

## 短视频处理 LanSoEditor-SDK 之 DrawPad 画板的方法说明

### 说明

一下文字是《视频编辑高级版本 LanSoEditor\_advance》SDK 内部的方法 DrawPad 画板的介绍。

- DrawPad 画板是 LanSoEditor\_SDK 为了简化视频处理而设计的一种视频编辑架构。
- 即媒体画板, 是一个容器, 意味着您可以向这个画板里绘制: 视频、图片、文字、线条、UI 等元素。我们把这些元素命名为 Layer 图层。当前有 VideoLayer、BitmapLayer、ViewLayer、CanvasLayer、MVLayr、CameraLayer、DataLayer。这些 Layer 都可以随时的投放进去, 并可以随时的删除。每个 Layer 均支持: 平移、缩放、旋转、RGBA 可调、滤镜等效果。让你像设计 UI 动画一样设计您个性的视频 APP。
- 比如,
  - 你把一个视频增加到 DrawPad 画板中, 则视频就会等比例自动缩放到 DrawPad 画板设定的大小, 起到缩放的效果;
  - 你把两个视频增加到 DrawPad 画板中, 则就是视频叠加;
  - 你把一个视频和图片增加到 DrawPad 画板中, 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 增加到 DrawPad 画板中, 更加呈现多样的效果;
  - 你把一个视频+UI 增加到 DrawPad 画板中, 则就会有涂鸦, 增加文字等效果;
  - 你把一张图片增加到 DrawPad 画板中, 则就是一种图片转换为视频;
  - 你把多张图片增加到 DrawPad 画板中, 并进行各种调节, 就是花样的照片影集;
  - 你把一个 UI 增加到 DrawPad 画板中, 则 UI 上炫酷的画面就会实时的转换为视频;
- 实现 DrawPad 画板的可调用类有:
  - DrawPadViewRender -----效果预览并实时保存处理视频画面的类。
  - DrawPadPictureExecute-----后台处理各种图片转换为视频的类。
  - DrawPadVideoExecute- -----后台处理视频和其他 Layer 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](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
2016-11-15	V1.9.0	全新的类:画板和图层,更易理解。
2017-02-21	V2.0.0	增加 CameraLayer 和 DataLayer, 增加一些方法说明。

## 方法说明

### DrawPad 画板基本类具有的方法

1. `public BitmapLayer addBitmapLayer(Bitmap bmp);`

**参数:**

bmp 图片的 bitmap 对象;

返回值: 一个 BitmapLayer 对象

**功能说明:**

从 DrawPad 画板中获取一个 BitmapLayer 对象, 获取后,您可以使用 BitmapLayer 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 BitmapLayer,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 DrawPad 画板的类.

2. `public ViewLayer addViewLayer();`

**参数:**

返回值: 一个 ViewLayer 对象

**功能说明:**

获得一个 ViewLayer, 创建后,您可以使用 ViewLayer 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 DrawPad 画板的类.

3. `public void removeLayer(Layer layer);`

**参数:**

返回值: 一个 ViewLayer 对象

**功能说明:**

从 DrawPad 画板渲染线程列表中移除并销毁这个 Layer,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 DrawPad 画板的类.

4. `public void setUpdateMode(DrawPadUpdateMode 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 DrawPad 画板的刷新帧模式, 这些模式包括:

{@link DrawPadUpdateMode#ALL\_VIDEO\_READY} DrawPad 画板中的所有视频帧都刷新后,更新 DrawPad 画板中一帧画面. 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 DrawPadUpdateMode#AUTO\_FLUSH} DrawPad 画板自动刷新, 这种模式需要设置自动刷新的帧率.试用在 DrawPad 画板中没有视频的场合,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 DrawPad 画板的类.

5. `public void pauseUpdateDrawPad 画板();`

**参数:**

无

**功能说明:**

暂停 DrawPad 画板的帧刷新;

在一些场景里,您需要开启 DrawPad 画板后,暂停下, 然后 add 各种 Layer 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 DrawPad 画板开始后,想连续获取多个 BitmapLayer 后,再开始刷新的场合.

可以在 DrawPad 画板中的任意地方调用.

