

TARTU ÜLIKOOL
Arvutiteaduse instituut
Informaatika õppekava

Johann Mattias Kammiste
Muusikaliste tugiradade mängija
Bakalaureusetöö (9 EAP)

Juhendaja:
Sven Aller, MSc

Tartu 2025

Muusikaliste tugiradade mängija

Lühikokkuvõte:

Tänapäeval esitatakse muusikat tihti koos tugiradadega. Kahjuks ei ole kättesaadaval odavaid ning töökindlaid variante, mille funktsionaalsus oleks võrreldav kallimatega. Antud bakalaureusetöö eesmärgiks oli luua rakendus, mille funktsionaalsus oleks samaväärne kallimate tugiradade seadmetega. Töös analüüsiti erinevaid tugiradade mängimise võimalusi ning selgitati välja kõige olulisemad komponendid. Programmi katsetati bändiproovis ning tagasiside põhjal muudeti programmi sisulist pool ning kirjeldati võimalikke edasiarendusi.

Võtmesõnad: Tugirada, Python, muusika

CERCS: P175 Informaatika, süsteemiteooria

Backing track player for live applications

Abstract:

Nowadays, music is often performed with backing tracks. Unfortunately, there are no cheap and accessible backtrack player options available whose functionality would be comparable to the more expensive ones. The aim of this bachelor's thesis was to create an application whose functionality would be comparable to the more expensive backing track machines. For this the thesis analyzed different options for playing backing tracks on the market and identified the most important components. The program was tested in a band rehearsal and based on the feedback, the content of the program was changed and possible further developments were described.

Keywords: Backing track, Python, music

CERCS: P175 Informatics, systems theory

Sisukord

Sissejuhatus.....	4
1. Tugiradade kasutus	5
2. Sarnased rakendused ja seadmed	6
2.1. Tavaline helifaili mängija	6
2.2. Apple MainStage	6
2.3. Ableton Live koos laiendustega	6
2.4. Cymatic Audio LP-16 ja uTrack24.....	7
2.5. iPad koos Multitracker programmiga	7
2.6. Loodava programmi kavandamine	7
3. Programmi ülesehitus.....	9
3.1. Programmi struktuur	9
3.2. Programmi <i>front-end</i>	10
3.3. Programmi <i>back-end</i>	12
3.4. Tehisintellekti kasutamine	13
4. Seadme loomine ning katsetamine.....	14
4.1. Seadme loomine.....	14
4.2. Seadme katsetamine.....	14
4.3. Tagasiside analüüs	14
5. Programmi edasiarendamine.....	16
6. Kokkuvõte.....	18
Viidatud kirjandus.....	19
Lisad.....	20
Litsents.....	22

Sissejuhatus

Muusika esitamine on arenenud võimenduseta sooloesitajatest suurte produktsioonidega maailmaturneedeni. Tihti, kui on soov oma muusika esitust täiendada, võetakse appi erinevaid inimesi, kellega musitseerida. Tänapäeva alternatiiv sellele on lisada enda etteastele taustamuusikat, mis mängib väliselt kandjalt sünkroonis esinejatega. Tugirada (ingl *backing track*) on väga levinud nii alustavatest ansamblitest kui ka suurte poppstaarideni. Kahjuks pole olnud veel head viisi, kuidas seda teha soodsamaks ja kättesaadavamaks, just alustavatele artistitele, mis pakuks ka kõiki põhivõimalusi, mida on olemas kallimatel variantidel. Nendeks on mitmete helifailide korraga taasesitamine, kavade haldus, väljundite muutmine ning põimimine.

Töö eesmärk on luua programm, mis aitab alustavatel ning hinnatundlikematel artistidel enda *live* šõudes kasutada tugiradasid. Programm ehitatakse nii, et seda saaks mugavalt kasutada eelkõige võimekama Raspberry Pi (või muu sarnase mikrokompuutri) peal, kuid on ka kasutatav nii Windows, macOS kui ka Linux operatsioonisüsteemidel. Rakenduse kavandamisel ning ehitamisel on arvestatud teiste sarnaste rakenduste ning seadmete võimaluste ja omadustega.

Esimeses peatükis kirjeldatakse, mis on tugirajad ning kuidas neid erinevates lavalistes ja muusikalistes kontekstides kasutatakse. Teises peatükis analüüsitakse sarnaseid rakendusi ja seadmeid ning kavandatakse sellest tulenevalt programmi vajalikud omadused. Kolmandas peatükis kirjeldatakse programmi ülesehitust. Selle käigus selgitatakse lahti ka töös kasutatud meetoodika. Neljandas peatükk kajastab, kuidas programm muuta eraldi seadmeks ning selle seadme katsetamist autori bändiproovis koos bändikaaslaste tagasisidega. Sellele järgnev peatükk pakub välja soovitusi ning võimalusi töö edasiarendamiseks. Töö lõpetab kokkuvõtte ja lisad.

1. Tugiradade kasutus

Selle töö eesmärk on aidata alustavatel muusikutel ja artistidel enda loomingut paremini tugiradadega ehk teisisõnu taustamuusika abil edastada. Sellisel moel musitseerimist kutsutakse ka põhja peale mängimiseks. Tugirada (ingl *backing track*) on helifail, mis koosneb tavaliselt ühest või mitmest erinevast instrumendist või rütmilistest helidest, mille saatel sünkroonis muusikud laval kaasa musitseerivad. Selle eesmärk on kontserdile anda lisaks kõlasid, mida oleks võimatu või ebapraktiline taasesitada, nendeks võivad olla näiteks kas keelpillikvarteti partii, mis on ette salvestatud stuudios, või muud võtmepartiid ja efektid, mida kuuldaja eeldab loo puhul kuulda.¹ Tänapäeval kasutavad tugiradu erinevad artistid, alustades alustavatest räpparitest kuni maailmastaarideni väljamüüdud maailmaturneedel. Danielsen ja Helseth [1] leidsid, et kuigi tugiradade kasutamine võib tekitada mulje, et kontserti ei esitata kohapeal elavalt, siis põhielementide esindatus laval on kuulaja jaoks rahuldav.

