



Šolski center Celje

Srednja šola za kemijo, elektrotehniko in računalništvo

ŠPORTNO GLASBENA APLIKACIJA

Heat beat

Raziskovalna naloga

Avtorji:

Dominik Zagrušovcem r-4-a

Nik Deželak r-3-a

Mentor:

Prof. Jaka Koren

Mestna občina Celje, Mladi za Celje

Celje, 2023

1 Zahvala

Zahvaljujeva se mentorju prof. Jaka Korenu za pomoč in vodstvo pri opravljanju raziskovalne naloge. Posebna zahvala gre vsem anketirancem, ki so nama odgovorili na anketo.

2 Povzetek

V tej raziskovalni nalogi je obravnavana izdelava aplikacije "Heat Beat", katere cilj je ugotoviti vpliv glasbe na človekovo stanje pri visoki fizični aktivnosti, ter morebitne nepričakovane učinke ki se pri tem lahko pojavijo.

Najina ideja je bila narediti aplikacijo, ki sinhronizira glasbo s hojo ali tekom. Pri tem sva naletela na veliko težav, večinoma pri izdelavi aplikacije. Ob koncu sva jih uspešno razrešila in odgovorila na hipoteze, ki sva si jih zastavila na to temo.

3 Abstract

This research paper goes through the creation and use of the "Heat Beat" application, the aim of which is to determine the impact of music on the human condition during high physical activity, as well as possible side effects that may occur.

Our goal was to build a working application that allows you to synchronize music with walking or running. While doing so we ran into some problems, mainly with the development of the application. At the end we managed to solve them and answer the hypothesis we came up with regarding this topic.

4 Kazalo vsebine

| | |
|------------------------|---|
| Uvod..... | 7 |
| Končni cilj..... | 7 |
| Metode dela..... | 7 |
| Hipoteze in cliji..... | 7 |

| | |
|------------------------------------------------------------|-------------------------------------|
| Planiranje aplikacije..... | 8 |
| Planiranje števca korakov | 9 |
| 1. ideja – merjenje višine | 9 |
| 2. ideja – merjenje hitrosti | 9 |
| 3. ideja – merjenje razlike hitrosti..... | 9 |
| 4. ideja – vgrajen števec | 9 |
| 5. ideja – Uporaba GPS-a..... | 10 |
| Prednosti in slabosti uporabe gps-a..... | 12 |
| Planiranje metode za določanje bpm pesmi..... | 12 |
| 1. Način – zaznavanje onset-a..... | 13 |
| Onset | 13 |
| Kako deluje zaznavanje onset-a..... | Error! Bookmark not defined. |
| Praktičen primer onset-a | 14 |
| Kako se zabeležijo podatki v mp3 datoteki | 14 |
| Prednosti in slabosti uporabe zaznavnika onset-a..... | 15 |
| 2. Način | 15 |
| Prednosti in slabosti samo vnosa bpm | 15 |
| Možne ideje..... | 16 |
| 1. Način – uporaba perifernih naprav | 16 |
| 2. Način – uporaba vgrajene kamere..... | 16 |
| Povezovanje programov med sabo | 17 |
| Praktični del | 18 |
| Android studio | 18 |
| Izbira med java in kotlinom..... | 18 |
| Izdelava bralca glasbe | 20 |
| Problemi bralca glasbe | 20 |
| Reševanje problema | 20 |
| Problemi pri implementiranju media store/media player..... | 20 |
| Izdelava pmb določevalnika | 21 |
| Problem shranjevanje podatkov | 22 |
| Implementacija števca korakov..... | 23 |
| 1. Ideja..... | 23 |

| | |
|----------------------------|-------------------------------------|
| 2. Ideja..... | 23 |
| Rešitev problemov | 23 |
| 1. Problem | 24 |
| 2. Problem | 28 |
| Povezovanje programov..... | 29 |
| Analiza | 30 |
| Lastne ugotovitve..... | 33 |
| Odgovori na hipoteze | 34 |
| Zaključek..... | Error! Bookmark not defined. |
| Viri slik | Error! Bookmark not defined. |
| Izjava..... | 38 |

5 Kazalo slik

| | |
|---------------------------------------------------------------------------------------------|-------------------------------------|
| Slika 1: prostorske koordinate v treh dimenzijah | 9 |
| Slika 2: sprememba hitrosti relativna z časom | 9 |
| Slika 3: prikaz aplikacije (strava), ki uporablja gps za zabeleževanje in prikaz poti | 10 |
| Slika 4: razmerje dolžine koraka v primerjavi z višino in spolom človeka | 11 |
| Slika 5: primer gps-a ki ne deluje pravilno | 12 |
| Slika 6: prikaz onset-a..... | Error! Bookmark not defined. |
| Slika 7: prikaz spremembe zvoka iz analognega v digitalnega | 14 |
| Slika 8: prikaz primerjave tihe in glasne glasbe v digitalni obliki..... | 15 |
| Slika 9: android pametna ura | 16 |
| Slika 10: prikaz aplikacije z obkroženim gumbom "SET" | Error! Bookmark not defined. |
| Slika 11: prikaz aplikacije pro pritisku na gumb "SET" | Error! Bookmark not defined. |
| Slika 12: tabela z vrednostimi časa, korakov ter povprečne hitrosti..... | 24 |
| Slika 13: grafa z prikazano odmerjeno hitrostjo ter povprečno hitrostjo..... | 24 |
| Slika 14: merjenje 3 povprečnih hitrosti | 25 |
| Slika 15: merjenje 4 povprečnih hitrosti | 26 |
| Slika 16: merjenje 4 povprečnih hitrosti | 26 |
| Slika 17: merjenje vrednosti trenutnega časa z celotno količino korakov | 27 |
| Slika 18: tabela vrednosti trenutnega časa in celotno količino korakov | 27 |
| Slika 19: merjenje vrednosti trenutnega časa z celotno količino korakov | 28 |
| Slika 20: prikaz aplikacije z poudarkom na slide bar | 29 |

6 Kazalo grafov

| | |
|----------------------------------------------------|----|
| Graf 1: poslušanje glasbe v prostem času | 30 |
| Graf 2: fizična aktivnost v tednu..... | 30 |
| Graf 3: tip fizične aktivnosti | 31 |
| Graf 4: vpliv glasbe na rezultate aktivnosti | 31 |
| Graf 5: uporaba telefona za poslušanje glasbe..... | 32 |

7 Kratice

Bmp: ang.: beats per minute – enota ki se v glasbi uporablja kot mera tempa v pesmi.

Gps: ang.: global positioning system – satelitski radijsko navigacijski sistem za določanje pozicije.

IDE: ang.: integrated development environment – integrirano razvojno okolje za izdelavo programov in aplikacij za določene sisteme.

Sql: ang.: structured query language - standardiziran programski jezik, ki se uporablja za upravljanje relacijskih baz podatkov.

8 Uvod

Čeprav se je človek v zadnjem stoletju prilagodil in navadil na omejujoče življenje, ki posamezniku ne dopušča popolne svobode v vsakdanjem življenju, to ne pomeni da smo se s tem sprijaznili. Ta se med drugim lahko najde tudi v telovadnici, v plavanju, plezanju, smučanju, drsanju, teku, ali pohodu. Vse te aktivnosti pa imajo nekaj skupnega. To je, da se vse aktivnosti izvajajo v nekem ritmu.

Človeka spremlja na primer ritem okolja ali pa ritem bitja srca.

Namen najine naloge je ugotoviti vpliv ritma na osebno počutje, raven moči in vzdržljivosti.

8.1 Končni cilj

Najini končni cilj je bil izdelava aplikacije, s katero se ne eni strani uporabniku meri njegov ritem pri hoji/teku, na drugi strani pa se izmeri ritem mp3 skladb. Uporabnikovem ritmu hoje/teka nato aplikacija prilagodi seznam predvajanih skladb.

8.2 Metode dela

Pri raziskovanju sva uporabila dve metodi dela:

- Metoda ankete, kjer sva udeležence spraševala o njihovi tedenski vadbi, če pri njej poslušajo glasbo in če mislijo da ima to vpliv na njihove rezultate
- Metodo raziskave kjer sva:
 - raziskala internet na to temo
 - zgradila aplikacijo, ki sva jo dala testiranim osebam, ki so soglašali s testom in nama vrnila rezultate.

8.3 Hipoteze in cilji

Zadala sva si hipoteze ki bi najbolj natančno opisale najino ugotovitev do katere sva hotela priti. Odgovore na hipoteze pa sva nato pridobila iz anketiranja udeležencev in preizkušanja aplikacije.

- 1 HIPOTEZA: poslušanje glasbe izboljša rezultate fizične aktivnosti
 - 1.1 HIPOTEZA: poslušanje glasbe izboljša možgansko delovanje pri fizični aktivnosti
 - 1.2 HIPOTEZA: poslušanje glasbe izboljša telesno vzdržljivost pri fizični aktivnosti
- 2 HIPOTEZA: vsaj 50% odstotkov ljudi pri fizični aktivnosti posluša glasbo
 - 2.1 HIPOTEZA: vsaj 20% poslušalcev misli da jim glasba pri aktivnosti pomaga/prinaša boljše rezultate
- 3 HIPOTEZA: Čeprav je bila to najina prva izkušnja z izdelavo aplikacije, sva menila da jo bova uspešno naredila.

CILJ: Uspešna izdelava športno-glasbene aplikacije.

9 Načrtovanje aplikacije

Aplikacija, ki sva si jo zamislila, bo imela dva osnovna dela že obstoječih aplikacij

- aplikacija za predvajanje glasbe
- aplikacija za merjenje hitrosti hoje/korakov.

Namenila jo bova praktični uporabi. Zato sva pri načrtovanju dala prednost namenu pred estetiko. Pozornost sva namenila enostavnosti in lahkemu razumevanju aplikacije. Tako jo bodo lahko razumeli in z lahkoto uporabljajo vsi uporabniki.

Aplikacija bo sestavljena iz treh glavnih delov:

- program za branje in štetje korakov,
- program za določanje bpm-ma glasbe,
- program ki bi ju povezal v delujočo aplikacijo.

