Kiterjesztésre fenntartva	NEM
MidiCableEvents	Kábel események	NEM
MidiSystemCommonMessage	Rendszerüzenetek	NEM
MidiSystemExclusive	Exkluzív üzenetek	IGEN
<b>MidiNoteOff</b>	<b>Hang vége</b>	<b>IGEN</b>
<b>MidiNoteOn</b>	<b>Hang kezdete</b>	<b>IGEN</b>
<b>MidiPolyphonicAftertouch</b>	<b>Hang lenyomás ereje</b>	<b>NEM</b>
MidiControlChange	Kontrol üzenet	IGEN
MidiProgramChange	Kontrol üzenet	IGEN
MidiChannelAftertouch	Teljes csatorna „ereje”	NEM
<b>MidiPitchWheel</b>	<b>Hajlítás</b>	<b>IGEN</b>

1. táblázat: A könyvtár által támogatott üzenetek, jelentésük és állapotuk a dokumentáció szerint. Vastaggal kiemelve a program által használt üzenetek.<sup>6</sup>

### 5.6.3 A GameLib komponens

Ahogy korábban már láttuk, a BCI-ből jövő adatokat nem közvetlen az alkalmazás, hanem egy külön erre a célra kifejlesztett szoftver fogadja. Ez a szoftver a háttérben fut, és folyamatosan kommunikál a hozzá csatlakoztatott BCI eszközzel.

Az adatokat használni kívánó többi alkalmazás ezen a szoftveren keresztül kérhet jelzést az egyes eseményekről. Ezek az események már nem nyers adatként, hanem valamilyen értelmezett utasításként kerülnek el a vezérelt alkalmazáshoz.

A GameLib komponens feladata, hogy kommunikáljon a BCI kezelő szoftverrel, és megszerezze az alkalmazás használata során keletkező, agyi aktivitást indikáló eseményeket. A kommunikáció IPC alapú, mely magas szintű objektumokkal van elfedve, így könnyítve a használatot. A GameLib használatához a játék kezdetén fel kell iratkoznunk, hogy innentől szeretnénk fogadni az egyes eseményeket. Ettől kezdve, ha az esemény, amire feliratkoztunk bekövetkezik, a BCI kezelő jelezni fog az saját alkalmazás számára egy callback függvénnyel. Ezt aztán az alkalmazás szempontjából megfelelően kezelhetjük.

<sup>6</sup> Forrás: [15]



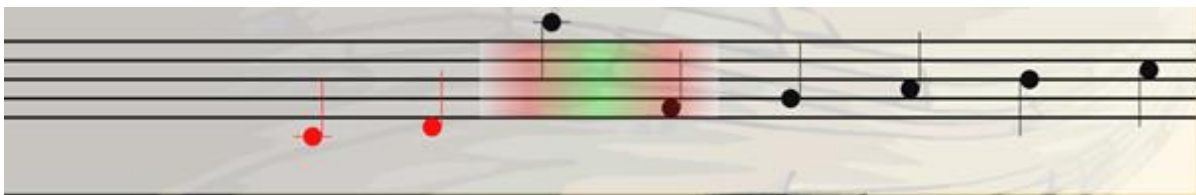
Mivel a BCI kezelő keretrendszer, és ezzel együtt a GameLib fejlesztése egy másik dolgozat témája, mélyebb ismertetésétől eltekintek.

#### 5.6.4 Egyéb, implementáció során felmerült problémák

Ebben a részben az implementáció során felmerült azon érdekességeket említem meg, melyek valamiért kiemelt figyelmet kaptak. Ezek többnyire Android specifikus kérdések, többségük a felhasználói felület egyedisége miatt merült fel problémaként. Ilyen például a külön kotta megjelenítő, a képernyő sarkaiban megjelenő animált teljesítménymérők, a leckék animált megjelenítését szolgáló, albumborító hatását keltő lapozó. Ezen komponensek mind egyedileg, az alkalmazás igényeihez szabva kerültek elkészítésre.

