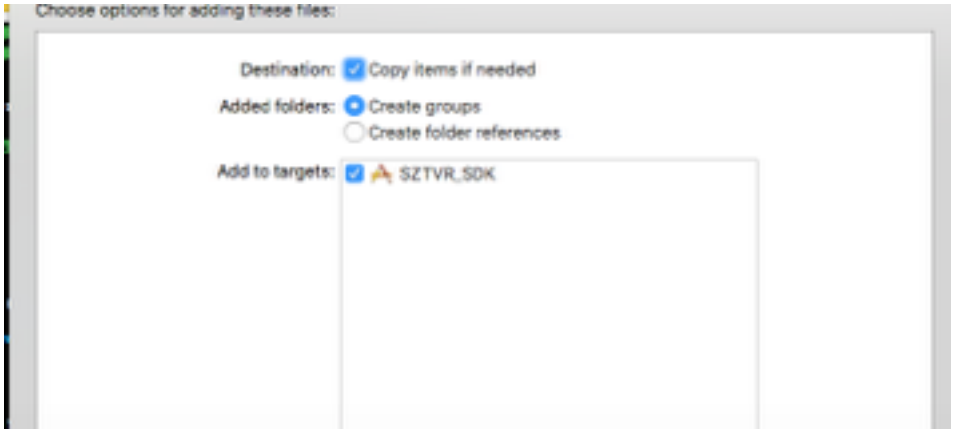
2016年12月16日 星期五

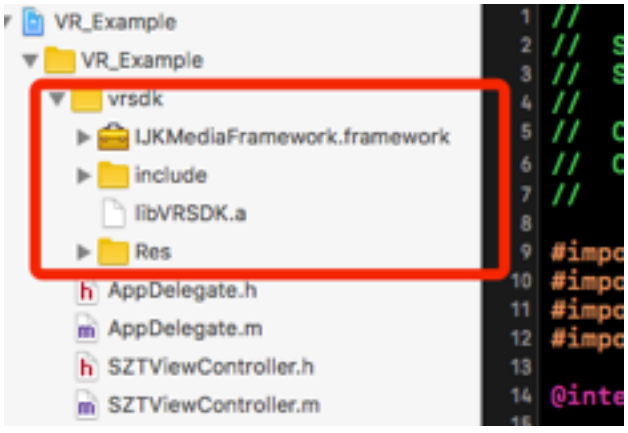
KongFu SDK for iOS

目录:

