

SVEUČILIŠTE U ZAGREBU
FILOZOFSKI FAKULTET
ODSJEK ZA INFORMACIJSKE ZNANOSTI
Akademska godina 2015/16

Bruno Tot

**PROTOTIP ZVUČNOG REPRODUKTORA ZA
PROVOĐENJE PERCEPCIJSKIH EKSPERIMENATA**

Završni rad

Mentor: Dr. sc. Nikolaj Lazić, izv. prof.

Zagreb, 2016.

Sadržaj

1	UVOD.....	2
2	POKUSI – OBJAŠNJENJE I PROBLEMATIKA.....	3
3	RJEŠENJE	6
3.1	PLANIRANJE, KRITERIJI I DIZAJN	6
3.2	TEHNIČKI PODACI	10
3.3	PREDNOSTI RJEŠENJA.....	11
3.4	BUDUĆNOST	11
3.	LITERATURA.....	13

Popis slika

Slika 1:	Prikaz grafičkog sučelja prototipa	7
----------	--	---

1 UVOD

Istraživanja su glavni pokretači napretka u znanosti. Ona potvrđuju ili pobijaju teze, stvaraju nove i pružaju uvid u svijet oko nas kao i nas same. Jedno od najvažnijih područja istraživanja vezano uz ljude i živa bića općenito jest percepcija. Više je područja percepcije, ali fokus ovog rada je na onoj slušnoj. Zahvaljujući tehnološkom napretku danas su dostupni mnogobrojni alati za provođenje raznih ispitivanja. Međutim, većina tih alata je upotrebljiva gotovo isključivo uz naprednije tehničko znanje i razumijevanje tehnologije.

Ta činjenica obeshrabruje i demotivira prvenstveno studente, ali i sve one koji žele provoditi percepcijske eksperimente da se tim alatima i koriste, pa pribjegavaju jednostavnijim rješenjima koja su nažalost često neprimjerena ili nedovoljno prilagođena potrebama eksperimentatora. Fokus ovog rada je upravo rješenje tog problema – softver kojeg će svi jednostavno i intuitivno moći koristiti neovisno o prethodnom poznavanju tehnologije. Prototip izrađen za ovaj rad služi kao primjer i prijedlog nekih koncepata i rješenja, ali je i u svom trenutnom stanju spreman za korištenje. Trenutna inačica softvera, kao i sve daljnje njegove inačice će uvijek biti besplatne za korištenje i distribuciju. Svi dodaci korišteni za njegovu izradu su korišteni u skladu s njihovim licencama te je svaka komercijalizacija softvera koja nije odobrena od strane autora kršenje autorskih prava.

2 POKUSI – OBJAŠNJENJE I PROBLEMATIKA

Znanstveno istraživanje je dugotrajan i složen proces. Ovisno o području i metodologiji dijelovi tog procesa se razlikuju. Jedan od ključnih dijelova su pokusi. Pokus¹ je izazivanje određene pojave u svrhu njenog zapažanja, promatranja i proučavanja. Pokusi su teorija provedena u praksu. Njima se preispituju pretpostavke i znanja, ali i stječu nova. Znatno se razlikuju ovisno o njihovom cilju iako svi eksperimenti imaju neke zajedničke komponente:

- Hipotezu – pretpostavku na temelju koje je eksperiment osmišljen i koja određuje očekivane rezultate.
- Ispitanike – ono nad kim ili čim se vrši eksperiment, ono od koga ili čega se očekuje reakcija ili neka indikacija promatrane pojave.
- Podražaj/e – ono čime se nastoji izazvati željena pojava kod ispitanika, bio to zvuk, slika, pokret, predmet ili nešto drugo na što ispitanici trebaju reagirati.
- Okolinu – uvjeti u kojima se provodi pokus. Mogu biti primjerice laboratorijski, nadzirani ili nenadzirani.