##### MidiVisualizer

Ezt a nevet kapta a kotta megjelenítéséért felelős külön vezérlőelem. A feladat az, hogy egy előre megadott hangjegy halmazon kívül az éppen lejátszott MIDI hang is megjeleníthető legyen egyazon kottán, különböző színekkel. A problémát tovább nehezítette, hogy a kottának animáltan, jobbról-balra mozogva kell megjelennie, igény szerint állítható tempóban.



18. ábra: A saját kotta megjelenítő.

Végül a megoldás egy saját nézet osztály implementálása lett. A képességei a következők:

- Egy előre megadható hangjegy halmaz és ütem alapján, a *start()* metódus hatására elindul a kotta automatikus görgetése.
- Ha az ütem menet közben változik, a megjelenítés újraszámolja a kotta helyzetét, átszámolja a hangjegyek közti helyet
- Egy interfészen keresztül bevezethető egy MIDI Note eseményeket generáló objektum. A megjelenítő a **Note On** és **Note Off** üzenetek alapján kirajzolja a lejátszott MIDI hangot.
- Megadható ütközési zóna: ha ezen a zónán belül egy éppen lejátszott és egy éppen leütött hangjegy ütközik, a hangjegy zöld lesz, különben piros.
- Egy callback interfészen keresztül kivezethető az ütközési zónában történt esemény, különböző paraméterekkel pl.: pontosság.

Az így elkészített saját View vezérlő bármikor könnyen hozzáadható az alkalmazás felhasználói felületéhez, és legtöbb paramétere az Android felületek leírására használt XML jelöléssel is állítható. A vezérlő kinézete a 18. ábra: látható.

## FragmentFlow

A FragmentFlow a jól ismert CoverFlow alapján készült vezérlőelem. A lényege, hogy Fragmenteket jelenít meg CoverFlow elrendezésben. Erre a leckeválasztó felületen van szükség. A leckék kiválasztása ugyanis egyszerű ilyen módon, de sajnos a hagyományos CoverFlow képességeinél magasabb elvárás is volt. Végül a legegyszerűbb megoldás az lett, hogy a megjelenített elemek külön Fragmentekek, melyek tetszőleges felületet nyújthatnak. A vezérlő kezelésére külön Adapter osztály is készült.



19. ábra: A FragmentFlow kinézete.

A vezérlő, mivel Fragmentekek kezelésére képes, nagyon sokszínűen és könnyen használható. A 19. ábra: egy példát mutat a használatára.

## MeterView

A MeterView a felhasználó teljesítményét visszajelző vezérlő a képernyő sarkán. Erre azért van szükség, mert az animált teljesítmény visszajelzés sokkalta hatékonyabb, és a felhasználó figyelmét is jobban vonzza az elem. A fejlesztett vezérlő gyakorlatilag egy mérő-óra, amely különböző mértékeket tud megjeleníteni, animált formában. Ez a P300 jel vizsgálatához szükséges azonnali jutalom feedbacket valósítja meg

## Előkövetelmények

Az előkövetelmények megjelenítésére is szükség volt egy hatékony módszerre, amely a tabletek gyorsan változó orientációját is követni képes. Erre ugyan külön vezérlő nem került kifejlesztésre, de az Android meglévő eszköztárát felhasználva ez sikerült: a megoldás egy ListView, egyedi Adapterrel, ami szükség szerint több-kevesebb elemet helyez egy sorban, és az összefüggő elemeket összeköti.



20. ábra: Az előkövetelmények megjelenése a gyakorlatok közt.