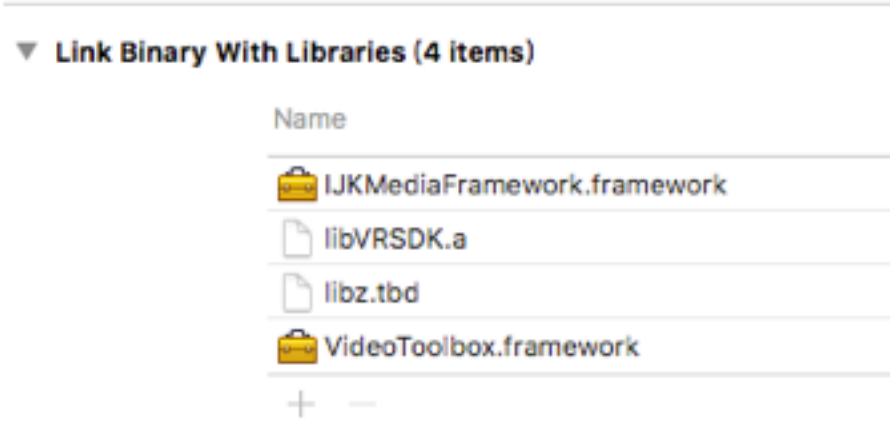
一、集成步骤

1、将vrsdk文件夹引入到到工程中，如下图：

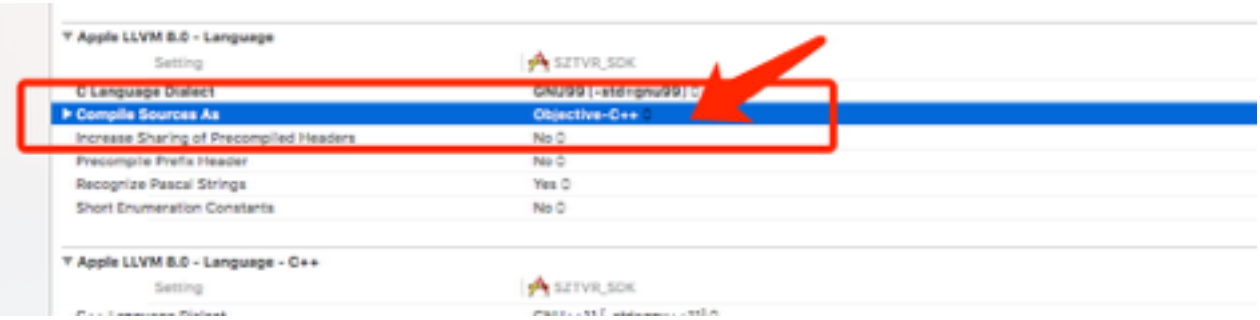




2、添加依赖库如下：



3、 Build Settings 里的Compile Sources As 改为Objectve-C++



二、SDK的主要类及其使用

2.1 SDK的入口 SZTLibrary

SZTLibrary是使用该SDK的入口，要开启SDK的相关功能，需要在相应的控制器或者View中实例化该类。

2.1.1 关键函数：

```
// 实例化sdk入口对象    — controlle or view
- (instancetype)initWithController:(UIViewController *)vc;
- (instancetype)initWithView:(UIView *)v;


// 设置单双屏幕模式 — 默认分屏
- (void)dispalyMode:(SZTModeDisplay)mode;

//设置交互模式 — 默认陀螺仪
- (void)interactiveMode:(SZTModeInteractive)mode;

// 畸变模式 - 默认无畸变
- (void)distortionMode:(SZTDistortion)mode;

// 开启焦点拾取 - 默认开启
- (void)setFocusPicking:(BOOL)isopen;

- // 开启点击拾取 - 默认关闭 (不能与焦点拾取同时存在)
- (void)setTouchEvent:(BOOL)isopen;

// 开启耳机点击事件 - 默认关闭
- (void)setEarPhoneTarget:(BOOL)isopen;

/**
 * 开启陀螺仪数据处理模式 - 默认使用gvr
 */
- (void)setSensorWithGvr:(SZTSensorMode)mode;
```

```
/**
 * 设置桶形畸变边缘黑边宽度，范围0.0~1.0 - 默认为0.9
 * 只有开启桶形畸变，设置参数才有效
 */
- (void)setValuesOfBlackEdge:(float)values;

/**
 * 设置双目间距（范围0.0 ~ 1.5）默认0.0，0.0的话没有立体效果，值越大立体效果越明显
 */
- (void)setBinocularDistance:(float)value;

/**
 * 重置屏幕 - 回到初始的方位（陀螺仪数据清除）
 */
- (void)resetScreen;

/**
 * 设置屏幕旋转方向 - 默认为右横屏 UIDeviceOrientationLandscapeRight
 * 设置错误会导致显示出错
 */
- (void)setOrientation:(UIInterfaceOrientation)orientation;

/**
 * 渲染帧数, 默认60帧
 */
@property(nonatomic, assign)int fps;

// 添加对象到场景中
-  (void)addSubObject:(SZTBaseObject *)object;

// 从场景中删除中对象
- (void)removeObject:(SZTBaseObject *)object;
```

2.1.2 使用方法：

具体设置使用方法如下图，所有的接口可动态设置，不需要提前设置好。

```
// 使用方法
- (void)loadSDK
{
    // 创建SDK
    self.sztLibrary = [[SZTLibrary alloc] initWithController:self];
    [self.sztLibrary distortionMode:SZTDistortionNormal]; // 无畸变
    [self.sztLibrary dispalyMode:SZTModeDisplayGlass]; // 分屏模式
    [self.sztLibrary interactiveMode:SZTModeInteractiveMotion]; //陀螺仪
    [self.sztLibrary setFocusPicking:YES]; // 开启热点交互
    [self.sztLibrary setEarPhoneTarget:YES]; // 开启线控模式
}
```

