

# 网易洞见

AR开发手册 · UNITY版



网易洞见

# 目录

## Table of Contents

安装需求.....	1
Unity AR SDK 使用示例 .....	1
1. 示例场景.....	1
2. 导出 IOS 工程 .....	3
3. 导出 Android APK .....	7
如何对 AR 进行操作和获取相关数据 .....	8
1. 启动 AR .....	8
2. 处理 InsightARInterface 事件回调 .....	8
3. 获取 AR 状态.....	8
4. 停止 AR.....	8
附录：接口详细说明.....	9

# 安装需求

## Unity:

Unity 5.6.2f1

## IOS:

XCode 9.1, IOS8 以上。

iPhone5s、iPad air 以上性能设备。

## Android:

AndroidStudio 2.3 或以上， Android 5.0 以上。

API Level >= 20, SDK BuildTools 版本>=23。

建议运行在中高性能设备上。

# Unity AR SDK 使用示例

## 1. 示例场景

将 InsightARSDK-1.4.4.unitypackage 导入新建的 Unity 工程中。

找到 Project/Scenes/Resources/Scenes 文件夹，打开 Demo 场景，如图 1 所示。

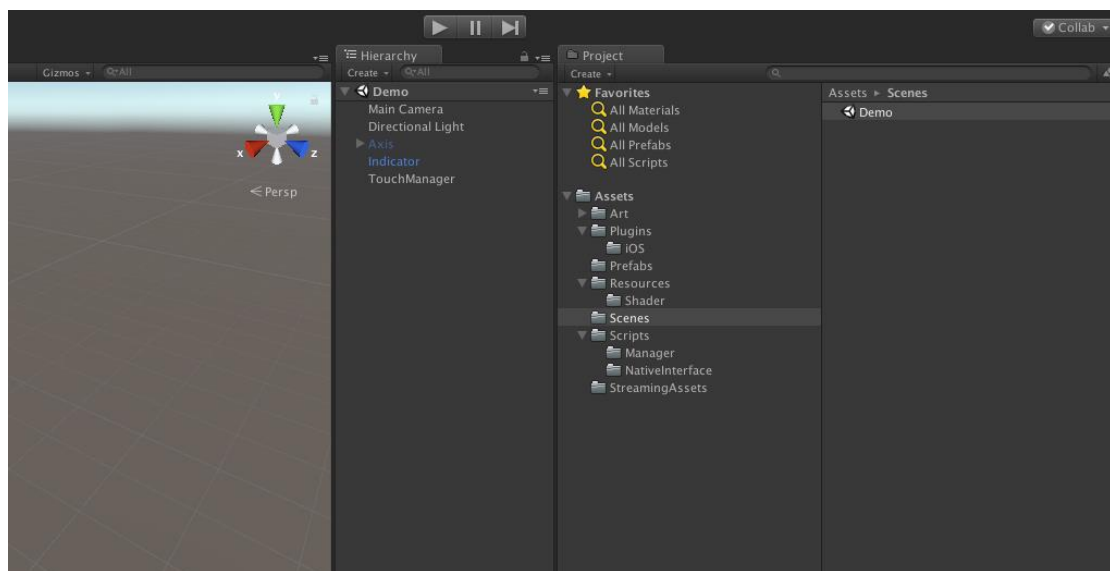


图 1 导入 InsightARSDK Unity Package

选中场景中的 **Main Camera**，上面挂有一个脚本 **InsightARManager.cs**。这个脚本是我们提供的一个例子，帮助了解如何操纵 AR 和获取相关信息。