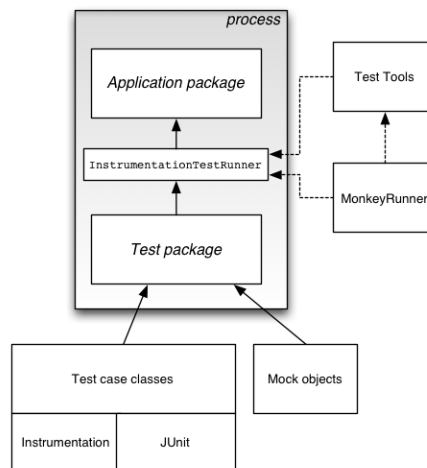
## 5.7 Az alkalmazás tesztelése

Az alkalmazás életciklusa során elengedhetetlen az elkészült komponensek és a komponensek együttműködésének folyamatos tesztelése. Ennek célja, hogy az elkészült program rejtett hibáit felszínre hozza, és felfedje az esetleges hiányosságokat.

Ezen felül, a készülő alkalmazás hatékonyságát is vizsgálni kell. Ugyanis hiába működik tökéletesen a szoftver, ha nincs gyakorlati haszna. Ennek elkerülése érdekében a program pedagógiai hatékonyságát is mérni kell valahogy. A fejezet második fejezetben ennek mikéntjéről is lesz szó.

### 5.7.1 Általános tesztelés

Az alkalmazás tesztelése a megszokott módon, a szoftver életciklusának legtöbb fázisában megtalálható. A szoftvertechnológiában elterjedt tesztelési folyamatok természetesen Android platformon is elérhetőek. A 21. ábra: szemlélteti a keretrendszer által nyújtott lehetőségeket.



21. ábra: Az Android platform által nyújtott tesztelési lehetőségek.<sup>7</sup>

Ez alapján a készülő komponenseket JUnit segítségével lehet tesztelni, ami alacsony szintű unit tesztek elvégzését segíti elő. Ezek segítségével az egyes moduloktól elvárt működést ellenőrizhetjük, az egyes erőforrásokat, környezeti függőségeket pedig ún. Mock objektumokkal helyettesíthetjük[18].

A tesztelt modulok együttműködésének helyességét aztán különböző módokon ellenőrizhetjük. A MonkeyRunner például az emulátoron elindított alkalmazást veti alá egyfajta stressz tesztelésnek: felhasználói eseményeket (pl. képernyő érintés stb.) generál, beállítástól függően akár több ezret.

<sup>7</sup> A képe forrása: [http://developer.android.com/tools/testing/testing\\_android.html](http://developer.android.com/tools/testing/testing_android.html)

### 5.7.2 A program hatékonyságának tesztelése

A helyes működés és funkcionalitás tesztelésén túl szükség van egyéb, nehezebben mérhető tulajdonságok ellenőrzésére is. Noha jelen szoftver esetében ezek a tesztek még jövőbe mutató elképzelések, mindenképpen szükségesnek érzem megemlíteni őket. Az elkészült szoftver esetén ilyenek lehetnek például a vizsgálandó tulajdonságok:

