

TARTU ÜLIKOOL  
Arvutiteaduse instituut  
Informaatika õppekava

**Kristiina Kõiv**  
**„Solfedžo õpitarkvara 1. klassile“**  
**Bakalaureusetöö (9 EAP)**

Juhendajad: Alo Peets, Taavi Duvin, Anne Villems

Tartu 2017

## „Solfedžo õpitarkvara 1. klassile“

### **Lühikokkuvõte:**

Käesoleva bakalaureusetöö eesmärk on luua eestikeelne mobiilne rakendus Androidi platvormile, mis muudaks 1. klassi muusikakooli õpilastel nootide ja rütmide õppimise protsessi huvitavamaks ning samas ka kergemaks. Töö on jaotatud kolmeks osaks. Esimeses peatükis koostatakse solfedžo õpitarkvara lühianalüüs. Teises peatükis tutvustatakse lugejale, kuidas on võimalik tuvastada klaveril mängitud heli Android seadmega, kasutades FFT algoritmi. Töö kolmandas osas antakse ülevaade valminud rakendusest ning koostatakse kasutusjuhend tarkvara kasutamiseks.

**Võtmesõnad:** Solfedžo, noodid, rütmid, Android, helitöötlus, FFT

**CERCS:** S281(Arvuti õpiprogrammide kasutamise metoodika ja pedagoogika)

## „Solfeggio learning software for first grade“

### **Abstract:**

The aim of this bachelor thesis is to develop mobile software on Android platform that will make learning process of notes and rythms more intriguing and easier for the first grade students in Estonian music schools. The thesis is divided into three parts. Chapter 1 gives an overview of solfeggio software analysis. Chapter 2 describes how it is possible to recognize note played on piano with Andorid device using FFT algorithm. Chapter 3 gives an overview of developed software and shows how to use it.

**Keywords:** Solfeggio, notes, rythms, Andorid, sound processing, FFT

**CERCS:** S281(Computer-assisted education)

# Sisukord

Sissejuhatus	4
1. Solfedžo õpitarkvara analüüs	5
1.1 Mis on solfedžo?	5
1.2 Solfedžo põhiteadmised	5
1.3 Probleemi püstitus	6
1.3 Funktsionaalsed nõuded	7
1.4 Sarnased lahendused	11
2. Nootide tuvastus Androidil	12
2.1 Digitaalheli salvestamine	12
2.2 Digitaalheli töötlus	13
2.3 FFT realiseering rakenduses	17
3. Ülevaade valminud rakendusest	20
3.1 Kasutatud tehnoloogiad	20
3.2 Rakenduse arhitektuur	21
3.3 Tarkvara kvaliteet ja jõudlustulemused	22
3.4 Kasutajaliides	23
3.5 Kasutusjuhend	23
Kokkuvõte	33
Kasutatud kirjandus	34
Litsents	36

# Sissejuhatus

Tänapäeval on väga paljudel muusikakooli 1. klassi õpilastel raske endale selgeks teha, mis on noodid, rütmid, kuidas kõlavad erinevad noodid ning kuidas teha vahet rütmidel. Kuna nootide tundmine ja rütmidest arusaamine on alus igale pilliõppele ja ka laulmisõppele, siis on väga oluline, et õpilased saaksid juba algusest peale tugeva muusikalise alushariduse. Muusikakoolides on selle jaoks loodud selline õppeaine nagu solfedžo.

IKT ehk info-ja kommunikatsioonitehnoloogia vahendite (sh nutitelefon) kasutamine õppetöös tõstab õpilaste motivatsiooni ja huvi õppeaine vastu ning muudab õpetajatele õpetamise kiiremaks [1]. Hiljuti Tartu Ülikooli haridusteaduste instituudi ja loodusteadusliku hariduse keskuse koostöös läbi viidud uuringust selgus, et 97,4%-l õpilastest on võimalik kasutada kodus õppimiseks nutitelefon ja 86,3%-l on võimalus teha seda ka koolis [2]. Praegu aga pole loodud eestikeelset mobiilset tarkvara Androidi platvormile, mis toetaks õpilaste solfedžoõpinguid ning aitaks lihtsustada õppimisprotsessi.

Käesoleva bakalaureusetöö eesmärk on luua eestikeelne mobiilne rakendus, mis muudaks 1. klassi muusikakooli õpilastel nootide ja rütmide õppimise protsessi huvitavamaks ning kergemaks.

Töö esimeses peatükis antakse ülevaade sellest, mis on solfedžo ning milliseid peamisi mõisteid peab tundma muusikakooli 1. klassi lõpetanud õpilane. Sõnastatakse probleem, mille lahendamiseks hakatakse looma rakendust. Antakse ülevaade rakenduse funktsionaalsetest nõuetest ning sarnastest lahendustest. Teises peatükis tutvustatakse lugejale, kuidas on võimalik tuvastada klaveril mängitud heli Android seadmega, kasutades FFT algoritmi. Töö kolmandas osas antakse ülevaade valminud rakendusest, kasutatud tehnoloogiatest ning koostatakse kasutusjuhend rakenduse kasutamiseks.

# 1. Solfedžo õpitarkvara analüüs

Käesolevas peatükis selgitatakse, mis õppeaine on solfedžo ning milliseid peamisi mõisteid peab tundma muusikakooli esimese klassi lõpetanud õpilane. Sõnastatakse probleem, mille lahendamiseks hakatakse looma rakendust. Samuti antakse ülevaade rakenduse funktsionaalsetest nõuetest ning sarnastest lahendustest.

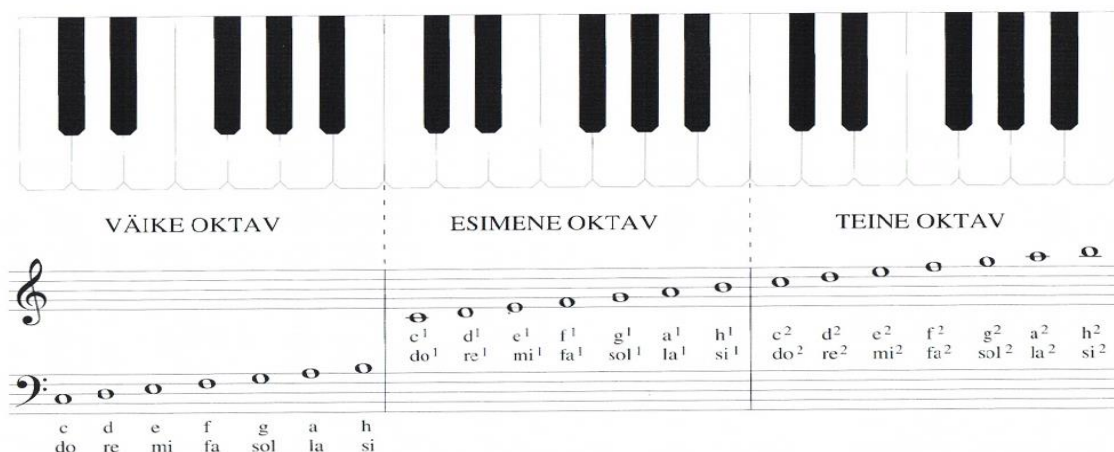
## 1.1 Mis on solfedžo?

Solfedžo on õppeaine muusikakoolides, kus õpitakse noote, rütme, kirjutatakse diktaate, õpitakse intervale, akorde. Lühidalt öeldes - solfedžo annab õpilastele muusikalised põhiteadmised, mis toetavad nende pilliõpinguid [3]. Samuti aitab muusikalise keele ehk noodikirja tundmine õpilastel väljendada enda loomingulist poolt, kirjutades erinevaid heliteoseid.











## 1.2 Solfedžo põhiteadmised

Kõige olulisemad põhiteadmised, mis tuleb solfedžo õppeaines esimeses klassis selgeks saada on järgnevad:

- nootide nimetused - C, D, E, F, G, A, H (või B) või do, re, mi, fa, sol, la, si [4];
- nootide asukoht viiuli- ja bassivõtmes (vt joonis 1);
- rütmifiguurid ehk rütmide tuvastamine pildi järgi (vt joonis 2);
- rütmide tuvastamine kõla järgi. [5]



Joonis 1. Nootide asukoht viiuli ja bassivõtmes koos nimetustega [6].

Pilt	Rütmi nimetus	Löökide arv	Pilt	Rütmi nimetus	Löökide arv
	TA	1 löök		TIRI-TIRI	1 löök
	TA-A	2 lööki		TI-TIRI	1 löök
	TA-A-A	3 lööki		TIRI-TI	1 löök
	TA-A-A-A	4 lööki		TAI-RI	1 löök
	TI-TI	1 löök		TA-I-TI	2 lööki

Joonis 2. Rütmide nimetused ja kestvus.

2016. aasta detsembris ilmus Eesti Muusikakoolide Liidu uus soovituslik solfedžo ainekava Eesti muusikakoolidele, kus on kirjutatud, et uue õppekava eesmärk on vähendada nooremas astmes (I-II klass) kirjalikke ülesandeid ja keskenduda rohkem kuulamisülesannetele ning astmetaju treenimisele [5]. Ka autor on oma rakenduse väljatöötamisel keskendunud muusikalist kuulmist arendavate ülesannete loomisele. Samuti on eesmärk vähendada kirjalike ülesannete mahtu läbi interaktiivsete õppemeetodite.

### 1.3 Probleemi püstitus

Väga paljudel muusikakooli 1. klassi õpilastel, kes pole varem muusikaga kokku puutunud, on alguses raske endale selgeks teha noodi- ja rütmiõpetuse alustalad. Tekib palju segadusi noodinimetustega, nootide paiknemisega noodijoonestikul ning klaveri klaviatuuril, rütminimetustega ja erinevate rütmide äratundmisega. Nii noodi- kui ka rütmiõpetuse õppimise aitaks kergemaks muuta see, kui õpilane saaks ise pidevalt harjutada ning enda teadmisi testida.

Kuna tänapäeval väga paljud noored kasutavad vabal ajal nutitelefoni, siis oleks hea, kui lisaks tunnis kasutatavatele õppevahenditele oleks olemas eestikeelne mobiilirakendus, mis aitaks neil kiiremini muusika alustalad selgeks saada. Samuti aitaks see muuta solfedžotunnid interaktiivsemaks ning õpilane saaks ka ise kodus või vabal ajahetkel harjutada. Kui noodi- ja rütmiõpetuse alustalad jäävad omandamata, siis on õpilasel väga raske edaspidi nii pilliõpingutes

kui ka solfedžo tundides. Sageli võib see viia ka selle olukorrani, kus õpilased kaotavad motivatsiooni ning jätavad õpingud pooleli. Samuti pole hetkel loodud eestikeelset mobiilset tarkvara Androidi platvormile, mis toetaks õpilaste solfedžoõpinguid.

Et muuta õpilaste solfedžoõpingud huvitavamaks ning kergemaks, on käesoleva töö raames loodud Androidi platvormile eestikeelne mobiilirakendus “Solfedžo õpitarkvara 1. klassile”, mis keskendub kahele peamisele õppeteemale - noodiõpetus ja rütmiõpetus. Noodiõpetuse osas saab õpilane õppida noodinimetusi, erinevate nootide kõla, nende paiknemist nii noodijoonestikul kui ka klaveri klaviatuuril, tuvastada klassiruumis oleval klaveril mängitud noote kõla järgi. Rütmiõpetuse osas saab õppida rütmide nimetusi, tuvastada rütme plaksutamise ja pildi järgi. Nii noodi-kui ka rütmiõpetuse kategoorias saab õppida ka mõisteid ning testida teadmisi.

### 1.3 Funktsionaalsed nõuded

Funktsionaalsete nõute koostamisel lähtus autor sellest, et rakendus oleks muusikakooli 1. klassi õpilastele kergesti mõistetav ning võimalikult lihtne kasutada.