Razlog i potreba za pokusima su očiti – jedini način da se nešto sa sigurnošću utvrdi jest da se i potvrdi u praksi.² Naravno, pritom se oslanja na prethodna saznanja i pretpostavke jer je nemoguće i besmisleno apsolutno svo prethodno stečeno znanje (koje nismo sami stekli već je „proslijeđeno“ obrazovanjem) preispitivati.

Osmisliti i provesti eksperiment nije jednostavno. Mnogobrojni su čimbenici koji utječu na rezultate te ih je potrebno što je više moguće nadzirati i kontrolirati. Ispitanici, na primjer, su iznimno važna komponenta (glede njihovih osobina); dob, spol, obrazovanje, jesu li i koliko upoznati s predmetom istraživanja ili znanstvenom disciplinom uz koju se ono veže su samo neke od karakteristika koje mogu utjecati na rezultate ispitivanja. Nadalje, uvjeti u kojima se odvija ispitivanje mogu (i najčešće imaju) izniman utjecaj na rezultate. Na primjer, ispitanici će zasigurno bolje čuti neki zvuk ako se nalaze u tihoj prostoriji (u laboratorijskim uvjetima) nego na hodniku fakulteta punog studenata koji razgovaraju ili idu nekamo.

¹ Wikipedia: Pokus, URL: <https://hr.wikipedia.org/wiki/Pokus> (24.09.2016.)

² Coleman, John: Introduction to Phonetics Experiments, URL: http://www.phon.ox.ac.uk/jcoleman/exp_intro.html (24.09.2016.)

Također, tu je važna i metoda, način na koji se provodi pokus; jesu li ispitanici u skupinama, kakva se tehnologija koristi i slično.

Na kraju, prikupljeni podaci se analiziraju i obrađuju te se na temelju dobivenih rezultata donose zaključci i objašnjenja o razlozima i načinu događanja neke pojave. Dakako, ponekad se ne može dobiti potpuno jasan odgovor – bilo radi prevelikog broja varijabli izvan kontrole ispitivača ili zbog neočekivanih pa i ponekad zbunjujućih rezultata. Pretpostavke i nove teorije nastale zbog toga mogu poslužiti kao temelj za buduće pokuse i istraživanja.

Kao student fonetike obvezan sam sudjelovati kao ispitanik u određenom broju eksperimenata. Većinu tih eksperimenata provode drugi studenti s viših godina i uglavnom se radi o percepcijskim eksperimentima. To su eksperimenti u kojima se ispitanicima puštaju određeni stimulusi (u ovom slučaju zvukovi) te se od ispitanika očekuje da ispune neke obrasce vezane uz podražaj. U slučaju fonetike to mogu biti snimke govora političara gdje se procjenjuju neke njihove osobine, različite kombinacije tonova i šumova kako bi se odredili efekti maskiranja („prikriivanja“ jednog zvuka drugim) ili vrlo kratki isječci riječi, slogova, čak i pojedinih glasova gdje se trajanje svakog stimulusa mjeri u milisekundama.

Alati kao što je Praat³ su primjereni za provođenje takvih eksperimenata uz uvjet da korisnik (ispitivač) jako dobro poznaje Praat i rad na računalu više od prosječnog korisnika. Međutim, neki od eksperimentatora jednostavno nemaju dovoljno tehničkog znanja ili bolje rečeno – vještine da bi se pravilno njima koristili, pogotovo kad je riječ o studentima koji tek stječu potrebno iskustvo i znanja. Dakako, u određenim slučajevima se radi i o tehnofobiji – strahu da će se sve urušiti pritiskom na ijedan gumb ili da će se neka nepopravljiva šteta nanijeti računalu ili stroju o kojem je riječ. U tim situacijama, ispitivači pribjegavaju korištenju raznoraznih improviziranih metoda kao što su korištenje uobičajenih medijskih reproduktora (Windows Media Player, VLC,...) koji svakako podržavaju širok spektar medijskih formata, ali nipošto nisu namijenjeni eksperimentima. Glavni problem je što ti programi nastoje što brže prikazati sadržaj korisniku da se ponekad može dogoditi da se preskoči nekoliko milisekundi početka ili da se sljedeći stimulus ne učita na vrijeme jer je trenutni vrlo kratak pa ga program ponovo reproducira i sl. To možda nije veliki problem kod svakodnevnog slušanja glazbe ili gledanja filmova, ali kada se radi o vrlo kratkim

