

短视频处理 LanSoEditor-SDK 之 MediaPool 的方法说明

说明

一下文字是《视频编辑高级版本 LanSoEditor_advance》SDK 内部的方法 MediaPool 的介绍。

- MediaPool 是 LanSoEditor_SDK 为了简化视频处理而设计的一种视频编辑架构。
- 即媒体“池子”，是一个容器，意味着您可以向这个“池子”里投放：视频、图片、文字、线条、UI 等元素。我们把这些元素命名为"Sprite"。当前有 VideoFilterSprite、VideoSprite、BitmapSprite、ViewSprite。这些 Sprite 都可以随时的投放进去，并可以随时的删除。每个 Sprite 均支持：平移、缩放、旋转、RGBA 可调效果。让你像设计 UI 动画一样设计您个性的视频 APP。
- 通俗的讲，您可以认为这是一个炒菜的锅，你放进去什么食材，就会呈现什么样的菜肴。放进去什么，就是什么,清晰明了。
- 比如,
你把一个视频投递到 MediaPool 中,则视频就会等比例自动缩放到 MediaPool 设定的大小，起到缩放的效果；
你把两个视频放到 MediaPool 中，则就是视频叠加；
你把一个视频和图片投递到 MediaPool 中，并在视频处理中，做各种平移缩放旋转的动作，就是视频和图片的各种效果；
你把一个视频+多个图片+UI 投递到 MediaPool 中，更加呈现多样的效果；
你把一个视频+UI 投递到 MediaPool 中，则就会有涂鸦，增加文字等效果；
你把一张图片投递到 MediaPool 中，则就是一种图片转换为视频；
你把多张图片投递到 MediaPool 中,并进行各种调节,就是花样的照片影集；
你把一个 UI 投递到 MediaPool 中，则 UI 上炫酷的画面就会实时的转换为视频；
- 实现 MediaPool 的可调用类有：
MediaPoolViewRender -----效果预览并实时保存处理视频画面的类。
MediaPoolPictureExecute-----后台处理各种图片转换为视频的类。
MediaPoolVideoExecute- -----后台处理视频和其他 Sprite 编辑的类。
MediaPoolVideoFilterExecute-----后台处理滤镜视频和其他 Sprite 编辑的类。
- 我们 SDK 已经应用在许多上线 APP 中，可联系我们，索取相关的上线 APP 信息。
- SDK 演示代码下载地址: https://github.com/LanSoSdk/LanSoEditor_advance

版本信息

| 日期 | SDK 版本号 | 更新说明 |
|------------|---------|-------|
| 2016-09-18 | V1.8.3 | 第一次编写 |
| | | |

方法说明

MediaPool 基本类具有的方法

1. `public BitmapSprite obtainBitmapSprite(Bitmap bmp);`

参数:

bmp 图片的 bitmap 对象;

返回值: 一个 BitmapSprite 对象

功能说明:

从 MediaPool 中获取一个 BitmapSprite 对象, 获取后,您可以使用 BitmapSprite 中的一些方法,比如是否显示,平移缩放旋转,RGBA 调节等.

您也可以一次获取多个 BitmapSprite,可以在获取的时候先设置为不显示,在你需要显示的时间点, 设置各种个性化的出现方式,如平移缓慢/或淡淡的渐亮出现/或缩放旋转着出现等等.

使用对象:

所有实现 MediaPool 的类.

2. `public ViewSprite obtainViewSprite();`

参数:

返回值: 一个 ViewSprite 对象

功能说明:

获得一个 ViewSprite, 创建后,您可以使用 ViewSprite 中的一些方法,比如是否显示,平移缩放等.

使用对象:

所有实现 MediaPool 的类.

3. `public void removeSprite(ISprite sprite);`

参数:

返回值: 一个 ViewSprite 对象

功能说明:

从 MediaPool 渲染线程列表中移除并销毁这个 Sprite,您可以在 remove 前做一些动作,比如飞出后再 remove;变暗后再 remove;淡淡的透明后再 remove 等.