#### **Funktsionaalsed nõuded rakenduse avalehele:**

1. Rakenduse avalehel peab olema kaks nuppu “ÕPI NOOTE” ja “ÕPI RÜTME”.
2. Avalehel peab olema paremal nurgas seadete nupp, kust kasutaja saab valida sobivaid noodinimetusi vastavalt solfedžotunnis kasutatavatele noodinimetustele:
  - 2.1. C, D, E, F, G, A, H
  - 2.2. C, D, E, F, G, A, B
  - 2.3. DO, RE, MI, FA, SOL, LA, SI

#### **Funktsionaalsed nõuded kategooriale “ÕPI NOOTE”:**

1. Antud kategooria peab jagunema viieks alamülesandeks:

##### **1.1. “ÕPI NOOTE PILDI JÄRGI” :**

- 1.1.1. Antud ülesande lehel peab olema paremas nurgas nupp “VALI NOODIVÕTPI”, kus kasutaja saab valida kahe valiku vahel - viiulivõti või bassivõti - vastavalt sellele, kas ta soovib arvata noote viiuli- või bassivõtmes. Vaikimisi on see valik alati viiulivõti.
- 1.1.2. Vastavalt eelnevale valikule (1.1.1) kuvatakse noodijoonestikul kasutajale pilt noodist.
- 1.1.3. Kasutajal peab olema võimalik ära arvata noot, vajutades õiget klahvi klaveri klaviatuuril, kusjuures klaveril peab olema 3 oktavit (väike, esimene ja teine oktav)
- 1.1.4. Kui kasutaja vastab õigesti, siis tuleb muuta klaveri klahv roheliseks ning näidata teksti “ÕIGE VASTUS”, vastasel korral muuta klahv punaseks ning näidata teksti “PROOVI VEEL”.
- 1.1.5. Kasutajal on võimalik vajutada nupule “JÄRGMINE” ainult siis, kui ta on õige vastuse valinud.

## **1.2. “ÕPI NOOTE KUULMISE JÄRGI”:**

- 1.2.1. Antud ülesande lehel peab olema paremas nurgas nupp “VALI OKTAV”, kus kasutaja saab valida kolme valiku vahel - esimene, teine või väike oktav - vastavalt sellele, kas ta soovib arvata esimese, teise või väikese oktavi noote kõla järgi. Vaikimisi on see valik alati esimene oktav.
- 1.2.2. Vastavalt eelnevale valikule (1.1.1) peab kasutajal olema võimalik vajutada nupule “PLAY”, mille tulemusena kuuleb kasutaja noodiheli.
- 1.2.3. Kasutajal peab olema võimalik ära arvata noot, vajutades õiget klahvi klaveri klaviatuuril, kusjuures klaveril peab olema 3 oktavit (väike, esimene ja teine oktav).
- 1.2.4. Kasutajal peab olema võimalik kuulata heli mitu korda.
- 1.2.5. Kui kasutaja vastab õigesti, siis tuleb muuta klaveri klahv roheliseks ning näidata teksti “ÕIGE VASTUS”, vastasel korral muuta klahv punaseks ning näidata teksti “PROOVI VEEL”.
- 1.2.6. Kasutajal on võimalik vajutada nupule “JÄRGMINE” ainult siis, kui ta on õige vastuse valinud.



### **1.3. “TUVASTA NOOT KÕLA JÄRGI”:**

- 1.3.1. Antud ülesande lehel peab olema 3 nuppu : “START”, “STOP”, “PLAY”
- 1.3.2. Kui kasutaja vajutab nupule “START”, siis rakendus peab alustama heli salvestamist.
- 1.3.3. Kui kasutaja vajutab nupule “STOP”, siis rakendus peab lõpetama heli salvestamise.
- 1.3.4. Kui kasutaja vajutab nupule “PLAY”, siis pärast heli salvestamist (1.3.1 ja 1.3.2) peab rakendus ära tundma päris klaveril valgetel klahvidel mängitud noodi 3 oktaavi ulatuses (väike, esimene, teine), kuvades nii noodi sageduse kui ka noodinimetuse.

### **1.4. “TEOORIA”**

- 1.4.1. Kasutajal peab olema võimalik õppida põhimõisteid nootide kohta teksti kujul.

### **1.5. “TESTI TEADMISI”**

- 1.5.1. Kasutajal peab olema võimalik lahendada testi, mis koosneb 6-st küsimusest.
- 1.5.2. Testi sooritamise järel näidatakse õigesti ja valesti vastatud küsimuste arvu ning parimat tulemust.

### **Funktsionaalsed nõuded kategooriale “ÕPI RÜTME”:**

- 1. Antud kategooria peab jagunema neljaks alamülesandeks:

#### **1.1. “ÕPI RÜTME PILDI JÄRGI” :**

- 1.1.1. Kasutajale kuvatakse pilt rütmist ning tal peab olema võimalik ära arvata,

mis rütmiga on tegemist, vajutades õigele nimetuse nupule.

- 1.1.2. Kui kasutaja vastab õigesti, siis tuleb muuta vastuse nupp roheliseks ning näidata teksti “ÕIGE VASTUS”, vastasel korral muuta nupp punaseks ning näidata teksti “PROOVI VEEL”.
- 1.1.3. Kasutajal on võimalik vajutada nupule “JÄRGMINE” ainult siis, kui ta on õige vastuse valinud.

## **1.2. “ÕPI RÜTME KUULMISE JÄRGI”:**

- 1.2.1. Kasutajal peab olema võimalik ära arvata rütm kuulmise järgi.
- 1.2.2. Antud ülesande lehel peab olema nupp “PLAY”, mille vajutamisel mängitakse kasutajale rütm ette plaksutuste animatsiooniga.
- 1.2.3. Kui kasutaja vastab õigesti, siis tuleb muuta vastuse nupp roheliseks ning näidata teksti “ÕIGE VASTUS”, vastasel korral muuta nupp punaseks ning näidata teksti “PROOVI VEEL”.
- 1.2.4. Kasutajal on võimalik vajutada nupule “JÄRGMINE” ainult siis, kui ta on õige vastuse valinud.
- 1.2.5. Kasutajal peab olema võimalik kuulata rütmi mitu korda.

## **1.3. “TEOORIA”**

- 1.3.1. Kasutajal peab olema võimalik õppida põhimõisteid nootide kohta teksti kujul.

## **1.4. “TESTI TEADMISI”**

- 1.4.1. Kasutajal peab olema võimalik lahendada testi, mis koosneb 8-st küsimusest.
- 1.4.2. Testi sooritamise järel näidatakse õigesti ja valesti vastatud küsimuste arvu ning parimat tulemust.

## 1.4 Sarnased lahendused

Hetkel pole loodud ühtegi eestikeelset mobiilirakendust Androidi seadmele, mis hõlmaks nii nootide kui ka rütmide õppimist. Inglise keeles on loodud nootide õppimiseks mobiilirakendus, kus kasutajal tuleb pildi järgi ära tunda noodi nimetus. Kuid kuna see rakendus on inglise keelne, siis Eesti algklassi õpilastel on raske selle järgi õppida. Samuti puudub rütmide õppimise osa. Antud rakendus on tasuline ning maksab Google Play Store'is 1,59 eurot [7].

Eesti keeles on valminud veebirakendus solf.ee, kus on kasutajal võimalik õppida noote ning rütme eesti keeles, kuid see kasutajaliides pole väga mobiilisõbralik ning vajab interneti olemasolu [8].

Inglise keeles on välja töötatud mobiilirakendus „Perfect Ear“, mis aitab õppida intervale, helistike äratundmist, rütme ja noote, kuid Eesti õpilastel on algklassides raske inglise keelest aru saada. Samuti tundub, et see rakendus on mõeldud rohkem edasijõudnutele [9] .

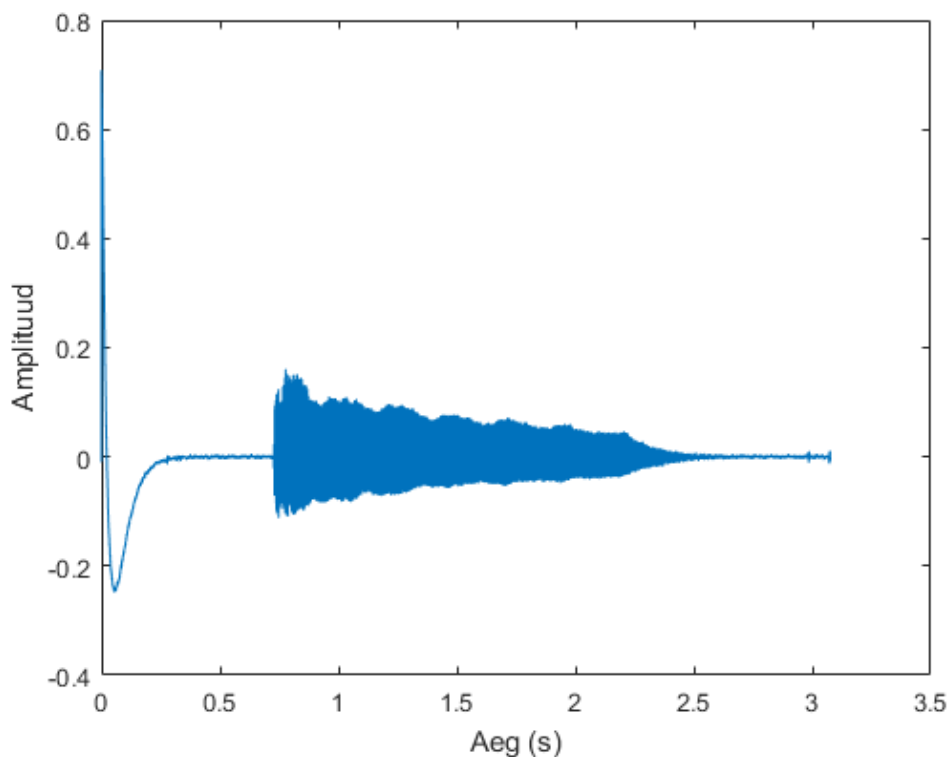
Autori eesmärk on luua Androidi platvormile mobiilne tarkvara, mis on eestikeelne ning aitaks solfedžoõpingutega alustajal selgeks saada nii noodid kui ka rütmid. Samuti on see rakendus kõigile tasuta kasutamiseks.

## 2. Nootide tuvastus Androidil

Loodav tarkvara peab omama nootide tuvastamise funktsionaalsust. Selleks kasutatakse enimlevinud meetodit FFT ehk kiiret Fourier' teisendust. Järgnevas peatükis antakse ülevaade digitaalheli salvestusprotsessist, töötlemisprotsessist ning noodituvastusprotsessist.

### 2.1 Digitaalheli salvestamine

Heliks nimetatakse elastses keskkonnas levivaid laineid, mida on võimalik kuulda [10]. Helisid on võimalik salvestada digitaalsele kujule (CD plaadile, arvutisse), kuid digitaalne salvestamine vähendab heli kvaliteeti. Digitaalsalvestuse protsessi käigus võetakse lainekujust perioodiliselt hetkväärtused, millele seatakse vastavusse täpsed arvud (vt joonis 3). [11]



Joonis 3. Digitaalse heli kujutus aja- ja amplituudi telje kaudu.

