# ARKit入门

——AR在iOS平台上的使用

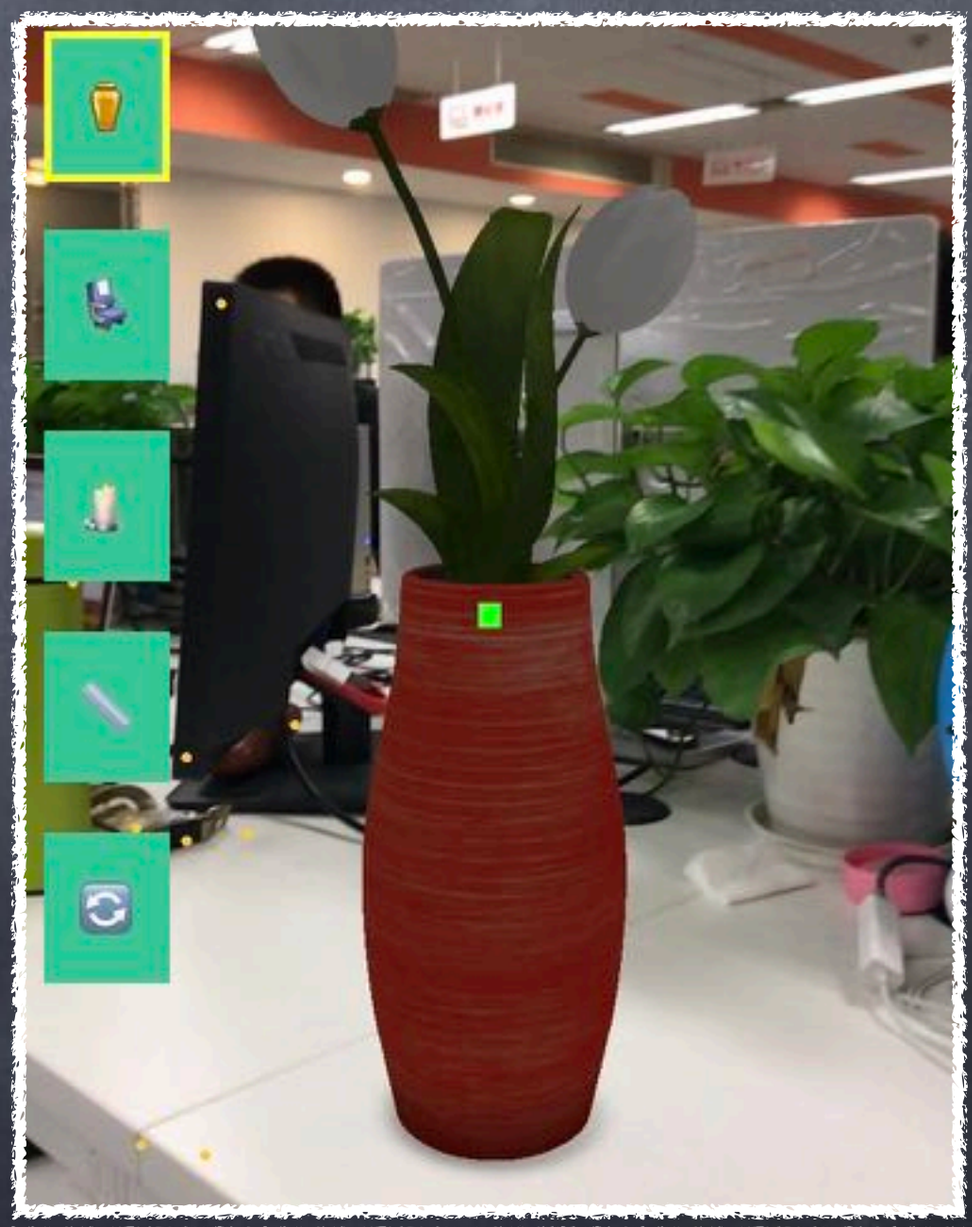
# 内容

- AR是什么
- ARKit是什么
- ARKit设备要求
- AR项目创建和ARDemo演示
- AR工作原理和工作流程
- ARKit的API介绍
- more

