

# HoloKit 快速上手开发手册

## HoloKit 快速上手开发手册

### 1 关于 MR 你需要知道

三个主要特点

与其他可视化环境相比

### 2 HoloKit 是什么？

2.1 产品简介

2.2 案例分享

2.3 相关产品链接

### 3 ARkit 简单介绍

3.1 背景与平台

3.2 ARKit 架构简述

3.3 我们需要的 Unity 与 ARKit

### 4 Unity 下的 HoloKit 开发入门

4.1 SDK 下载与分支简介

4.2 HoloKit 的 scene 所在目录

4.3 自带 Demo 场景下的 hierarchy 简介

### 5 简单 Demo 和真机测试教程 (附视频录像)

#### 5.1 制作一个简单场景

准备阶段 (iOS)

准备阶段 (Android)

快速开始 - Universal (iOS and Android)

快速开始 - ARKit v1.5 Only (iOS 11.3)

创建你自己的场景

#### 5.2 从 Unity 打包至 iPhone

相关 Q&A

# 1 关于 MR 你需要知道

MR（简称混合现实），包括**增强现实和虚拟现实**，指的是合并现实和虚拟世界而产生的新的可视化环境，在新的可视化环境里物理和数字对象共存，并实时互动

它是虚拟现实技术的进一步发展，该技术通过在虚拟环境中引入现实场景信息，在虚拟世界、现实世界和用户之间搭起一个交互反馈的信息回路，以**增强用户体验的真实感**

## 三个主要特点

- 结合了虚拟和现实
- 在现实中获得虚拟体验
- 实时运行

## 与其他可视化环境相比

- 混合现实（MR）的实现需要在一个能与现实世界各事物相互交互的环境中
- 如果一切事物都是虚拟的那就是VR领域
- 如果展现出来的虚拟信息只能简单叠加在现实事物上，那就是AR
- MR的关键点就是与现实世界进行交互和信息的及时获取



## 【相关阅读】

知乎《VR、AR和MR的区别？》：<https://www.zhihu.com/question/39157545>

互动百科：<http://www.baik.com/wiki/%E6%B7%B%E5%90%88%E7%8E%B0%E5%AE%9E>

## 2 HoloKit 是什么？

### 2.1 产品简介

HoloKit是一款**头戴式MR眼镜**，用镜片组合投影智能手机屏幕画面的方式，以极低成本实现了“虚拟与现实混合”的MR效果。仅需要一部手机和MR软件，HoloKit就可以为用户带来混合现实的体验，同时为开发者和使用者提供低门槛上手、高质量体验。



同时，HoloKit拥有目前市面较高的视野范围的76度FOV（视场角），支持ARKit/ARCore等AR开发工具，具备多人联机能力，并配备手柄以实现更多的交互。游戏、教育、培训等方向都是未来的应用场景。

在当前市场中，HoloKit凸显了其在消费者级别设备中的优势地位。对于开发者，HoloKit是一款入手便宜，上手简单，非常便于开发的MR设备，在短时间内就可以实现并体验混合现实效果。

