趣看

IOS录播、编辑SDK手册（1.1.0）

内部资料，请勿外传

目录

1. 系统介绍及快速使用 3

1.1 项目初始配置 3

1.1 SDK的使用前言 3

1.1.1 录制视频使用流程 3

1.1.2 录制过程中的可选操作 4

1.1.2 短视频集成简介 4

1.1.2 短视频中的一些问题 4

二、IOS客户端录播接口 4

2.1 IPLocalCameraSDK接口 4

2.1.1 初始化SDK以及配置日志文件 5

2.1.2 设置视频清晰度 6

2.1.3 视频码率设置 7

2.1.4 设置帧率 8

2.1.5 开启摄像头，设置手机旋转方向 8

2.1.6 关闭摄像头 9

2.1.7 开始录播 9

2.1.8 停止录播 10

2.1.9 开启或关闭闪光灯 10

2.1.10 开启或关闭声音 10

2.1.11 切换前后摄像头 11

2.1.12 切换摄像头聚焦模式 11

2.1.13 手动聚焦 11

2.1.14 摄像机画面放大 12

2.1.15 设备是否正在录制视频 12

2.1.16设置画幅比 13

2.1.17设置防抖 13

2.1.18 修改视频倍速 14

2.1.20 检查是否支持某种分辨率 14

2.1.21 获取SDK的版本号 14

2.1.22 使用SDK进行拍照截图 15

2.1.23 使用SDK获取录制的文件名 15

三、短视频播放器，对应类 16

3.1设置播放的视频队列 16

3.2开始启动播放器 17

3.3停止播放器 17

3.4继续或暂停播放器 18

3.5查看当前播放器是否暂停 18

3.6异步调整当前播放时间 18

3.7同步调整到指定时间播放 19

3.8获取当前播放时间 19

3.9判断播放器是否已经播放到末尾了 19

3.10修改播放器的转场效果 20

3.11修改视频的播放速度（只能在单个文件编辑时候使用） 20

3.12修改视频的旋转角度 20

3.13修改原始声音大小 21

3.14修改背景声音大小 21

3.15修改配音声音大小 21

3.16设置背景音地址 22

3.17关闭背景音 22

3.18设置配音地址 22

3.19取消配音地址 23

3.20获取视频当前截图 23

3.21修改当前视频的滤镜 23

3.22修改亮度 24

3.23修改对比度 24

3.24修改饱和度 24

3.25修改饱和度 25

3.26修改曝光 25

3.27修改色调 25

四、短视频制作，合成QKGPUMovieExportSession.h 26

4.1视频制作的主要方法，包括视频制作进度监听，视频制作过程中状态回调监听，以及视频制作过程中，视频贞回调 26

4.2开始制作 27

4.3取消制作 28

4.4修改原始声音大小 28

4.5修改背景声音大小 28

4.6修改配音声音大小 29

4.7设置背景音地址 29

4.8关闭背景音 29

4.9设置配音地址 30

4.10取消配音地址 30

4.11修改亮度 30

4.12修改对比度 31

4.13修改饱和度 31

4.14修改饱和度 31

4.15修改曝光 32

4.16修改色调 32

4.17Mp3转为Pcm背景音 32

5、短视频Demo的简单介绍 33

5.1 demo中界面的介绍 33

5.2 草稿的生成参考demo中的saveDrafts方法。 33

# 系统介绍及快速使用

## 1.1 项目初始配置

* 录播sdk，负责调用设备摄像头录制指定分辨率及码率的视频，并保存到本地。
* **项目必须将在build settings中找到 Other Linker Flags ,设置为-ObjC**
* **将IPRecordCamera文件夹中的文件都都copy到项目中。**
* **QKLocalRecord.framework 是动态库， 添加至Embedded Binaries中。无需添加其他外接库**

## 1.1 SDK的使用前言

* 使用SDK前要先调用[IPLocalCameraSDK initAppkey: username:]设置Appkey,设置上传文件的用户名选填，不知道的填test即可。
* 由于一次录播时间有可能会很长，所以sdk对录像的录制进行了分片处理，目前设定为5分钟生成一个mp4的录像文件。

