

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Žiga Vučko

**Poigritev procesa poučevanja in
učenja glasbene teorije**

MAGISTRSKO DELO

MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: prof. dr. Saša Divjak

SOMENTOR: doc. prof. dr. Peter Šavli

Ljubljana, 2018

AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

©2018 ŽIGA VUČKO

ZAHVALA

Za vsa strokovno podporo in pomoč pri izdelavi magistrskega dela bi se zahvalil izr. prof. dr. Matiji Maroltu, prof. dr. Saši Divjaku, doc. prof. dr. Petru Šavliju, as. dr. Matevžu Pesku in viš. pred. dr. Alenki Kavčič. Zahvalil bi se še materi Poloni in očetu Darku, ki sta me spodbujala skozi vsa leta izobraževanja. Prav tako pa bi se zahvalil Tini ☆ za pomoč pri oblikovanju izgleda aplikacije in vse spodbude tekom pisanja magistrskega dela.

Žiga Vučko, 2018

V spomin, čast in slavo Gaiusu Juliusu,
Claudii Octavii, Cleopatri II in Lii.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Problem	1
1.2	Cilji in ideja	1
1.3	Struktura dela	3
2	Pregled sorodnih del	5
2.1	Poigritev	5
2.2	Sorodne rešitve za učenje glasbene teorije	13
3	Uporabljene tehnologije in orodja	21
3.1	Laravel	21
3.2	Sass	22
3.3	Vue.js	23
3.4	Timer Progress	27
3.5	MIDI	28
4	Mobilna aplikacija	31
4.1	Tehnični opis	31
4.2	Podatkovna baza	33
4.3	Uporabniški vmesnik	35
4.4	Igra intervalov	39

KAZALO

4.5	Poigritev aplikacije	46
4.6	Modul za učitelje	48
5	Rezultati in analiza	51
5.1	Intervju	51
5.2	Testiranje aplikacije	53
6	Sklepne ugotovitve	71
A	Primeri programske kode	75
A.1	Jezik stilskih predlog Sass (sintaksa SCSS) z uporabo notacije BEM	75
A.2	Vzpostavljanje ogrodja knjižnice MIDI.js	76
A.3	Logika zagona in ustavljanja predvajanja zaporedja tonov . . .	76
A.4	Inicializacija in zagon odštevalnika časa knjižnice Timer Progress	77
A.5	Algoritem za generiranje vprašanj pri igri intervalov	78
B	Vprašalniki	81
B.1	Intervju	81
B.2	Vprašalnik o profilu	82
B.3	Vprašalnik o aplikaciji	83
B.4	Vprašalnik za kontrolno skupino	84

Seznam uporabljenih kratic

kratica	angleško	slovensko
API	Application programming interface	aplikacijski programski vmesnik
BEM	Block Element Modifier	blok-element-določilo
CSS	Cascading Style Sheets	kaskadne stilske predloge
DOM	Document Object Model	objektni model dokumenta
HTML	Hyper Text Markup Language	jezik za označevanje nadbesedila
JSON	JavaScript Object Notation	objektni zapis JavaScript
MIDI	Musical Instrument Digital Interface	digitalni vmesnik glasbenih inštrumentov
MVC	Model-view-controller	model-pogled-kontroler
NPM	Node Package Manager	upravitelj paketov Node
PHP	PHP: Hypertext Preprocessor	PHP: predprocesor nadbesedila
REST	Representational State Transfer	predstavitveni prenos stanja
SPA	Single-page application	aplikacija z eno stranjo
SQL	Structured Query Language	strukturirani povpraševalni jezik
SVG	Scalable Vector Graphics	razširljiva vektorska grafika
URL	Uniform Resource Locator	enolični krajevnik vira
W3C	World Wide Web Consortium	konzorcij svetovnega spleta
XML	Extensible Markup Language	razširljiv označevalni jezik

Povzetek

Naslov: Poigritev procesa poučevanja in učenja glasbene teorije

Proces poučevanja in učenja glasbene teorije v Sloveniji še vedno poteka na zelo konvencionalen način: z inštrumentom, papirjem in svinčnikom. Kljub razvoju informacijske tehnologije v zadnjih desetletjih se to ni bistveno spremenilo.

Za naslovitev tega problema smo se osredotočili na implementacijo mobilne interaktivne personalizirane izobraževalne aplikacije za poučevanje in učenje solfeggia, ki je kot element glasbene teorije prisoten na vseh nivojih glasbenega šolstva. Aplikacijo smo obogatili z elementi poigritve, kot so značke, nivoji, točke in lestvice, obenem pa smo implementirali igro za učenje intervalov. Naša glavna raziskovalna usmeritev je bila preverjanje učinkov poigritve na proces poučevanja in učenja glasbene teorije, zato smo z učenci Konservatorija za glasbo in balet v Ljubljani izvedli testiranje, rezultate pa nato ovrednotili. Rezultati kažejo, da je poigritev težak proces, a vendar nam je s pomočjo elementov poigritve do določene mere uspelo angažirati uporabnike, prav tako pa se je potrdila naša predpostavka, da si učenci želijo drugačne pristope k učenju glasbene teorije, kot so jim na voljo sedaj.

Aplikacija kot pripomoček k izvajanju učnega procesa glasbene teorije kaže velik potencial. Njena ključna prednost je, da učenci lahko vadijo sami, kadarkoli imajo čas in niso vezani na samo prisotnost v razredu. Z nadaljnjim aktivnim razvojem bi aplikacija lahko postala pomemben katalizator zanimanja za področje glasbe v Sloveniji.

Ključne besede

e-izobraževanje, mobilna aplikacija, poigritev, učenje glasbene teorije, solfeggio

Abstract

Title: Gamification of the Music Theory Teaching and Learning Process

The process of teaching and learning music theory in Slovenia is still performed in a very conventional way: through instrument, paper and pencil. Despite the development of information technology in recent decades, this has not changed significantly.

In order to address this problem, we focused on the implementation of a mobile interactive personalized educational application for teaching and learning music theory, especially ear-training which is present at all school levels. We enriched the application with gamification elements, such as badges, levels, points and leaderboards. At the same time we implemented a game for learning intervals. In our research we were trying to discover how gamification affects the process of teaching and learning music theory. To that purpose we carried out a couple of testing sessions with the students of the Ljubljana Music and Ballet Conservatory and evaluated the results afterwards. They indicate that gamification is a difficult process, however we managed to engage a certain amount of users with the help of the gamification elements. Moreover, our assumption that students want different approaches to learning music theory, proved to be correct.

The application shows great potential as a helper tool for the teaching and learning process of music theory. Its main advantage is that students can practice on their own whenever it is suitable for them and they are not dependent only on classroom work. With further active development, the application could become an important catalyst of interest in the field of

music in Slovenia.

Keywords

e-learning, mobile application, gamification, music theory learning, ear-training

Poglavje 1

Uvod

1.1 Problem

Proces učenja glasbene teorije v Sloveniji je s hitrim razvojem računalniške in informacijske tehnologije v zadnjih desetletjih doživel določene napredke. Kljub temu še vedno ne obstaja primerno interaktivno izobraževalno okolje za področje glasbene teorije, temveč le nabor aplikacij, ki naslavljaajo specifične probleme s tega področja, nobena izmed njih pa ne ponudi celostne rešitve v obliki nadgradljive in personalizirane izobraževalne platforme. Večina aplikacij ponuja zgolj glasbene vadnice, katerih naloge so slabo prilagojene učnemu programu slovenskih glasbenih šol [1, 2].

1.2 Cilji in ideja

V delu smo poskušali nasloviti ta manjko in se osredotočiti na izdelavo izobraževalnega orodja za poučevanje in učenje glasbene teorije. Poseben podatek smo pri tem namenili poigritvi učnega procesa, zato je bilo glavno raziskovalno vprašanje, ki smo si ga zastavili: “Kako poigritev vpliva na proces poučevanja in učenja glasbene teorije?”. Raziskovanje vplivov poigritve na proces učenja je relativno novo področje, zato v sorodnih delih ne obstajajo točno določene smernice za razvoj. Kljub temu nam je uspelo

identificirati glavne poudarke, na katere smo bili pri izdelavi lastne rešitve pozorni. Mednje spadajo primeren izbor elementov poigritve glede na njihov učinek, vpliv poigritve na motivacijo in angažiranost uporabnikov ter identifikacija različnih tipov uporabnikov.

Da bi odgovorili na zastavljeno raziskovalno vprašanje smo si kot primarni cilj zadali izdelavo mobilne izobraževalne platforme za poučevanje in učenje glasbene teorije. Slednjega smo podrobneje razdelili v posamezne zahteve, in sicer po **odprtokodnosti rešitve** (dostopnost vsakomur, ki bi aplikacijo želel na kakršenkoli način uporabiti ali nadgraditi), **enostavni nadgradljivosti** (razširjanje izobraževalne platforme z novimi tipi iger in dodajanje novih elementov poigritve mora biti poenostavljeno), **poigritvi platforme** (vključitev elementov poigritve, kot so značke, točke in lestvica, nivoji, večigralski način itd. za povečanje motivacije uporabnikov za udeleževanje na platformi) in **posebitvi uporabniške izkušnje** (vključitev možnosti za prilagajanje izgleda in drugih nastavitev znotraj platforme glede na osebne preference uporabnika ter prilagodljiva težavnost nalog glede na stopnjo izobrazbe). Naša ideja in želja je bila, da bi dodajanje novih tipov nalog in s tem povečevanje njenega dometa postalo stalnica in da bi se sama aplikacija razvila v pomemben izobraževalni pripomoček in katalizator zanimanja za področje glasbe v Sloveniji.

Ker je bil poseben poudarek namenjen cilju poigritve učnega procesa, smo se odločili za izdelavo iger v obliki kvizov. Pri pregledu literature smo opazili, da so najbolj pogosto uporabljeni poigritveni elementi, ki se uporabljajo v podobnih aplikacijah za učenje, točkovanje ter lestvice, značke in nivoji, zato smo si tudi sami kot cilj zadali implementacijo le-teh. Ker naš namen ni bil implementirati čim večjega števila različnih iger, temveč vzpostavitev celotnega ogrodja in preverjanje smiselnosti zastavljenega koncepta, smo se v vsebinskem smislu omejili na naloge, kjer morajo učenci prepoznati in zapisati intervale glede na zaigrano zaporedje tonov pri solfeggiu. Ker smo proces generiranja intervalov želeli popolnoma avtomatizirati, a obenem ohraniti smiselnost generiranih nalog, smo se odločili za kombinacijo eksper-

tnega znanja z naključnostjo. Pri generiranju nalog smo skušali zagotoviti njihovo skladnost s posodobljenim učnim načrtom za solfeggio Ministrstva za izobraževanje, znanost, kulturo in šport [3].

Da bi preverili, če smo zastavljene cilje dosegli, smo aplikacijo testirali na ciljnih skupinah dijakov Srednje glasbene šole Konservatorija za glasbo in balet v Ljubljani in pri tem zbirali statistike o njihovi uspešnosti. Rezultate testiranja smo nato ovrednotili in tako sklenili krog za iterativno izboljševanje same aplikacije, kot tudi procesa poigritve učenja.

1.3 Struktura dela

V prvem delu magistrske naloge bomo pregledali sorodna dela, pri čimer bo poudarek namenjen sorodnim rešitvam za učenje glasbene teorije ter sorodnih del in eksperimentov s področja poigritve. V drugem delu podajamo opis uporabljenih tehnologij in orodij, s pomočjo katerih smo implementirali mobilno aplikacijo. V tretjem delu sledi tehnični in vizualni opis mobilne aplikacije. Četrty del se osredotoča na rezultate in njihovo analizo, v petem in zadnjem delu pa podajamo sklepne ugotovitve.

Poglavje 2

Pregled sorodnih del

2.1 Poigritev

2.1.1 Definicija poigritve in razlika med poigritvijo in učenjem, ki temelji na igri

Poigritev (angl. *gamification*) je definirana kot uporaba elementov igre v neigralskem okolju. Gre za širok pojem, ki običajno vključuje mehaniko dodeljevanja značk, nagrad in dosežkov, zbiranje točk, ki lahko predstavljajo nadomestilo tradicionalnim ocenam, ter možnost izbire lastne poti pri procesu učenja [4, 5, 6]. S poigritvijo želimo doseči večjo angažiranost uporabnikov za uporabo platforme preko spodbujanja določenih vzorcev v njihovem vedenju, s končnim ciljem potencialno pozitivnega vpliva na njihovo znanje (v primeru učencev na njihov učni uspeh) [7, 8, 4, 9].

Pomembno je, da ločimo med pojmom poigritev in učenjem, ki temelji na igri (angl. *game-based learning*). Poigritev se ne ukvarja direktno z igro, temveč se aplicira na ravni platforme oz. aplikacije, medtem, ko se učenje, ki temelji na igri uporablja za oplemenitenje izkušnje učenja preko transformacije učnega procesa v obliko igre [10, 11, 12]. V Tabeli 2.1 bolj natančno navajamo razlike med poigritvijo in učenjem, ki temelji na igri [12, 13].

Tabela 2.1: Razlike med poigrivitvijo in učenjem, ki temelji na igri.

Primerjava	Poigritev	Učenje, ki temelji na igri
cilj	privzemanje motivacije skozi igro	doseg ciljev znotraj same igre
igralec	identifikacija preko profila (ime, slika)	uporabnik je predstavljen z osebo znotraj igre
tehnike	nivoji, značke, točke, lestvice, profilne slike (angl. <i>avatars</i>), digitalni denar, tekmovanje z drugimi	zgodba, čustva, cilji znotraj igre, močno odvisno od same igre
pozitivni učinki	boljša uporabniška oz. učna izkušnja, boljše izobraževalno okolje, takojšnji odziv, spodbujanje sprememb v vedenju uporabnika	povečanje sposobnosti pomnjenja, pomaga pri hitrem strateškem razmišljanju, razvija koordinacijo med rokami in očmi, pomaga pri izgradnji veščin, kot je npr. orientacija
nagrade	doseganje točk, osvajanje značk, doseganje nivojev	notranji občutek izpoljenosti, premaganje ovir in s tem pridobitev novih znanj
čas implementacije	relativno hitro	dolgotrajen razmislek in razvoj, kako določen proces učenja spremeniti v igro
vsebina	dodajanje elementov v izobraževalno platformo ali druge podobne sisteme	vsebina je vključena v zgodbo igre

2.1.2 Elementi poigritve in njihovi učinki

Literatura navaja, da je proces učenja in osvajanja glasbene teorije zelo zahteven [14]. Zaradi tega obstaja možnost upada učne motivacije in v skrajnem primeru opustitve zanimanja za stroko [15]. Ta problem se vse bolj pogosto rešuje s poigritvijo učnega procesa in učenjem, ki temelji na igri [7].

Poznamo različne elemente poigritve, in sicer točke, lestvice, dosežki oz. značke, nivoji, zgodbe oz. teme, jasni cilji, povratne informacije, nagrade, napredek, izzivi, personalizacija in časovniki. Prevladujoči elementi v večini študij so točke, lestvice, značke in nivoji. Glavni razlog, ki botruje temu dejstvu je, da je implementacija in vključitev slednjih elementov v obstoječo platformo relativno hitra in ne preveč zahtevna [16]. V Tabeli 2.2 je natančen pregled in opis konceptov poigritve in njim pripadajočih elementov ter njihovega potenciala pri oblikovanju poigritvenega sistema na platformi [9].

2.1.3 Motivacija in angažiranost uporabnikov

Poigritev ima lahko velik čustven in socialen vpliv na uporabnike, saj sistem nagrajevanja in socialni mehanizmi tekmovanja zanje predstavljajo dodatno motivacijo. Sistemi nagrajevanja v obliki značk in doseganja nivojev naj bi bili inovativen, zabaven in spodbujajoč način k predstavitvi napredka znotraj izobraževalne poigrene platforme. Lestvice prav tako vplivajo na motivacijo uporabnikov, saj je njihovo delo javno ovrednoteno in ker na ta način dobijo perspektivo o tem, kje se, glede na uspeh, napram ostalim uporabnikom, trenutno nahajajo [11, 9].

Po drugi strani pa lahko trdimo, da omenjeni pozitivni učinki na motivacijo uporabnikov preko poigritve ne vplivajo na vse na enak način. Določenim uporabnikom sistem poigritve ne predstavlja dovolj velikega izziva, da bi jih motiviral k sodelovanju na platformi. Prav tako pa obstajajo tudi uporabniki, na katere poigritev vpliva celo negativno, saj ne želijo tekmovati z ostalimi uporabniki na platformi in ne želijo, da bi bili njihovi dosežki, kot so

Tabela 2.2: Pregled elementov poigritve in njihovega potenciala pri oblikovanju poigritvenega sistema.

Koncept	Elementi	Uporabnost v sistemu
dosežek (napredovanje)	točke, značke, nivoji, lestvice, napredek	Uporabniki občutijo zadovoljstvo ob pridobljenih dosežkih, kar jih dodatno motivira. Lestvice, točke in značke povečujejo socialni status.
nagrade	bonusi, povečana moč (angl. <i>power-up</i>), zbirke	Tesno povezane z dosežki. Nagrade se podeljujejo ob zaključku določene zbirke akcij ali pa ob nekih časovnih intervalih. Nudijo zunanjo motivacijo in so odraz vloženega časa, napora in pridobljenih veščin oz. znanj.
zgodba	potovanje po izzivih v obliki zgodbe	Uporabniki so postavljeni v pustolovski scenarij z namenom povečanja motivacije. Izkušnja učenja je predstavljena v pripovedni podobi. Uporabnik preko svojih odločitev vpliva na razplet zgodbe.
čas	odštevalnik (angl. <i>timer</i>), urnik	Odštevalniki časa ustvarijo občutek nujnosti. Urnik, ki uporabniku pove kdaj mora določeno nalogo opraviti in v kakšnem vrstnem redu, mu pomaga k boljšemu fokusiranju.
personalizacija	profilna slika, poimenovanje	Možnost izbire in modifikacije profilne slike. Zamenjava tem znotraj platforme za boljše naslavljanje potrebe po personalizaciji in povečanje zadovoljstva pri uporabi. Uporabniki lahko uporabljajo tudi vzdevke.

npr. značke in točke razvidne na lestvici, vidne vsem ostalim uporabnikom [11, 17].

Na podlagi teh spoznanj lahko identificiramo različne tipe uporabnikov. Dominguez in ostali [11] navajajo štiri različne tipe igralcev na osnovi njihovega udejstvovanja na platformi v smislu uspešnosti, zanimanj, nivoja socializacije in doseganja elementov poigritve kot so npr značke. Slednji so: uničevalci oz. morilci, (manj kot 1%), uporabniki osredotočeni na dosežke (okoli 10%), uporabniki, ki se primarno ukvarjajo z druženjem oz. socializacijo z drugimi uporabniki (približno 80%) in raziskovalci, ki se bolj kot na tekmovanje in socializacijo osredotočajo na odkrivanje (okoli 10%).

Opisane kategorije uporabnikov niso medsebojno izključujoče, kar pomeni, da lahko uporabniki hkrati pripadajo več tipom. Vendar raziskave kažejo, da ima velika večina uporabnikov en prevladujoči tip, ki definira uspešnost njegovega udejstvovanja na platformi. Razčlenitev uporabnikov v različne kategorije tako lahko bistveno pripomore pri gradnji sistema za poigritve, saj lahko uporabnikom različnih kategorij na bolj personaliziran način ponudimo uporabniško izkušnjo. Tako lahko uničevalce z različnimi indici usmerjamo v večigralski način, kjer bodo lahko dobili pozitivne občutke ob zmagi nad drugimi uporabniki, uporabnikom, ki se osredotočajo na dosežke, bolj pogosto prikazujemo njihovo uspešnost v obliki doseganja nivojev in točk in njihove dosežke v obliki značk, uporabnikom, ki so primarno prisotni na platformi zaradi socializacije, lahko ponudimo elemente kot so možnost pogovora z drugimi uporabniki in možnost komentiranja, raziskovalcem pa damo na voljo prvine, kjer lahko neovirano in v netekmovalnem duhu odkrivajo igro in elemente, povezane z njo [11, 9].

Omejitev sistemov za e-učenje s pedagoškega stališča je, da ne oddaja čustvenega naboja in zato ne more angažirati uporabnikov (v tem primeru učencev), kot bi to lahko storil učitelj, če bi učenje potekalo v živo. Za zapolnitev te vrzeli oz. reševanje tega problema morajo omenjeni sistemi uporabiti drugačne načine motivacije uporabnikov [10].

Z vpeljavo poigritve v učno platformo želimo spodbuditi bolj učinkovito in

angažirano učenje uporabnikov. V Munteanini študiji [10] je opisan Fogg-ov model vedenja, ki je osnovan na predpostavki, da se uporabniki, v primeru igranja iger, do računalnikov obnašajo na enak način, kot bi se do ljudi. Fogg-ov model vedenja tako trdi, da za spremembo ali sprožitev določenega vedenjskega vzorca pri uporabniku obstajajo 3 pogoji: motivacija, sposobnost za rešitev dane naloge in spodbuda (namig oz. sporočilo uporabniku, da mora v določenem trenutku opraviti neko akcijo oz. dejanje; poimenujemo ga lahko tudi kot signal oz. opomin). Vsi ti elementi morajo biti prisotni, da je angažiranje uporabnika uspešno.

Z elementi poigritve ne želimo nadomestiti uporabnikove notranje motivacije, ki je močnejša in bolj dolgoročne narave, z zunanjo motivacijo, temveč preko spodbujanja zunanje spodbujati notranjo. Tako bi uporabnik preko kombinacije notranje in zunanje motivacije potencialno lahko dosegel boljši uspeh [10, 11].

Kljub temu, da poigritev lahko poveča motivacijo in vključenost uporabnika, obstaja na njen račun tudi kritika, da se preveč osredotoča na zunanjo spodbude v obliki nagrad, medtem ko bi dejanska želja po angažiranju morala izvirati iz uporabnikove notranje motivacije. Hakulinen in ostali [4] tako trdijo, da bi poigritev lahko zmanjšala notranjo motivacijo z zamenjavo le-te z zunanjo. Po drugi strani pa v Munteanini študiji [10] menijo, da lahko poigritev poveča notranjo motivacijo, če je slednja implementirana na pravi način oz. imajo elementi poigritve velik pomen za uporabnike.

Vključenost oz. angažiranost uporabnikov je ključna metrika za ugotavljanje uspeha poigritve. Tu imamo v mislih predvsem vključenost uporabnikov v učne aktivnosti, saj velja, da je vključenost korelirana njihovimi učnimi uspehi. Mednje sodijo zadovoljstvo, vztrajnost in akademski dosežki [10, 8].

Običajno vključenost opisujemo s tremi različnimi komponentami: vedenjsko vključenostjo, afektivno vključenostjo (nanaša se na afekte uporabnika) in spoznavno (kognitivno) vključenostjo.

Vedenjsko vključenost povezujemo s pozitivnim vedenjem, odsotnostjo motečega vedenja in sodelovanjem v učnih aktivnostih. Kaže se v obliki

napora, vztrajnosti in pozornosti. Afektivna vključenost se osredotoča na pripravljenost za opravljanje dela in vključuje zanimanje za učne aktivnosti, užitek in pozitiven odnos do učenja. Spoznavna vključenost pa se navezuje na uporabnikovo željo in pripravljenost za dosego globokega poznavanja področja oz. prodobitve ekspertnega znanja. To implicira, da so uporabniki pripravljeni preseči normalne zahteve, ki jim jih postavljajo naloge v učnem procesu [8].