³ Praat, URL: <http://www.fon.hum.uva.nl/praat/> (24.09.2016.)

stimulusima gdje je svaka milisekunda važna, sve što izostane može utjecati na percepciju pa tako i na rezultate istraživanja. Također, ako se radi u eksperimentima većeg opsega s puno kratkih isječaka može doći do zbrke s redoslijedom pa ispitanici mogu pogriješiti prilikom ispunjavanja upitnika ili odgovaranja misleći da je na redu sljedeći stimulus, ali ustvari program ponavlja prethodni jer se idući nije uspio učitati. Upravo tome sam i sam svjedočio sudjelovavši u jednom takvom eksperimentu. Eksperiment je bio vezan za percepciju utjecaja vokala na konsonante. Kao ispitanici smo dobili upitnik od nekoliko stranica na kojima smo morali označiti za svaki primjer koji smo vokal čuli. Radilo se o oko 1200 iznimno kratkih primjera (trajanja 100-200 milisekundi, a možda i manje) koji su bili reproducirani redom. Ispitivači su koristili neki od običnih reproduktora (najvjerojatnije Windows Media Player) te se dogodilo upravo prethodno opisano – program je reproducirao zvukove čim je stigao i ponavljao prethodne dok su se sljedeći učitali. To je uzrokovalo velike teškoće u praćenju svim ispitanicima zbog nejednakih kao i često premalih pauza nakon svakog primjera, a ispitivači nakon zaustavljanja reprodukcije nisu bili u mogućnosti odrediti koji je bio zadnji reproduciran primjer. U kasnijem razgovoru s mentorom sam saznao da Praat⁴ nudi i mogućnosti postavljanja eksperimenata za koje nisam znao jer nije bilo potrebe. Kasnije se ispostavilo da ispitivači nisu vični tehnologiji pa su se snašli na najjednostavniji način. Tako sam dobio ideju za alat koji bi bio svima pristupačan i jednostavan za korištenje i koji bi automatizirao postupak a ispitivačima dao potrebne povratne informacije.

⁴Praat, URL: <http://www.fon.hum.uva.nl/praat/> (24.09.2016.)