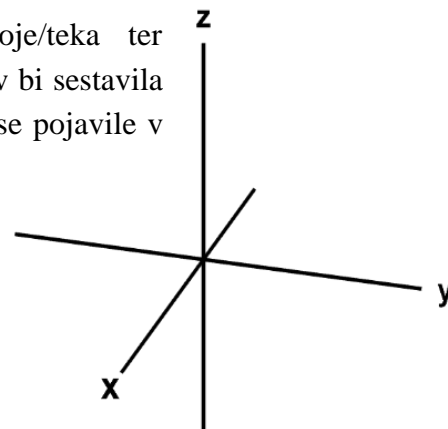
10 Planiranje števca korakov

10.1 1. ideja – merjenje višine

Najina prva ideja je bila uporaba senzorja za gibanje, in iz tega sestaviti algoritem ki bi prepoznal kdaj uporabnik hodi/teče, ali izvaja druge fizične aktivnosti. To bi naredila tako, da bi zabeležila kakršen koli premik ki bi se zgodil v smeri Z koordinate.

Nato bi pridobila začetne podatke z beleženjem najine hoje/teka ter opazovala pri kakšnih vrednostih se je zgodil korak. Iz teh podatkov bi sestavila algoritem ki bi jih preštel. Istočasno pa bi pazila na težave, ki bi se pojavile v primerih:

- štetja koraka kadar ga uporabnik ni naredil
- ne štetja koraka kadar ga je uporabnik naredil
- počasno delovanje aplikacije
- delovanje aplikacije za osebe drugačne višine, sloga hoje...

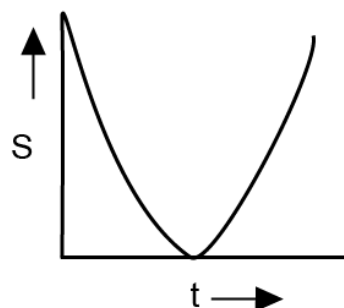


Slika 1: prostorske koordinate v treh dimenzijah

10.2 2. ideja – merjenje hitrosti

Druga ideja bi tudi uporabila senzor za gibanje. Tokrat pa bi se zabeležila hitrost naprave.

Najino mnenje je bilo da ko človek stopi, se v premiku navzgor zgodi velika takojšna razlika hitrosti, ko pa začne padati pa hitrost pada počasneje. To obliko hitrosti bi nato lahko primerjala s hitrostjo ki jo sprejema senzor, ter pri tem ugotovila če je to korak ali ne.



Slika 2: sprememba hitrosti relativna z časom

10.3 3. ideja – merjenje razlike hitrosti

Tretja ideja je bila bolj enostavna. Za to idejo ni potrebnih zahtevnih funkcij, ampak samo prepoznavanje ritma. Delovala bi tako, da bi uporabnik kliknil kdaj hoče začeti aktivnost. Nato bi zabeleževala vrednosti senzorja, ki bi jih po določenem času prebrala in primerjala. Korak bi štela vsakič, ko bi se zgodil prestop med hitrostjo, kot prikazano na sliki 2. ($s^- \Rightarrow s^+$).

Na žalost ta rešitev prinaša veli problem. Skoraj nemogoče je narediti števec, ki bi upošteval samo določene prestopne (zaradi občutljivosti senzorja in človeške napake) ter obenem predvideti sloge hoje različnih oseb.

10.4 4. ideja – vgrajen števec

Na srečo sva imela še četrto opcijo. Ta je bila že vgrajeni števec korak ki ga imajo prenosne naprave.

Z uporabo že vgrajenega števca bi z lahkoto brala korake, nato pa jih delila s preteklim časom da bi dobila bpm.

10.5 5. ideja – Uporaba GPS-a

Na misel nama je tudi prišlo, da bi uporabila vgrajen gps ter zabeležila pot ki jo je uporabnik opravil od izbrane točke. Ta način beleženja poti je viden v veliko aplikacijah in je zelo enostaven ter interaktiven. Ima možnosti prikazati zemljevid in uporabnikovo pot ki jo je naredil.

Po izdelani raziskavi povprečen ženska naredi 26 inch/ 66cm in moški 31 inch/78cm v enem koraku. [1]



Slika 3: prikaz aplikacije (strava), ki uporablja gps za zabeleževanje in prikaz poti

| Height | Women's Step Length (inches) | Women's Stride Length (inches) | Men's Step Length (inches) | Men's Stride Length (inches) |
|--------------|------------------------------|--------------------------------|----------------------------|------------------------------|
| 5 ft. 0 in. | 24.8 | 49.6 | 24.9 | 49.8 |
| 5 ft. 1 in. | 25.2 | 50.4 | 25.3 | 50.63 |
| 5 ft. 2 in. | 25.6 | 51.2 | 25.7 | 51.46 |
| 5 ft. 3 in. | 26.0 | 52.0 | 26.1 | 52.29 |
| 5 ft. 5 in. | 26.4 | 52.9 | 26.6 | 53.12 |
| 5 ft. 5 in. | 26.8 | 53.7 | 27.0 | 53.95 |
| 5 ft. 6 in. | 27.3 | 54.5 | 27.4 | 54.78 |
| 5 ft. 7 in. | 27.7 | 55.3 | 27.8 | 55.61 |
| 5 ft. 8 in. | 28.1 | 56.2 | 28.2 | 56.44 |
| 5 ft. 9 in. | 28.5 | 57.0 | 28.6 | 57.27 |
| 5 ft. 10 in. | 28.9 | 57.8 | 29.1 | 58.1 |
| 5 ft. 11 in. | 29.3 | 58.6 | 29.5 | 58.93 |
| 6 ft. 0 in. | 29.7 | 59.5 | 29.9 | 59.76 |
| 6 ft. 1 in. | 30.1 | 60.3 | 30.3 | 60.59 |
| 6 ft. 2 in. | 30.6 | 61.1 | 30.7 | 61.42 |
| 6 ft. 3 in. | 31.0 | 62.0 | 31.1 | 62.25 |
| 6 ft. 4 in. | 31.4 | 62.8 | 31.5 | 63.08 |
| 6 ft. 5 in. | 31.8 | 63.6 | 32.0 | 63.91 |

Slika 4: razmerje dolžine koraka v primerjavi z višino in spolom človeka

Program bi lahko izdelala tako, da bi uporabnika vprašala za spol ter njihovo višino, saj bi te podatke uporabila in se nato navzkrižno sklicevala po podatkih v tabeli.

10.5.1 Prednosti in slabosti uporabe gps-a

Gps je zelo natančno orodje ki lahko na 3-10m natančno določi lokacijo z uporabo satelitov (od 7 do 8). Zaradi tega je ena od najboljših načinov beleženja poti, hitrosti in korakov. Tudi veliko drugih aplikacij ga uporablja za te namene. To pa še ne pomeni da je nezmojljiv.

Poleg prednosti ki jih prinaša:

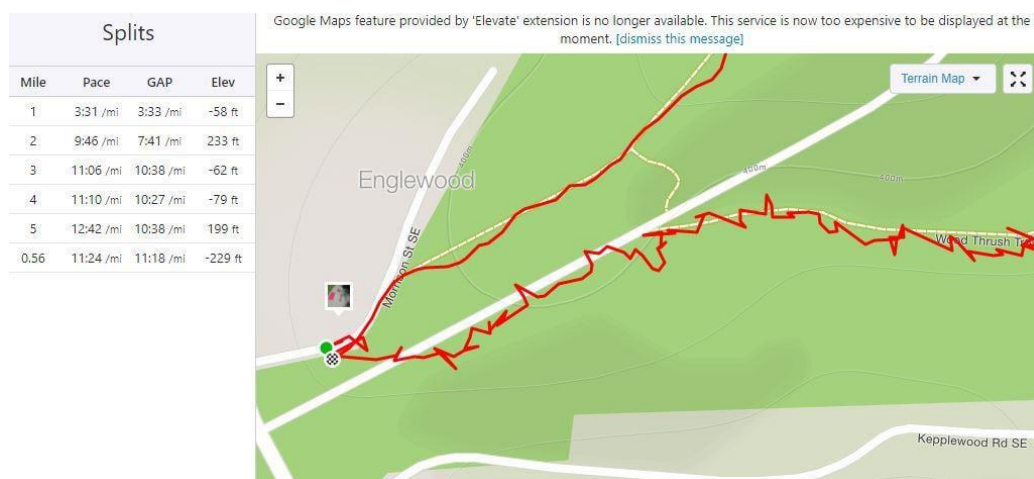
- jasna in lahka uporaba
- delovanje povsod po svetu
- prikaz že opravljene poti
- štetje dolžine
- natančna določitev pozicije
- ...

Se lahko najdejo tudi slabosti:

- povečana uporaba baterije
- napake pri določitvi pozicije
- napake na zemljevidu
- slabe kvaliteta naprave
- premalo satelitov
- slab signal za pridobitev koordinat

Poleg teh se pojavijo napake v drugih področjih, na primer ko med osebo in sateliti stoji stavba, gora, gozd... v tem primeru je možno da signal od satelitov ne pride do uporabnika, tako da gps neha delovati. [2]

11



Slika 5: primer gps-a ki ne deluje pravilno

Planiranje metode za določanje bpm pesmi

Za določanje bpm-a pesmi sva na začetku imela kar nekaj idej.

Kot pri števcu korakov sva ugotovila da že obstajajo programi ki to znajo narediti.

Postavila sva si izziv, da ta del programa narediva sama, brez tuje pomoči.