2.1.4 Testiranje uspešnosti poigritve in njeni vplivi

V člankih s področja ovrednotenja učinkov poigritve na proces učenja avtorji navajajo, da igre v splošnem ne veljajo več za nekaj obskurnega oz. nekaj, kar predstavlja igranje in ne delo, temveč kot pomembna spodbuda k učenju [10, 11, 18, 19, 20]. Zato ne preseneča dejstvo, da je področje e-učenja oz. področje izobraževanja dandanes primarna tarča za implementacijo poigritvenega sistema [16, 11, 21, 22]. Ostala področja uporabe poigritve s ciljem povečanja vključenosti oz. angažiranja uporabnikov so poslovne aplikacije, poslovni sistemi znotraj organizacij, aplikacije na področju zdravstva in zdravega življenja, itd. [16].

Učinke poigritve raziskovalci običajno testirajo preko osebnih intervjujev, vprašalnikov odprtega in zaprtega tipa, z zbiranjem podatkov o vedenju uporabnikov tekom uporabe aplikacije oz. platforme, različnimi eksperimenti, kot so A/B testiranja, fokusne skupine, itd. Večina opravljenih študij je kvantitativne narave, zgolj manjšina pa se poslužuje tudi kvalitativne analize. To implicira, da je večina raziskav na področju vplivov poigritve usmerjena na zbiranje statistik in nadaljnje sklepanje o vedenju uporabnikov [16, 11].

Z metodološkega vidika raziskave velikokrat niso idealno izvedene. Tipični problemi so [16]:

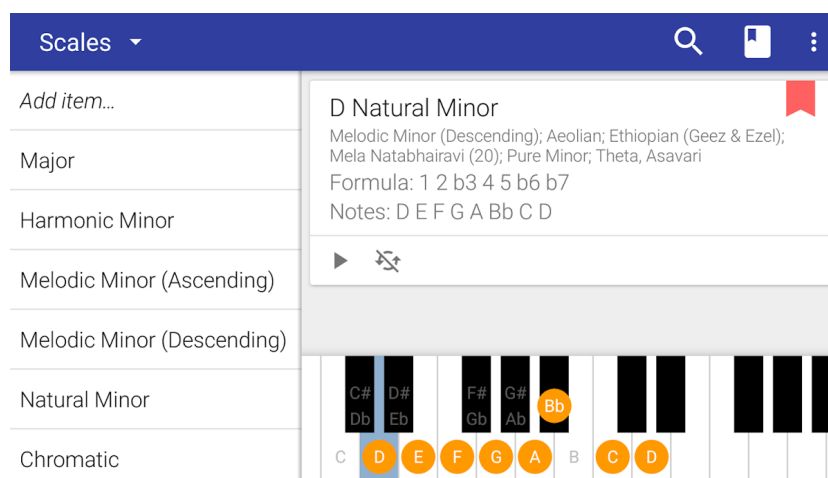
- testiranje na premajhnem vzorcih uporabnikov,
- izvedba eksperimenta brez definiranja kontrolne skupine (rezultati temeljijo zgolj na ovrednotenju podatkov zbranih o uporabnikih),

- velikokrat so rezultati predstavljeni samo deskriptivno in ne v kvantitativni obliki,
- časovna obdobja, v katerih so eksperimenti izvedeni so kratka (kratkoročni rezultati so lahko zavajajoči).

Študije ugotavljajo, da se v določenem številu primerov izkaže, da poigritev ni dolgoročen proces, temveč gre pozitivne učinke v veliki meri pripisati dejstvu, da je bila vpeljava poigritve nekaj novega, svežega in da so uporabniki to sprva z navdušenjem sprejeli, čez čas pa je nivo njihovega zanimanja upadel. Pomembno je omeniti tudi dejstvo, da v kolikor je bila poigritev platforme že izpeljana in se je izkazala za ne tako uspešno, kot je bilo pričakovati, ima naknadni odvzem elementov poigritve lahko negativen vpliv na dejavnost tistih uporabnikov, ki so poigritev pozitivno sprejeli. Kot razlog se navaja odpor do izgube že pridobljenih elementov poigritve, kot so npr. značke, nivoji, točke ipd. [16].

Nekatere študije izvedene na platformah za e-učenje prav tako kažejo, da rezultati poigritve niso vedno pozitivni. Učenci, ki so na platformi, v okviru poigritve, dosegali dobre rezultate, so se slabše odrezali pri preverjanjih na konvencionalen način v razredu, četudi je bila njihova začetna motivacija visoka [11, 23]. Pri platformah za e-učenje s področja glasbe obstaja možnost, da so nianse pri rezultatih odvisne tudi od izkušenosti oz. neizkušenosti uporabnikov pri uporabi modela klaviature in pri povezavi tega modela z njihovim dožemanjem tonskih višin.

V splošnem pa večina študij navaja, da so bili njihovi testi poigritve samo deloma uspešni, testi s popolnoma uspešnimi rezultati so v veliki manjšini. Iz opisanega gre torej sklepati, da je poigritev težak proces, močno odvisen od domene, v kateri je implementiran, za maksimizacijo uspešnosti poigritve pa moramo biti pozorni na raznolike dejavnike, kot so različni tipi uporabnikov, njihova notranja in zunanja motivacija ter vključenost oz. angažiranost, hkrati pa se moramo za izboljšavo sistema poigritve poslužiti metodološko pravilno izpeljane analize [16, 23].



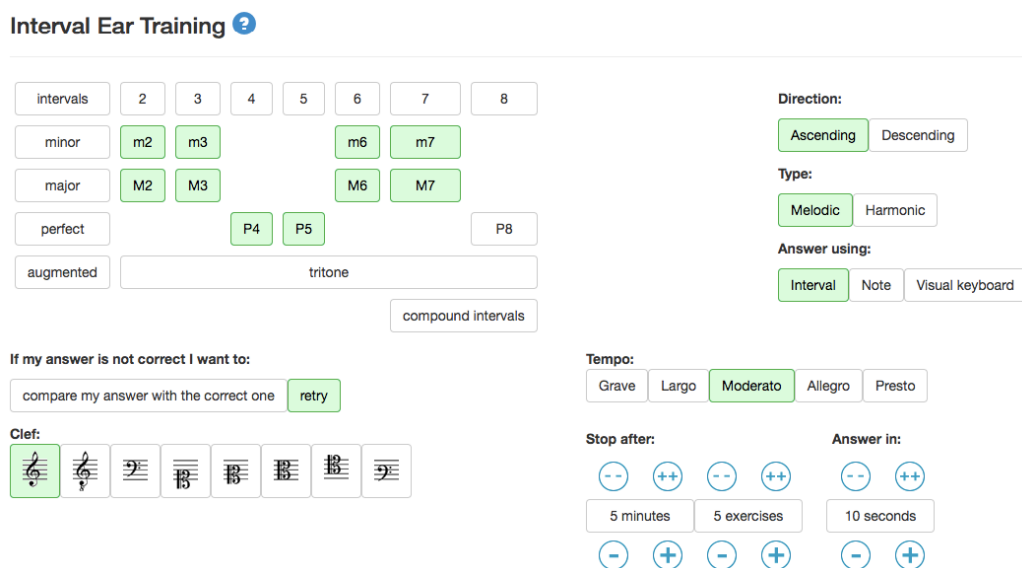
Slika 2.1: Aplikacija *My Piano Assistant*.

2.2 Sorodne rešitve za učenje glasbene teorije

Obstoječi sistemi za vajo glasbene teorije v Sloveniji skušajo na neprivlačen način poigrati učenje. Izdelani so v zastarelih tehnologijah s slabo grafično podobo. Prav tako ne ponujajo poglobljenih statistik uspešnosti reševanja nalog, se ne prilagajajo posameznikovemu nivoju znanja in ga ne vodijo skozi proces učenja [1, 2].

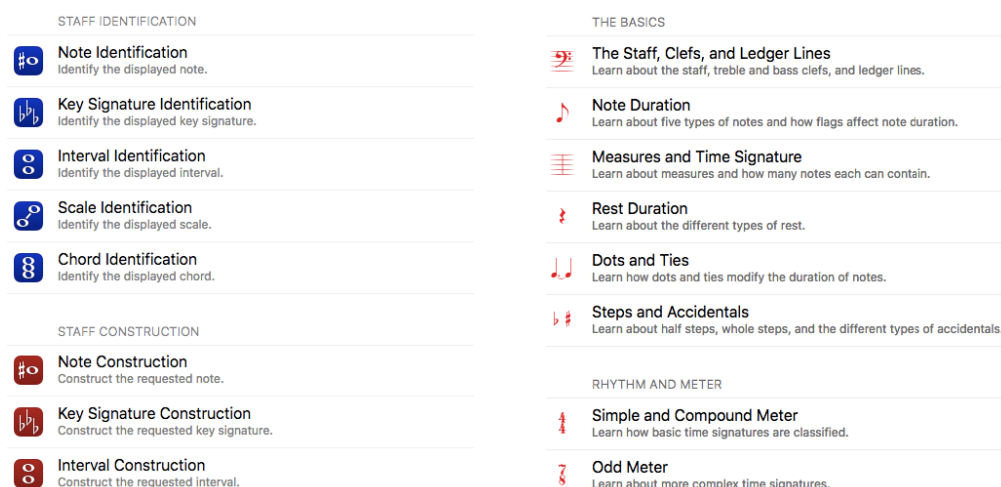
V svetovnem merilu je področje boljše pokrito, saj obstajajo zanimive spletne, namizne in mobilne aplikacije, ki se na konstruktiven način lotevajo interaktivnega poučevanja in vaje glasbene teorije.

1. **My Piano Assistant** (Slika 2.1): Kot že prevod imena sporoča, predstavlja pripomoček pri učenju igranja klavirja. Uporabniku prikazuje akorde, lestvice in intervale ter omogoča predvajanje izbranega zvoka. Na voljo je zgolj v obliki mobilne aplikacije [24].
2. **Teoria** (Slika 2.2): Implementira vadnice intervalov, not, lestvic, melodij, harmonij in ritmičnega nareka ter ponuja statistiko uspešnosti reševanja in s tem možnost analize. Preko modula za učitelje omogoča prilagajanje nalog. Prav tako učencem ponuja tudi lekcije, kjer so na interaktiven predstavljen teoretične teme. Na voljo je v obliki spletne

Slika 2.2: Aplikacija *Teoria*.

aplikacije [25].

3. **musictheory.net** (Slika 2.3): Uporabnikom ponuja vadbice not, intervalov, lestvic in akordov, ne omogoča pa kasnejše analize reševanja v obliki statistik. Sistem je osredotočen zgolj na učenca samega in kot tak ne ponuja modula za učitelje za delo s skupino učencev. Kot številne ostale aplikacije tudi ta ponuja teoretične lekcije s pordočja glasbe. Na voljo je v obliki spletne in mobilne aplikacije [26].
4. **Musition** (Slika 2.4): Aplikacija ponuja vaje branja not, določanja intervalov, akordov, ritma, lestvic in inštrumentov ter vaje iz področja skladanja. Izmed vseh omenjenih sistemov ima največ globine, saj so naloge, ki jih ponuja, prilagojene učnim načrtom glasbenih šol v Avstraliji, Novi Zelandiji, Veliki Britaniji in ZDA, učence pa vodi skozi učni proces z elementi poigrivke in ponuja paleta statistik za poglobljeno ovrednotenje rezultatov. Prav tako implementira modul za učitelje, kjer učiteljem omogoča kreiranje tečajev z izbranimi vadnicami, v katere lahko vključujejo različne razrede učencev. Poleg samih vadnic v



(a) Seznam vadnic.

(b) Teoretična poglavja.

Slika 2.3: Aplikacija *musictheory.net*.

obliki pisnih in slušnih nalog učencem ponuja tudi teoretične lekcije. Na voljo je v obliki spletne, namizne in mobilne aplikacije [27].

5. **Perfect Ear** (Slika 2.5): Implementira prilagodljive vadbice intervalov, lestvic, akordov in ritma. Ponuja tudi vaje melodičnega nareka ter omogoča treniranje absolutnega posluha. Obenem se lahko učenci znotraj aplikacije seznanjajo s teorijo glasbe v obliki člankov. Na voljo je v obliki mobilne aplikacije [28].
6. **iRealPro** (Slika 2.6): Aplikacija omogoča ustvarjanje, urejanje in zbiranje akordov za različne instrumente. Prav tako ponuja interaktivno vadbico za učenje instrumentov kot so kitara, ukulele in klavir. Je bolj praktične narave kot ostale aplikacije in kot taka ne ponuja teoretičnih vadnic, statistik in modula za učitelje. Na voljo je v obliki namizne in mobilne aplikacije [29].
7. **MyEarTraining** (Slika 2.7): Platforma je namenjena uporabi v šolah, privatnim učiteljem in glasbenikom kot posameznikom. Ponuja vadbice intervalov, akordov, lestvic, melodij, itd. Naloge imajo različne

The screenshot displays the Musition5 application interface. At the top, there is a navigation bar with icons for HOME, PRACTICE, PEOPLE, MESSAGES, SYLLABUS, TESTS, COURSES, REPORTS, LIBRARY, NOTATION, SETTINGS, and HELP. The main content area is divided into three panels:

- Syllabus List:** A list of syllabi including Complete Syllabus, AP* Music Theory, Band, Complete Syllabus V5, Florida State Music Teachers Association, Jazz, Solfege - Fixed Do, Solfege - Moveable Do, Solfege - Moveable Do La, Trinity Guildhall, UK - GCSE, and Western Australia.
- Topics:** A list of topics including Advanced Note Reading, Advanced Progressions, Artists & Performers, Beaming, Cadences, Chord Progressions, Chord/Scale Relationships, Chords, Clef Transposition, Composers, Compositional Devices, Concepts, Diatonic Chords, Drum Sticking, Drum Styles, Enharmonic Notes, Guitar Symbols, Instrument Keys, Instrument Knowledge, Instrument Range, Instrument Recognition, Instrument Transposition, Intervals, Jazz Chord Symbols, Jazz Chords, Jazz Forms, and Jazz Scale Degrees.
- Levels:** A list of levels from Level 1 to Level 7.

Below the Syllabus List, there is a **Syllabus Details** section for the selected 'Complete Syllabus'. It includes fields for Name, Description, and Options. The options include 'Rhythm Terminology' with sub-options for 'US style (half, quarter notes)' and 'UK style (minims, crotchets)'. The 'Level Details' section for 'Level 1' includes a Name field, a Description field, a Content slider set to 'Generated questions only', and a 'Test only' checkbox.

User: WILSON, TIM (TIM) (Cloud school RS2016)

(a) Učni programi.

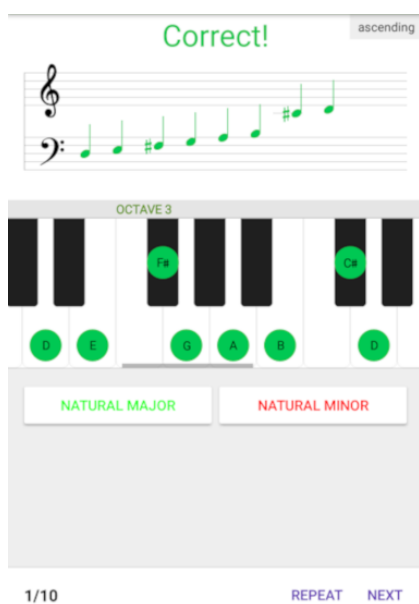
The screenshot displays the Musition5 application interface. At the top, there is a navigation bar with icons for HOME, PRACTICE, PEOPLE, MESSAGES, SYLLABUS, TESTS, COURSES, REPORTS, LIBRARY, NOTATION, SETTINGS, and HELP. The main content area is divided into three panels:

- Courses:** A list of courses including Music Theory Course (Level 1-5), Jazz Theory Course (Level 1-3), AP* Music Theory Course (Level 1-2, Exam Level), Band Theory Course (Novice - Level 1-3, Intermediate - Level 1-3, Advanced - Level 1-3), Trinity Guildhall Theory Course (Grade 1-5), and VET Theory Course.
- Course Details:** A detailed view of the 'Music Theory Course - Level 1'. It includes fields for Name, Description, Group, Options (Allow course to be resumed later, Allow students to view entries, Maximum attempts: 5), Results (Allow students to view results, Show results upon completion), Entries (Contains 74 entries), Timeslots (Available at any time), and Copyright.
- Classes:** A list of classes including 10A, 11A, 12A, 7A, 7B, 7C, 8A, 8B, 8C, 9A, 9B, and Default Class.

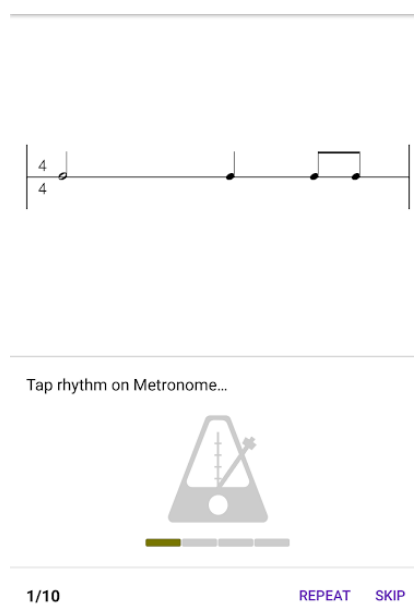
User: WILSON, TIM (TIM) (Cloud school RS2016)

(b) Modul za učitelje.

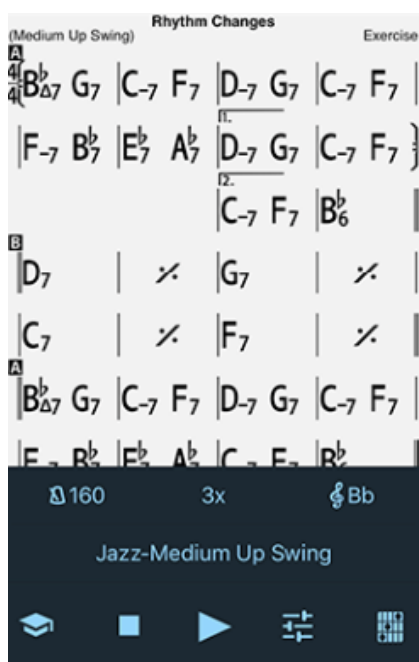
Slika 2.4: Aplikacija Musition.



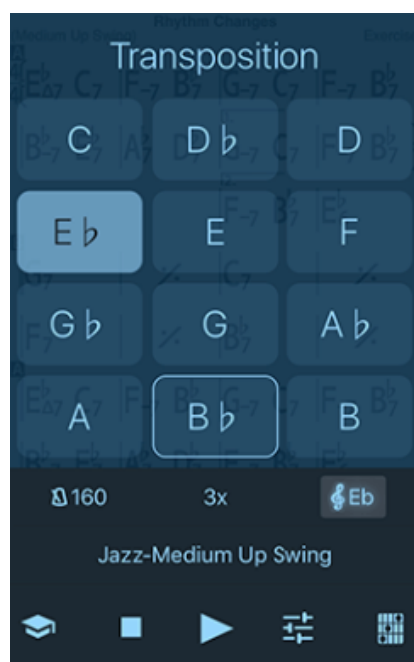
(a) Vaja lestvic.



(b) Vaja ritma.

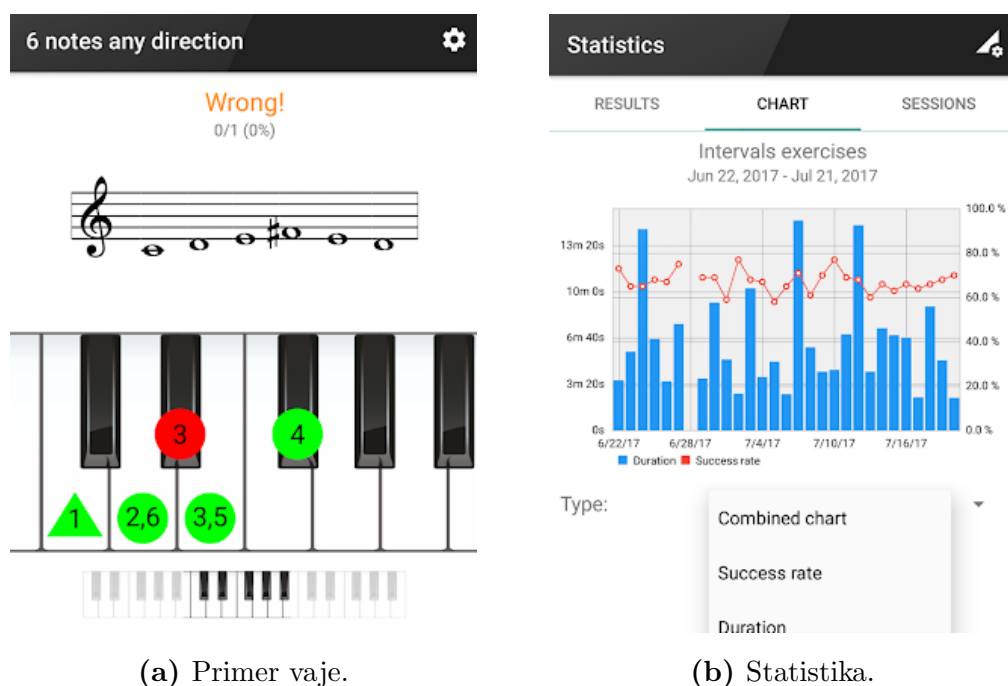
Slika 2.5: Aplikacija *Perfect Ear*.

(a) Urejanje akordov.



(b) Transpozicija.

Slika 2.6: Aplikacija *iRealPro*.



Slika 2.7: Aplikacija *MyEarTraining*.

nivoje težavnosti in so s strani uporabnika lahko podrobno parametrizirane. Uporabnikom je na voljo poglobljena statistika, kjer lahko prepoznajo svoje prednosti in slabosti. Implementira tudi osnoven modul za učitelje, kjer učitelji lahko pregledujejo uspehe svojih učencev. Na voljo je v obliki spletne, namizne in mobilne aplikacije [30].

8. **Sibelius** (Slika 2.8): Aplikacija se močno razlikuje od ostalih, saj ne ponuja vadnic, temveč predstavlja pripomoček pri komponiranju in aranžiranju skladb. Obenem ponuja kreiranje delovnih listov za uporabo v izobraževanju s področja glasbene teorije. Na voljo je v obliki namizne aplikacije [31].

