Dokumentácia

Aplikácia Volume Meter

Kristián Lepka, 5ZYR22

Popis a analýza riešeného problému

Cieľom práce bolo vytvoriť Android aplikáciu pre meranie hlasitosti zvuku. Vzhľadom na hardvér v mobilných zariadeniach je možné merať iba približné hodnoty hlasitosti. Toto ale postačuje k určeniu, či je dané prostredie hlučné a škodlivé pre sluch. Aplikácia sa dá teda využiť najmä k orientačnému meraniu a zisteniu škodlivosti hlučného prostredia.

Cieľom bolo implementovať:

* Meranie výšky hlasitosti nahrávaného zvuku
* Vyjadrenie nameranej hodnoty pomocou budíka, číselnej hodnoty a slovne
* Vyjadrenie nameranej hodnoty pomocou grafu
* Prehľad o štatistikách ako maximálna nameraná hodnota, minimálna nameraná hodnota, priemerná nameraná hodnota a aktuálna nameraná hodnota
* Kalibrácia aplikácie pri prvom spustení
* Jednoduché ovládanie a navigácia

Podobné aplikácie

Podobných aplikácii sa v Google Play store nachádza (minimálne)niekoľko desiatok. Väčšina skontrolovaných obsahuje dizajn budíka s ručičkou a meranie v reálnom čase spolu s históriou merania v podobe grafu. Niektoré poskytujú aj možnosť uložiť nahrávku do zariadenia a rôzne témy. Pre bližšie porovnanie som si vybral aplikáciu Sound Meter od Abc Apps.

Aplikácia poskytuje číselné vyjadrenie hluku ako aj prirovnanie k rôznym hlučným udalostiam. Meranie je možné pozastaviť, kalibrovať, zobraziť históriu a vymazať ju. Obsahuje pár nastavení a tmavú/svetlú tému.

A screenshot of a device

Description automatically generated with medium confidence

Obrázok 1 Aplikácia Sound Meter od Abc Apps

A screenshot of a phone

Description automatically generated with medium confidence

Obrázok 2 Aplikácia Sound Meter od Abc Apps

Návrh riešenia

Základ aplikácie tvorí trieda MediaRecorder a vytvorené abstrakcie nad touto triedou.

MediaRecorder je pomerne komplikovaná trieda na rýchle použitie. Je to tzv. state machine – v závislosti od toho, čo práve robíme (nahrávanie, pauza v nahrávaní), sú niektoré metódy nedostupné, resp. ak ich zavoláme dostaneme IllegalStateException.

A picture containing text, diagram, screenshot, circle

Description automatically generated

Obrázok 3 Stavový diagram triedy MediaRecorder

MediaRecorder je pred použitím potrebné aj správne inicializovať. Nesprávne poradie volania metód dokáže spôsobiť výnimku. Preto som považoval za nevyhnutné vytvoriť triedu ktorá obsluhu MediaRecorder-a zjednoduší.

Táto trieda je SoundRecorder. Zjednodušuje obsluhu za použitia 3 public metód – start, pause, stop – ktoré ovládajú MediaRecorder. Metóda getDecibelValue dokáže vypočítať na základe amplitúdy signálu hlasitosť v decibeloch.

Hodnota v decibeloch sa počíta pomocou vzorca

, kde predstavuje nameranú amplitúdu a maximálnu nemerateľnú hodnotu. V prípade triedy MediaRecorder je táto hodnota 32767. Kvalita audio nahrávky triedy MediaRecorder je 16 bitov – to znamená že dynamický rozsah nahrávky je 96db. Z toho vyplýva že samotný rozsah nameranej hlasitosti je -96db až 0db pričom -96 predstavuje najmenší možný signál a 0db najvyšší. Takto nameraná hodnota je posunutá o hodnotu offset pri kalibrácií. Výsledok dobre odhaduje reálnu hodnotu hlasitosti. Pre ďalšie vysvetlenie som spolu s dokumentáciou zahrnul PDF dokument z internetu.

