# 目的

能从监听的录音数据中分析出哪些是“人声”

# 途径

## API

Android的AudioRecord提供了这样的API：public int read (byte[] audioData, int offsetInBytes, int sizeInBytes)  
则，我们可以监听到扬声器接收到的声音数据（最终数据会放入audioData中），接下来，你需要对这个audioData中的数据进行分析，以判定，这部分数据是否是“人”的声音

## 分析audioDate

1.你要先从audioData中拿到一个个的sample  
sample是什么：样本，也许，你还听说过samplerate这个词语，它代表的是 一段时间内样本被采集的次数，声音最终被播放成什么样子，跟sample有很大的关系  
sample占用了多少个字节：这个，你需要学习一个wav文件的header，推荐你一个：https://ccrma.stanford.edu/courses/422/projects/WaveFormat/  
    你会发现wav是个一个大杂缸，你会发现什么是PCM的wav

2.在得出了sample之后，我们将进行如下运算（假设，1个sample占用了2个字节）  
    int sample = 0;  
    float average;  
    for (int i = 0; i < audioDataLength; i += 2) {  
        sample = Math.abs(audioData[i] | (audioData[i + 1] << 8));  
        // 平均的sample  
        average += 1.0f \* sample / audioDataLength;  
    }  
有必要再说一下，java中的byte/short/int等等，就是signed的，也就意味着，audioData[i] | (audioData[i + 1] << 8)这个值有可能是负数，所以，请你不要惊奇  
而负数的存在，也是正常的，它代表了一个我们都不想去了解的专业概念  
最终，你得到了一个值average，来代替一个数组audioData

3.如果直接拿average作为条件，来判定：刚监听到的声音数据是否是人声，也是可以的，但是，你会发现：如果采用这种方式，那么局限性太大，即是：你的代码需要频繁的更改阀值，以确保在不同的device上面，能够监听到声音  
如何打破这种局限性，使用一个你不想接触的概念：分贝  
网上有很多，如何将pcm data转换为分贝的方法，下面说一下，我采用的这种（也意味着，你可以找到更准确的方法，所以，欢迎更新该文档）：  
float db = (float) Math.abs(10 \* Math.log10(average \* average / 32767 / 32767));  
非常的简单，最终，你从average得到了一个分贝值。  
你会发现整个过程：audioData -> average -> db

## 结论

你可以拿分贝来作为判定条件，我的实验结果是：无论device的操作系统是2.x还是4.x，只要位于[20,40]区间内，你都可以认为，你监听到的是人声  
也请你注意：[20,40]并不是完美的，我实验了将近10-台机器，涵盖了samsung,htc,huawei，均是正确的  
但是我也找到了：对于huawei u8800+，区间为[50,60]，所以，如果你找到了更准确的方法，欢迎更新该文档

## 项目代码

com.oooo3d.lib.animation.record.RecordPlayer.java