11.1 1. Način – zaznavanje onset-a

Najina prva ideja je bila uporabiti detektor Onset-a (začetka)

11.1.1 Onset

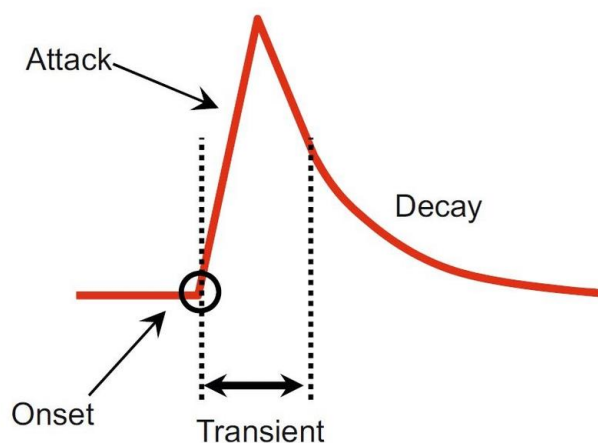
Onset se nanaša na začetek glasbene note ali drugega zvoka. Vse glasbene note imajo začetek, vendar ne vključujejo nujno začetnega prehoda.

Slika 6: prikaz onset-a

11.1.2 Pristopi k odkrivanju onset-a

Lahko delujejo kot časovni , frekvenčni , fazni ali kompleksni način in vključujejo iskanje:

- Povečanje spektralne energije
- Spremembe spektralne porazdelitve energije (spektralni tok) ali faze
- Spremembe zaznane višine - npr. z uporabo polifonega algoritma za zaznavanje višine
- Spektralni vzorci, ki jih prepoznajo tehnike strojnega učenja, kot so nevronske mreže.



Enostavnejše tehnike, kot je zaznavanje povečanja amplitude časovne premice, lahko običajno vodijo do prevelikega števila lažno pozitivnih ali lažno negativnih rezultatov.

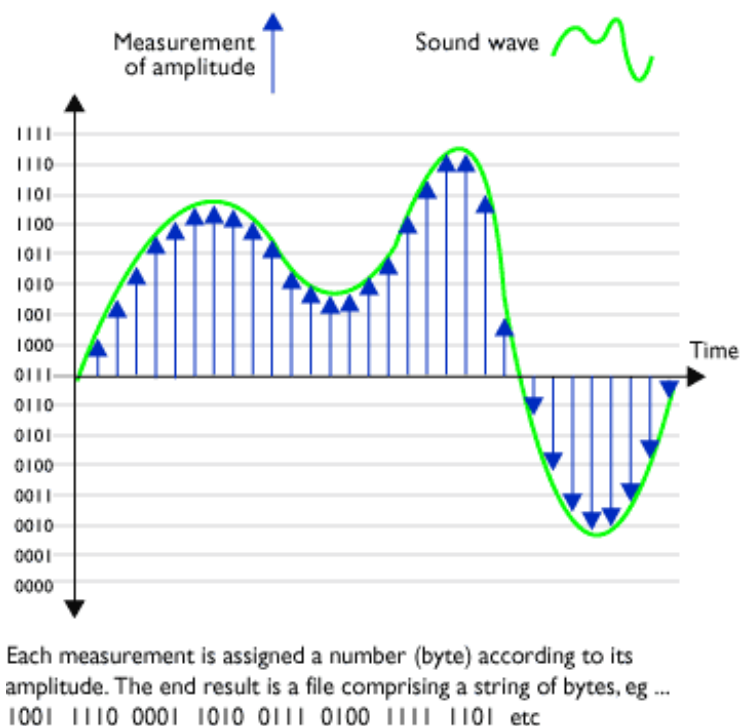
Cilj je pogosto oceniti zagone podobno kot bi jih človek: zato se lahko uporabijo psiho akustično motivirane strategije. Včasih je detektor začetkov lahko omejen na določeno področje (odvisno od predvidene uporabe), na primer usmerjen na zaznavanje udarnih začetkov. Z ožjim fokusom je lahko bolj preprosto doseči zanesljivo zaznavanje. [3]

11.2 Praktičen primer onset-a

Če bi želeli izdelati program za zaznavanje onset-a, se bi tega lahko lotili tako, da bi najprej spremenili mp3 datoteko v človeku berljivo obliko.

11.2.1 Kako se zabeležijo podatki v mp3 datoteki

Preden zvok lahko zabeležimo, ga moramo spremeniti iz analognega v digitalnega. To naredimo s kompresijskim sistemom, ki spremeni vibracije v določene vrednosti in jih shrani na disk, kot videno na sliki.

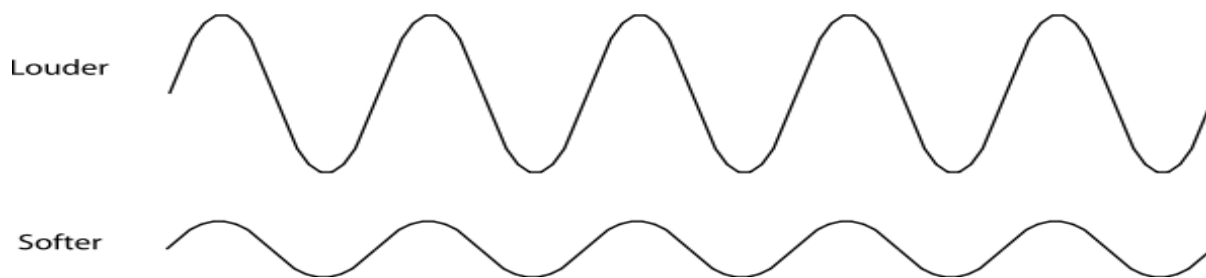


Slika 7: prikaz spreembe zvoka iz analognega v digitalnega

Vse vrednosti v mp3 datoteki so napisane na enakem časovnem intervalu. Zaradi tega pa je mogoče te vrednosti izraziti kot številke in jih med seboj primerjati. Z uporabo algoritma je mogoče zaznati velike spremembe v zaporedju in te označiti kot onset (začetek). S tem se zabeleži tudi zaporedno mesto bita na katerem se onset začne in tako pridobimo čas spremembe. Ti časi se nato zapišejo v novo datoteko. Nazadnje se primerja dolžina mp3 datoteke z številom onsetov da se pridobi število onsetov na časovno enoto ali bpm.

11.2.2 Prednosti in slabosti uporabe zaznavnika onset-a

Način uporabe onset-a prinaša veliko prednosti, glavne od teh je avtomatsko določanje bpm-a pesmi, kar je nepredstavljivo bolje, še posebej če ima uporabnik veliko pesmi.



Slika 8: prikaz primerjave tihe in glasne glasbe v digitalni obliki

Prinaša pa tudi slabosti, kot so napake pri branju datoteke. Te se lahko pojavijo pri mehanskih okvarah, okvarjenih mp3 datotekah (v katerih ni dovolj podatkov za natančno določitev) ali pa v samem delovanju programa. [4]

Zgornja Slika prikazuje spremembo amplitude/ digitalnih vrednosti ali onsetov tišje glasbe v primerjavi z glasnejšo. Zaradi tega lahko nastanejo napake pri branju onsetov, saj jih računalnik ne more zaznati v primeru da se pojavi pesem ki je enako glasna daljši čas.

11.3 2. Način

Včasih ne gre vse po načrtu. Ker najina prva ideja ni delovala, sva poiskala lažji način.

Ugotovila sva, da uporabniki radi poslušajo glasbo. To pa pomeni, da isto skladbo poslušajo večkrat. Zaradi tega sva si zamislila program ki bi vsaki pesmi zabeležil bpm tako, da bi utripe (onset-e) vnesel uporabnik sam.

11.3.1 Prednosti in slabosti samo vnosa bpm

Samo vnos bpm-a omogoča najbolj točen način pri katerem ne obstaja programskih napak. Zaradi tega nima nobenih slabosti, ki jih ima detektor onset-a. To pa ne pomeni da je brez slabosti. Pri večjih količinah pesmi, je samo vnos problematičen zaradi količine le teh. Največja slabost je ta, da deluje samo za tiste datoteke, ki jih je uporabnik sam izbral in jim opredelil bpm.

12 Možne ideje

Poleg originalnega bistva aplikacije sva hotela uporabnikom ponuditi tudi druge opcije, ki jim lahko pomagajo pri njihovi aktivnosti. Ena od teh je bila možnost zaznavanja srčnega utripa.

12.1 1. Način – uporaba perifernih naprav

Prva možnost je uporaba perifernih naprav, katere lahko zaznavajo srčni utrip. Android je do zdaj izdelal ogromno naprav, ki lahko nadzirajo uporabnikov srčni utrip. Te vključujejo zapestnice, pametne ure (smart watches) in tudi slušalke.



Slika 9: android pametna ura

Glavni razlog da se nisva odločila za to opcijo je ta, da nimava dovolj denarja, s katerim bi te naprave nabavila in testirala.

12.2 2. Način – uporaba vgrajene kamere

Nekatere Android naprave lahko zaznavajo srčni utrip preko kamere. To je mogoče s pomočjo tehnologije, imenovane optično zaznavanje srčnega utripa.

Ta tehnologija izkorišča dejstvo, da se barvni spekter kože spreminja s pulziranjem krvi v kapilarah. Pri tem pulziranju se spreminjata barvni ton in intenziteta kože, kar se odraža tudi na svetlobi, ki se odbija od kože. Kamere, ki so vgrajene v telefone in tablične računalnike, lahko zaznajo te spremembe in na ta način izračunajo srčni utrip.

Na žalost ta možnost ni primerjena cilju najine naprave, saj bi se od uporabnika zahtevala konstantna pozicija, kjer bi njegov prst pil položen pred kamero in primerno osvetljen, kar pri fizični aktivnosti ni mogoče.

Po premisleku sva se odločila da bi bila ta opcija za merjenje srčnega utripa preveč časovno potratna in nezanesljiva, saj je pri tem največja ovira zmogljivost in natančnost same naprave.

13 Povezovanje programov med sabo

Najina aplikacija je zgrajena na novi ideji, ki povezuje že prej izdelane aplikacije. Te so: števec korakov ter detektor bpm. Za glavni komponenti sva že našla rešitev. Narediti pa sva morala še izgled aplikacije.

Aplikacijo ki sva si jo zamislila bi vsebovala knjižnico mp3 datotek, ki jih aplikacija zazna na sistemu. Ob tem pa tudi gumb z katerim se uporabnikova seja z uporabo aplikacije začne, to je, kadar hoče da začne sinhronizirati glasbo z lastno aktivnostjo.