3 RJEŠENJE

3.1 PLANIRANJE, KRITERIJI I DIZAJN

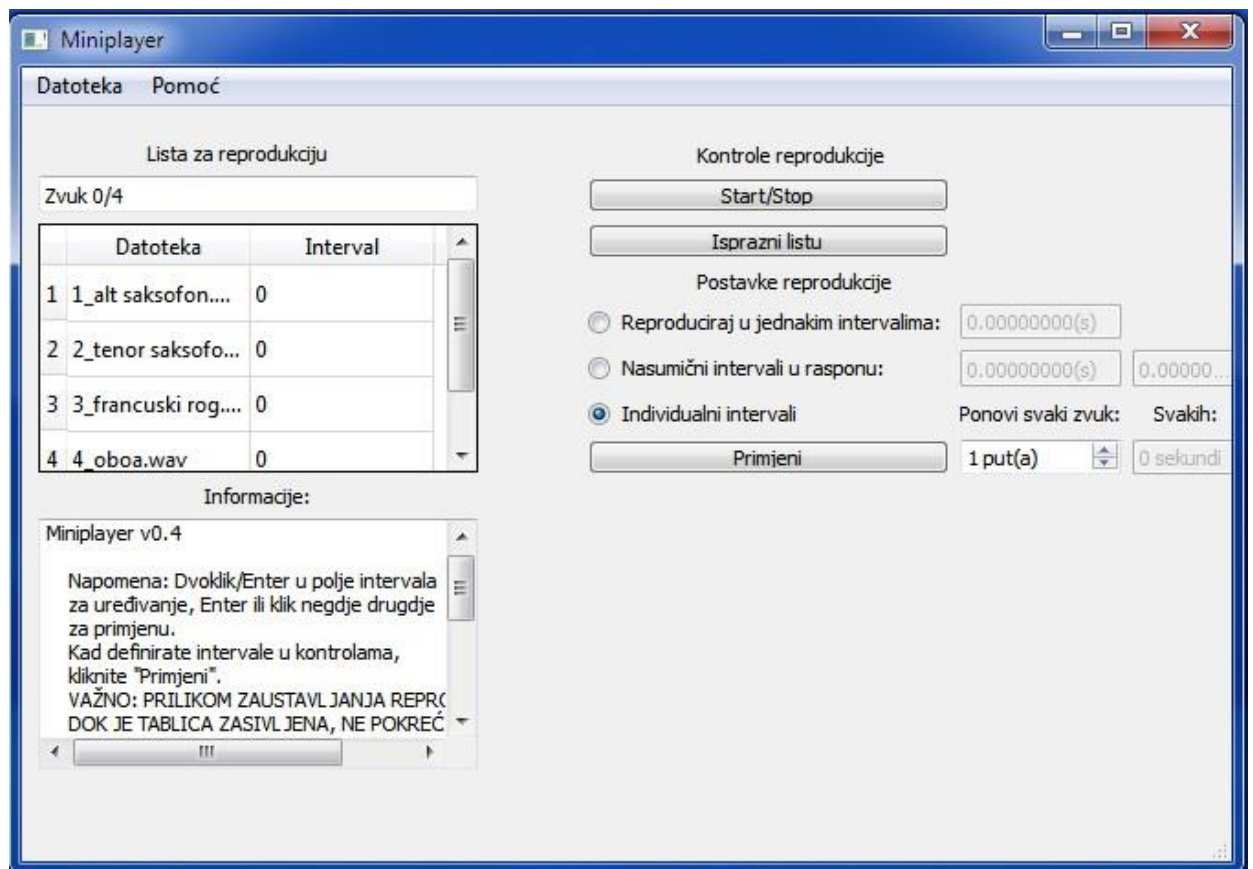
Prije početka izrade trebalo je razmotriti mogućnosti i odabrati optimalno rješenje. Program je trebao zadovoljiti određene kriterije:

1. Jednostavnost izrade – zbog ograničenog vremena i iskustva jezik i alati za izradu morali su biti što jednostavniji i intuitivniji, ali i pružiti dovoljnu razinu kontrole.
2. Kompatibilnost – program je morao biti kompatibilan sa što više platformi kako bi se što veći broj korisnika mogao služiti njime.
3. Jednostavnost sučelja – grafičko sučelje je moralo biti jasno i minimalističko kako bi se svi korisnici mogli lako snaći i efikasno koristiti program.
4. Veličina i jednostavnost instalacije – što manja aplikacija za jednostavniju distribuciju te što manji broj koraka potreban za postavljanje i pokretanje programa.

Što se samih alata i programskog jezika tiče, vremensko ograničenje i iskustvo su bili odlučujući faktori pa je za izradu odabran *Python 2.7.11*, a kao okruženje je korišten *Spyder* koji je zajedno s mnogim modulima došao u paketu znanom kao *Anaconda (2.4.1)* što je uvelike olakšalo konfiguraciju i instalaciju svih potrebnih komponenata. Za grafičko sučelje odabran je modul *PyQt4* zbog svoje podrške za velik broj platformi te izrazitu jednostavnost korištenja u odnosu na mogućnosti koje nudi. Važno je napomenuti da je *PyQt4* modul besplatan isključivo u nekomercijalne svrhe.