Digitaalset salvestust käsitletakse allikas [11], mille põhjal on koostatud alljärgnev lõik. Digitaalse salvestuse kvaliteet sõltub diskreetimissagedusest ehk *sample rate*'st. Diskreetimissagedus näitab, mitu hetkväärtust ehk diskreeti ehk *sample*'it sekundis salvestatakse. Seda mõõdetakse hertsides (Hz). Mida suurem on diskreetimissagedus, seda

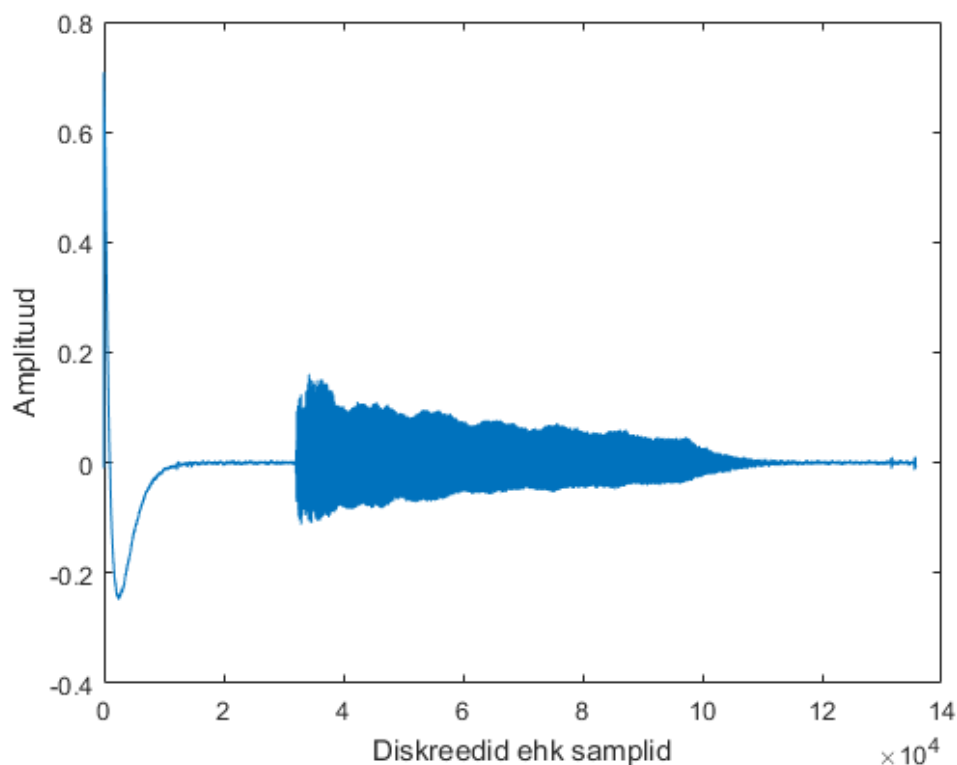
kvaliteetsem on salvestus, kuid samas suurendab see ka andmemahutu. Kõige rohkem on levinud diskreetimissagedus 44100 Hz, mida kasutatakse nii CD plaatide, MP3-failide kui ka paljude teiste heliformaatide salvestamiseks. Samuti mõjutab digitaalse salvestuse kvaliteeti bitisügavus. Bitisügavus näitab, mitut bitti kasutatakse ühe diskreedi kujutamiseks. Mida rohkem bitte kasutatakse, seda täpsem on diskreedi esitus. Kõige levinum bitisügavus on 16 bitti, mille tulemusena saab võnkeamplituudi hetvärtuse kirjeldamiseks kasutada 65535 erinevat 0 ja 1 kombinatsiooni [12]. Seda kasutavad enamus audioformaadid.

Kõige levinum audio salvestamise formaat on PCM ehk *pulse-code modulation*. Seda kasutatakse toore (*raw*) tihendamata audiofaili salvestamiseks. PCM formaat on toetatud enamus tarkvarade poolt ning seda on võimalik hiljem töödelda. [13]

## 2.2 Digitaalheli töötlus

Igale klaveril mängitud noodile vastab mingi kindel sagedus (näiteks esimese oktaavi A sagedus on 440 Hz, jne). Digitaalheli töötlemise tulemusena on võimalik leida helis domineeriv sagedus ning selle põhjal tuvastada noote.

Kui heli on salvestatud digitaalsele kujule, siis koosneb see teatud arvust diskreetidest ning igale diskreedile vastab mingi konkreetne arv (näiteks 0.0041, -0.0040, 0.0089, -0.0094, 0.0126, 0.0064, 0.6437, jne) . Alljärgnevalt on toodud ühe digitaalheli (noodi C ehk do) kujutus enne töötlusprotsessi, kus igale x-telje väärtusele ehk diskreedile vastab amplituud (vt joonis 4).



Joonis 4. Noodi C ehk do kujutus amplituudi ja diskreetide telje kaudu.

Noodi sageduse leidmiseks tuleb teisendada ajapõhised andmed (diskreedid ehk *samplid*) sageduspõhisteks. Seda aitab teha algoritm DFT ehk *Discrete Fourier Transform* ehk diskreetne Fourier' teisendus. Algoritm jagab helifaili väikesteks osadeks ehk sageduse bin'ideks, kusjuures bin'e on täpselt sama palju kui oli algselt diskreete. Iga k-nda bin'i arvutamiseks kasutatakse valemit:

$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{\frac{-j2\pi kn}{N}} = \sum_{n=0}^{N-1} x_n \cdot \left( \cos\left(-2\pi k \frac{n}{N}\right) + i \sin\left(-2\pi k \frac{n}{N}\right) \right)$$

kus    N - helifaili diskreetide arv;  
       n - töödeldava diskreedi number (0, ..., N-1);  
        $x_n$  - signaali väärtus ajahetkel n;  
       k - praegune sagedus (0, ..., N-1 Hz);  
        $X_k$  - k's sageduse bin. [14]

DFT on väga aeglane ( $\Theta(n^2)$ ). Seetõttu kasutatakse selle teostamiseks kiiremat versiooni ehk FFT'd ehk Fast Fourier Transform'i ehk kiiret Fourier' teisendust ( $\Theta(n \log n)$ ). See põhineb samuti DFT tehnoloogial, ainult et selle puhul vähendatakse oluliselt arvutusmahtu.  $\sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi kn/N}$  summa diskreedid jagatakse pidevalt kaheks - paaritu indeksiga diskreedid ja paarisindeksiga diskreedid - ja nii iga summa puhul:

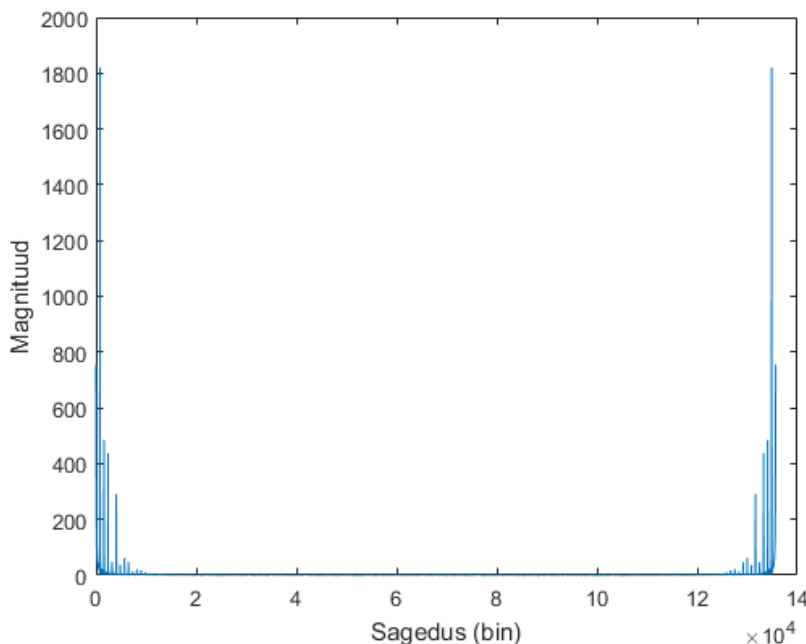
$$X_k = \sum_{n=0}^{N-1} x_n \cdot e^{-j2\pi kn/N} = \sum_{m=0}^{\frac{N}{2}-1} x_{2m} \cdot e^{-j2\pi k(2m)/N} + \sum_{m=0}^{\frac{N}{2}-1} x_{2m+1} \cdot e^{-j2\pi k(2m+1)/N}$$

. [14, 15]

Peale FFT algoritmi rakendamist vastab igale sageduse bin'ile üks kompleksarv (näiteks:  $-7.2802 + 0.0000i$ ,  $-7.0020 + 1.9807i$ ,  $-6.2523 + 3.7933i$ ,  $-5.1101 + 5.2997i$ , jne). Edasi tuleb teisendada kompleksarvud reaalarvu kujule ehk leida iga bin'i kompleksarvu magnituud ehk moodul, kasutades valemit:

$$\sqrt{A_k^2 + B_k^2} \quad [16].$$

Pärast teisendust saame kanda andmed uuesti graafikule, ainult et nüüd on ajatelje asemel sagedustelg bin'idega (vt joonis 5).

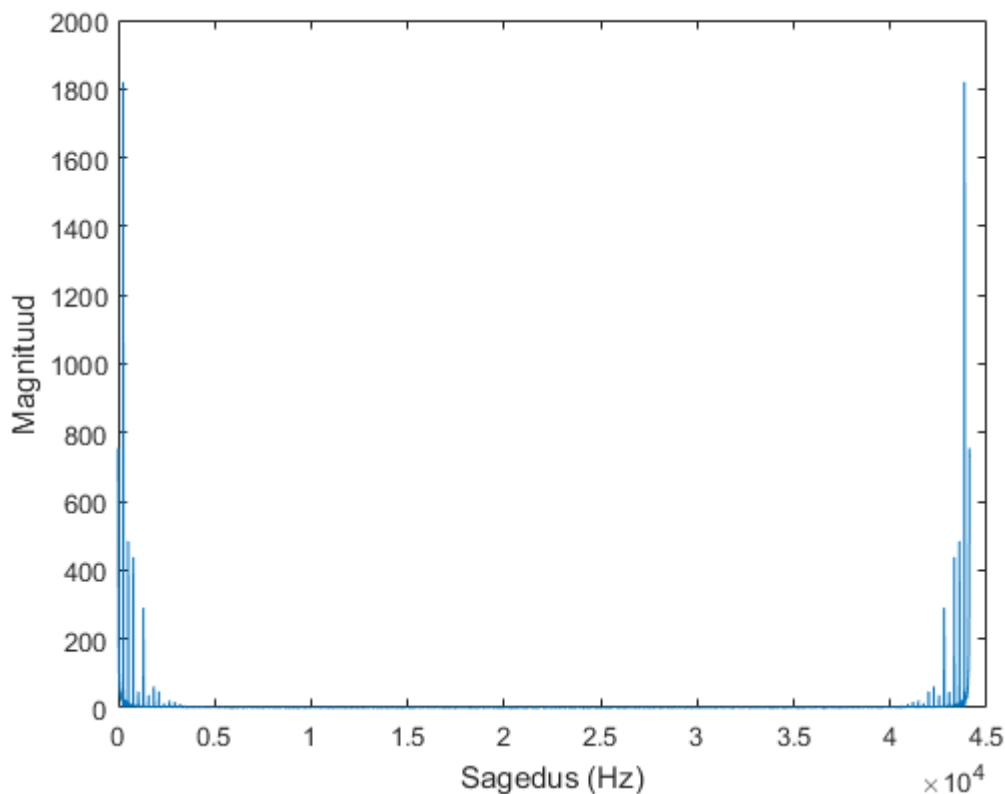


Joonis 5. Noodi C ehk do kujutus magnituudi ja sagedustelje (bin) kaudu.

Noodi tuvastamiseks on vaja leida kõige suurema magnituudiga bin'i indeks (vt joonis 5). Peale indeksi leidmist on tuleb leida selle indeksi kohal oleva bin'i sagedus hertsides (vt joonis 6). Teades heli salvestamise diskreetimissagedust ning sageduse bin'ide arvu helisalvestuses, saame arvutada ühe bin'i väärtuse hertsides. Selleks tuleb korrutada bin'i indeks diskreetimissagedusega ning jagada bin'ide arvuga:

$$F_k = \frac{k * \text{diskreetimissagedus}}{\text{bin'ide arv}}, \quad (1)$$

kus k on otsitava bin'i indeks [16].



