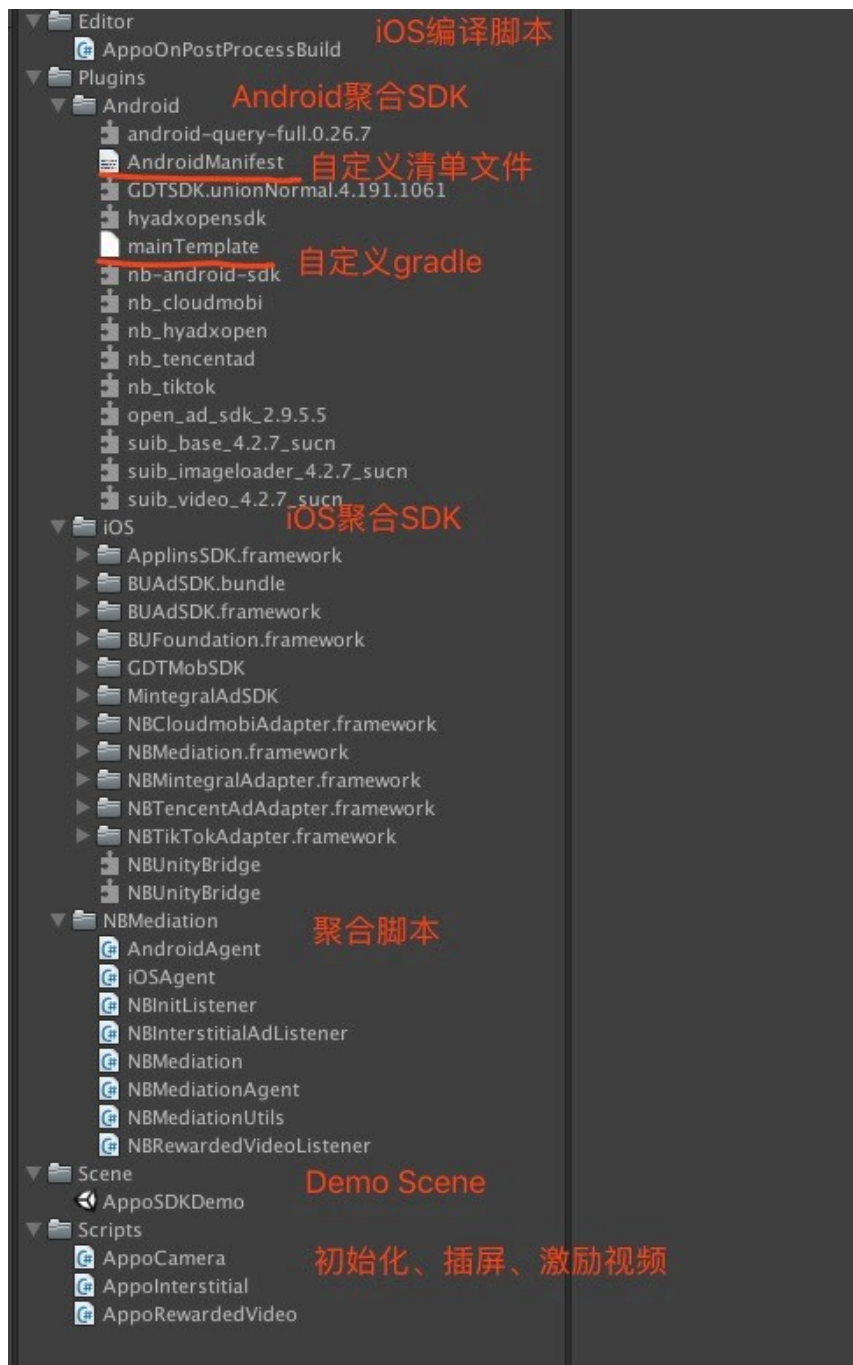
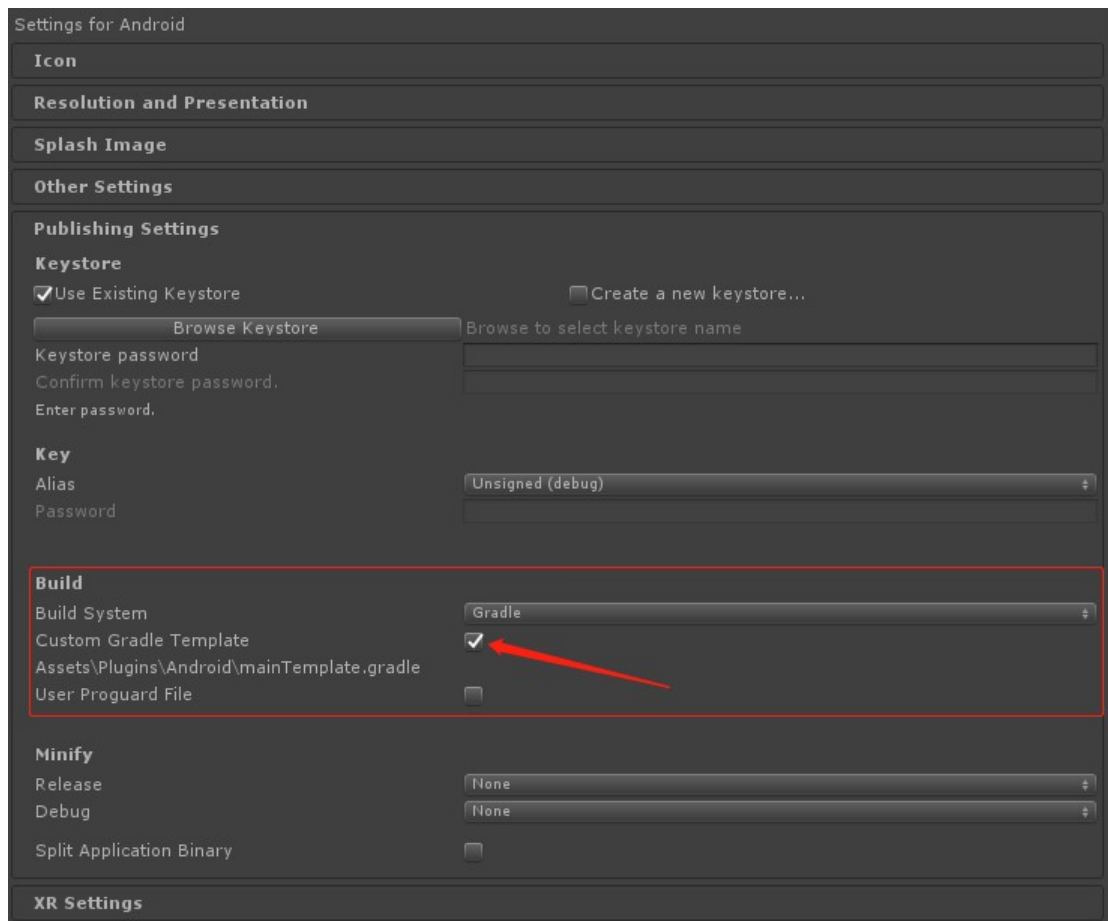