AR

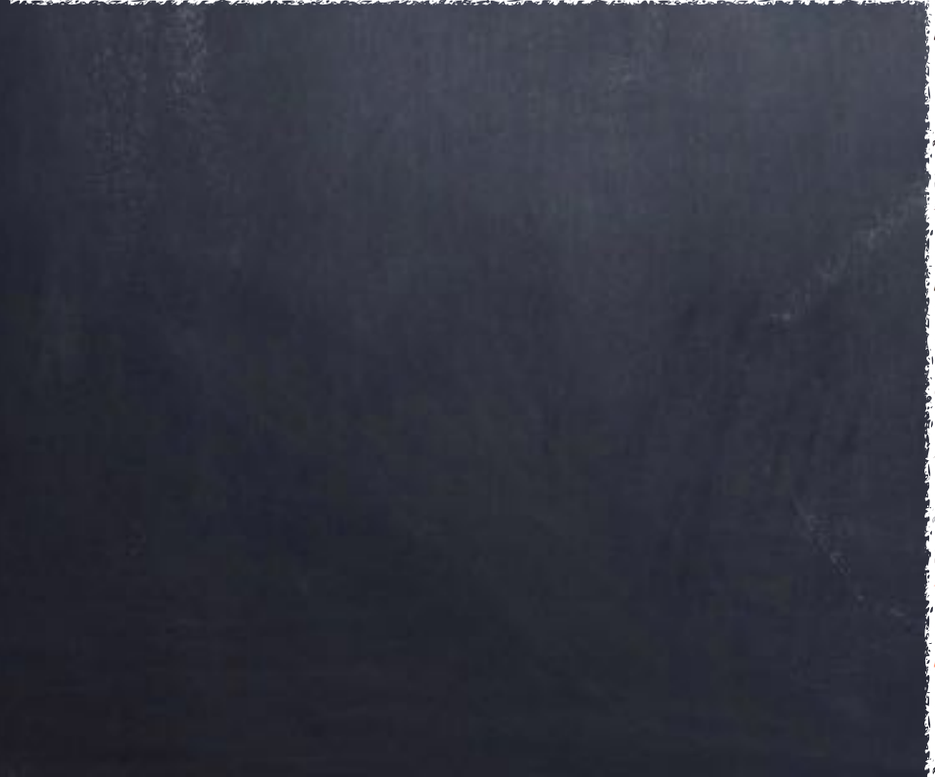
# AR

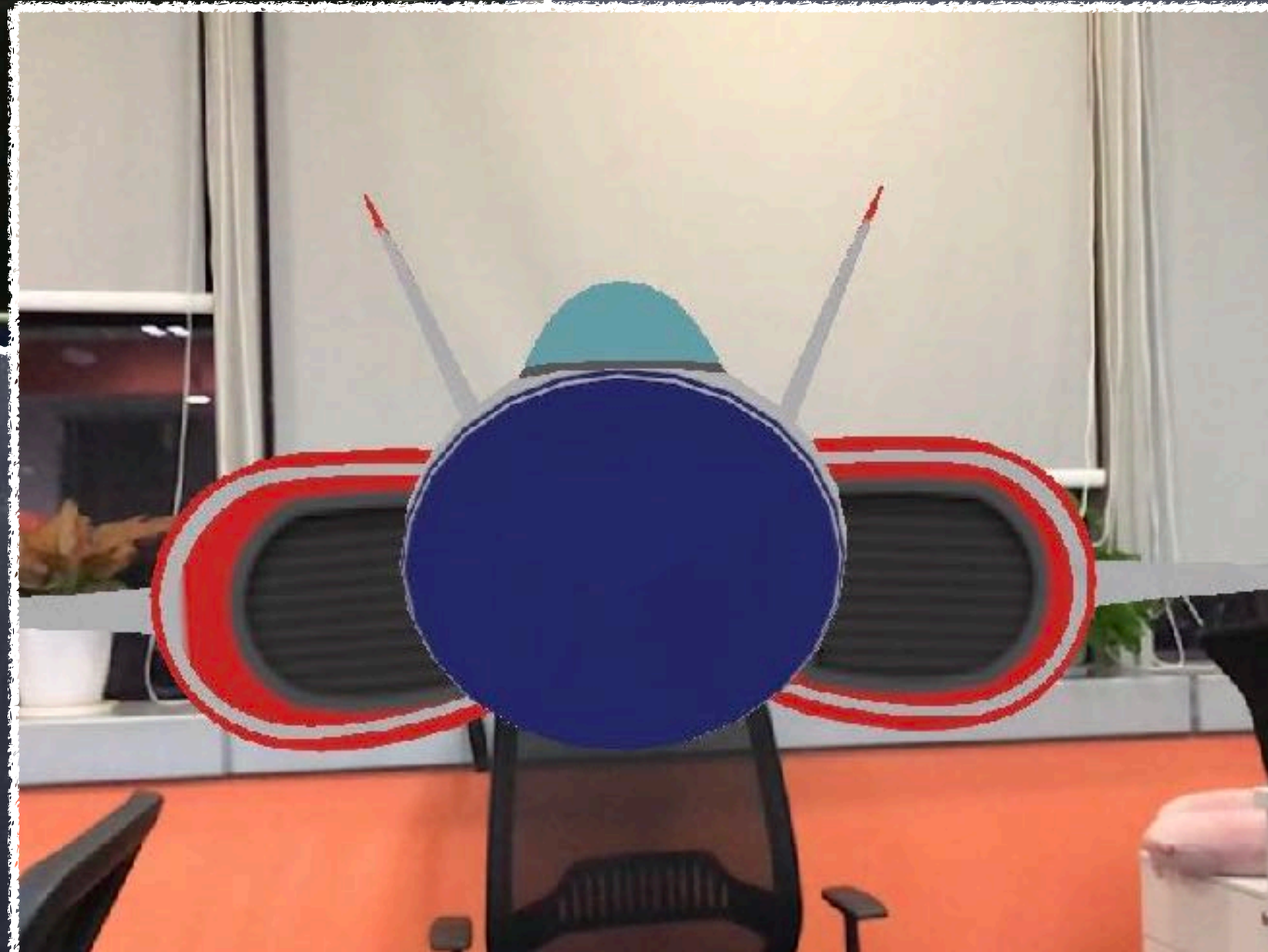
增强现实技术（Augmented Reality，简称 AR），是一种实时地计算摄影机影像的位置及角度并加上相应图像、视频、3D模型的技术，这种技术的目标是在屏幕上把虚拟世界套在现实世界并进行互动。




中国联通
下午7:03
100%
back
Mt 59fps
♦5 ▲4.73K


下午7:03
back
Mt 60fps
♦4 ▲5.59K





# AR实现技术和步骤

AR场景实现所需要的技术和步骤一般包括以下几部分：

- 捕获现实环境的图像：如摄像头
- 三维建模：3D模型的建立
- 传感器追踪：主要追踪现实世界动态物体的六轴变化，这六轴分别是X、Y、Z轴位移及旋转。其中位移三轴决定物体的方位和大小，旋转三轴决定物体显示的区域。
- 坐标识别及转换：3D模型显示在现实图像中不是单纯的frame坐标点，而是一个三维的矩阵坐标。
- 交互：AR还可以与虚拟物体进行交互

# ARKit

# ARKit

- ARKit是2017年6月6日，苹果发布iOS11系统所新增框架,它能够帮助我们以最简单快捷的方式实现AR技术功能。
- ARKit框架提供了两种AR技术，一种是基于3D场景(SceneKit)实现的增强现实，一种是基于2D场景(SpriktKit)实现的增强现实。
- 要想显示AR效果，必须要依赖于苹果的游戏引擎框架（3D引擎SceneKit，2D引擎SpriktKit），主要原因是游戏引擎才可以加载物体模型。
- 虽然ARKit框架中视图对象继承于UIView，但是由于目前ARKit框架本身只包含相机追踪，不能直接加载物体模型，所以只能依赖于游戏引擎加载ARKit。