Juhul kui kasutatakse tugiradu bändikoosseisus, kus on olemas trummar, siis on oluline, et oleks olemas eraldi *click track* ehk metronoomi helirada, mida kuulevad ainult bändiliikmed, et nad oskaksid orienteeruda ja olla samas tempos teiste instrumentaalsete ja vokaalsete tugiradadega. Nende teiste tugiradade pealt võivad aga kõlada näiteks spetsiaalselt eelsalvestatud instrumendid, näiteks kitarrid, viiulid, ning ka vokaalid, mida kuulevad nii bändiliikmed kui ka publik. *Click track* ongi kõige levinum tugirada, mida kasutatakse just seetõttu, et hoida tempot. Kui kasutatakse ainult *click track*'i, siis piisab ka tavalisest metronoomist, millest tuleb heliväljund.

Tugirada võib koosneda ka mitmest erinevast failist. Sellisel juhul tihti soovitakse aga mingit süsteemi või programmi, kus neid saaks hallata ja nende jaoks ka erinevaid füüsilisi väljundeid määrata, et helitehnikul oleks võimalus neid vastavalt enda soovile ning bändi visioonile publikule helindada. Sellisel juhul on aga vaja rohkem füüsilisi väljundeid, mida ei saa näiteks tavalisest arvutist kätte ilma, arvutile oleks ühendatud külge helikaart, millel oleks palju heliväljundeid.

Sõnad *live*, *click* ning *backtrack* on muusikalised terminid, mida kasutatakse ülemaailmselt ning eesti keeles ei ole nendele tekkinud head vastet, mis oleks üheselt mõistetav.

¹ https://en.wikipedia.org/wiki/Backing_track, vaadatud 03.05.2025

2. Sarnased rakendused ja seadmed

Valmiva programmi eesmärk on abistada artistidel muusika esitamist elava esitusena ehk *live*'is ning seetõttu tuleks analüüsida ka sarnaseid rakendusi ja seadmeid, mis aitavad seda saavutada. Antud valikud on tehtud autori kogemuse põhjal, mida ta on näinud Eesti tippartistide kasutuses.

2.1. Tavaline helifaili mängija

Tavaliseks helifaili mängijaks saab olla kasvõi nutitelefon, kuhu on *playlist*'i pandud helifailid, mida soovitakse kontserdil mängida. Selle süsteemi kõige suuremad puudused on füüsiliste väljundite arv ning piiratud failidehaldus. Sellise mängija puhul on tavaliselt vasak kanal *click*'i väljund ning parem on esitatav lugu, kuid siis kaob kahjuks ära võimalus kasutada stereoefekte ning lugu ei kõla nii avaralt.

2.2. Apple MainStage

Apple Mainstage on programm, mis on loodud just *live*'is kasutamiseks². Tehniliselt on ta DAW ehk *Digital Audio Workstation* ehk digitaalne helitööjaam, mis on programm, mille eesmärk on helifailide salvestamine ning nende töötlemine. Tihti kasutatakse Mainstage'i klahvpillimängijad, et erinevaid süntesaatori- ja klaveriheliseid vallata ning neid kergelt vahetada, kuid sellel programmil on väga palju erinevaid võimalusi veel. Meie eesmärgi tarbeks on Mainstage'is võimalus kasutada Playback pistikprogrammi (ingl *plug-in*), kuhu saab importida erinevaid helifaile ning neile kõikidele määrata erinevaid efekte ning väljundeid. Selle programmi ainukene piiraja oleks sel juhul füüsiliste väljundite olemasolu, kuid mida rohkem füüsilisi väljundeid soovida, seda suuremat helikaarti on vaja. Lisaks on vaja kindlasti Apple'i arvutit, sest see töötab ainult macOS operatsioonisüsteemi peal ning Apple'i arvutid on tavaliselt kallimad kui teised arvutid.

2.3. Ableton Live koos laiendustega

Ableton Live on ka DAW, kuid selle suur eelis on see, et see töötab ka erinevatel operatsioonisüsteemidel. Et aga tugiradu saaks mugavalt *live*'is esitada ning hallata, on selle jaoks soovitatav osta juurde laiendus, mille eesmärk on kavade (ingl *setlist*) loomine ning

² <https://www.apple.com/mainstage/>, vaadatud 02.05.2025

nende haldamine. Üks neist on näiteks AbleSet, mille hinnaks on \$149 USD³. Muu funktsionaalsus sarnaneb Mainstage'iga.