The image displays a screenshot of the Sibelius music notation software interface, showing five staves of music notation for different instruments: Flugelhorn, Clarinet, Bagpipe, Vibes, and another instrument. The notation is annotated with various musical features and their corresponding software capabilities:

- Flugelhorn:**
 - create instruments in traditional or custom order
 - custom stem lengths, beam angles and stem symbols
 - note names stemless notes within notes that playback
 - unusual beaming including stemlets
 - hidden cautionary key signatures
- Clarinet:**
 - transposing instruments have correct key signature
 - bar numbers on specific staves
 - articultations inside & outside the staff
 - custom beaming
 - trills that playback
 - many kinds of ornament that playback
 - highlighted sections (to view this switch on View > Invisibles > Highlights)
 - winged repeats
 - tremolos between 2 notes
 - dense chords & clusters
 - many ledger lines
- Bagpipe:**
 - lots of different notehead designs, plus custom noteheads
 - feathered beaming
 - ossia staves
 - dynamics within hairpins
 - complex dynamic markings
- Vibes:**
 - any number of grace notes, of any value
 - complex time signatures
 - bracketed cautionary accidentals
 - colored symbols
 - longs
 - slash notation

At the bottom right, there is a legend for colored symbols, including a red treble clef, a blue bass clef, a blue circle with a cross, a blue circle with a dot, a red circle with a cross, a green circle with a cross, a blue circle with a cross, and a blue circle with a cross.

Slika 2.8: Aplikacija *Sibelius*.

Poglavje 3

Uporabljene tehnologije in orodja

3.1 Laravel

Laravel [32] je najbolj priljubljeno odprtokodno PHP spletno ogrodje namenjeno razvoju spletnih aplikacij. Sloni na razvijalskem vzorcu MVC, med njegove glavne prednosti pred ostalimi PHP spletnimi ogrodji pa spadajo ekspresivna sintaksa kode, ki sledi principom objektnega programiranja, modularni paketni sistem z upravljalnikom paketov Composer, različne možnosti dostopa do relacijskih podatkovnih baz in orodje za objektno-relacijsko preslikavo Eloquent, orodja za pomoč pri nameščanju in vzdrževanju rešitve in številne možnosti za generiranje ponavljajočih se delov programske kode preko sistema artisan. Obenem ogrodje vključuje delujočo logiko za avtentikacijo in avtorizacijo, usmerjanje, upravljanje s sejami, logiko za migriranje in sejanje podatkovnih baz, logiko za testiranje ter orodje Blade za upravljanje s predlogami.

3.1.1 SVG

SVG [33] je vektorski format slike zapisan v formatu XML. Podpira dvodimenzionalno grafiko, interaktivno vsebino in animacije. SVG je specificiran

z odprtim standardom, ki ga razvija W3C.

Slike SVG in njim pripadajoča logika so definirane v tekstovnih datotekah XML, kar omogoča poizvedovanje po njihovi vsebini, indeksiranje in kompresijo. Tako kot vsaka datoteka XML, lahko tudi sliko SVG ustvarimo in spreminjamo z običajnimi urejevalniki besedila, hkrati pa jo lahko prikažemo kot sliko v enem od programov za prikazovanje slik.

SVG lahko vključimo v vsebino kode HTML kot `svg` element. Slednji predstavlja vsebnik za SVG grafiko. Z različnimi specifičnimi elementi lahko znotraj vsebinka izrisujemo poti, kroge, elipse, kvadrate, pravokotnike, besedilo, itd. Vse te elemente lahko tudi poljubno animiramo, kot to velja za veliko večino osnovnih elementov HTML.

Vsi moderni brskalniki, vključujoč Google Chrome, Safari, Mozilla Firefox, Internet Explorer, Microsoft Edge in Opera, imajo podporo za prikazovanje in izris elementov SVG. V zadnjem času SVG postaja vedno bolj priljubljena izbira za prikazovanje slik v brskalnikih, saj se slike SVG lahko neomejeno skalirajo brez izgube.

3.2 Sass

Programski jezik stilskih predlog Sass [34] predstavlja izboljšavo jezika stilskih predlog CSS. Gre za dinamični jezik stilskih predlog, čigar koda se interpretira ali pretvori (angl. *compile*) v kodo jezika CSS. Pretvorba datoteke iz jezika Sass v CSS ustvari veljavno CSS kodo, ki je kompatibilna z vsemi modernimi brskalniki. Sass vsebuje naslednje mehanizme, ki ga bogatijo v primerjavi s CSS: spremenljivke, gnezdenje razredov, integracijo pomožnih razredov iz drugih datotek (angl. *mixins*), operatorje in funkcije.

Obstajata dve različici sintakse jezika Sass. Pri prvi različici za ločitev delov kode (npr. razredov) uporabljamo zamike, za ločitev pravil pa znake za novo vrstico. Datoteke napisane v sintaksi slednje različice uporabljajo končnico `.sass`. Pri drugi različici za ločitev delov kode uporabljamo zavite oklepaje in zaklepaje, za ločitev pravil oz. lastnosti znotraj bloka pa podpičja.

Njena sintaksa je enaka kot pri jeziku CSS, imenuje pa se SCSS (končnice datotek napisane v slednji sintaksi so `.scss`).

Za lažje razvijanje obstajajo Sass pretvorniki, ki v realnem času pretvarjajo kodo zapisano v Sass oz. SCSS sintaksi v kodo jezika CSS.

3.2.1 Notacija BEM

Notacijo BEM [35] se uporablja za poimenovanje razredov stilskih predlog in prinaša boljše semantiko v programsko kodo. Gre za način poimenovanja razredov, ki temelji na konsistentnosti in posledično prinaša večjo urejenost stilskih predlog in manjše težave pri pretvarjanju datotek dinamičnih stilskih predlog, kot je npr. Sass, v CSS. Notacija je uporabna predvsem za velike, kompleksne projekte, kjer je organizacija kode ključna za učinkovitost (čas pisanja kode, količina pisanja kode in čas, ki ga bo brskalnik uporabil za interpretacijo kode, kar pomeni čas nalaganja spletne strani).

Prednosti notacije BEM so strukturiran jezik stilskih predlog in strukturiran uporabniški vmesnik. Metoda, za razliko od ostalih notacij CSS prinaša večjo jasnot in omočoga zelo dobro arhitekturo razredov stilskih predlog preko terminologije, ki je hitro razpoznavna. Blok (angl. *block*) predstavlja samostojno entiteto, ki sama po sebi nosi pomen (npr. `header`, `checkbox`, `input`, itd.), element je del bloka, ki samostojno ne more obstajati in je semantično povezan z blokom (npr. `menu item`, `list item`, `header title`, itd.), medtem ko določilo (angl. *modifier*) predstavlja lastnost bloka oz. elementa in se uporablja predvsem za spremembo izgleda ali logike delovanja elementa (npr. `disabled`, `highlighted`, `checked`, itd.).

Primer uporabe jezika stilskih predlog Sass (sintaksa SCSS) in notacije BEM je razviden v Dodatku A.1.

3.3 Vue.js

Ogrodje Vue.js [36] je namenjeno razvoju klienta oz. uporabniškega vmesnika v okviru gradnje spletnih aplikacij. Gre za popularno ogrodje napi-

sano v jeziku JavaScript, ki je bilo ustvarjeno z namenom organizacije in poenostavitve razvoja spletnih aplikacij. Vue.js skuša narediti prvine razvoja spletnih uporabniških vmesnikov bolj dostopne (mednje spadajo komponente, razhroščevanje, deklarativen uporabniški vmesnik, osveževanje v realnem času, itd.). Omogoča inkrementalno prilagodljivo arhitekturo. Njene jedrne knjižnice se osredotočajo na izris elementov v brskalniku ter grupiranje elementov v skupke, ki jih imenujemo komponente. Le-te so modularne narave in se jih da na preprost način vključiti v katerokoli spletno aplikacijo na poljubno mesto. To pomeni, da je ogrodje primerno tudi za delno nadgradnjo že obstoječih spletnih aplikacij. Napredne prvine, ki jih potrebujemo za razvoj kompleksnih aplikacij, kot so npr. usmerjanje (angl. *routing*) in upravljanje stanja (angl. *state management*), so dostopne preko podpornih knjižnic in paketov, ki jih Vue.js uradno podpira.

3.3.1 Predloge

Ogrodje Vue.js uporablja sistem za predloge osnovan na jeziku HTML. Sistem omogoča deklarativno vezavo v brskalniku izrisanega modela DOM (hierarhično urejeno drevo objektov, ki so preslikava hierarhične strukture HTML dokumenta) na lastnosti ustvarjene instance ogrodja Vue. Ker ogrodje omogoča osveževanje pogleda v realnem času (reaktivnost), se z namenom hitrejšega delovanja ob vsaki spremembi stanja uporabniškega vmesnika izračuna minimalno število sprememb, ki so potrebne za manipulacijo modela DOM in ponoven izris strani v brskalniku. Vue prevede predloge v funkcije za izris modela DOM, rezultat pa predstavlja veljavno HTML kodo, ki jo znajo razčleniti vsi brskalniki.

Sintaksa predlog je obogatena s različnimi možnostmi za interpolacijo besedila, atributov in logičnih izrazov napisanih v jeziku JavaScript in z direktivami, ki predstavljajo posebno vrsto atributov s predpono `v-`. Vrednosti atributov direktiv so izrazi napisani v jeziku JavaScript. Naloga direktiv je, da v realnem času aplicirajo spremembe na model DOM, če se vrednost izraza v atributu spremeni. Primer direktive je npr. `v-if="expression"`, ki

nadzoruje ali bo izbrani element prikazan v modelu DOM in s tem izrisan v brskalniku.

3.3.2 Reaktivnost

Ogrodje Vue.js uporablja nevpadljiv sistem za dinamično osveževanje vsebine na spletnih straneh, ki temelji na konceptu reaktivnosti (angl. *reactivity*). Ko se določen podatek oz. lastnost komponente, ki je vezan oz. reaktiven spremeni, se avtomatsko posodobi tudi pogled spletne strani. To pripomore k temu, da je upravljanje stanja lahko in intuitivno. Ponoven izris modela DOM v brskalniku je optimiziran zahvaljujoč dejstvu, da sistem za osveževanje hrani in spremlja vrednosti vezanih lastnosti, tako da se venomer posodobi zgolj minimalna množica elementov.

3.3.3 Komponente

Komponente so ena od najmočnejših prvin ogrodja Vue.js. Pri velikih spletnih aplikacijah je za večjo preglednost in s tem lažji razvoj potrebno celotno aplikacijo razdeliti na manjše, modularne, samozadostne, hierarhično urejene dele, ki so hkrati primerni za večkratno uporabo na različnih delih aplikacije. Ta problem naslavlja komponente, ki jih lahko na zelo enostaven način vključimo v obstoječo strukturo kode HTML. Posamezna komponenta lahko poleg lastne kode HTML vsebuje tudi lastne stilske predloge CSS in lastno logiko v obliki funkcij napisanih v programskem jeziku JavaScript. Vse funkcije in predloga HTML si med seboj delijo kontekst komponente, na katerega so vezane številne lastnosti komponente, kot npr. `data` (interno definirane reaktivne spremenljivke komponente) in `e1` (korenski HTML element komponente).

Vsaka komponenta gre skozi različne življenjske faze. Na določenih stopnjah se kličejo ti.i funkcije življenjskega cikla (angl. *lifecycle hooks*), ki jih lahko razvijalec uporabi za dodatni nadzor nad izvajanjem logike komponente. Tako se ob inicializaciji kliče funkcija `created`, po vstavitvi kompo-

nente v model DOM se sproži funkcija `mounted`, ob posodobitvi komponente v modelu DOM imamo na voljo funkcijo `updated`, ob uničenju komponente pa se sproži funkcija `destroyed`, ki omogoča dodatno sproščanje potencialno uporabljenih virov. To so zgolj najpomembnejše funkcije, obstajajo pa številne druge, ki dovoljujejo še večjo zrnatost pri nadzoru delovanja.

3.3.4 Usmerjanje

Moderne spletne aplikacije so na strani klienta implementirane v obliki SPA aplikacij, ki s strani spletnega strežnika sprejmejo zgolj eno HTML spletno stran (običajno `index.html`). Slednja referencira datoteko stilskih predlog CSS in datoteko, ki vsebuje logiko za izvajanje spletne strani v jeziku JavaScript. Spletna stran se po tej osnovni inicializaciji ne osvežuje več in zato lahko hrani lokalno stanje. Za premik na določeno podstran tako ne kličemo več spletnega strežnika, da bi nam le-ta serviral HTML vsebino, temveč uporabljamo interni usmerjevalnik.

Interni usmerjevalnik ogrodja Vue.js lahko implementiramo s pomočjo uradno podprte knjižnice `vue-router` [37]. Prehajanje med podstranmi je v grobem ekvivalentno menjavi komponent. Običajno se na osnovni strani definira HTML element `router-outlet`, ki je specifičen usmerjevalniku. Igra vlogo vsebnika komponent, saj mu za menjavo podstrani zgolj zamenjamo Vue komponento. Vsaka stran lahko vsebuje več omenjenih vsebnikov, vsakega izmed njih pa lahko posebej nadzorujemo in tako istočasno izmenjujemo več komponent hkrati. Vse te menjave se dogajajo brez osveževanja same spletne strani, zato lokalnega stanja nikoli zares ne izgubimo.

Poudariti velja tudi dejstvo, da se ob menjavi komponent zamenja tudi dejanski naslov URL, tako, da lahko uporabnik uporablja osnovne funkcije brskalnikov, kot sta npr. shranjevanje zaznamkov in navigacija naprej ter nazaj, brez skrbi, da bi se stran dejansko osvežila. To je bil eden od bistvenih problemov usmerjanja SPA aplikacij, ki pa je sedaj v celoti rešen.

3.3.5 Upravljanje stanja

Velike spletne aplikacije lahko z rastjo postanejo pretirano kompleksne. Ta problem se v SPA aplikacijah naslavlja z uporabo sistema za uporabljanje stanja, ki omogoča, da si komponente med seboj izmenjujejo informacije. Skupno skladišče stanja hkrati zmanjšuje število klicev na spletni strežnik, saj lahko informacijo, ki jo potrebujemo v več komponentah s spletnega strežnika pridobimo le enkrat, nadalje pa jo zgolj pridobimo iz skladišča.

Ogrodje Vue.js za upravljanje stanja uporablja knjižnico imenovano Vuex [38]. Njena pravila zagotavljajo, da se stanje spreminja oz. mutira na vnaprej predpisan način. Spremembe stanja so omogočene preko mutacij (sinhrone operacije) in akcij (asinhrono operacije). Mutacije spreminjajo zgolj lokalno shranjene informacije, medtem ko akcije potencialno komunicirajo s spletnim strežnikom oz. njegovim vmesnikom API in po pridobitvi informacij interno izvajajo mutacije. Sprememba stanja aplikacije običajno vpliva na pogled, ki se prikazuje v brskalniku. Komponente običajno vežejo svoje lokalne lastnosti na lastnosti stanja aplikacije, zato se sprememba pogleda zgodi avtomatsko (princip reaktivnosti).

3.4 Timer Progress

Timer Progress [39] je lastna knjižnica, ki temelji na elementih SVG, omogoča pa grafični prikaz odštevalnika časa. Ob inicializaciji instance razreda knjižnice lahko razvijalec nastavlja lastnosti povezane z izgledom odštevalnika, časom odštevanja in načinom odštevanja časa (omejen ali neomejen). Način, v katerem je čas omejen, je uporaben predvsem za spletne strani, ki zastavljajo neko časovno omejeno vprašanje. Prav tako lahko odštevalnik časa, v načinu, ko čas ni omejen, uporabimo za indikator, da se vsebina spletne strani še nalaga (angl. *loader*). Primer uporabe knjižnice je razviden v Dodatku A.4. Knjižnica je kompatibilna z vsemi modernimi brskalniki.

3.5 MIDI

MIDI [40] je elektronski standardni protokol, ki omogoča snemanje, shranjevanje in igranje digitalne glasbe. V formatu MIDI ni shranjena sama glasba (signal), temveč različne informacije o njej. Z MIDI formatom upravljamo z različnimi ukazi, najbolj pogosta pa sta `note on` in `note off`. Slednja sprejemata parametre, kot so višina tona (angl. *pitch*), moč oz. hitrost (angl. *velocity*), s katero je bila nota zaigrana, kanal (angl. *channel*), na katerem je bila nota zaigrana - določa vrsto inštrumenta, itd. V eni datoteki lahko shranimo do 16 kanalov, vsak izmed njih pa ima lahko izbran različni inštrument (barvo zvoka).

Prednost MIDI formata je v tem, da je z njim relativno preprosto kombinirati različne inštrumente, spreminjati lastnosti not, shranjevati različne posnetke, itd. Najpomembnejša prednost pa je v tem, da so datoteke, ki so zapisane v MIDI formatu zelo majhne. Zato je zelo primeren za simulacijo inštrumentov na računalniku.

3.5.1 MIDI.js

Odprtokodna knjižnica MIDI.js [41], implementirana v jeziku JavaScript, se uporablja za predvajanje zvokov na spletnih straneh.

Za delovanje potrebuje t.i. zvočne predloge (angl. *soundfonts*), ki vsebujejo podatke o vseh možnih tonih za pozamezne inštrumente. Zvočne predloge, ki jih aplikacija potrebuje, se prenesejo na stran klienta, določene pa so z izbranimi inštrumenti (npr. klavir, trobenta, klarinet, kitara, violina, saksofon, itd.). Vsakemu izmed 16 kanalov, ki so na voljo, lahko določimo program, ki predstavlja izbrani inštrument, ter glasnost zvoka. Logika inicializacije knjižnice je razvidna v Dodatku A.2.

Po opravljenih začetnih nastavitvah lahko s preprostimi metodami, kot sta `noteOn` in `noteOff` predvajamo zvoke tonov. Obe sprejemata parametre, ki predstavljajo kanal, višino tona in zakasnitev (angl. *delay*). Prva metoda prav tako sprejme parameter, ki predstavlja moč oz. hitrost pritiska.

Za predvajanje shranjenega zaporedja tonov knjižnica MIDI.js ne ponuja rešitve, zato se lahko v tem primeru poslužimo lastne rešitve. Pri tem se sprehodimo po seznamu tonov in kličemo ustrezni metodi `noteOn` in `noteOff` s časovno zakasnitvijo, ki je določena z relativnim časovnim odmikom tona od začetka predvajanja. Za zaustavitev predvajanja se prav tako poslužimo lastne metode. Logika obeh metod je razvidna v Dodatku A.3.

Poglavje 4

Mobilna aplikacija

Glavni cilj, ki smo si ga zastavili in s katerim smo želeli preverili vpliv poigrivitve na proces učenja glasbene teorije, je bila izdelava mobilne izobraževalne platforme in njeno ovrednotenje. Pri razvoju smo sledili trem glavnim smernicam, in sicer odprtosti rešitve, enostavni nadgradljivosti aplikacije in poigrivni platforme.

4.1 Tehnični opis

Z željo po doseganju čim širše populacije in zmanjšanju števila platform, katerim bi aplikacijo morali prilagajati oz. za katere bi jo morali razviti (npr. Windows, Linux, OS X, Android, iOS, ...), smo se odločili za implementacijo aplikacije v spletnem okolju (uporaba aplikacije je torej možna v spletnih in mobilnih brskalnikih). Z odločitvijo, da bomo slednjo razvili za spletno okolje, smo tako aplikacijo lahko razvili zgolj za eno okolje in jo po potrebi malenkostno prilagodili za različne brskalnike. S tem smo dosegli zelo veliko število potencialnih uporabnikov, saj imajo vsi namizni računalniki in vse mobilne naprave dandanes naložen spletni oz. mobilni brskalnik.

Aplikacija je bila na strežniški strani implementirana s pomočjo strežniškega programskega jezika PHP in njemu pripadajočega ogrodja Laravel. Strežnik komunicira s podatkovno bazo MySQL, pri čimer se za poenostavitev ra-

zvoja uporablja objektno relacijska preslikava med podatkovno bazo in modeli ogrodja Laravel. Klient je razdeljen na dva dela, in sicer na tistega, ki se ukvarja z registracijo in avtentikacijo in uporablja predloge Blade ogrodja Laravel, ter na tistega, v katerega vstopimo po avtentikaciji in vključuje vso ostalo logiko - implementiran s pomočjo ogrodja Vue.js in njemu pripadajočega sistema komponent, usmerjevalnika ter sistema za upravljanje stanja. Vsa komunikacija s spletnim strežnikom po opravljeni avtentikaciji in inicializaciji osnovne komponente ogrodja Vue.js poteka preko asinhronih spletnih zahtevkov na strežniški vmesnik API, ki vrača podatke v formatu JSON. Spletni strežnik je torej v svojem jedru tipa RESTful, omenjeni klici pa so naslovljeni na naslove URL določenih spletnih virov.

Pri implementaciji so bili uporabljeni najnovejši pristopi razvoja spletnih aplikacij preko uporabe ogrodij Laravel in Vue.js, podpornih knjižnic axios [42], Lodash [43], Moment.js [44], node-sass [45] ter notacije BEM. Je enostavno razširljiva, saj npr. dodajanje novih iger v aplikacijo terja zgolj dodajanje komponente Vue na klientu ter dopolnitev logike določenih kontrolerjev na strežniku.

Aplikacija je prav tako fleksibilna, saj se za vzpostavitev okolja uporabljata orodji za upravljanje s paketi, NPM in Composer, in njima pripadajoči nastavitveni datoteki `package.json` in `composer.json`, kar omogoča enostavno prenosljivost projekta v različna okolja in enostavno nameščanje rešitve na produkcijski strežnik. Aplikacija je odprtokodne narave in vsem dostopna na portalu GitHub (uporablja sistem za upravljanje z verzijami Git) na spletnem naslovu <https://github.com/volkmaster/trubadur>. Uporaba tehnologije Git omogoča, da na projektu dela več razvijalcev hkrati in dodaja projektu popolno transparentnost, saj sistem hrani zgodovino modifikacij programske kode, obenem pa podpira reverzibilnost. Kakršnekoli težave z aplikacijo se lahko naslovijo na strani <https://github.com/volkmaster/trubadur/issues>.