## 1、导入 Unity-Plugin



- i) 导入 NBMediation\_UnityPlugin.unitypackage。
- ii) 导入 3rdSDK.unitypackage 三方广告 SDK，也可以参考 iOS 和 Android 文档导入。。

## 2、Android 自定义 gradle 编译



## 3、SDK 初始化

```
NBMediation.Agent.init(slot_id, new AdTimingInitListener()); class
AdTimingInitListener : NBInitListener
{
    public void onError(string message)
    {
        Debug.LogError("UnityApp" + message);
    }
    public void onSuccess()
    {

```

```
        Debug.Log("UnityApp Init onSuccess");  
    }  
}
```

参考 AppoCamera.cs

## 4、开屏参考 AppoCamera.cs 文件

开屏广告位不在聚合后台配置，为直接调用 SDK 传入参数，目前只支持穿山甲。

```
NBMediation.Agent.showSplash("5040915", "光速清理", "887364502");
```

## 5、激励视频参考 AppoRewardedVideo 文件

激励视频填充库存由 SDK 决定，只需要调用 isReady 接口查看是否有填充，不需要管理预加载。

```
NBMediation.Agent.setRewardedVideoListener(new RewardedVideoListener());
```

```
if (NBMediation.Agent.isRewardedVideoReady()) {  
    NBMediation.Agent.showRewardedVideo();  
}
```

参考 AppoRewardedVideo.cs

## 6、插屏参考 AppoInterstitial 文件

插屏填充库存由 SDK 决定，只需要调用 isReady 接口查看是否有填充，不需要管理预加载。

```
NBMediation.Agent.setInterstitialListener(new InterstitialAdListener());

if (NBMediation.Agent.isInterstitialReady()) {
    NBMediation.Agent.showInterstitial ();
}
```