Za zvuk je nakon duljeg istraživanja i testiranja izabran *PyAudio*. Prvotno je bio korišten *winsound* modul koji je standardan modul Pythona, ali je utvrđeno da ne podržava uničivanje (*eng. threading*) poglavito zbog nemogućnosti komuniciranja (dojave stanja) kako bi se reprodukcija mogla nadzirati. Osim navedenih modula korišteni su još i standardni moduli *threading* – za uničivanje, *sys* – za komuniciranje aplikacije i sustava (kao što je zatvaranje aplikacije), *wave* – za čitanje .wav datoteka, *random* – trenutno za odabir intervala iako su moguće i druge implementacije u budućnosti te *time* za pauziranje reprodukcije. Može se primijetiti kako je većina korištenih modula standard što je uvelike olakšalo razvoj te zapečatilo Python kao prvi izbor u ovom projektu. Sučelje je izrađeno u QT Designeru kako bi se dodatno uštedjelo na vremenu te je prilikom svake promjene kod iznova bio generiran u Pythonu. Za izgradnju aplikacije tj. pretvorbu u .exe korišten je *PyInstaller 3.2*, a za dodatnu

kompresiju je korišten *UPX* što doduše utječe na brzinu pokretanja aplikacije. Nakon raznih kombinacija elemenata sučelja, odabran je konačan dizajn.



Slika 1: Prikaz grafičkog sučelja prototipa

Kao što se može vidjeti, svaka komponenta ima svoj naziv, a sve napomene i informacije su istaknute u posebnom prozoru. Neki elementi još nemaju svoju pravu namjenu pa služe samo kao primjer. To se prvenstveno odnosi na element „Informacije“ čija je prvotna namjena bila prikazivati podatke o selektiranim datotekama, ali se obavijesna funkcija pokazala korisnijom. Također, izbornik „Pomoć“ sadrži opciju „O aplikaciji“ koja samo obavještava korisnika da se radi o prototipu, ali se jednostavno može prenamijeniti ako se aplikacija nastavi razvijati.

Izbornik „Datoteka“ sadrži dvije opcije: „Otvori...“ i „Izlaz“. Sve su opcije samoobjašnjive; „Otvori...“ otvara novi prozor u kojem se mogu selektirati i učitati sve podržane datoteke (u ovom slučaju isključivo .wav). Prozor za odabir datoteka je već definiran unutar *PyQt4* modula te njegov izgled varira ovisno o platformi. Opcija „Izlaz“ zatvara aplikaciju te prethodno zaustavlja reprodukciju kao i dobro poznati crveni „X“ u

gornjem desnom uglu prozora. Obje opcije imaju i kratice (redom: Ctrl+O kao standard te Ctrl+Q za „Izlaz“).

Nadalje, sučelje se može podijeliti na dva glavna dijela – prikaz reprodukcije i upravljanje reprodukcijom. U prikazu reprodukcije nalazi se naslovljena lista za reprodukciju koja prikazuje sve učitane datoteke te njihove intervale, sve označeno brojevima s lijeve strane zbog lakšeg snalaženja s većim brojem datoteka. Trenutno se nazivi datoteka ne mogu uređivati niti se redoslijed može mijenjati, ali se prilikom otvaranja datoteka može odrediti jer on ovisi o redoslijedu selektiranja. Lista je zapravo tablica s dva stupca – prvi je već spomenut i sadrži naziv datoteke a drugi interval. Interval je pauza (u sekundama) između dva zvuka. Svi intervali se mogu uređivati i ručno upisivati u obliku cijelog ili decimalnog broja. Ako uneseni interval nije ispravan broj, program će prilikom reprodukcije kada dođe do tog mjesta upozoriti korisnika i prekinuti reprodukciju. Ako je iz nekog razloga unesen negativan broj, program će uzeti apsolutnu vrijednost pa neće doći do pogreške. To vrijedi za sva polja u kojima se unose intervali. Treba napomenuti da se jedino posljednji red ne može uopće uređivati i njegov interval je uvijek 0 jer je to posljednji zvuk te se reprodukcija može zaustaviti odmah po završetku zvuka. Lista također sadrži i brojač koji je rješenje prethodno navedenog problema snalaženja i nemogućnosti određivanja posljednje reproduciranog zvuka u slučaju zaustavljanja prije kraja. Budući da se lista generira ovisno o broju datoteka, prilikom samog otvaranja aplikacije brojač će prikazivati 0/0, a prije prve reprodukcije će prikazivati 0/X gdje je X broj datoteka u listi. Nakon zaustavljanja reprodukcije, brojač se neće vratiti na 0 već će i dalje prikazivati redni broj posljednjeg zvuka. To ne znači da je zvuk reproduciran do kraja već da se barem počeo reproducirati; tijekom pauze brojač ostaje na prethodnom broju. Prilikom pokretanja reprodukcije, brojač se uvijek vraća na trenutni zvuk. Ispod liste se nalazi element „Informacije“ koji je prethodno već opisan.