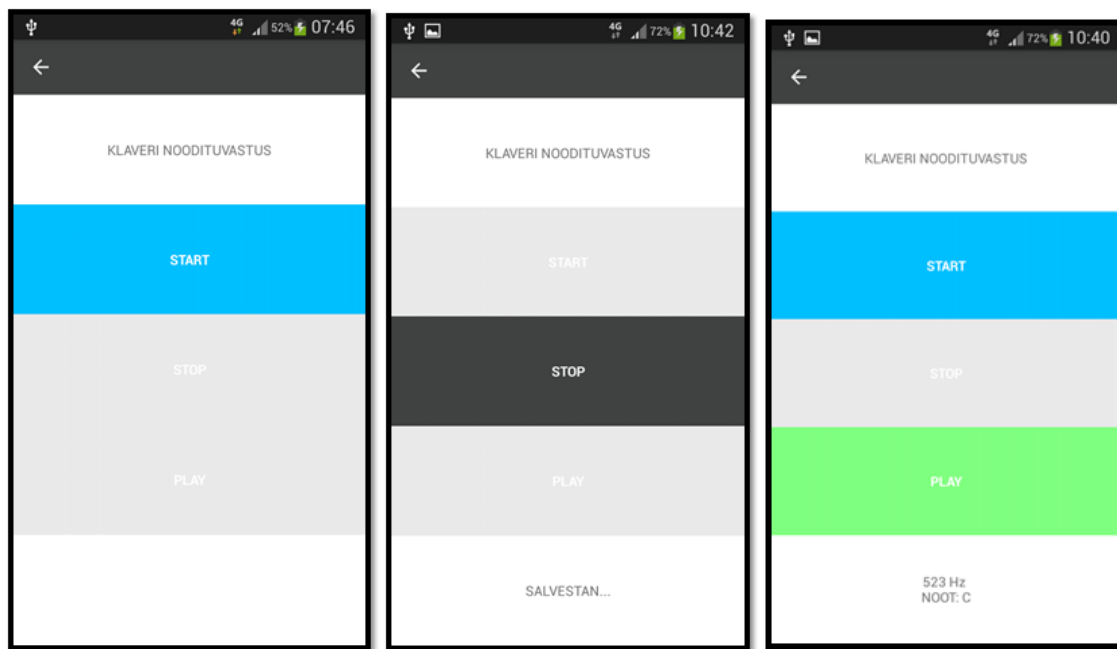
Joonis 6. Noodi C ehk do kujutus magnituudi ja sagedustelje (Hz) kaudu.

Kui suurima magnituudiga bin'i sagedus on leitud (Hz), võrreldakse seda kokkuleppeliste nootide sagedusega. Sageduste ühtimise korral on noot tuvastatud.



## 2.3 FFT realiseering rakenduses

Noodituvastuse protsessi jaoks on vaja esmalt salvestada klaveril mängitud noot digitaalheliks. Rakenduses näeb nootide tuvastuse tegevus välja järgmine (vt joonis 7):



Joonis 7. Klaveri noodituvastuse tegevus rakenduses.

Heli salvestuseks kasutatakse PCM formaati, kuna see on väga väikeste kadudega ning seda on võimalik hiljem töödelda. (vt joonis 8, rida 145-149). Nutitelfoni mikrofonist andmete kättesaamiseks kasutatakse Java klassi *AudioRecord*, mis aitab Java rakendustel lugeda andmeid audio salvestamise riistvaralt meetodiga *read()*. Enne heli salvestamist pannakse paika salvestamisprotsessi olulised faktorid (vt joonis 8, rida 152-162):

1. salvestuse audioallikaks Androidi seadmel on mikrofoni;
2. diskreetimissageduseks (*sample rate*) 44100 Hz (see on ainus diskreetimissagedus, mille puhul on garanteeritud, et salvestamine toimib kõikide Android seadmetega);
3. heli tuleb ühest kanalist sisse ehk *CHANNEL\_IN\_MONO*;
4. audio formaadiks valime *ENCODING\_PCM\_16BIT* (see on hetkel ainuke audio andmete formaat, mis on toetatud kõikide Android seadmete poolt);
5. puhvri suuruse leiame *AudioRecord* sisseehitatud funktsiooni abil *getMinBufferSize*, et vältida seda olukorda, kus puhvri suuruseks määratakse väiksem arv, kui see olla tohiks,

mille tõttu ei looda *AudioRecord* objekti. [17,18]

```
143 //http://android-er.blogspot.com/2014/04/audiorecord-and-audiotrack-and-to.html
144 private void startRecord() {
145     File file = new File(Environment.getExternalStorageDirectory(), "recordedFile.pcm");
146
147     try {
148         DataOutputStream dataOutputStream = new DataOutputStream(
149             new BufferedOutputStream(new FileOutputStream(file)));
150
151
152         int bufferSize = AudioRecord.getMinBufferSize(
153             44100,
154             AudioFormat.CHANNEL_IN_MONO,
155             AudioFormat.ENCODING_PCM_16BIT);
156
157         AudioRecord audioRecord = new AudioRecord(
158             MediaRecorder.AudioSource.MIC,
159             44100,
160             AudioFormat.CHANNEL_IN_MONO,
161             AudioFormat.ENCODING_PCM_16BIT,
162             bufferSize);
163
164         audioRecord.startRecording();
165
166         short[] audioData = new short[bufferSize];
167
168         while (isRecording()) {
169             int numberOfShort = audioRecord.read(audioData, 0, bufferSize);
170             for (int i = 0; i < numberOfShort; i++) {
171                 try {
172                     dataOutputStream.writeShort(audioData[i]);
173                 } catch (IOException e) {
174                     e.printStackTrace();
175                 }
176             }
177         }
178         audioRecord.stop();
179
180         dataOutputStream.close();
181     } catch (Exception e) {
182         e.printStackTrace();
183     }
184     runOnUiThread(() -> {
187         answerTW.setText("");
188     });
189
190 }
191 }
```

Joonis 8. Heli salvestamine Android Studios.

Kui digitaalne helifail on salvestatud, siis helifaili töötlus algab pärast seda, kui kasutaja vajutab nupule “Play” (vt joonis 7). Kogu helifail koos oma diskreetidega loetakse sisse ujukoma massiivi ehk *float array*’sse. Massiivile rakendatakse FFT algoritmi kasutades Java teeki JTransforms-3.1-with-dependencies [19]. Tulemuseks on massiiv kompleksarvudega (vt joonis 9, read 225-228). Et saada kätte noodi sagedus, tuleb esmalt leida kompleksarvude magnituudid ehk moodulid, leida üles kõige suurema magnituudiga kompleksarvu indeks ning rakendada valemit (1).

Kui helifaili sagedus on käes, siis tuleb seda võrrelda klaveri nootidele mõeldud sagedustega [20].

```
224 //https://gist.github.com/jongukim/4037243
225 DoubleFFT_1D doubleFFT_1D = new DoubleFFT_1D(dataForFFT.length);
226 double[] fft = new double[dataForFFT.length * 2];
227 System.arraycopy(dataForFFT, 0, fft, 0, dataForFFT.length);
228 doubleFFT_1D.complexForward(fft);
229
230 //http://stackoverflow.com/a/7675171
231 double [] magnitude=new double[dataForFFT.length/2];
232 for(int j=0; j< (dataForFFT.length/2-1);j++){
233     double re = fft[2*j];
234     double im = fft [2*j+1];
235     magnitude[j] =Math.sqrt (re*re + im*im);
236 }
237 double max_mag= Double.NEGATIVE_INFINITY;
238 int max_ind= -1;
239 for(int k=0; k< (dataForFFT.length/2-1);k++){
240     if(magnitude[k]>max_mag){
241         max_mag=magnitude[k];
242         max_ind=k;
243     }
244 }
245 frequency = max_ind * 44100 / dataForFFT.length;
```

Joonis 9. FFT algoritmi rakendamine sisseloetud helifailile ning noodi sageduse tuvastamine.

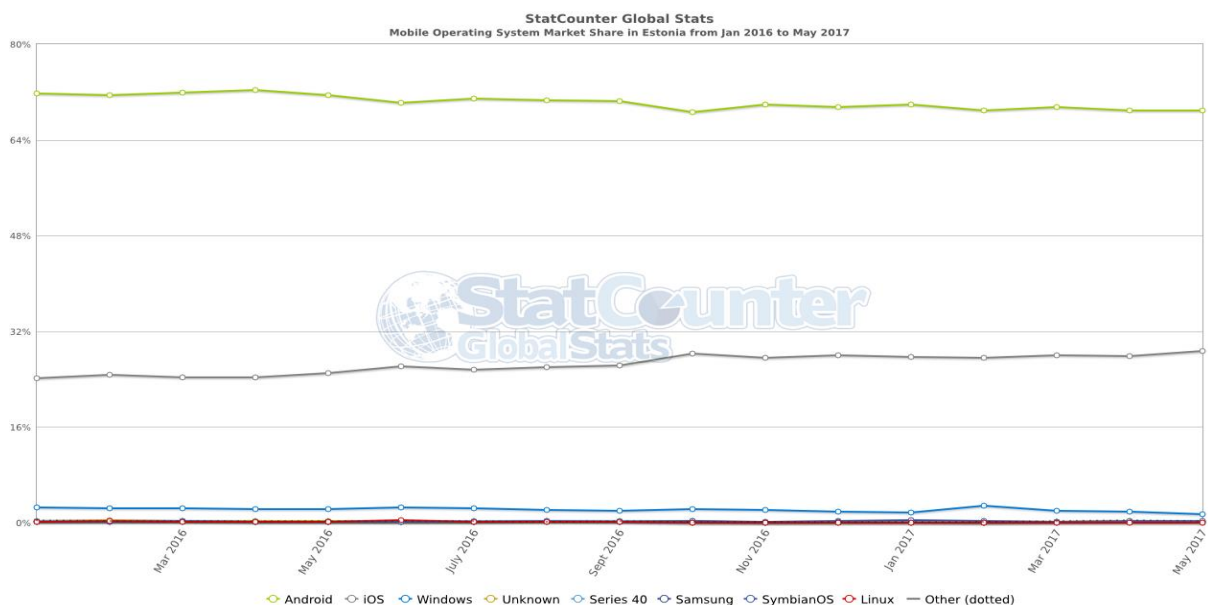
Valminud funktsionaalsus suudab tuvastada ainult ühe noodi korraga, st et kui kasutaja mängib klaveril kaks nooti korraga, siis seda ei tuvastata. Samuti tuleks salvestusprotsess läbi viia vaikusel, et vältida taustamüra helituvastusel. Kõige paremini töötab helituvastuse funktsioon süntesaatoriga. Süntesaatoriga tuvastamine on täpsem kui tavalise klaveriga tuvastamine, kuna tavaline klaver võib ajapikku minna häälest ära, st et nootide sagedused pole enam kooskõlas kokkuleppeliste nootide sagedustega. Samuti mõjutab nootide tuvastust see, kuidas klaver on häälestatud. Rakendus suudab tuvastada klaveri valgete klahvide noote kolme oktaavi (väike, esimene ja teine) ulatuses, kusjuures iga noodi sageduse puhvriks on ligikaudu pool tooni. Samuti on üheks algoritmi miinuskohaks ka see, et antud algoritmi tulemusena tuvastatakse süntesaatoril küll õige noot, kuid algoritm ajab segamini oktavite vahelised sagedused, nt süntesaatori peal tuvastab algoritm nii väikese oktaavi kui ka esimese oktaavi c sageduseks 523 Hz, kuigi tegelikult üks väikese oktaavi c sagedus peaks olema 130 Hz. Üheks suureks plussiks on see, et rakendus suudab tuvastada noote ka siis, kui kasutaja laulab õiget nooti.

### 3. Ülevaade valminud rakendusest

Töö käigus valmis rakendus “Solfedžo õpitarkvara 1. klassile”, mis muudab õpilaste solfedžoõpingud muusikakoolis huvitavamaks ning kergemaks. Keskendutakse kahele peamisele teemale - noodiõpetus ja rütmiopeetus. Rakenduse kood on üles laaditud GitHub’i [21].

