nuptxiaoli

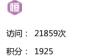
走走停停

个人资料



nuptxiaoli0518

关注 发私信



等级: 第13244名

原创: 162篇 转载: 79篇 译文: 0篇 评论: 55条

文章搜索

文章分类

Java基础 (17) JavaScript (10)

算法 (20) C++ (18)

C++Primer学习 (10)

Linux驱动开发 (2)

网络 (1)

设计模式 (12)

webrtc (11)

unix网络编程 (2)

linux基础 (22)

cocos2dx (2)

apue (3) html (4)

数据库 (8)

数据件 (0)

音视频 (2)

c基础 (16)

web (6)

搭建环境 (1)

文章存档

2016年04月 (7)

2016年03月 (8)

2016年02月 (3)

【免费公开课】Android APP开发之真机调试环境实现 有奖试读一漫话程序员面试求职、升职加薪、创业与生活

WebRTC原生音频采集

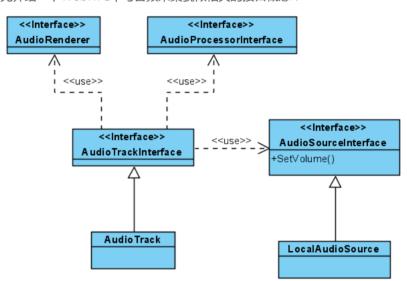
2015-05-12 22:45 170人阅读 评

1. WebRTC原生音频采集

目录(?)

先介绍一下WebRTC中与音频采集貌似相关的接口概念:

[+]



结构上看起来是不是和视频Track的结构类似?不过前面提过,如果你以对称的思维,在此结构中找的采集源和输出源,那就肯定无功而返了,LocalAudioSource对AudioSourceInterface的实现就是了音频源,那音频处理接口AudioProcessorInterface和输出接口AudioRenderer都成了无米之炊了这,可能类似于AudioCapturer的框架正在实现的途中,也可能这些接口有别的用处,比如远程音频就暂且搁置,先记下有这回事吧。这里只谈WebRTC本地音频的采集处理。前面介绍音视频接口的语频的采集由AudioDeviceModule接口统一封装:

| 2016年01月 | (13) |
|----------|------|
| 2015年12月 | (18) |

展开

阅读排行

| Java基础 Day01 Java介绍 | (395) |
|---------------------|-------|
| DFS求迷宫问题 | (335) |
| Java基础 Day14 泛型 | (307) |
| Day14 自定义泛型类的使用 | (290) |
| NetEQ 算法 | (287) |
| ORACLE 第4节 多表查询 | (283) |
| RAII手法封装互斥锁 | (273) |
| 快速排序 | (257) |
| Dijkstra算法,求单源最短路径 | (253) |
| javascript是什么? | (251) |
| | |

评论排行

| 设计模式C++实现 (7) —— | (3) |
|--------------------|-----|
| 设计模式C++实现(8)—— | (3) |
| 设计模式C++实现 (4) —— | (2) |
| 全排列 | (1) |
| Dijkstra算法,求单源最短路径 | (1) |
| 快速排序 | (1) |
| DFS求迷宫问题 | (1) |
| Dijkstra算法 | (1) |
| 全排列方法求解八皇后问题 | (1) |
| 动态规划之数字三角形 | (1) |
| | |

推荐文章

*4月28--30日:一起去北展揭开电信网络诈 骗的神秘而纱

*Binder工作机制

* Java Web基础知识之Filter:过滤一切你 不想看到的事情/a>

*Untiy Native Render Plugin在VR中的绘 制(二): 透明排序

*随机过程--Metropolis-Hastings算法

*Fresco图片库研读分析

最新评论

JavaScript 对象

Dre--amer: 很实用的网页技术之一

动态规划之数字三角形

Dre--amer : 看来楼主比较喜欢算法

Dre--amer : 来点文字描述就好了

快速排序

Dre--amer: 算法的基础,不错

Dijkstra算法,求单源最短路径(包含路径) Dre--amer:运筹学中接触过最短路径这些

概念

Dijkstra算法

Dre--amer:以前好像学过这个算法

BFS求解迷宫问题

Dre--amer: 很高深的样子, 学习了

全排列

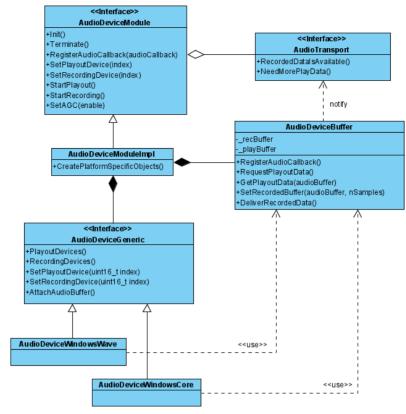
Dre--amer:数之间的排序,很有用,赞-

全排列方法求解八皇后问题

Dre--amer: 篇法中的经典案例

DFS求迷宫问题

Dre--amer:感谢楼主分享,了解了



AudioDeviceModule是个大而全的接口,恨不得将所有音频相关的接口都封装在里面(实际也差不 枚举音频采集设备(Record)和播放设备(Playout)、设置当前的采集设备/播放设备、开始/停山 设置音频增益控制开关(AGC)等。AudioTransport是个关键的对外接口,负责音频数据的传入(NeedMorePlayData方法,供Playout使用)和輸出(调用RecordedDataIsAvailable方法,数据的 生)。