14 Praktični del

Delo se je začelo v programu Unity. Najprej na nivoju dokazovanja koncepta, ter učenju kako upravljati z zvokom. Za začetek sva uporabila dve različni skladi shranjeni v obliki .Wav datoteke s 3D ozadjem, ki ni imelo nobenega praktičnega pomena.

Nato sva se odločila program predstaviti in izdelati v android studio, saj sva slišala da je to zelo dobro integrirano razvojno okolje (IDE) za izdelavo android aplikacij. Preden pa vsa lahko začela pisati aplikacijo, sva se morala naučiti kako program deluje.

14.1 Android studio

Android Studio je integrirano razvojno okolje (IDE) za razvijanje mobilnih aplikacij za operacijski sistem Android. Razvil ga je Google in je osnovno orodje za razvijalce aplikacij Android.

Vsebuje številne funkcije, ki omogočajo razvijalcem aplikacij Android, da ustvarijo zmogljive, intuitivne in zanesljive mobilne aplikacije. Med najpomembnejšimi funkcijami so:

- Gradle Build System: sistem za avtomatizirano gradnjo aplikacij, ki olajša razvijanje, preizkušanje in distribucijo aplikacij.
- Layout Editor: orodje za vizualno oblikovanje uporabniškega vmesnika, ki omogoča razvijalcem, da hitro in enostavno ustvarijo privlačen in funkcionalen uporabniški vmesnik.
- Emulator: orodje, ki omogoča preizkušanje aplikacij na virtualni napravi Android, kar je zelo koristno pri razvijanju aplikacij za različne naprave in različne različice operacijskega sistema Android.
- Debugger: orodje za odpravljanje napak, ki omogoča razvijalcem, da hitro najdejo in odpravijo napake v svojih aplikacijah.

Android Studio je na voljo brezplačno in je odprtokodni projekt. Razvijalci lahko uporabljajo Android Studio za razvoj mobilnih aplikacij za različne namene, vključno z igranjem, komunikacijo, poslovanjem, učenjem in zabavo.

Android studio omogoča uporabo glavno dveh različnih programskih jezikov. To sta kotlin in Java.

14.1.1 Izbira med java in kotlinom

Java in Kotlin sta programska jezika, ki se pogosto uporabljata za razvoj mobilnih aplikacij Android. Sta objektno orientirana jezika, ki jih podpira Android Studio.

Java je bil prvi programski jezik, ki se je uporabljal za razvoj mobilnih aplikacij Android. Je zelo priljubljen programski jezik, ki se uporablja na različnih področjih, kot so razvoj programske opreme, spletni razvoj, mobilni razvoj in podobno. Java je poleg tega tudi enostaven in dober za začetnike.

Prednosti Jave:

- Stabilen in zanesljiv jezik z veliko knjižnicami
- Enostaven za učenje
- Velika skupnost razvijalcev, kar pomeni, da obstaja veliko dokumentacije in pomoči

Slabosti Jave:

- Včasih lahko postane nekoliko zapleten, še posebej pri večjih projektih
- Nekatere funkcije lahko zahtevajo veliko kode
- Včasih lahko postane počasen

Kotlin po drugi strani je relativno nov programski jezik, ki se vedno bolj uporablja za razvoj mobilnih aplikacij Android. Kotlin je bil razvit v JetBrainsu leta 2011 kot alternativa Javi in se osredotoča na poenostavitev in izboljšanje razvoja aplikacij. Kotlin ponuja nekatere napredne funkcije, ki jih Java ne podpira, kar olajša delo razvijalcem.

Prednosti Kotlin:

- Kotlin je bolj čitljiv in razumljiv, kar olajša razumevanje kode
- Kotlin je znan po svoji čitljivosti kode in preprostosti
- Boljša podpora za funkcionalno programiranje

Slabosti Kotlin:

- je relativno nov programski jezik, kar pomeni, da obstaja manjša skupnost razvijalcev in manj dokumentacije
- lahko zahteva nekoliko več kode v primerjavi z Javo, saj ima nekatere napredne funkcije

V primerjavi lahko rečemo, da je Kotlin v mnogih pogledih boljši od Jave, saj je bolj preprost za uporabo, ponuja nekatere napredne funkcije in je bolj berljiv in razumljiv. Kljub temu sva se odločila uporabljati Javo, saj je zaloga dokumentacije ki je na voljo veliko večja.

15 Izdelava bralca glasbe

Preden je bila aplikacija pripravljena določati bpm pesmi, jih je morala najprej zbrati. Za to funkcijo sva uporabila že vgrajeni media player¹. [5]

15.1 Problemi bralca glasbe

Med izdelavo aplikacije sva prišla do podobnega problema kot pri prejšnji verziji, to je, da ima program dostop samo do mp3 datotek, ki sva jih osebno naložila na aplikacijo, namesto da bi aplikacija sama od sebe zaznala datoteke na napravi, kar je bil cilj bralca glasbe.

15.2 Reševanje problema

Ker takšne aplikacije obstajajo na vsaki novejši napravi, sva vedela da jo je mogoče narediti, zato sva se lotila iskanja. Našla sva MediaStore² ki je omogočal uporabo zunanjih podatkov. [6]

15.3 Problemi pri implementiranju media store/media player

Po veliko poskusih in eksperimentiranju z media store in media player-jem, ni bilo veliko napredka. Zato sva se odločila za hitrejši in boljši način, ki je vključeval odprtokodno predlogo za glasbeno aplikacijo. Do spletne strani na kateri je bila ni več dostopa, saj so jo odstranili.[7]

Za ta pristop sva se odločila, saj je najin namen bil zgraditi aplikacijo, ki združuje že dve obstoječi aplikaciji da bi naredila nekaj inovativnega. Pri tem sva si vzela to aplikacijo kot podlago, na kateri sva gradila svoje ideje.

Aplikacija uporablja MediaStore, da dobi ime ter URI³ vseh avdio datotek na napravi, ter jih shrani v objekt⁴, ki jih nato shrani v polje. URI lahko nato uporabimo da dostopamo do datoteke ki jo predamo Media Playerju da jo predvaja. [8]

¹ MediaPlayer je razred v Android Studiu, ki omogoča predvajanje avdio datotek. Ponuja različne metode in funkcije kot so predvajanje, zaustavitev, premor in podobno.

² MediaStore je vgrajena aplikacija v Androidu, ki omogoča dostop do različnih vrst medijskih datotek. Uporablja centralno bazo podatkov za shranjevanje in upravljanje z medijskimi datotekami v napravi, kar omogoča lažji dostop in uporabo teh datotek. Ponuja različne opcije kot so pridobitev podatkov o datotekah ter dodajanje in spreminjanje datotek.

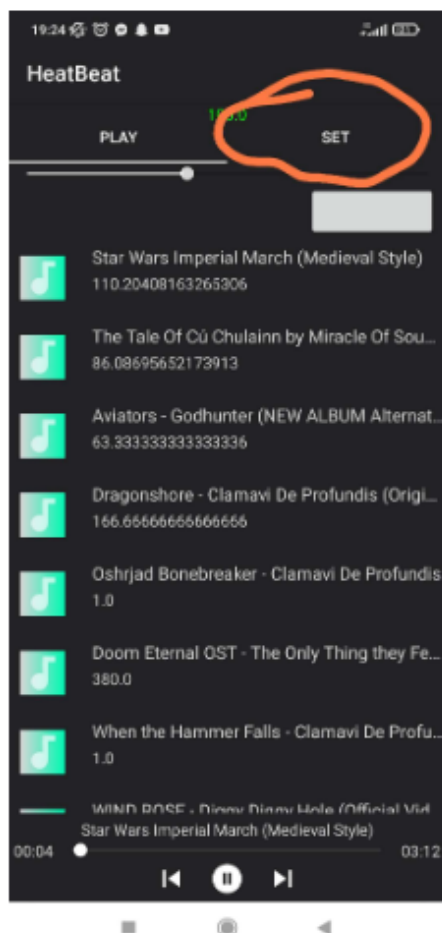
³ URI (enotni identifikator vira) je niz znakov, ki identificira vir, kot je spletna stran, datoteka ali strežnik. Sestavljen je iz dveh delov: sheme, ki določa vrsto vira, ter informacije, kot so ime in naslov. Uporablja se za dostop do virov preko spleta ali lokalne naprave. V Androidu se URI uporablja za določanje lokacije datotek ter za izvajanje drugih operacij, kot so dostop do spletnih storitev in baz podatkov.

⁴ V programiranju se objekt nanaša na posamezno pojavitev ali instanco razreda. Razred je predloga ali načrt, ki opisuje lastnosti, metode in obnašanje, ki jih ima lahko določen objekt. Objekt ima konkretne vrednosti teh lastnosti in izvaja metode razreda.

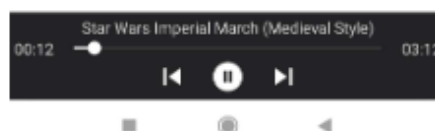
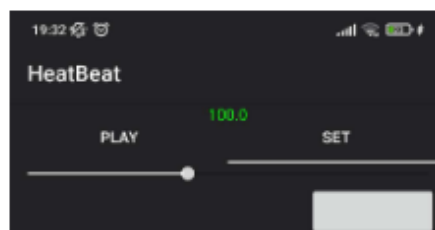
16 Izdelava vpisa bpm

Po uspešno narejeni glasbeni aplikaciji je bil čas za izdelavo opisa bpm-a. To bi naredila tako, da bi vsaki datoteki pripisala svoj ritem oziroma hitrost. Za to bi bil potreben naslednji proces.

Uporabnik klikne na gumb z imenom "SET" na kar se prikaže velik gumb na sredini ekrana in shrani točni trenutek, ko je uporabnik kliknil na gumb "SET" kot prikazano na spodnjih Slikah.



Slika 10: prikaz aplikacije z obkroženim gumbom "SET"



Slika 11: prikaz aplikacije po pritisku na gumb "SET"

Ob pritisku na sredinski gumb se spremenljivka ki šteje dotike inkrementira za ena vrednost in spremeni barvo gumba, da uporabniku signalizira uspešen pritisk.