#### 3.1 Kasutatud tehnoloogiad

Kuna eesmärgiks oli teha mobiilirakendus, mida saaksid kasutada võimalikult paljud Eesti õpilased, siis tuli uurida erinevate mobiilsete operatsioonisüsteemide turuseisukohta Eestis. Kaks kõige populaarsemat mobiilset operatsioonisüsteemi 2016. aasta oktoobris olid Andorid ning iOS, kuid kuna Androidi turuosakaal Eestis oli 68,62% ning iOSi oskaal 28,28%, siis langes otsus Androidi kasuks (vt joonis 10) [22]. Samuti aitas otsusele kaasa ka see, et Android on olnud stabiilne turu liider juba väga pikka aega ning kuna autoril polnud varem kogemust Androidi platvormile rakenduse tegemisega, siis tundus see talle põneva väljakutsena.



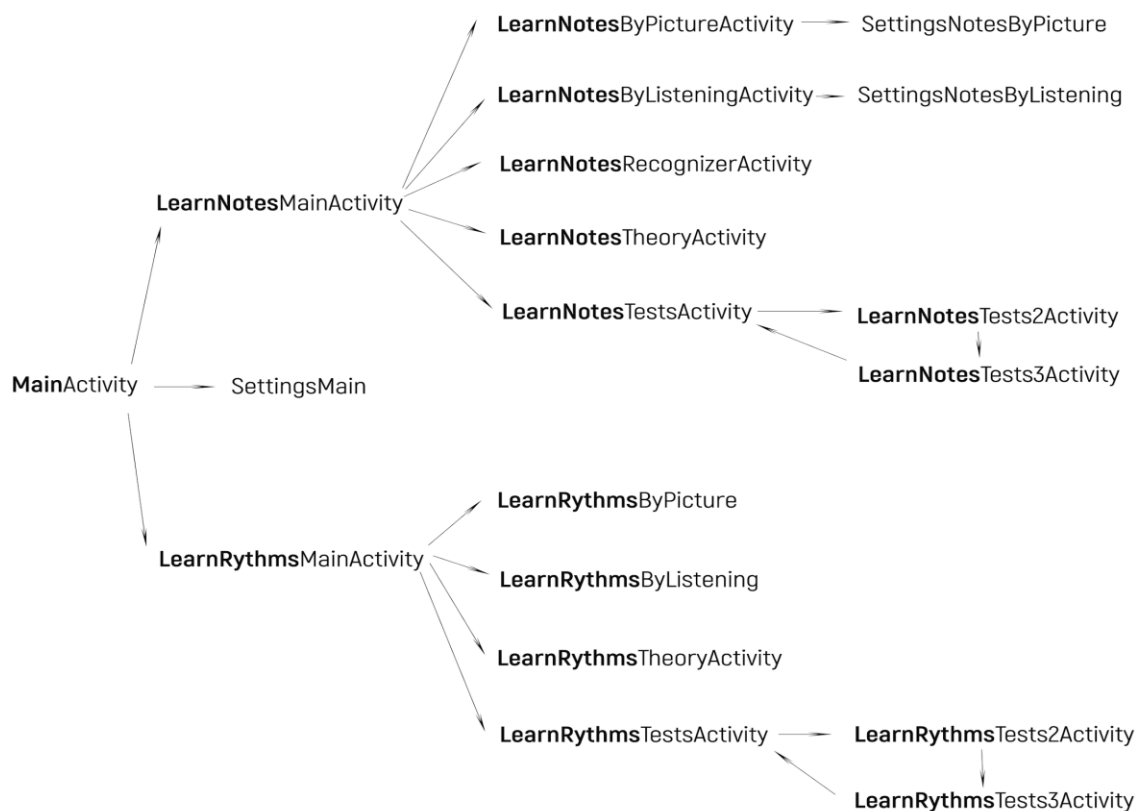
Joonis 10. Mobiili operatsioonisüsteemide turuseis Eestis 2016 jaanuar kuni 2017 mai [22].

Androidi rakenduse tegemiseks kasutati Android Studiot ning programmeerimiskeeleks oli Java. Androidi platvormi versiooni valimisel lähtuti seisukohast, et tuleks valida selline versioon, mida

toetaksid enamuse mobiiliseadmetest. Seega valiti versioon 4.0.3 Ice Cream Sandwich (miinimum API level 15), mida toetas kõige suurem osa aktiivsetest Google Play Store'i seadmetest (97,4%) [23].

## 3.2 Rakenduse arhitektuur

Loodud rakendus koosneb ühest *MainActivity*st ehk ühest peamisest tegevusest, mis jaguneb omakorda kaheks alamtegevuseks - *LearnNotesMainActivity* ja *LearnRhythmsMainActivity* - vastavalt kahele peamisele õpikategooriale “ÕPI NOOTE” ja “ÕPI RÜTME”. Lisaks on *MainActivity* juures ka *SettingsMain* tegevus, mis lubab kasutajal avalehel muuta nootide nimetuste seadeid. Edasi jaotuvad *LearnNotesMainActivity* ja *LearnRhythmsMainActivity* omakorda alamtegevusteks (vt joonis 11).



Joonis 11. Rakenduse ülesehitus.

Projekti kood on jaotatud kolme kausta - *manifests*, *java* ja *res* - mis automaatselt genereeritakse

projekti loomisel. *Manifests* kausta all paikneb fail *AndroidManifest.xml*, kus hoitakse vajalikku infot rakenduse ja tegevuste kohta, samuti saab seal määrata ära erinevaid *permission*'e ehk lubasid, näiteks kui tahetakse, et rakenduses oleks võimalik helisid salvestada, siis tuleb lisada rida:

```
<uses-permission android:name="android.permission.RECORD_AUDIO" /> [24].
```

*Java* kausta alla on kirja pandud kogu rakenduse loogika osa ehk erinevad tegevused (*activity*'d).

*Res* kaust jaguneb omakorda kuueks kaustaks, kus *drawable* kausta alla on paigutatud pildid ja üksikud kujunduselementide stiilid *.xml* failidena, *layout* kausta all on kogu rakenduse paigutuslik ja kujunduslik osa, *menu* kausta all on defineeritud menüü koos menüüelementidega, *raw* kaustas on kõik helifailid ning *values* kaustas on eraldi välja toodud eraldi fail nii sõnede (*strings.xml*), stiilide (*styles.xml*), värvide (*colors.xml*) kui ka erinevate dimensioonide (*dimens.xml*) defineerimiseks.

Lisaks on kasutatud projektis välist *library*'t ehk teeki *JTransforms-3.1-with-dependencies.jar* [19], et kasutada sealset FFT algoritmi helitöötluseks.

### 3.3 Tarkvara kvaliteet ja jõudlustulemused

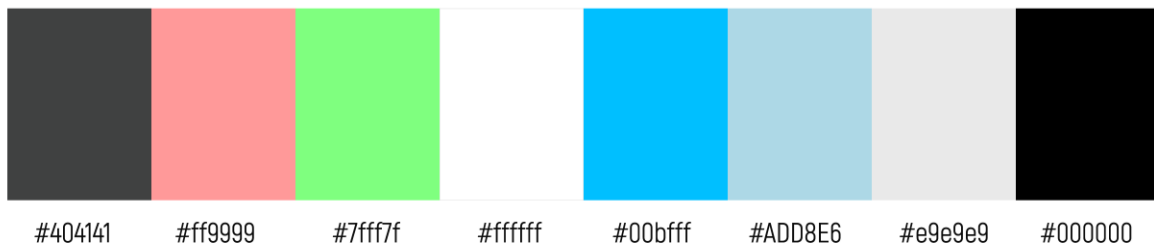
Tarkvara kvaliteedi kontrollimiseks testiti rakendust kõige madalama API leveliga (15) ning sihtgrupile suunatud API leveliga (25). Lisaks uuriti millised on uuemate API levelite erinevused võrreldes vanematega. Näiteks selgus testimise käigus, et uuemates API'des on vaja küsida rakenduse kasutajalt eraldi luba heli salvestamiseks. Enam ei piisanud lihtsalt *Manifest.xml* faili *user permission*'i ehk kasutajaloo kirjutamisest. Lisaks testiti rakendust ka musta kasti testimise meetodil (*black-box testing*), kus rakendust testis inimene, kes polnud varem seda kasutanud.

Rakenduses oli klaveri programmeerimisel vaja kasutada väga palju helifaile, mille laadimine klahvivajutuse ajal muutis rakenduse väga aeglaseks. Selle parandamiseks otsustati kasutada Java klassi *SoundPool*. See võimaldab laadida kõik helifailid enne uue *Activity* ehk tegevuse alustamist mälli, vähendades CPU ehk protsessori koormust. Tegevuse juurest lahkudes tuleb kutsuda esile funktsiooni *release*, et vabastada mälu. Samuti kutsuti koodis iga *onDestroy*

meetodi juures funktsiooni *finish*, et vabastada kõik ressursid, mida enam vaja pole. See aitab suurendada vaba mälu osakaalu.

### 3.4 Kasutajaliides

Kasutajaliidese kujundamisel on kasutatud järgmist värvipaletti (vt joonis 12):



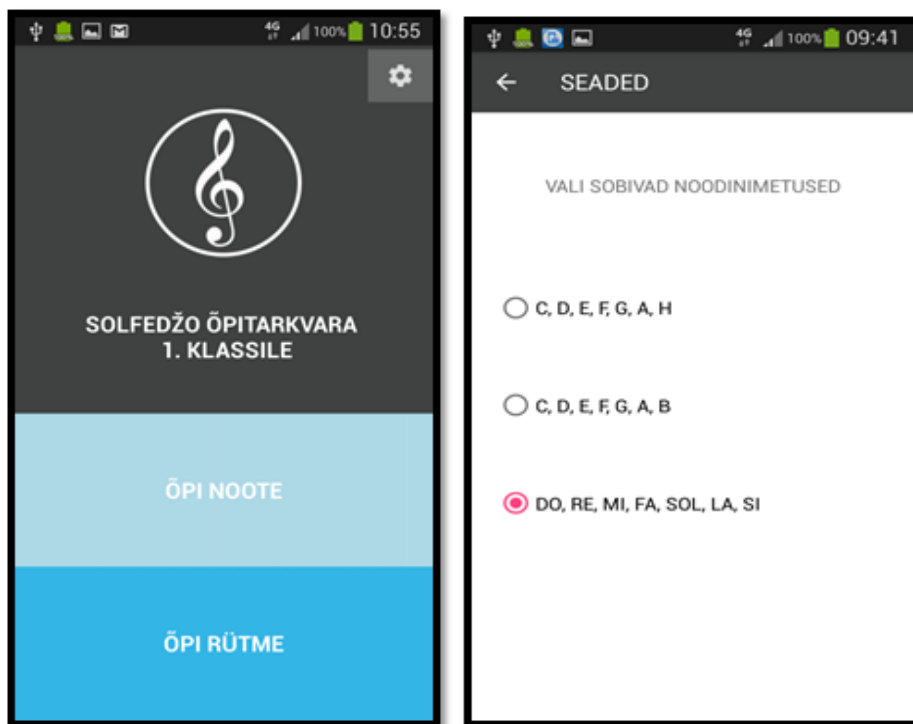
Joonis 12. Rakenduse värvikombinatsioonid.

Kogu tekst rakenduses on kirjutatud trükitähtedega, et vältida seda olukorda, kus mõni esimese klassi õpilane ei oska veel kirjatähti lugeda.

Piltide ja logode tegemiseks kasutati Adobe PhotoShop CS6’e ning Material Design lehekülje ikoone [25, 26]. Rakenduses oleva klaveri noodi helid on saadud leheküljelt FreeSound [27].

