

什么是ARKit?

ARKit是苹果推出的移动AR开发框架。使用ARKit，开发者可以创造出富有沉浸感的体验，将虚拟的2D或者3D数字内容和通过摄像头捕捉到的现实场景合二为一。

ARKit跟其它AR开发框架如Vuforia的最大的区别就是，ARKit主要是基于无标记追踪的。是的，你不需要什么追踪卡牌。除此之外，ARKit可以把任何带有A9或更高级别芯片的苹果产品转换成支持无标记追踪的AR设备。想想看，在很短的时间内会有数以亿计的苹果设备通过ARKit升级为AR设备。

ARKit能做什么？

如果你是一个iOS开发老手，而且使用过苹果的其它开发框架，可能会担心需要消耗大量的时间和精力才能让ARKit正常运转。其实不然，ARKit的使用非常简单，对于原生开发来说，只需要添加几行代码就好。ARKit将完成最麻烦的工作，开发者只需要专注于创建具有沉浸感的AR体验就好。

让我们看看ARKit能做些什么吧。

Tracking（追踪）

跟HoloLens类似，ARKit可以实时追踪真实世界空间中的设备运动。它会使用设备所采样的运动数据自动更新虚拟数字内容的坐标。具体实现的原理是通过移动虚拟场景中的虚拟摄像机视角来匹配设备的运动。

Scene understanding(场景感知)

为了获取无标记的追踪，ARKit会创建并管理所检测的表面的虚拟地图。当ARKit检测到新的表面，或者是所检测到表面发生了变化，或者是被删除时，都会触发跟ARKit绑定的特殊事件。通过这种方式，开发者可以很好的控制场景，包括如何让虚拟的数字内容和表面进行交互。

Light estimation(光线评估)

ARKit使用当前视频帧的内容来评估实时的环境光线状况，并自动调整虚拟数字内容的亮度，以便和环境光相匹配。当然，前提是你所使用的虚拟数字内容使用的是PBS材质（基于物理的材质）。关于PBS，我们将在后面的内容中一起来了解。

Scene interaction（场景互动）

ARKit还提供了hit-test（碰撞检测）功能，可以让我们自由的和场景中的3D内容进行互动。更令人激动的是，我们还可以将这种互动和传统的基于多点触摸的手势相结合。而在AR应用中采用这种交互方式也是苹果官方的人机交互指南中所推荐的方式。

当然，考虑到iOS还有其它功能强大的框架，比如Core ML或是Vision，甚至是SiriKit，我们完全可以从用户体验出发，综合使用各种框架打造更好的AR体验。

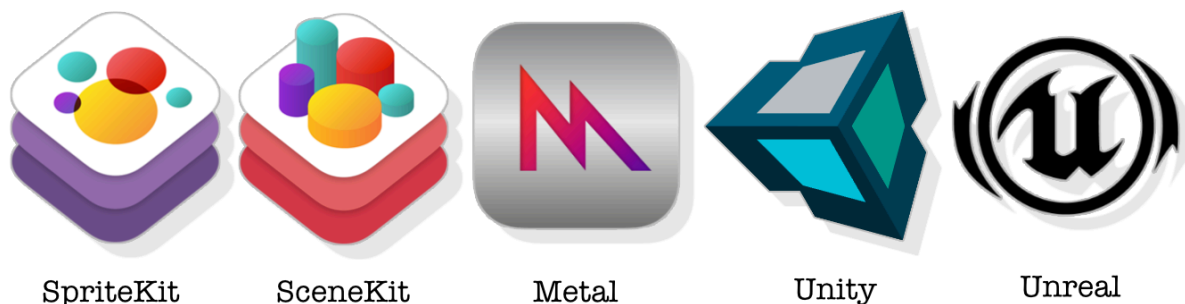
Metric scaling system（公制测量系统）

为了测量场景中的距离和大小，ARKit使用了非常精确的公制测量系统。你不用考虑所使用的是哪种图像API，只需要知道，在ARKit中1个单位就等于1米。

Rendering integrations（渲染框架整合）

ARKit并没有提供自己的图像渲染API，它所提供的只是追踪信息和空间表面检测。因此，如果需要在场景中提供图像信息，我们需要用到SpriteKit,SceneKit或是Metal。

除此之外，最知名的商用游戏开发引擎Unity和Unreal也通过插件的形式提供了对ARKit的支持。



ARKit的局限性

虽然ARKit已经进化到了第二代，但是作为一项新的仍在演化中的技术，当前ARKit仍然存在着各种局限性。我们必须了解ARKit的这些局限性，从而更好的设计和开发iOS平台上的AR应用或游戏。

1.表面检测比较耗时

因为苹果使用的仍然是iOS设备上的传统后置摄像头来获取场景信息，因此表面检测以及真实场景感知需要消耗一定的时间，而这个时间是用户可以明显感觉到的。为此，我们需要通过各种方式降低用户的预期，并在表面检测期间引导用户进行其它的操作，或是耐心

等待。如果不这样做的话，大家辛辛苦苦开发的AR应用游戏很可能就要躺在虚拟世界的空间里面吃土了~

2.运动处理存在延迟

设备运动会带来画面的抖动，同时也让ARKit难以准确感知所检测到的场景。此外，当现实世界的场景晃动的时候，ARKit的运动处理也会消耗更多的时间。因此，当设备存在较大运动幅度的时候，ARKit将不得不针对性的减缓处理速度。

3.弱光环境下的检测效果堪忧

当用户在弱光环境下体验AR应用时，也会遇到很多问题。首先，智能设备上所使用的摄像头没有人眼那么敏锐的感光度。因此，在弱光环境下，ARKit进行场景检测时会遇到很大的困难。最坏的情况下，ARKit可能无法成功分辨出任何表面。如果你作为人类都很难在某个环境下看清东西，ARKit的表现只会十倍百倍的糟糕😓