Trieda SoundRecorder umožňuje aj ukladanie nahrávky do súboru alebo jej zahodenie. Ďalšou podstatnou vlastnosťou je property callibrationOffset ktorá upravuje nameranú hlasitosť.

A screen shot of a computer

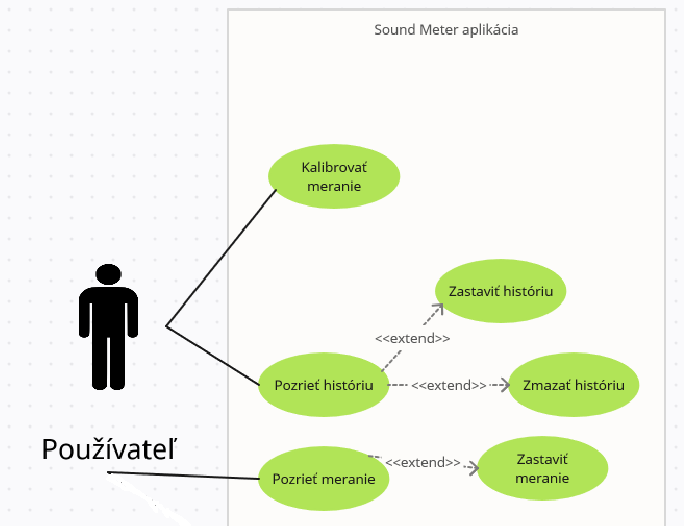
Description automatically generated with medium confidence

Obrázok 4 Inicializácia triedy MediaRecorder

Ďalšiu úroveň funkcionality poskytuje trieda VolumeRecorder. Táto trieda obsahuje ako field inštanciu SoundRecorder a poskytuje ešte jednoduchšie ovládanie. Nahrávanie sa ovláda dvoma metódami – switchRecording a stopRecording. Metóda switchRecording zapne alebo pozastaví nahrávanie a metóda stopRecording ho zruší úplne. Zaujímavým prídavkom sú udalosti (Event). Trieda event je inšpirovaná jazykom C#, táto mechanika mi prišla užitočná. Kód pre triedu Event<T> som našiel na stackOverflow a mierne ho upravil aby lepšie sedel mojím potrebám. Eventy v tejto triede slúžia na informovaní o zmenách v nahrávaní – nameranie novej hodnoty, zmena stavu nahrávania, zastavenie nahrávania a zmena popisného reťazca. Ďalšia funkcionalita je periodické získavanie hlasitosti počas nahrávania. Dĺžka periódy môže byť zmenená. Popisný reťazec je jednoduchý slovný popis aktuálnej hlasitosti – slovné prirovnanie napr. k vysávaču.

V hlavnej aktivite som využil

Diagram prípadov použitia



Obrázok 5 Diagram prípadov použitia aplikácie

Diagram tried

Zahrnutý aj ako samostatný obrázok k dokumentácií. Popisuje základnú štruktúru programu pre rýchle pochopenie vzťahov medzi triedami.

A picture containing text, diagram, screenshot, line

Description automatically generated

Obrázok 6 Diagram tried aplikácie

Popis jednotlivých častí aplikácie

Aktivity a fragmenty

* MainActivity
  + hlavná aktivita aplikácie
  + obsahuje fragmenty aplikácie
  + zodpovedá za prepínanie medzi fragmentmi
  + inicializácia spodného a horného menu, akcie tlačidiel
  + kontrola povolení pri spustení aktivity
  + horný action bar obsahuje akcie pre kontrolu nahrávania
  + spodné menu slúži na navigáciu v aplikácií
* MainFragment

A screenshot of a device

Description automatically generated with medium confidence

Obrázok 7 Obrazovka s MainFragment

* + hlavný fragment aplikácie
  + obsahuje ovládanie pre nahrávanie zvuku
  + obsahuje check box pre výber uloženia nahrávky a slider pre úpravu rýchlosti merania – nová hodnota každých x milisekúnd
  + hlavný budík pre zobrazenie nameranej hodnoty hlasitosti
  + slovné prirovnanie a číselná hodnota
  + metóda pre nastavenie vzhľadu budíka
  + použitá knižnica simple-gauge-android
* HistoryFragment