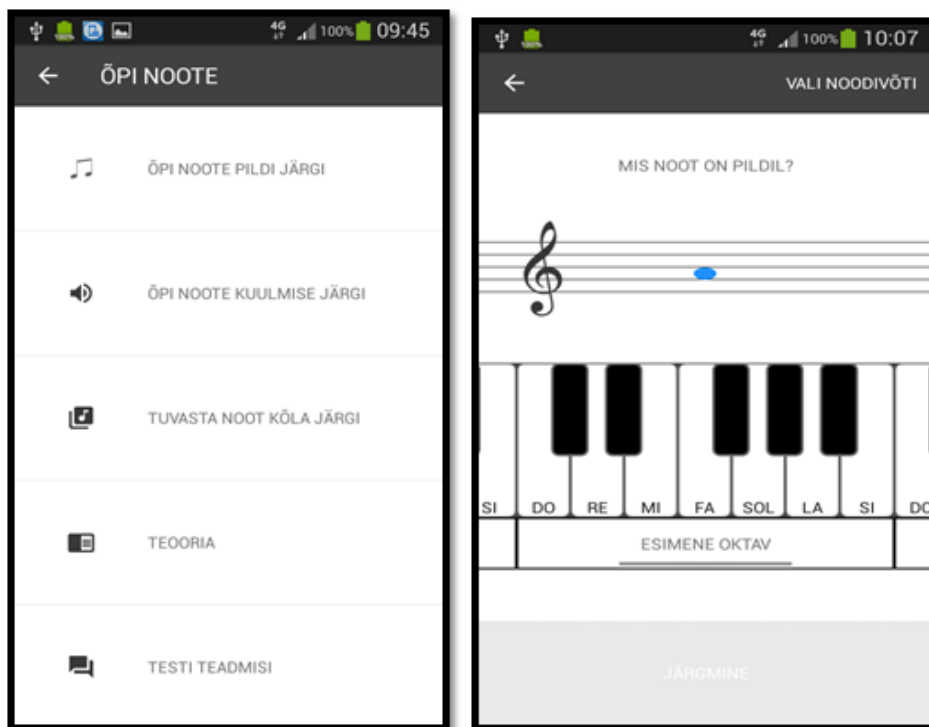
### 3.5 Kasutusjuhend

Rakenduse avamisel avaneb kasutajale avaleht, kus kasutajal on võimalik valida kahe peamise kategooria vahel “Õpi noote” ja “Õpi rütme”. Vajutades avalehe paremal nurgas olevale seadete nupule, avaneb vaade “Seaded”, kus kasutajal on võimalik muuta rakenduses kuvatavate nootide nimetusi (vt joonis 13).



Joonis 13 . Rakenduse avaleht ning seadete leht.

Kui kasutaja tahab õppida noote, siis tuleb vajutada nupule “Õpi noote”. Avaneb järgnev vaade (vt joonis 14):



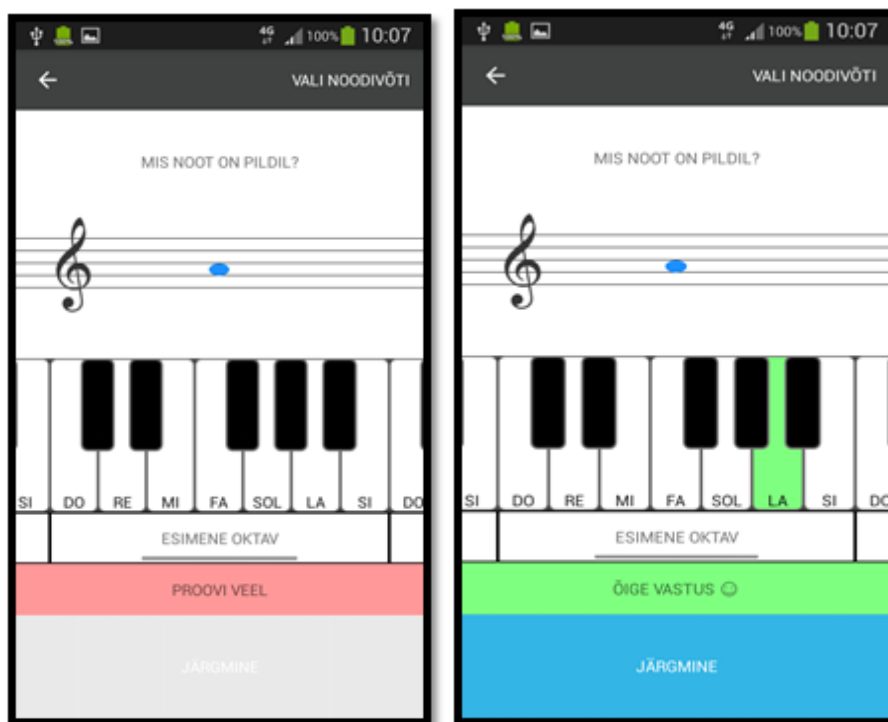
Joonis 14 . Vasakul kategooria „Õpi noote“ avaleht, paremal “Õpi noote pildi järgi” vaade.



Edasi saab kasutaja valida viie alamülesande vahel:

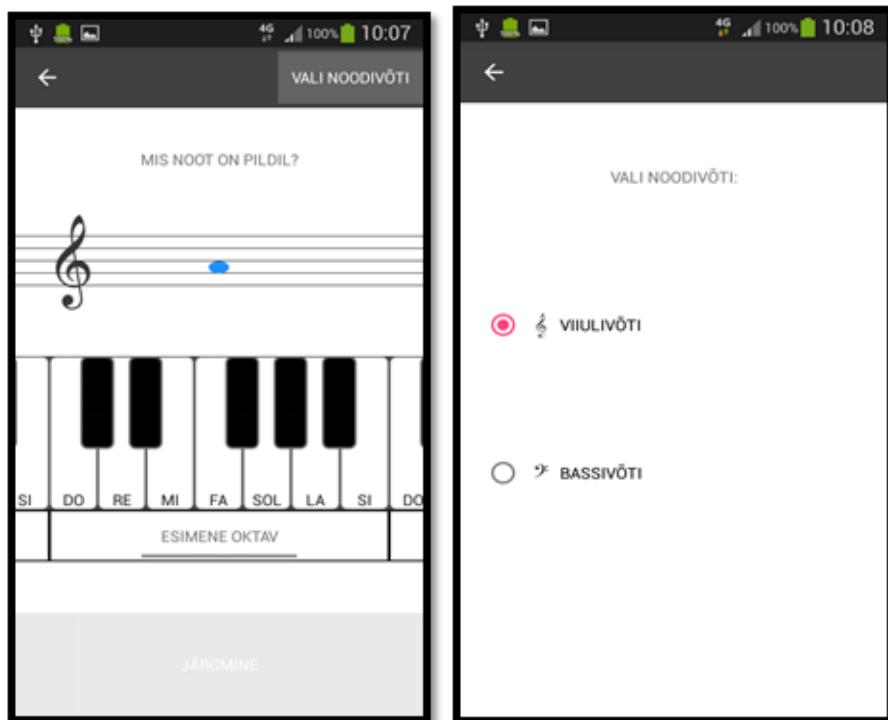
1. Kui kasutaja vajutab nupule “Õpi noote pildi järgi”, siis on kasutajal võimalik õppida noodijoonestikul oleva pildi järgi noodinimetusi ja nende paiknemist klaviatuuril kolme oktavi ulatuses (vt joonis 14). Klaveri klaviatuuri saab näputõmbega liigutada nii vasakult paremale kui ka paremalt vasakule.

Kui kasutaja ei tunne pildil olevat nooti ära ning vajutab valele klaveri klahvile, siis kuvatakse kasutajale tekst “Proovi veel”. Kui kasutaja vastab õigesti, siis kuvatakse tekst “Õige vastus” ning õige vastuse klahvi taustavärv muutetakse roheliseks (vt joonis 15). Kasutaja ei saa minna uut nooti ära arvama enne, kui ta on vastanud õigesti. Pärast õiget vastust kuvatakse kasutajale nupp “Järgmine” ning peale selle vajutust kuvatakse kasutajale uus juhuslik noot noodijoonestikul.



Joonis 15 . Vale ja õige vastuse kuvamine kasutajale.

Kasutajal on võimalik õppida lisaks viiulivõtme nootidele ka bassivõtme noote. Selleks tuleb vajutada paremal nurgas nupule “Vali noodivõti” ning sealt valida sobiv noodivõti (vt joonis 16).



Joonis 16. Noodivõtme seadete muutmise.

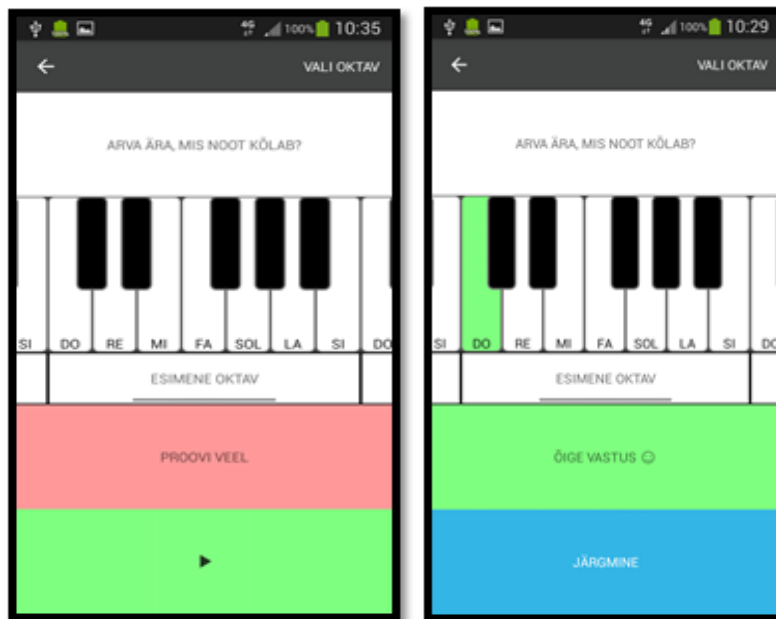
2. Kui kasutaja vajutab nupule “Õpi noote kuulmise järgi”, siis on tal võimalik õppida noote kuulmise järgi (vt joonis 17). Vaikimisi on alati selleks ainult esimese oktavi noodid).



Joonis 17. Nootide õppimine kuulmise järgi.

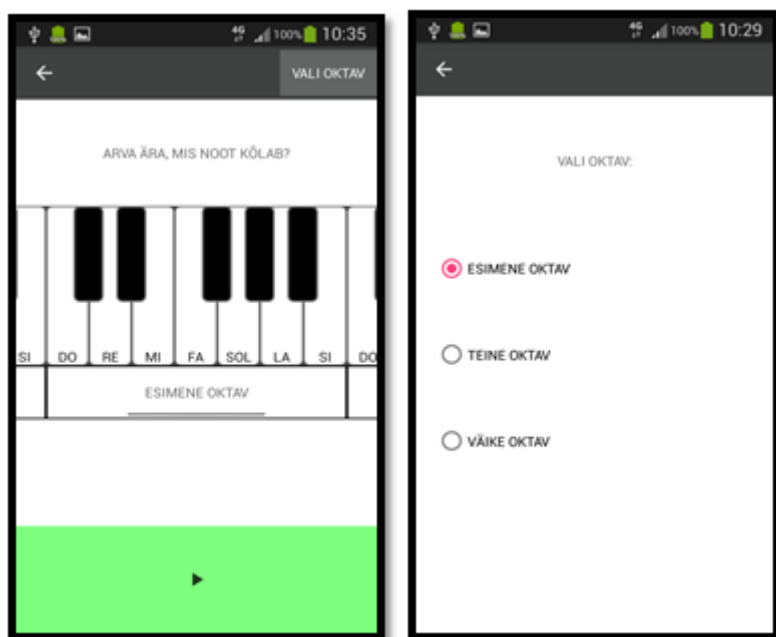
Kui kasutaja vajutab rohelisele “Play” nupule, siis mängitakse talle ette ühe noodi heli. Kasutaja peab ära arvama, mis heliga on tegemist, vajutades klaveril õigele klahvile. Vale

vastuse korral kuvatakse kasutajale tekst “Proovi veel” ning õige vastuse korral “Õige vastus”(vt joonis 18). Kui kasutaja vastab õigesti, kuvatakse talle nupp “Järgmine” ning genereeritakse uus noot.



