bewantbe / audio-analyzer-for-android

A fork of audio-analyzer-for-android in Google code, with a lot of enhancement.

245 commits | **2** branches | **13** releases | **4** contributors | Apache-2.0

Branch: **master** | **New pull request** | **Create new file** | **Upload files** | **Find file** | **Clone or download**

bewantbe Add CalibrationLoad.java wich is missing in previous commit. ··· Latest commit `faab219` on 7 May

FFTLibrary	Fix some typo and Lint warnings.	9 months ago
Release	Add release build.	8 months ago
audioSpectrumAnalyzer	Add CalibrationLoad.java wich is missing in previous commit.	7 months ago
doc	Add release build.	8 months ago
gradle/wrapper	Update User Manual Text and Preferences Context	9 months ago
util	Remove unused and ugly colormaps.	9 months ago
.gitignore	Remove unused and ugly colormaps.	9 months ago
LICENSE	Add license notes.	2 years ago
README.md	Boost version number. Add code notes in readme.	8 months ago
README.old	Fix some typo and Lint warnings.	9 months ago
build.gradle	Gradle android Plugin v2.3.0-rc1 causes high CPU in device, and cause…	9 months ago
gradlew	Convert the whole project (Eclipse) to Gradle project.	3 years ago
gradlew.bat	Convert the whole project (Eclipse) to Gradle project.	3 years ago

settings.gradle	Convert the whole project (Eclipse) to Gradle project.	3 years ago

README.md

Audio Spectrum Analyzer for Android

> A fork of Audio spectrum Analyzer for Android (See README.old for its original readme)

This software shows the frequency components' magnitude distribution (called spectrum) of the sound heard by your cell phone. Can be used to help tuning musical instrument or tone in singing, (tentative) measure environmental noise and sound revent education or experiments.

You can install this app through **Google Play Store**: Audio Spectrum Analyzer. Comments are welcome.

Features

- Show spectrum or spectrogram in real-time, with decent axis labels.
- Linear, Logarithm and (Musical) Note frequency axis support.
- You can put a cursor in the plot, for measurement or as a marker.
- Easy gestures to fine exam the spectrum: i.e. pinch for scaling and swipe for view move.
- Show peak frequency in a moderate accuracy (FFT + interpolation).
- Show dB or A-weighting dB (dBA), although not suitable for serious application.
- Possible to take averages of several spectrum then plot, make the spectrum smoother.
- You may record the sound (while analyzing!) to a WAV file (PCM format). Then you can deal with it with your favorite tool.

- Support all recorder sources except those need root privilege (see list in Android reference: MediaRecorder.AudioSource)
- Support all possible sampling rates that your phone is capable. e.g. useful to find out the native (or best) sampling format for you phone.

Installation Requirements

- > = Android 2.2 (API Level 8). The recent version need Android 2.3 (API Level 9).
- External storage (e.g MicroSD card), if you want to record the sound.

Development

`git clone` then open it use Android Studio. Install the SDK platform if requested (e.g. rev 116 needs API level 20), or tune the `compileSdkVersion` to the value that fits your needs.

Code structure

The whole program structure is roughly follows the MVC model:

AnalyzerActivity.java is the controler, as the main activity, it receives user inputs and system events, then sent corresponding commands to views or sampling and analyzing procedures.

AnalyzerViews.java is the view in MVC. It is used to manage (initialization, display, refresh) UI texts, buttons, dialogs and graphics. *AnalyzerGraphic.java* is a main sub-view which manage display of spectrum(*SpectrumPlot.java*) and spectrogram(*SpectrogramPlot.java*).

SamplingLoop.java is more or less the "model" part. It performs the sampling and FFT analysis, and inform the graphics update.

Processing of audio samples

The data process loop is located in `run()` in *SamplingLoop.java* (after commit c9e430b (Feb 06, 2017), but basicly this process didn't change since the initial commit), as well as the trigger of graphics refresh.

In every loop of `while (isRunning)`, it reads a chunk of audio samples by

```
record.read(audioSamples, 0, readChunkSize);
```

, then "stream" it to `STFT.java` by

```
stft.feedData(audioSamples, numOfReadShort);
```

which calculates RMS and FFT whenever enough data is collected. The view is then informed through

```
activity.analyzerViews.update(spectrumDBcopy);
```

which ultimately calls `invalidate()` of the graphic view to request an update, then the `AnalyzerGraphic.onDraw(Canvas c)` will be called automatically.

Notes on project import for old revision (rev <= 115)

Import eclipse project to Android Studio (tested in Android Studio 1.1.0 with OpenJDK-7 v2.5.4)

- As Gradle-based projects (recommended)

    i. git clone *repo-path* audio-analyzer-for-android

    ii. Copy the standard library project "android-support-v7-appcompat" to "audio-analyzer-for-android/android-support-v7-appcompat".

iii. Modify "audio-analyzer-for-android/audioAnalyzer/project.properties", change "android.library.reference.2=../../../workspace/android-support-v7-appcompat" to "android.library.reference.2=../android-support-v7-appcompat".
iv. Click "Import project" in the welcome dialog box of Android Studio.
v. Choose the sub-directory "audio-analyzer-for-android/audioAnalyzer".
vi. Choose a name for Destination Directory. Next.
vii. (check the two "Replace ... when possible") Finish. Then import-summary.txt will be generated.

You should have a workable copy now.

- Or, as IntelliJ "classic" projects

i. git clone *repo-path* audio-analyzer-for-android
ii. Click "Import project" in the welcome dialog box of Android Studio.
iii. Choose the directory "audio-analyzer-for-android".
iv. Select "Create project from existing sources", Next.
v. Next.
vi. Uncheck the directories that end with "gen", codes there are auto generated. Next.
vii. (review libraries found) Next.
viii. (review suggested module structure) Next.
ix. (select project SDK) Next.

10. (review frameworks) Finish.

```
Now if you build the project, you will get an error "Cannot resolve symbol '@style/Theme.AppCompat'".
This theme is in "android.support.v7.appcompat", I don't know how to import these values yet. Just
choose another built-in theme will make it work. e.g. change to
android:theme="@android:style/Theme.Black".
```

Thanks

The code Audio spectrum Analyzer for Android gives me a good starting point, for learning Java and write this software (that I desired long ago).