### 2.2 案例分享

MR视觉效果



## 玩家联机互动



## 新版HoloKit



## 2.3 相关产品链接

### HoloKit中文官方网址

<https://holokit.163.com/>

### HoloKit英文官方网址

<https://holokit.io/>



## 3 ARkit 简单介绍

### 3.1 背景与平台

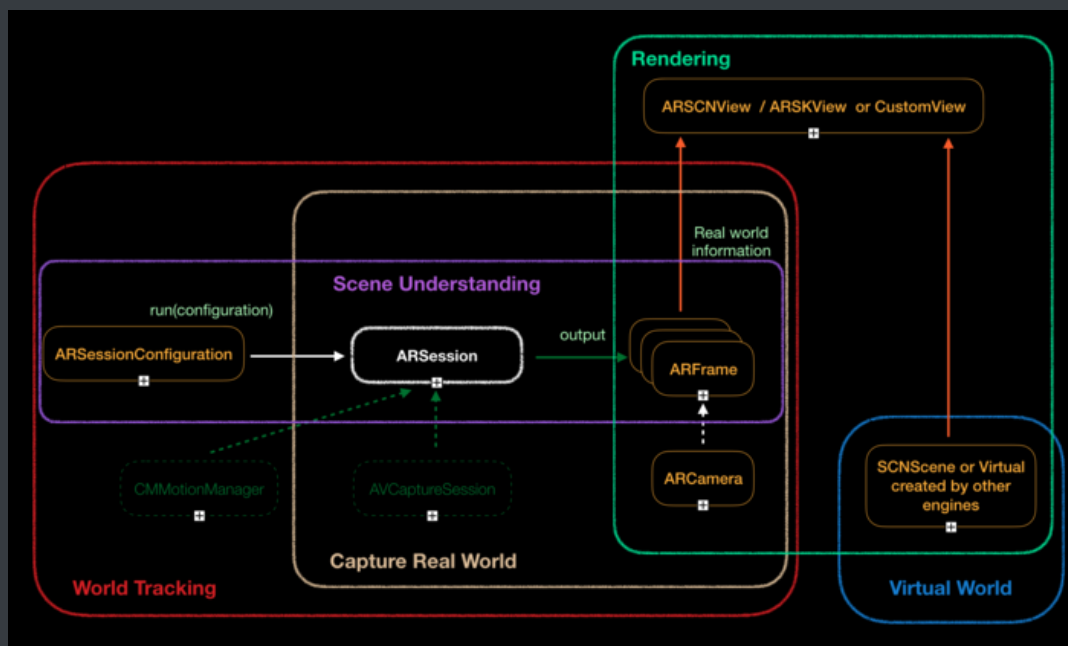


ARKit 是苹果 WWDC2017 中发布的用于开发iOS平台 AR 功能的框架。ARKit 为上一节中提到的 AR 系统架构中各个部分都提供了实现方案，并且为开发者提供了简单便捷的 API，使得开发者更加快捷的开发 AR 功能。

基于 unity 的 ARKit 开发的使用需要一定的软硬件设施：

- 软件：
  - Unity的AR Kit需要**Unity 5.6.1p1**或更高版本的补丁版本（建议使用 **Unity 2017**）
  - 需要 **iOS 11** 或更高版本
  - 需要 **XCode 9** 测试版或更高版本，其中要包含 ARKit Framework 的 iOS SDK
- 硬件：
  - 处理器为 A9 及以上的 iPhone 或 iPad 设备 (iPhone 6s 为 A9 处理器)

### 3.2 ARKit 架构简述



上图列出了 ARKit API 中的几个主要的类，如 `ARSession`、`ARSessionConfiguration`、`ARFrame`、`ARCamera` 等。并依据各个类的功能进行了模块划分：红色(World Tracking)、蓝色(Virtual World)、土色(Capture Real Wrold)、紫色(Scene Understanding)、绿色(Rendering)。

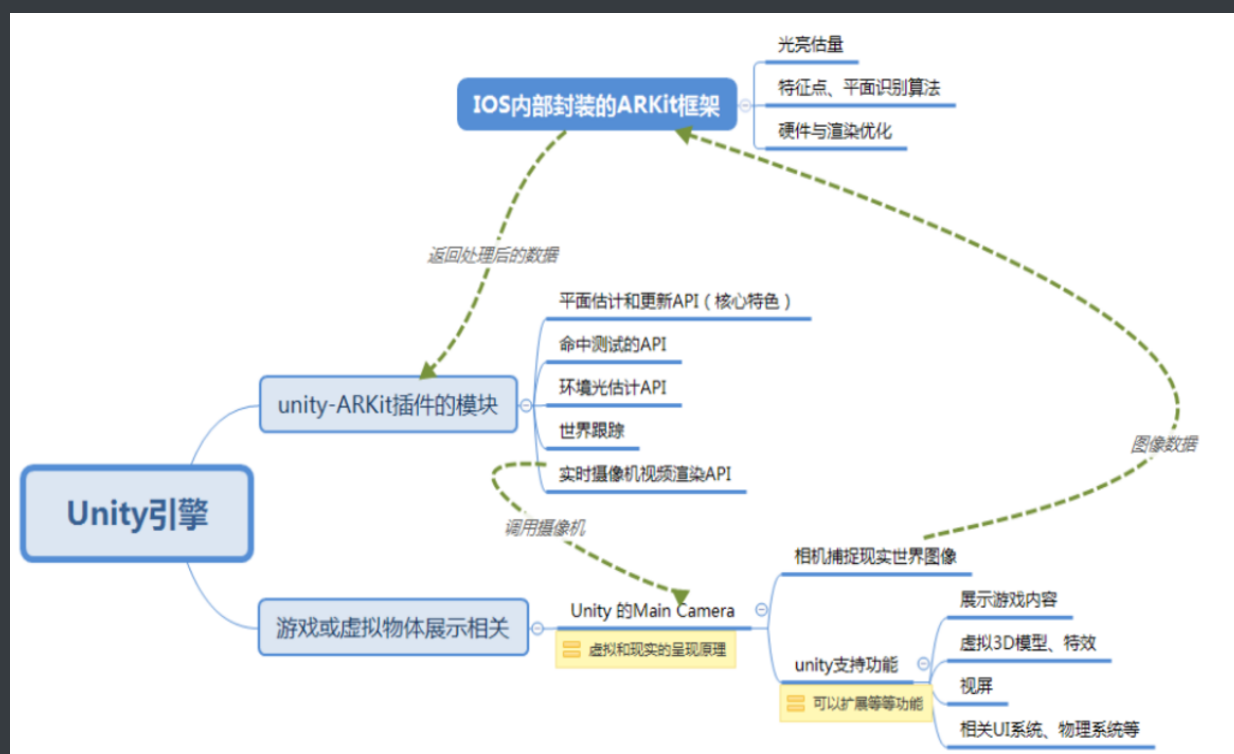
对于上图，ARSession 是核心整个ARKit系统的核心，ARSession 实现了世界追踪、场景解析等重要功能。而 ARFrame 中包含有 ARSession 输出的所有信息，是渲染的关键数据来源。