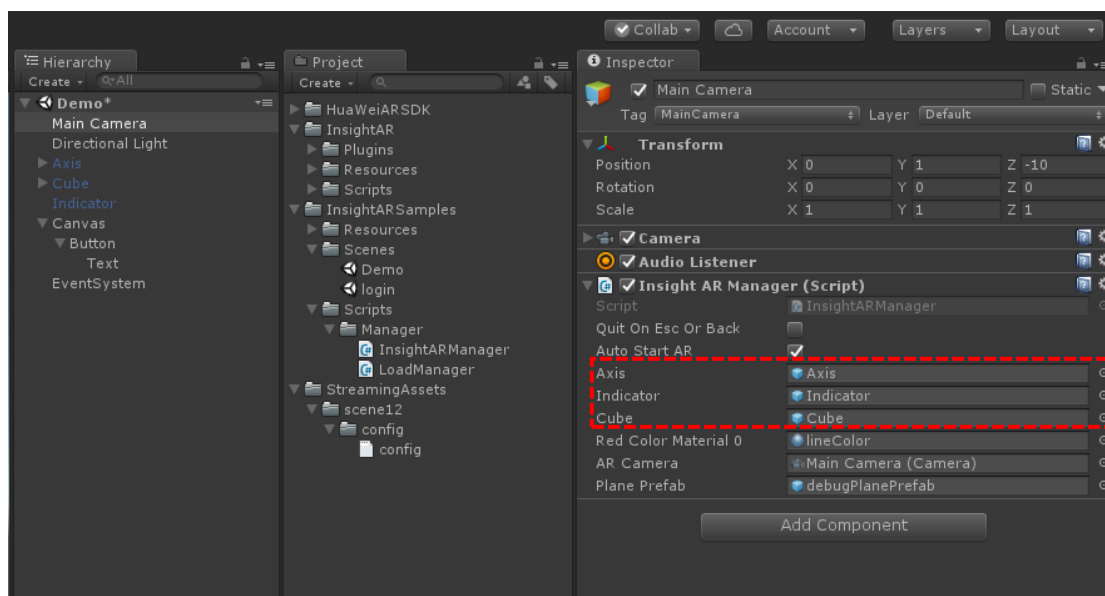


图 2 InsightARManager.cs 脚本

脚本上挂三个 Prefab：Axis，Indicator 和 Cube。

当 Demo 场景在手机上运行以后，Axis 将会显示在世界坐标系的原点。Indicator 会显示在检测出的虚拟平面上。Cube 可以通过点击屏幕，放置在 Indicator 的位置上。