2.2 设置视频播放对象 - SZTVideo

SZTVideo可支持的播放器：

AVPlayer：

ijkplayer：已经支持硬解码 / 软解码。做vr视频直播需要用ijkplayer。

2.2.1 关键函数

```
/**
 * 初始化-  avplayer 视频资源/模式
 * @param url 视频地址
 * @param vrVideoMode 视频模式
 */
```

```
- (instancetype)initAVPlayerVideoWithURL:(NSURL *)url VideoMode:
(SZTRenderModel)vrVideoMode;


/**
 * 初始化-  avplayer 视频资源/模式
 * @param playitem 资源管理集
 * @param vrVideoMode 视频模式
 */
- (instancetype)initAVPlayerVideoWithPlayerItem:(AVPlayerItem
*)playerItem VideoMode:(SZTRenderModel)vrVideoMode;



/**
 * 初始化-  ijkplayer 视频资源/模式
 * @param url 视频地址
 * @param vrVideoMode 视频模式
 * @param isVideoToolBox 是否开启硬解码
 */
-  (instancetype)initIJKPlayerVideoWithURL:(NSURL *)url VideoMode:
   (SZTRenderModel)renderModel isVideoToolBox:(BOOL)key;


#pragma mark video player属性

@property(nonatomic, assign)float duration;

@property(nonatomic, assign)float currentTime;

- (void)seekToTime:(float)time;

- (void)pause;

- (void)stop;
```

- (void)play;

2.2.2 代理回调事件

SZTVideoDelegate

// 加载视频失败

- (void)errorToLoadVideo:(SZTVideo *)video;

当加载视频对象失败时，需要重新删除对象重建。

// 视频准备播放

- (void)videoIsReadyToPlay:(SZTVideo *)video;

2.2.3 使用方法

videoMode的枚举类型如下：

```
// 渲染模型
typedef NS_ENUM(NSInteger, SZTRenderModel) {
    SZTVR_2D,                                   // 2d
    SZTVR_SPHERE,                               // 全景
    SZTVR_STEREO_SPHERE_LEFT_RIGHT,             // 立体全景 - 左右
    SZTVR_STEREO_SPHERE_UP_DOWN,                // 立体全景 - 上下
    SZTVR_PLANE,                                // 平面
    SZTVR_STEREO_PLANE_LEFT_RIGHT,              // 立体平面 - 上下
    SZTVR_STEREO_PLANE_UP_DOWN,                 // 立体平面 - 左右
}
```

1、ijkplayer 创建全景视频：

```
// 用ijk播放器创建全景视频对象
NSURL *url = [NSURL URLWithString:@"http://vrkongfu.oss-cn-hangzhou.aliyuncs.com/movie/1/110.mp4"];
self.sztVideo = [[SZTVideo alloc] initIJKPlayerVideoWithURL:url VideoMode:SZTVR_SPHERE isVideoToolBox:YES];
[self.sztLibrary addSubObject:self.sztVideo];
```

2、avplayer 创建平面视频：

```
NSURL *url = [NSURL URLWithString:@"http://o9u6o2a06.bkt.clouddn.com/1208.m3u8"];
self.sztVideo = [[SZTVideo alloc] initAVPlayerVideoWithURL:url VideoMode:SZTVR_PLANE];
[self.sztVideo setObjectSize:16.0 Height:9.0]; // 设置对象的尺寸大小
[self.sztVideo setPosition:0.0 Y:0.0 Z:-10.0]; // 设置对象在空间中的坐标位置
[self.sztLibrary addSubObject:self.sztVideo];
```

2.2.4 注意事项

如果创建的对象渲染模型为平面的（包括下面介绍的SZTImageView、SZTVideo、SZTGif等）（即为：

SZTVR_PLANE,

SZTVR_STEREO_PLANE_LEFT_RIGHT,

SZTVR_STEREO_PLANE_UP_DOWN），需要设置对象坐标，否则看不到实例出来的对象.（记住，摄像机的位置在(0.0,0.0,0.0)的位置，朝向是(0,0,-1), 所有对象创建后初始位置处于(0.0,0.0,0.0)的位置，所以当设置的对象渲染模式为平面且没有设置坐标的话，摄像机和物体重合了，所以看不到渲染对象）。

2.3 设置图片对象 - SZTImageView

SZTImageView可直接支持:

本地路径地址：

网络url地址：内部做了缓存操作，占位图等。

2.3.1 关键函数

/ * 实例话对象*/

- (instancetype)initWithMode:(SZTRenderModel)renderModel;