参考 AppoInterstitial.cs

## 7、Banner 参考 AppoBanner 文件

加载 banner 需要在初始化之后进行

i)loadBanner 加载 banner

ii)使用 isBannerReady 判断是否加载成功 iii)使用

showBanner 展示 banner iv)然后在 OnDestroy 时，使

用 hideBanner 销毁 banner

参考 AppoBanner.cs

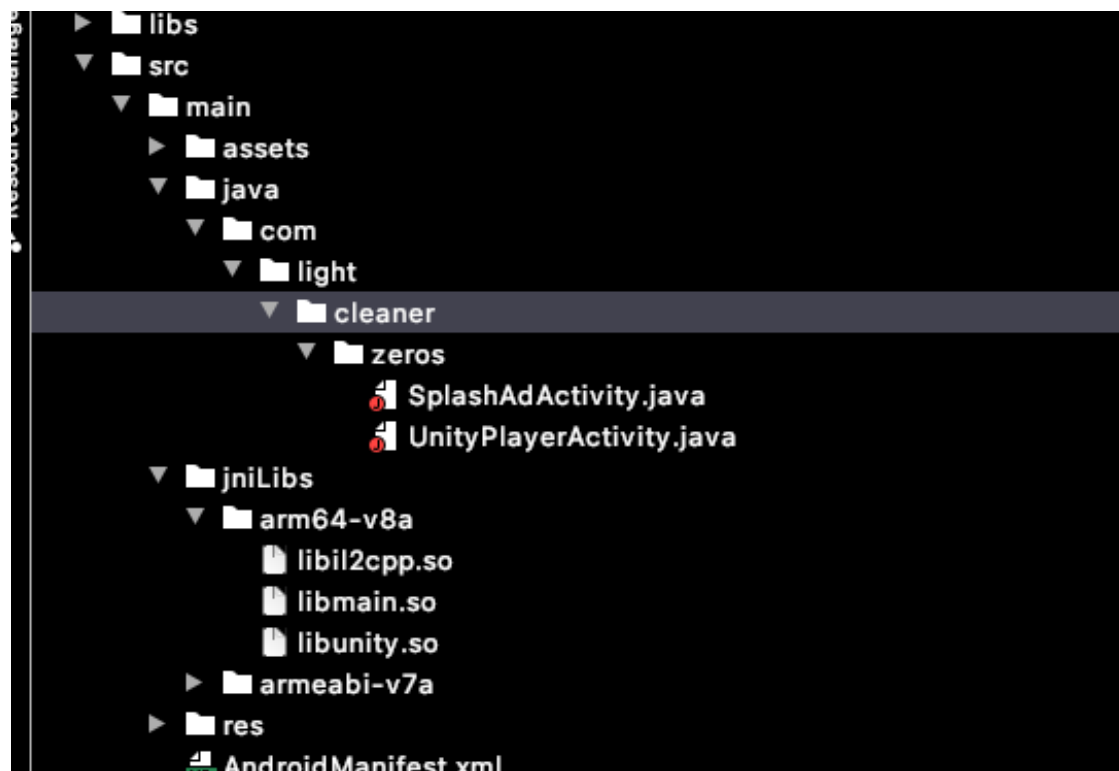
## 8、开屏

为保证广告效果，开屏需要从原生上接入，将开屏 Activity 作为 LaunchActivity。

- 1、老版本用户，移除 Unity 中的初始化 NBMediation.Agent.init 的调用与回调。
- 2、更新 U3D Plugins 的 AndroidManifest.xml，添加 SplashAdActivity 并设置为启动 Activity

```
<activity android:name=".SplashAdActivity"
    android:theme="@android:style/Theme.DeviceDefault.NoActionBar"
    android:configChanges="keyboardHidden|screenSize|orientation"
    android:screenOrientation="portrait"
    tools:ignore="LockedOrientationActivity">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

- 3、Unity Build Settings->Export Project 导出 Android Studio 工程。
- 4、打开导出的工程，加入 SplashAdActivity.java。负责初始化聚合，加载插屏，并返回 UnityPlayerActivity。



- 5、打开 AndroidManifest.xml，去掉 UnityPlayerActivity 的 intend-filter。

```

<application android:name="android.support.multidex.MultiDexApplication" and
<activity android:label="@string/app_name" android:screenOrientation="full
<intent-filter>
<del>    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
</del>
</intent-filter>
</activity>
<!--Necessary-->
<activity android:name="com.colorfast.kern.view.InnerWebViewActivity">
</activity>
<activity android:name=".SplashAdActivity" android:theme="@android:style/T
    <intent-filter>
    <action android:name="android.intent.action.MAIN" />
    <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
<!--for cloudssp rewardvideo-->
<activity android:name="com.colorfast.kern.view.RewardedVideoActivity" and
<activity android:name="com.colorfast.kern.view.InterstitialActivity" />
<meta-data android:name="unity.build-id" android:value="1f1cce4f-06eb-4684
<meta-data android:name="unity.splash-mode" android:value="0" />
<meta-data android:name="unity.splash-enable" android:value="True" />
</application>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

```

删除

检查

6、build.gradle 设置版本至少为 3.5.1

```

dependencies {
    classpath
    'com.android.tools.build:gradle:3.5.1'
}

```

如有 minSDKVersion 警告，请在 AndroidManifest.xml 和 build.gradle 中移除。

Activity.

7、开发者可在 SplashAdActivity 中加入 Launch Image，默认为黑色背景。