使用对象:

所有实现 MediaPool 的类.

4. public void setUpdateMode(MediaPoolUpdateMode mode,int fps);

参数:

mode: 当刷新模式, 具体值参考如下.

fps: 一秒钟的刷新频率

功能说明:

设置当前 MediaPool 的刷新帧模式, 这些模式包括:

{@link MediaPoolUpdateMode#ALL_VIDEO_READY} MediaPool 中的所有视频帧都刷新后,更新 MediaPool 中一帧画面. 适用在多个视频有同样的视频帧率的情况下,当前采用这种模式 ;

{@link MediaPoolUpdateMode#FIRST_VIDEO_READY} MediaPool 维护一个列表,当列表中第一个视频帧刷新后,更新 MediaPool 中一帧画面,当前没有用到;

{@link MediaPoolUpdateMode#ANYONE_VIDEO_READY}MediaPool 中任意一个 VideoSprite 的视频帧刷新后,更新 MediaPool 中一帧画面,当前没有用到;

{@link MediaPoolUpdateMode#AUTO_FLUSH} MediaPool 自动刷新, 这种模式需要设置自动刷新的帧率.试用在 MediaPool 中没有视频的场合,比如图片影集/把 UI 独立转换为视频等场合.

使用对象:

所有实现 MediaPool 的类.

5. public void pauseUpdateMediaPool();

参数:

无

功能说明:

暂停 MediaPool 的帧刷新;

在一些场景里,您需要开启 MediaPool 后,暂停下, 然后 obtain 各种 Sprite 后,安排好各种事宜后,再让其画面更新,则用到这个方法,比如 在 MediaPool 开始后,想连续获取多个 BitmapSprite 后,再开始刷新的场合.

可以在 MediaPool 中的任意地方调用.

当 MediaPool 运行时,会判断当前是否暂停刷新帧, 如果暂停了,则不再刷新 帧, 但可以正常的 obtain 各种 Sprite 和 remove 各种 sprite.

可以用 {@link #resumeUpdateMediaPool()}来恢复 MediaPool,

使用对象:

所有实现 MediaPool 的类.

6. public void resumeUpdateMediaPool();

参数:

无

功能说明:

恢复 MediaPool 的 帧刷新过程. 和 {@link #pauseUpdateMediaPool()} 配对使用.

使用对象:

所有实现 MediaPool 的类.

7. public void setMediaPoolProgressListener(onMediaPoolProgressListener listener);

参数:

onMediaPoolProgressListener 的监听对象

功能说明:

设置 MediaPool 在执行的时的监听, MediaPool 每执行完一帧画面,会调用这个 Listener,返回的 timeUs 是当前画面的时间戳(单位是微秒),可以利用这个时间戳来做一些变化,比如在几秒处缩放,在几秒处平移等

您可以参考我们 SDK 中的大多数例子来做操作.

使用对象:

所有实现 MediaPool 的类.

8. `public void setMediaPoolCompletedListener(onMediaPoolCompletedListener listener);`

参数:

onMediaPoolCompletedListener 的监听对象

功能说明:

设置 MediaPool 处理完后的监听.

使用对象:

所有实现 MediaPool 的类.

9. `public void startMediaPool();`

参数:

无

功能说明:

开始 MediaPool 的执行.

使用对象:

所有实现 MediaPool 的类.

10. `public void releaseMediaPool();`

参数:

无

功能说明:

结束 MediaPool 的执行.

使用对象:

所有实现 MediaPool 的类.

MediaPoolViewRender 具有的方法

11. `public MediaPoolViewRender(Context context,int glwidth, int glheight);`

参数:

context: Android 系统的 Context 对象

glwidth MediaPool 的宽度

glheight MediaPool 的高度.

功能说明:

构造方法, `MediaPoolViewRender` 用在视频预览和后台统一处理的场合, 用户需要预览着各种效果, 一边设置的场合.