用户将这些MP4文件上传到阿里云服务器后，可以通知我们的服务器，对这些文件进行文件合并，最终可以在服务器上生成一个完整的视频文件。

Demo的作用就是在本地建立一个数据库，管理这些文件及上传状态。

* 视频转码功能目前支持转码的文件格式为：MP4与mov

### 1.1.1 录制视频使用流程

1、调用+(void)initSdkSaveLog:(int)saveLog;配置日志信息并且初始化SDK。

2、配置录像分片完成的接收回调+(void)getFileEndRecord…（Demo中，在设置回调之前，先生成了一个分片的统一标示名，用来标示录像文件都属于一次录播的）

2、（可选）设置logo +(void)setLocalUserImage…

3、（可选）设置分辨率+(void)setLocalCameraEncodeType:…

4、（可选）设置码率+(void)setLocalCameraVideoBitRate…

5、（可选）设置帧率+(void)setLocalVideoFrame…

6、开启摄像头+(int) startLocalCamera…

7、（可选）设置声音增益效果+(void) changeLocalGainValue…

8、启动录制+(int)startLocalRecord…

9、暂停录制+(void)stopLocalRecord…

10、关闭摄像头+(void) stopLocalCamera

### 1.1.2 录制过程中的可选操作

切换前后摄像头、设置远近焦距、设置自动聚焦或者手动聚焦、关闭或开启声音、关闭或者开启闪光灯。

### 1.1.2 短视频集成简介

1、初始化SDK，[IPLocalCameraSDK initAppkey: username:]设置Appkey,没有用户名的传@”test”即可。

2、复制IPRecordCameraSDK文件夹下的文件，以及newClip文件夹下的文件，到项目中。

3、QKLiveAndRecord.framework是动态库，需要在target-》General-》Embedded Binaries中，添加。

4、根据自身项目，修改newClip 中的部分公用配置即可。

### 1.1.2 短视频中的一些问题

1、草稿的保存和还原，QKMoviePartDrafts 是草稿文件对应的bean文件，其中dict\_save 中保存的是 短视频的所有信息，其他字端保存的是草稿信息。

SubTitlesSql，展示了如何把QKMoviePartDrafts 保存到数据库中，并且重新获取出来转化为草稿的操作。

2、短视频处理的QKClipController，可以通过2种方式创建。

通过视频列表创建，列表可以为空

-(id)init:(NSMutableArray\*)array;

通过草稿文件创建，草稿对象不能为空

-(id)initWithDrafts:(QKMoviePartDrafts\*)movieDrafts;

# 二、IOS客户端录播接口

## 2.1 IPLocalCameraSDK接口

APP KEY的设置进行了更改，支持更多方式，APPkey 与代理分开，可以单独设置。并且独立返回初始化结果。在设置过代理后，appkey设置错误的返回信息也会通过代理通知。

1.0.7.160119中，优化了视频传输，减少了视频卡顿的情况。

### 2.1.1 初始化SDK以及配置日志文件

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void)initSdkSaveLog:(int)saveLog;** |
| 参数 | **[int]saveLog:** 是否在本地形成日志文件 |
| 返回值 |  |
| 说明 | 设置本地是否以文件形式保存日志，  Macintosh HD:Users:yang:Desktop:CA99F9CF-99B2-4BAD-818E-9F878DDA645E.png |

### 2.1.2 设置视频清晰度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void) setLocalCameraEncodeType:(NSInteger)encodeType |
| 参数 | **[in]** encodeType**,**视频清晰度  Macintosh HD:Users:yang:Desktop:BE560599-D574-4018-8054-AA7287DD6EE0.png |
| 说明 | 在用户调用startLocalCameraNoEncodeType，开启摄像头前，设置视频的清晰度，如果未设置或者设置错误参数，将默认为标清 |

### 2.1.3 视频码率设置

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void) setLocalCameraVideoBitRate:(NSInteger)videobitrate; |
| 参数 | **[in]** videobitrate 设置视频码率参数  Macintosh HD:Users:yang:Desktop:21028FDF-7A81-4AF3-B4F8-7C61BF1864D2.png |
| 说明 | 在用户调用startLocalCameraNoEncodeType，开启摄像头前，设置视频的码率，如果未设置或者设置错误参数，将默认为kLocalBitRate2000 |

### 2.1.4 设置帧率

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(BOOL) setLocalVideoFrame:(int)videobitrate; |
| 参数 | **[in]** videobitrate设置视频帧率参数 int类型，参数值10-30 |
| 返回值 | **BOOL NO**设置失败 **YES**设置成功 |
| 说明 | 设置视频的帧率，默认帧率**20** |