AudioDeviceModuleImpl实现了AudioDeviceModule接口,创建的时候调用CreatePlatformSpe 建平台相关的AudioDeviceGeneric接口实现。该接口抽象了音频的采集和播放逻辑,在Windows³ 案:

- AudioDeviceWindowsWave实现的是传统的Windows Wave APIs方案。
- AudioDeviceWindowsCore实现的是Vista之后才支持的Windows Core Audio APIs方案。

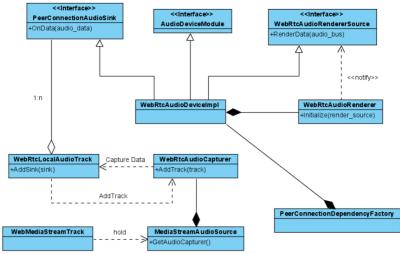
此外,AudioDeviceModuleImpl还维护了一个AudioDeviceBuffer对象来管理音频数据的缓冲区, 口AudioTransport交互。比如:

- 当AudioDeviceWindowsWave或者AudioDeviceWindowsCore需要播放音频数据的时候, AudioDeviceBuffer的RequestPlayoutData方法请求播放数据,然后通过GetPlayoutData方 数据。AudioDeviceBuffer的RequestPlayoutData就是调用AudioTransport接口的NeedMo 求待播放的音频流数据。
- 当AudioDeviceWindowsWave或者AudioDeviceWindowsCore采集到音频数据后,会调用 的SetRecordedBuffer方法将采集到的音频数据传递进去,然后调用DeliverRecordedData方 发方法就是通过调用AudioTransport接口的RecordedDataIsAvailable来实现。

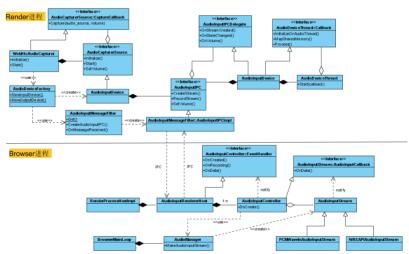
总之,音频采集模块处处都透露出大而全的结构设计。如果可以,真的应该细化一下概念设计,比如 播放逻辑分离、音频输入和输出的接口拆分等等,那样才能谈得上结构设计。

2. Chromium对WebRTC的音频采集适配

根据WebRTC的本地音频接口设计, Chromium提供了一个WebRtcAudioDeviceImpl类来实现Au 接口,该类对象由PeerConnectionDependencyFactory负责创建和维护,结构如下:



如图所示,WebRtcAudioDeviceImpl摒弃了原生的AudioDeviceModuleImpl实现中大而全的设计和音频渲染逻辑分开,分别对应于WebRtcAudioCapturer和WebRtcAudioRenderer。WebRtcAuWebRtcAudioRendererSource接口的RenderData方法向WebRtcAudioDeviceImpl请求音频流数WebRtcAudioDeviceImpl将该请求转发给前面提到的对外交互接口AudioTransport。WebRtcAu频采集逻辑,它将采集到的数据通过WebRtcLocalAudioTrack对象所持有的PeerConnectionAudi去,WebRtcAudioDeviceImpl正是实现了该接口来接收音频采集数据,然后也是通过AudioTrans至于WebRtcAudioCapturer对象的持有者MediaStreamAudioSource和WebMediaStreamTrack就行,它们是Chromium对HTML5媒体流的实现接口。接下来仔细分析一下WebRtcAudioCapturWebRtcAudioRenderer两个关键类,毋庸置疑,它们都涉及到了特定平台实现,而且在Chromiur和Browser进程。和介绍Chromium视频采集的模式一样,由于不是本文重点,这里只列出结构图,你有开发上的需要,可以照着该结构图细看源码。



这是WebRtcAudioCapturer采集音频数据的结构,牵涉到跨进程通信,结构还是非常复杂的。We的结构就不准备介绍了,因为Chromium的这块设计非常具备对称性,基本上图中类命名中的Input 多是WebRtcAudioRenderer的架构了。



- 上一篇 WebRTC的问题
- 下一篇 WebRTC Channel概念