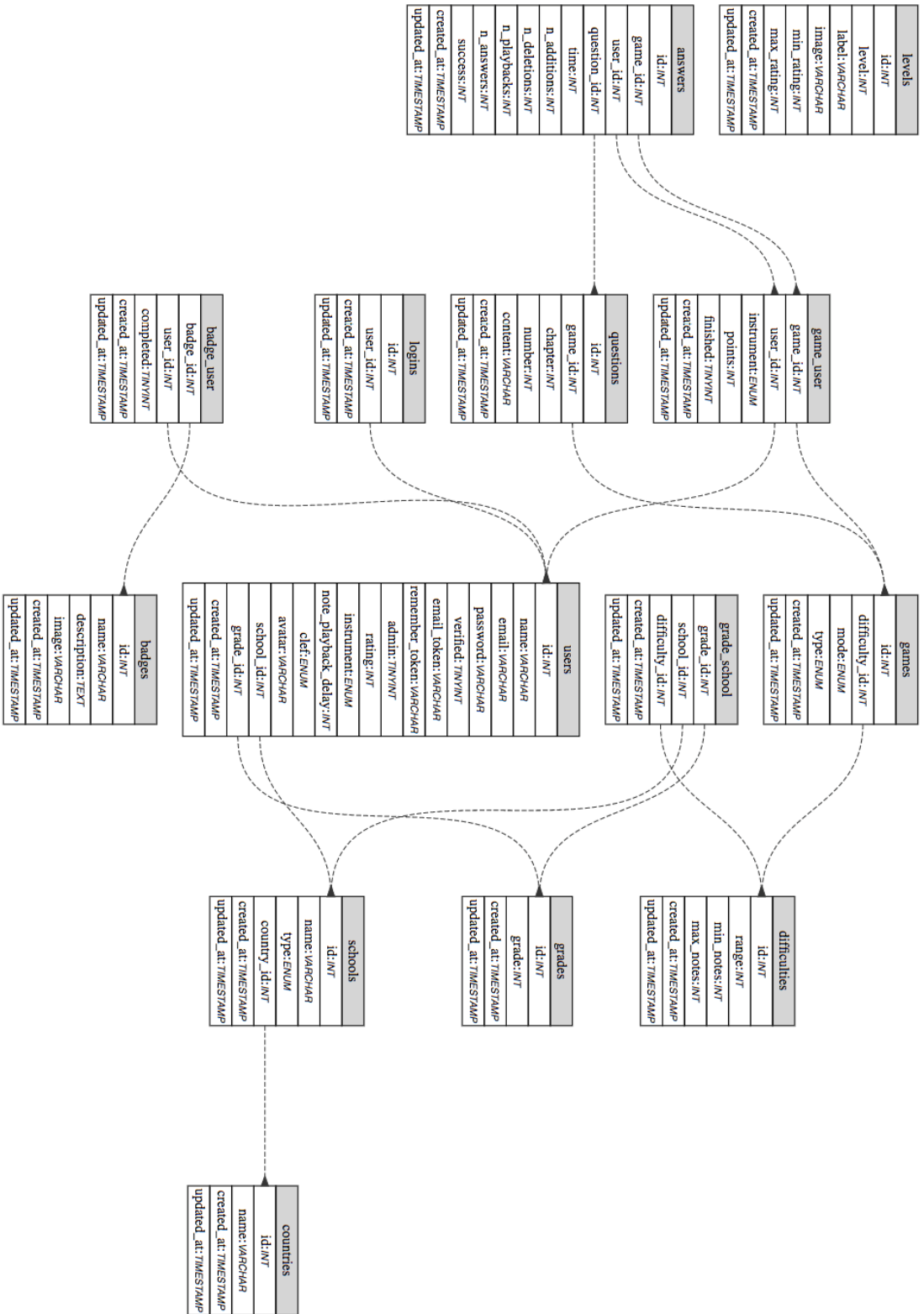
Globalno dostopna testna aplikacija je nameščena v okviru oblačne spletne platforme Heroku, na naslovu <https://trubadur.herokuapp.com>, med-

tem ko je globalno dostopna produkcijska aplikacija nameščena na spletnem strežniku Laboratorija za grafiko in multimedijo Fakultete za računalništvo in informatiko Univerze v Ljubljani. Nahaja se na spletnem naslovu `https://trubadur.si`.

4.2 Podatkovna baza

Aplikacija uporablja sistem za upravljanje z relacijskimi podatkovnimi bazami MySQL. Za začetno definicijo sheme podatkovne baze in za vse nadaljnje spremembe le-te uporabljamo migracije. Pri razvoju aplikacije smo za hitrejšo testiranje implementirali sejalice, ki so razvojno podatkovno bazo napolnili s testnimi podatki. Diagram podatkovne baze je razviden na Sliki 4.1.

Tabela `users` hrani podatke o uporabnikih, z njo povezana tabela `logins` pa beleži prijave uporabnikov. Vsakemu uporabniku sta pripisani šola (tabela `schools`), ki je locirana v določeni državi - tabela `countries` in razred (tabela `grades`), ki preko vmesne tabele `grade_school` določita nivo težavnosti iger uporabnika. Podatkovna logika povezana z igrami je shranjena v tabelah `games` (igre), `questions` (vprašanja), `answers` (odgovori) in vmesni tabeli `game_users`. Zapis v slednji predstavlja udeležbo določenega uporabnika v posamezni igri. Logiko vmesne tabele uporabimo, ker lahko zaradi večigralskega načina v posamezni igri sodeluje več uporabnikov. Glede na dejstvo, da smo v aplikaciji implementirali elemente poigritve, posamezni uporabnik zbira točke, ki vplivajo na nivo, katerega je trenutno dosegel. Informacije o slednjih so shranjene v tabeli `levels`. Z doseganjem določenih ciljev oz. izpoljevanjem specifičnih nalog se lahko uporabnik poteguje za doseg značk (tabela `badges`). Ker vsak uporabnik lahko pridobi več značk, le-te pa lahko pritičejo mnogim uporabnikom, obstaja še vmesna tabela `badge_user`.



Slika 4.1: Diagram podatkovne baze aplikacije.

4.3 Uporabniški vmesnik

4.3.1 Zgradba klienta

Klient je tehnološko gledano razdeljen na dva dela, kar je razvidno na Sliki 4.2. V prvem delu je implementirana logika za registracijo in avtentikacijo, komponente oz. pogledi pa so implementirani s pomočjo orodja za predloge im. Blade. Tej odločitvi je botrovalo dejstvo, da je bila ob inicializaciji novega projekta ogrodja Laravel, avtentikacijska logika - z njo povezani modeli, kontrolerji in pogledi - do določene mere že implementirana. Slednjo je bilo tako lažje prilagoditi naši aplikaciji, kot v nasprotnem primeru, ko bi morali logiko implementirati od samega začetka in se s tem izpostavili določenim varnostnim tveganjem, ki jih avtomatsko zgenerirana začetna logika že naslavlja. Prvi del klienta je tako sestavljen iz štirih pogledov, in sicer `Register` (registracija), `Login` (prijava), `ForgottenPassword` (pozaobljeno geslo) in `ResetPassword` (ponastavitev gesla).

Drugi del aplikacije je skoraj v celoti sestavljen iz hierarhično urejenih komponent ogrodja Vue.js, vanj pa vstopimo takoj po opravljeni avtentikaciji oz. prijavi. Takrat spletni brskalnik inicializira SPA aplikacijo Vue. Od tega trenutka naprej torej navigacija po aplikaciji poteka s pomočjo internega usmerjevalnika ogrodja Vue.js, hkrati pa se hrani lokalno stanje, saj se spletna stran ne osvežuje. Komponente Vue so hierarhično urejene in kot se kot take gnezdiyo med seboj. Na desni strani Slike 4.2 je hierarhija oz. gnezdenje komponent razvidno preko uporabe različnih barv za različne nivoje.

Na najvišjem nivoju se nahaja t.i. komponenta `App`, katero sestavljata komponenti `Dashboard` (predstavlja namizje oz. domačo stran aplikacije) in `HeaderMenu` (vsebuje glavo in meni aplikacije in vsebuje navigacijsko logiko).

Komponenta `Dashboard` nadalje igra vlogo vsebnika za komponente `GameTypes` (v njej izberemo tip igre, ki jo želimo igrati), `GameStatistics` (prikazuje podrobno statistiko odigrane igre), `Profile` (vsebuje profil uporabnika), `Leaderboard` (prikazuje lestvico uporabnikov urejeno glede na število zbranih točk) in `Settings` (vsebuje skupne nastavitve za igre).

Vsaka igra je lahko igrana v treh različnih načinih (vaja, en igralec in več igralcev), med katerimi uporabnik lahko izbira v komponenti `GameModes`, katere vsebnik je že omenjena komponenta `GameTypes`. Izbiri načina igre sledi komponenta, ki je specifična izbranemu tipu igre, v našem primeru obstaja zgolj komponenta `Intervals`, ki vsebuje izgled in logiko igre intervalov. Slednja je zgrajena iz komponent `Stave` (notno črtovje) in `Keyboard` (klaviatura), pri čimer notno črtovje deluje kot vsebnik za komponento `Note`, ki predstavlja posamezno noto, ki se izrisuje na črtovju.

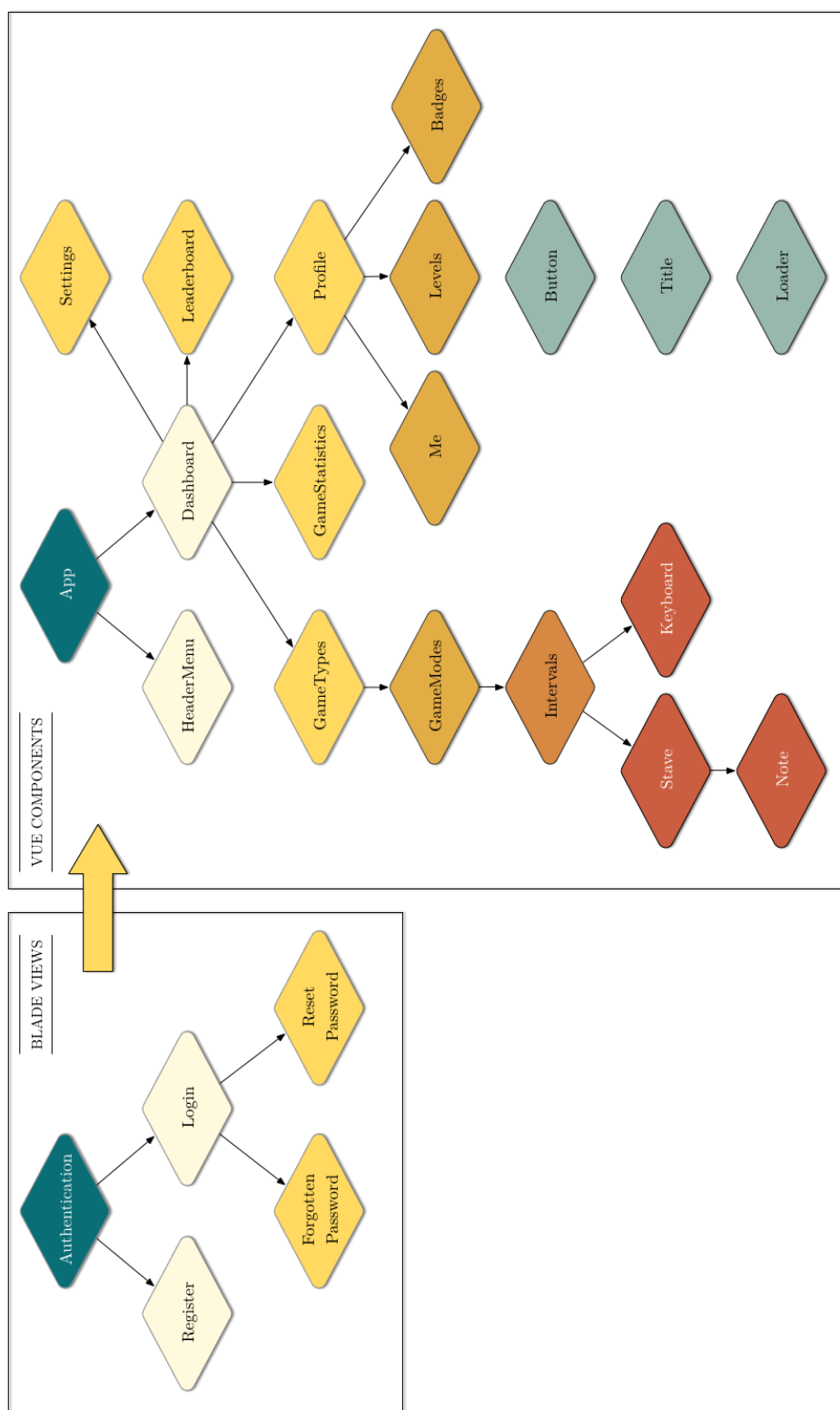
Komponenta `Profile` se nadalje deli v tri komponente in sicer `Me` (vsebuje informacije o uporabniku, njegovi uspešnosti in ponuja možnost za zamenjavo šole in razreda), `Levels` (prikazuje nivoje, ki jih je uporabnik z zbiranjem točk že dosegel in tiste, ki jih še lahko doseže) in `Badges` (prikazuje informacije o že opravljenih in neopravljenih dosežkih oz. značkah uporabnika).

Nabor komponent sestavljajo še tri dodatne komponente, ki pa niso direktno prisotne v hierarhiji, temveč se uporabljajo kot nekakšne pomožne komponente, ki jih večkrat uporabimo. Mednje spadajo komponente `Button` (predstavlja gumb), `Title` (predstavlja naslov oz. poudarjeno besedilo) in `Loader` (vsebuje grafični indikator, da je vsebina strani v procesu nalaganja).

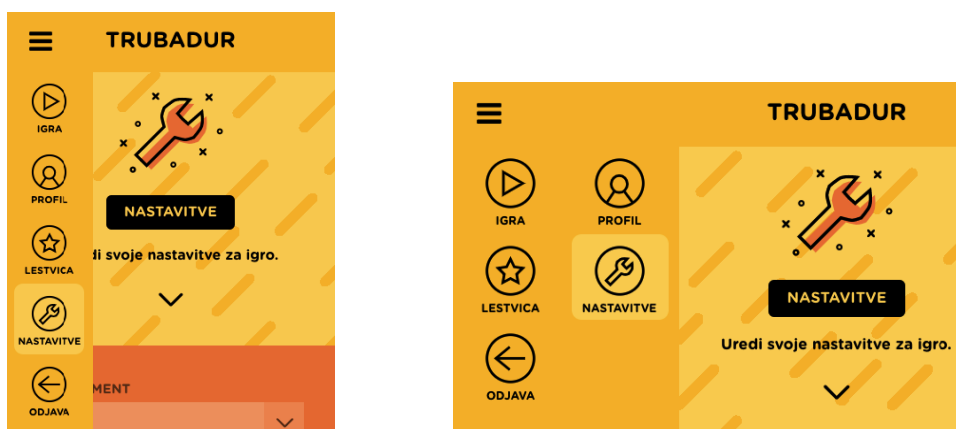
4.3.2 Navigacija

Uporabnik lahko za navigacijo po aplikaciji uporablja gumbе (primer na Sliki 4.4a). Prav tako lahko uporablja gumba nazaj in naprej, ki sta interni del vseh brskalnikov.

Večjo fleksibilnost pri navigaciji aplikaciji dodaja stranski meni (menu), ki je običajno skrit, odpremo pa ga lahko s klikom na ikono v levem zgornjem kotu glave (angl. *header*). Odprt meni in glava v pokončni (angl. *portrait*) orientaciji zaslona sta razvidna na Sliki 4.3a. Če se uporabnik trenutno nahaja na eni od strani, ki je navedena kot opcija v meniju, potem se mu kot indikacija obarva ozadje te opcije. V ležeči (angl. *landscape*) orientaciji zaslona odprt meni in glava dobita prilagojen izgled zaradi pomanjkanja vertikalnega prostora (Slika 4.3b). Prednost stranskega menija je možnost



Slika 4.2: Diagram pogledov Blade in komponent Vue, ki so sestavni del klienta aplikacije.



(a) Odprt meni in glava v pokončni orientaciji zaslona.

(b) Odprt meni in glava v ležeči orientaciji zaslona.

Slika 4.3: Meni in glava.

navigacije iz katerekoli strani na katerokoli drugo stran, saj je meni vedno prisoten. Dodatna navigacijska prvina pa je klik na naslov aplikacije, ki se nahaja v glavi. Slednji uporabnika preusmeri na domačo stran.

4.3.3 Vizualna podoba

Vizualna podoba aplikacije je preprosta, saj to omogoča boljše preglednost in lažjo uporabo. Kljub preprostosti je bil velik poudarek pri izdelavi aplikacije dan prav vizualni podobi, saj smo želeli doseči privlačno podobo, ki bi bila v skladu z modernimi pristopi k oblikovanju. Primarne barve v aplikaciji so naslednje barve in njihove izpeljanke: svetlejša rumena (#FFD15E), temnejša rumena (#FDBB2F), oranžna (#EB7D3D), zelena (#6CC075) in modra (#72B7AE). Ozadje na večini strani in podstrani je sestavljeno iz svetlorumenega in temno-rumenega vzorca. Opcije v meniju so vizualno obogatene z ikonami, kar lahko vidimo na Sliki 4.3a.

Z željo po konsistentnosti izgleda celotne aplikacije, smo za gumbe in naslove uporabili posebne oblike. Primer gumba je razviden na Sliki 4.4a, medtem ko naslov lahko vidimo na Sliki 4.5a. Določene strani imajo poleg naslova prisotno še ikono, ki vizualno poudari njen pomen (primer na Sliki



(a) Navigacijski gumb za premik na stran *intervali*.

(b) Zavihki na strani *profil*.

Slika 4.4: Navigacijski gumbi.

4.5b). Podobno velja za stran, kjer uporabnik izbira način v katerem želi igrati igri, saj je poleg besedila vsakemu izmed načinov dodeljena še specifična ikona (Slika 4.6b).

V okviru profilne strani so uporabniku na voljo trije zavihki, vsakemu izmed njih pa pripada ena od postrani *jaz*, *nivoji* ali *dosežki*. Za lažjo ločnico med tremi zaključenimi celotami, vsaka izmed njih uporablja eno dominantno barvo (razvidno na Sliki 4.4b). Tako zavihku *jaz* pripada oranžna barva, zavihku *nivoji* modra barva, zavihku *dosežki* pa zaznamuje zelena barva.

Skozi celotno aplikacijo se uporablja pisava **Gotham Rounded**, in sicer v treh različicah: **light**, **book** in **bold**. Z manjšo raznolikostjo pisav smo prav tako želeli doseči večjo konsistentnost v izgledu aplikacije.

Aplikacija je trenutno vizualno prilagojena veliki večini tabličnih in mobilnih zaslonov, kar je bila tudi naša primarna ciljna skupina. Izgled je torej prilagojen širini zaslona (angl. *responsive design*).

Aplikacija je prav tako prilagojena in brez težav deluje v najbolj pogosto uporabljenih mobilnih brskalnikih, kot so Google Chrome, Mozilla Firefox in Safari.

4.4 Igra intervalov

V obsegu magistrskega dela smo si za preizkus zamišljenega koncepta za implementacijo izbrali igro, pri kateri uporabniki lahko preizkusijo in izboljšajo svoje znanje ugotavljanja intervalov. V pričakovanju, da bodo na platformi v prihodnosti implementirane še druge igre drugega tipa, je bila logika na

(a) Primer naslova na strani *igre*.(b) Naslovna ikona na strani *lestvica*.**Slika 4.5:** Naslovi na straneh.

strežniku in klientu prilagojena tej ideji.

Če uporabnik želi igrati igro, se preko gumba na domači strani ali gumba v meniju premakne na stran, kjer lahko izbere tip igre, ki jo želi igrati. V našem primeru je to zgolj igra intervalov, kot je razvidno na Sliki 4.6a. Po izbiri tipa igre se premakne na stran, kjer izbere način, v katerem želi igrati igro: vaja, en igralec ali več igralcev (Slika 4.6b). Načini se razlikujejo med seboj, tako po poteku igre, kot tudi po točkovanju. Pri igri za vajo odgovorov ne upoštevamo v okviru sistema točkovanja, prvotno pa je namenjena učenju in ne tekmovanju, kot velja za ostala dva načina. Težavnost posameznega vprašanja oz. naloge je definirana s poltonskim razponom intervalov in številom not, ki jih mora uporabnik vnesti na notno črtovje.

Pri igri za vajo lahko uporabnik poljubno nastavlja omenjena parametra, pri čimer njegovi odgovori niso ne točkovani, ne časovno omejeni, medtem ko so pri načinu enega igralca vprašanja zastavljena glede na težavnost (določena na podlagi izbrane šole in razreda oz. letnika), razdeljena pa so v tri kategorije. V vsaki izmed kategorij mora uporabnik odgovoriti na osem vprašanj, težavnost pa se stopnjuje s kategorijami. Zaenkrat so definirane zgolj 3 skupine šol, in sicer nižje glasbene šole, srednje glasbene šole in šole glasbe na univerzitetnem nivoju. Vrednosti parametrov igre glede na izbrano šolo in razred oz. letnik so navedene v Tabeli 4.1. Način igre za več igralcev je

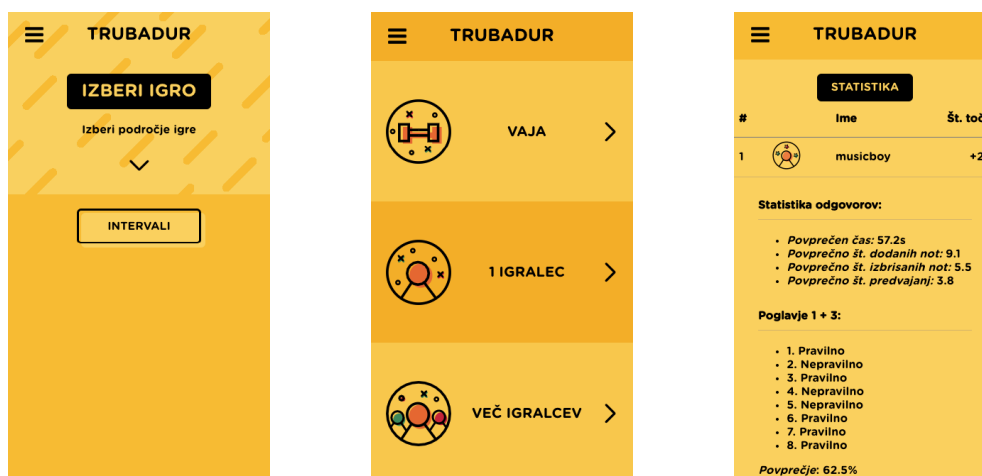
Tabela 4.1: Vrednosti parametrov igre intervalov glede na izbrano šolo in razred oz. letnik.

šola	razred/ letnik	poltonski razpon intervalov	število not
nižja	vsi	3	2 do 4
srednja	1.	5	4 do 6
	2.	12	4 do 6
	3.	12	6 do 8
	4.	12	6 do 8
univerzitetna	vsi	12	6 do 8

opisan v podpoglavju 4.4.3. Odgovori so časovno omejeni na dve minuti. Če ta čas preteče in uporabnik ni uspel pravilno odgovoriti na vprašanje, ga igra preusmeri na novo vprašanje. Hkrati velja, da ima uporabnik na voljo omejeno število odgovorov na posamezno vprašanje (od 2 do 5), ki se stopnjuje s težavnostjo vprašanj.

Po koncu igre je uporabnik avtomatsko preusmerjen na stran, kjer se mu prikaže podrobna statistika reševanja nalog (primer na Sliki 4.6c). Statistika zajema izračunano število točk, ki jih je uporabnik v igri zbral (pozitivno ali negativno število), povprečen čas, ki ga je potreboval za odgovarjanje na vprašanja, povprečno število dodanih in izbrisanih not in povprečno število predvajanja zaporedja tonov. Nadalje statistika zajema še podroben prikaz pravilnosti oz. nepravilnosti pri odgovarjanju na posamezna vprašanja znotraj kategorij in deleže (v procentih) pravih odgovorov po kategorijah in skupno.

Pri izračunu števila točk, ki jih je uporabnik v igri zbral, se upoštevajo sledeči faktorji: število generiranih tonov v zaporedjih, maksimalni poltonski razpon intervalov, pravilnost odgovora na posamezno vprašanje, število dodajanj in brisanj not na notnem črtovju in čas, ki ga je uporabnik potreboval



(a) Izbira tipa igre.

(b) Izbira načina igranja.

(c) Statistika igre intervalov.