2.4. Cymatic Audio LP-16 ja uTrack24

Cymatic Audio poolt toodetud LP-16 ja uTrack24 olid kaks seadet, mille ainus eesmärk oligi tugiradade taasesitamine. Nende ülesehitus oli lihtne ning olid väga kasutajasõbralikud. LP-16 on väiksem, omab 16 TRS väljundit⁴, kuid uTrack24 on 1U suurune, omab 24 väljundit koos XLR pesadega⁵. Esitatavad lood saab teha eraldi programmiga, kus saab ka määrata, millisest väljundist tuleb mis helifail. Paljud Eesti tippartistid kasutavad siiani neid seadmeid, kuid kahjuks aga lõpetati nende seadmete tootmine ära ning järelturul ei ole neid ka enam palju järel. Tootmise lõpu põhjuseks oli emafirma REIDEL-i sulgemine ning sellega kaasnevalt ka Cymatic Audio. [2]

2.5. iPad koos Multitracker programmiga

Multitracker on programm, mis on loodud helifailide taasesitamiseks *live* kontekstis. Selle jaoks on vaja iPadi ning ka soovi korral välist helikaarti. Idee on sama mis Cymatic Audio toodetel, et saad luua kavasid ning laule, millel saab muuta ka väljundeid vastavalt soovidele. Programmi saab alla laadida tasuta, kuid kõikideks vajaminevateks funktsioonideks on vaja see osta⁶.

2.6. Loodava programmi kavandamine

Käesolevas töös kasutatakse ära eelnevalt kirjeldatud rakenduste ja seadmete analüüsi käigus ette tulnud võimalused, et kujundada kasutajatele võimalikult palju kasutoov programm tugiradade esitamiseks. Lisaks on autor lisanud enda poolt mõned vajadused, millest arvab, et kasutajal oleks kasu. Analüüsi põhjal on vajalikud funktsioonid järgnevad:

- Helifailide importimine ning levinumate failitüüpide toetamine (MP3, WAV, AIFF jms);
- kavade loomine ning nende haldamine;
- heliväljundite määramine ning põimimine;

³ <https://ableset.app/>, vaadatud 02.05.2025

⁴ <https://cymaticaudio.com/product/lp-16-live-player-16-track-backing-track-system/>, vaadatud 03.05.2025

⁵ <https://cymaticaudio.com/product/utrack24-live-recorder-player-interface/>, vaadatud 03.05.2025

⁶ <http://www.multitracker.eu/>, vaadatud 03.05.2025

- töökindel ning mitut operatsioonisüsteemi (Windows, macOS, Linux) toetav;
- kergesti mõistetav ning kasutajasõbralik.

3. Programmi ülesehitus

See peatükk kirjeldab programmi struktuuri, tehnilisi vahendeid ning kasutatud metoodikat. Samuti on välja toodud programmi erinevad vaated ning sisemeetodite töö. Programm luuakse esialgselt inglise keeles, et keskenduda ka välismaisele kasutajaskonnale. Valminud rakendusega saab tutvuda GitHubi aadressil <https://github.com/johannmkammiste/btplayer>, kus on ka õpetus, kuidas programmi ise kasutada.

3.1. Programmi struktuur

Kavandataval programmil peab olema eesrakendus, millega kasutaja suhtleb, ning tagarakendus, kus toimub kogu tehniline töötlus. Kahel osal on ka tarvilik omavaheline suhtlus. Seega saab programmi jagada kolmeks osaks:

- *front-end* (eesrakendus), mis hõlmab endas kasutajaga suhtlust kasutades HTML-i (ingl *Hypertext Markup Language*), JavaScripti ning CSS-i (ingl *Cascading Style Sheets*);
- *back-end* (tagarakendus), mis hõlmab endas helifailide töötlust, esitamist ning muid mäluoperatsioone Pythonis;
- ees- ning tagarakenduse vaheline suhtlus Flaskis.

Selleks, et *front-end* ja *back-end* saaksid omavahel suhelda, kasutatakse ühendusülina veebiraamistiku Flaski, mis põhineb Werkzeugi WSGI-1 ja Jinja2-1 ning on loodud olema kiiresti ja lihtsalt kasutatav. Flaskiga saab kutsuda välja Pythoni funktsioone kasutaja suhtlemisel veebilehega [3]. Rakenduse sisu kuvamiseks kasutatakse moodulit pywebview, mis käivitab uue brauseri akna, mis on ülejäänud operatsioonisüsteemi brauseritest eraldatud⁷. Programm on mõeldud toimima individuaalselt ühe arvuti peal, mis tähendab, et kuigi on tegu nii-öelda veebilehega, siis programmi funktsionaalsus tuleneb arvutist, mille peal see on tööle pandud. Programmi kasutamiseks on nõutud suhtlus veebilehega ning heliväljundi(te) olemasolu, kuid programmi kavandamisel on mõeldud ka selle peale, et saaks põimida erinevaid heliväljundeid kokku. See tähendab, et ei ole vaja mitte ühte suurt helikaarti, vaid võib ka olla mitu erinevat väikest helikaarti ning kõikidele nendele saab määrata oma enda väljundkanal.