Po kratkem obdobju klikanja na gumb v ritmu s pesmijo, lahko uporabnik klikne na gumb "PLAY", kar ga bo prestavilo na prejšnji zavihek (Slika 9). Ob enem pa se poleg tega tudi izračuna povprečen tempo glasbe, kjer se število dotikov sredinskega gumba deli z časom, ki ga je uporabnik porabil na "SET" zavihku.

16.1 Problem shranjevanje podatkov

Ob izhodu iz aplikacije sva prišla do naslednje težave. Vsi bpm_ji, ki so bili vpisani, so se izbrisali. Potrebovala sva način za shranjevanje trenutnih podatkov.

Potrebovala sva Bazo podatkov ali ekvivalentno datoteko, kjer bi shranjevala želene podatke. Izbrala sva SQLite⁵.

Z njim sva kreirala tabelo v kateri sva shranila imena pesmi in njihove ocenjene hitrosti. Ob zagonu aplikacije se nato prej omenjeni podatki preberejo iz podatkovne baze in uporabijo za prikaz pesmi in njihovih hitrosti.

⁵ SQLite je lahka, vgrajena relacijska baza podatkov, ki shranjuje podatke v tabelah in vzpostavlja relacije med njimi. Podatki v tabelah so organizirani v vrstice in stolpce, ki jih je mogoče indeksirati in poiskati. Uporablja se v mobilnih aplikacijah, vključno z Androidom, saj je zasnovan za delovanje na napravah z omejenimi viri in ima zelo majhno velikost datoteke ter porabi malo sistemskih virov.

17 Implementacija števca korakov

Ko sva začela delati na tem delu aplikacije, sva predvidevala da bo delo preprosto, saj obstaja veliko aplikacij, ki ponuja opcijo za štetje korakov. Na žalost sva se hudo motila.

17.1 1. Ideja

Delo sva začela z razvojem od začetka, saj sva bila prepričana da lahko sama ustvariva števec korakov. Pri tem sva uporabila acelerometer⁶ da sva si zabeležila smeri pospeškov ki so delovali na napravo (prej omenjen 2. način). [9]

Ob testiranju sva ugotovila, da bi za natančno mero rabila zabeležiti in procesirati veliko količino podatkov katere sva shranjevala v niz. Skrbelo pa naju je to, kako nepravilno in nenatančno ta metoda deluje.

17.2 2. Ideja

Ko sva opazila da nujna prva ideja ne bo zadostovala, sva si zamislila podobno verzijo, kjer bi vzela čim več vrednosti iz acelerometra ter bi jih obdelala na isti način, s tem da bi tokrat zaradi več vrednosti najina meritev bila bolj natančna.

To se je izkazalo da je neuporabno, saj je program uporabljal veliko virov, s tem da rezultati še zmeraj niso bili pravilni.

17.3 Rešitev problemov

Medtem ko sva se ukvarjala z raznimi načini izdelave aplikacije, sva ugotovila, da imajo naprave že vgrajen senzor za hojo. Sprva se nisva hotela uporabiti obstoječih rešitev. Po premisleku sva sprejela dejstvo, da najina rešitev ne bo boljša od profesionalnih.

Senzor ki sva ga uporabila deluje tako, da vsakih 5 sekund posodobi število korakov, ne šteje pa, kako hitro nekdo hodi. Zato sva morala števec narediti sama.

Začela sva tako, da vsakič ko se število korakov posodobi, se takratna vrednost odšteje od prejšnje. Na primer: če je števec korakov do zdaj preštel 1540, pred 5 sekundami pa 1520, je razlika 20 korakov. Iz tega sledi da je hitrost hoje 4 koraki na sekundo. Na žalost pa tudi to ni tako preprosto, saj sva pri tem srečala 2 večja problema.

⁶ Acelerometer je naprava, ki meri vibracije ali pospešek gibanja naprave. Sila, ki jo povzročijo vibracije ali sprememba gibanja (pospešek), povzroči, da masa "stisne" piezoelektrični material, ki proizvede električni naboj, ki je sorazmeren sili, ki deluje nanj.

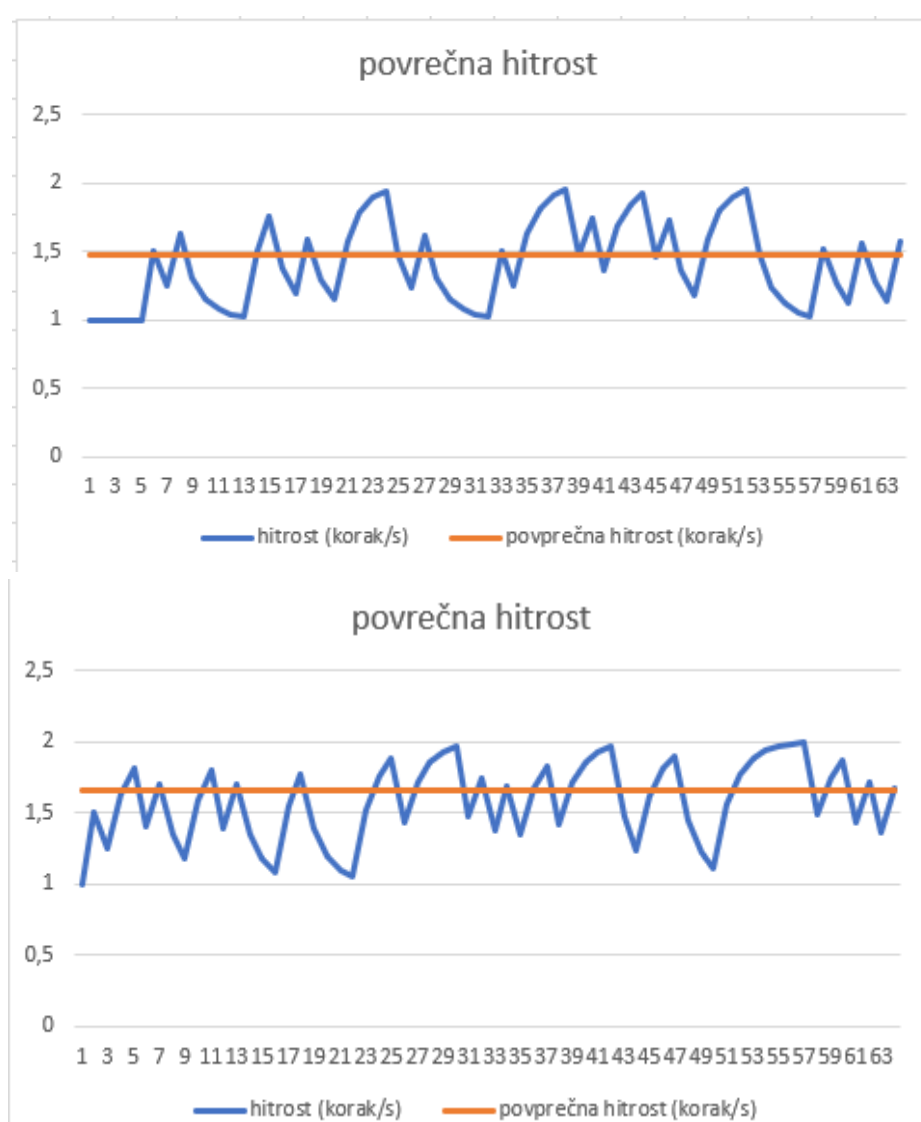
17.4 1. Problem

1. problem je v obliki nenatančnosti števca zaradi slabega načina štetja korakov. To je najlažje razložiti na primeru:

Recimo da oseba hodi 1.5 korakov na sekundo in števec posodobi vrednost vsako sekundo. Ker je števec zelo nepredvidljiv sva za naslednji primer uporabila naključno ponastavitev, kjer je v vsaki sekundi prebral 1 ali 2 koraka.

| | | | | | | | | | | | | | | | | | | | | |
|---|-------------------|---|-----|------|------|------|------|-----|------|------|------|------|-----|-----|------|------|------|------|------|------|
| 1 | čas (s) | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | |
| 2 | korak | 1 | 2 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 2 | 2 | 1 | 2 | 1 | 1 | 1 | 2 | 2 | 1 |
| 3 | hitrost (korak/s) | 1 | 1,5 | 1,25 | 1,63 | 1,81 | 1,41 | 1,7 | 1,35 | 1,18 | 1,59 | 1,79 | 1,4 | 1,7 | 1,35 | 1,17 | 1,09 | 1,54 | 1,77 | 1,39 |

Slika 10: tabela z vrednostimi časa, korakov ter povprečne hitrosti



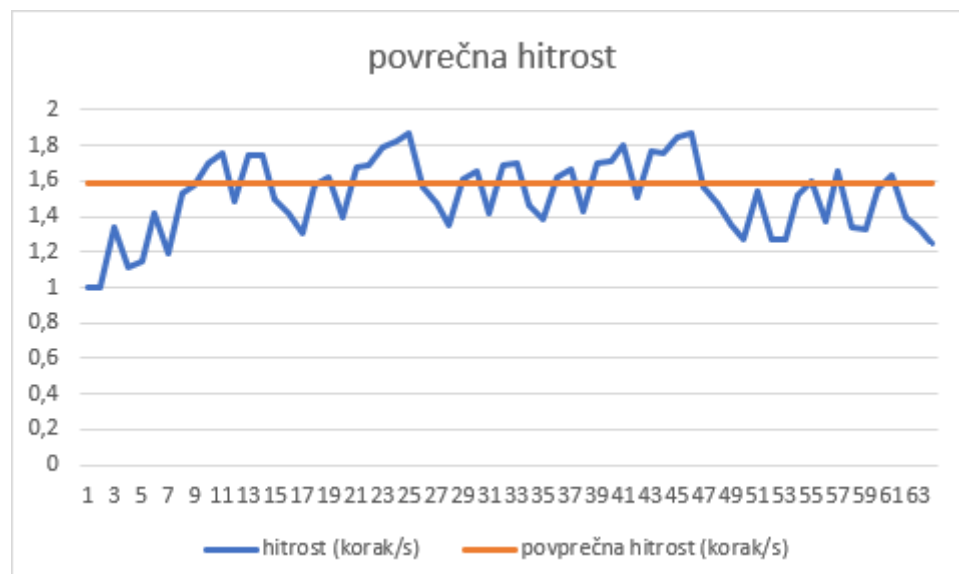
Slika 11: grafa z prikazano odmerjeno hitrostjo ter povprečno hitrostjo

V zgornjih primerih sva hitrosti dobila tako, da sva korak delila z preteklim časom, nato ga seštel s povprečno hitrostjo prejšnjega časa in končno delila z 2 da sva dobila povprečno hitrost. V zgornjem primeru bi to zgledalo tako, da sva rumeni korak (2), delila z rdečim časom (1), da sva dobila hitrost (n korakov/ 1 sekundo), ter to seštel s povprečno hitrostjo prejšnje zelene meritve (1) ter delila z 2, da sva dobila povprečno hitrost.