// 设置 图片纹理 多种方法创建，具体详看代码注释

- (void)setupTextureWithImage:(UIImage *)imageName;

- (void)setupTextureWithUrl:(NSString *)fileUrl Paceholder:(UIImage *)paceholder;

- (void)setupTextureWithColor:(UIColor *)color Rect:(CGRect)frameSize;

- (void)setupTextureWithFileName:(NSString *)fileName;

- (void)setTextureWithLeftImage:(UIImage *)LeftImage RightImage:(UIImage *)rightImage;

- (void)setTextureWithLeftUrl:(NSString *)leftUrl RightUrl:(NSString *)rightUrl Paceholder_Left:(UIImage *)paceholder_left Paceholder_Right:(UIImage *)paceholder_right;

2.3.2 使用方法

1、创建全景图

```
// 创建全景图
SZTImageView *back = [[SZTImageView alloc] initWithMode:SZTVR_SPHERE];
[back setupTextureWithImage:[UIImage imageNamed:@"test0.jpg"]];
[self.sztLibrary addSubObject:back];
[back setRotate:-90.0 X:0.0 Y:1.0 Z:0.0]; // 绕Y轴旋转图片-90度
```

2、创建平面图

```
image = [[SZTImageView alloc] initWithMode:SZTVR_PLANE];
[image setupTextureWithImage:imageName];
[image setObjectSize:4.0 Height:4.0];
[self.sztLibrary addSubObject:image];
[image setPosition:0.0 Y:5.0 Z:-30.0];
```

2.4 设置Gif对象 - SZTGif

SZTGif可支持:

本地路径地址/网络url地址等；

2.4.1 关键函数

// 加载gif图片

- (void)setupGifWithFileUrl:(NSString *)fileUrl;

- (void)setupGifWithGifName:(NSString *)gifName;

- (void)setupGifWithGifPath:(NSString *)pathName;

/**

```
 * 播放重复次数
 */
@property(nonatomic, assign)int repeatTimes;

/**
 * gif播放速率
 */
@property(nonatomic, assign)float speed;
```

2.4.2 事件回调方法 block

```
/**
 * 播放完毕回调,每走完一次gif图序列回调一次
 */
- (void)gifDidFinishedCallback:
(gifDidFinishedBlockParam)block;
```

2.4.3 使用方法

```
SZTGif *sztGif = [[SZTGif alloc] init];
sztGif.repeatTimes = 1;
NSURL * left = [NSURL URLWithString:@"http://www.gifs.net/Animation11/Food_and_Drinks/Fruits/App
NSString *pngPath = [left absoluteString];
[sztGif setupGifWithFileUrl:pngPath];
[sztGif setObjectSize:8 Height:8];
[sztGif setPosition:-14.0 Y:-3.0 Z:-25];
[sztGif gifDidFinishedCallback:^(SZTGif *gif) {
    SZTLog(@"gif - 播放完毕回调");
}];
[self.SZTLibrary addSubObject:sztGif];
```

2016年12月16日 星期五

2.5 设置Label对象 - SZTLabel

2.5.1 关键函数

/**设置文字内容*/

@property(nonatomic, strong)NSString *text;

/**设置字体颜色*/

@property(nonatomic, strong)UIColor *fontColor;

/** 每一行字的数量 ， 超过数量自动换行*/

@property(nonatomic, assign)int lineNumber;

/** 是否右对齐 */

@property(nonatomic, assign)BOOL isRightAlign;

/** 是否背景不透明 */

@property(nonatomic, assign)BOOL isOpaque;

/** 清晰度 默认为2.0 值越大越清晰，效率越低*/

@property(nonatomic, assign)float definition;

2.5.1 使用方法

```
SZTLabel *label = [[SZTLabel alloc] init];
label.text = @"测试使用，测试使用，测试使用，测试使用，测试使用";
label.lineNumber = 10;
label.fontColor = [UIColor redColor];
[label setObjectSize:16.0 Height:4.5];
[label setPosition:0.0 Y:0.0 Z:-25.0];
[self.sztLibrary addSubObject:label];
```

2.6 设置3D模型对象 - SZTObjModel

2.6.1 关键函数

/ *实例化obj对象*/

- (instancetype)initWithPath:(NSString *)path;