# 那ARKit究竟是啥

# ARKit

NEW

Mobile AR platform

High-level API

IOS (A9 and up)

移动端的AR平台

高级API

iOS设备：处理器A9及以上

# ARKit三层核心技术


快速稳定的世界定位 ， 包括实时运算，运动定位，无需预设（软硬件）

# ARKit三层核心技术


平面和边界感知 碰撞测试和光线估算，让虚拟内容和现实环境无缝衔接。

# ARKit三层核心技术


支持各种渲染制作工具，目标就是简单易用，和其它插件融合度好。

# ARKit应用架构和使用


苹果AR工程师总结起来也是超级好用：

首先创建一个ARSession。

然后就是设置你的ARsession configuration。

# AR设备要求

# AR环境要求

- Xcode版本：Xcode9及以上
- iOS系统：iOS11及以上
- iOS设备：处理器A9及以上（6S机型及以上）
- MacOS系统：10.12.4及以上（安装Xcode9对Mac系统版本有要求）

# AR项目创建和ARDemo演示

# AR项目创建

- 打开Xcode9版本，新建一个工程，选择Augmented Reality APP (Xcode9新增),点击next。
- 项目包含技术选择SceneKit（3D）,如果2D选择SpriteKit。
- 输入项目名字选择保存，此时,Xcode会自动为我们生成一段极其简洁的AR代码。

# ARDemo演示

- 3D游戏模版Demo
- AR2D模版Demo
- AR3D模版Demo

# ARDemo演示

- 3D场景里面模型旋转 Demo
- 太阳系 Demo
- 3D模型跟随移动 Demo
- AR识别平地 Demo
- HOMEHero Demo
- 全景照片查看 Demo
- 全景视频播放 Demo

