

短视频处理 LanSoEditor-SDK 之 Layer 图层的方法说明

说明

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

Layer 是一个抽象类, 用来实现所有图层的公共方法。**DrawPad** 是一个画板, **Layer** 是向这个画板内绘制的各种图层 **Layer**, 目前继承它的有:

1. **BitmapLayer**: 用来实现对图片的各种操作, 从 DrawPad 中获取, 在 DrawPad 中释放, 可以同时获取多个 Bitmap, 然后做照片影集, 视频贴纸, 增加背景, 也可以单张图片, 把一张图片转换为视频等等。
2. **VideoLayer**: 用来实现对视频的各种操作. 从 DrawPad 中获取, 使用完毕后, 在 DrawPad 中释放, 可以从 DrawPad 中获取一个, 也可以获取多个, 从而实现对单个视频或多个视频操作, 每个视频也可以在处理中增加平移, 缩放, 旋转, RGBA 调节(渐暗或渐亮)等, 如果源视频的宽高比例和 DrawPad 设置的宽高不成比例, 则会自动缩放源视频的画面比例, 来用来实现视频叠加, 视频压缩, 视频转换等。
3. **ViewLayer**: 用来实现把 UI 界面上的各种信息绘制到 DrawPad 中, 从 DrawPad 中获取, 使用完后, 在 DrawPad 中释放, 主要有两个方法. `onDrawViewBegin()` 和 `onDrawViewEnd()`. 我们在 Demo 中做了很多的举例, 比如视频上涂鸦、视频上增加文字、视频上增加滚动图片、视频上 3D 动画等等。也可以直接把 UI 录制成视频, 我们的举例《浪漫情诗》。您可以自由的发挥。
4. **CanvasLayer**: 因 Android 的 View 机制只能在 UI 线程中创建 UI 界面, 但 Canvas 可以在其他线程绘制的特点, 我们把 Android 的 Canvas 类绘制功能封装成一个图层, 以方便您在后台执行各种绘制文字等操作。
5. **CameraLayer**: 摄像头图层, 用来把手机的摄像机当成一个图层, 绘制到 DrawPad 中, 增加后, 会根据您设置 DrawPad 的尺寸来选择一个合适的分辨率来显示, 您可以选择摄像机的画面是平铺到整个 Drawpad 还是用竖屏录制, 支持横屏的全屏显示。
6. **MVLayer**: 把美工做好的 MV 动画作为一个图层, 增加到 DrawPad 中。
7. **DataLayer**: 把您的 RGBA 数组类型的数据作为一个图层, 增加到 DrawPad 中, 因需求的多样性, 您有不同的图像的来源, 这里方便您自定义一个的内容, 支持实时 push 数据。

以上 Layer, 可单独投递到 DrawPad 中, 也可以任意的组合来投递, 比如同时投递图片, 投递视频, 投递 View, 然后在 DrawPad 某个进度时间点上删除某个 Layer, 都可以.

短视频 APP 就是在于个性化. 我们的 Layer 你可以任何的组合, 任意的投递/或释放/或调节. 来构建您个性化的 APP.

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, 或访问我们的网站 www.lansongtech.com

SDK 演示代码下载地址: https://github.com/LanSoSdk/LanSoEditor_advance

版本信息

日期	SDK 版本	更新说明
2016-09-14	V1.8.2	第一次编写.
2016-10-13	V1.8.5	增加说明文字
2016-11-15	V1.9.0	采用全新的类, 画板和图层
2017-02-17	V2.0.0	增加 CameraLayer, ViewLayer, DataLayer

方法说明

1. public void setScale(float scale);

参数:

scale 缩放值, 小于 1.0 为缩小, 大约 1.0 为放大

功能说明:

对 Layer 进行缩放.

如是 BitmapLayer, 则是相对图片的宽高来进行缩放.

如是 VideoLayer/ViewLayer 则是针对 Layer 的宽高进行缩放.

使用对象:

所有实现 Layer 的类.

2. public void setRotate(float angle);

参数:

angle 0---360 度

功能说明:

设置 Layer 旋转角度

0---360 度, 如果超出 360 度, 则取余计算. 相对于源图像的旋转..

使用对象:

所有实现 Layer 的类.

3. public void setPosition(float xpos, float ypos);

参数:

xpos x 轴的位置

ypos y 轴的位置, 可参考我们的 demo 实现.

功能说明:

设置该 Layer 的中心点在整个 DrawPad 中的位置,

设置过后,在下一帧画面刷新时呈现效果

注意:不是 android 坐标系的左上角位置,可通过 DrawPad 的宽度和高度来转换为 android 的系统的坐标系.

使用对象:

所有实现 Layer 的类.

4. public void setAlphaPercent(int percent);

参数:

Percent 百分比,范围从 0—100

功能说明:

设置画面中每个像素的 RGBA 中的 A 值(透明度)的百分比, 即把一个像素中的 A 值乘以这个百分比,设置过后,在下一帧画面刷新时呈现效果

在视频进行中, 来实现比如淡入淡出的效果.

使用对象:

所有实现 Layer 的类.

5. public void setRedPercent(int percent);

参数:

Percent 百分比,范围从 0--100

功能说明:

设置画面中每个像素的 RGBA 中的 R 值(透明度)的百分比, 即把一个像素中的 R 值乘以这个百分比,设置过后,在下一帧画面刷新时呈现效果

使用对象:

所有实现 Layer 的类.