#### 【相关阅读】

苹果官方ARKit开发文档：<https://developer.apple.com/documentation/arkit>

不错的ARKit讲解《直击苹果 ARKit 技术》：<https://www.jianshu.com/p/7faa4a3af589>

### 3.3 我们需要的 Unity 与 ARKit



虽然 ARKit 提供的 API 较为简单，但看到上面整个框架后，对于初识整个体系的开发者来说，还是会觉着有些庞大。然而，对于有unity开发经验的人来说，ARKit的相关开发十分容易上手，在后面我们将会为大家提供10分钟完成一个可用Demo的制作教程。

#### 【相关阅读】

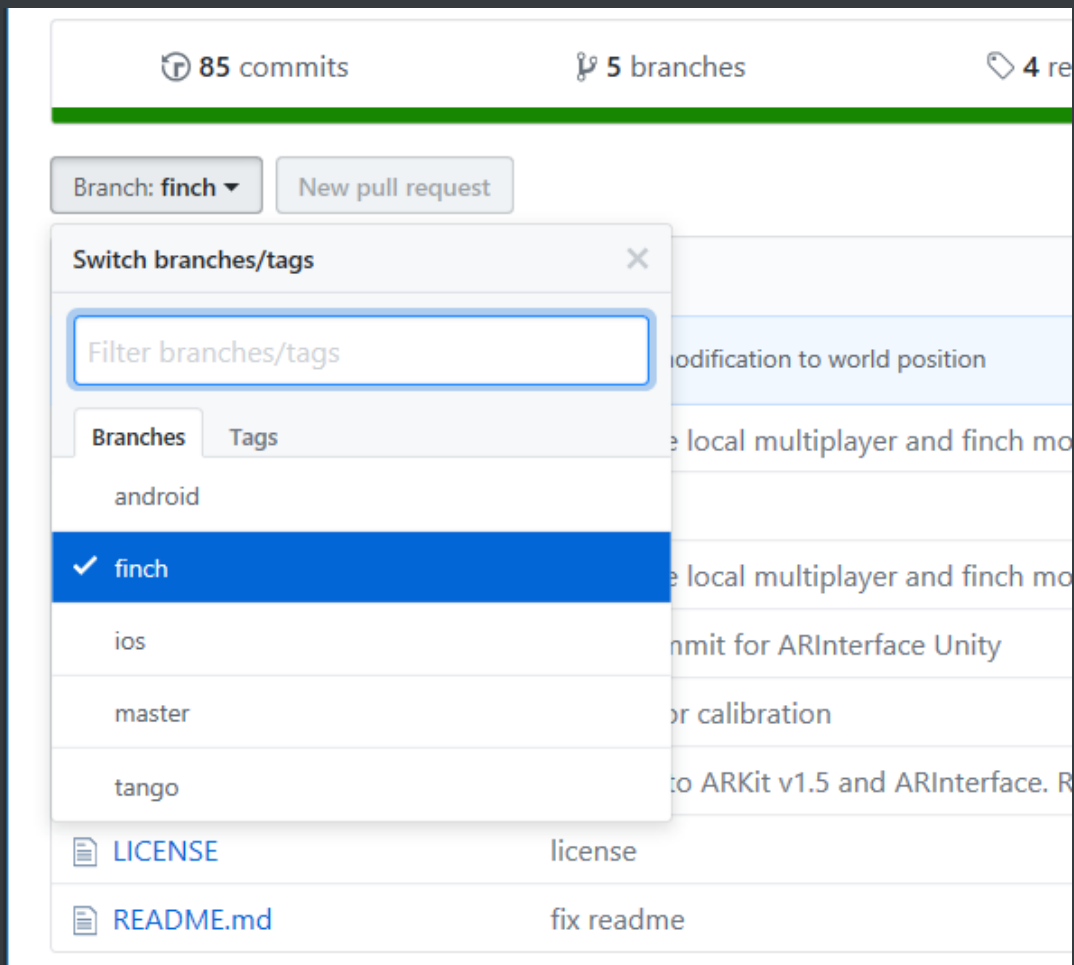
ARKit的Unity开发插件下载：<https://assetstore.unity.com/packages/essentials/tutorial-projects/unity-arkit-plugin-92515>