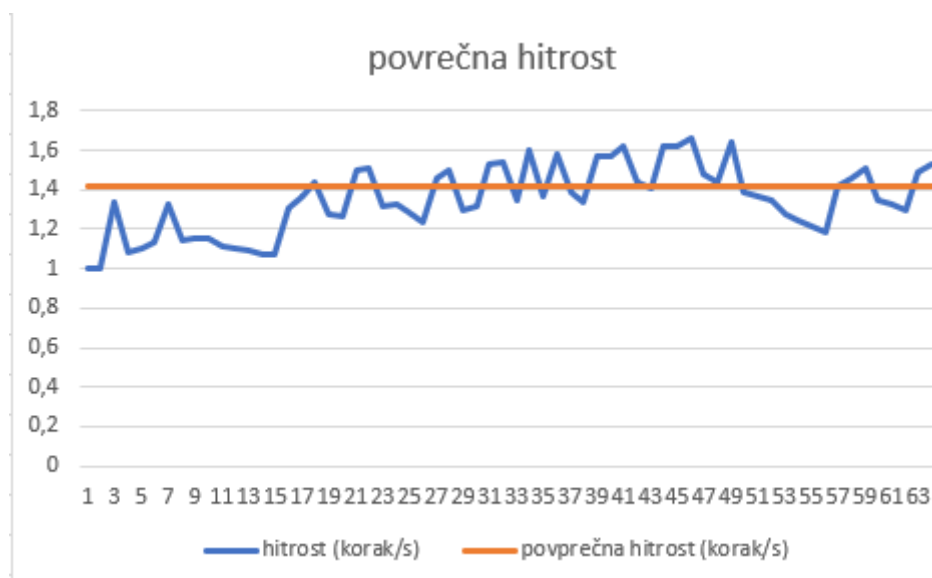
$$\frac{(\text{št. korakov/pretekel čas}) + \text{povprečna hitrost prejšnje meritve}}{2}$$

Kot je razvidno na grafih, ta opcija ne prinese zelo dobre hitrosti v trenutku meritve, čeprav je povprečna hitrost približno pravilna (± 1).

Eksperimentirala sva tudi z drugimi metodami. Poskusila sva vzeti povprečje večjih hitrosti skupaj:



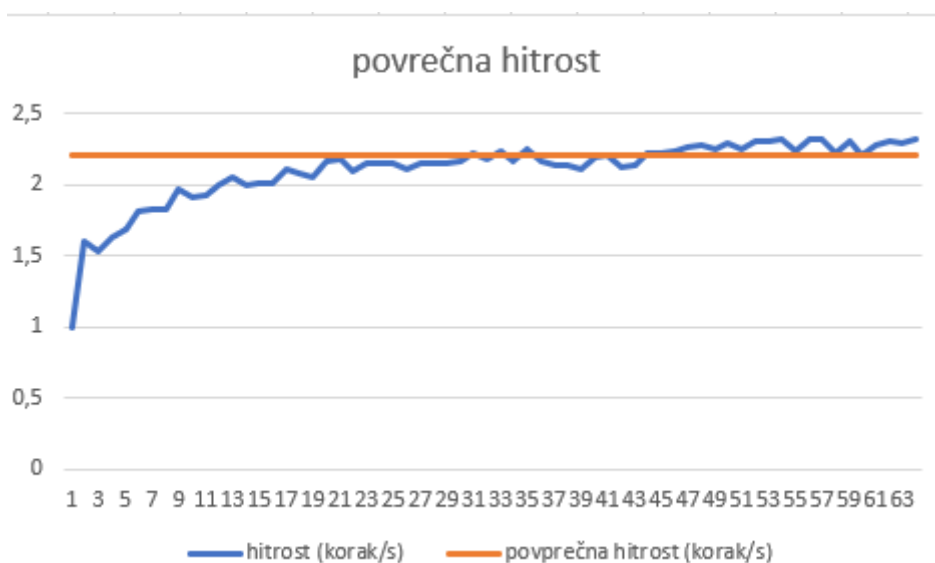
Slika 12: merjenje 3 povprečnih hitrosti



Slika 13: merjenje 4 povprečnih hitrosti

pri eksperimentiranju sva videla, da čeprav je mera postajala bolj in bolj natančna z večimi hitrostmi ki sva jih uporabila, je še zmeraj bilo videti 0.5 napake.

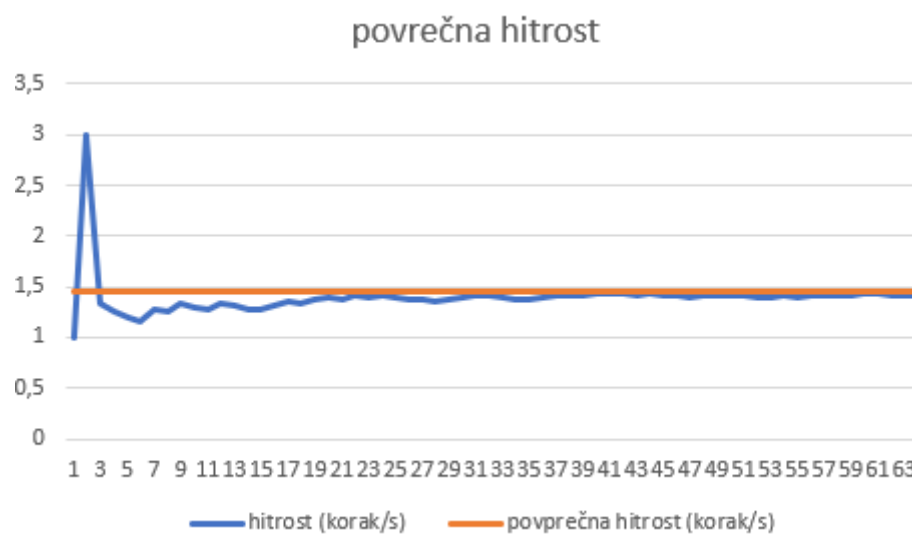
Po testiranju sva tudi ugotovila da je razlika bila veliko manjša, ko sva uporabila večja števila:



Slika 14: merjenje 4 povprečnih hitrosti

pri zgornjem primeru sva uporabila meritve, pri katerih je oseba hodila 10-12 korakov na 5 sekund, medtem pa so formule ostale enake. Iz tega sva ugotovila da so meritve na začetku največje, nato pa se počasi prilagajajo, čeprav nikoli ne uspejo priti do točnega povprečja.

Zato sva iskala druge načine merjenja in odkrila veliko bolj učinkovito metodo.



Slika 15: merjenje vrednosti trenutnega časa z celotno količino korakov

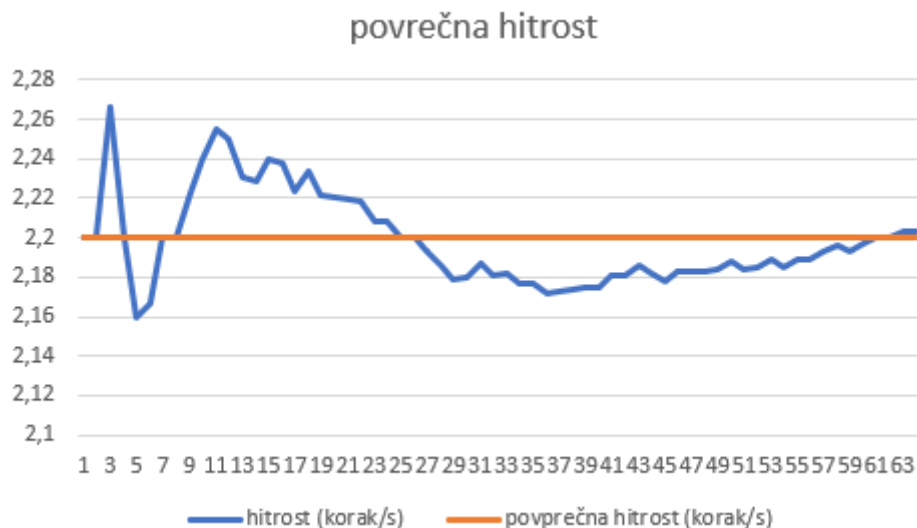
| | | | | | | | | | | | | | | | | | |
|-------------------|---|---|------|-----|-----|-----|------|-----|------|-----|------|-----|------|-----|------|-----|------|
| čas (s) | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
| korak | 2 | 4 | 5 | 6 | 7 | 9 | 11 | 12 | 13 | 15 | 16 | 18 | 19 | 21 | 22 | 24 | 25 |
| hitrost (korak/s) | 1 | 2 | 1,67 | 1,5 | 1,4 | 1,5 | 1,57 | 1,5 | 1,44 | 1,5 | 1,45 | 1,5 | 1,46 | 1,5 | 1,47 | 1,5 | 1,47 |

Slika 16: tabela vrednosti trenutnega časa in celotno količino korakov

Pri tej metodi sva namesto ene časovne enote v kateri senzor posodobi število korakov vzela celoten čas ki je pretekel in ga delila z vsemi koraki ki so do zdaj bili narejeni.

$$\frac{\text{št. korakov}}{\text{pretekel čas}}$$

To nama je opazno povečalo natančnost, še posebej po daljših meritvah.