# AR工作原理和执行流程

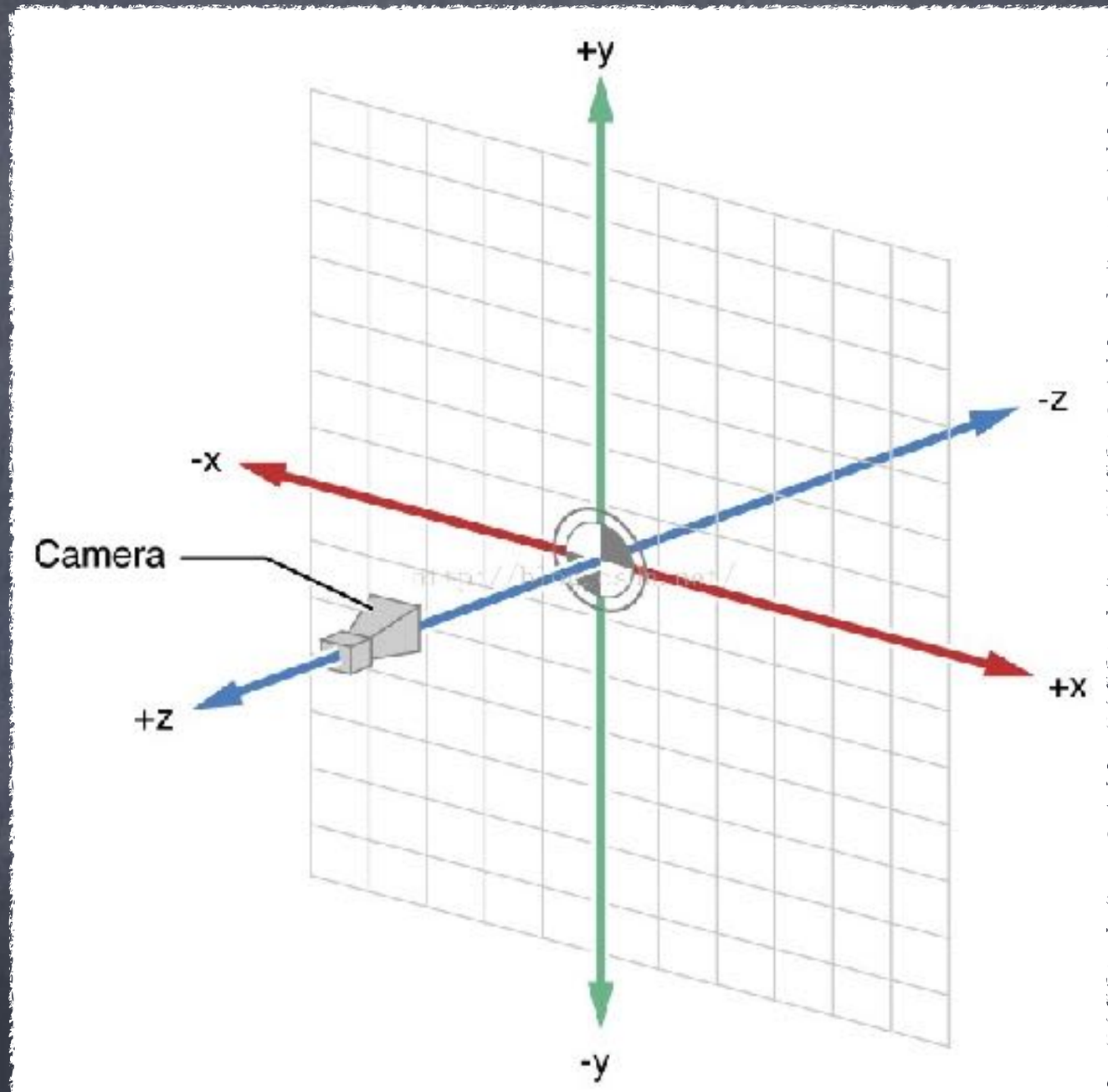
# AR工作原理

- 初次接触 ARKit ,很多人会为其复杂的架构关系而感到畏惧。这里简介的方式介绍一下苹果原生AR（虚拟增强现实）， ARKit并不是一个独立就能够运行的框架，而是必须要SceneKit一起用才可以，换一句话说，如果只有ARKit，而没有SceneKit，那么ARKit和一般的相机没有任何区别。两者配合实现AR功能。