4.平滑或者表面纹理不明显的表面检测效果堪忧

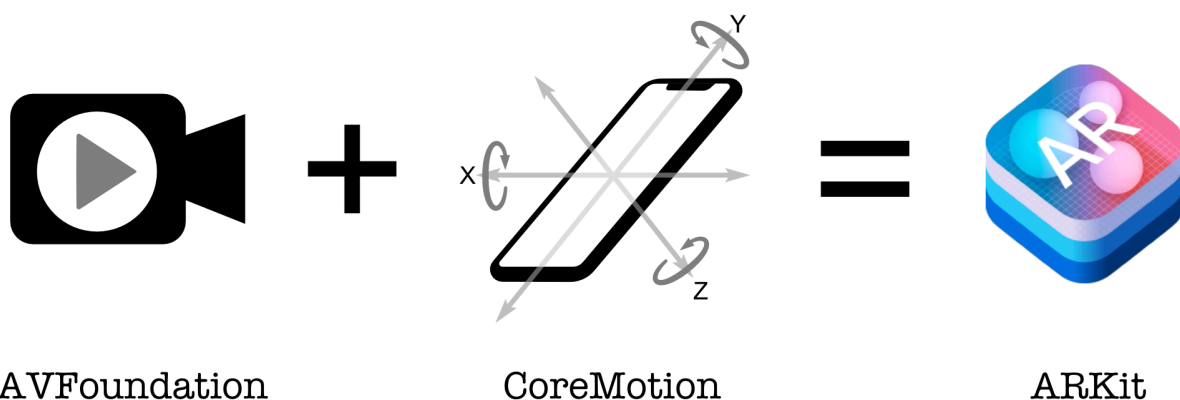
目前来看，ARKit对平滑或者表面纹理不明显的表面检测效果非常糟糕。如果某个表面上的纹理信息越丰富，ARKit就越容易检测并追踪表面的运动。作为开发者来说，我们的任务就是让用户在使用AR应用之前了解到这一局限。否则，当用户想在自己打磨一新黝黑发亮的大理石地板上放置虚拟家具的时候，会感到绝望的。

5.Ghost effect（鬼影）

在使用了AR一段时间后，我们会注意到某些虚拟内容的显示似乎违背了物理规律。例如，虚拟的数字宠物可能会穿透真实世界的物体，就像幽灵一样穿过一扇紧闭的大门。类似这样的情境会严重破坏AR体验的效果。因此，我们需要鼓励用户尽量在明亮开放的空间，或是在一个干净整洁的桌子上面体验AR应用。

ARKit背后的技术

当我们深入探究ARKit背后的技术时，会发现很多东西其实在之前都已经接触过，而且也已经应用了相当长的时间。



这张图告诉我们，ARKit其实没那么神奇，它只是将传统的AVFoudnation框架和CoreMotion框架结合在一起，然后添加一些其它的功能而已。

ARKit使用VIO(Visual Inertial Odometry，也就是所谓的视觉惯性里程计技术)来追踪设备的运动，并感知真实世界的空间。VIO融合了来自摄像头传感器的基于AVFoundation的输入信息，以及通过CoreMotion获取的设备运动信息。如果你之前对这两个框架有一定的了解，那么很容易就会明白ARKit识别和追踪场景的技术原理。

当然，如果你之前没有不太了解这两个框架，也不要恐慌。Don't panic!我们会带着大家来一起一步步了解ARKit。

ARKit的学习资源

虽然ARKit还是一个全新的不断演化中的技术框架，但是其广阔的应用场景和酷炫的效果还是很快吸引了大量的开发者。

与此同时，苹果官方也提供了大量的技术资料，方便开发者快速入门和掌握ARKit。

1.苹果官方网站上的ARKit文档

Apple Developer Documentation for ARKit: <https://developer.apple.com/documentation/arkit>

2.苹果官网上的AR应用人机交互指南

Human Interface Guidelines for AR: <https://developer.apple.com/design/human-interface-guidelines/ios/system-capabilities/augmented-reality/>

3.WWDC 2018上关于ARKit的视频

What's New in ARKit 2: <https://developer.apple.com/videos/play/wwdc2018/602/>

Understanding ARKit Tracking and Detection: <https://developer.apple.com/videos/play/wwdc2018/610/>

Integrating Apps and Content with AR Quick Look: <https://developer.apple.com/videos/play/wwdc2018/603/>

Inside SwiftShot: Creating an AR Game: <https://developer.apple.com/videos/play/wwdc2018/605/>

Platforms State of the Union: <https://developer.apple.com/videos/play/wwdc2018/102/>

4.ARKit的官方demo

Interactive Content with ARKit: https://developer.apple.com/library/archive/samplecode/InteractiveContent/Introduction/Intro.html#//apple_ref/doc/uid/TP40017667-Intro-DontLinkElementID_2

SwiftShot: Creating a Game for Augmented Reality: https://developer.apple.com/documentation/arkit/swiftshot_creating_a_game_for_augmented_reality

小结:

在本章的内容中，我们已经学到了以下指示：

- 1.什么是AR
- 2.传统的基于标记的无标记追踪技术之前的区别
- 3.什么是ARKit，它能干什么，以及不能干什么。
- 4.ARkit不是一个图像处理的类库，它是一个框架，可以帮开发者轻松实现AR应用。

在下一章的内容中，我们将开始学习如何在Xcode中创建你的第一款ARKit应用。

让我们下一课再见~