Slika 17: merjenje vrednosti trenutnega časa z celotno količino korakov

Pri zgornjem primeru sva uporabila meritev ki je štela korake vsakih 5 sekund, povprečno od 10 do 12. z najino novo metodo sva prišla do maksimalno ± 0.1 razlike, kar je bila velika izboljšava od prejšnje metode.

17.5 2. Problem

Problem se pojavi, če na primer oseba hodi dlje časa. Število preteklih meritev je veliko, zato se pri trenutne spremembi hitrosti povprečna hitrost ne spremeni.

Zato sva morala narediti limiter, ki bi preveril če se je zgodila velika sprememba v količini korakov od prejšnje meritve do trenutne. Ta limiter sva nastavila na 20%. v primeru da bi se število korakov povečalo ali pomanjšalo za več kot 20%, se bi vse vrednosti o preteklem času in količini korakov zbrisale in začele na novo pridobivati od 0.

18 Povezovanje programov

Ko sva končno izdelala obe komponenti programa, sva ju morala povezati. To sva naredila z implementirano vrednostjo, ki je obdržala uporabnikovo željeno hitrost pesmi.

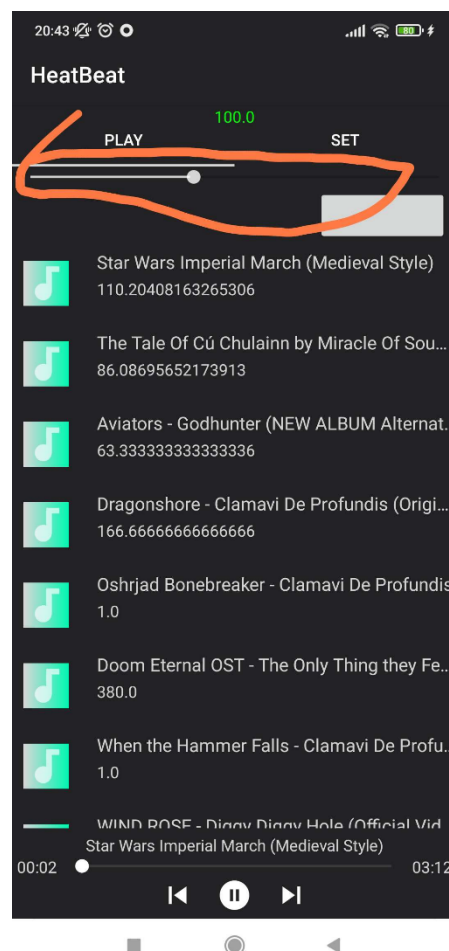
To sva pridobila od hitrosti hoje, število korakov na minuto.

Na začetku sva uporabljala samo preprosto spremenljivko nato pa sva naredila poseben objekt ki nadzira hitrost, da so spremembe hitrosti lažje opazne.

Ob koncu pesmi gre zanka skozi celotno polje pesmi s podanimi bpm-i in izbere tisto datoteko, katere bpm je najbližji trenutni hitrosti uporabnika ter jo začne predvajati.

Glavni in najbolj dosegljivi način za prilagoditev želene hitrosti je drsnik na vrhu ekrana (prikaz na sliki 20).

Omogoča razpon med 0 in 250, saj ljudje običajno tečejo med 150-180 korakov na minuto.



Slika 18: prikaz aplikacije z poudarkom na slide bar

19 Analiza

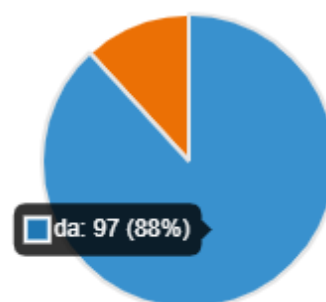
Pri najini analizi sva hotela biti temeljita, zato sva najino anketo poslala celotni srednji šoli kemije, elektrotehnike in računalništva.

Zahvaljujoč temu sva dobila 110 odgovorov, kar je po najinem mnenju zadostovalo za pridobitev pravih odgovorov.

Za prvo vprašanje naju je zanimalo koliko anketirancev posluša glasbo v prostem času.

1. Ali poslušate glasbo v prostem času?

[More Details](#)



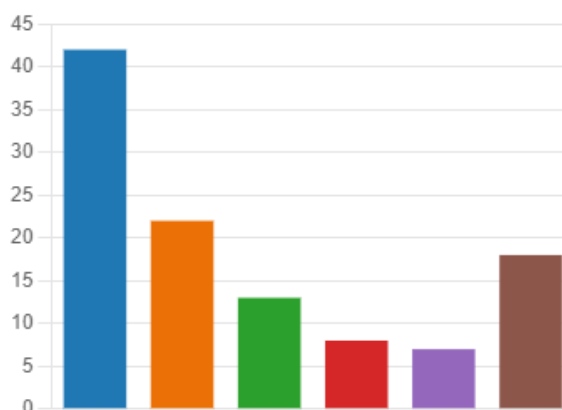
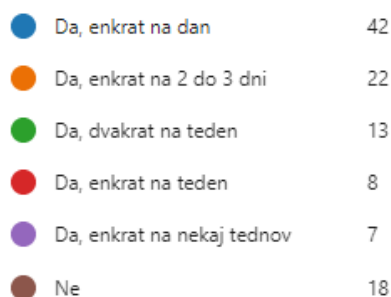
Graf 1: poslušanje glasbe v prostem času

Pri tem sva bila presenečena da okoli 12% ljudi ne posluša glasbe v prostem času.

Drugo vprašanje je služilo ugotovitvi, koliko udeležencev je fizično aktivnih in kako pogosto so anketiranci udeleženi v teh aktivnostih.

2. Ali redno telovadite?

[More Details](#)



Graf 2: fizična aktivnost v tednu

Iz tega vprašanja sva ugotovila da povprečno 16% udeležencev sploh ne telovadi in okoli 6% telovadi samo enkrat na nekaj tednov. Iz tega izhaja ugotovitev da je približno 78% (85) udeležencev vsaj enkrat na teden aktivnih.

Tretje vprašanje je služilo za določevanje tipa fizične aktivnosti ki ga anketiranci izvajajo.

3. Kako telovadite?

[More Details](#)

| | |
|---------------------------|----|
| ● Cardio(tek, hoja itd) | 48 |
| ● Uteži(dvigovanje uteži) | 44 |
| ● Other | 36 |



Graf 3: tip fizične aktivnosti

Tukaj sva bila srečno presenečena, saj je približno 44% odstotkov vseh udeležencev izvajalo vadbo za katero je najina aplikacija bila namenjena.

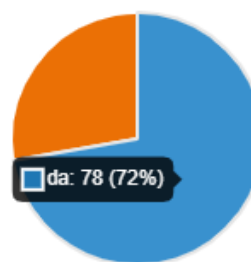
Pri četrtem vprašanju sva poskušala ugotoviti odgovor na najino hipotezo, kjer sva namerava izvedeti če anketiranci mislijo da ima glasba vpliv na njihove rezultate vadbe in tudi če pri vadbi poslušajo glasbo.

4. Ali mislite da ima glasba kakšen učinek na vaše rezultate pri vadbi?

[More Details](#)

[Insights](#)

| | |
|------|----|
| ● da | 78 |
| ● ne | 30 |



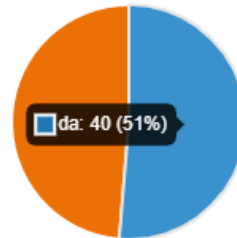
Graf 4: vpliv glasbe na rezultate aktivnosti

Presenetljivo nama je bilo da več kot 70% misli da ima glasba vpliv na njihove rezultate. Pri tem je tudi pomembno pomisliti na rezultate 1. vprašanja, saj je od 71% udeležencev kateri so odgovorili na to vprašanje z "da", 95% takšnih ki so odgovorili tudi na 1. vprašanje z "da". Iz česar lahko tudi sklepamo da večina ljudi ki posluša glasbo, pa tudi nekateri ki ne, meni da jim pomaga.

Nato sva poskušala ugotoviti, če anketiranci za predvajanje glasbe uporabljajo telefon.

6. Ali je ta glasba nameščena na vaš telefon?

[More Details](#)



Graf 5: uporaba telefona za poslušanje glasbe

Z temi informacijami sva ugotovila da je samo 50% takšnih, ki bi lahko najino aplikacijo uporabljali. Zaradi nepredvidenih razlogov sva tudi opazila pomanjšanje števila anketirancev.

20 Lastne ugotovitve

Za testiranje aplikacije sva poleg naju dobila še X udeležencev. Aplikacija sama od sebe deluje kot normalna aplikacija, v kateri lahko predvajaš in iščeš pesmi, poleg tega pa tudi meriš hitrost hoje. Deluje hitro in zanesljivo.

Pri testiranju aplikacije sva ugotovila da je najbolj nezaželeni del ta, da more uporabnik sam vnesti bpm pesmi. Na žalost tega problema nisva rešila. Ta del bova še skušala rešiti v prihodnosti.

Rezultati aplikacije so bili zelo zadovoljivi, saj je občutek teka pri glasbi ki se tako natančno usklajuje z koraki neverjeten in izredno priporočljiv.

21 Odgovori na hipoteze

1 HIPOTEZA: poslušanje glasbe izboljša rezultate fizične aktivnosti

Po testiranju lahko potrdiva, da poslušanje glasbe lahko nekaterim pomaga pri vzdržljivosti ter osredotočenosti na aktivnost.

1.1 HIPOTEZA: poslušanje glasbe izboljša možgansko delovanje pri fizični aktivnosti

Te hipoteze nisva mogla dokončno potrditi, saj nimava primerne opreme. Iz lastnih izkušenj pa lahko trdima da je pod vplivom glasbe osredotočenost na določeno stvar lažja.

