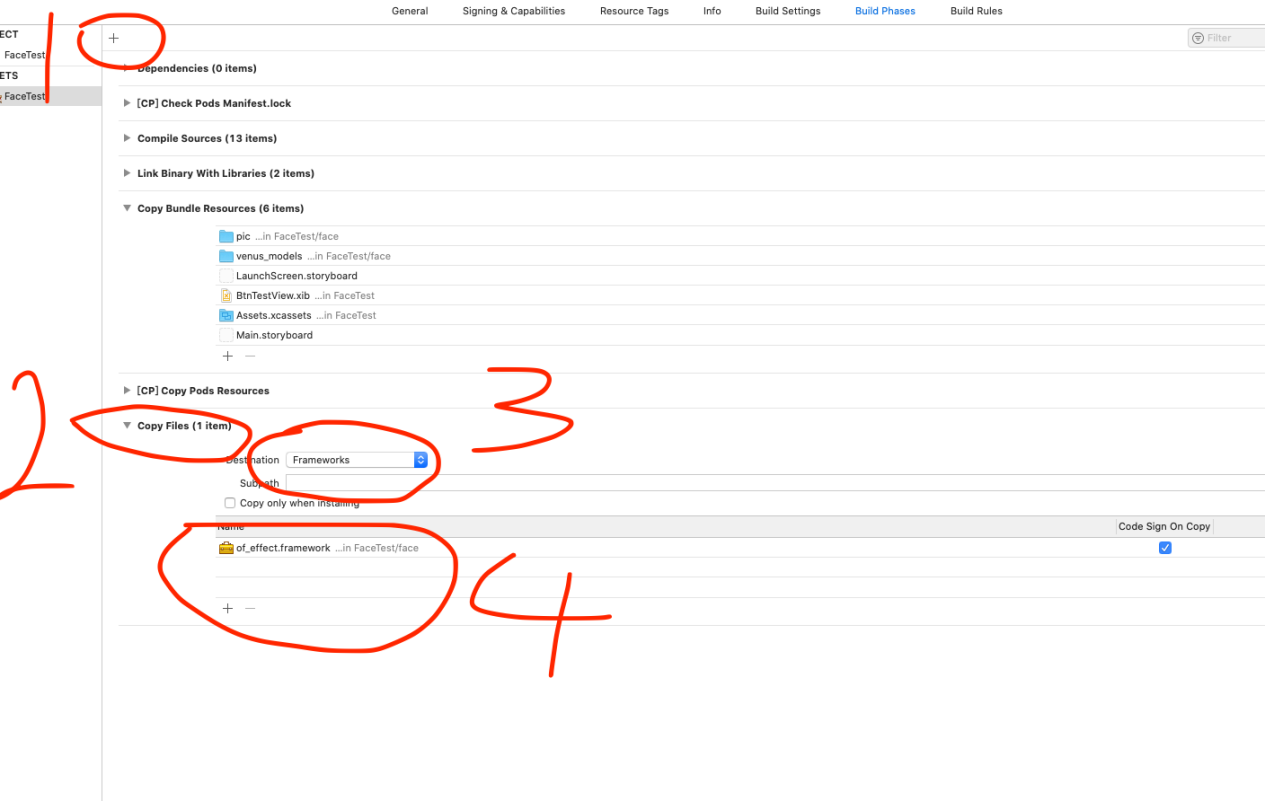
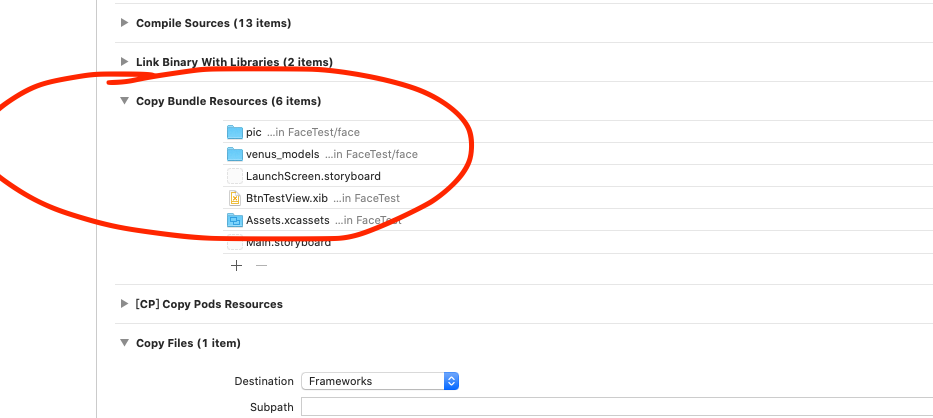
美颜 iOS 接入流程，根据 demo 工程