/**设置obj模型的纹理图*/

- (void)setupTextureWithImage:(UIImage *)image;

2.6.2 使用方法

```
NSString *path = [[NSBundle mainBundle] pathForResource:@"shitou_01.obj" ofType:nil];
SZTObjModel * obj = [[SZTObjModel alloc] initWithPath:path];
[obj setupTextureWithImage:[UIImage imageNamed:@"shitou_D.jpg"]];
[self.sztLibrary addSubObject:obj];
[obj setPosition:0.0 Y:0.0 Z:-10.0];
```

2.6.3 注意事项

3d模型不需要设置objSize（设置了也无效），如果需要设置模型的大小，则只能设置纵深z值或setScale。

2016年12月16日 星期五

2.7 设置对象点击 / 热点拾取 - SZTTouch

2.7.1 关键函数

```
/** 实例化，传入要点击的对象*/
- (instancetype)initWithTouchObject:(SZTRenderObject *)obj;

/** 不需要点击后，必须销毁，否则会内存泄漏*/
- (void)destory;
```

2.7.2 回调事件

```
typedef void(^willTouch)(GLKVector3);
typedef void(^didTouch)(GLKVector3);
typedef void(^endTouch)(GLKVector3);

/ * 将要拾取对象 － 刚进入选取区域 */
- (void)willTouchCallBack:(willTouch)block;
/*进度条读完后选中对象*/
- (void)didTouchCallback:(didTouch)block;
/ * 将要离开对象*/
- (void)endTouchCallback:(endTouch)block;
```

2.7.3 使用方法

```
image = [[SZTImageView alloc] initWithMode:SZTVR_PLANE];
[image setupTextureWithImage:imageName];
[image setObjectSize:4.0 Height:4.0];
[self.sztLibrary addSubObject:image];
[image setPosition:0.0 Y:5.0 Z:-30.0];

SZTTouch *touch = [[SZTTouch alloc] initWithTouchObject:image];
[touch willTouchCallBack:^(GLKVector3 vec) {
    SZTLog(@"will touch!");
}];

[touch didTouchCallback:^(GLKVector3 vec) {
    SZTLog(@"did touch!");
}];

[touch endTouchCallback:^(GLKVector3 vec) {
    SZTLog(@"leave!");
}];
```

以上只是给图片添加了热点事件，也可以给视频，gif图，模型等添加点击，只需要传对应的对象即可。传出来的vec3为空间中的坐标比例。

2.7.4 注意事项

使用热点拾取对象，能触发block3个状态的回调；若用点击拾取对象的话，只能触发didTouchCallBack。

2.8 对象做基础的动画(MoveTo, scaleTo, RotationTo等)

2.8.1 关键函数

// 移动

- (void)moveTo:(SZTBaseObject *)object Time:(float)time PosX:(float)x posY:(float)y posZ:(float)z finishBlock:(didFinishBlock)block;

- (void)moveBy:(SZTBaseObject *)object Time:(float)time PosX:(float)x posY:(float)y posZ:(float)z finishBlock:(didFinishBlock)block;

// 缩放

- (void)scaleTo:(SZTBaseObject *)object Time:(float)time scaleX:(float)x scaleY:(float)y scaleZ:(float)z finishBlock:(didFinishBlock)block;

- (void)scaleBy:(SZTBaseObject *)object Time:(float)time scaleX:(float)x scaleY:(float)y scaleZ:(float)z finishBlock:(didFinishBlock)block;

// 旋转

- (void)rotateTo:(SZTBaseObject *)object Time:(float)time radians:(float)radians rotateX:(float)x roteateY:(float)y rotateZ:(float)z finishBlock:(didFinishBlock)block;