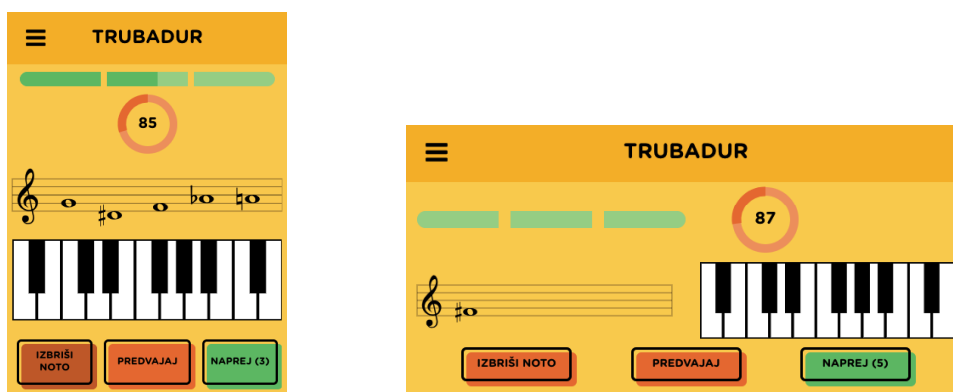
Slika 4.6: Izbira igre in statistika.

za odgovor na posamezno vprašanje.

4.4.1 Izgled igre

Uporabniški vmesnik igre intervalov sestoji iz naslednjih elementov:

- vrstice, ki prikazuje napredek (vizualno razdeljena v 3 skupine, ki predstavljajo 3 poglavja igre),
- odštevalnika časa, ki prikazuje čas, ki je uporabniku preostal za odgovor na vprašanje,
- notnega črtovja, kjer so izpisane note in klaviature, s pomočjo katere uporabnik vnaša note,
- kontrolnih gumbov za izbris zadnje note, ponovno predvajanje zaporedja tonov in odgovor (prikazuje tudi omejitev števila odgovorov, ki jih ima uporabnik še na voljo),
- obvestil, ki se izpišejo v primerih, ko uporabnik vnese premalo not ali ko je odgovor napačen.



(a) Izgled igre v pokončni orientaciji.

(b) Izgled igre v ležeči orientaciji.

Slika 4.7: Uporabniški vmesnik igre intervalov v različnih orientacijah.

Izgled igre intervalov je prilagojen različnim velikostim in orientacijam zaslona. Slika 4.7a prikazuje uporabniški vmesnik v pokončni orientaciji, Slika 4.7b pa v ležeči orientaciji.

4.4.2 Generator vprašanj

Delovanje algoritma za generiranje vprašanj pri igri intervalov je razvidno v Dodatku A.5. Od težavnosti naloge oz. vprašanja je odvisno število generiranih tonov in poltonski razpon intervalov v generiranem zaporedju tonov. Algoritem sprva generira naključen ton v obsegu od A3 do C5 in ga doda v vzorec. Generiranje zaporedja tonov nato poteka iterativno, pri čimer se vedno upošteva lastnosti predhodno generiranega tona. Koraki generiranja posameznega tona so sledeči:

1. Izračuna se poltonski razpon, ki nam je na voljo glede na predhodno generirani ton.
2. Z uteženo naključnostjo se izbere smer naslednjega tona (višja / nižja tonska višina).
3. Potencialno se omeji izračunani poltonski razpon glede na težavnost

naloge.

4. V omejenem razponu se naključno izbere naslednji ton, razlika v tonski višini med tem in predhodnim tonom pa predstavlja interval.
5. Če se je ton že dvakrat pojavil v generiranem vzorcu, se zavrže, v nasprotnem primeru se doda v vzorec.

Da smo preverili pravilno delovanje generatorja, smo generirali milijon zaporedij. Statistika generiranih tonov je razvidna v Tabeli 4.2a, statistika generiranih intervalov pa v Tabeli 4.2b. Opazimo, da je večina tonov generiranih z enako verjetnostjo, le pri robnih tonih opazimo rahel padec v verjetnosti pojavitve v zaporedju. Iz statistike generiranih intervalov pa je razvidno, da verjetnost pojavitve intervala pada premo sorazmerno z razdaljo med dvema tonoma oz. med njunima višinama. Edina opazna razlika je tu pri čisti prvi, saj smo namenoma zmanjšali verjetnost njene pojavitve in s tem preprečili, da bi bila generirana vprašanja prelahka.

4.4.3 Večigralski način

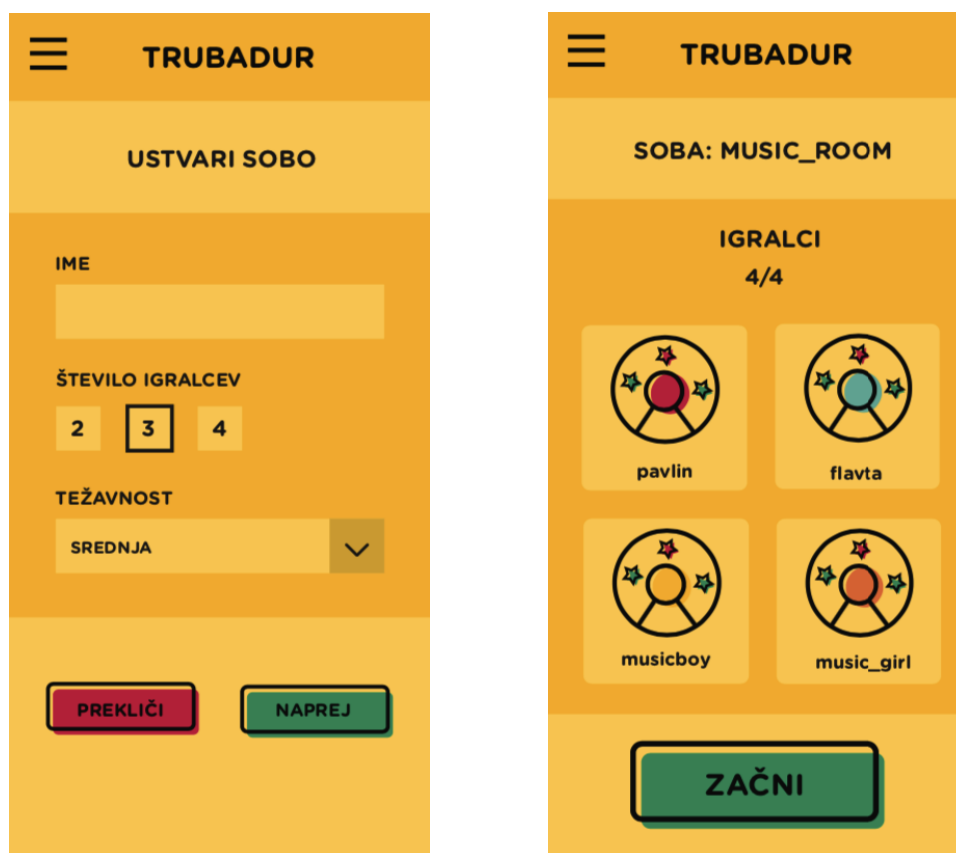
Večigralski način (angl. *multiplayer*) omogoča igralcem, da v realnem času tekmujejo med seboj v izbrani igri. Njegova implementacija izvira iz želje po povečanju motiviranosti in angažiranosti uporabnikov za uporabo platforme in predstavlja pomemben del procesa poigritve. V večigralskem načinu lahko uporabniki ustvarijo novo igro, pri čimer morajo v obrazec vnesti ime igre, število igralcev (od 2 do 4) in težavnost igre (Slika 4.8a). Ko ta korak opravijo, morajo počakati, da se njihovi igri pridružijo še drugi uporabniki. Po ustaljeni praksi se to čakanje opravi v t.i. sobi (angl. *lobby*, kar ni dobeseden prevod, vendar se v kontekstu večigralskih iger beseda soba v angleščino vedno prevaja s tem izrazom). Ko se v sobi (Slika 4.8b) zbere zadostno število uporabnikov, lahko vodja sobe oz. tisti, ki je igro ustvaril, le-to tudi zažene. V večigralskem načinu je čas za odgovor na vprašanje prav tako kot pri ostalih načinih igre omejen, naloga tekmovalcev pa je, da kar se da hitro

Ton	Delež	Interval	Delež
A#3	5 %	čista prima	6 %
B3	5,63 %	zvečana prima /	12 %
C4	6,13 %	mala sekunda	
C#4	6,57 %	velika sekunda	11 %
D4	6,63 %	mala terca	11 %
D#4	6,65 %	velika terca	10 %
E4	6,69 %	čista kvarta	9 %
F4	6,7 %	zvečana kvarta /	8 %
F#4	6,69 %	zmanjšana kvinta	
G4	6,69 %	čista kvinta	8 %
G#4	6,67 %	mala seksta	7 %
A4	6,63 %	velika seksta	6 %
A#4	6,55 %	mala septima	5 %
B4	6,14 %	velika septima	4 %
C5	5,62 %	čista oktava	3 %
C#5	5 %		

(a) Statistika generiranih tonov.

(b) Statistika generiranih intervalov.

Tabela 4.2: Statistika za milijon generiranih zaporedij tonov.



(a) Obrazec za ustvarjanje nove igre v večigralskem načinu.

(b) Soba, kjer uporabniki čakajo na začetek igre v večigralskem načinu.

Slika 4.8: Igra v večigralskem načinu.

odgovorijo na vprašanje, saj so točke, ki jih za pravilni odgovor prejmejo, v veliki meri odvisne od vrstnega reda odgovorov. Če torej tekmovalec odgovori prvi izmed vseh, bo pri tistem vprašanju prejel več točk kot ostali, ki so odgovorili kasneje.

4.5 Poigritev aplikacije

Da bi med uporabniki dosegli večjo motiviranost, smo se izobraževalno platformo odločili obogatiti z izbranimi elementi poigritve. Uporabnik znotraj platforme z igranjem iger dosega točke, le te pa direktno vplivajo na njegovo

mesto na lestvici (angl. *leaderboard*) in na dosego nivojev (angl. *levels*). Ob koncu vsake igre se posameznemu uporabniku izračunajo točke (angl. *points*), ki jih je pri dani igri dosegel (lahko so pozitivne ali negativne). Pri tem se upoštevajo težavnost nalog (v primeru intervalov upoštevamo razpon intervala in število not v zaporedju), čas potreben za odgovor na vprašanja, število dodanih in izbrisanih not (penalizira golo poskušanje) in ali je uporabnik pravilno odgovoril na vprašanje (število poskusov pri posameznem vprašanju je odvisno od težavnostnega nivoja, ki ga je uporabnik posredno izbral preko izbire šole in razreda oz. letnika, nikoli pa ne preseže števila 5).

Uporabniku se informacije, ki so osnovane na elementih poigritve prikazujejo že na domači strani, prikazani po prijavi (Slika 4.9a). Podrobne informacije o sebi si lahko uporabnik ogleda v profilu oz. znotraj njegovih zavihkov. Mednje spadajo *jaz*, *nivoji* in *dosežki*. V zavihku *jaz* (Slika 4.9c in Slika 4.9d) uporabnik lahko vidi svojo profilno sliko, uporabniško ime, število točk, ki jih je zbral, številčni in grafični prikaz doseženega nivoja ter do tri zadnje dosežene značke. Prav tako lahko zamenja profilno sliko, uporabniško ime, šolo in razred oz. letnik.

Vsebina zavihka *nivoji* (Slika 4.10a) predstavlja grafični prikaz doseženih nivojev. Iz tega uporabnik lahko razbere katere pomembne nivoje je že dosegel in katere še mora oz. kateri še sledijo. Definirali smo naslednje nivoje (urejeni v naraščajočem vrstnem redu): “Recital za Glasbeno Mladino v Cankarjevem domu”, “Nastop s Komornim godalnim orkestrom Slovenske Filharmonije”, “Snemanje v studiih RTV Slovenija”, “Solo nastop z Orkestrom Slovenske Filharmonije”, “Sodelovanje v Evropskem mladinskem orkestru”, “Solo nastop v koncertni dvorani Concertgebouw v Amsterdamu”, “Igranje na evropski turneji s koncertno agencijo European Artistic Services”, “Finale Evropskega tekmovanja mladih glasbenikov” in “Zmaga na Evropskem tekmovanju mladih glasbenikov”.

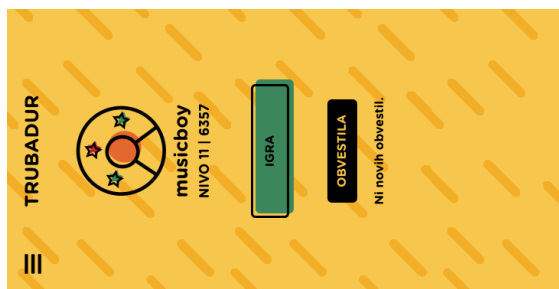
V zadnjem zavihku *značke* uporabnik vidi katere dosežke je že opravil in katerih še ni. Vsak izmed dosežkov je predstavljen s sliko in z opisom, ki definira pogoje za njegovo izpolnitev. Primer dosežkov je razviden na Sliki

4.10b. Uporabniki lahko pridobijo naslednje značke: “Igra brez napake”, “Igra s 50% točnostjo”, “Igra končana v 25 minutah”, “Dokončana igra 3 dni zapored”, “Dokončana igra 7 dni zapored”, “Dokončana igra z vsemi različnimi inštrumenti”, “Prijava 3 dni zapored”, “Prijava 7 dni zapored” in “Zmaga v večigralski igri”.

Na lestvici (Slika 4.9b), ki je urejena padajoče glede na zbrane točke, si lahko uporabnik ogleda, kako uspešen je napram drugim uporabnikov, ki prav tako uporabljajo platformo. Če izbere enega od uporabnikov, ga to pripelje na profilno stran slednjega, kjer si lahko ogleda vse informacije o njem, ki se nanašajo na elemente poigritve, enako kot je to lahko storil zase.

4.6 Modul za učitelje

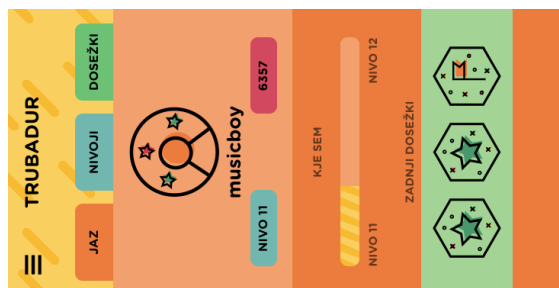
Modul za učitelje je prvenstveno namenjen pedagogom, ki želijo spremljati, nadzorovati in voditi potek učenja in ovrednotiti rezultate učencev pri reševanju nalog znotraj platforme. Učitelj tako lahko učencem razpiše domačo nalogo, ki jo morajo rešiti do določenega roka. Pri tem natančno opredeli tip in obseg naloge, tekom reševanja in po preteku roka za reševanje, pa ima na voljo rezultate o uspešnosti učencev. Na podlagi slednjih se lahko odloči o nadaljnjih korakih izobraževanja.



(a) Domača stran.

MESTO	IME	TOČKE
1	musicboy	6537
2	Antonio Vivaldi	1452
3	Ludwig van Beethoven	1450
4	Wolfgang Amadeus Mozart	1302
5	Joseph Haydn	1024

(b) Lestvica uporabnikov.

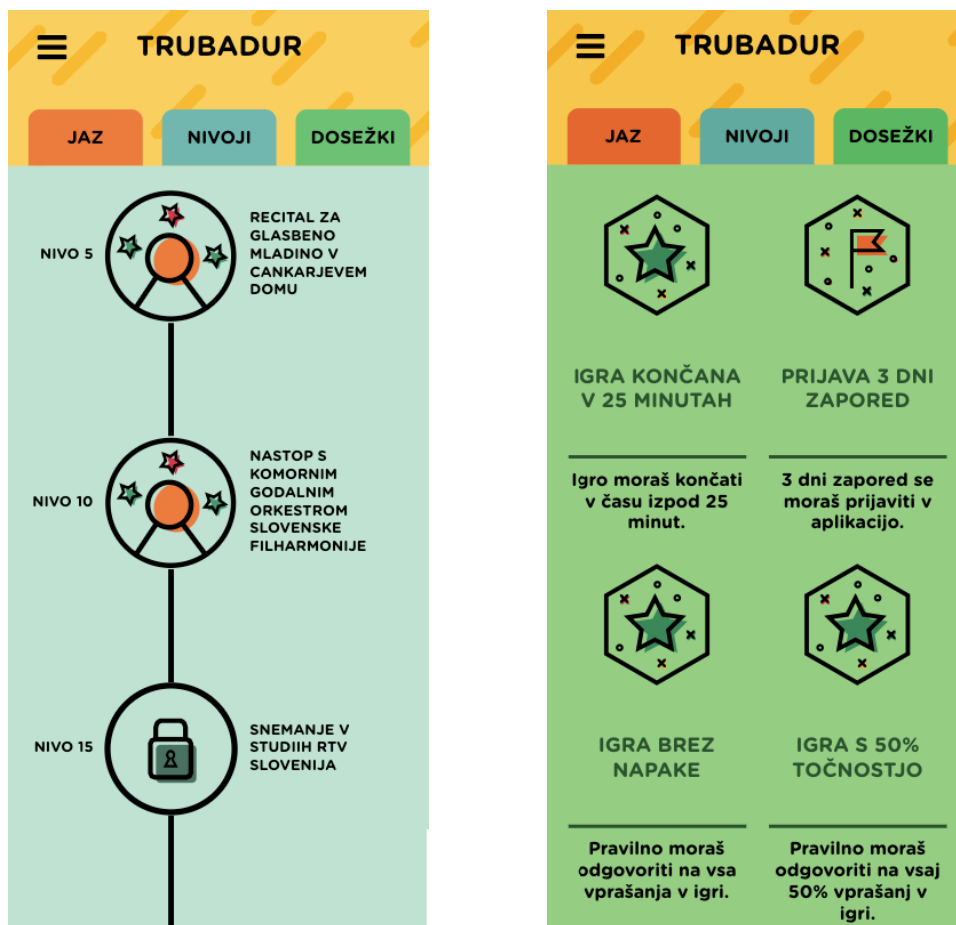


(c) Profil uporabnika v zavihku jaz.



(d) Zamenjava profile slike.

Slika 4.9: Elementi poigrive.

(a) Zavihek *nivoji*.(b) Zavihek *dosežki*.

Slika 4.10: Elementi poigritve.

Poglavje 5

Rezultati in analiza

Aplikacijo smo testirali na izbrani skupini dijakov 1. in 2. letnikov Srednje glasbene šole Konservatorija za glasbo in balet Ljubljana.

5.1 Intervju

Po izdelavi prototipa aplikacije smo izvedli intervju z dvema učencema, s katerim smo želeli, še pred samim testiranjem aplikacije, pridobiti povratno informacijo o potencialnih problemih oz. frustracijah, ki bi jih uporabniki morda doživljali pri uporabi aplikacije.

Intervju je bil razdeljen v štiri sklope (natančen opis v Dodatku B.1):

- **Profil (aktivno spraševanje):** Učenec je odgovarjal na vprašanja povezana z veščinami, ki jih ima pri uporabi mobilnih aplikacij (tudi bolj specifično - pri uporabi mobilnih aplikacij s področja glasbe).
- **Uspešnost opravljanja nalog (pasivno opazovanje):** Učencu smo zastavili manjše cilje, ki jih je moral izpolniti, pri tem pa smo se vprašali, če je do danega cilja prišel in koliko časa je za to potreboval. Če mu to ni uspelo, nas je zanimalo, kako daleč mu je uspelo priti.
- **Učinkovitost igre (pasivno opazovanje):** Učenec je moral odigrati

igro intervalov, mi pa smo pri tem opazovali, koliko pomoči je potreboval pri reševanju nalog, kakšnega tipa je bila pomoč in če je le-ta bila verbalna, kakšna vprašanja je zastavil. Obenem smo opazovali, če je pred začetkom igre prebral navodila in kako dolgo jih je bral ter ali je med samim reševanjem vedel, na kateri točki igre se nahaja (ali je imel informacijo o napredku).

- **Zadovoljstvo (aktivno spraševanje):** Učencu so bila zastavljena vprašanja o igri, ki so se nanašala na njeno zahtevnost, zabavnost, frustracije, ki jih je morebiti doživel ob igranju igre ali uporabi aplikacije, itd.

Pri opravljanju nalog učenca v splošnem nista imela težav in sta do cilja prišla zelo hitro. Eden od ciljev, ki jima je predstavljal težave je bil "Pojdi na domačo stran", saj v meniju ni bilo te opcije in bi morala za izpolnitev cilja pritisniti na izpisano ime aplikacije v glavi. Drugi cilj, kjer sta učenca imela težave je bil "Postavi mobilno napravo v ležeč položaj in se poskušaj orientirati po aplikaciji". Pri slednjem je bilo opaziti, da se del menija (ki je bil malenkost prevelik), skriva na dnu, kar jima je otežilo prehajanje med podstranmi aplikacije.

Pri igranju igre intervalov je bilo opaziti, da sta učenca porabila različno količini časa za seznanitev z igro preko navodil. Prvi učenec je le-temu posvetil dovolj časa in zato tudi pri samem reševanju igre ni imel nobenih omembe vrednih težav, drugi učenec pa ni dovolj podrobno prebral navodil in je imel zato težave pri reševanju nalog. Eden od problemov je bil, da ni vedel, da mora za zapis not uporabiti klaviaturo, in je zato skušal vnesti note s pritiskanjem na notno črtovje.

V sklopu aktivnega spraševanja o zadovoljstvu pri igri sta učenca izpostavila, da igra ni preveč zahtevna. Izpostavila sta, da če je neko vprašanje pretežko, ga lahko še vedno rešiš s pomočjo ponovnega predvajanja in poskušanja. Igra se jima ni zdela pretirano zabavna, želela bi si več različnih načinov igre in več iger. Poudarila sta, da se jima igra zdi boljši način učenja intervalov, kot konvencionalni način in da bi igro uporabljala za vajo pred

testom. Kot glavno pomanjkljivost oz. slabost igre sta izpostavila nizko kakovost zvoka inštrumentov in dejstvo, da je največja možna pavza med toni prekratka. Eden od učencev je kot prednost igre izpostavil vnašanje not na notno črtovje preko klaviature, saj lahko na ta način vadi igranje klavirja, ki ni njegov primarni inštrument. Ugotovili smo tudi, da bi učenca v navodila dodala informacijo o številu not, ki jih moraš izbrati pri posameznem vprašanju določenega poglavja. Za konec je eden od učencev predlagal še, da bi obstajala možnost za zaustavitev predvajanja zaporedja tonov, saj je bila do tedaj implementirana zgolj možnost za začetek predvajanja in ne za zaustavitev pred koncem. Drugi učenec pa je prelagal, da bi imela igra v načinu za vajo na voljo paleto različnih nastavitvev, ki bi jih lahko med igro dinamično prilagajal, in bi s tem na hiter način prehajal med različnimi načini igranja igre.