1. 新建工程， FaceTest demo，需要 pod install --verbose
2. 导入美颜的 framework
3. 创建一个 copy file
4. 选择 frameworks 选型
5. 点击 加号， 添加 framework
6. 导入资源库

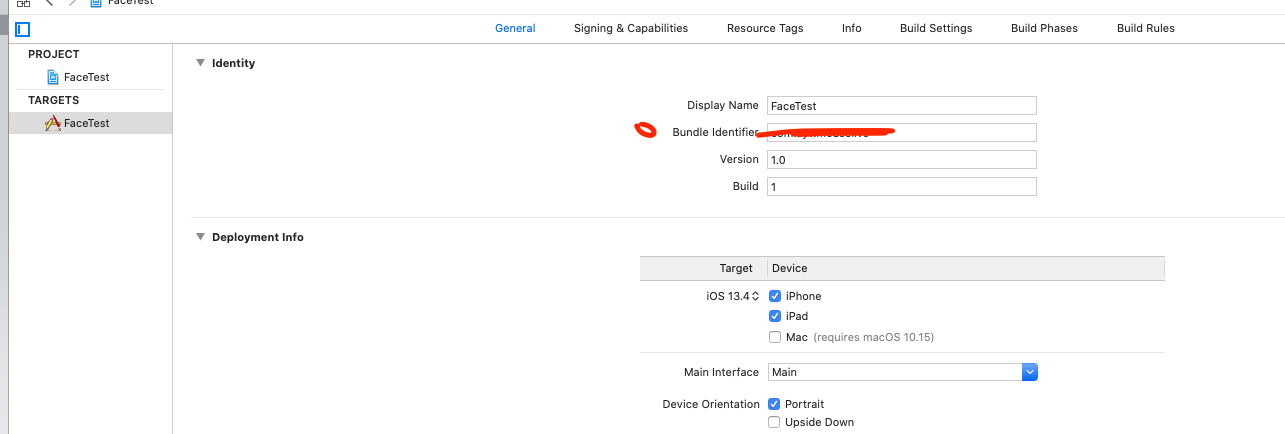


1. 如工程所示，在 SYAppId.m 下，添加 appid， token， kOFSDKSerialNumber，其中

Appid + token 是 thunder 使用

kOFSDKSerialNumber 是美颜使用

1. iOS 的 bundle id 请使用申请美颜功能的



1. 接入美颜
2. 检测字串

// 1. 检测字串， 检测申请的 kOFSDKSerialNumber 字串

[[SYEffectRender sharedRenderer] checkSDKSerailNumber:kOFSDKSerialNumber];

2） 注册摄像头回调

// 2. 注册摄像头回调

[[SYThunderManager sharedManager] registerVideoCaptureFrameObserver:self];

此出比较麻烦，registerVideoCaptureFrameObserver 在 thunder 需要延迟注册，这里先做保存，在onjoinRoomSuccess 时 才真正注册，详细情况 SYThunder 中实现，

- (void)setup 和

-(void)registerVideoCaptureFrameObserver:(nullable id<ThunderVideoCaptureFrameObserver>)delegate

这两个方法

1. 设置美颜的默认路径，beautys.zip 文件是必须要存在的，是美颜的基础模型

// 3. 设置 default 路径

[[SYEffectRender sharedRenderer] setDefaultBeautyEffectPath:[self findPathWithName:@"pic/Beauty/beautys"]];

1. 设置一些特效，filter， 手势，帽子或者猫耳朵 等特效

// 4. 要等到 10s，等待回调一次 onVideoCaptureFrame 创建出来才可以

dispatch\_after(dispatch\_time(DISPATCH\_TIME\_NOW, (int64\_t)(10.f \* NSEC\_PER\_SEC)), dispatch\_get\_main\_queue(), ^{

[[SYEffectRender sharedRenderer] loadFilterEffectWithEffectPath:[self findPathWithName:@"pic/Filter/42f10960-94c3-4b93-b5d9-ce813fb709bd"]];

[[SYEffectRender sharedRenderer] loadGestureEffectWithEffectPath:[self findPathWithName:@"pic/Gesture/gesture\_ok"]];