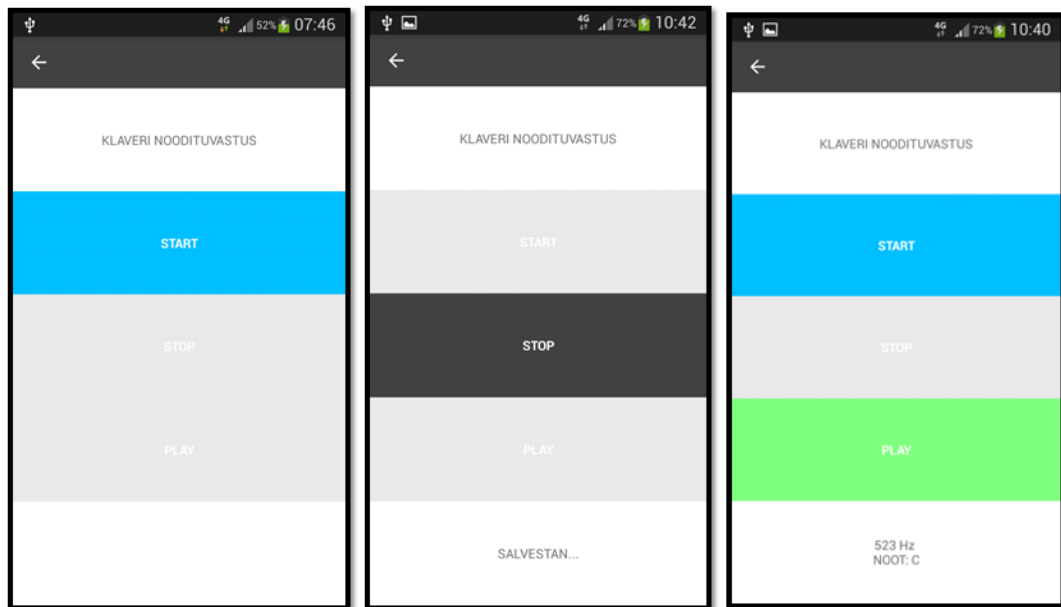
Joonis 18. Õige ja vale vastuse kuvamine.

Lisaks esimese oktavi nootidele on kasutajal võimalik õppida ka teise ja väikese oktavi noote, vajutades paremal nurgas olevale nupule “Vali oktav”, kusjuures tagasi minnes navigeeritakse kasutajal klaveri klaviatuur valitud oktavi juurde (vt joonis 19).



Joonis 19. Oktavi seadete muutmine.

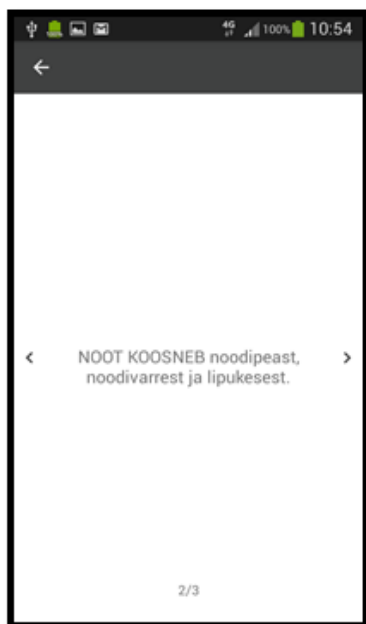
3. Kui kasutaja vajutab nupule “Tuvasta noot kõla järgi”, siis on tal võimalik tuvastada klassis oleval klaveril (esimese, teise ja väikese oktaavi ulatuses) mängitud noodi nimetus (vt joonis 20).



Joonis 20. Klaveri noodituvastus.

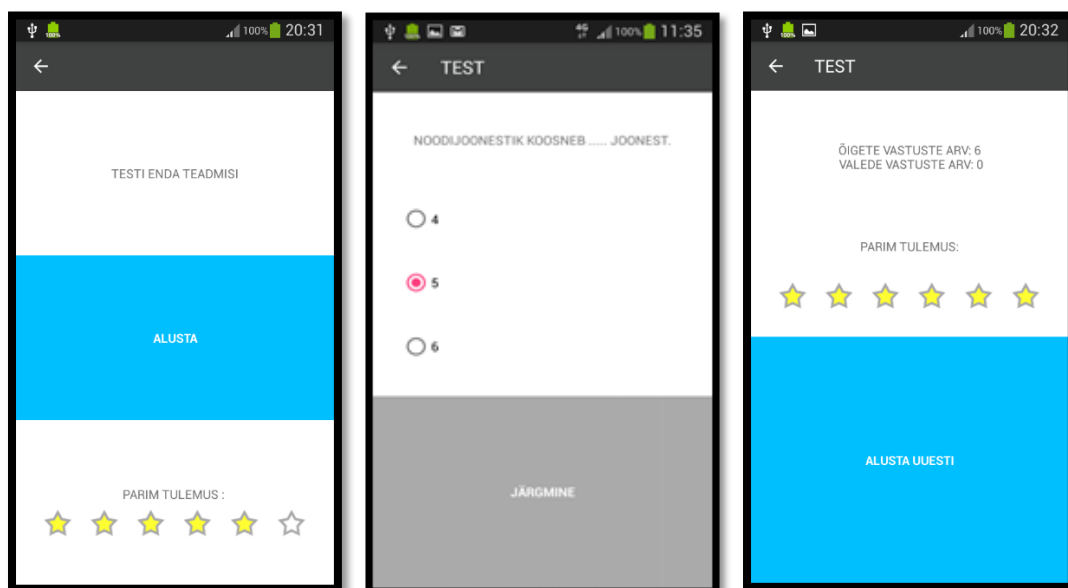
Selleks tuleb vajutada kõigepealt nupule “Start”, pärast mida tuleks mängida klaveril ainult üks heli, kuna tuvastamissüsteem suudab tuvastada helisid ainult ühe kaupa. Et salvestust lõpetada, tuleb vajutada nupule “STOP”. Pärast nupule “Play” vajutamist kuvatakse kasutajale nii noodi sagedus Hz kui ka noodi nimetus.

4. Kui kasutaja vajutab nupule “Teooria”, siis seal on tal võimalik õppida nootidega seotud mõisteid. Kasutajal on võimalik liikuda mõistete vahel nii nuppudega kui ka näputõmmetega (vt joonis 21).



Joonis 21. Teooria õppimine rakenduses.

5. Kui kasutaja vajutab nupule “Testi teadmisi”, siis on tal võimalik testida enda teoreetilisi teadmisi nootide kohta. Test koosneb kaheksast küsimusest ning samuti kuvatakse kasutajale ka läbi aegade kõige paremat tulemust tähekeste kujul, kus kollane täht näitab õigesti vastatud küsimuste arvu ning valge valesti vastatud küsimuste arvu (vt joonis 22).

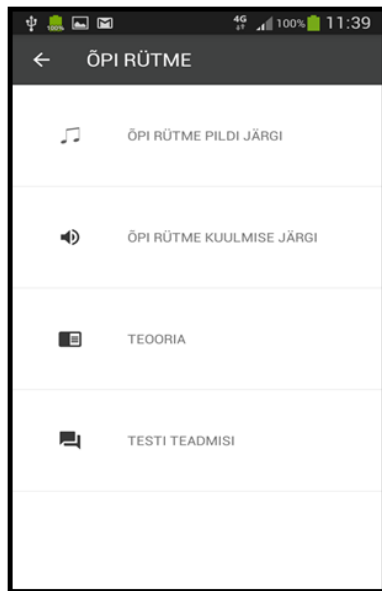


Joonis 22. Testi sooritamine.

Kui kasutaja vajutab nupule “Alusta”, siis kuvatakse talle järjest erinevaid küsimusi ning

igal küsimusel on täpselt üks õige vastus. Kui kasutaja on enda vastuses kindel, siis tuleb vajutada nuppu “Järgmine”. Kui kõikidele küsimustele on vastatud, kuvatakse kasutajale nii õigesti kui ka valesti vastatud küsimuste arv ja parim tulemus.

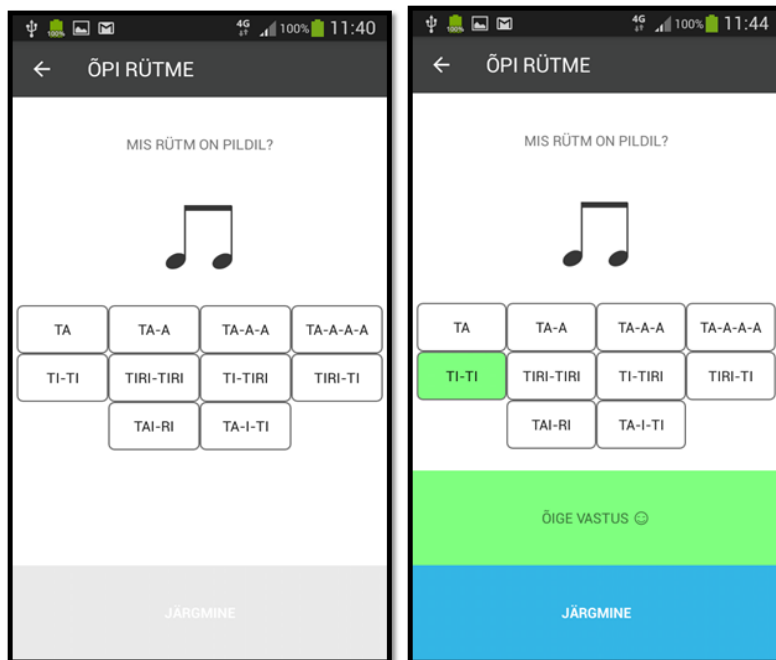
Kui kasutaja soovib õppida rütme, siis tuleb avalehel vajutada nupule “Õpi rütme”. Avaneb järgnev vaade (vt joonis 23):



Joonis 23. Kategooria “Õpi rütme” avaleht

Kasutajal on võimalik valida nelja erineva ülesande vahel:

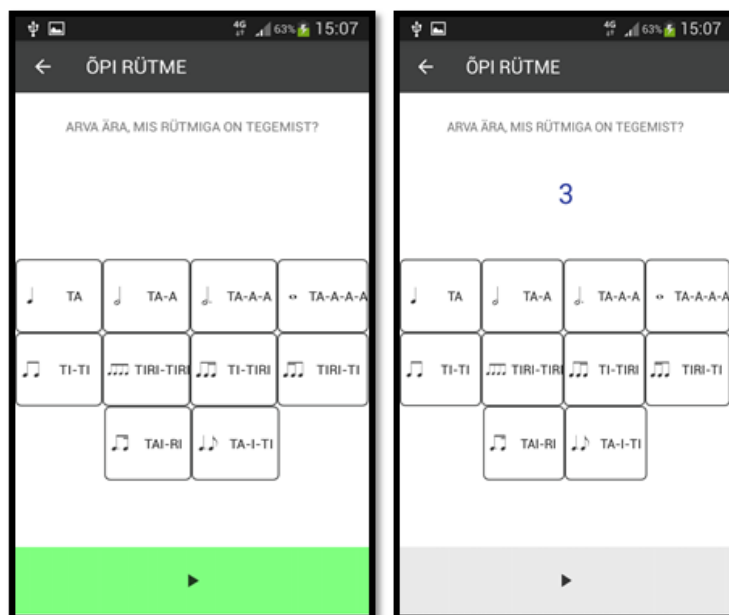
1. Kui kasutaja vajutab nupule “Õpi rütme pildi järgi”, siis on tal võimalik õppida rütmide nimetusi pildi järgi (vt joonis 24).



Joonis 24. Vaade “Õpi rütmepildi järgi”.

Kui kasutaja arvab valesti, siis kuvatakse talle tekst “Proovi veel” nii kaua, kuni ta arvab ära õige vastuse. Kui kasutaja arvab õige rütmi ära, siis kuvatakse tekst “Õige vastus”. Pärast õiget vastust kuvatakse kasutajale nupp “Järgmine” ning peale selle vajutust genereeritakse uus juhuslik rütmi pilt.