- használhatóság
- pedagógiai hatékonyság

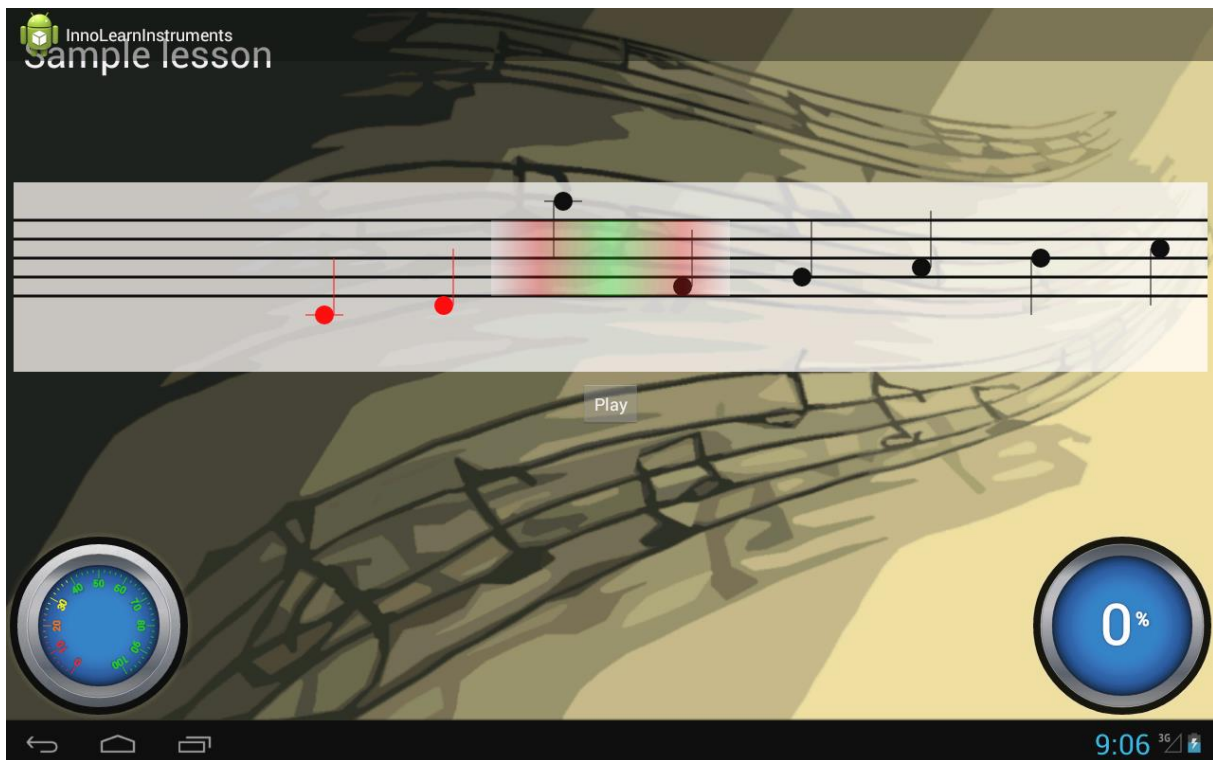
A használhatóság alatt itt az értem, hogy a szoftver egy átlagos felhasználó számára könnyen és egyértelműen használható. Napjainkban, különösen az okostelefon alkalmazások körében, a felhasználók számára ez az egyik legfontosabb tulajdonság. Ha a szoftver felülete nem ergonomikus, nem sokan fogják használni azt. Éppen ezért mára már komoly technikák segítik a szoftverek használhatósági tesztelését[18].

A tanulási hatékonyság mindenképpen a szoftver legfontosabb tulajdonsága. Éppen ezért a Zeneakadémiával együttműködve folyik a program zenepedagógiai ellenőrzése.

## 6 Összefoglalás

### 6.1 Eredmények

A projekt eredményeként létrejött alkalmazás a célként kitűzött követelményeknek teljes mértékben megfelel. Az alkalmazás képes tetszőleges MIDI hangszer kezelésére és a felhasználó a hangszerrel leckéket ill. leckéken belüli gyakorlatokat teljesít. A játék közben a felhasználó valós idejű visszacsatolást kap eredményéről, és az adott gyakorlat ezek alapján adaptívan változik.

