# XEngine接入文档

## iOS (cocoapods)

1. 添加pod源：

```
    source 'https://github.com/cosmos33/MMSpecs.git'
```

2. 添加pod依赖

```
    pod 'MMXEngine', '4.4.7.20210125.1445'
```

3. 创建渲染视图，启动引擎并加载资源

```
#import <MMXEngine/XSKEngine.h>
#import <MMXEngine/XSKEngine+Lua.h>
#import <MMXEngine/XEGameView.h>
#import <MMXEngine/XSKScriptBridge.h>

@interface ViewController () <XEGameViewDelegate>

@property (nonatomic, strong) XEGameView *gameView;

@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    //创建GameView
    _gameView = [[XEGameView alloc] initWithFrame:self.view.bounds];
    [self.view addSubview:_gameView];
    //设置GameView代理
    _gameView.delegate = self;
    //启动GameView
    [_gameView start];

}

//GameView启动回调
- (void)onStart:(XSKEngine *)engine {
    //设置引擎日志开关
    [engine setLogEnable:YES];
    //添加引擎资源搜索路径
```

```
    NSString *path = [[[NSBundle mainBundle] bundlePath]
stringByAppendingString:@"/GameRes"];
    [engine addLibraryPath:path];

    [engine.scriptBridge regist:self forHandler:@"LiveGameHandler"];
    //执行游戏启动脚本
    [engine execteGameScriptFile:@"app"];
}
```

# Android

1. 在工程根目录的build.gradle中添加maven源

```
    repositories {
        maven { url 'https://cosmos1602.bintray.com/cosmos/' }//陌陌对外仓库
        maven { url 'https://cosmos1602.bintray.com/mediax/' }//陌陌对外仓库
    }
```

2. 在app的budile.gradle中添加依赖项

```
    implementation 'com.cosmos.baseutil:cpp_shared:r17c'
    implementation 'com.cosmos.mediax:xerecord:4.4.7.20210125.1445'
    implementation 'com.cosmos.mediax:xescene:4.4.7.20210125.1445'
```

3. 引擎全局初始化，在Application的onCreate方法中为引擎传入上下文对象

```
    XE3DEngine.setApplicationContext(this);
```

4. 创建渲染视图，启动引擎并加载资源

```
    private XEGameView gameView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        verifyStoragePermissions(this);
        XE3DEngine.setApplicationContext(this);
        gameView = findViewById(R.id.gameView);
        gameView.setCallback(this);
        gameView.startGame();
    }

    //渲染视图创建回调
    @Override
    public void onRenderViewCreate(View view) {
        GLSurfaceView renderView = (GLSurfaceView) view;
```

```
        //do something if needed
    }

    //引擎启动回调
    @Override
    public void onStart(XE3DEngine engine) {
        engine.addLibraryPath("/sdcard/demo");
        engine.getScriptBridge().regist(this, "LiveGameHandler");
        engine.getScriptEngine().startGameScriptFile("app");
    }

    //引擎启动失败回调
    //可能导致引擎启动失败的原因：
    //1. 引擎启动前没有设置全局的上下文对象
    //2. 如果有so打包时排出，改为运行时下载的场景。可能为so下载失败。或so版本与客户端不匹配
    @Override
    public void onStartFailed(String errorMsg) {
        Toast.makeText(this, "引擎启动失败 " + errorMsg,
Toast.LENGTH_LONG).show();

    }

    //渲染尺寸改变回调
    @Override
    public void onRenderSizeChaged(int width, int height) {

    }

    //引擎so下载进度，仅针对陌陌客户端
    @Override
    public void onEngineDynamicLinkLibraryDownloadProcess(int percent, double
speed) {

    }

    //同步Bridge方法
    public String func1(String arg) {
        return "Java同步返回";
    }

    //异步Bridge方法
    public void func2(String arg, ScriptBridge.Callback callback) {
        callback.call("Java异步回调");
    }
```

# 运行DEMO

1. iOS DEMO

进入iOS目录 执行pod update，执行成功后直接编译运行即可

2. Android

在Android目录下执行push_res.sh脚本，将资源推入手机SD卡。
使用AndroidStudio打开Android工程，直接运行即可

3. 游戏示例


点击屏幕挥棒击球
500m
900m
200m