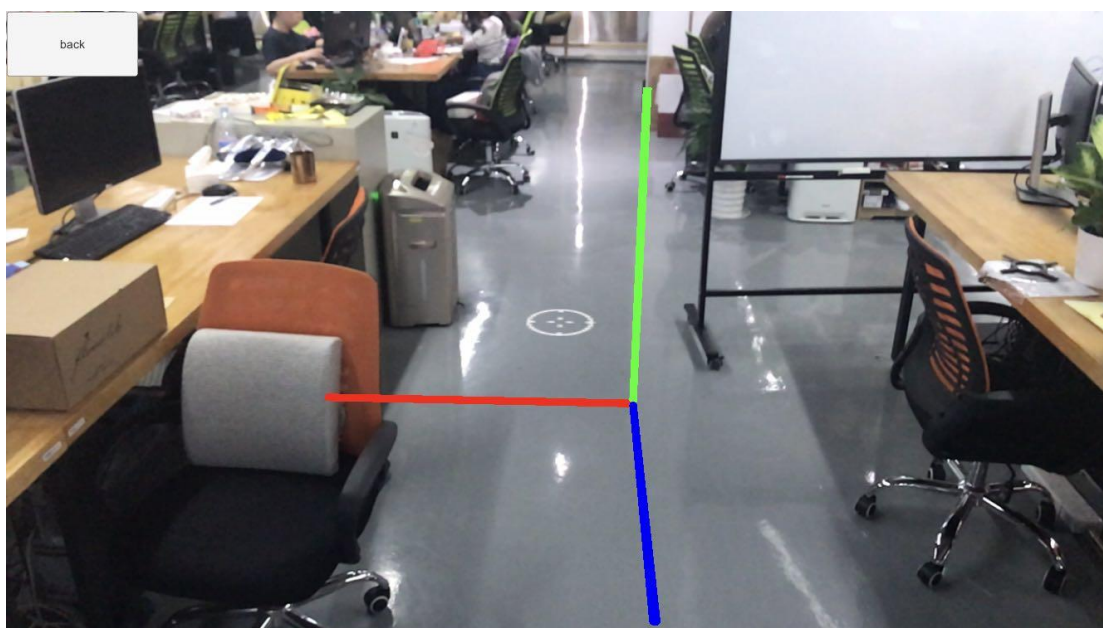


图 3 Demo 场景运行在手机上的截屏 - Axis 和 Indicator



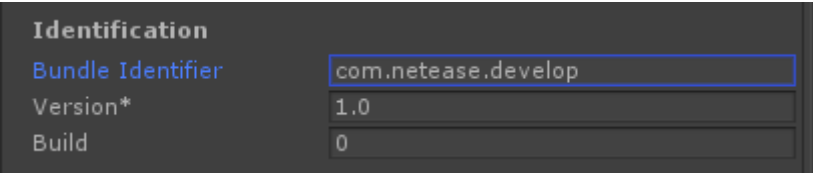


图 6 修改后的 Bundle Identifier

在 **Mac** 上使用 XCode 打开导出的工程。双击 Unity-iPhone.xcodeproj。

MapFileParser.sh	2017年6月11日 上午12:46	1 KB	Shell Script
opencv2.framework	今天 上午9:44	--	文件夹
process_symbols.sh	2017年6月17日 上午9:01	373 字节	Shell Script
Unity-iPhone	今天 上午10:17	--	文件夹
Unity-iPhone Tests	今天 上午10:17	--	文件夹
Unity-iPhone.xcodeproj	今天 上午10:43	132 KB	Xcode Project
UnityData.xcassets	今天 上午10:17	--	文件夹

图 7 导出的 Xcode 工程

打开工程之后，连接 iPhone 手机，配置 Code Signing，选择一个 Team 账号。如果之前没有添加过账号，需要通过登录 Apple ID 来添加。