《Unity与ARKit插件实践评测总结》：[https://blog.csdn.net/aooosss2012/article/details/76273443#OLE\\_LINK1](https://blog.csdn.net/aooosss2012/article/details/76273443#OLE_LINK1)

《Unity 与 ARkit 开发基本特征和案例分享》：<http://www.52vr.com/article-1774-1.html>

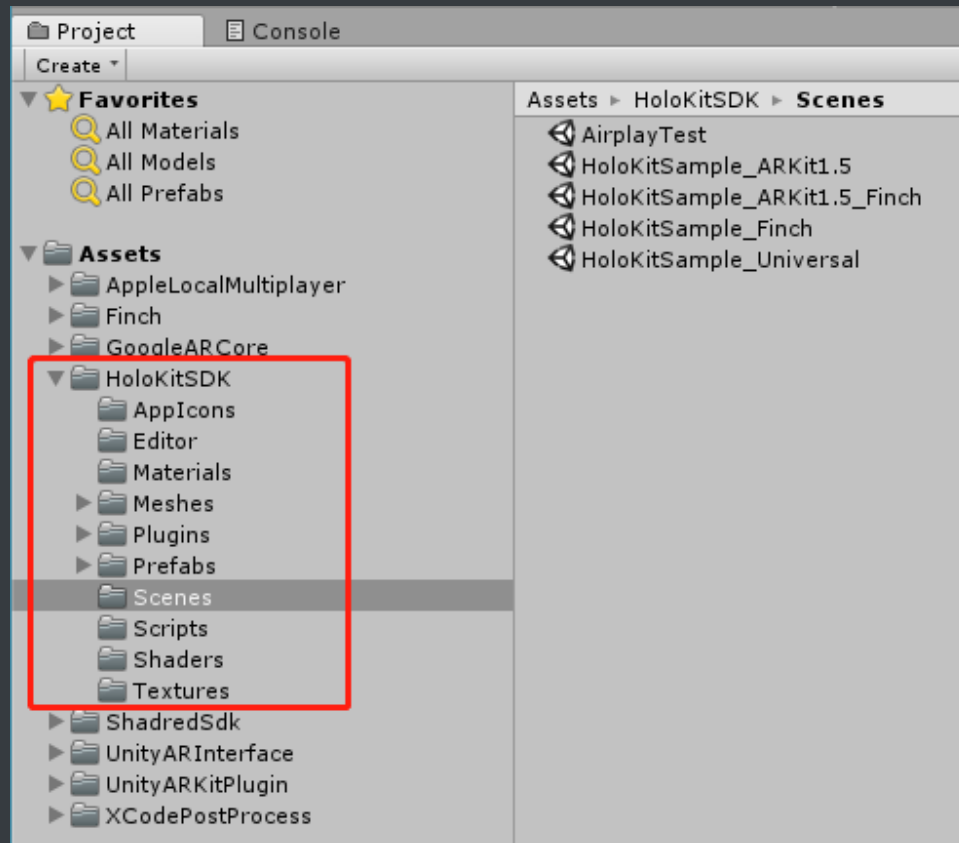
## 4 Unity 下的 HoloKit 开发入门

### 4.1 SDK 下载与分支简介



- 请进入HoloKit官方Github上下载最新的HoloKitSDK,地址如下  
[github.com/holokit/](https://github.com/holokit/)
- 推荐选择finch分支下的HoloKit SDK进行开发
- 该分支下的SDK包含如下内容
  - ARKit 1.5
  - ARCore
  - HoloKit
  - Finch手柄（3dof与6dof均可）
  - Apple联机