使用对象:

`MediaPoolViewRender`

12. `public void setDisplaySurface(Surface surface);`

参数:

`surface` : `Surface` 类型的对象.

功能说明:

设置预览显示的 `Surface`, 这个 `surface` 一般来自 `TextureView`, 当前您能也可以根据自己的需求来设置.

可以参考我们的开放源代码文件 `MediaPoolView` 中的设置.

如果您不需要 UI 界面, 但需要播放视频或音频, 则可以不设置该 `surface`, 但 `MediaPool` 的工作条件是, `surface` 和 `setEncoderEnable` 一定要有一个设置的.

使用对象:

`MediaPoolViewRender`

13. `public void setEncoderEnable(int encW,int encH,int encBr,int encFr,String encOut);`

参数:

`encW` 录制视频的宽度, 如果之前设置的 `glWidth` 和 `glheight` 于这个宽高不同, 则会在录制的过程中缩放到编码的宽高

`encH` 录制视频的高度

`encBr` 录制视频的 `bitrate`

`encFr` 录制视频的 帧率, 如有主视频, 则用主视频的刷新频率.

`encOut` 录制视频的输出路径

功能说明:

设置在预览的过程中, 同时开始录制, 这样画面中的任何变化, 都会实时的保存起来, 做到所见既所得的方式.

使用对象:

`MediaPoolViewRender`

14. `public void setUseMainVideoPts(boolean use);`

参数:

`use` : 布尔类型, 是否使用. 默认为 `false`

功能说明:

设置是否使用主视频的时间戳为录制视频的时间戳, 默认第一次获取到的 `VideoSprite` 或 `FilterSprite` 为主视频.

如果您传递过来的是一个完整的视频, 只是需要在此视频上做一些操作, 操作完成后, 时长等于源视频的时长, 则建议使用主视频的时间戳, 如果视频是从中间截取一般开始的则不建议使用

默认是这里为 `false`;

使用对象:

`MediaPoolViewRender`

15. public VideoSprite obtainVideoSprite(int width, int height);

参数:

width 视频的画面宽度 建议用 {@link MediaInfo#vWidth}来赋值

height 视频的画面高度, 此高度和宽度是即将填入到获取到 VideoSprite 中的视频宽高.

返回值: 返回创建好的 VideoSprite 对象.

功能说明:

获取一个视频的 VideoSprite 使用完毕后,通过{@link #removeSprite(ISprite)}来注销掉.

使用对象:

MediaPoolViewRender

16. public VideoFilterSprite obtainVideoFilterSprite(int width, int height, GPUImageFilter filter);

参数:

width 视频的画面宽度 建议用 {@link MediaInfo#vWidth}来赋值

height 视频的画面高度, 此高度和宽度是即将填入到获取到 VideoSprite 中的视频宽高.

filter 当前滤镜效果对象.

返回值: 返回创建好的 VideoFilterSprite 对象.

功能说明:

从 MediaPoolViewRender 中获取一个滤镜视频的 Sprite 对象.

使用对象:

MediaPoolViewRender

17. public VideoSprite obtainMainVideoSprite(int width, int height);

参数:

width 视频的画面宽度 建议用 {@link MediaInfo#vWidth}来赋值

height 视频的画面高度, 此高度和宽度是即将填入到获取到 VideoSprite 中的视频宽高.

返回值: 返回创建好的 VideoSprite 对象.

功能说明:

获取一个主视频的 VideoSprite, 如 MediaPool 用在对已经存在的视频做处理,则需要先在 MediaPool 建立后第一个调用.

当前渲染线程是通过主视频来显示频率来刷新的. 使用完毕后,通过{@link #removeSprite(ISprite)}来注销掉.

使用对象:

MediaPoolViewRender

18. public boolean switchFilterTo(VideoFilterSprite sprite, GPUImageFilter filter);

参数:

sprite 使用到的视频滤镜 Sprite,

filter 需要切换到的 filter 对象. 返回值: 返回创建好的 VideoSprite 对象.