### 2.1.5 开启摄像头，设置手机旋转方向

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(int) startLocalCamera:(UIView \*)view Orientation:(AVCaptureVideoOrientation)Orientation Camera:(NSInteger)cameraValue recordPath:(NSString\*)recordPath; |
| 参数 | **[in]view**，录播画面所在的页面，**self.view;**  **[in]Orientation** 废弃了，填**0**即可  **[in]Camera 0** 前置摄像头 **1**后置摄像头  **[in]** recordPath录像相对路径，可以传**@“record"**，表示存在**record**文件夹下 |
| 返回值 | **0:**表示开启摄像头成功  **-1:**表示开启摄像头失败 |
| 说明 | 用户开始摄像头，必须在初始化之后，及录播之前调用， |

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void)setAppOrientation:(UIInterfaceOrientation)orientation; |
| 参数 | **[in]** orientation，手机的导航条方向**;** |
| 说明 | 用户设置手机当前的导航条方向，对应的会进行横竖屏视频的采集， |

### 2.1.6 关闭摄像头

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void)stopLocalCamera; |
| 参数 |  |
| 返回值 |  |
| 说明 | 关闭摄像头必须在**stopRtmp** 之后调用 |

### 2.1.7 开始录播

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(int) startLocalRecord** |
| 参数 |  |
| 返回值 | Int : -2 ：appkey验证失败 ，-1：已经开始录播了， 0：开启成功 |
| 说明 | 该函数用来开启录播，必须要在开启摄像头后调用 |

### 2.1.8 停止录播

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void) stopLocalRecord; |
| 参数 |  |
| 返回值 |  |
| 说明 | 该函数用来停止录播 |

### 2.1.9 开启或关闭闪光灯

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(BOOL) switchLocalFlash:(int)flashFlag; |
| 参数 | [in] flashFlag ：0：关闭闪光灯1：开启闪光灯 |
| 返回值 | **YES**：关闭或开启闪光灯的操作成功 **NO**：开启或关闭闪光灯的操作失败 |
| 说明 | 在调用**+start**Local**Camera()**之后，就可以调用该函数开启或关闭闪光灯。默认未开启闪光灯 |

### 2.1.10 开启或关闭声音

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void) switch**Local**Audio :(int) audioType;** |
| 参数 | **[in] audioType: 0:**关闭音频 **1:**打开音频 |
| 返回值 |  |
| 说明 | 在调用**start**Local**Camera()**之后，就可以调用该函数开启或关闭声音。默认开启声音 |

### 2.1.11 切换前后摄像头

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(BOOL) switch**Local**ChangeCamera : (int) cameraValue;** |
| 参数 | **[in] cameraValue** ：**0:**启用前置摄像头 **1:**启用后置摄像头 |
| 返回值 | **YES**表示切换到目标摄像头成功  **NO**表示切换到目标摄像头失败 |
| 说明 | 在调用**start**Local**Camera()**之后，就可以调用该函数切换前后摄像头。默认后置摄像头 |

### 2.1.12 切换摄像头聚焦模式

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(BOOL) switch**Local**FocusMode : (int) focusType;** |
| 参数 | **[in] focusType: 0** 自动聚焦 **1** 手动聚焦 |
| 返回值 | **YES**：表示支持这种聚焦模式，并设置成功  **NO:**表示不支持 |
| 说明 | 在调用**start**Local**Camera()**之后，就可以调用该函数设置摄像头聚焦模式。默认自动聚焦 |

### 2.1.13 手动聚焦

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void) manual**Local**Focus:(CGPoint) touchPoint TouchView:(UIView \*)view;** |
| 参数 | **[in] touchPoint :** 聚焦在当前**view**的坐标  **[in] TouchView**当前的**view** |
| 返回值 |  |
| 说明 | 在调用**switch**Local**FocusMode(1)**后，切换到手动聚焦的模式，就可以调用该函数使手机的摄像头修改聚焦的焦点位置。 |

### 2.1.14 摄像机画面放大

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void) zoom**Local**Out : (float) scale** |
| 参数 | **[in] scale** 范围**1-10**， |
| 返回值 |  |
| 说明 | 该接口仅作用于**ios7.0**并且**iphone5**以上设备。  在调用**start**Local**Camera()**之后，就可以设置摄像机放大的倍数。 |