6. public void setGreenPercent(int percent);

参数:

Percent 百分比,范围从 0--100

功能说明:

设置画面中每个像素的 RGBA 中的 G 值(透明度)的百分比, 即把一个像素中的 G 值乘以这个百分比,设置过后,在下一帧画面刷新时呈现效果

使用对象:

所有实现 Layer 的类.

7. public void setBluePercent(int percent);

参数:

Percent 百分比,范围从 0--100

功能说明:

设置画面中每个像素的 RGBA 中的 B 值(透明度)的百分比, 即把一个像素中的 B 值乘以这个百分比, 设置过后, 在下一帧画面刷新时呈现效果

使用对象:

所有实现 Layer 的类.

8. public void setVisibility(int visibility);

参数:

Visibility: Layer. VISIBLE 或者 Layer. INVISIBLE

功能说明:

当前的 Layer 是否显示, {@value #VISIBLE}表示显示,{@value #INVISIBLE}表示暂时不显示. 用显示和不显示,配上当前的视频的处理进度时间戳, 可以让一个 Layer 出现闪烁变化的效果.

使用对象:

所有实现 Layer 的类.

9. public int getVisibility();

参数:

返回值: Layer. VISIBLE 或者 Layer. INVISIBLE

功能说明:

获取当前 Layer 是否显示的状态, {@value #VISIBLE}表示显示,{@value #INVISIBLE}表示暂时不显示..

使用对象:

所有实现 Layer 的类.

10. public void setLogID(String flag);

参数:

flag: logID 的字符串.String 类型.

功能说明:

因为有可能同时创建多个相同类型的 Layer,这个用来为每个 Layer 设置一个 ID 号, 可以在调试打印的时候,用来区分不同的 Layer

使用对象:

所有实现 Layer 的类.

11. public String getLogID();

参数:

返回值: 用 setLogID 设置的字符串.String 类型.

功能说明:

获取当前的 ID

使用对象:

所有实现 Layer 的类.

12. public int getWidth();

参数:

当前 Layer 的宽度.int 类型.

功能说明:

获取当前 Layer 的宽度

使用对象:

所有实现 Layer 的类.

13. public int getHeight();

参数:

当前 Layer 的高度.int 类型.

功能说明:

获取当前 Layer 的高度

使用对象:

所有实现 Layer 的类.

14. public float getRotation();**参数:**

返回值:当前 Layer 的旋转角度, float 类型.

功能说明:

获取当前 Layer 的旋转角度.

使用对象:

所有实现 Layer 的类.

15. public float getPositionX();**参数:**

返回值:当前 Layer 的 X 坐标点, float 类型.

功能说明:

返回当前位置的 X 坐标值.

使用对象:

所有实现 Layer 的类.

16. public float getPositionY();**参数:**

返回值:当前 Layer 的 Y 坐标点, float 类型.

功能说明:

返回当前位置的 Y 坐标值.

使用对象:

所有实现 Layer 的类.

17. public float getScaleX();**参数:**

返回值: 当前 Layer 在 X 方向上的缩放值, float 类型.

功能说明:

获取当前 Layer 在 X 方向上的缩放值, 可参考 {@link #setScale(int)}

注意,这里不是比例,而是缩放值,是相对于源画面的缩放

使用对象:

所有实现 Layer 的类.

18. public float getScaleY();**参数:**

返回值: 当前 Layer 在 Y 方向上的缩放值, float 类型.

功能说明:

获取当前 Layer 在 Y 方向上的缩放值, 可参考 {@link #setScale(int)}

注意,这里不是比例,而是缩放值,是相对于源画面的缩放

使用对象:

所有实现 Layer 的类.

19. public void setRotate0();

参数:

无

功能说明:

以源视频为基础,设置视频画面旋转角度为 0 度, 即恢复到原视频显示

使用对象:

VideoLayer

20. public void setRotate90();

参数:

无

功能说明:

以源视频为基础,设置视频画面旋转角度为 90 度

使用对象:

VideoLayer

21. public void setRotate180();

参数:

无

功能说明:

以源视频为基础,设置视频画面旋转角度为 180 度

使用对象:

VideoLayer

22. public void setRotate270();

参数:

无

功能说明:

以源视频为基础,设置视频画面旋转角度为 270 度

使用对象:

VideoLayer

23. public boolean switchFilter(GPUImageFilter filter);

参数:

filter 需要切换的滤镜对象.

返回值: 切换成功返回 true, 失败返回 false

功能说明:

切换滤镜,您可以在画面走动中,任意时刻切换滤镜,切换后的滤镜,将在下一帧刷新时显示.

注意:切换后的 Filter 不再支持 RGBA 的调节,如要调节,可以设置滤镜为 RGB 的类型来调节

使用对象:

所有实现 Layer 的类.

24. public Canvas onDrawViewBegin();

参数:

返回值: Canvas. 我们有很多个 demo 举例,以方便您的使用.

功能说明:

获取一个 Canvas,开始绘制.

使用对象:

ViewLayer

25. public void onDrawViewEnd();

参数:

无

功能说明:

绘制结束, 我们有很多个 demo 举例, 以方便您的使用.

使用对象:

ViewLayer