- ARkit主要实现相机捕捉显示世界图像，SceneKit主要负责显示虚拟3D模型。

# 前提一：了解SceneKit三维坐标系

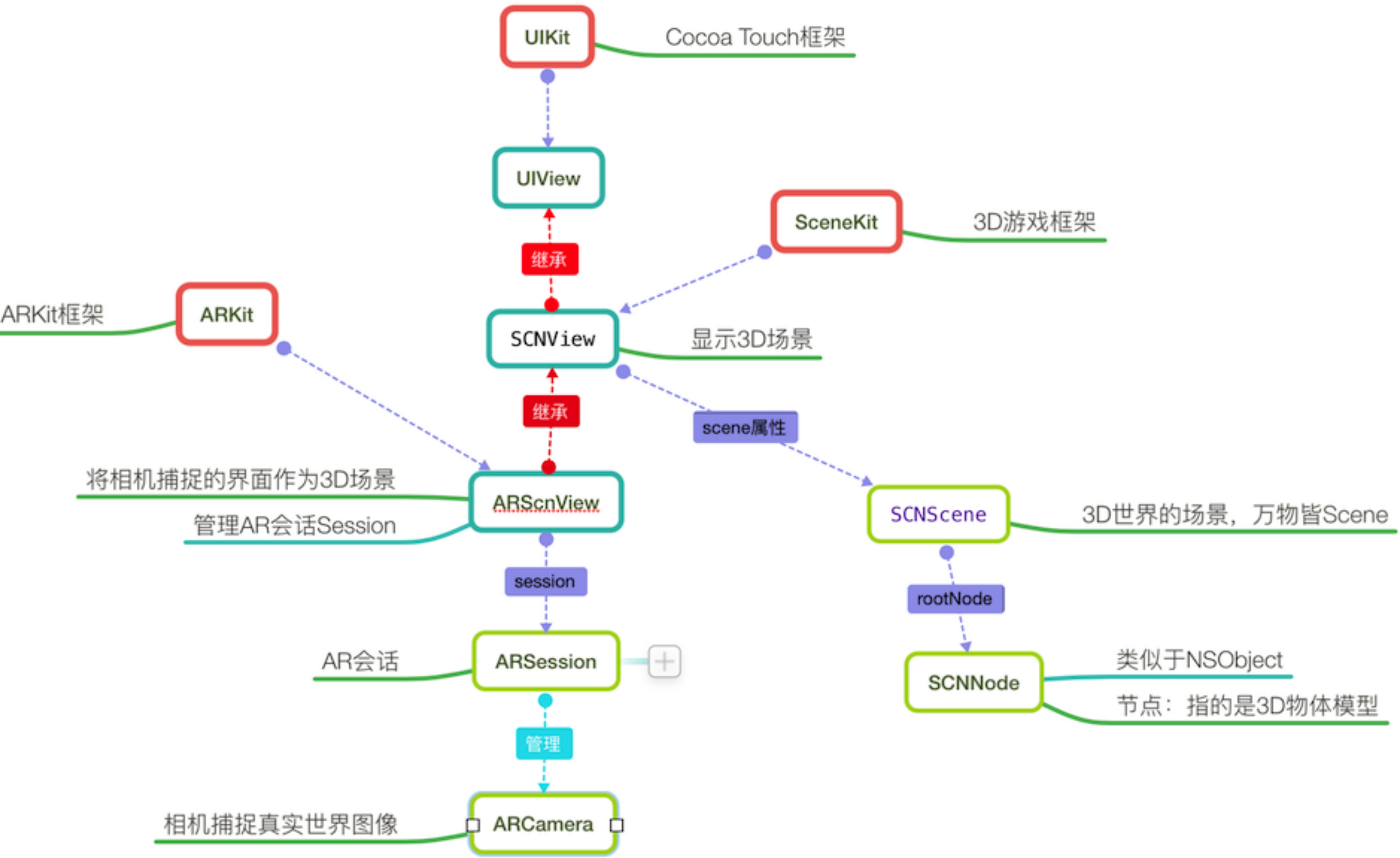
很清楚的看到，SceneKit 中的坐标系是右手坐标系(笛卡尔坐标系)，如果需要与其他3D框架共享数据，先了解其框架是右手坐标系还是左手坐标系。其实也很好转化，就是Z 轴的正负不一样而已。



# 前提二：SceneKit重要类

| 类 / 协议 | 描述 |
| --- | --- |
| SCNView & SCNSceneRenderer | 类似UIView，用来显示 SceneKit 的内容，定义了一些代理方法，可以用 addSubView 方法添加到 UiView 中. |
| SCNScene | SceneKit 内容的容器. 你可以从3D建模工具生成的.dae文件中加载一个场景，或者用代码创建一个，然后把它显示在视图上. |
| SCNNode | 一个场景的基本构建块，你可以把摄像机，灯光，几何体附加到节点上 |
| SCNGeometry | 一个可以连接到一个节点的三维物体。一个几何体（有时称为模型或网格）只定义了一个可见物体的形状。要定义对象的表面颜色图案，你必需要给几何体要附加材料。然后给材料贴图，或者上色，这个几何体表面才会有颜色，或者图案。你可以从3D建模工具生成的.dae文件中加载一个几何体，也可以用代码创建，SceneKit 提供了几种常见几何体，是SCNGeometry 的子类，比如长方体，球，圆柱球等等，后面我们会写一个demo会把官方提供的几何体给大家列出来，给大家一个直观的感受。 当然我们也可以用三维坐标，法向量自定义几何体，也可以讲一个2D 图案转化成一个具有深度(厚度)的三维几何体。后面应该专门有一篇会讲到利用贝塞尔曲线将一个2D 图案转化成一个具有深度(厚度)的三维几何体。 |
| CNMaterial | 材质，由于在3D建模工具中呈现球形，所以也叫材质球。上色，贴图全靠它。 |
| SCNLight | 光源可以附加到节点上，在渲染场景中提供着色. |
| SCNCamera | 虚拟摄像机可以附加到节点上，提供了一个场景的视图。 |