### 2.1.15 设备是否正在录制视频

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(NSInteger)cameraIsRecord;** |
| 参数 |  |
| 返回值 | Int:1 正在录制 0未录制 |
| 说明 | 设备是否正在录制视频 |

### 2.1.16设置画幅比

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void)changeRenderOutPut:(int)type;** |
| 参数 |  |
| 返回值 | Int:type : 0 ：16x9，1: 4x3， 2: 1x1  Macintosh HD:Users:yang:Desktop:8E03A7CD-6EF0-4E3F-AC83-87433482EA9A.png |
| 说明 | 设置画幅比 |

### 2.1.17设置防抖

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(void)setStabilization:(BOOL)stabilization;;** |
| 参数 |  |
| 说明 | 设置防抖非常吃机器性能，建议在iphone8以上机器开启1080P的防抖。 |

### 2.1.18 修改视频倍速

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(BOOL)changeSpeed:(double)speed;** |
| 参数 |  |
| 说明 | 修改视频倍速 |

### 2.1.20 检查是否支持某种分辨率

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(BOOL)checkEncodeType:(int)type font:(BOOL)font;** |
| 参数 |  |
| 说明 | 检查是否支持某种分辨率 |

### 2.1.21 获取SDK的版本号

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | **+(NSString\*)getVersion;** |
| 参数 |  |
| 返回值 | 返回当前**sdk**的版本号 |
| 说明 |  |

### 2.1.22 使用SDK进行拍照截图

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void) takeLocalPhoto:(getImgAddress)getaddress address:(NSString\*)address; |
| 参数 | **[**getImgAddress**]** getaddress拍照成功后的回调  **[**NSString**]** address 照片的存放路径 |
| 回调说明 | typedef void (^getImgAddress)(NSString \*img\_address,NSString \*img\_name,NSData \*img);  拍照完成后的block回调，  返回img\_address 图片完整路径，  img\_name 图片名称，  img 图片数据 |
| 说明 | 该接口用于拍照截图 |

### 2.1.23 使用SDK获取录制的文件名

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | +(void)getFileEndRecord:(getFileEndRecord)endrecord; |
| 参数 | **[**getFileEndRecord**]** endrecord某个片段录播完成后的回调 |
| 回调说明 | typedef void (^getFileEndRecord)(NSString \*filename,NSInteger filetime,NSString \*type);  结束当前录播片段的时候的回调  NSString \*filename 文件片段的名称，名称的规则是:yyyyMMddHHmmss\_第几个片段.mp4,例如：20161220141516\_2.mp4 这表示的是2016年12月20日14点15分16秒开始录制录播的，第二个片段。  Type：文件类型，目前只有Localvideo一种。 |
| 说明 | 用户拍摄一个视频可能可以拍几个小时，但是几个小时的视频文件过大，这里我们在录播过程中，每隔一段时间就会保存一个视频文件，实现文件分片。 |

# 三、短视频播放器，对应类

## 3.1设置播放的视频队列

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setLocalFiles:(NSArray \*)array;//设置播放的视频 |
| 参数 | **[**NSArray**]** array**:** 视频列表，列表中是**QKPlayerFileInfo** 对象 |
| 返回值 |  |
| 说明 | 设置视频的播放队列。不支持重复调用，如果array发生变化，请重新初始化一个新的播放器 |

QKPlayerFileInfo解析：

@property const char\* pcFilePath;

@property(copy,nonatomic)NSString \*strFilePath;

@property(nonatomic) float startTime; //视频在视频队列中的开始时间

@property(nonatomic) float endTime; //视频在视频队列中的结束时间

@property(nonatomic)float softStartTime;//视频的开始时间

@property(nonatomic) float softEndTime; //视频的结束时间

@property(nonatomic) float speed;

@property(nonatomic) int filterType;//0 原始 1怀旧 2、素描 3清新 4恋橙 5稚颜 6青柠 7film 8微风 9萤火虫

@property(nonatomic) int useOrientation;//旋转角度

@property(nonatomic) int length; //视频地址二进制的长度

@property BOOL isImage;

@property BOOL isTitle;

@property float width;

@property float height;