Na podlagi spoznanj intervjuja smo popravili izgled aplikacije v ležeči orientaciji in dodali gumb za prehod na domačo stran v meni. Obenem smo skušali izboljšati navodila, tako da učenci ne bi imeli težav med samim reševanjem, prav tako pa smo omogočili možnost, da si učenec naknadno lahko ogleda navodila med igranjem igre. Na podlagi ocene, da vprašanje lahko rešuješ s poskušanjem, smo število odgovorov omejili na 2 do 5 (odvisno od težavnosti igre). Pripombo o nekakovostnem zvoku inštrumentov smo naslovili z boljšimi zvočnimi posnetki, obenem pa smo podaljšali najdaljšo možno pavzo med predvajanimi toni. Upoštevali smo tudi dodatni ideji učencev in dodali možnost zaustavitve predvajanja zaporedja tonov in igro v načinu za vajo popolnoma spremenili v skladu s predlogom učenca.

S tem smo že pred testiranjem zaključili prvi krog iteracije izboljševanja aplikacije.

5.2 Testiranje aplikacije

Nadalje smo izvedli metodo A/B testiranja [46], pri čimer smo za posamezen letnik definirali testno in kontrolno skupino. Učenci v testnih skupinah so

aplikacijo v okviru uradnega testiranja uporabili dvakrat, medtem ko učenci iz kontrolnih skupin slednje niso uporabljali. Učenci iz testnih skupin so se pri uporabi aplikaciji prijavi v platformo, se po njej orientirali in na koncu seveda odigrali igro intervalov. Na samem koncu so vsi učenci, tako iz testnih, kot iz kontrolnih skupin, rešili preverjanje iz znanja intervalov na konvencionalen način - kot to delajo običajno, s pomočjo klavirja in zapisom not na papir. S tem preverjanjem smo želeli preveriti, če obstaja korelacija med uporabo aplikacije in vajo intervalov na naši platformi in uspehom pri preverjanju znanja intervalov na konvencionalen način. Zanimalo nas je torej, če obstaja razlika v uspehu pri preverjanju med učenci iz testne in kontrolne skupine.

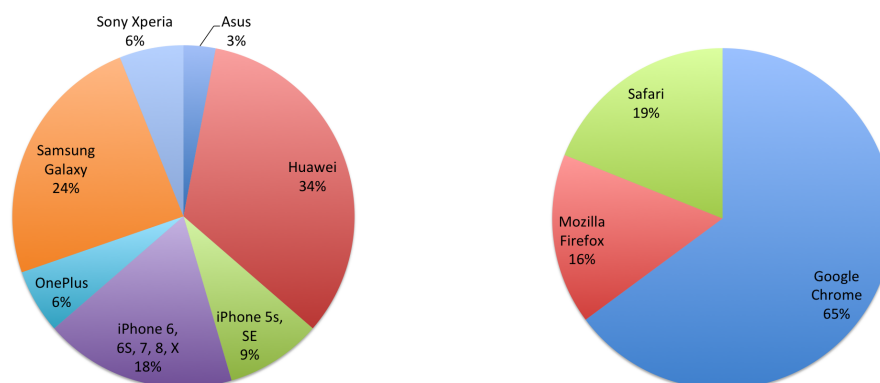
Učencev 1. letnika v testni skupini je bilo 6, v kontrolni skupini pa 5, medtem ko je bilo učencev 2. letnika v testni skupini 13, v kontrolni skupini pa 9. Učencev v testnih skupinah je bilo torej 19, učencev v kontrolnih skupinah pa 14.

5.2.1 Vprašalnik o profilu

Učenci obeh skupin so rešili profilni vprašalnik (Dodatek B.2), v katerem so poleg navedbe svoje starosti odgovorili še na vprašanja o svojih navadah glede igranja računalniških ali mobilnih iger in glede uporabe mobilnih aplikacij. S tem vprašalnikom smo skušali ugotoviti, kako dobro se učenci znajdejo v mobilnih aplikacijah in kakšne implikacije bi to morebiti imelo na uporabo naše mobilne aplikacije.

Na podlagi profilnega vprašalnika smo izvedeli, katere mobilne naprave in katere mobilne brskalnike uporabljajo učenci, kar nam je dalo dodatno informacijo o potrebni prilagoditvi uporabniškega vmesnika. Na podlagi podatkov, ki so grafično prikazani v tortnih diagramih 5.1a in 5.1b, smo tako lahko ocenili, kakšne širine zaslonov mora naša aplikacija podpirati in katere mobilne brskalnike moramo podpreti.

Opazimo, da je največ učencev uporabilo napravo znamke Huawei, sledita pa ji Samsung Galaxy in različni tipi naprav iPhone. Delež ostalih



(a) Deleži mobilnih naprav, ki so jih uporabljali učenci med testiranjem.

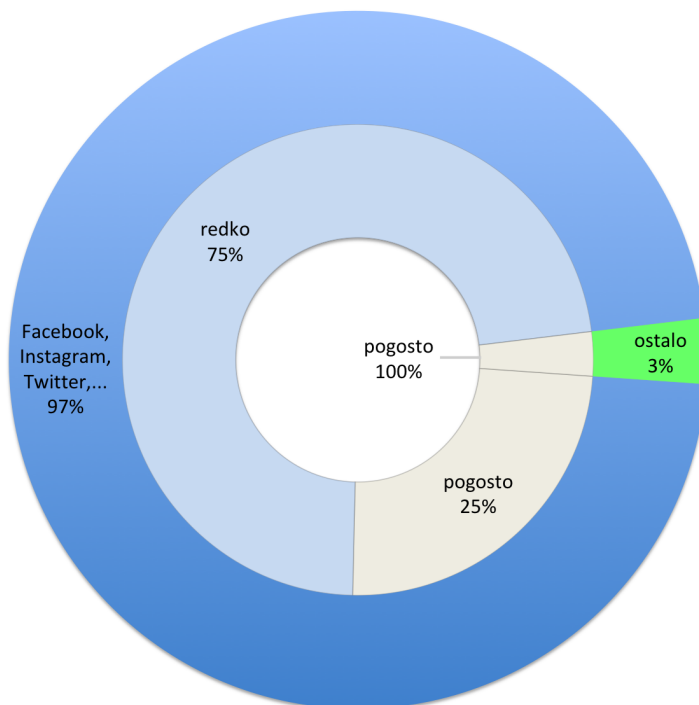
(b) Deleži mobilnih brskalnikov, ki so jih uporabljali učenci med testiranjem.

Slika 5.1: Mobilne naprave in brskalniki.

mobilnih naprav je relativno majhen, zato ga pri analizi nismo upoštevali. Mobilne naprave smo razdelili v dve skupini s ciljem identifikacije minimalne in maksimalne širine zaslona, ki jo moramo podpreti. Med naprave, ki imajo bistveno nižjo širino zaslona in so v fizično manjše smo uvrstili iPhone 5s in SE, med naprave z večjo širino in višjo ločljivostjo zaslona pa smo umestili telefone Huawei, Samsung Galaxy in različne tipe naprav iPhone (6, 6S, 7, 8 in X). Prva skupina nam je dala informacijo o minimalni širini zaslona, ki jo moramo še podpreti, druga skupina pa nasprotno o maksimalni širini.

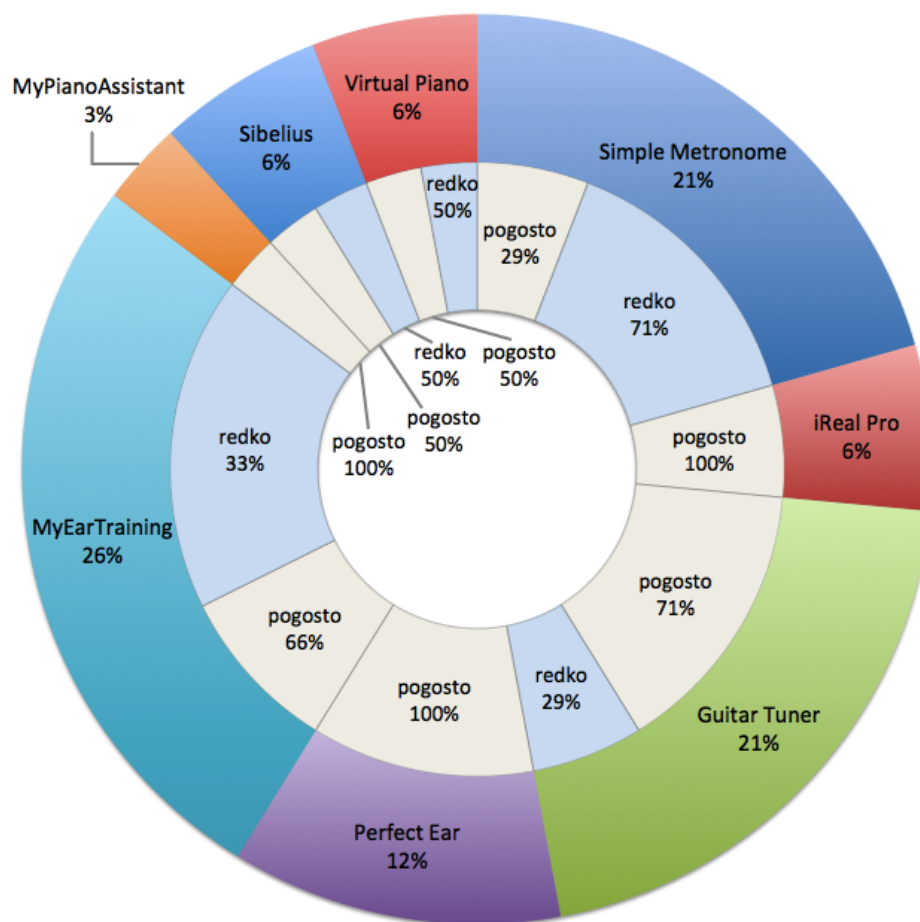
Pri analizi deležov uporabljenih mobilnih brskalnikov opazimo, da Google Chrome zavzema večinski delež, medtem ko imata Safari in Mozilla Firefox prav tako nezanemarljiv delež. Na podlagi tega dejstva smo se odločili podpreti vse tri brskalnike.

Rezultati profilnega vprašalnika nam prav tako ponujajo informacijo o aplikacijah, ki jih učenci uporabljajo. To nam pove, kako večji uporabniki mobilnih naprav so učenci. Podatki so grafično prikazani na tortnih diagramih 5.2 in 5.3. Opazimo, da učenci, ki uporabljajo mobilne aplikacije, v veliki večini uporabljajo aplikacije, ki se navezujejo na socialna omrežja



Slika 5.2: Deleži mobilnih aplikacij, ki jih uporabljajo učenci.

(Facebook, Instagram, Twitter, itd.), glavnina izmed njih pa to počne redko. Kar se tiče aplikacij za učenje glasbe oz. aplikacij, ki se navezujejo na glasbo v splošnem, med najbolj pogosto uporabljene spadajo MyEarTraining, Perfect Ear, Simple Metronome in Guitar Tuner. Prvi dve spadata v skupino aplikacij za učenje glasbe, medtem ko zadnji dve predstavljata aplikaciji, ki delujeta kot pripomoček glasbeniku. Pomembno je poudariti, da zgolj aplikacijo Perfect Ear, ki spada med aplikacije za učenje glasbe, učenci uporabljajo pogosteje kot redkeje. Obenem velja, da 10 učencev, ki so rešili vprašalnik, aplikacij za učenje glasbe oz. drugih aplikacij za glasbo sploh ne uporabljajo.



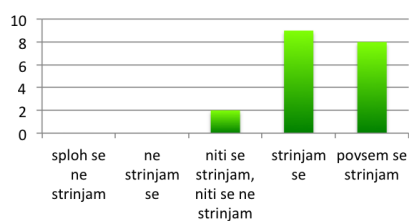
Slika 5.3: Deleži mobilnih aplikacij s področja glasbe, ki jih uporabljajo učenci.

5.2.2 Vprašalnik o aplikaciji

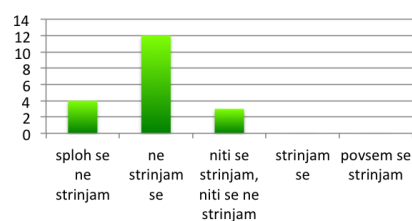
Po končanih dveh iteracijah testiranja aplikacije so učenci testnih skupin iz obeh letnikov rešili anketni vprašalnik, ki se navezuje na njihovo izkušnjo pri uporabi naše aplikacije (angl. *user experience*). Vprašalnik je v grobem razdeljen v 3 sklope, in sicer uporabniškega, tehničnega in pedagoškega. Vprašanja z uporabniškega področja se navezujejo na uporabniško izkušnjo pri uporabi aplikacije in na izgled uporabniškega vmesnika, vprašanja s tehničnega področja preverjajo mnenje uporabnikov glede zmogljivosti in odzivnosti aplikacije, vprašanja s pedagoškega področja naslavljajo učni aspekt uporabe aplikacije.

Še posebej nas je zanimal uporabniški aspekt vprašalnika, saj je ta del vključeval vprašanja v zvezi s poigritvijo aplikacije. Izkazalo se je, da je v povprečju večini učencev všeč uporabniški vmesnik oz. izgled aplikacije kot platforme (Slika 5.4a), obenem pa se je večina opredelila v prid dejstvu, da je uporabniški vmesnik igre intervalov uporaben (Slika 5.4c). Glede navigacije po aplikaciji med učenci obstaja prevladujoče mnenje, da je razdelitev vsebine po podstraneh smiselna (Slika 5.4b), da informacije, ki jih iščejo, najdejo hitro (Slika 5.4e) in da z orientacijo po aplikaciji nimajo velikih težav (Slika 5.4d). Prav tako učencem prilagajanje uporabniških nastavitev ne predstavlja velike ovire (Slika 5.4f). Kar se tiče vprašanj, ki se direktno navezujejo na elemente poigritve, je opaziti manj splošnega odobravanja. Učenci so v povprečju popolnoma neopredeljeni glede vprašanja, ali jih želja po doseganju značk in nivojev spodbuja k reševanju nalog (Slika 5.4g). Prav tako kar velik odstotek učencev ne mara dejstva, da lahko drugi uporabniki platforme vidijo njihove točke na lestvici (Slika 5.5a), a gre obenem omeniti, da jim je v splošnem sistem točkovanja in tekmovanja z drugimi všeč (Slika 5.4h). Ker v času raziskave aplikacija še ni podpirala večigralskega načina, smo učencem zastavili vprašanje, če bi aplikacijo uporabljali bolj pogosto, če bi le-ta način igre bil na voljo. Glede tega so v povprečju nekako neopredeljeni oz. ni zaznati pretirane želje po tem načinu igre (Slika 5.5b).

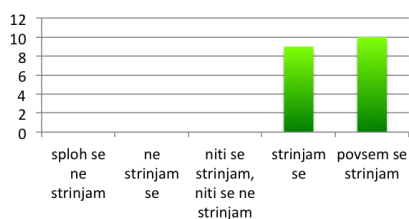
V okviru tehničnega sklopa vprašalnika so učenci večinsko izrazili mnenje,



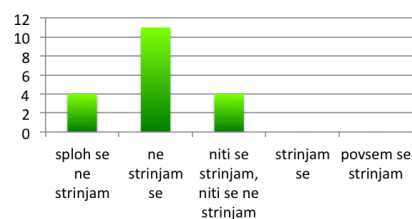
(a) Všeč mi je izgled (dizajn) uporabniškega vmesnika.



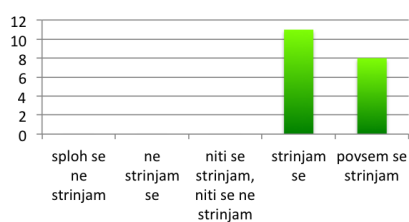
(b) Razdelitev vsebine po podstraneh ni smiselna.



(c) Uporabniški vmesnik igre (klavir, notno črtovje in ostali gumbi) je uporaben.



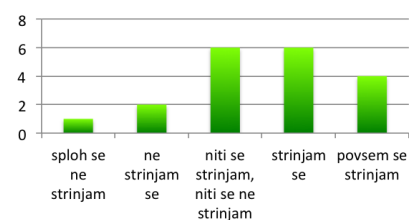
(d) Imam težave z orientacijo in navigacijo po aplikaciji.



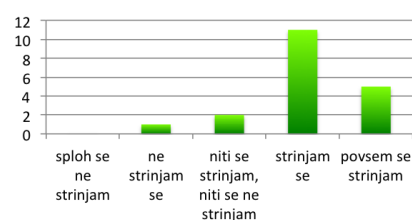
(e) Hitro najdem informacije, ki sem jih iščem.



(f) Prilagoditev nastavitvev igre in profila (šola, razred) mi predstavlja težave.

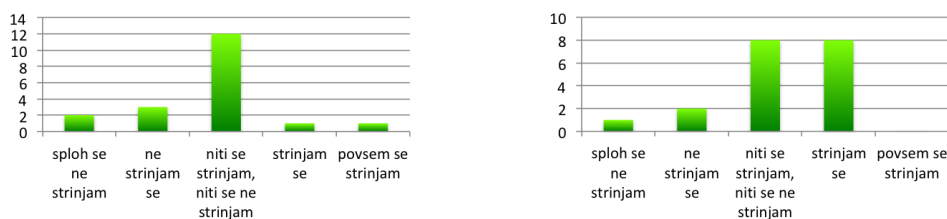


(g) Želja po doseg/izpolnitvi značk (badges) in nivojev (levels) me spodbuja k reševanju nalog.



(h) Všeč mi je sistem točkovanja (le-stvica) in tekmovanje z drugimi.

Slika 5.4: Histogrami odgovorov na vprašanja s področja uporabniške izkušnje.



(a) Ni mi vseč dejstvo, da vsi lahko vidijo točke na lestvici.

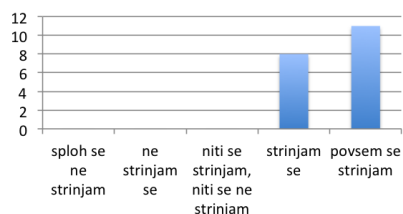
(b) Če bi bil na voljo večigralski način igre, bi bolj pogosto uporabljal aplikacijo.

Slika 5.5: Histogrami odgovorov na vprašanja s področja uporabniške izkušnje (nadaljevanje).

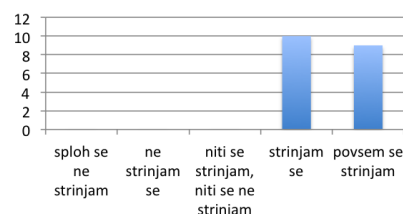
da uporaba aplikacije ni zahtevna (Slika 5.6a) in da so se njene uporabe hitro naučili (Slika 5.6f). Obenem ugotavljamo, da se je določen odstotek učencev med uporabo aplikacije počutil frustriranega (Slika 5.6g), prav tako pa so določeni učenci izrazili mnenje, da aplikacija ni dovolj odzivna (Slika 5.6d). V zvezi s tehničnim aspektom same igre, med učenci obstaja prevladujoče mnenje, da so navodila za igro potrebna (Slika 5.6c) in da so jim bila v pomoč (Slika 5.6b) ter da med samo igro niso začitili časovnega pritiska (Slika 5.6e).

Učenci so se v okviru pedagoškega sklopa vprašanj večinsko opredelili, da jih aplikacija spodbuja k učenju intervalov (Slika 5.7a) in da bi igro priporočili prijatelju ali znancu (Slika 5.7b), hkrati pa so po večini dokaj pozitivno odgovorili na vprašanje, ali bodo aplikacijo uporabljali še naprej (Slika 5.7g). Obenem so učenci izrazili mnenje, da ima aplikacija na voljo premalo različnih tipov nalog (Slika 5.7d) in se po drugi strani strinjali s trditvijo, da bi jih večja raznolikost iger bolj pritegnila k sodelovanju in udejstvovanju na učni platformi (Slika 5.7c). Kar se tiče zadovoljstva z igranjem igre, ugotavljamo, da so se učenci povečini med igranjem zabavali (Slika 5.7f), vendar se jim je sama igra zdela dovolj zahtevna oz. jim je predstavljala izziv (Slika 5.7e).

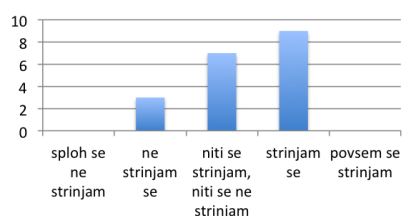
Rezultati kažejo na to, da je izgled uporabniškega vmesnika naletel na odobravanje in da je bila uporabniška izkušnja večine učencev zelo pozitivna. Na podlagi tega sklepamo, da je bil naš poudarek izgledu aplikacije tekom



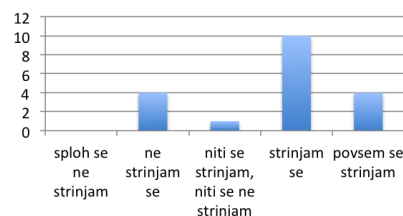
(a) Uporaba aplikacije ni zahtevna.



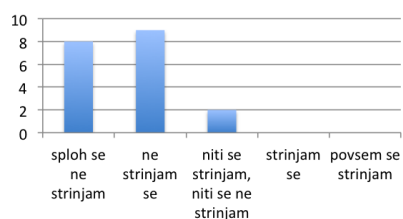
(b) Navodila za igranje so ustrezna (so mi bila v pomoč).



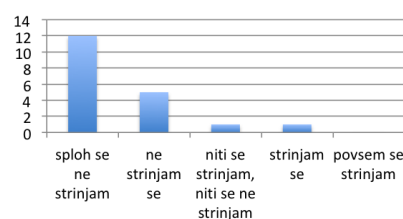
(c) Navodil za uporabo sploh ne bi potreboval/a.



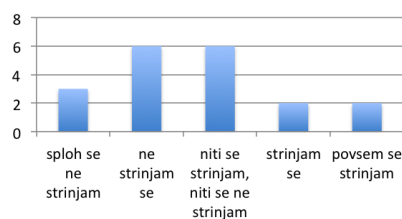
(d) Aplikacija je odzivna (deluje hitro).



(e) Za reševanje nalog imam premalo časa.

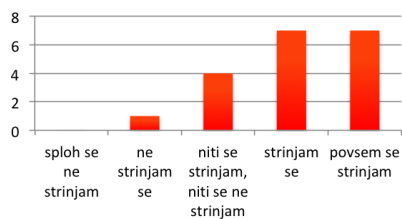


(f) Za učenje uporabe aplikacije potrebujem preveč časa.

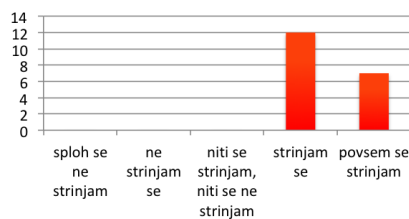


(g) Med uporabo aplikacije se na trenutke počutim frustriranega.

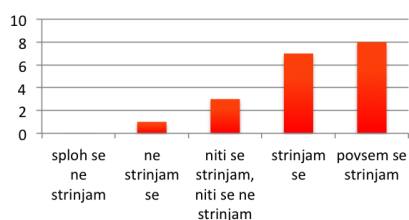
Slika 5.6: Histogrami odgovorov na vprašanja s tehničnega področja.



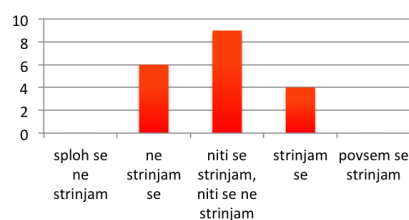
(a) Aplikacija me spodbuja k učenju intervalov.