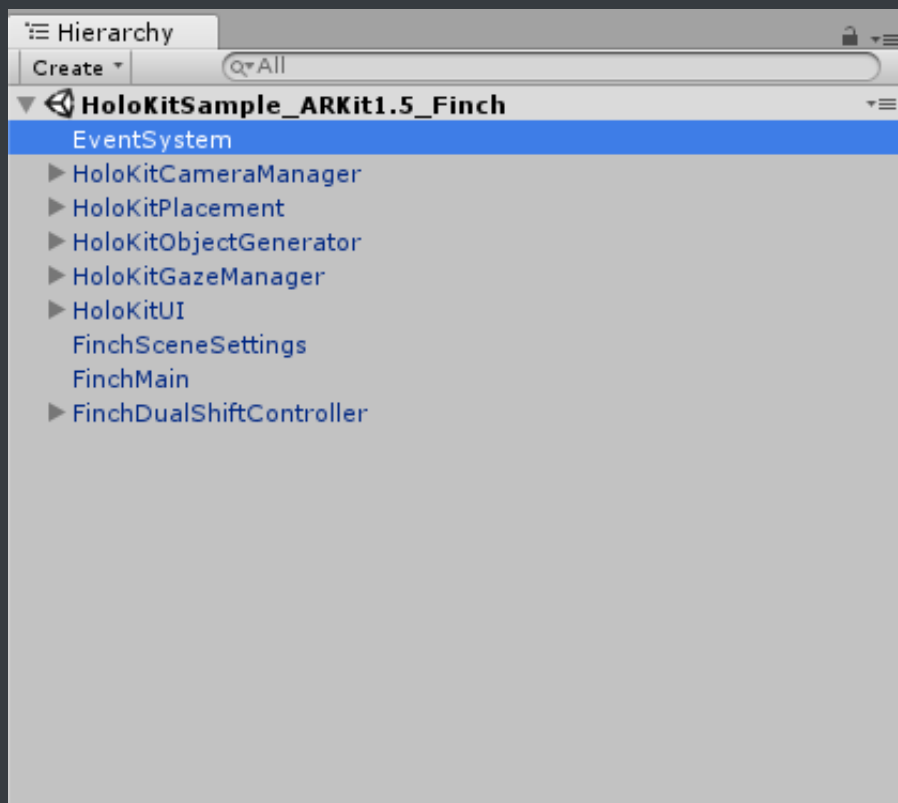
## 4.2 HoloKit 的 scene 所在目录



- HoloKitSample\_ARKit1.5 —— 包含ARKit1.5的功能
- HoloKitSample\_ARKit1.5\_Finch —— 包含ARKit1.5及Finch手柄的功能
- HoloKitSample\_Finch —— 包含ARKit1.0及Finch手柄的功能
- HoloKitSample\_Universal —— 包含ARKit1.0及ARCore的功能



### 4.3 自带 Demo 场景下的 hierarchy 简介



- HoloKitCamera — 最基础的组件来使你的游戏响应ARKit 追踪
  - 它有一个子物体VideoSeeThroughCamera, 使你的游戏有类似VR的体验。它有两个额外的 Camera "Left Eye", "Right Eye", 来为HoloKit渲染立体的呈像, 在"HoloKitCameraRig"物体中, 你可以设置默认的观看模式和切换模式的键
- HoloKitObjectGenerator
  - DisplayPlanes — 当ARKit检测到地面时可以动态生成可碰撞的地面, 而且你可以设置一个 Plane Prefab作为它的设置对象来可视化生成平面
  - PointCloud — 用于可视化ARKit检测的点云, 是为了调试的目的
  - AmbientLight — 定向光, 强度由ARKit控制。强度将根据当前环境照明的实际情况自动调整
- HoloKitPlacementRoot — 一个方便的预制件, 帮助你把你的物体放在地上
- HoloKitGazeManager — 一个方便的预制件来帮助你发射射线来触发一些交互

## 5 简单 Demo 和真机测试教程 (附视频录像)

【操作教学视频录像传送门】

### 1. HoloKit简介

[http://v.youku.com/v\\_show/id\\_XMzYyMjQxNzUyMA==.html](http://v.youku.com/v_show/id_XMzYyMjQxNzUyMA==.html)

### 2. 什么是MR

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2MDk0NA.html](http://v.youku.com/v_show/id_XMzYyMjl2MDk0NA.html)

### 3. HoloKit SDK的下载和导入

[http://v.youku.com/v\\_show/id\\_XMzYyMjl5OTM2NA.html](http://v.youku.com/v_show/id_XMzYyMjl5OTM2NA.html)

### 4. HoloKit SDK简单介绍

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2NTIwMA.html](http://v.youku.com/v_show/id_XMzYyMjl2NTIwMA.html)

### 5. HoloKit SDK的相关配置

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2NjE4OA.html](http://v.youku.com/v_show/id_XMzYyMjl2NjE4OA.html)

### 6. HoloKit Xcode项目打包到iPhone上

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2ODA3Ng.html](http://v.youku.com/v_show/id_XMzYyMjl2ODA3Ng.html)

### 7. HoloKit重要组件的介绍

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2OTAyMA.html](http://v.youku.com/v_show/id_XMzYyMjl2OTAyMA.html)