// 贝塞尔曲线轨迹点

// 三次

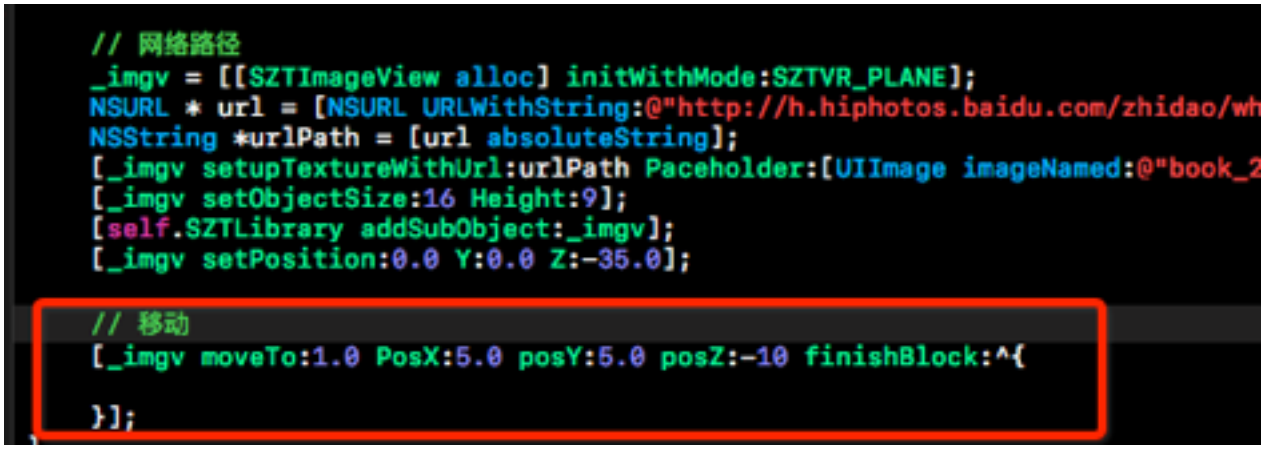
- (void)bezierTo:(SZTBaseObject *)object Time:(float)time PointEnd:(Point3D)pointEnd ControlPoint1:(Point3D)point1 ControlPoint2:(Point3D)point2 finishBlock:(didFinishBlock)block;

// 二次

- (void)bezierTo:(SZTBaseObject *)object Time:(float)time PointEnd:(Point3D)pointEnd ControlPoint1:(Point3D)point1 finishBlock:(didFinishBlock)block;

2.8.1 使用方法

1、移动方法



2、缩放方法

```
// 网络路径
_imgv = [[SZTImageView alloc] initWithMode:SZTVR_PLANE];
NSURL * url = [NSURL URLWithString:@"http://h.hiphotos.baidu.com/zhidao/wh%3
NSString *urlPath = [url absoluteString];
[_imgv setupTextureWithUrl:urlPath Paceholder:[UIImage imageNamed:@"book_256
[_imgv setObjectSize:16 Height:9];
[self.SZTLibrary addSubObject:_imgv];
[_imgv setPosition:0.0 Y:0.0 Z:-35.0];

// 缩放
[_imgv scaleTo:1.0 scaleX:0.5 scaleY:2.0 scaleZ:1.0 finishBlock:^{

}];
```

3、旋转方法

```
// 网络路径
_imgv = [[SZTImageView alloc] initWithMode:SZTVR_PLANE];
NSURL * url = [NSURL URLWithString:@"http://h.hiphotos.baidu.com/zhidao/wh%3D600%2C800/sig
NSString *urlPath = [url absoluteString];
[_imgv setupTextureWithUrl:urlPath Paceholder:[UIImage imageNamed:@"book_256px.png"]];
[_imgv setObjectSize:16 Height:9];
[self.SZTLibrary addSubObject:_imgv];
[_imgv setPosition:0.0 Y:0.0 Z:-35.0];

// 旋转
[_imgv rotateTo:1.0 radians:50 rotateX:1.0 roteateY:0.0 rotateZ:0.0 finishBlock:^{

}];
```

4、贝塞尔轨迹

2.9 修改对象的图像效果(美白，对比度，滤镜等)

```
// 网络路径
_imgv = [[SZTImageView alloc] initWithMode:SZTVR_PLANE];
NSURL * url = [NSURL URLWithString:@"http://h.hiphotos.baidu.com/zhidao/wh%3D600%2C800/sign=d8d47aee8435e5dd9879add946f6
NSString *urlPath = [url absoluteString];
[_imgv setupTextureWithUrl:urlPath Paceholder:[UIImage imageNamed:@"book_256px.png"]];
[_imgv setObjectSize:16 Height:9];
[self.SZTLibrary addSubObject:_imgv];
[_imgv setPosition:0.0 Y:0.0 Z:-35.0];

Point3D pointEnd;
pointEnd.x =-10.0;
pointEnd.y =-15.0;
pointEnd.z = -15.0;

Point3D point1;
point1.x = 8.0;
point1.y = -5.0;
point1.z = -15.0;

Point3D pointEnd1;
pointEnd1.x = 0.0;
pointEnd1.y = 0.0;
pointEnd1.z =-15.0;

[_imgv bezierTo:2.0 PointEnd:pointEnd ControlPoint1:point1 finishBlock:^{
    [_imgv bezierTo:2.0 PointEnd:pointEnd1 ControlPoint1:point1 finishBlock:^{

    }];
}];
```