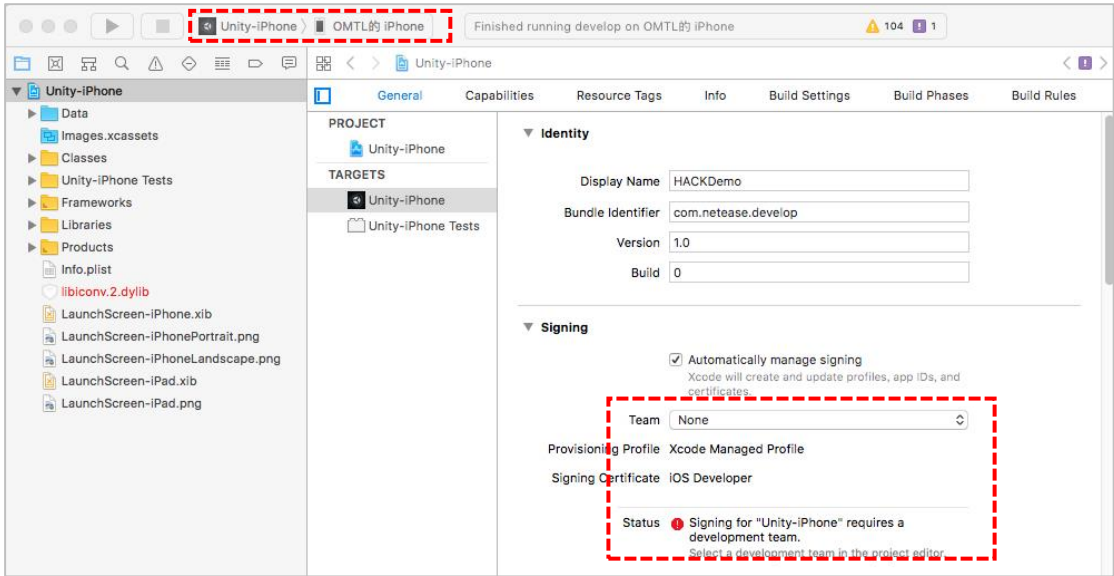


图 8 Code Singing 需要被配置



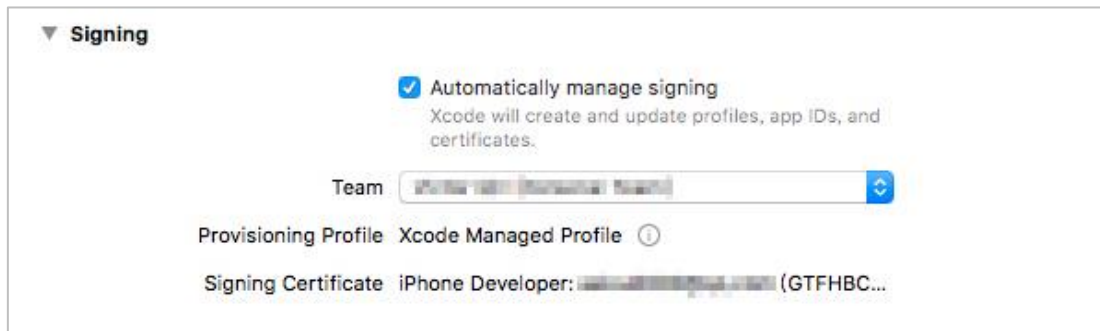


图 9 Code Signing 配置好后

接着，找到我们提供的 opencv2.framework.zip 文件，解压后将 opencv2.framework 拖入到 XCode 工程的 Frameworks 之中。

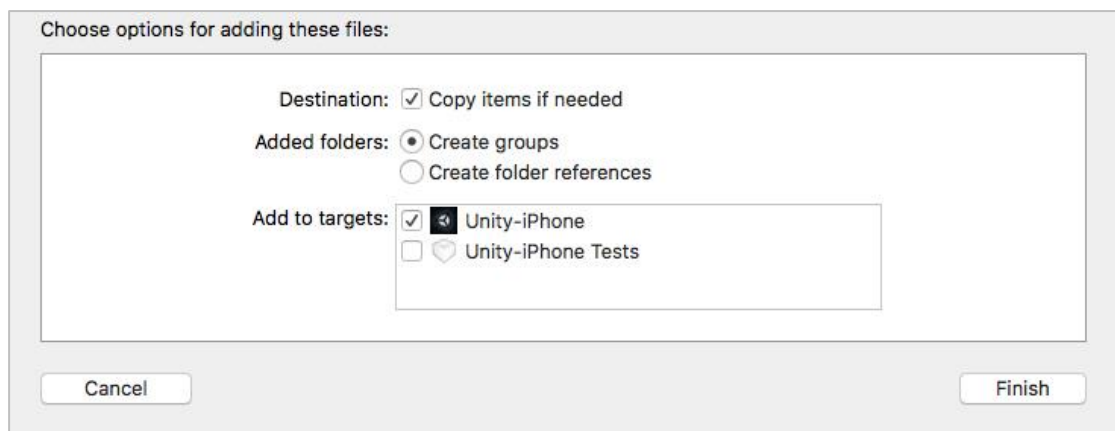


图 10 导入 opencv2.framework 后弹出的窗口，点击 Finish 即可

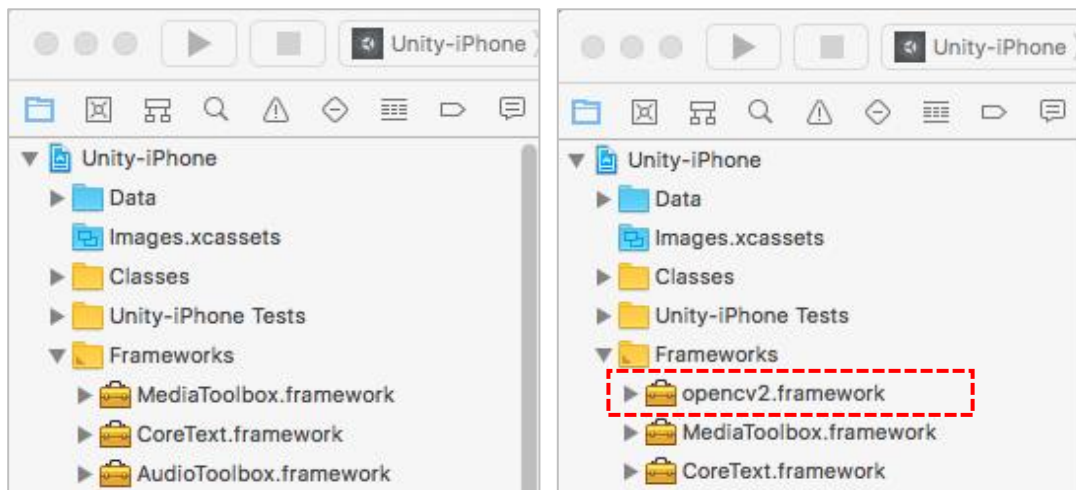


图 11 opencv2.framework 拖入 Frameworks 前后

在某些情况下，按照上面的步骤添加 OpenCV 库以后，依然无法通过编译。选中 opencv2.framework，将 Target Membership 中 Unity-iPhone 前的勾去掉，然后重新勾选上，即可通过编译。