返回值: 切换成功,返回 true,失败返回 false

功能说明:

切换从 MediaPool 对象中获取到 VideoFilterSprite 对象的滤镜效果.支持在视频预览中实时的切换.

使用对象:

MediaPoolViewRender

MediaPoolPictureExecute 后台执行方法说明

19. public MediaPoolPictureExecute(Context ctx,int glwidth,int glheight,int duration,int framerate,int bitrate,String dstPath) ;

参数:

ctx 语境,android 的 Context

glwidth MediaPool 的的宽度

glheight MediaPool 的的高度

duration 视频时长

framerate 帧率

bitrate 编码视频所希望的码率,比特率,设置的越大,则文件越大, 设置小一些会起到视频压缩的效果.

dstPath 编码视频保存的路径.

功能说明:

MediaPool 的图片转换为视频的后台执行.

使用对象:

MediaPoolPictureExecute

MediaPoolVideoExecute 后台执行方法说明

20. public MediaPoolVideoExecute(Context ctx,String srcPath,int glwidth,int glheight,int framerate,int bitrate,String dstPath);

参数:

ctx 语境,android 的 Context

srcPath 主视频的路径

glwidth MediaPool 的的宽度

glheight MediaPool 的的高度

bitrate 编码视频所希望的码率,比特率.

dstPath 编码视频保存的路径.

功能说明:

创建在后台调用 MediaPool 来处理视频的构造方法.

使用对象:

MediaPoolVideoExecute

21. public void setUseMainVideoPts(boolean use);

参数:

use :布尔类型, 是否使用.默认为 false

功能说明:

设置是否使用主视频的时间戳为录制视频的时间戳, 如果您传递过来的是一个完整的视频, 只是需要在此视频上做一些操作, 操作完成后,时长等于源视频的时长, 则建议使用主视频的时间戳, 如果视频是从中间截取一般开始的则不建议使用

默认是这里为 false;

使用对象:

MediaPoolVideoExecute 和 MediaPoolVideoFilterExecute

22. public VideoSprite obtainVideoSprite(String videoPath,int vWidth,int vHeight);

参数:

videoPath 叠加视频的完整路径

vWidth 该视频的宽度

vHeight 该视频的高度.

返回值: 从 MediaPool 中获得的 VideoSprite 对象.

功能说明:

获取一个 VideoSprite 并返回, 在 MediaPoolVideoExecute 中获取,则等于是在主视频中另外叠加上一个视频.

使用对象:

MediaPoolVideoExecute 和 MediaPoolVideoFilterExecute

23. public VideoFilterSprite obtainVideoFilterSprite(String videoPath,int vWidth,int vHeight,GPUImageFilter filter);

参数:

videoPath 叠加视频的完整路径

vWidth 叠加视频的完整路径

vHeight 该视频的高度.

filter 对即将获取到的 VideoFilterSprite 设置的滤镜对象.

返回值:获取到的 VideoFilterSprite 对象.

功能说明:

获取一个 VideoFilterSprite 并返回, 在 MediaPoolVideoExecute 中获取,则等于是在主视频中另外叠加上一个视频,并可以设置其滤镜效果..

使用对象:

MediaPoolVideoExecute 和 MediaPoolVideoFilterExecute

MediaPoolVideoFilterExecute 后台执行方法说明

24. public MediaPoolVideoFilterExecute(Context ctx,String srcPath,int mpoolW,int mpoolH,int bitrate,String dstPath,GPUImageFilter filter);

参数:

vctx 语境,android 的 Context
srcPath 主视频的路径
mpoolW MediaPool 的宽度.
mpoolH MediaPool 的高度.
bitrate 编码视频所希望的码率,比特率.
dstPath 编码视频保存的路径.
filter 需要的滤镜效果对象

功能说明:

创建在后台调用 MediaPool 来处理视频的构造方法.

使用对象:

MediaPoolVideoFilterExecute

注意: MediaPoolVideoFilterExecute 中的多个方法和 **MediaPoolVideoExecute** 一样,这里不再赘述,请参照以上方法.