1.2 HIPOTEZA: poslušanje glasbe izboljša telesno vzdržljivost pri fizični aktivnosti

Iz ankete ter testiranja lahko to hipotezo potrdiva.

2 HIPOTEZA: vsaj 50% odstotkov ljudi pri fizični aktivnosti posluša glasbo

Odgovor na to hipotezo je "da", saj glasbo pri fizični aktivnosti posluša povprečno 72% ljudi.

2.1 HIPOTEZA: vsaj 20% poslušalcev misli da jim glasba pri aktivnosti pomaga/prinaša boljše rezultate

Odgovor na to hipotezo je presenetljivi "da", saj skoraj vsi ki pri vadbi poslušajo glasbo tudi menijo da jim ta pomaga.

3 HIPOTEZA: Čeprav je bila to najina prva izkušnja z izdelavo aplikacije, sva menila da jo bova uspešno naredila.

Na žalost je ta hipoteza napačna, saj sva se naučila da je izdelava tudi lahke aplikacije zelo težka.

Poleg hipotez sva si zadala cilj ki je bil zgraditi delujočo aplikacijo, ki pa sva ga uspešno izpopolnila.

22 Zaključek

Kljub težavam, ki so se nenehno pojavljale, sva uspešno izdelala delujočo verzijo aplikacije na kar sva zelo ponosna. Največ problemov sva imela pri implementiranju dobre rešitve za pridobivanje bpm-a pesmi, saj je večina že narejenih programov napisana v drugih jezikih, kot so python ali C.

Ob tem sva morala biti pozorna na izgled aplikacije, saj enostavnost uporabe pomemben faktor izkušnje uporabnika.

Meniva da je aplikacija v trenutnem stanju zadovoljiva čeprav so izboljšave še možne. Izboljšavo glede pridobivanja bmp-ja bova skušala rešiti v prihodnosti.

Ob delu na raziskovalni nalogi sva se veliko naučila in odkrila nove stvari. Na primer: veliko ljudi ne samo posluša glasbo pri vadbi, ampak tudi meni da jim to pomaga. Zato sva vesela, da si lahko zdaj pomagajo z najino aplikacijo.

1. Seznam literature

ACCELEROMETER. (15. december 2022). Pridobljeno iz Android: https://developer.android.com/reference/android/hardware/Sensor#TYPE_ACCELEROMETER[9]

android studio. (16. August 2022). Pridobljeno iz youtube: <https://www.youtube.com/watch?v=DG7OYSTxPUA>

audio. (12. September 2022). Pridobljeno iz GitHub: <https://github.com/dmcennis/jAudioGIT>

audio conversion. (23. February 2023). Pridobljeno iz planetoftunes: <http://www.planetoftunes.com/digital-audio/how-do-analogue-to-digital-converters-work.php>[3]

audio processing. (12. September 2022). Pridobljeno iz GitHub: <https://github.com/JorenSix/TarsosDSP/blob/master/README.textile>

audio processing. (12. September 2022). Pridobljeno iz sapandiwakar: <https://sapandiwakar.in/audio-processing-on-android-using-tarsosdsp/>

audio processing. (12. September 2022). Pridobljeno iz stackoverflow: <https://stackoverflow.com/questions/36971839/tarsosdsp-clap-detection>

audio processing. (14. September 2022). Pridobljeno iz 0110.be: https://0110.be/posts/TarsosDSP_on_Android_-_Audio_Processing_in_Java_on_Android

GPS. (23. Febraury 2023). Pridobljeno iz hellotracks: <https://hellotracks.com/en/blog/How-to-Improve-your-GPS-Accuracy/>[2]

IDE. (4. March 2022). Pridobljeno iz Android: <https://developer.android.com/studio/intro/update>

IDE's. (4. November 2022). Pridobljeno iz appradar : <https://appradar.com/blog/mobile-game-engines-development-platforms>

java. (25. February 2023). Pridobljeno iz stack overflow: <https://stackoverflow.com/questions/2808535/round-a-double-to-2-decimal-places>

Java dependencies. (25. September 2022). Pridobljeno iz jar-download: https://jar-download.com/maven-repository-class-search.php?search_box=javafx.sound.sampled.UnsupportedAudioFileException

media player. (2. February 2022). Pridobljeno iz Android: <https://developer.android.com/guide/topics/media/mediaplayer>[5]

Media Store. (2. February 2022). Pridobljeno iz Android: <https://developer.android.com/reference/android/provider/MediaStore>[6]

mediaplayer. (12. December 2022). Pridobljeno iz *11zon*:
https://www.11zon.com/android/android_mediaplayer.php [7]

mobile IDE's. (4. November 2022). Pridobljeno iz g2: <https://www.g2.com/categories/mobile-development-platforms>

motion sensor. (14. decemeber 2022). Pridobljeno iz Android:
https://developer.android.com/guide/topics/sensors/sensors_motion

notification. (2. December 2022). Pridobljeno iz Android:
<https://developer.android.com/develop/ui/views/notifications/build-notification#groovy>

onset. (21. May 2022). Pridobljeno iz wikipedia: [https://en.wikipedia.org/wiki/Onset_\(audio\)](https://en.wikipedia.org/wiki/Onset_(audio)) [3]

pitch analysis. (23. december 2022). Pridobljeno iz stackoverflow:
<https://stackoverflow.com/questions/31231813/tarsosdsp-pitch-analysis-for-dummies>

reasearch for running. (24. february 2023). Pridobljeno iz Marathon Handbook:
<https://marathonhandbook.com/average-stride-length/#:~:text=This%20research%20also%20reports%20that,stroke%20length%20is%2062%20inches> [1]

refresh button. (4. March 2022). Pridobljeno iz OnlineWebFonts:
<https://www.onlinewebfonts.com/icon/72363>

reverting a project. (8. January 2023). Pridobljeno iz Stack Overflow:
<https://stackoverflow.com/questions/38163380/how-to-revert-project-back-to-a-previous-commit-in-android-studio#:~:text=Android%20Studio%20-%3E%20Version%20Control%20-,change%2C%20commit%20-%3E%20done>

running research. (25. Januray 2023). Pridobljeno iz Strength Running:
<https://strengthrunning.com/2020/02/best-running-cadence-step-rate/#:~:text=If%20you%27re%20running%20at,be%20220%20steps%20per%20minute!> [1]

running research. (25. January 2023). Pridobljeno iz TrainingPeaks:
<https://www.trainingpeaks.com/blog/finding-your-perfect-run-cadence/#:~:text=Most%20recreational%20runners%20will%20have,seen%20in%20runners%20who%20overstride> [1]

sensor. (14. december 2022). Pridobljeno iz Android:
https://developer.android.com/guide/topics/sensors/sensors_overview

Sound. (23. February 2023). Pridobljeno iz earmaster: <https://www.earmaster.com/music-theory-online/ch03/chapter-3-1.html> [4]

URI. (23. December 2022). Pridobljeno iz stackoverflow:
<https://stackoverflow.com/questions/13209494/how-to-get-the-full-file-path-from-uri> [8]

23 Viri slik

IMAGE. (21. Decemeber 2022). Pridobljeno iz mathinsight:
https://mathinsight.org/cartesian_coordinates

IMAGE. (18. Decemeber 2022). Pridobljeno iz Toppr Ask:
<https://www.toppr.com/ask/question/a-ball-is-thrown-vertically-upwards-which-of-thefollowing-plots-represents-the-speedtime-graph-of/>

IMAGE. (26. December 2022). Pridobljeno iz planet of tunes:
<http://www.planetoftunes.com/digital-audio/how-do-analogue-to-digital-converters-work.php>

IMAGE. (23. December 2022). Pridobljeno iz earmaster: <https://www.earmaster.com/music-theory-online/ch03/chapter-3-1.html>

IMAGE. (29. December 2022). Pridobljeno iz GRAN FONDO Cycling Magazine:
<https://granfondo-cycling.com/strava-routen-news/>

IMAGE. (2. January 2023). Pridobljeno iz research gate:
https://www.researchgate.net/figure/Onset-of-the-envelope-of-the-audio-signal-2_fig1_315835543

IMAGE. (2. February 2023). Pridobljeno iz Marathon Handbook:
<https://marathonhandbook.com/average-stride-length/#:~:text=This%20research%20also%20reports%20that,stroke%20length%20is%2062%20inches>

IMAGE. (13. February 2023). Pridobljeno iz reddit:
https://www.reddit.com/r/Strava/comments/cnpn1n/gps_malfunction_at_the_beginning_of_a_run/

IZJAVA*

Mentor/-ica Jaka Koren v skladu z 20. členom Pravilnika o organizaciji mladinske raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje, zagotavljam, da je v raziskovalni nalogi z naslovom HeatBeat, katere avtorja sta Dominik Zagrušovcem in Nik Deželak.

- besedilo v tiskani in elektronski obliki istovetno,
- pri raziskovanju uporabljeno gradivo navedeno v seznamu uporabljene literature,
- da je za objavo fotografij v nalogi pridobljeno avtorjevo dovoljenje in je hranjeno v šolskem arhivu,
- da sme Osrednja knjižnica Celje objaviti raziskovalno nalogo v polnem besedilu na knjižničnih portalih z navedbo, da je raziskovalna naloga nastala v okviru projekta Mladi za Celje,
- da je raziskovalno nalogo dovoljeno uporabiti za izobraževalne in raziskovalne namene s povzemanjem misli, idej, konceptov oziroma besedil iz naloge ob upoštevanju avtorstva in korektnem citiranju,
- da smo seznanjeni z razpisni pogoji projekta Mladi za Celje.

Celje, 5.4.2023



Podpis mentorja

Podpis odgovorne osebe

*

POJASNILO

V skladu z 20. členom Pravilnika raziskovalne dejavnosti »Mladi za Celje« Mestne občine Celje je potrebno podpisano izjavo mentorja (-ice) in odgovorne osebe šole vključiti v izvod za knjižnico, dovoljenje za objavo avtorja (-ice) fotografskega gradiva, katerega ni avtor (-ica) raziskovalne naloge, pa hrani šola v svojem arhivu.