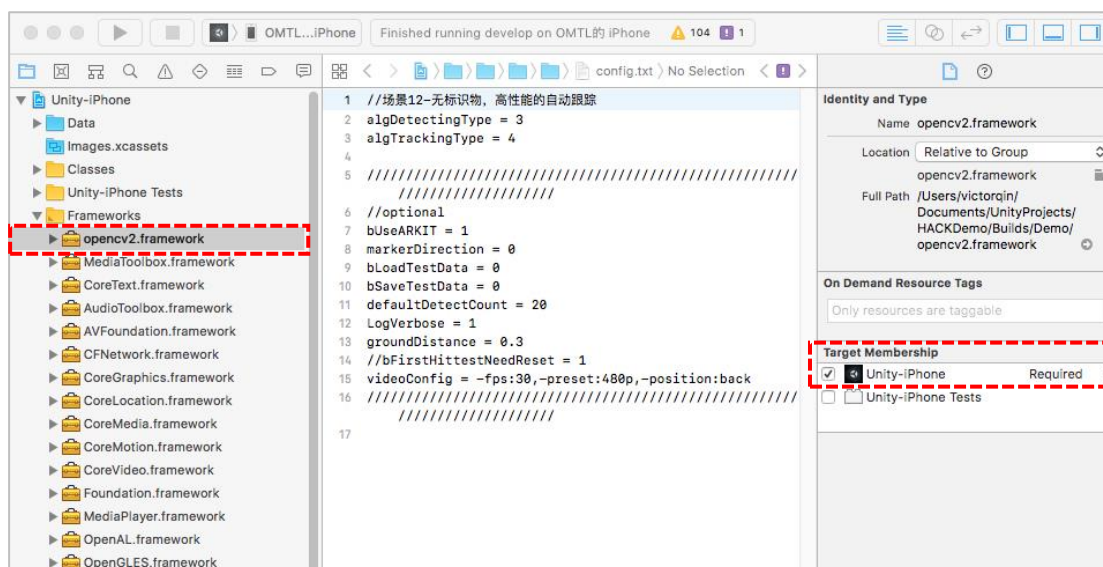


图 12 重新勾选 Target Membership

添加 OpenCV 库之后，由于使用 AR 需要获取相机的权限，需要添加相应的权限。找到 Info.plist 文件，点击 Information Property List 旁边的加号按钮，添加一个属性。在 Key 的一栏填入 **NSCameraUsageDescription**。Value 一栏填入请求相机权限的描述文字。