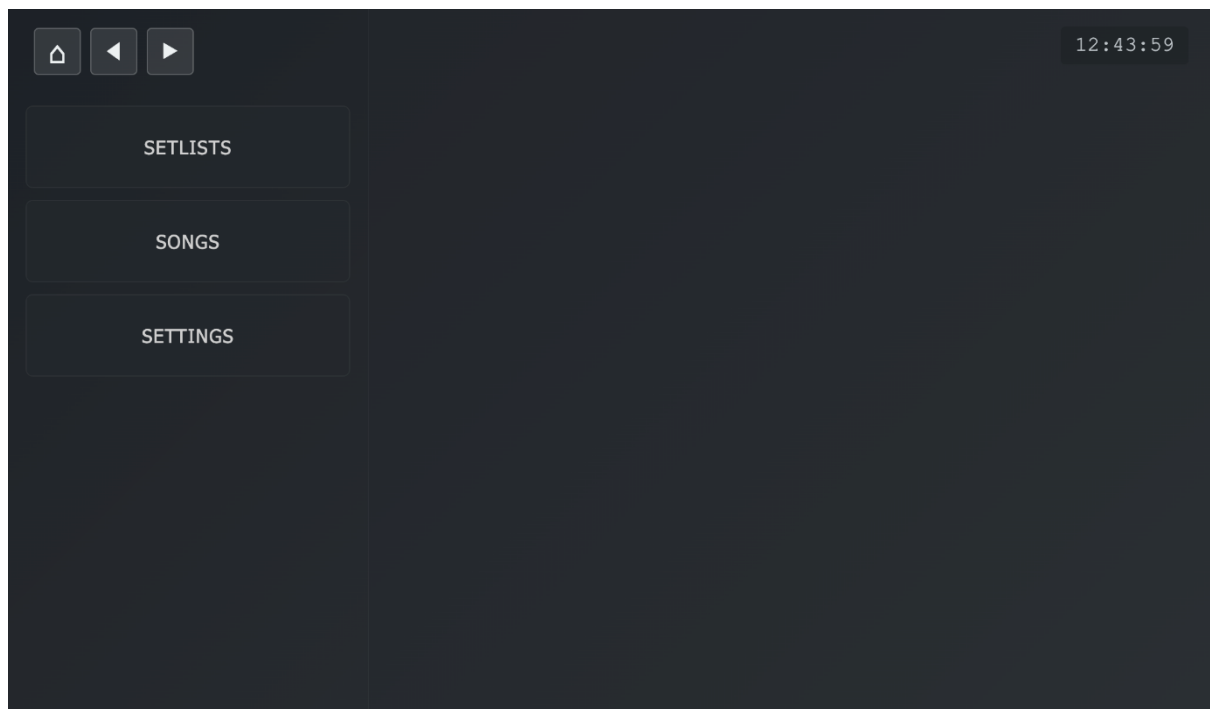
⁷ <https://pywebview.flowrl.com/>, vaadatud 14.05.2025

3.2. Programmi *front-end*

Front-end tegutseb veebilehena, mille kaudu saab kasutaja programmiga suhelda. *Front-end*’iks valiti HTML, JavaScript ning CSS, sest seda on lihtne erinevatele ekraanisuurustele kohendada ning toetab erinevaid operatsioonisüsteeme. Stack Overflow eelmise aasta küsitluse järgi on need keeled kõige enam levinumad keeled kõikide vastajate vahel [4]. Programmi kujundus on lihtne, et keskenduda programmi funktsionaalsusele, mitte niivõrd välimusele. Veebilehed on alati jaotatud kaheks osaks – vasakul pool saab teha valikuid ning paremal pool on sisuline osa. Programmi on võimalik ka kontrollida klaviatuuri abil.

Programmi tööle pannes avaneb esimese vaadena veebileht, kus on näha erinevaid asju (vt joonis 1):

- *SETLISTS* nupp, kust avaneb kavade vaade;
- *SONGS* nupp, kust avaneb lugude vaade;
- *SETTINGS* nupp, kust avaneb seadete vaade.
- 3 navigatsiooni nuppu, mis on vastavalt kodu ehk peavaade, tagasi ning edasi;
- üleval paremal nurgas on ka reaalaajas kell.



Joonis 1. Programmi pealeht.

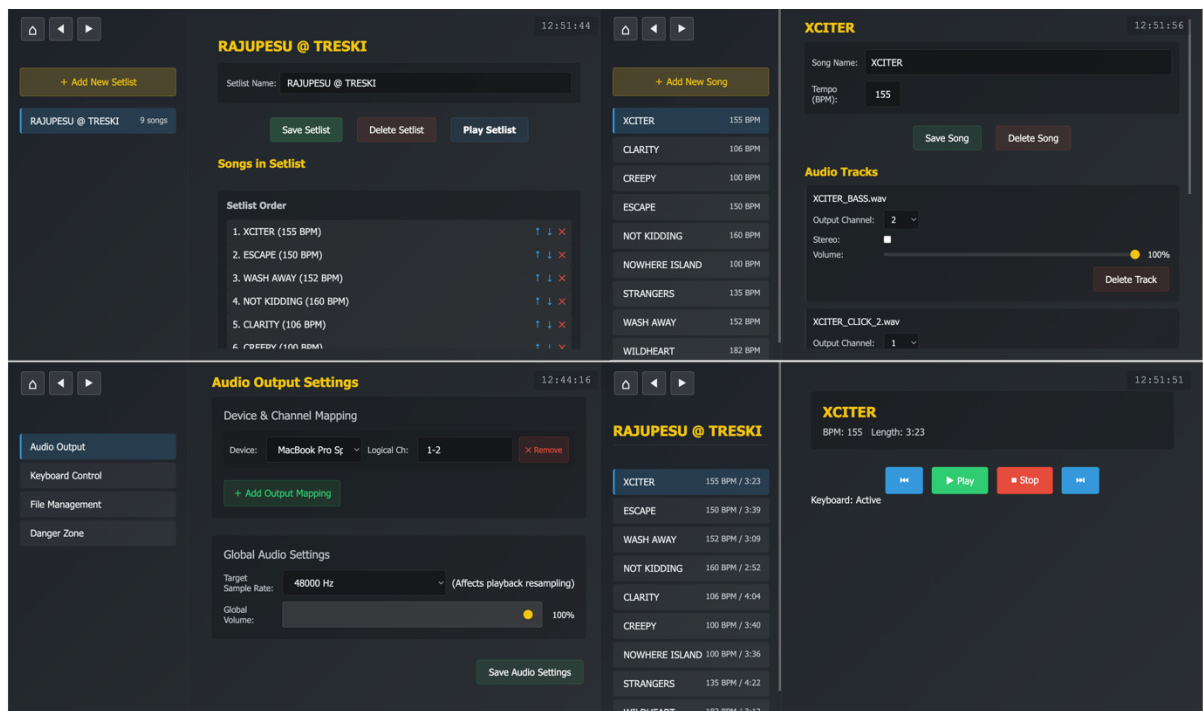
Kolmele põhinupule vajutades saab ligi erinevatele programmi funktsioonidele. *Setlists* vaates on vasakul poolel näha uue kava loomise nuppu ning selle alla tekivad ka kõik kavad, mida on kasutaja salvestanud. Uue kava nupule vajutades või olemasoleva kava valimisel avanevad paremal poolel selle kava valikud (vt joonis 2). Seal saab muuta kava nime, seda salvestada, kustutada, esitada ning ühtlasi lugusid lisada ning nende järjekorda muuta.

