

蓝松短视频 SDK--IOS 版集成步骤

感谢您使用我们的 SDK， 以下是 SDK 的集成步骤和其他说明， 供您参考。

集成步骤

1. 把 LanSongEditor.framework 拖到您工程中，弹出对话框，选中 copy；
2. 在 Build Phases 的<Link Binary With Libraries>中增加 libz.tbd 和 libbz2.tbd；
3. 在 Build Setting 的 Build Options 中的 **Enable Bitcode 选项设置为 No**；
4. 在您代码的开始处初始化我们的 SDK (比如 AppDelegate.m 的 didFinishLaunchingWithOptions 函数中):

```
[LanSongEditor initSDK:NULL];
```

5. 在需要使用我们函数和类的地方,包括头文件:

```
#import <LanSongEditorFramework/LanSongEditor.h>
```

SDK 中相关库文件说明

- 我们 SDK 当前使用了 ffmpeg 和 GPUImage, 两个开源项目, 其实 ffmpeg 为 3.0 版本.
- 因 IOS 版本的 GPUImage 过于强大, 大部分开发者会自己增加 GPUImage 中的各种滤镜, 我们 SDK 同样适用了 GPUImage 作为滤镜的实现, 为了不至于您在编译中遇到错误, 我们把 SDK 中的 GPUImage 名字修改为 LanSongXXXX, 只是英文字母的修改, GPUImage 的运行原理和每个类, 每个函数的使用基本一致, 我们尽量不改动原有的代码, 只是作为 SDK 的滤镜部分而已。

细节如下:

1, 把 GPUImage 修改为: LanSong; kGPU 修改为:kLanSong

2, 把 GPUImageOutput.h 中的几个队列封装函数, 名字去掉“ing”字母, 例如:
runSynchronouslyOnVideoProcessingQueue 修改为:
runSynchronouslyOnVideoProcessQueue

3, 一些特定类的修改

GLProgram 修改为: ShaderProgram ;

kColorConversion 修改为: kColorConvert ;

setColorConversion 修改为: setColorConvert ;

4, 其他一些单个函数的修改:

stillImageDataReleaseCallback 修改为: stillImageDataReleaseListener

dataProviderUnlockCallback 修改为: dataProviderUnlockListener

dataProviderReleaseCallback 修改为: dataProviderReleaseListener

int16WithBytes 修改为: lanSongInt16WithBytes

5, 如在编译中, 遇到别的问题, 请把完整的 Log 信息发给我们, 方法是: 点击错误[红色感叹号]: 右键选择" Reveal in Log" 然后复制所有的内容, 发送给我们, 即可.