Na desnoj strani prozora nalazi se „upravljač reprodukcijom“. Podijeljen je u dva dijela – kontrole reprodukcije i postavke reprodukcije. Prvi dio za sada sadrži samo dva gumba koja su također samoobjašnjiva. Gumb „Start/Stop“ pokreće i zaustavlja reprodukciju, a njegova kratica je razmaknica jer je to i inače standard u većini reproduktora za tu funkciju. Reprodukcija uvijek ide od početka redom po listi. Gumb „Isprazni listu“ briše sve datoteke sa svih lista (za sada su dvije) te vraća aplikaciju u stanje jednako onom pri prvom pokretanju zbog čega je odabrana kratica Ctrl+Z za tu funkciju jer je slična spasonosnoj funkciji „undo“ u drugim aplikacijama. Ako se funkcija aktivira dok reprodukcija još traje, reprodukcija će biti zaustavljena kako bi se izbjegle pogreške.

Drugi dio se odnosi na postavke. Valja primijetiti da se postavke kao i ostale funkcije otključavaju samo u slučaju da lista nije prazna (iz očitih razloga) te se također zaključavaju zajedno s listom za vrijeme reprodukcije.

Prije svega, ponuđena su tri načina određivanja intervala:

1. Jednaki za sve
2. Nasumični u zadanom rasponu
3. Ručno određeni intervali

Posljednja opcija je ujedno i zadana pri prvom pokretanju. Odabirom svake od opcija otvaraju se različite mogućnosti. Prva opcija će za svaku datoteku odrediti interval koji se upiše u polje s njene desne strane koje se otključava samo kada je ta opcija odabrana (kao i sva ostala polja vezana uz druge opcije). Drugi izbor ima dva polja koja određuju najmanju i najveću vrijednost raspona. Posebna metoda je definirana kako bi omogućila odabir nasumičnog decimalnog broja što je inače bilo moguće samo za cijele brojeve. Ova opcija će čak i cijeli broj pretvoriti u decimalni (u tom slučaju imat će jednu decimalu), a ukoliko je unesen decimalan broj u bilo koje od polja, uzet će broj s najviše decimalnih mjesta te generirati (zapravo odabrati) nasumičan decimalan broj s toliko decimalnih mjesta. Drugim riječima: ako je prvi broj 1, a drugi 2.3456789, funkcija će prvi broj pretvoriti u 1.0 nakon čega će usporediti broj decimalnih mjesta oba broja. Ispostavit će se da 2.3456789 ima više decimalnih mjesta pa će onda i rezultat biti broj između 1.0000000 i 2.3456789 uključujući i oba krajnja broja. Zbog intuitivnosti su podržani i brojevi koji umjesto decimalne točke imaju zarez (1,0). Te brojeve će program ispraviti i staviti decimalnu točku. Ta provjera je dodana zbog toga što netko još možda ima naviku pisanja zareza. Bez obzira na izbor načina određenja intervala uvijek je moguće ručno promijeniti bilo koju stavku osim posljednje koja je uvijek 0. Treba primijetiti i da pri otvaranju datoteka (bez iznimke) su svi intervali postavljeni na 0. To ne znači da neće biti nikakve pauze između zvukova, jer to u trenutnoj implementaciji nije moguće, već da će se idući zvuk reproducirati čim to bude moguće. To vrijedi za sve postavke intervala.