[[SYEffectRender sharedRenderer] loadStickerEffectWithEffectPath:[self findPathWithName:@"pic/Sticker/sticker\_aral"]];

});

其他的特效，请查看 filterArray， gestureArray， stickerArray

5） 这段代码可以不用动，是接受 摄像机的回调，并调用美颜的流程

- (ThunderVideoCaptureFrameDataType)needThunderVideoCaptureFrameDataType

{

return THUNDER\_VIDEO\_CAPTURE\_DATATYPE\_TEXTURE;

// return THUNDER\_VIDEO\_CAPTURE\_DATATYPE\_PIXELBUFFER;

}

- (CVPixelBufferRef)onVideoCaptureFrame:(EAGLContext \*)glContext PixelBuffer:(CVPixelBufferRef)pixelBuf

{

if (!pixelBuf) {

return pixelBuf;

}

CVPixelBufferRef outPixelBuf = [[SYEffectRender sharedRenderer] renderPixelBufferRef:pixelBuf context:glContext];

return outPixelBuf;

}

- (BOOL)onVideoCaptureFrame:(EAGLContext \*)context PixelBuffer:(CVPixelBufferRef)pixelBuffer SourceTextureID:(unsigned int)srcTextureID DestinationTextureID:(unsigned int)dstTextureID TextureFormat:(int)textureFormat TextureTarget:(int)textureTarget TextureWidth:(int)width TextureHeight:(int)height

{

if (pixelBuffer) {

[[SYEffectRender sharedRenderer] renderPixelBufferRef:pixelBuffer context:context sourceTextureID:srcTextureID destinationTextureID:dstTextureID textureFormat:textureFormat textureTarget:textureTarget textureWidth:width textureHeight:height];

}

return YES;

}

6） 在退出房间的时候，销毁美颜，参照 dealloc

// 2. 清除美颜

[[SYEffectRender sharedRenderer] destroyAllEffects];

以上就是根据 demo 工程，美颜的接入流程

====================================

下面是一些接入说明：

### MouseLive-iOS 接入流程：

1. 最主要的类是 util 文件夹下的类和 SYEffectProtocol 协议类，单独提取可以给业务用

2. 可以根据需要是否用 SYEffectsDataManager 下载类和模型，若不需要需要删除 SYEffectRender 类里的 595行的默认加载美颜特效方法

3. 应用启动的时候在 appdelegate 调用 checkSDKSerailNumber 校验 sn，成功之后才可用

4. thunder 需要在 thunderbolt 模式注册代理 registerVideoCaptureFrameObserver

5. 第一个代理 needThunderVideoCaptureFrameDataType 需要返回一个默认值（最好返回 THUNDER\_VIDEO\_CAPTURE\_DATATYPE\_TEXTURE，性能最好）

6. 若第一个代理返回 THUNDER\_VIDEO\_CAPTURE\_DATATYPE\_TEXTURE，需要实现方法，直接返回 YES

```

- (BOOL)onVideoCaptureFrame:(EAGLContext \*)context PixelBuffer:(CVPixelBufferRef)pixelBuffer SourceTextureID:(unsigned int)srcTextureID DestinationTextureID:(unsigned int)dstTextureID TextureFormat:(int)textureFormat TextureTarget:(int)textureTarget TextureWidth:(int)width TextureHeight:(int)height;

```

在方法实现里调用 SYEffectRender 渲染方法，会渲染目标纹理

```

[[SYEffectRender sharedRenderer] renderPixelBufferRef:pixelBuffer context:context sourceTextureID:srcTextureID destinationTextureID:dstTextureID textureFormat:textureFormat textureTarget:textureTarget textureWidth:width textureHeight:height];

```

7. 若想实现某个特效可以直接调用 SYEffectRender 里的加载方法

```

/// 加载美颜特效

/// @param effectPath 特效地址

- (void)loadBeautyEffectWithEffectPath:(NSString \*)effectPath;

/// 加载滤镜特效

/// @param effectPath 特效地址

- (void)loadFilterEffectWithEffectPath:(NSString \*)effectPath;

/// 加载贴纸特效

/// @param effectPath 特效地址

- (void)loadStickerEffectWithEffectPath:(NSString \*)effectPath;

/// 加载手势特效

/// @param effectPath 特效地址

- (void)loadGestureEffectWithEffectPath:(NSString \*)effectPath;

```

> \*\*注意：\*\*

>

> 美颜和滤镜的强度、默认、最大最小值需要业务调用方法从特效包中获取