# ARKit和SceneKit关系



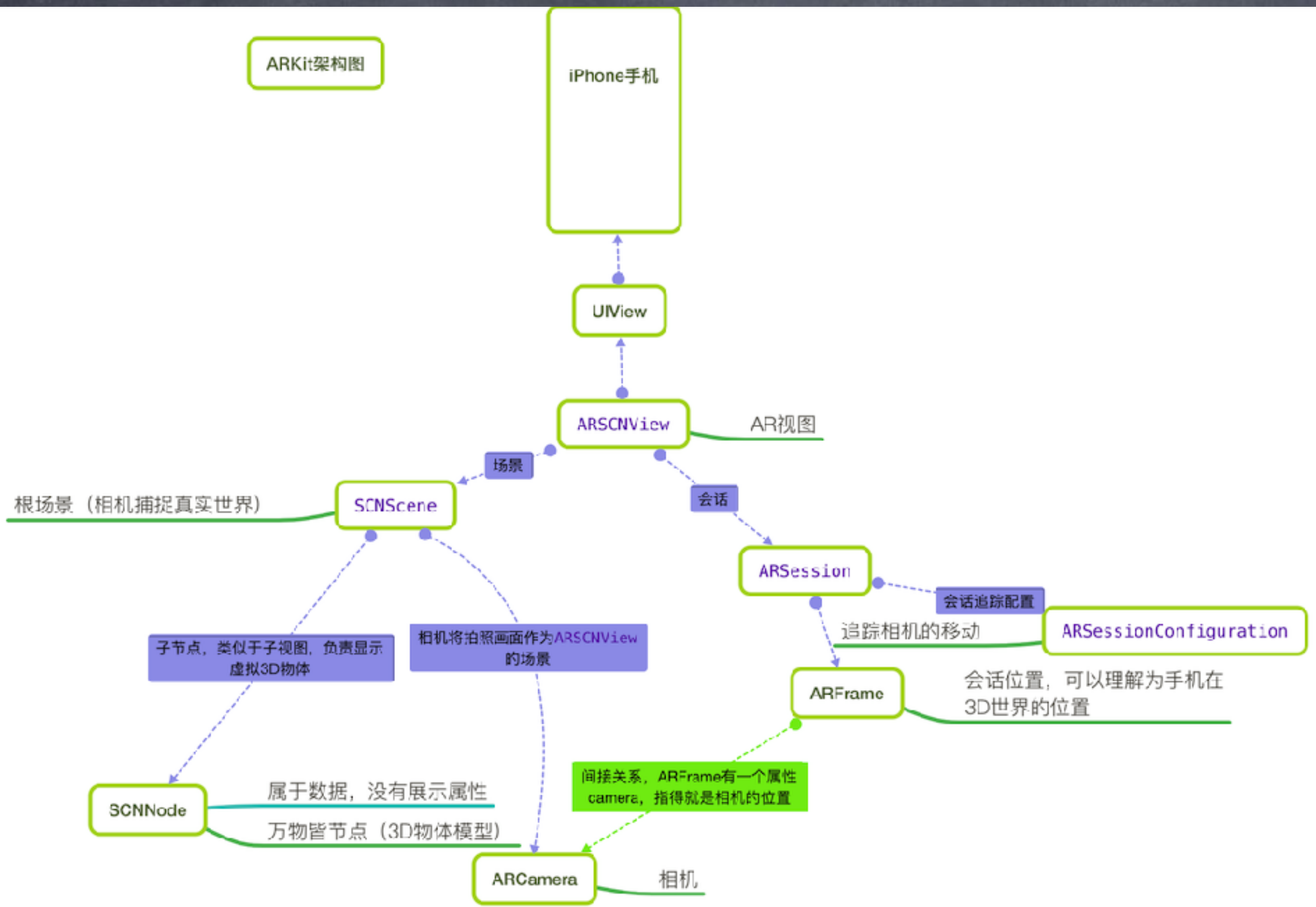
# ARKit和SceneKit关系

- ARKit框架中中显示3D虚拟增强现实的视图ARSCNView继承于SceneKit框架中的SCNView,而SCNView又继承于UIKit框架中的UIView。
- UIView的作用是将视图显示在iOS设备的window中，SCNView的作用是显示一个3D场景，ARSCNView的作用也是显示一个3D场景，只不过这个3D场景是由摄像头捕捉到的现实世界图像构成的。
- ARSCNView只是一个视图容器，它的作用是管理一个ARSession。
- 在一个完整的虚拟增强现实体验中，ARKit框架只负责将真实世界画面转变为一个3D场景，这一个转变的过程主要分为两个环节：由ARCamera负责捕捉摄像头画面，由ARSession负责搭建3D场景。
- 在一个完整的虚拟增强现实体验中，将虚拟物体现实在3D场景中是由SceneKit框架来完成中：每一个虚拟的物体都是一个节点SCNNode,每一个节点构成了一个场景SCNScene无数个场景构成了3D世界。

# ARKit和SceneKit关系（总结）

- ARKit捕捉3D现实世界使用的是自身的功能，这个功能是在iOS11新增的。而ARKit在3D现实场景中添加虚拟物体使用的是父类SCNView的功能，这个功能早在iOS8时就已经添加（SceneKit是iOS8新增），可以简单的理解为：ARSCNView所有跟场景和虚拟物体相关的属性及方法都是自己父类SCNView的。

# AR工作流程


ARKit架构图
iPhone手机
UIView
ARSCNView
AR视图
场景
会话
根场景（相机捕捉真实世界）
SCNScene
ARSession
会话追踪配置
ARSessionConfiguration
追踪相机的移动
子节点，类似于子视图，负责显示虚拟3D物体
相机将拍照画面作为ARSCNView的场景
ARFrame
会话位置，可以理解为手机在3D世界的位置
间接关系，ARFrame有一个属性camera，指得就是相机的位置
SCNNode
属于数据，没有展示属性
万物皆节点（3D物体模型）
ARCamera
相机

# AR工作流程

- ARSCNView加载场景SCNScene。
- SCNScene启动相机ARCamera开始捕捉场景。
- ARCamera捕捉场景后，ARSCNView开始将场景交给Session。
- Session通过管理ARSessionConfiguration实现场景追踪并返回ARFrame。