A screen shot of a graph

Description automatically generated with medium confidence

Obrázok 8 Obrazovka s HistoryFragment

* + obsahuje graf histórie nameraných hodnôt
  + história je zobrazená pokiaľ je nahrávanie v stave Started/Paused
  + v stave Stopped graf nezobrazuje žiadne dáta
  + história zobrazuje dáta 10 sekúnd do minulosti
  + použitá knižnica MPAndroidChart
* InfoFragment

A screenshot of a cell phone

Description automatically generated with medium confidence

Obrázok 9 Obrazovka s InfoFragment

* + informácie o maximálnej, minimálnej a priemernej nameranej hodnote
  + informuje aj o aktuálnej hodnote hlasitosti
  + dáta sa zresetujú keď sa nahrávanie úplne zastaví
* CalibrationFragment
  + zobrazený pri prvom štarte aplikácie
  + slúži pre kalibráciu merania

A screenshot of a phone

Description automatically generated with medium confidence

Obrázok 10 Obrazovka s CalibrationFragment

AndroidX Komponenty

Využitý je ViewModel, DataBinding, LiveData a LifeCycles

A picture containing text, screenshot, font

Description automatically generated

Obrázok 11 Príklad využitia View Modelu, HistoryViewModel.kt

A screen shot of a computer code

Description automatically generated with low confidence

Obrázok 12 Príklad využitia Data Bindingu, fragment\_main.xml

A picture containing text, screenshot, font

Description automatically generated

Obrázok 13 LiveData využité vo View Modeli, SoundMeterViewModelBase.kt

A screen shot of a computer code

Description automatically generated with low confidence

Obrázok 14 Využitie Life Cycle Ownera vo fragmente, HistoryFragment.kt

View Modely

* MainViewModel
* CalibrationViewModel
* InfoViewModel
* HistoryViewModel
* abstraktná trieda SoundMeterViewModelBase

Použité knižnice

MPAndroidChart

* knižnica pre vytváranie a zobrazovanie rôznych druhov grafov v Android a iných aplikáciach
* vzhľad jednotlivých grafov sa dajú upraviť mnohými spôsobmi
* knižnica poskytuje animácie, kombinované grafy, legendy, dotykové akcie a kompatibilitu aj s iOS a Xamarinom
* v projekte je konkrétne využitý LineChart a vlastný štýl
* <https://github.com/PhilJay/MPAndroidChart>

A screen shot of a computer program

Description automatically generated with low confidence

Obrázok 15 Stylovanie grafu typu LineChart

simple-gauge-android

* knižnica obsahujúca 4 meniteľné budíky
* jednoduché použitie a dobré prispôsobenie vzhľadu
* použitá v hlavnom fragmente MainFragment, konkrétne FullGauge
* <https://github.com/Gruzer/simple-gauge-android>

A screen shot of a computer program

Description automatically generated with medium confidence

Obrázok 16 Nastavovanie zón a dizajnu budíka FullGauge

Využitie senzorov

Využitý je mikrofón pre nahrávanie audia. Bolo potrebné vyžiadať práva k nahrávaniu v Android manifeste. Pre ukladanie súboru s nahrávkou bolo potrebné aj právo pre prístup k súborom.

A picture containing text, screenshot, font

Description automatically generated

Obrázok 17 Vyžiadanie práv v AndroidManifest.xml

Zoznam použitých zdrojov

[stackoverflow.com](https://stackoverflow.com/)

[geeksforgeeks.org](https://www.geeksforgeeks.org/)

[androidforums.com](https://androidforums.com/)

[youtube.com](https://www.youtube.com/)

[weeklycoding.com/mpandroidchart-documentation](https://weeklycoding.com/mpandroidchart-documentation/)

[javadoc.jitpack.io/com/github/PhilJay/MPAndroidChart/v3.1.0/javadoc](https://javadoc.jitpack.io/com/github/PhilJay/MPAndroidChart/v3.1.0/javadoc/)

[github.com/Gruzer/simple-gauge-android#readme](https://github.com/Gruzer/simple-gauge-android#readme)