Dalje slijede postavke ponavljanja. Ponavljanja se odnose na svaki zvuk, ne na listu u cjelini. Dakle, određuje se koliko će puta svaki zvuk biti ponavljen prije nego se prijede na sljedeći. Minimalan broj je 1 a maksimalan 10 (jer se ne čini potrebnim više). Logično je zaključiti da će se svaki zvuk reproducirati barem jedanput, a ne da će nakon prve

reprodukcije uslijediti još jedna. Moguće je i odrediti interval između ponavljanja. On je neovisan o ostalim intervalima i koristi se samo ako je broj ponavljanja veći od 1 što se očituje i otključavanjem polja za njegovo određenje čim se poveća broj ponavljanja. U ovom slučaju, ako je unesena pogrešna vrijednost moguća su dva ishoda:

1. U slučaju da je primjerice upisan tekst umjesto brojke, program će javiti da interval nije ispravan broj i prekinuti reprodukciju.
2. U slučaju da je polje prazno ili decimalan broj sa zarezom, program će za interval uzeti 0 i nastaviti s reprodukcijom.

Posljednji, ali nikako najmanje važan element je gumb „Primjeni“ koji još jednom izvršava sve provjere postavki te primjenjuje zadane intervale na listu. Ne treba naglašavati da se prije reprodukcije trebaju sve postavke „Primijeniti“ jer je to vidljivo u programu. Iznimka su ručno uneseni intervali koji se odmah primjenjuju. Može se izvući pravilo: Ako nije u listi, ne priznaje se.

Trenutno je zbog tehničkih ograničenja dodano upozorenje da se reprodukciju ne pokreće sve dok je tablica zaključana (zasivljena) jer to može uzrokovati probleme, ali o tome više u nastavku.

3.2 TEHNIČKI PODACI

Aplikacija uz kompresiju zauzima 10.8 MB što je solidno poboljšanje u odnosu na 14.2 MB bez kompresije (što manje, to bolje) i prilikom pokretanja bez učitavanja datoteka zauzima oko 50.7 MB RAM-a. Količina zauzete memorije ovisi o dva faktora:

1. O veličini datoteka na redu za reprodukciju
2. O broju ponavljanja

Program je konstruiran tako da pri početku reprodukcije učita prvih X datoteka Y puta. X je broj datoteka (trenutno je određeno 3), Y je broj ponavljanja (od 1 do 10). Dakle, ako su odabrana 2 ponavljanja, program će svaku datoteku učitati 2 puta u memoriju. Već je prije rečeno da su ustvari dvije liste aktivne, ali su u reprodukciji tri. Treća lista je ona u koju se zapravo učitavaju zvukovi za reprodukciju; ona je privremena i stvara se i briše po potrebi. Razlog zbog kojeg dolazi do problema u reprodukciji (i s time i mogućeg rušenja programa) je metoda *sleep()* iz modula *time* koja zaustavlja operacije na određeno vrijeme. Problem je u

tome što proces ili u ovom slučaju nit koja je tako zaustavljena nije u mogućnosti ništa drugo raditi dok god ne prođe zadano vrijeme. Ono što bi se moralo dogoditi je to da dok nit čeka, istovremeno provjerava oznake za početak odnosno kraj te u slučaju da se označi kraj (za vrijeme reprodukcije se ponovo aktivira gumb „Start/Stop“) nit se prekine. No, ta provjera je u ovom slučaju moguća jedino kada prođe zadano vrijeme čekanja tj. zadani interval. Za vrijeme pisanja ovog rada se osmislilo nekoliko mogućih rješenja tog problema, ali nijedno još nije implementirano ni testirano. Također, treba napomenuti da je trenutno aplikacija zbog vremenskog i tehničkog ograničenja dostupna samo za 64 bitne Windows operativne sustave.