- 给ARSCNView的Scene添加一个子节点（3D物体模型）。
- ARSessionConfiguration捕捉相机3D位置的意义就在于能够在添加3D物体模型的时候计算出3D物体模型相对于相机的真实的矩阵位置。

# ARKit的API介绍

# ARKit的API介绍

- ARAnchor
- ARCamera
- ARError
- ARFrame
- ARHitTestResult
- ARLightEstimate
- ARPlaneAnchor

- ARPointCloud
- ARSCNView
- ARSession
- ARSessionConfiguration

more

# SceneKit旋转

- 每个SCNNode 都有自身的三维坐标系，用CABasicAnimation就是让SCNNode绕自身的三维坐标轴旋转，所以要特别注意是坐标轴，不是这个SCNNode的几何中心。一般SceneKit 的自带的几个几何体的坐标系原点(0,0,0)就是这个它的几何中心，比如说SCNBox；SCNSphere等等，所以看上去跟绕几何中心旋转一模一样。

- CABasicAnimation

```
// Rotate the moon

    animation = [CABasicAnimation
animationWithKeyPath:@"rotation"];

    animation.duration = 1.5;

    animation.toValue = [NSValue
valueWithSCNVector4:SCNVector4Make(0, 1, 0, M_PI * 2)];

    animation.repeatCount = FLT_MAX;

    [_moonNode addAnimation:animation forKey:@"moon rotation"];
```

- 其实SceneKit 框架有自己的动画API

```
[_earthNode runAction:[SCNAction repeatActionForever:[SCNAction
rotateByX:0 y:2 z:0 duration:1]]];
```

# 其他问题

- 图片和视频全景展示
- SceneKit高级使用（粒子系统、碰撞检测）
- AR使用场景和后续发展。。。

# Thanks!

# 资料地址

- 文档：http://10.10.13.12/showdoc/index.php/Home/Item/show?item_id=21
- demo：http://10.10.13.28/AllenMu/ARDemo.git