@property NSInteger type; //转场动画：0 没有动画 1:左侧划入 2右侧划入 3:淡出 4:闪白 5闪黑

关于时间的详细解释：

例如一个视频队列中有3个视频：

一个视频的 总时长是 20秒

我们要播放其中的10秒至15秒，那么这个QKPlayerFileInfo的startTime = 10；endTime = 15；

这个视频在视频队列中的实际时长为：softStartTime = 0；softEndTime = 5 ；

第二个视频的总时长是30秒，我们要播放其中的21秒至28秒，那么这个QKPlayerFileInfo的startTime = 21；endTime = 28；

这个视频在视频队列中的实际时长为：softStartTime = 5（上一个视频的结束时间）；softEndTime = 12（上一个视频的结束时间+自身的持续时间） ；

第三个视频的总时长是30秒，我们要播放其中的20秒至26秒，那么这个QKPlayerFileInfo的startTime = 20；endTime = 26；

这个视频在视频队列中的实际时长为：softStartTime = 12（上一个视频的结束时间）；softEndTime = 18（上一个视频的结束时间+自身的持续时间） ；

## 3.2开始启动播放器

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)startPlayerWithTime:(double)time; |
| 参数 | time：播放器的起始播放时间 |
| 返回值 |  |
| 说明 | 开始启动播放器，与stopPlayer相对应，一个播放器的生命周期只有一次startPlayer和一次stopPlayer，暂停和继续播放，通过pause：接口控制 |

## 3.3停止播放器

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)stopPlayer; |
| 参数 |  |
| 返回值 |  |
| 说明 | 停止播放器，与startPlayer相对应，一个播放器的生命周期只有一次startPlayer和一次stopPlayer，暂停和继续播放，通过pause：接口控制 |

## 3.4继续或暂停播放器

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)pause:(bool)isPause; |
| 参数 | **[**BOOL**]** pause**: YES**：暂停播放器 **NO**：继续播放 |
| 返回值 |  |
| 说明 | 暂停或者继续播放 |

## 3.5查看当前播放器是否暂停

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(bool)getPauseState; |
| 参数 |  |
| 返回值 | **[**BOOL**] YES**：暂停 **NO**：播放中 |
| 说明 |  |

## 3.6异步调整当前播放时间

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | - (void)seekToTime:(double)startTime; |
| 参数 | **[**double**]** startTime**:** 播放的时间点 |
| 返回值 |  |
| 说明 | 异步跳转到指定时间播放 |

## 3.7同步调整到指定时间播放

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | - (void)seekToTimeSync:(double)startTime; //调整时间 |
| 参数 | **[**double**]** startTime**:** 播放的时间点 |
| 返回值 |  |
| 说明 | 同步跳转到指定时间播放，会堵塞ui线程，特定时候用到 |

## 3.8获取当前播放时间

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(float)getCurrentTime; |
| 参数 |  |
| 返回值 | **[**float**]** 当前的播放时间点 |
| 说明 |  |

## 3.9判断播放器是否已经播放到末尾了

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(BOOL)isPlayerEnd; / |
| 参数 |  |
| 返回值 | **[**BOOL**]** YES：播放到末尾了，NO：没有播放到末尾 |
| 说明 |  |

## 3.10修改播放器的转场效果

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setChangeMovieTransfer:(NSArray \*)array; |
| 参数 | **[**NSArray**]** array**:** 视频列表，列表中是**QKPlayerFileInfo** 对象 |
| 返回值 |  |
| 说明 | 设置替换所有视频的转场效果 |

## 3.11修改视频的播放速度（只能在单个文件编辑时候使用）

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeSpeed:(double)speed; |
| 参数 | **[**double**]** speed**:** 播放速度，不能为**0**，最好在**0.5~2**之间 |
| 返回值 |  |
| 说明 | 设置替换当前视频的播放速度 |

## 3.12修改视频的旋转角度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeUseOrientation:(int) useOrientation; |
| 参数 | **[**int**]** useOrientation**:** 旋转角度，**0** 不旋转 **1**：**90**，**2** ：**180**，**3** ：**270** |
| 返回值 |  |
| 说明 | 修改视频的旋转角度 |