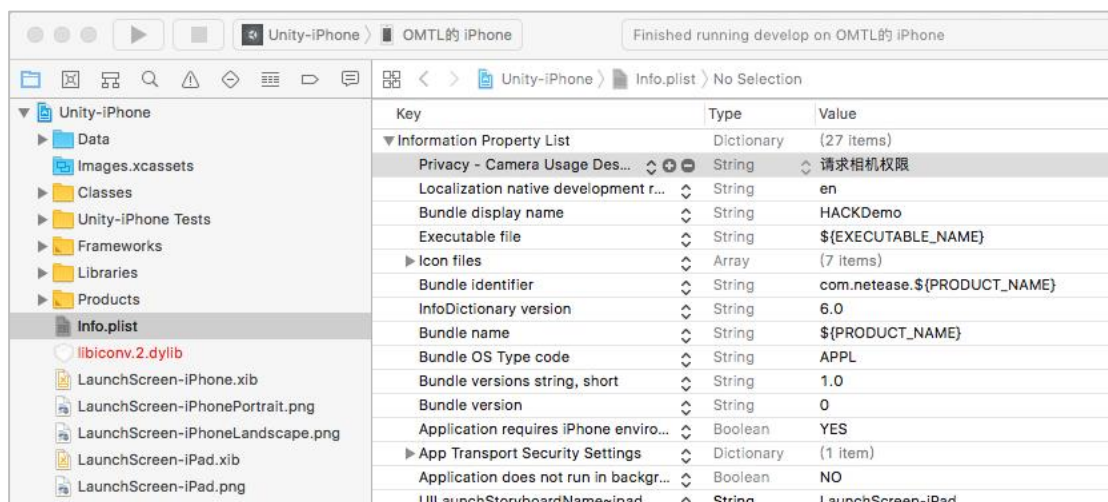


图 13 添加相机权限属性后

完成以上配置之后，确保 iPhone 手机已经连接到 Mac，点击编译并执行运行即可。务必确保手机已连接互联网，用于验证密钥，否则 AR 无法启动。





### 3. 导出 Android APK

对于 Android 来说，由于可以直接导出 APK，相对简单。

如图所示设置 Player Settings：

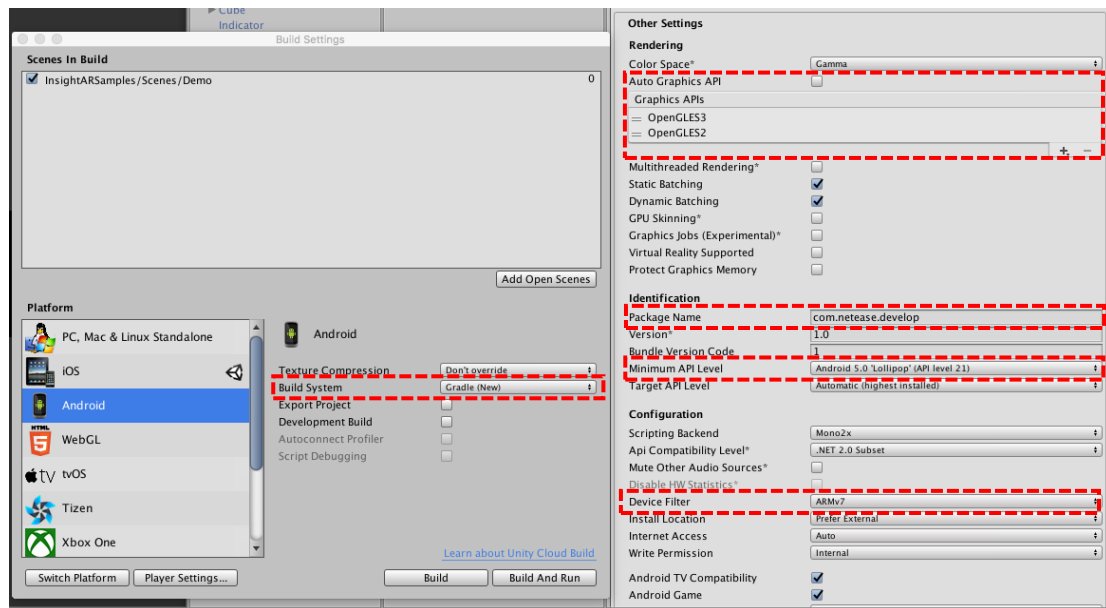


图 14 Unity 导出 Android 配置预览图

**Build System** 选择 Gradle (New)。

**Auto Graphics API** 不要勾选，并在 Graphics APIs 列表中移除 Vulkan。

更改 **Package Name**。

**Minimum API Level** 选择 Android5.0。

**Device Filter** 选择 ARMv7。

将 Android 手机连接电脑，然后点击 **Build And Run**。

如果 Build 过程中提示需要配置 Android 环境，配置方法参见：

<https://docs.unity3d.com/Manual/android-sdksetup.html>

# 如何对 AR 进行操作和获取相关数据

InsightARInterface 类是获取 AR 相关数据的主要 API。

下面我们将以 InsightARManager.cs 作为示例，介绍一下如何对 AR 进行操作和获取数据。

## 1. 启动 AR

如下代码用于启动 AR：

```
m_ARInterface = InsightARInterface.GetInterface();
InsightARSettings setting = new InsightARSettings()
{
    configPath = _algPath,
    appKey = "AR-905E-782934EC990A-a-f",
    appSecret = "1AfDaV7ALI"
};
m_ARInterface.StartAR(setting);
m_ARInterface.SetupCamera(m_ARCamera);
```

configPath 是算法配置文件的路径。这个文件在 StreamingAssets\scene12\config 中可以找到。其中包含对算法的一些配置。appKey 和 appSecret 是密钥，在 AR 启动时，会通过网络来验证密钥。如果验证不成功，AR 将无法启动。