2.9.1 关键函数

/** 修改滤波器*/

- (void)changeFilter:(SZTFilterMode)filterMode;

以下是一些效果的枚举类型

```
typedef NS_ENUM(NSInteger, SZTFilterMode) {
    SZTVR_NORMAL,           // 普通
    SZTVR_LUMINANCE,        // 像素色值亮度平均，图像黑白（黑白效果）
    SZTVR_PIXELATE,         // 马赛克
    SZTVR_EXPOSURE,         // 曝光（美白）
    SZTVR_DISCRETIZE,       // 离散
    SZTVR_BLUR,             // 模糊
    SZTVR_BILATERAL,        // 双边模糊
    SZTVR_HUE,              // 饱和度
    SZTVR_POLKADOT,         // 像素圆点花样
    SZTVR_GAMMA,            // 伽马线
    SZTVR_GLASSSPHERE,      // 水晶球效果
    SZTVR_CROSSHATCH,       // 法线交叉线
};
```

2016年12月16日 星期五

以下是对应效果的参数设置：

```
#pragma mark - SZTVideoFilter property
// SZTVR_PIXELATE 模式
@property(nonatomic, assign)float particles;

// SZTVR_BLUR 模式
@property(nonatomic, assign)float radius;

// SZTVR_HUE 模式
@property(nonatomic ,assign)float hueAdjust;

// SZTVR_POLKADOT 模式
@property(nonatomic ,assign)float fractionalWidthOfPix
@property(nonatomic ,assign)float aspectRatio;
@property(nonatomic ,assign)float dotScaling;

// SZTVR_CROSSHATCH 模式
@property(nonatomic, assign)float crossHatchSpacing;
@property(nonatomic, assign)float lineWidth;

// SZTVR_EXPOSURE 模式
@property(nonatomic ,assign)float exposure;

// SZTVR_GAMMA 模式 (0.0 ~ <1.0 变亮 && >1.0 变暗)
@property(nonatomic ,assign)float gamma;

// SZTVR_GLASSSPHERE 模式
@property(nonatomic ,assign)float refractiveIndex;
```

2.9.2 使用方法

```
SZTVideo *videoObj = [[SZTVideo alloc] initAVPlayerVideoWithURL:[NSURL URLWithString:@"http://vrkon
[videoObj setObjectSize:1920 Height:1080];
[videoObj setPosition:0 Y:0 Z:-30];
[self.sztLibrary addSubObject:videoObj];

// 黄白效果
[videoObj changeFilter:SZTVR_EXPOSURE];
videoObj.exposure = 0.5;

// 模糊
[videoObj changeFilter:SZTVR_BLUR];
videoObj.radius = 0.02;
```

以上的这些效果可以用在任何对象中，只需要对应的对象调用改接口就可以。

2.10 json脚本添加vr场景 - ScriptUI

2.10.1 关键函数

```
/**
 * 实例化对象，需要传入主json脚本
 */
- (instancetype)initWithJson:(NSString *)jsonPath;

/**
 * 设置脚本内部视频播放器类型 -- 默认为IJkplayer
 */
- (void)setVideoPlsyer:(VideoPLayerMode)playerMode;

/**
 * 传入热更新json脚本 -- 后续直播使用，还未实现
 */
- (void)loadHotUpdateJson:(NSString *)jsonPath;

-  (void)destory;
```

2.10.2 使用方法

```
/*** json 脚本***/
ScriptUI *script = [[ScriptUI alloc] initWithJson:@"vrkongfu.json"];
[script setVideoPlsyer:SZT_AVPlayer];
[self.sztLibrary addSubObject:script];
```

2.10.3 注意事项

如果脚本对象是全局变量的话，退出场景时，需要先调用destory方法，在置为nil，否则会照成内存泄漏。