## 3.13修改原始声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setOrgSoundValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改原始声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 3.14修改背景声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setBackSoundValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改背景声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 3.15修改配音声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void) setRecordValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改配音声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 3.16设置背景音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setBackgroundAudio:(NSString\*)path; |
| 参数 | [NSString] path: 背景音地址 |
| 返回值 |  |
| 说明 | 设置背景音地址 |

## 3.17关闭背景音

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)closeBackgroundAudio; |
| 参数 |  |
| 返回值 |  |
| 说明 | 关闭背景音 |

## 3.18设置配音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setRecordAudio:(NSString\*)path; |
| 参数 | [NSString] path: 配音地址 |
| 返回值 |  |
| 说明 | 设置配音地址 |

## 3.19取消配音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)closeRecordAudio; |
| 参数 |  |
| 返回值 |  |
| 说明 | 取消配音地址 |

## 3.20获取视频当前截图

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | - (UIImage \*)thumbnailImageAtCurrentTime; |
| 参数 |  |
| 返回值 |  |
| 说明 | 获取视频当前截图 |

## 3.21修改当前视频的滤镜

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeFilter:(int)filterType; |
| 参数 | [int] filterType: 滤镜类型， |
| 返回值 |  |
| 说明 | 修改当前视频的滤镜 |

## 3.22修改亮度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeBrightness:(CGFloat)brightness; |
| 参数 | [CGFloat] brightness： -1~1，0是正常色 |
| 返回值 |  |
| 说明 | 修改亮度 |

## 3.23修改对比度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeContrast:(CGFloat)contrast; |
| 参数 | [CGFloat] contrast：0.0 to 4.0，1.0是正常色 |
| 返回值 |  |
| 说明 | 修改对比度 |

## 3.24修改饱和度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeSaturation:(CGFloat)saturation; |
| 参数 | [CGFloat] saturation：0.0 to 2.0， 1是正常色 |
| 返回值 |  |
| 说明 | 修改饱和度 |

## 3.25修改饱和度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeSharpness:(CGFloat)sharpness; |
| 参数 | [CGFloat] sharpness：-4.0 to 4.0 , 0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改锐度 |

## 3.26修改曝光

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeExposure:(CGFloat)exposure; |
| 参数 | [CGFloat] exposure：-10.0 to 10.0, with 0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改曝光 |

## 3.27修改色调

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeHue:(CGFloat)hue; |
| 参数 | [CGFloat] hue： 0 ~ 360，0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改色调 |

# 四、短视频制作，合成QKGPUMovieExportSession.h

## 4.1视频制作的主要方法，包括视频制作进度监听，视频制作过程中状态回调监听，以及视频制作过程中，视频贞回调

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(id)init:(NSString\*)v\_strSavePath progress:(GetProgress)progress state:(GetChangeState)state width:(double)width height:(double)height array:(NSArray\*)array\_movies gpuview:(GPUTransferAndSubTitleView \*)gpuview; |
| 参数 | NSString\*v\_strSavePath:导出文件的位置  GetProgress progress：进度追踪的block  GetChangeState state：视频制作中的状态回调  Double width：视频目标的宽度  Double height：视频目标的高度  NSArray\* array：需要处理的视频队列, 中包含的对象为QKPlayerFileInfo  GPUTransferAndSubTitleView gpuview：视频字幕的所属界面  参数详解：  typedef typedef void (^GetProgress)(double progress);  获取进度progress : 0.0 – 1.0 之间  typedef void (^GetChangeState)(ChangeState state,NSError \*error);  制作中的状态回调：  typedef NS\_ENUM(NSInteger,ChangeState) {  ChangeStateNomal = 0, //正常状态  ChangeStateFinish = 1, //制作完成  ChangeStateCancel = 2, //制作取消  ChangeStateError = 3, //制作出错  ChangeStateUnknown = 4， //未知错误  ChangeStateErrorAppkey = 5 //Appkey不正确  }; |
| 返回值 |  |
| 说明 | 视频制作的主要方法，包括视频制作进度监听，视频制作过程中状态回调监听，以及视频制作过程中，视频贞回调处理 |

## 4.2开始制作

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)startEncode; |
| 参数 |  |
| 返回值 |  |
| 说明 | 开始制作 |

## 4.3取消制作

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)cancelClipAvasset; |
| 参数 |  |
| 返回值 |  |
| 说明 | 取消制作 |

## 4.4修改原始声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setOrgSoundValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改原始声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 4.5修改背景声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setBackSoundValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改背景声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 4.6修改配音声音大小