当 DrawPad 画板运行时,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 add 各种 Layer 和 remove 各种 Layer.

可以用 {@link #resumeUpdateDrawPad()}来恢复 DrawPad 画板,

**使用对象:**

所有实现 DrawPad 画板的类.

6. `public void resumeUpdateDrawPad();`

**参数:**

无

**功能说明:**

恢复 DrawPad 画板的 帧刷新过程. 和 {@link #pauseUpdateDrawPad()} 配对使用.

**使用对象:**

所有实现 DrawPad 画板的类.

7. `public void setDrawPadProgressListener(onDrawPadProgressListener listener);`

**参数:**

onDrawPadProgressListener 的监听对象

**功能说明:**

设置 DrawPad 画板在执行的时的监听, DrawPad 画板每执行完一帧画面,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 DrawPad 画板的类.

8. `public void setDrawPadCompletedListener(onDrawPadCompletedListener listener);`

**参数:**

onDrawPadCompletedListener 的监听对象

**功能说明:**

设置 DrawPad 处理完后的监听.

**使用对象:**

所有实现 DrawPad 画板的类.

9. `public void startDrawPad();`

**参数:**

无

**功能说明:**

开始 DrawPad 画板的执行.

**使用对象:**

所有实现 DrawPad 画板的类.

10. `public void releaseDrawPad();`

**参数:**

无

**功能说明:**

结束 DrawPad 画板的执行.

**使用对象:**

所有实现 DrawPad 画板的类.

## DrawPadViewRender 具有的方法

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

**参数:**

context: Android 系统的 Context 对象

glwidth DrawPad 画板的宽度

glheight DrawPad 画板的高度.

**功能说明:**

构造方法, DrawPadViewRender 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:**

DrawPadViewRender

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 DrawPadView 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 DrawPad 画板的工作条件是,surface 和 setEncoderEnable 一定要有一个设置.

**使用对象:**

DrawPadViewRender

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:**

DrawPadViewRender

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 VideoLayer 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:**

DrawPadViewRender

## 15. public VideoLayer addVideoLayer(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 VideoLayer 中的视频宽高.

返回值: 返回创建好的 VideoLayer 对象.

**功能说明:**

获取一个视频的 VideoLayer 使用完毕后,通过{@link #removeLayer(Layer)}来注销掉.

**使用对象:**

DrawPadViewRender

## 16. public VideoLayer addMainVideoLayer(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 VideoLayer 中的视频宽高.

返回值: 返回创建好的 VideoLayer 对象.

**功能说明:**

获取一个主视频的 VideoLayer, 如 DrawPad 用在对已经存在的视频做处理,则需要 在 DrawPad 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 #removeLayer(Layer)}来注销掉.

**使用对象:**

DrawPadViewRender

## 17. public boolean switchFilterTo(Layer Layer, GPUImageFilter filter);

**参数:**

Layer 使用到的视频滤镜 Layer,

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

返回值: 切换成功,返回 true,失败返回 false

**功能说明:**

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

**使用对象:**

DrawPadViewRender

## DrawPadPictureExecute 后台执行方法说明

## 18. public DrawPadPictureExecute(Context ctx,int glwidth,int glheight,int duration,int framerate,int bitrate,String dstPath);

**参数:**

ctx 语境,android 的 Context  
glwidth DrawPad 画板的宽度  
glheight DrawPad 画板的高度  
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:**

DrawPad 画板的图片转换为视频的后台执行.

**使用对象:**

**DrawPadPictureExecute**

## DrawPadVideoExecute 后台执行方法说明

19. public DrawPadVideoExecute(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 DrawPad 画板的宽度  
glheight DrawPad 画板的高度  
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 DrawPad 来处理视频的构造方法.

**使用对象:**

**DrawPadVideoExecute**

20. 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:**

**DrawPadVideoExecute**

21. `public VideoLayer addVideoLayer(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 `DrawPad` 中获得的 `VideoLayer` 对象.

**功能说明:**

获取一个 `VideoLayer` 并返回, 在 `DrawPadVideoExecute` 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:**

`DrawPadVideoExecute`