3.3 PREDNOSTI RJEŠENJA

Glavna prednost glavna prednost ovog rješenja je jednostavnost. Svi ga mogu koristiti bez ikakvih prethodnih koraka – poput Praat⁵, samo se preuzme i pokrene. Činjenica da je lokaliziran na hrvatski olakšava korištenje i onima koji ne barataju dovoljno dobro stranim jezicima (prvenstveno engleskim s obzirom da je većina alata na tom jeziku). Nadalje, činjenica da zauzima nešto manje od 11 MB znači da ga je lako preuzeti čak i na sporijim internetskim vezama, a s obzirom da zauzima nešto manje od 51 MB RAM-a znači da se čak i stariji uređaji te oni slabijih karakteristika mogu nositi s njim. Također, modul korišten za izradu sučelja podržava velik broj platformi, uključujući i mobilne (Android, iOS,...) što znači da su potrebne minimalne prilagodbe za izradu inačica za njih. Program je moguće lako prevesti i na druge jezike. Isto tako, sučelje bi trebalo biti u potpunosti kompatibilno sa čitačima ekrana pa ga mogu koristiti i osobe oštećenog vida (iako se to još mora potvrditi u praksi). Budući da je moguće odrediti pauze između zvukova, broj ponavlja i pauze između ponavljanja, gubi se potreba za „ručnim“ spajanjem zvukova. Sve što je potrebno je pripremiti primjere i učitati ih u program. Ukratko: maksimalna kompatibilnost – minimalna zahtjevnost.

3.4 BUDUĆNOST

Prototip je jednostavan, ali funkcionalan i ima mnogo potencijala za daljnji razvoj. Ako se aplikacija pokaže korisnom, zasigurno će se i dalje proširivati. Prvi korak bi svakako bio optimiziranje koda koji u sadašnjem obliku ima svojih nedostataka. Softver bi morao biti što

⁵ Praat, URL: <http://www.fon.hum.uva.nl/praat/> (24.09.2016.)

brži i zauzimati što manje resursa (što je donekle potpomognuto generatorima). Također je moguće proširenje mogućnosti programa. Prvenstveno se može dodati podrška za reprodukciju drugih formata zvuka (MP3, FLAC, OGG,...) što u ovoj izvedbi nije namjerno učinjeno. Isto tako, može se proširiti opseg programa; jedna od ideja je mogućnost pohranjivanja svih datoteka korištenih u eksperimentu kao i povezanih podataka u jednu datoteku koja bi se kasnije mogla učitati natrag u program ili dijeliti mrežom što bi omogućilo lakše provođenje potpuno jednakih eksperimenata od strane različitih ispitivača kao i jednostavno izvođenje u više navrata bez potrebe za ponovnim postavljanjem. Ta datoteka bi mogla primjerice biti .zip arhiva s dodanom .json datotekom što bi omogućilo i kompresiju za uštedu prostora i jednostavnu pohranu nužnih podataka. Osim toga mogla bi se napraviti poslužiteljsko-klijentska inačica programa u kojoj bi ispitivač s jednog računala upravljao reprodukcijom, a na drugim računalima bi ispitanici mogli ispunjavati upitnike koji bi automatski slali podatke poslužitelju (ispitivačevoj strani aplikacije) koja bi te podatke u konačnici obradila i prikazala ispitivaču.

Mogućnosti su mnogobrojne, ali sve ovisi o potrebama i zahtjevima korisnika. Još jednom treba napomenuti da se radi o aplikaciji prvenstveno za netehničare. Za sve naprednije potrebe od onih koje ova aplikacija nudi, svakako se preporučuju drugi alati kao što je Praat⁶ koji je djelomice i služio kao inspiracija za izradu ovog alata i baš kao Praat, ako ova aplikacija bude korištena, uvijek će biti potpuno besplatna.

⁶ Praat, URL: <http://www.fon.hum.uva.nl/praat/> (24.09.2016.)

3. LITERATURA

Wikipedia: Pokus

<https://hr.wikipedia.org/wiki/Pokus>

(24.09.2016.)

Coleman, John: Introduction to Phonetics Experiments

http://www.phon.ox.ac.uk/jcoleman/exp_intro.html

(24.09.2016.)

Praat

<http://www.fon.hum.uva.nl/praat/>

(24.09.2016.)