### 8. HoloKit手柄的简单介绍

[http://v.youku.com/v\\_show/id\\_XMzYyMjl3MzQxMg.html](http://v.youku.com/v_show/id_XMzYyMjl3MzQxMg.html)

### 9. HoloKit联机的简单介绍

[http://v.youku.com/v\\_show/id\\_XMzYyMjl2OTgzNg.html](http://v.youku.com/v_show/id_XMzYyMjl2OTgzNg.html)

## 5.1 制作一个简单场景

### 准备阶段 (iOS)

#### 1. 注册成为**苹果开发者**

#### 2. 一台支持 ARKit v1.5 的设备并且iOS系统为11.3以上

#### 3. XCode 9.3.x 你需要去你的苹果开发者账号处下载

#### 4. Unity2017.3.1f3 或更新的版本, 确保你已安装iOS组件

- 转换你的目标平台为iOS并且点击Switch Platform
- 打开“File”->“Build Settings”并且点击“Build”.确保以下参数是正确的.
- Player Setting -> Other Settings
  - Arctecture:ARM64
  - Camera Usage Description:"HoloKit need use Camera".

## 准备阶段 (Android)

1. 一台支持ARCore的设备并且系统版本为Android7.0以上 (API Level 24)
  - 准备你的设备
  - 启用developer options
  - 启用USB debugging
2. 下载Android Studio并且相应Android SDK 已经安装
3. Unity2017.3.1f3或更新的版本, 确保你已安装Android组件
  - 转换你的目标平台为Android并且点击Switch Platform
  - 点击“Player Settings ”来打开“Android Player Settings”, 之后改变下面的选项
    - Other Settings > Multithreaded Rendering: Off
    - Other Settings > Package Name: 一个不同的app ID 并且看起来像是Java 的包名, 例如 com.example.helloAR
    - Other Settings > Minimum API Level: Android 7.0 或更高
    - Other Settings > Target API Level: Android 7.0 or 7.1
    - XR Settings > ARCore Supported: On

## 快速开始 – Universal (iOS and Android)

1. 在一个新的Unity项目中导入“HoloKitSDK”文件夹在 "Assets"文件夹下
2. 打开案例场景"HoloKitSDK/Examples/HoloKitSample\_Universal".
3. 构建你的场景并且运行。这一场景可以同时运行在iOS和Android
4. 在app运行之后, 你应该看到一个方块漂浮在空中的某一处, 你可以将屏幕中心处的点对准球体看一下效果
5. 这个app可以扫描地面, 并且你可以点击屏幕去放置方块于地面上
6. 你可以触碰“AR/MR”按钮去转换成HoloKit模式

## 快速开始 – ARKit v1.5 Only (iOS 11.3)

1. 在一个新的Unity项目中导入“HoloKitSDK”文件夹在 "Assets"文件夹下
2. 打开案例场景"HoloKitSDK/Scenes/HoloKitSample\_ARKit1.5".
3. 你可以扫描垂直的地面
4. 你可以扫描下面这个图像标记

"Assets/UnityARKitPlugin/Examples/ARKit1.5/UnityARImageAnchor/ReferenceImages/unitylog  
owhiteonblack.png".