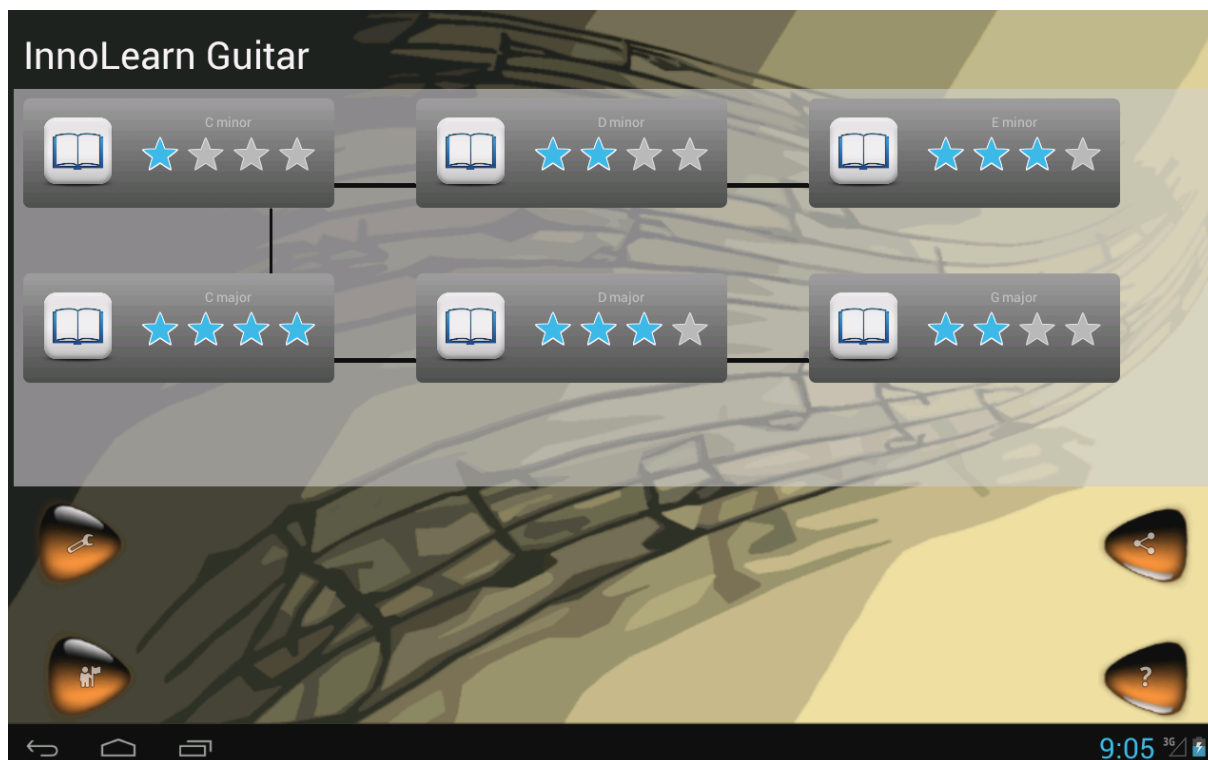


22. ábra: A megvalósított alkalmazás gyakorló felülete.

Az alkalmazás a feedbacken kívül megvalósít számos, tanulási kedvet növelő funkciót is, így a felhasználók a tanulási folyamat során valódi játékelményel találkozhatnak. A főbb interaktív elemek a következők:

- achievementek
- toplista a barátokkal
- a gyakorlatok és a leckék egy értékelési rendszer alapján értékeltek
- játék során interaktív animációk és hanghatások kísérik az egyes eseményeket
- a játék során a pillanatnyi teljesítményt különböző kijelzők színesítik

Összességében az alkalmazás képes a felhasználóknak új tudás átadására, olyan módon, hogy közben a játékos motivációja megmarad, a versenyszelleme egyre jobb eredmények elérésére készíti és a tanulást egy játékos folyamat részeként valósítja meg.



23. ábra: A gyakorlatválasztó, az előkövetelményekkel.

## 6.2 Továbbfejlesztési lehetőségek

Habár a játék a kitűzött célokat megvalósítja, a tanulás nem jöhet létre a megfelelő tudásanyag nélkül. Későbbi cél, hogy a játék tartalmát egy szakemberek által összeállított csomag adja, így a hatékony tanulási folyamat bizonyítottan hatásos tartalommal ötvöződne.