Songs vaates ilmneb sarnane pilt nagu ka kavade vaates – seal leidub *Add New Song* nupp ning ka lugude loetelu. Klakkides emmale-kummale avaneb vaade, kus saab muuta laulu nime, tempot ning lisada sellele helifaile. Helifaile lisades saab muuta iga helifaili suhtelist helitugevust, selle helikanali väljundit ning määrata ka, kas faili esitamine on stereos või monos. Ühtlasi on võimalik ka laule kustutada ning salvestada muudatusi.

Settings lehel asetsevad programmi seaded. Vasakul poolel on näha:

- heliväljundite seadeid, kus saab muuta diskreetimissagedust, globaalset helitugevust ning valida ka seadmeid, millest heli hakkab kõlama, ühtlasti saab valida mitut erinevat heliseadet ning valida mis heliseadmest kõlavad millised määratud loogilised kanalid;
- klaviatuuriga juhtimise seadeid, kus saab muuta klaviatuuri otseteid, näiteks seda, et klahv Enter on *Play* nupp;
- failihaldust, kus saab avada failide kausta;
- ohtlikku tsooni, kus saab taastada programmi tehaseseaded ning ka kustutada kõik laulud ning kavad.

Kavade vaates *Play Setlist* nupule vajutades avaneb kavade esitamiskaart. Vasakul poolel on lood selles järjekorras, mis kasutaja määras, ning paremal pool on näha laulu nime, selle tempot, kestvust ning *PLAY*, *STOP*, eelneva laulu ning järgneva laulu nupp. Nende nuppude funktsioone saab kontrollida ka klaviatuuriga.



Joonis 2. Programmi erinevad vaated. Vasakult: kavade, lood, seadmete haldus ning esitamine.

3.3. Programmi *back-end*

Kogu programmi loogiline ning funktsionaalne osa toimib Pythoniga ning *front-end*'iga suhtlemisel kasutatakse Flask raamistikku. Python valiti, kuna see on helifailide töötlemisel üks parimaid ning populaarsemaid keeli [5]. Põhilise töö teeb ära teek sounddevice, mille eesmärk on heli töötlemine ning esitamine, kasutades vooge (ingl *stream*). Neid vooge saab ka määrata erinevatele heliseadmetele, mis oli ka üks töö peamise kriteeriumeid.

Programmi *back-end*'i ülesehitus on järgnev: Pythoni ning Flaski funktsionaalsus asub `app.py` failis, HTML failid *templates* kasutas, programmi mälu asub *data* kaustas ning JavaScript, CSS ja salvestatavad helifailid asuvad *static* kaustas. Programmi mälu salvestatakse JSON failidesse, kuna nii Python kui ka JavaScript oskavad seda hästi lugeda.

Kõik kasutaja väljakutsutud funktsioonid käivad läbi Flaski, mis seejärel saab Pythoni koodi käivitada. Neid tehakse läbi */api* kutsete. Kui kasutaja lisab uue laulu, siis kutsub ta välja kutse, mis Pythonis salvestab selle laulu koos vajaliku infoga *songs.json* faili, kust seda saab hiljem tagasi lugeda. Sarnaselt töötavad ka kavade ning seadete salvestamine, vastavalt *setlists.json* ning *settings.json* failidesse.

Programmi haldab 3 erinevat Python faili. Nendeks on *run.py*, *app.py* ning *audioplayer_module.py*. *Run.py* haldab programmi jooksumist ning kutsub välja *app.py*, mis

omakorda kutsub välja *audioplayer_module.py*, kus asetseb helifailide esitamise loogika. Failide haldamine ning kogu ees- ja tagarakenduse suhtlust haldab *app.py*. Pywebview moodulit kasutab *run.py*, mis teeb lihtsaks programmi jooksutamise.

3.4. Tehisintellekti kasutamine

Nii ees- kui tagarakenduse loomisel on kasutatud tehisintellekti abi. TI kasutamise eesmärk oli aidata töö autoril luua baasfailid ning baasfunktsioonid, et hõlpsamini tegeleda töö sisulise poolega. Viipa „Make me basic front-end templates for my backing track player app. I am developing it using Flask and Python. Make sure CSS is separate from HTML. I want the page to be divided into 2 parts: left and right side. Left side is a navigation menu and shows options and right side is where the content is“ kasutati, et luua algeline HTML ning CSS (Google Gemini 2.5 Pro (preview)). Viibaga „Make me JavaScript and CSS for modals on my page (the notifications that pop up when I do an action, so a global notification)“ saadi teavituste aknad, mis tulevad ülesse, kui vajutada näiteks „*Save Audio Settings*“ (Google Gemini 2.5 Pro (preview)). Ühtlasi kasutati viipa „Help me make a backend for the program using sounddevice. Make sure multiple audio files can be played on the same device.“, et luua esialgne tagaprogrammi kood (Google Gemini 2.5 Pro (preview)). Nende algsete failide ning koodipõhja peale ongi programm ehitatud.