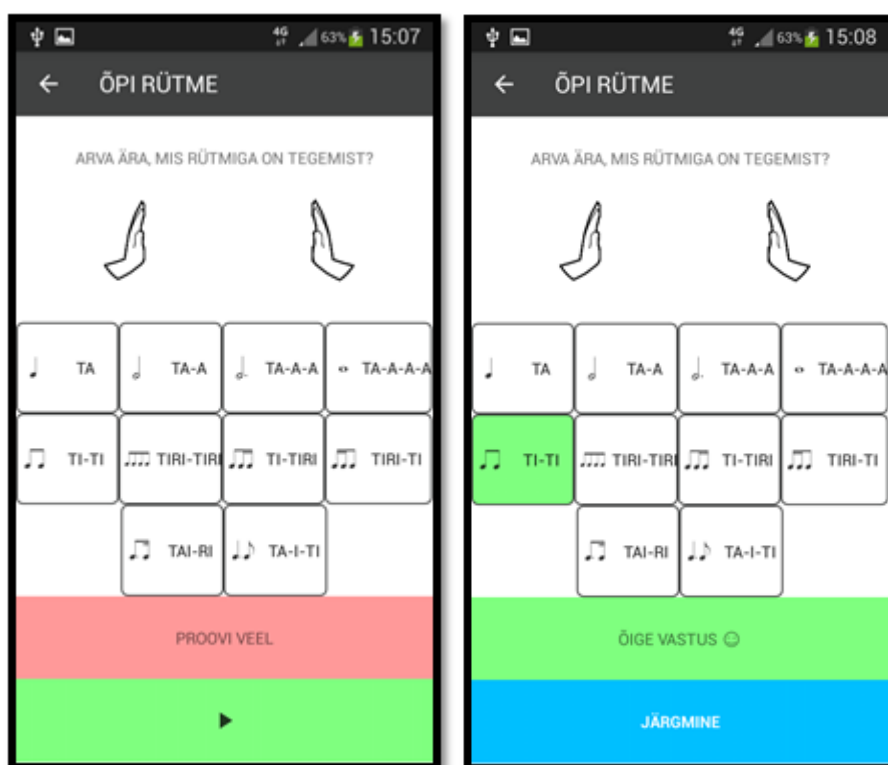
2. Kui kasutaja vajutab nupule “Õpi rütme kuulmise järgi”, siis kasutajal on võimalik kuulmise järgi ära arvata, mis rütmiga on tegemist (vt joonis 25).



Joonis 25. Vaade “Õpi rütme kuulmise järgi”

Kui kasutaja vajutab rohelisele nupule “Play”, siis kõigepealt ilmuvad ekraanile numbrid 3, 2, 1, mis tähendavad seda, rütm esitatakse peale seda, kui loendamine on jõudnud üheni. Pärast seda ilmuvad ekraanile käed, mis plaksutavad rütmi, mis tuleb ära arvata.

Kui kasutaja arvab valesti, siis kuvatakse talle tekst “Proovi veel” ning samuti saab kasutaja kuulata rütmi heli uuesti (vt joonis 26). Kui kasutaja arvab õige rütmi ära, siis muudetakse rütmi pilt roheliseks ning kuvatakse tekst “Õige vastus”. Pärast õiget vastust kuvatakse kasutajale nupp “Järgmine” ning peale selle vajutust genereeritakse uus juhuslik rütm.



Joonis 26. Õige ja vale vastuse kuvamine.

3. Kui kasutaja vajutab nupule “Teooria”, siis on tal võimalik õppida rütmidega seonduvaid mõisteid. Kasutajal on võimalik liikuda mõistete vahel nii nuppude kui ka näputõmmetega. Põhimõte on sama nagu “Õpi noote” kategooria teoorial (vt joonis 21)
4. Kui kasutaja vajutab nupule “Testi teadmisi”, siis on tal võimalik testida enda teadmisi rütmide kohta. Põhimõte on sama nagu “Õpi noote” kategooria testil (vt joonis 22).



# Kokkuvõte

Antud bakalaureusetöö eesmärgiks oli luua eestikeelne mobiilne rakendus Androidi platvormile, mis muudaks 1. klassi muusikakooli õpilastel nootide ja rütmide õppimise protsessi huvitavamaks ning samas ka kergemaks. Rakenduses on kasutajal võimalik õppida noote ja rütme nii kuulmise kui ka pildi järgi. Samuti on olemas rütmide ja nootide õppimise teooria osa, kust saab õppida erinevaid mõisteid ning hiljem enda teadmisi testida, lahendades testülesandeid. Rakendus võimaldab kasutajal tuvastada klaveril mängitud noote. Kõigepealt salvestatakse heli, siis töödeldakse seda ning töötluse tulemusel leitud heli sagedust võrreldakse kokkuleppeliste nootide sagedustega.

Töös koostati solfedžo õpitarkvara analüüs, kus anti lühiülevaade solfedžost, sõnastati rakenduse funktsionaalsed nõuded ning anti ülevaade sarnastest lahendustest. Lisaks kirjeldati digitaalheli salvestamise ja töötlemisprotsessi ning seletati, kuidas tuvastada digitaalheli järgi noodi sagedust. Töös toodi välja rakenduses kasutatud tehnoloogiad, arhitektuur ning loodi ka kasutusjuhend tarkvara kasutamiseks.

Rakenduse loomise käigus jõuti arusaamale, et nootide tuvastamine algoritm töötab kõige paremini ainult süntesaatoriga testides. Klaverite peal testides sõltub tulemus palju sellest, kas klaver on hääles või ei ole. Samuti jõudis autor järeldusele, et antud lahenduse nootide ja rütmide õppimise osa võiks laadida tulevikus Google Play poodi. Nii jõuaks see paljude 1. klassi õpilasteni, kes vajavad abi õppimisel.

# Kasutatud kirjandus

- [1] Prei E. IKT vahendite kasutusaktiivsus Eesti üldhariduskoolides. Tallinn. 2013  
[https://www.innovatsioonikeskus.ee/sites/default/files/tekstifailid/Sihtgrupi\\_kysitus\\_2012\\_2.pdf](https://www.innovatsioonikeskus.ee/sites/default/files/tekstifailid/Sihtgrupi_kysitus_2012_2.pdf)  
(21.04.2017)
- [2] Hiiesalu T. Nutiandmed muudavad iga klassi arvutiklassiks. *Õpetajate Leht*, 2016.  
<http://opleht.ee/2016/12/nutiseadmed-muudavad-iga-klassi-arvutiklassiks/> (21.04.2017)
- [3] Elleri Muusikakooli noorteosakond. Mis tund on solfedžo?  
<http://ellerikool.wixsite.com/noorteosakond/solfedo> (04.01.2017)
- [4] Semlek L. Muusika elementaarteooria õpik. Tallinn. 2000.
- [5] Eesti Muusikakoolide Liit. Solfedžo ainekava. 2016.  
[http://www.vorumuusikakool.ee/dokumendid/EML\\_solfedzo\\_ainekava\\_2016\\_dets.pdf](http://www.vorumuusikakool.ee/dokumendid/EML_solfedzo_ainekava_2016_dets.pdf) (08.05.2017)
- [6] Ülenurme Muusikakool. Õppematerjalid. Solfedžo. Klaviatuur koos bassivõtmega.  
<http://muusikakool.net/files/03-klaviatuur-koos-bassivõtmega-sc5686b9c0cbc2e.pdf> (27.04.2017)
- [7] 1 Learn music read music rakendus.  
<https://play.google.com/store/apps/details?id=org.anddev.android.solfa&hl=et> (28.04.2017)
- [8] solf.ee veebirakendus. <http://solf.ee/> (28.04.2017)
- [9] Perfect Ear- Ear Trainer  
<https://play.google.com/store/apps/details?id=com.evilduck.musiciankit&hl=et> (28.04.2017)
- [10] Ilmar Eiskop ja Aleksander Sillart. Akustika ja helitehnika. Tallinn „Valgus“. 1988
- [11] Pihlap M. Digitaalne heli. Tartu Ülikool.  
[https://et.wikipedia.org/wiki/Digitaalne\\_heli](https://et.wikipedia.org/wiki/Digitaalne_heli) (Toimetamisel) (20.04.2017)
- [12] Rinde A. Multimeediumi Õppematerjale. Helid arvutis.  
[http://www.cs.tlu.ee/~rinde/mm\\_materjal/pdf/mm\\_audio\\_wave.pdf](http://www.cs.tlu.ee/~rinde/mm_materjal/pdf/mm_audio_wave.pdf) (25.04.2017)
- [13] Rinde A. Multimeediumi Õppematerjale. Audiokompressioon.  
[http://www.cs.tlu.ee/~rinde/mm\\_materjal/pdf/mm\\_audio\\_compression.pdf](http://www.cs.tlu.ee/~rinde/mm_materjal/pdf/mm_audio_compression.pdf) (20.04.2017)
- [14] Diskreetne Fourier' teisendus.  
[https://et.wikipedia.org/wiki/Diskreetne\\_Fourier%27\\_teisendus](https://et.wikipedia.org/wiki/Diskreetne_Fourier%27_teisendus) (26.04.2017)
- [15] VanderPlas J. Understanding the FFT algorithm.  
<https://jakevdp.github.io/blog/2013/08/28/understanding-the-fft/> (25.04.2017)
- [16] How to get frequency from fft?  
<http://stackoverflow.com/a/7675171> (25.04.2017)
- [17] Android Developers. AudioRecord.  
<https://developer.android.com/reference/android/media/AudioRecord.html> (24.03.2017)

- [18] Android Developers. AudioFormat.  
[https://developer.android.com/reference/android/media/AudioFormat.html#ENCODING\\_PCM\\_16BIT](https://developer.android.com/reference/android/media/AudioFormat.html#ENCODING_PCM_16BIT)  
(24.03.2017)
- [19] Piotr Wendykier. JTransforms teek.  
<https://sites.google.com/site/piotrwendykier/software/jtransforms> (24.03.2017)
- [20] Musical Note to Frequency Conversion Chart  
<https://www.audiology.org/sites/default/files/ChasinConversionChart.pdf> (28.04.2017)
- [21] GitHub. <https://github.com/KristiinaK/SolfedzoSoftware> (10.05.2017)
- [22] StatCounter Global Stats. Mobile Operating System Market Share in Estonia  
<http://gs.statcounter.com/os-market-share/mobile/estonia/#monthly-201601-201610> (28.04.2017)
- [23] Android Studio. <https://developer.android.com/studio/index.html> (24.04.2017)
- [24] Android Developers. App Manifest.  
<https://developer.android.com/guide/topics/manifest/manifest-intro.html> (28.04.2017)
- [25] Adobe Photoshop CS6.  
<http://www.adobe.com/ee/products/photoshop.html> (24.04.2017)
- [26] Material Design.  
<https://material.io/icons/> (17.04.2017)
- [27] FreeSound.  
<https://www.freesound.org/people/jobro/packs/2489/> (18.03.2017)

# Litsents

## **Lihtlitsents lõputöö reprodutseerimiseks ja lõputöö üldsusele kättesaadavaks tegemiseks**

Mina, Kristiina Kõiv,

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) enda loodud teose „Solfedžo õpitarkvara 1. klassile“, mille juhendajad on Alo Peets, Taavi Duvin ja Anne Villems,
  - 1.1.reprodutseerimiseks säilitamise ja üldsusele kättesaadavaks tegemise eesmärgil, sealhulgas digitaalarhiivi DSpace-is lisamise eesmärgil kuni autoriõiguse kehtivuse tähtaja lõppemiseni;
  - 1.2.üldsusele kättesaadavaks tegemiseks Tartu Ülikooli veebikeskkonna kaudu, sealhulgas digitaalarhiivi DSpace'i kaudu kuni autoriõiguse kehtivuse tähtaja lõppemiseni.
2. olen teadlik, et punktis 1 nimetatud õigused jäävad alles ka autorile.
3. kinnitan, et lihtlitsentsi andmisega ei rikuta teiste isikute intellektuaalomandi ega isikuandmete kaitse seadusest tulenevaid õigusi.

Tartus, **11.05.2017**