További érdekes lehetőség lenne a szoftver kibővítése úgy, hogy egyszerre több hangszert támogasson. Ezzel lehetőség nyílna csoportos gyakorlásra, amely egy zenekari próba hangulatát idézné. Ennek a bővítésnek az első állomása a további hangszertípusok specifikus támogatása, mivel a jelenlegi verzió ugyan minden MIDI-eszközzel együttműködik, de a felhasználói felület és a leckék felépítése főleg billentyűs és húros hangszereknek kedvez.

Egy saját leckeszerkesztő készítése a haladóbb felhasználóknak kedvezne: így kiélhetnék a kreatív vágyaikat, és az elkészült saját leckéket barátaikkal megosztva bezsebelhetnék az elismerést. Ez szintén a játékedv és a motiváció növekedésével járna, ami szintén elősegíti a hatékonyabb tanulást.



## Irodalomjegyzék

- [1] Clark, Ruth C., and Richard E. Mayer. E-learning and the science of instruction: Proven guidelines for consumers and designers of multimedia learning. Wiley.com, 2011.
- [2] Meinou H. de Vries, Catrin Ulte, Pienie Zwitserlood, Barbara Szymanski, Stefan Knecht, Increasing dopamine levels in the brain improves feedback-based procedural learning in healthy participants: An artificial-grammar-learning experiment, *Neuropsychologia*, Volume 48, Issue 11, September 2010, Pages 3193-3197, ISSN 0028-3932, <http://dx.doi.org/10.1016/j.neuropsychologia.2010.06.024>.
- [3] Acey Kreisler Boyce, Antoine Campbell, Shaun Pickford, Dustin Culler, and Tiffany Barnes. 2011. Experimental evaluation of BeadLoom game: how adding game elements to an educational tool improves motivation and learning. In *Proceedings of the 16th annual joint conference on Innovation and technology in computer science education (ITiCSE '11)*. ACM, New York, NY, USA, 243-247. DOI=10.1145/1999747.1999816 <http://doi.acm.org/10.1145/1999747.1999816>
- [4] Acey Boyce, Katelyn Doran, Antoine Campbell, Shaun Pickford, Dustin Culler, and Tiffany Barnes. 2011. BeadLoom Game: adding competitive, user generated, and social features to increase motivation. In *Proceedings of the 6th International Conference on Foundations of Digital Games (FDG '11)*. ACM, New York, NY, USA, 139-146. DOI=10.1145/2159365.2159384 <http://doi.acm.org/10.1145/2159365.2159384>
- [5] Michael Eagle and Tiffany Barnes. 2009. Experimental evaluation of an educational game for improved learning in introductory computing. *SIGCSE Bull.* 41, 1 (March 2009), 321-325. DOI=10.1145/1539024.1508980 <http://doi.acm.org/10.1145/1539024.1508980>
- [6] Luis von Ahn. 2013. Duolingo: learn a language for free while helping to translate the web. In *Proceedings of the 2013 international conference on Intelligent user interfaces (IUI '13)*. ACM, New York, NY, USA, 1-2. DOI=10.1145/2449396.2449398 <http://doi.acm.org/10.1145/2449396.2449398>
- [7] Gerényi Gábor, Sík Zoltán, *Midi – Alapozás*, Pixel Graphics Kft, 1992
- [8] Wolpaw, Jonathan R., et al. "Brain-computer interface technology: a review of the first international meeting." *IEEE transactions on rehabilitation engineering* 8.2 (2000): 164-173.
- [9] Vallabhaneni, Anirudh, Tao Wang, and Bin He. "Brain—Computer Interface." *Neural engineering*. Springer US, 2005. 85-121.
- [10] Berger, Hans. *Über das Elektrenkephalogramm des Menschen*, Volume 106, Pages 577-584, ISSN 0003-9373, <http://dx.doi.org/10.1007/BF01987726>
- [11] Vidal, J.J., "Real-time detection of brain events in EEG," *Proceedings of the IEEE* , vol.65, no.5, pp.633,641, May 1977 doi: 10.1109/PROC.1977.10542