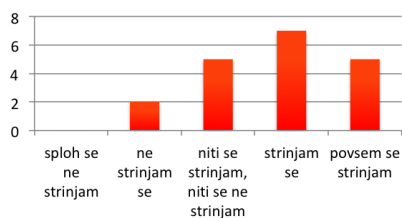
(b) Igro bi za učenje priporočil/a prijatelju/znancu.



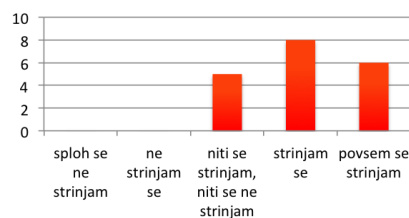
(c) Če bi imela aplikacija na voljo več različnih iger, bi me bolj pritegnila k sodelovanju.



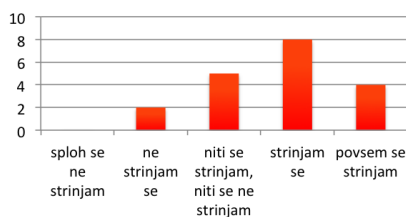
(d) Aplikacija ima na voljo dovolj različnih tipov nalog.



(e) Igra se mi ne zdi zahtevna.



(f) Igra se mi zdi zabavna.



(g) Aplikacijo bom uporabljal/a še naprej.

Slika 5.7: Histogrami odgovorov na vprašanja s pedagoškega področja.

razvoja smiseln.

Rezultati obenem kažejo na to, da učencev elementi poigritve, kot so značke, nivoji, točke in lestvica, niso posebej navdušili oz. jim niso dodatno motivirali k udejstvovanju na platformi in k igranju iger. Potencialni krivci za to so kratko časovno obdobje testiranja (približno en mesec) in dejstvo, da je bilo učencem na voljo premajhno število različnih iger.

Obenem je pomembno poudariti, da so tipi uporabnikov aplikacije lahko zelo različni, kar močno vpliva na njihovo motivacijo. Določeni uporabniki se namreč raje socializirajo, določeni raje tekmujejo, nekateri radi zbirajo značke, spet drugi pa radi raziskujejo. Glede na dejstvo, da se pri sami implementaciji nismo posebej osredotočili na različne tipe uporabnikov in da so bili izzivi, s katerimi so se soočili, za vse enaki, je to morda tudi eden od razlogov za nizko stopnjo angažiranosti uporabnikov za uporabo aplikacije.

Iz rezultatov lahko sklepamo, da je bila implementacija aplikacije v tehničnem smislu zadovoljiva, pomanjkanje odzivnosti in frustracije, ki so jih določeni učenci doživeli pri uporabi, pa lahko pripišemo široki paleti mobilnih naprav in mobilnih brskalnikov, ki so jih uporabili. Naprave uporabljajo različne operacijske sisteme in imajo različne resolucije zaslona, zato se morata tako izgled kot tudi logika delovanja aplikacije prilagoditi vsem posebnostim. Skozi čas bi se z iterativnim izboljševanjem aplikacije te težave dalo nasloviti in odpraviti.

Večina učencev je izrazila mnenje, da jih aplikacija spodbuja k učenju intervalov, da bo aplikacijo uporabljala še naprej in da bi jo priporočila prijateljem in znancem. Na podlagi teh dejstev sklepamo, da je uporabnikom všeč ideja aplikacije, ki jim ponuja drugačen način učenja glasbene teorije, kot so ga vajeni iz šolskih klopi.

5.2.3 Vprašnik za kontrolno skupino

Z namenom primerjave rezultatov končnega testiranja smo učence iz kontrolnih skupin povprašali, koliko ur na teden, so tekom našega testiranja aplikacije na testni skupini, vadili prepoznavanje intervalov na konvencio-

nalni način. Izkazalo se je, da so učenci intervale doma vadili zelo malo, saj so v povprečju učenci 1. letnika vadili manj kot eno uro na teden, učenci 2. letnika pa sploh nič. Skupno gledano so učenci obeh letnikov skupaj vadili manj kot 20 minut na teden.

Na podlagi tega sklepamo, da bi aplikacija morala ponujati večjo raznolikost iger, predvsem tudi bolj zahtevne vaje, saj so se intervali pri večini učencev izkazali za lahek izziv.

Učencem smo prav tako zastavili vprašanje, ali bi uporabljali igro za vajo intervalov v obliki spletne oz. mobilne aplikacije, če bi jim bila le-ta na voljo. Skoraj tri četrtine učencev so na vprašanje odgovorile pritrdilno, medtem ko je preostanek učencev ostal neopredeljen.

Ta informacija potrjuje našo predpostavko, da obstaja potreba po interaktivni izobraževalni platformi za učenje glasbene teorije.

5.2.4 Primerjava rezultatov testne in kontrolne skupine

Učenci testnih in kontrolnih skupin so na koncu raziskave rešili preverjanje znanja intervalov na konvecionalni način (s pomočjo klavirja in z zapisom not na papir). Učenci testne skupine 1. letnika so v povprečju dosegli 69,8%, medtem ko so učenci kontrolne skupine 1. letnika dosegli zgolj 60,6%. Opazimo torej skoraj 10% razliko med skupinama. Pri učencih 2. letnika je ta razlika bistveno manjša (okoli 1%), saj so učenci iz testne skupine v povprečju dosegli 73,4%, učenci iz kontrolne skupine pa 72,2%.

Na podlagi rezultatov končnega preverjanja smo še izvedli Mann-Whitneyjev U-test, s katerim smo želeli preveriti, če je porazdelitev podatkov različna za testne in kontrolne skupine. V primeru, da bi bila statistična značilnost oz. pomembnost dovolj majhna ($<0,05$), bi lahko zavrnilo ničelno hipotezo. Slednjo opredelimo kot trditev, da v splošnem uporabniki naše aplikacije, ki igrajo igro intervalov dosegajo boljše rezultate kot tisti, ki naše aplikacije ne uporabljajo.

Analiza je pri učencih 1. letnika pokazala statistično značilnost 0,177, kar

pomeni, da ničelne hipoteze ne moremo zavriniti, saj se porazdelitev podatkov po skupinah ne razlikuje dovolj. Pri učencih 2. letnika dobimo statistično značilnost 0,896, kar nakazuje na dejstvo, da so podatki po skupinah skoraj popolnoma enako porazdeljeni in da ponovno ne moremo zavriniti ničelne hipoteze. Analiza skupnih rezultatov za oba letnika nam pričakovano ponudi enak rezultat, torej nezmožnost zavrnitve ničelne hipoteze, saj je v tem primeru statistična značilnost enaka 0,418.

Ker je statistična značilnost v vseh primerih večja od 0,05 se moramo vzdržati vsakršnega sklepanja iz našega vzorca na populacijo in lahko rezultate interpretiramo zgolj na nivoju vzorca. Torej lahko sklepamo le, da so učenci testnih skupin v povprečju dosegli boljše rezultate, kot učenci kontrolnih skupin zaradi uporabe naše aplikacije oz. igranja igre intervalov v okviru naše platforme, ne moremo pa trditi, da bi se to v splošnem pokazalo pri vseh potencialnih uporabnikih naše platforme.

5.2.5 Dodatni rezultati, pridobljeni pri uporabi aplikacije

V aplikaciji smo merili številne statistike, s katerimi smo poskušali dobiti sliko o tem, kako uporabniki uporabljajo platformo in kako se znajdejo pri igranju igre intervalov. Mednje spadajo:

- beleženje prijav uporabnikov, s čimer smo želeli izvedeti, kako pogosto uporabniki uporabljajo aplikacijo,
- beleženje doseženih točk, nivojev in značk, kar nam je dalo informacijo o vplivih poigritve na motivacijo uporabnikov za uporabo platforme,
- beleženje statistik igre intervalov, kjer smo preko rezultatov merjenja števila dodanih oz. odvzetih not, števila ponovnih predvajanj zaporedja tonov, števila odgovorov in časa, ki so ga uporabniki porabili za odgovor na vprašanje, skušali izboljšati pravila igre in prilagoditi uporabniški vmesnik.

Pri analizi števila prijav učencev v aplikacijo smo ugotovili, da učenci aplikacije izven šolskega okvira oz. izven okvira raziskave skorajda niso uporabljali, saj je povprečno število prijav v aplikacijo na učenca znašalo zgolj 1,84. Zgolj določeni posamezniki, ki so v splošnem kazali več zanimanja za učenje in tudi pri reševanju nalog med testiranjem dosegali boljše rezultate, so aplikacijo uporabljali tudi v prostem času. Ta podatek prav tako kaže na dejstvo, da je bila v splošnem motivacija uporabnikov za udejstvovanje na platformi zelo nizka.

Iz zabeleženih statistik, ki smo jih zbirali med igranjem igre intervalov, lahko razberemo, da so se učenci z igranjem izboljšali oz. so postali bolj vešč igranja igre. Pri tem je pomembno poudariti, da smo podatke analizirali posebej za učence 1. in 2. letnika, saj je bila za skupini težavnost vprašanj in posledično tudi vrednost statistik različna.

Na Sliki 5.8a opazimo, da se je povprečni čas, ki so ga učenci potrebovali za odgovor na vprašanje iz 1. na 2. igro pri 1. letnikih malenkost zmanjšal, pri 2. letnikih pa se je več kot prepolovil.

Podobno opazimo za statistike, ki merijo povprečno število dodanih not na vprašanje (Slika 5.8b), povprečno število izbrisanih not na vprašanje (Slika 5.8c) in povprečno število ponovnih predvajanj zaporedja tonov (Slika 5.8d). Pri vseh omenjenih kategorijah opazimo drastični upad povprečnih vrednosti predvsem pri 2. letniku.

Pri statistiki, kjer smo merili povprečje odgovorov, kjer je učencem v zastavljenem časovnem okviru in z omejitvijo število možnih odgovorov, na koncu uspelo pravilno odgovoriti (Slika 5.8e), opazimo, da so učenci obeh letnikov iz 1. na drugo igro napredovali do te mere, da so na vse odgovore odgovorili pravilno.

Splošno izboljšanje uspeha učencev v opisanih kategorijah pa je povzeto v statistiki, ki meri povprečno število točk, ki so jih učenci zbrali pri določeni igri (Slika 5.8f). Pri izračunu točk tako upoštevamo čas, število dodanih in odvzetih not, dejstvo ter upoštevamo dejstvo, ali je učenec na odgovor na koncu odgovoril pravilno ali napačno. Obenem upoštevamo tudi težavnost

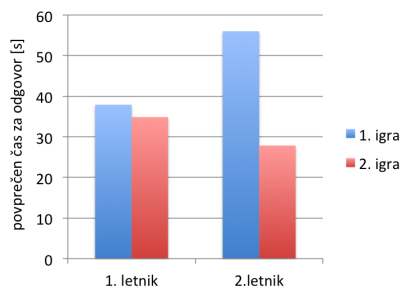
igre, zato na sliki opazimo, da so učenci 2. letnikov v povprečju zbrali več točk, kot učenci 1. letnikov.

Ti rezultati govorijo v prid naši implementaciji uporabniškega vmesnika igre intervalov, saj so učenci relativno hitro osvojili potrebna znanja za reševanje nalog.

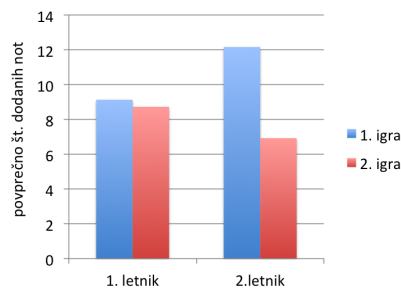
Pri učencih, ki so aplikacijo uporabljali tudi izven šolskega (testnega) okvira, smo prav tako opazili trend splošnega izboljšanja v statističnih kategorijah, kjer opravljamo meritve. Na grafu 5.9 opazimo, da se je za izbrano skupino 5 učencev povprečen čas, ki so ga potrebovali za odgovor na vprašanje v igri intervalov, s časom zmanjševal. Funkcija ni padajoča (saj pri padanju niha), vendar kljub temu zaznamo zmerno padanje povprečnega časa ob naraščanju števila odigranih iger. Po drugi strani lahko na grafu 5.10 vidimo trend naraščanja števila točk, ki so jih isti učenci zbrali pri igranju igre. Prav tako kot v primeru prejšnje funkcije opazimo, da funkcija niha in zato ni naraščajoča, vendar zaznamo zmerno naraščanje števila zbranih točk ob naraščanju števila odigranih iger. Na podlagi teh rezultatov lahko sklepamo, da so visoki motivirani učenci hitro osvojili večšine, ki so jih potrebovali za uspešno reševanje nalog.

V okviru beleženja statistik v aplikaciji smo zbirali podatke o tem, kako uspešni so bili učenci pri pridobivanju oz. doseganju poigritvenih elementov, kot so značke in nivoji. Na Sliki 5.11a je razvidno, da je bila največkrat dosežena značka “Igra s 50% točnostjo”, sledita pa ji znački “Igra končana v 25 minutah” in “Igra brez napake”. Na podlagi tega lahko sklepamo, da je večina učencev hitro in uspešno reševala vprašanja v igri intervalov, ni pa kontinuirano uporabljala platforme, saj značk, kot sta “Dokončana igra 7 dni zapored” in “Prijava 7 dni zapored” ni dosegel nobeden izmed učencev.

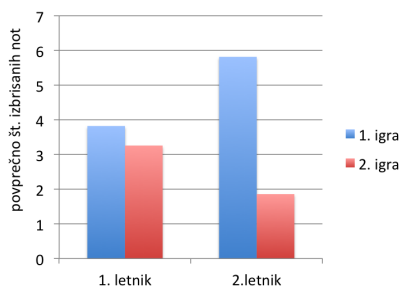
Slika 5.11b pa nam prikazuje, katere nivoje so učenci dosegli. Opazimo, da je večina učencev dosegla nivo 2 do 4, nekaj pa se jih je na platformi udejstvovalo več kot drugi in so tako dosegli tudi višje nivoje od 5 do 7. Sklepamo lahko torej, da so nivoji, kot element poigritve, pozitivno vplivali na približno četrtino učencev, ki so uporabljali aplikacijo.



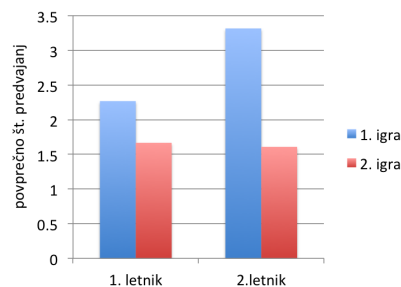
(a) Povprečen čas, ki so ga učenci potrebovali za odgovor na vprašanje.



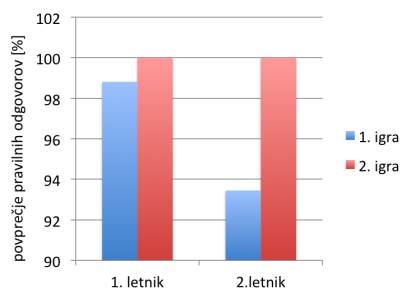
(b) Povprečno število dodanih not pri posameznem vprašanju.



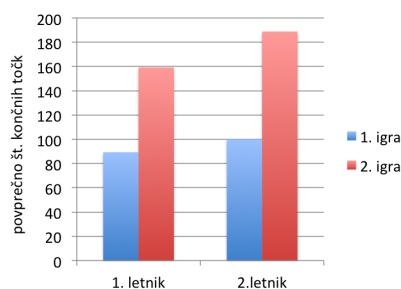
(c) Povprečno število izbranih not pri posameznem vprašanju.



(d) Povprečno število ponovnih predvajanj pri posameznem vprašanju.

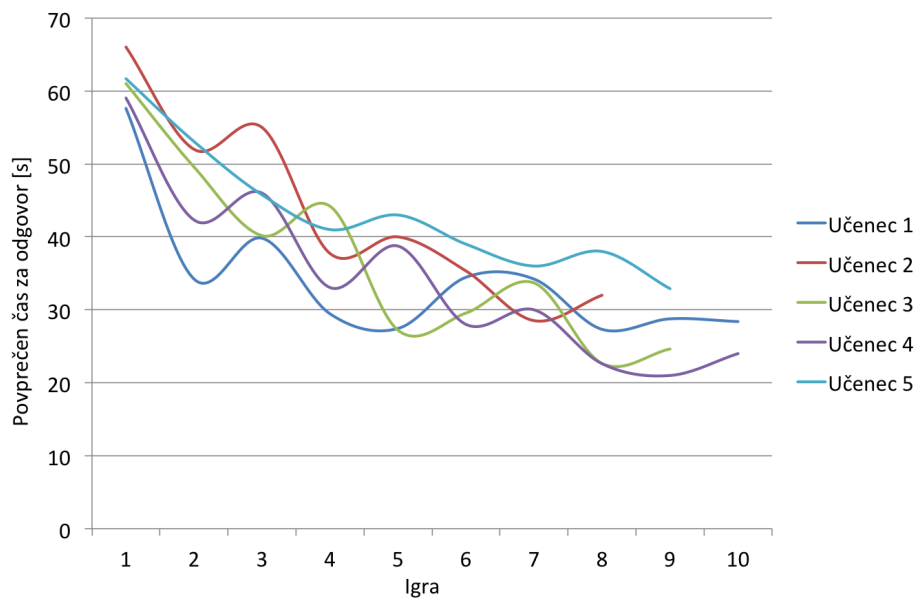


(e) Povprečen uspeh pri odgovoru na posamezno vprašanje.

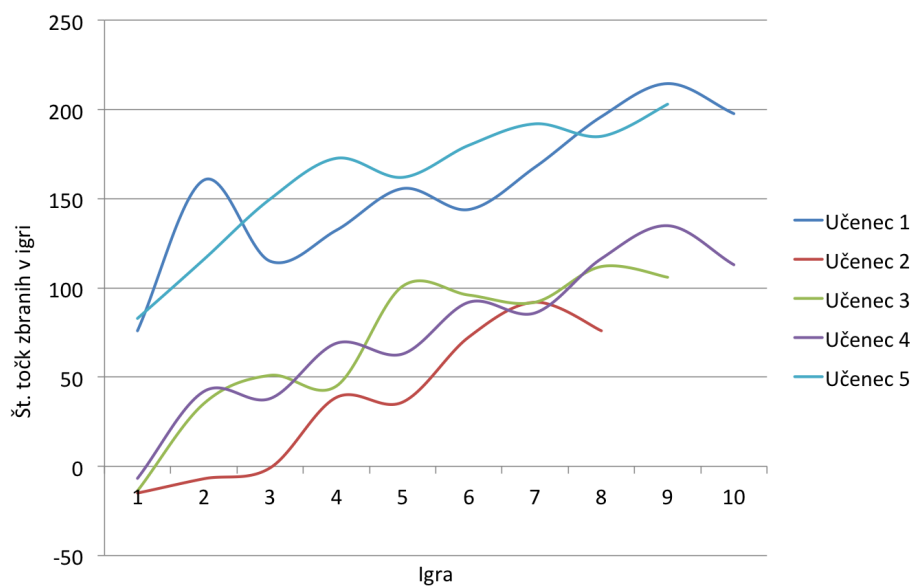


(f) Povprečno število zbranih točk za posamezno igro.

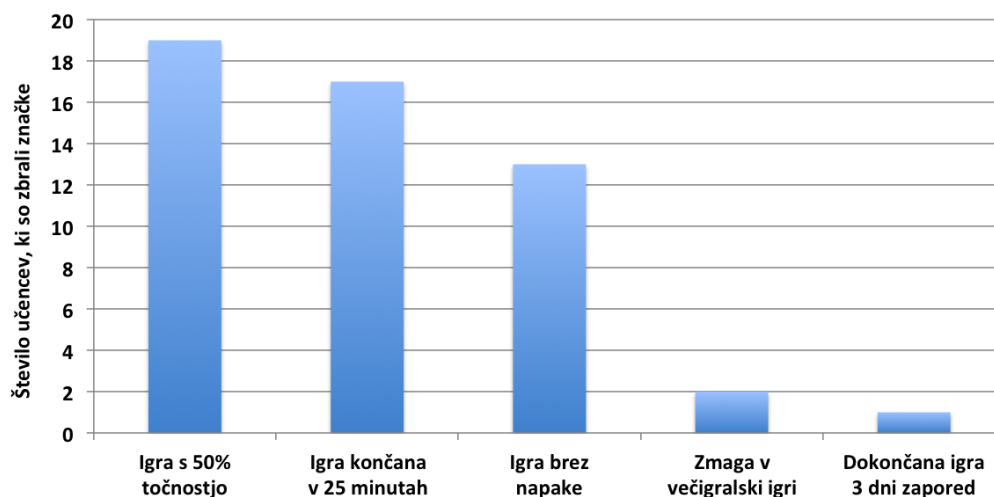
Slika 5.8: Primerjava 1. in 2. igre učencev obeh letnikov.



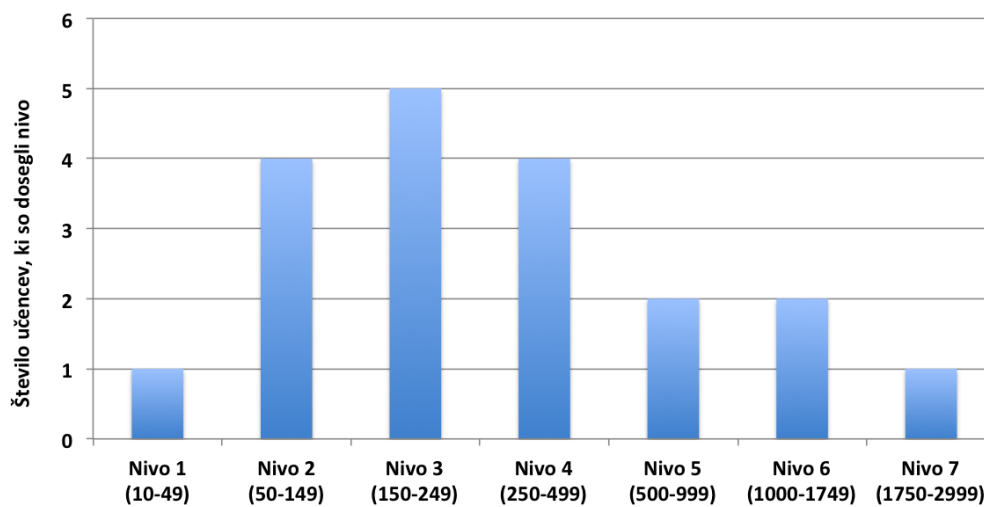
Slika 5.9: Povprečen čas za odgovor v odvisnosti od števila odigranih iger.



Slika 5.10: Število zbranih točk v odvisnosti od števila odigranih iger.



(a) Število učencev, ki so zbrali določeno vrsto značk v okviru poigritve aplikacije.



(b) Število učencev, ki so dosegli določen nivo v okviru poigritve aplikacije.

Slika 5.11: Statistika pridobivanja oz. doseganja poigritvenih elementov.