4. Seadme loomine ning katsetamine

See peatükk kirjeldab, kuidas programm muuta eraldiseisvaks seadmeks Raspberry Pi-d või muud sarnast mikrokompuutrit kasutades ning selle katsetamist autori bändiproovis koos bändikaaslaste tagasisidega.

4.1. Seadme loomine

Töö eesmärk oli programmile leida reaalne kasutus ning selle realiseerimiseks on tarvis programmile teha eraldi seade, mille eesmärk ongi heli tagasiesitamine *live*'is. Selle jaoks kasutas autor Asus Tinker Board 2 koos 7 tollise puutetundliku ekraani ning kahe väikese USB helikaardiga. Õpetus, kuidas sarnaselt seda järgi teha, leiab GitHubi lehelt ning ka lisast 2. Kokku läks see autorile maksma alla 150€, mis on märgatavalt vähem kui muud sarnase funktsionaalsusega süsteemid.

4.2. Seadme katsetamine

Seadet katsetati ansambliga Rajupesu, mis viljeleb räpp-rokki. Kuna ansambli bassimängija viibib ülikooliõpingute tõttu Inglismaal, oli see hea võimalus ülejäänud bändimeestel prototüüpi katsetada. Autor oli loonud kava, mis koosneb 9 laulust ning iga laul koosneb 3 failist: *click*-fail (monos), basskitarri fail (monos) ning muud instrumendid (stereos). Metronoomi fail ning basskitarri fail said vastavalt väljundid 1 ja 2 ning muud instrumendid said väljundid 3 ja 4. Seade võeti proovi kaasa ning bänd ühendas seadme ning alustati prooviga. Lugu alustades selgus suur probleem, et kuna kasutati kahte erinevat helikaarti, siis nende digitaalne kellaajastus ei sünkroniseerunud, mis tõttu nende helid mängisid erinevatel hetkedel. Digitaalne kellaajastus (ingl *digital clocking*) tagab, et analoogse signaali diskreteerimine (ingl *sampling*) digitaalseks toimuks täpsetel ja ühtlastel intervallidel, et kindlustada täpne muundamine ning taasesitamine [6]. Proovis lahendati see mure sellega, et rakendust kasutati hoopis macOS operatsioonisüsteemi kasutava arvutiga, sest sellel operatsioonisüsteemil on võimalik luua virtuaalne heliseade, mis suudab mitme heliseadme digitaalseid kellaajastusi sünkroniseerida.

4.3. Tagasiside analüüs

Proovis jäid bändikaaslased programmi visuaalse ning tehnilise poolega rahule. Mõlemad tagasisidestajad leidsid, et programm oli kergesti kasutatav ning üheselt mõistetav. Üks tagasisidestajatest arvas, et programmi visuaalne pool oli lihtne ja selge. Ta tõi ka välja, et bassimängija puudulikkust suutis rakendus efektiivselt asendada. Autor palus bändikaaslastel

programmi ka kodus katsetada. Üks tagasiside andjatest raporteeris, et tal esines raskusi Windowsil programmi kasutamisel ning soovis näha programmist väljumise nuppu. Sellele probleemile leidis suhteliselt kerge lahendus - pywebview moodul võimaldab akent sulgeda ning šsellega kaasnevalt ka ülejäänud programmi peatada. Tavalise veebibrauseri puhul sulgeb rakendus programmi brauseri akna, kuid ei peata programmi. Teine tagasisidestaja märkis, et talle meeldis, kuidas rakendus toetas mitut erinevat operatsioonisüsteemi (vt lisa 1).

Tagasisidest ning proovi kogemusest tulenevalt otsustas autor, et *back-end* vajab suuri muudatusi, et ei oleks tarvilik virtuaalset heliseadet ise luua. Virtuaalset heliseadet oleks vaja luua juhul, kui kahe helikaardi digitaalsed kellaajastused ei ühti.

5. Programmi edasiarendamine

Selles peatükis on välja toodud võimalused programmi ning seadme edasi arendamiseks. See põhineb bändiproovis tekkinud tagasisidel ning ka töö käigus tekkinud ideedel ja probleemidel.

Heliväljundite sünkroonimise probleemi lahendamiseks pidi autor sounddevice moodulist loobuma. Selle asemel leidis ta vabavaralise lahenduse nimega pybass, täpsemalt modpybass moodul, mis interpreteerib BASS-i. BASS on heliteek, mis on mõeldud kasutamiseks tarkvaras mitmel platvormil. Selle eesmärk on pakkuda võimsaid ning tõhusaid funktsioone sãmplite, voogude ja muusika salvestamise jaoks. [7] Probleemi lahenduseks pakub BASS lisateeki BASSMIX, mis võimaldab erinevaid heliseadmeid sünkroniseerida. Pythonis realiseerimiseks kasutatakse moodlit modpybass [8], mis võimaldab hõlpsasti kasutada heliteegi funktsioone. Kahjuks avastati, et selle toetus ARM protsessoritega macOS arvutitele ning Linux arvutitele oli puudulik ning seetõttu pidi moodulit uuendama. Uuendatud mooduliga saab lähemalt tutvuda ning seda ka alla laadida GitHubist lingil <https://github.com/johannmkammiste/modpybassmacos>. Kasutati ka viipa „Convert my existing backend to work with pybass, make sure multi-device is still supported and synchronize the clocks!“, et aidata muuta vana sounddevice'i põhine *back-end* uuele modpybass põhisele (Google Gemini 2.5 Pro (preview)).