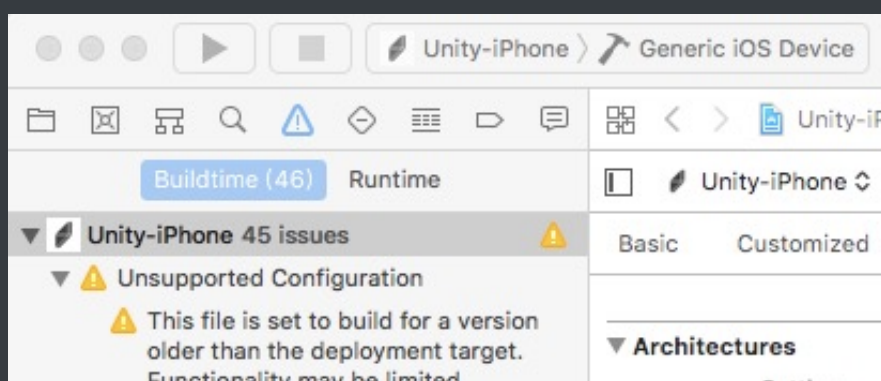


## 创建你自己的场景

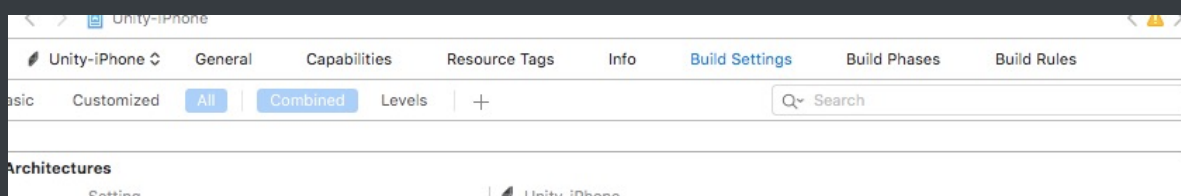
1. 在Unity中创建一个新的场景
2. 拖拽在 "HoloKitSDK/Prefabs/Universal"中的所有的物体进入场景
3. 放置任何你喜欢的东西在 "HoloKitPlacementRoot"下面，并且你的模型应该是和 "DebugCube"相似的大小，之后可以按照自己的想法选择是否删除"DebugCube".
4. 构建并运行你的场景
5. 如果你不喜欢 ambient light，请关闭在你场景中的 ambient light

## 5.2 从 Unity 打包至 iPhone

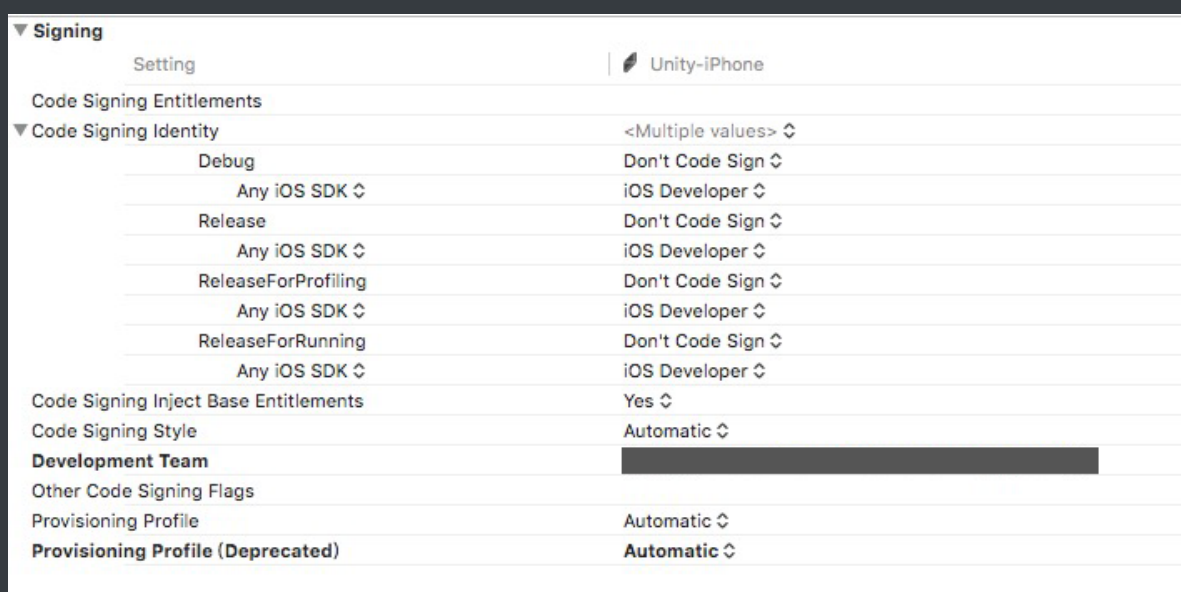
- 当你依照5.1中的步骤打包出一份文件时，这一份即是XCode项目
- 你只需双击该文件夹下的Unity-iPhone.xcodeproj文件既可打开该XCode项目
- 点击如下Unity-iPhone，进行相关设置



- 首先点击上方的"Build Setting"