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void) setRecordValue:(float)value; |
| 参数 | [float]value: 音量大小，建议设置值 0～1之间 |
| 返回值 |  |
| 说明 | 修改配音声音大小，建议设置值 0～1之间。可以超过1，超过即放大音量 |

## 4.7设置背景音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setBackgroundAudio:(NSString\*)path; |
| 参数 | [NSString] path: 背景音地址 |
| 返回值 |  |
| 说明 | 设置背景音地址 |

## 4.8关闭背景音

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)closeBackgroundAudio; |
| 参数 |  |
| 返回值 |  |
| 说明 | 关闭背景音 |

## 4.9设置配音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)setRecordAudio:(NSString\*)path; |
| 参数 | [NSString] path: 配音地址 |
| 返回值 |  |
| 说明 | 设置配音地址 |

## 4.10取消配音地址

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)closeRecordAudio; |
| 参数 |  |
| 返回值 |  |
| 说明 | 取消配音地址 |

## 4.11修改亮度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeBrightness:(CGFloat)brightness; |
| 参数 | [CGFloat] brightness： -1~1，0是正常色 |
| 返回值 |  |
| 说明 | 修改亮度 |

## 4.12修改对比度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeContrast:(CGFloat)contrast; |
| 参数 | [CGFloat] contrast：0.0 to 4.0，1.0是正常色 |
| 返回值 |  |
| 说明 | 修改对比度 |

## 4.13修改饱和度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeSaturation:(CGFloat)saturation; |
| 参数 | [CGFloat] saturation：0.0 to 2.0， 1是正常色 |
| 返回值 |  |
| 说明 | 修改饱和度 |

## 4.14修改饱和度

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeSharpness:(CGFloat)sharpness; |
| 参数 | [CGFloat] sharpness：-4.0 to 4.0 , 0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改锐度 |

## 4.15修改曝光

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeExposure:(CGFloat)exposure; |
| 参数 | [CGFloat] exposure：-10.0 to 10.0, with 0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改曝光 |

## 4.16修改色调

|  |  |
| --- | --- |
| **类型** | **描述** |
| 函数 | -(void)changeHue:(CGFloat)hue; |
| 参数 | [CGFloat] hue： 0 ~ 360，0.0 是正常色 |
| 返回值 |  |
| 说明 | 修改色调 |

## 4.17Mp3转为Pcm背景音

背景音必须是Pcm的文件。NewAudios中提供了把Mp3转为Pcm的方法

/\*

把Mp3转成PCM

\*/

-(**void**)Mp3ToPcm:(NSString\*)mp3Path pcmPath:(NSString\*)pcmPath{

dispatch\_async(dispatch\_get\_global\_queue(0, 0), ^{

OSStatus state = [QKRecordMP3ToPcm mixAudio:mp3Path toFile:pcmPath preferedSampleRate:44100];

**if**(state == noErr){

NSLog(@"setMp3Path success");

}**else**{

NSLog(@"MP3存在异常");

}

});

}

# 5、短视频Demo的简单介绍

## 5.1 demo中界面的介绍

短视频的ViewController是QKClipController类。

所有View界面的控制类

//视频列表

@property(strong,nonatomic) MoviePartCollection \*collect\_clip;

//转场动画

@property(strong,nonatomic) MovieChooseTransfer \*transfer; //转场动画的类

//视频裁剪的界面

@property(strong,nonatomic) ClipPartView \*clipMovie;

//字幕所在界面

@property(strong,nonatomic) SubTitleCollectionView \*subTitle;

//字幕动画控制界面

@property(strong,nonatomic) SubTitleShowAnimationView \*subPlayView;

//配音、背景音调节的界面

@property(strong,nonatomic) RecordControlView \*recordView;

//音乐数据管理类

@property(strong,nonatomic) RecordInfoController \*recordInfo;

//滤镜选择页面

@property(strong,nonatomic) FilterChooseView \*filterView;

//调色的页面

@property(strong,nonatomic) ChangeFilterColor \*changeColor;

每个模块对应创建方法和使用的方法，可单独使用。

## 5.2 草稿的生成参考demo中的saveDrafts方法。

最终，草稿会变成一个json字符串，保存到数据库中，需要还原的时候，可以把json重新转化成对象复原。