Kuna autoril esines probleeme pywebview mooduliga Linuxil, siis lõi ta ka võimaluse kasutada programmi operatsioonisüsteemile allalaetud veebibrauseritega. Selleks on olemas eraldi fail *kiosk.py*, mis jooksub ainult *back-end*'i ning *front-end* tuleb ise avada. Seda lahendust kasutas autor ka oma viimases katsetuses, et luua täiesti eraldiseisev seade, mis käivitaks programmi sel hetkel, kui arvuti tööle läheb. Ühtlasi alustab siis ka operatsioonisüsteem kohe veebibrauseri õige veebiaadressiga.

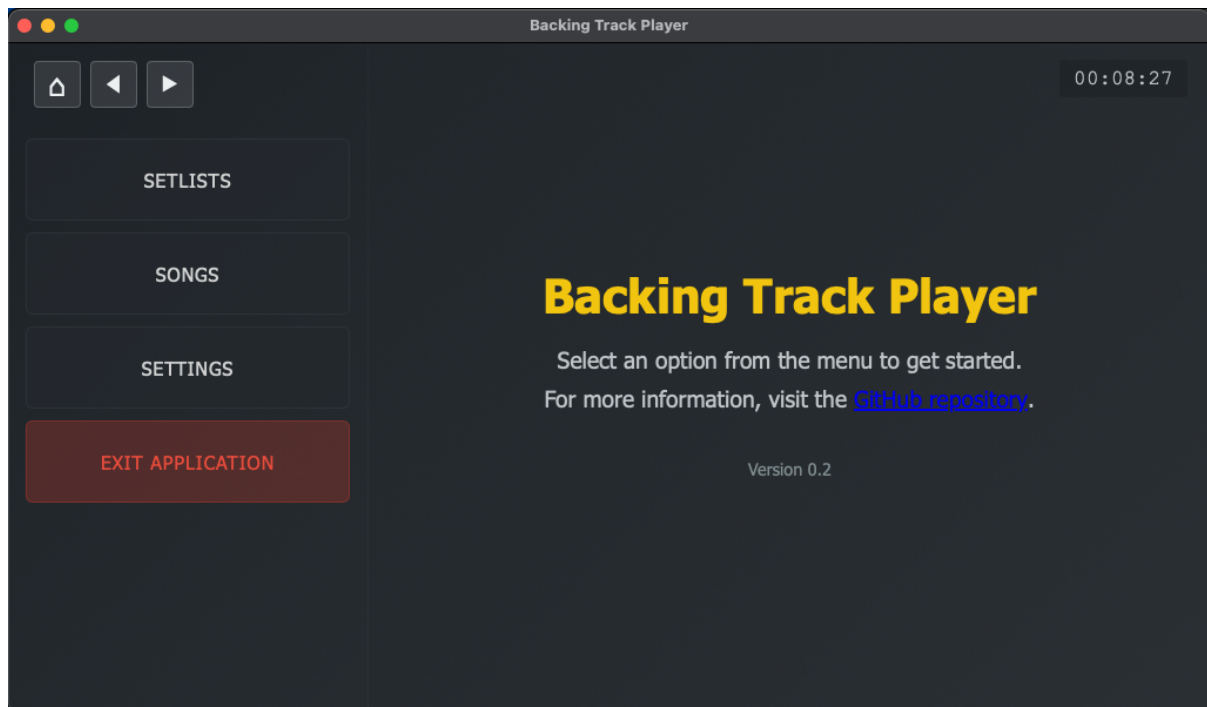
Programmile oleks võimalik lisada veel juurde juhtimisvõimalusi. Üks variantidest oleks Raspberry Pi või muude mikrokompuutrite mitmeotstarbelist sisend/väljundit ehk GPIO'd (ingl *general purpose input/output*) kasutada. Pythonis on nende jaoks loodud palju erinevaid moduleid, näiteks RPi.GPIO, mis on mõeldud just nimelt Raspberry Pi'ga kasutamiseks⁸. Sellisel juhul saaks lisada nuppe, mis näiteks alustaksid või vahetaksid lugu. Sellist sarnast lahendust on kasutanud Jorge, et luua kitarrri efektipedaali laadne toode, mis mängib kontsertidel MP3 faile⁹. Teine võimalus oleks lisada üldine MIDI funktsionaalsus. Sellisel

⁸ <https://pypi.org/project/RPi.GPIO/>, vaadatud 15.05.2025

⁹ <https://hackaday.io/project/7278-mp3-pedal-for-gigs>, vaadatud 15.05.2025

juhul saaks näiteks DJ kontrollritega või muude MIDI seadmetega juhtida programmi tööd. Lisaks saaks võimaldada bändisisestel pillide helieelseadete (ingl *preset*) muutmist vastavalt loole, kui teiste bändiliikmete tehnika seda toetab. Näiteks saaks iga looga saata ühe kindla MIDI signaali, mis vahetaks klavnpillimängija klavnpillil ning kitarristi kitarrisignaali protsessoril *preset*'i, mis sobiks antud looga.

Lõplik programmi koduleht sisaldab nüüd ka lühikest tutvustust (vt joonis 3).



Joonis 3. Valminud programmi koduleht uuendustega.

Viimased edasiarendused oleks Androidi ning IOS'i toetamine, teiste keelte, näiteks eesti keele, toetamine. Viimane võimalus, et programm teha raudkindlaks, kirjutada programm keeles C++¹⁰, mis oleks kõige töökindlam arvestades asjaolu, et paljud kui mitte enamus helifirmade tooted kasutavad JUCE raamistikku, mis on kirjutatud C++'is¹¹.