## 2. 处理 InsightARInterface 事件回调

可以对平面识别事件添加相应的回调函数，做出相应的处理。

```
InsightARInterface.planeAddedAction += OnPlaneAdded;
InsightARInterface.planeUpdatedAction += OnPlaneUpdated;
InsightARInterface.planeRemovedAction += OnPlaneRemoved;
```

## 3. 获取 AR 状态

根据 InsightARInterface 的 CurrentState，获取 AR 当前的状态。例如：

```
labelStatus.text = "Status:" + m_ARInterface.CurrentState;
if (m_ARInterface.CurrentState == InsightARState.Init_Fail)
{
    labelStatus.text += "\n";
}
```

## 4. 停止 AR

```
m_ARInterface.StopAR();
```

当程序停止或者退出时，调用 InsightARInterface 的 StopAR()函数，停止相机，结束算法逻辑。

# 附录：接口详细说明

## InsightARInterface

InsightARInterface 是 Unity 中使用 InsightAR 的主要入口。

返回值	方法/变量名	参数	说明
<b>InsightARInterface</b>	GetInstance	()	static 方法获得 ARInterface 实例
<b>InsightARState</b>	CurrentState		Static 变量，获得当前 AR 状态
<b>Action</b>	planeAddedAction		static Action，当平面增加时,可以得到通知
<b>Action</b>	planeUpdatedAction		static Action，当平面更新时,可以得到通知
<b>Action</b>	PlaneRemovedAction		static Action，当平面移除时,可以得到通知
<b>void</b>	StartAR	InsightARSetting arSetting	启动 AR 功能
<b>void</b>	StopAR	()	停止 AR 功能
<b>void</b>	ResetAR	String path	重新启动 AR，参数是提供切换场景
<b>void</b>	Update	()	更新 AR
<b>void</b>	UpdateARBackground	()	更新 AR 背景
<b>void</b>	SetupCamera	Camera arCamera	设置 AR Camera
<b>void</b>	SwitchCamera	()	切换前后相机（仅限洞见内部版本）
<b>InsightARUserHitAnchor</b>	GetHitTestResult	Touch touch	根据屏幕的 Input.touch 来获得点击点对应的真实世界的坐标信息
<b>InsightARUserHitAnchor</b>	GetHitTestResult	float pointX, float pointY	根据屏幕的相对坐标获得点击点对应的真实世界的坐标信息

## InsightARSettings

<b>struct</b>	<b>InsightARSettings</b>	<b>AR 设置</b>
<b>string</b>	configPath	算法配置路径
<b>string</b>	appKey	AppKey
<b>string</b>	appSecret	AppSecret

## InsightARState

enum InsightARState AR 运行状态

Uninitialized	AR 未启动，未初始化
Initing	AR 初始化中，打开相机，联网，加载等
Init_OK	AR 初始化成功，每次必然出现一帧
InitFail	AR 启动失败，额可以通过 InsightARReason 获得失败原因
Detecting	AR 检测中
Detect_OK	AR 检测成功，在 Tracking 前一定有一帧
Detect_Fail	AR 检测失败
Tracking	AR 跟踪中
Track_Limited	AR 跟踪受限制，可能相机被挡住，或者剧烈运动中
Track_Fail	AR 跟踪失败
Track_Stop	AR 跟踪停止

## InsightARAnchorType

enum InsightARAnchorType AR 锚点类型

Plane	0	平面类型。当前 anchor 描述的是一个平面
UserHitTest	1	用户点击类型。当前 anchor 描述的是一个用户点击点
Marker_2d	2	2D 图片识别
Object_3D	3	3D 物体识别（仅限洞见内部版本）

## InsightARUserHitAnchor

struct InsightARUserHitAnchor 描述了点击后得到的跟踪锚点的信息

string	identifier	锚点的 ID
Vector3	position	锚点位置
Quaternion	rotation	锚点的方向
int	isvalid	平面是否有效

## InsightARPlaneAnchor

struct InsightARPlaneAnchor 描述了一个检测到的平面

string	identifier	平面的 ID
Vector3	center	平面中心位置
Vector	extent	平面的大小
Quaternion	rotation	平面的方向
int	isvalid	平面是否有效