Poglavje 6

Sklepne ugotovitve

V okviru izdelave magistrske naloge smo dosegli cilj implementacije mobilne izobraževalne aplikacije za učenje glasbene teorije. Prav tako smo zadostili zastavljenim zahtevam po odprtokodnosti rešitve, enostavni nadgradljivosti z novimi tipi iger, poigritvi platforme in posebitvi uporabniške izkušnje. S pomočjo aplikacije smo uspešno izvedli testiranje z dijaki 1. in 2. letnika Srednje glasbene šole Konservatorija za glasbo in balet v Ljubljani. Z ovrednotenjem rezultatov testiranja smo odgovorili na raziskovalno vprašanje: “Kako poigritev vpliva na proces poučevanja in učenja glasbene teorije?”.

Analiza rezultatov A/B testiranja kaže na to, da lahko govorimo o pozitivnih učinkih poigritve na podlagi vzorca, ki smo ga uporabili pri testiranju, rezultatov pa ne moremo posplošiti na celotno populacijo. Učenci testnih skupin so sicer dosegli boljše rezultate pri končnem preverjanju znanja intervalov, kot učenci kontrolnih skupin, vendar ne moremo trditi, da se bodo pozitivni učinki poigritve preslikali na nove uporabnike naše platforme.

Glavni pomanjkljivosti naše študije sta dejstvi, da rezultati temeljijo na relativno majhnem vzorcu (33 učencev) in da je bilo testiranje izvedeno v kratkem časovnem obdobju ob koncu šolskega leta (približno v enem mesecu), ko so bile ocene že pridobljene in je tempo pouka nekoliko padel.

Na podlagi analize podatkov, ki smo jih zbirali med uporabo aplikacije, lahko sklepamo, da so določeni učenci, ki so pokazali željo in višjo motivacijo

za uporabo aplikacije in igranje iger, skozi čas postajali bolj večči igranja igre intervalov. Pri njih namreč opazimo dva trenda, in sicer, da je čas, ki so ga potrebovali za odgovor na posamezno vprašanje, padal s številom odigranih iger in da je število točk, ki so jih zbrali v igri, naraščalo s številom odigranih iger. Hkrati so ti učenci bolj pogosto pridobivali oz. dosegali elemente poigravitve, kot so značke in nivoji, obenem pa so se aktivno potegovali za višjo uvrstitev na lestvici.

Učencem je bila zelo všeč uporabniška izkušnja, ki so jo doživeli pri uporabi aplikacije, kar so dodatno podkrepili s trditvijo, da bi aplikacijo priporočili tudi prijateljem in znancem. Uporabnikom je torej všeč ideja aplikacije, ki jim ponuja drugačen način učenja glasbene teorije, kot so ga vajeni iz šolskih klopi.

Po drugi strani pa iz rezultatov anketnega vprašalnika lahko razberemo, da je ena od glavnih pomanjkljivosti naše rešitve precejšno število različnih iger, kar je negativno vplivalo na uporabo aplikacije oz. motiviranost uporabnikov za igranje iger. Med anketiranci obstaja prevladujoče mnenje, da bi z nadaljnjim razvojem aplikacije in njenim nadgrajevanjem v obliki novih iger dosegli višjo stopnjo angažiranosti uporabnikov za uporabo platforme.

Pomembno je poudariti, da so tipi uporabnikov aplikacije lahko zelo različni, kar močno vpliva na njihovo motivacijo. Glede na dejstvo, da se pri sami implementaciji nismo posebej osredotočili na različne tipe uporabnikov in da so bili izzivi, s katerimi so se soočili, za vse enaki, je to morda tudi eden od razlogov za nizko stopnjo angažiranosti uporabnikov za uporabo aplikacije. Določeni uporabniki se namreč raje socializirajo, določeni raje tekmujejo, nekateri radi zbirajo značke, spet drugi pa radi raziskujejo. V sklopu prototipa bi bilo težko nasloviti potrebe vseh teh različnih skupin uporabnikov, vsekakor pa to predstavlja eno od potencialnih izboljšav aplikacije, ki bi lahko imela pozitiven vpliv na motivacijo uporabnikov.

V vsebinskem smislu bi v prihodnje veljalo aplikacijo razširiti izven intervalnega treninga na posebno področje melodije. To bi dosegli z uvedbo tonov različnih notnih vrednosti (npr. doba in njena prva in druga podde-

litev), kar bi muzikalno obogatilo zvočno vsebino aplikacije, ter z različnimi nivoji kromatike, kot so npr. čista diatonika, mutacija med durom in molom, stabilne alteracije, labilne alteracije itd.

Izpostavili bi tudi dejstvo, da so bile informacije, ki smo jih pridobili z izvedbo osebnih intervjujev zelo koristne. Na njihovi podlagi smo izboljšali izgled uporabniškega vmesnika in uporabniško izkušnjo, obenem pa odpravili veliko tehničnih težav. Sklepamo, da bi bilo v prihodnje potrebno opraviti še več intervjujev z uporabniki in tako pridobiti dodatne koristne informacije za iterativen razvoj aplikacije.

Potencialno izboljšavo aplikacije prav tako vidimo v razvijanju modula za učitelje, preko katerega bi slednji lahko nadzorovali in vodili učni proces učencev. S tem bi aplikacija lahko postala pomemben izobraževalni pripomoček za poučevanje glasbene teorije.

Sklepamo, da je poigritev zelo težak proces, močno odvisen od domene v kateri je implementiran, za maksimizacijo uspešnosti poigritve pa moramo biti pozorni na raznolike faktorje, kot so različni tipi uporabnikov, njihova notranja in zunanja motivacija, ter vključenost oz. angažiranost. V magistrskem delu smo določene kriterije poigritve zadovoljivo naslovili, saj smo dosegli cilj implementacije izobraževalne platforme obogatene z elementi poigritve, a obenem na podlagi testiranj in ovrednotenja identificirali še mnogo izzivov, ki jih bo potrebno nasloviti v prihodnosti.

Dodatek A

Primeri programske kode

A.1 Jezik stilskih predlog Sass (sintaksa SCSS) z uporabo notacije BEM

```
1 <style lang="scss" scoped>
2 @import '../..//sass/variables/index';
3
4 .leaderboard__table-row {
5     height : $row-height;
6     padding : 10px 5px;
7 }
8
9 .leaderboard__table-row--selected {
10     background-color : $light-green;
11     color : $dark-gray;
12 }
13
14 .leaderboard__table-row--disabled {
15     background-color : $silver;
16     color : $gray;
17 }
18 </style>
```

A.2 Vzpostavljajanje ogrodja knjižnice MIDI.js

```
1 MIDI.loadPlugin({
2   soundfontUrl: '/soundfonts/',
3   instruments: this.instruments.map(instrument =>
4     instrument.soundfont),
5   targetFormat: 'mp3',
6   onSuccess: () => {
7     for (let name of this.instruments) {
8       let instrument = this.instruments[name]
9       MIDI.setVolume(instrument.channel, this.
10         VOLUME)
11       MIDI.programChange(instrument.channel,
12         MIDI.GM.byName[instrument.soundfont].
13         number)
14     }
15   }
16 })
```

A.3 Logika zagona in ustavljanja predvajanja zaporedja tonov

```
1 play() {
2   if (!this.playing) {
3     this.playing = true
4     this.noteIds = []
5
6     for (let i = 0; i < this.notes.length; i++) {
7       let id = setTimeout(() => {
8         MIDI.noteOn(this.instrument.channel,
9           notes[i].pitch, this.NOTE_VELOCITY
10            , this.NOTE_DELAY)
11         setTimeout(() => {
```

```
10         MIDI.noteOff(this.instrument.  
11             channel, notes[i].pitch, this.  
12             NOTE_DELAY)  
13  
14         this.noteIds.push(id)  
15     }  
16  
17     this.playingId = setTimeout(() => {  
18         this.playing = false  
19     }, this.notes.length * this.NOTE_DELAY)  
20 }  
21 },  
22 stop() {  
23     if (this.playing) {  
24         for (const id of this.noteIds) {  
25             clearTimeout(id)  
26         }  
27         clearTimeout(this.playingId)  
28         this.playing = false  
29     }  
30 }
```

A.4 Inicializacija in zagon odštevalnika časa knjižnice Timer Progress

```
1 const timer = new TimerProgress({  
2     'container' : document.getElementById('timer'),  
3     'width-container' : 300,  
4     'height-container' : 200,  
5     'stroke-width' : 20,
```

```
6   'color-container' : '#5e5e5e',
7   'color-circle'   : '#c4c3c3',
8   'color-path'     : '#2c2c2c',
9   'color-text'     : '#2c2c2c',
10  'color-alert'    : '#dc143c',
11  'font-family'    : 'GothamRounded-Light'
12 }, 2)
13 timer.run(7000, 3000)
```

A.5 Algoritem za generiranje vprašanj pri igri intervalov

```
1 function generateIntervalsQuestion($range, $maxNotes,
2   $chapter, $nChapters) {
3   $pitches = ['A#3', 'B3', 'C4', 'C#4', 'D4', 'D#4',
4     'E4', 'F4', 'F#4', 'G4', 'G#4', 'A4', 'A#4',
5     'B4', 'C5', 'C#5'];
6
7   $pitchOccurrences = [];
8   foreach ($pitches as $pitch) {
9     $pitchOccurrences[$pitch] = 0;
10  }
11
12  $sample = [];
13
14  // randomly generate first note and add it to the
15  // sample
16  $pitch = $pitches[array_rand($pitches)];
17  $sample[] = $pitch;
18
19  // generate consecutive notes each one based on
20  // the previous note
21  $nNotes = $maxNotes - $nChapters + $chapter;
```

```
17     for ($i = 1; $i < $nNotes; $i++) {
18         $pitchIndex = array_search($pitch, $pitches);
19
20         // define possible range (steps allowed to
           the top / bottom)
21         $topRange = count($pitches) - $pitchIndex -
           1;
22         $bottomRange = $pitchIndex;
23         $rangeSum = $topRange + $bottomRange;
24
25         // choose direction of the interval by using
           weighted random
26         $direction = weightedRandom(['down' =>
           $bottomRange / $rangeSum, 'up' =>
           $topRange / $rangeSum]);
27
28         // potentially limit the range of the
           interval with the predefined range
29         $range = $direction == 'down' ? min($range,
           $bottomRange) : min($range, $topRange);
30
31         // randomly choose the actual range
32         $nSemitones = rand(0, $range);
33         $intervalIndex = $direction == 'down' ? (
           $pitchIndex - $nSemitones) : ($pitchIndex
           + $nSemitones);
34
35         // based on the range find the pitch
36         $pitch = $pitches[$intervalIndex];
37
38         // check if the pitch satisfies the defined
           constraints
39         if ($pitchOccurrences[$pitch] === 2 || (($i
```

```
    === 1 || $i === $nNotes - 1) && $sample[$i
    - 1] === $pitch)) {
40     $i--;
41     continue;
42 }
43
44 // add the pitch to the sample
45 $sample[] = $pitch;
46
47 // keep track of how many times each of the
    pitches already occurred in the sample
48 $pitchOccurrences[$pitch]++;
49 }
50
51 return $sample;
52 }
```

Dodatek B

Vprašalniki

B.1 Intervju

1. Profil (aktivno spraševanje)
 - Koliko si star/a?
 - Ali igraš računalniške igre?
 - Uporabljaš mobilne aplikacije?
 - Ali si kdaj uporabljal/a mobilne aplikacije za učenje? S področja glasbe? Če da, katere in kako pogosto?
2. Uspešnost opravljanja nalog (pasivno opazovanje)
 - Prijavi se v aplikacijo.
 - Preglej svoj profil in preveri, če so vsi podatki pravilni.
 - Pojdi na domačo stran.
 - Poglej na katerem mestu se nahajaš na lestvici.
 - Pojdi v nastavitve in zamenjaj inštrument.
 - V nastavitvah prilagodi pavzo med predvajanimi toni.
 - Odjavi se iz aplikacije.

- Registriraj se s svojim imenom in e-mail naslovom, potrdi ga, nato se ponovno vpiši v aplikacijo.
 - Postavi mobilno napravo v ležeč položaj in se poskušaj orientirati po aplikaciji.
3. Učinkovitost igre (pasivno opazovanje)
- Izberi igro “Intervali” > “1 igravec” in pred začetkom igranja preberi navodila.
 - Reši vprašanja (bolje, če hitreje).
4. Zadovoljstvo (aktivno spraševanje)
- Se ti je zdela igra zahtevna?
 - Se ti je zdela igra zabavna?
 - Bi igro uporabljal/a še naprej? Zakaj?
 - Se ti igra zdi boljši način učenja intervalov kot konvencionalni način? Koliko sicer vadiš intervale?
 - Kaj ti ni bilo všeč pri igri?
 - Kaj ti je bil najbolj všeč pri igri?
 - Ali so bila navodila dobro napisana? Si jih razumel/a že preden si začel igrati? Kaj si spoznal/a šele med igro in bi želel/a, da bi bilo napisano že v navodilih?
 - Imaš še kakšen predlog oz. pripombo?

B.2 Vprašalnik o profilu

Vprašanja so odprtega tipa.

1. Koliko si star/a?
2. Ali igraš računalniške igre / igre na konzolah? Če da, kako pogosto?

3. Ali uporabljaš mobilne aplikacije? Če da, katere in kako pogosto?
4. Ali si kdaj uporabljal mobilne aplikacije za učenje s področja glasbe? Če da, katere in kako pogosto?
5. Kateri mobilni telefon uporabljaš (znamka, model) in kateri spletni brskalnik na mobilnem telefonu (npr. Chrome / Safari / Firefox / drugo)?

B.3 Vprašalnik o aplikaciji

Vprašanja od 1. do 24. so zaprtega tipa, možni odgovori pa so “povsem se strinjam”, “strinjam se”, “niti se strinjam, niti se ne strinjam”, “ne strinjam se” in “sploh se ne strinjam”. Vprašnji 25. in 26. sta odprtega tipa.

1. Uporaba aplikacije ni zahtevna.
2. Navodila za igranje so ustrezna (so mi bila v pomoč).
3. Navodil za uporabo sploh ne bi potreboval/a.
4. Aplikacija je odzivna (deluje hitro).
5. Za reševanje nalog imam premalo časa.
6. Za učenje uporabe aplikacije potrebujem preveč časa.
7. Aplikacija me spodbuja k učenju intervalov.
8. Med uporabo aplikacije se na trenutke počutim frustriranega.
9. Igro bi za učenje priporočil/a prijatelju/znancu.
10. Če bi imela aplikacija na voljo več različnih iger, bi me bolj pritegnila k sodelovanju.
11. Aplikacija ima na voljo dovolj različnih tipov nalog.

12. Igra se mi ne zdi zahtevna.
13. Igra se mi zdi zabavna.
14. Aplikacijo bom uporabljal/a še naprej.
15. Všeč mi je izgled (dizajn) uporabniškega vmesnika.
16. Razdelitev vsebine po podstraneh ni smiselna.
17. Uporabniški vmesnik igre (klavir, notno črtovje in ostali gumbi) je uporaben.
18. Imam težave z orientacijo in navigacijo po aplikaciji.
19. Hitro najdem informacije, ki sem jih iščem.
20. Prilagoditev nastavitvev igre in profila (šola, razred) mi predstavlja težave.
21. Želja po dosegu/izpolnitvi značk (badges) in nivojev (levels) me spodbuja k reševanju nalog.
22. Všeč mi je sistem točkovanja (lestvica) in tekmovanje z drugimi.
23. Ni mi všeč dejstvo, da vsi lahko vidijo točke na lestvici.
24. Če bi bil na voljo večigralski način igre, bi bolj pogosto uporabljal aplikacijo.
25. Koliko ste uporabljali igro doma za vajo? (v urah na teden)
26. Imaš še kakšno dodatno zamisel, pripombo?

B.4 Vprašalnik za kontrolno skupino

Vprašanja so odprtega tipa.

1. Koliko ur na teden ste vadili intervale na običajen način (klavir / druga aplikacija / ...)?
2. Ali bi uporabili igro za vajo intervalov (v obliki spletne / mobile aplikacije), če bi bila na voljo?

Literatura

- [1] Spletišče Pojmovnik, Dosegljivo: http://muzikologija.ff.uni-lj.si/ptg/index_slo.html, Dostopano: 7. 1. 2018.
- [2] Spletišče Pojmovnik teorije glasbe, Dosegljivo: <http://pojmovnik.fri.uni-lj.si/>, Dostopano: 7. 1. 2018.
- [3] P. Šavli, A. Bervar, A. Čopi, Posodobljeni učni načrt. Solfeggio, Ministrstvo za izobraževanje, znanost, kulturo in šport, 2013.
- [4] L. Hakulinen, T. Auvinen, A. Korhonen, The effect of achievement badges on students' behavior: An empirical study in a university-level computer science course 10.
- [5] K. M. Kapp, The gamification of learning and instruction: game-based methods and strategies for training and education, John Wiley & Sons, 2012.
- [6] T. Shea, Gamification: Using Gaming Technology for Achieving Goals, Digital and information literacy, Rosen Pub., 2013.
- [7] C. Wagner, Digital gamification in private music education, *Antistasis* 7 (1) (2017) 115–122.
- [8] M. Ibanez, A. Di-Serio, C. Delgado-Kloos, Gamification for engaging computer science students in learning activities: A case study, *IEEE Transactions on Learning Technologies* 7 (3) (2014) 291–301.

-
- [9] F. Layth Khaleel, N. Ashaari, T. S. M. T. W. Tengku Wook, A. Ismail, Gamification elements for learning applications 6, no. 6, 2016.
- [10] C. Muntean, Raising engagement in e-learning through gamification.
- [11] A. Dominguez, J. S. de Navarrete, L. de Marcos, L. Fernandez-Sanz, C. Pages, J.-J. Martinez-Herraiz, Gamifying learning experiences: Practical implications and outcomes, *Computers & Education* 63 (2013) 380–392.
- [12] R. Al Azawi, M. Al Bulshi, F. Al Farsi, Educational gamification vs. game based learning: Comparative study 7 (2016) 131–136.
- [13] C. Perrotta, G. Featherstone, H. Aston, E. Houghton, Game-based learning: Latest evidence and future directions, NFER Research Programme: Innovation in Education. Slough: NFER.
- [14] M. Maria, M. Fradinho Duarte de Oliveira, G. Waddell, Game-based learning of musical instruments: A review and recommendations, 10th European Conference on Games Based Learning, University of the West of Scotland, Paisley, Scotland.
- [15] G. Denis, P. Jouvelot, Motivation-driven educational game design: Applying best practices to music education, in: *Proceedings of the 2005 ACM SIGCHI International Conference on Advances in Computer Entertainment Technology, ACE '05*, ACM, New York, NY, USA, 2005, pp. 462–465.
- [16] J. Hamari, J. Koivisto, H. Sarsa, Does gamification work? – a literature review of empirical studies on gamification, in: *2014 47th Hawaii International Conference on System Sciences*, 2014, pp. 3025–3034.
- [17] J. Hamari, Do badges increase user activity? a field experiment on the effects of gamification, *Computers in Human Behavior* 71 (2017) 469 – 478.

-
- [18] R. Van Eck, Digital game-based learning: Still restless, after all these years 50 (2015) 12–28.
- [19] R. Van Eck, Digital game-based learning: It's not just the digital natives who are restless, *EDUCAUSE review* 41 (2) (2006) 16.
- [20] E. Sanchez, S. Young, C. Jouneau-Sion, Classcraft: from gamification to ludicization of classroom management, *Education and Information Technologies* 22 (2) (2017) 497–513.
- [21] T. Connolly, M. Stansfield, T. Hainey, An application of games-based learning within software engineering 38 (2007) 416–428.
- [22] S. C. Ng, A. K. Lui, W. S. Lo, *An Interactive Mobile Application for Learning Music Effectively*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2013, pp. 148–157.
- [23] D. Gebremichael, An evaluation of gamification to assess students' learning on their understanding of first year computer science programming module, A dissertation submitted in partial fulfilment of the requirements of Dublin Institute of Technology for the degree of M.Sc. in Computing, Advanced Software Development, Advanced Software Development (2016).
- [24] Aplikacija My Piano Assistant, Dosegljivo: <https://play.google.com/store/apps/details?id=com.alexpp.mpa&hl=en>, Dostopano: 15. 9. 2018.
- [25] Aplikacija Teoria, Dosegljivo: <http://teoria.com>, Dostopano: 15. 9. 2018.
- [26] Aplikacija musictheory.net, Dosegljivo: <https://www.musictheory.net/>, Dostopano: 15. 9. 2018.
- [27] Aplikacija Musition, Dosegljivo: <https://www.risingsoftware.com/musition/>, Dostopano: 15. 9. 2018.

-
- [28] Aplikacija Perfect Ear, Dosegljivo: <https://play.google.com/store/apps/details?id=com.evilduck.musiciankit&hl=en>, Dostopano: 15. 9. 2018.
- [29] Aplikacija iRealPro, Dosegljivo: <https://irealpro.com/>, Dostopano: 15. 9. 2018.
- [30] Aplikacija MyEarTraining, Dosegljivo: <https://www.myeartraining.net/>, Dostopano: 15. 9. 2018.
- [31] Aplikacija Sibelius, Dosegljivo: <https://www.avid.com/sibelius>, Dostopano: 15. 9. 2018.
- [32] Programska knjižnica Laravel, Dosegljivo: <https://laravel.com/>, Dostopano: 15. 9. 2018.
- [33] Scalable Vector Graphics, Dosegljivo: https://en.wikipedia.org/wiki/Scalable_Vector_Graphics, Dostopano: 15. 9. 2018.
- [34] Programska knjižnica Sass, Dosegljivo: <http://sass-lang.com/>, Dostopano: 15. 9. 2018.
- [35] Block Element Modifier, Dosegljivo: <http://getbem.com/introduction/>, Dostopano: 15. 9. 2018.
- [36] Programska knjižnica Vue.js, Dosegljivo: <https://vuejs.org/>, Dostopano: 15. 9. 2018.
- [37] Programska knjižnica Vue Router, Dosegljivo: <https://router.vuejs.org/>, Dostopano: 15. 9. 2018.
- [38] Programska knjižnica Vuex, Dosegljivo: <https://vuex.vuejs.org/>, Dostopano: 15. 9. 2018.
- [39] Programska knjižnica Timer Progress, Dosegljivo: https://github.com/volkmaster/timer_progress, Dostopano: 15. 9. 2018.

-
- [40] Protokol MIDI, Dosegljivo: <https://en.wikipedia.org/wiki/MIDI>, Dostopano: 15. 9. 2018.
- [41] MIDI.js, Dosegljivo: <https://github.com/mudcube/MIDI.js/>, Dostopano: 15. 9. 2018.
- [42] Programska knjižnica axios, Dosegljivo: <https://github.com/axios/axios>, Dostopano: 15. 9. 2018.
- [43] Programska knjižnica Lodash, Dosegljivo: <https://lodash.com/>, Dostopano: 15. 9. 2018.
- [44] Programska knjižnica Moment.js, Dosegljivo: <https://momentjs.com/>, Dostopano: 15. 9. 2018.
- [45] Programska knjižnica node-sass, Dosegljivo: <https://github.com/sass/node-sass>, Dostopano: 15. 9. 2018.
- [46] R. Kohavi, R. Longbotham, Online Controlled Experiments and A/B Testing, Springer US, Boston, MA, 2016, pp. 1–8.