- [12] Mason, S.G.; Birch, G.E., "A general framework for brain-computer interface design," *Neural Systems and Rehabilitation Engineering*, IEEE Transactions on , vol.11, no.1, pp.70,85, March 2003 doi: 10.1109/TNSRE.2003.810426.
- [13] Meunier, Regine, et al. "Pattern-Oriented Software Architecture, Volume 1: A System of Patterns." (1996).
- [14] Meier, Reto. *Professional Android 4 application development*. John Wiley & Sons, 2012.
- [15] Kaoru Shoji. *USB MIDI Driver for Android 3.1 or later*,  
<https://github.com/kshoji/USB-MIDI-Driver>
- [16] Package documentation: javax.sound.midi,  
<http://docs.oracle.com/javase/7/docs/api/javax/sound/midi/package-summary.html>
- [17] Ekler Péter, Fehér Marcell, Forstner Bertalan, and Kelényi Imre. *Android-alapú szoftverfejlesztés*. SZAK Kiadó, 2012.
- [18] Myers, Glenford J., Corey Sandler, and Tom Badgett. *The art of software testing*. John Wiley & Sons, 2011.



## A. Függelék

<b>EEG</b>	agyi tevékenység mérésére szolgáló módszer
<b>hangszer</b>	Általános értelemben zenélésre használt eszköz, a most azonban a kiválasztott hangszert értjük alatta.
<b>lecke</b>	Egy anyagrészt tartalmazó egység, melynek elvégzése közelebb visz az adott hangszer elsajátításához.
<b>értékelés</b>	A leckére kapható jutalom. Lásd „csillag”.
<b>szabad gyakorlás</b>	Nem egy komplett leckéből, hanem szabadon választott témából gyakorló feladatok megoldása. A felhasználó csak <b>pontokat</b> kaphat érte.
<b>mérföldkő</b>	Olyan speciális lecke, amely előírja bizonyos leckék, lecke csoportok előzetes teljesítését.
<b>pont</b>	A leckék/gyakorlások alatt elért <b>százalékos teljesítmény</b> pontként jóváíródik a felhasználó <b>profilján</b> .
<b>szint</b>	Mivel a felhasználók fejlődése a pontok alapján nehezebben követhető, és nem lineáris ezért a szerzett pontok alapján szintek is megjelennek a profilon.
<b>gyakorló felület</b>	Az a felület ahol a gyakorlás zajlik. Itt történik az agyi és hangszeres visszacsatolások mérése, a gyakorlás befejeztével pedig az értékelés.
<b>agyi visszacsatolás</b>	A felhasználó agytevékenységének mérése adódó információ. Pl.: idegesség, mert a lecke túl nehéz neki, unalom stb.
<b>hangszeres visszacsatolás</b>	A hangszerekből jövő hang mérésén alapszik. Megmutatja, hogy milyen pontos a megszólalt hang.
<b>százalékos teljesítmény</b>	Egy lecke során lejátszott hangok visszacsatolása során egy százalékos pontosságot határozunk meg. Az összes lejátszott hang átlaga adja meg a százalékos teljesítményt.
<b>csillag</b>	A lecke százalékos teljesítménye alapján, ha a lecke sikeres 1-4 csillaggal jutalmazható.
<b>profil</b>	A felhasználó adatainak összefoglaló nézete.
<b>megosztás</b>	A felhasználó megoszthat a profiljáról egy képet különböző közösségi oldalakon.

<b>achievement</b>	Ha a felhasználó bizonyos értékelendő teljesítményt ér el (pl. 100%-ra játszik el egy dalt.) akkor egy kitüntetéssel jutalmazzuk, ami látszik a profilján.
<b>vizsga</b>	Egy mérföldkő akkor teljesített, ha a felhasználó elvégzi a hozzá tartozó vizsgát.