¹⁰ <https://audiodev.blog/newbie-resources/>, vaadatud 03.05.2025

¹¹ <https://juce.com/>, vaadatud 14.05.2025

6. Kokkuvõte

Töö eesmärgiks oli luua programm ning ka seade, mis aitaks muusikutel kontsertidel muusikaliselega põhjaga ehk tugiradadega koos mängida. Töö käigus analüüsiti sarnaseid rakendusi ning kasutati nende parimaid omadusi uue programmi loomisel.

Võrreldes sarnaste rakendustega on programm üldiselt lihtne, võimekas, kasutajasõbralik ning tasuta. Programm on kirjutatud Pythonis ning kasutab BASS heliteeki failide taasesitamiseks ja Flask'i, et luua sellele programmile *front-end*. Ühtlasi on see mõeldud erinevatel platvormidel töötamiseks ning seetõttu ei ole piiranguid arvutite operatsioonisüsteemile. Programmi testimiseks loodi seade, mille eesmärk oli ainult antud programmi jooksutada. Selleks kasutati ASUS Tinker Board 2'te koos puuetundliku ekraani ning kahe väikese USB helikaardiga, mille kogu maksumus tuli alla 150€.

Programmi ning seadet testiti räpp-rokk bändi proovi kontekstis, kus bändiliikmed said anda tagasisidet. Bändikaaslastele meeldis programmi funktsionaalsus ning tõsteti esile programmi kasutajaliidese selgust ning kasutatavust. Tagasisides toodi välja ka puudujäägid ning ka edasiarendamise võimalused. Puudujäägid said kõrvaldatud ning edasiarendamise võimalusteks on näiteks GPIO ja MIDI toetamine, teiste keelte, näiteks eesti keele, toetamine ning C++'ile üleminek. Kõik sellele tööle püstitatud eesmärgid said täidetud ning realiseeritud.

Viidatud kirjandus

- [1] Danielsen, A; Helseth, I. Mediated Immediacy: The Relationship between Auditory and Visual Dimensions of Live Performance in Contemporary Technology-Based Popular Music. *Rock Music Studies*. 2016, 3 (1), 24-40. <https://doi.org/10.1080/19401159.2015.1126986>
- [2] Malkowski B. Cymatic Audio Closes: What Options Exist for Playback Techs? 2022. <https://www.gearnews.com/cymatic-audio-closes-what-options-exist-for-playback-techs/> (14.05.2025)
- [3] Flask. <https://flask.palletsprojects.com/en/stable/> (03.05.2025)
- [4] Stack Overflow 2024 Developer Survey. <https://survey.stackoverflow.co/2024/technology> (03.05.2025)
- [5] Wilczek, J. Top 5 Languages for Audio Programming. 2023. <https://thewolfound.com/top-5-languages-for-audio-programming/> (03.05.2025)
- [6] What is Clocking? <https://audient.com/tutorial/what-is-clocking/> (15.05.2025)
- [7] BASS. <https://www.un4seen.com/> (13.05.2025)
- [8] modpybass. <https://github.com/peppy0510/modpybass> (13.05.2025)

Lisad

1. Bändikaaslaste tagasiside programmile

„Rakendust oli kerge ja mugav kasutada ning selle kasutamine oli lihtsasti arusaadav. Windowsi peal esines rakenduse jooksumisega mõned probleemid ning näiteks ei leidnud ma viisi kuidas rakendust sulgeda selle siseselt, ilma alt + f4 kasutamata. Kui rakenduse sain käima, täitis see protsessi, mida sellelt oli oodata täiuslikult.“ – Bändikaaslane 1

„Bändilugude backtracki rakendus oli üllatav, kuid teretulnud lisand meie viimasesse bändiproovi. Kuna meie bändis hetkel puudub bassimees, oli selline efektiivne rakendus heaks asenduseks, et bändiproovid saaksid siiski toimuda. Rakendus tundus UI poolelt „puhas” ning kergesti mõistetav. Samuti oli meeldiv näha, et rakendus töötas mitme erineva OSiga.“ – Bändikaaslane 2

2. Õpetus, kuidas luua eraldiseisev seade

Vaja on: mikrokompuutrit (Raspberry Pi, Asus Tinker Board vms), SD-kaarti, puutetundliku ekraani, klaviatuuri, internetiühendus.

1. Lae alla DietPi (<https://dietpi.com>) ning järgi selle õpetust, kuidas seda oma mikrokompuutrile installeerida. Kasuta vaikimisi *root* kasutajat.
2. Lülita sisse helimoodul ning *dietpi-software*'ist lae alla chromium, python3, git ning xcfе.
3. Laadi alla GitHubist programm ning kasuta neid koodikäske, et kõik valmis seadistada.

```
wget https://github.com/johannmkammiste/btplayer/archive/refs/heads/main.zip
```

```
unzip main.zip
```

```
cd backingtrackplayer
```

```
apt install python3-venv
```

```
python3 -m venv venv
```

```
source venv/bin/activate
```

```
pip install -r requirements.txt
```

```
mv chromium-autostart.sh /var/lib/dietpi/dietpi-software/installed/chromium-autostart.sh
```

```
dietpi-autostart 11
```

4. Tee arvutile restart ning arvuti peaks koheselt avama programmi.

Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Johann Mattias Kammiste

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose Muusikaliste tugiradade mängija, mille juhendaja on Sven Aller, reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada Tartu Ülikooli digitaalarhiivi kuni autoriõiguse kehtivuse lõppemiseni;
2. annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi kaudu Creative Commons'i litsentsiga CC BY NC ND 4.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni;
3. olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile;
4. kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Johann Mattias Kammiste

15.05.2025