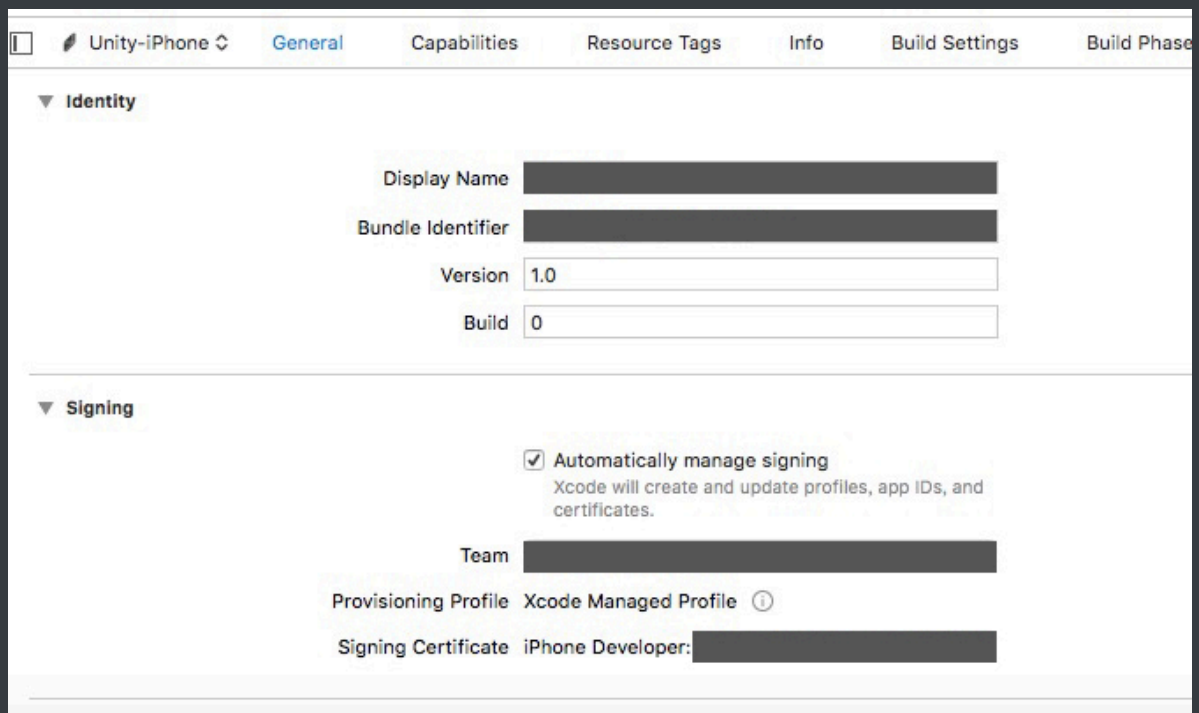


- 找到下面的Signing,如果您是第一次打包XCode项目，请先加入您的开发者签名

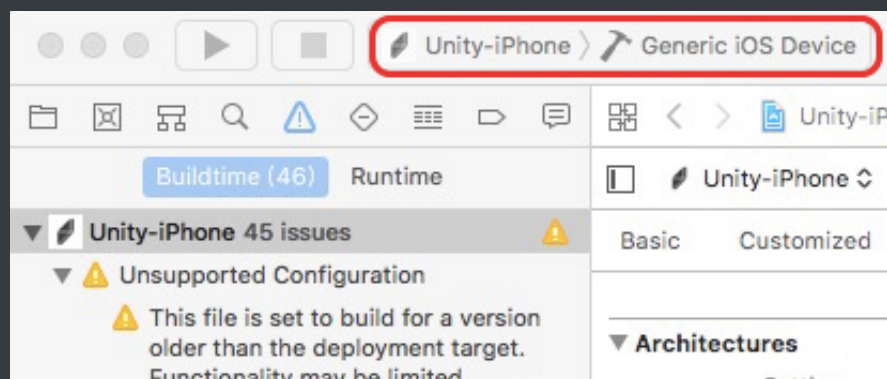


- 点击Code Signing Identity,选中自己的开发者签名

- 之后，点击上方General



- 点击Signing下的 Automatically manage signing，这一项是检查您的Bundle Identifier是否与别人的重复，该项检查需要在联网下进行，如果重复，请修改Bundle Identifier，并且再一次取消再勾选Automatically manage signing
- 如果验证成功，即可开始打包，此时需要使用连接线将iPhone连上mac,并点击左上角的Unity-Iphone选择自己的设备进行打包



- 等待几分钟即可将您的项目打包到您的设备上



## 相关 Q&A

### 1.Q: 如果没有Mac电脑和iPhone该怎么参与开发?

A: 开发过程可以在自己电脑上运行, 我们会在比赛期间提供官方Mac电脑和iPhone供打包测试

### 2.Q: 请问不会Unity开发经验可以参加吗?

A: Unity入门是非常简单的, 只要有编程基础就很好上手, Unity支持C#和JS, 并且可视化操作, 上手简单

### 3.Q: 请问在开发过程中有什么需要注意的地方吗?

A: 颜色方面, 由于HoloKit实现原理是在屏幕中把背景完全切换为黑色, 把模型的光通过折射来显示在人眼中的现实空间, 所以暗色黑色无法显示出来; 蓝色系在HoloKit中会出现半透明效果; 如果有不懂的地方可以咨询现场开发人员

### 4.Q: 请问参加比赛有什么福利吗?

- 我们会提供给每一队参赛选手一套HoloKit开发设备
- 表现优异的团队网易游戏会提供官方支持(资金, 技术支持, 美术支持, 策划支持), 扶持完成AR创新项目
- 表现优异的个人可获得网易游戏Offer
- 优秀作品团队可获得丰富鼓励奖金