

ARKit从入门到精通（4）-ARKit全框架API大全



坤小 (/u/b5c2ad9cbe7b) [+ 关注](#)

2017.06.12 20:32* 字数 1093 阅读 4497 评论 9 喜欢 18 阅读 4497 评论 9 喜欢 18

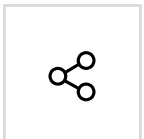
(/u/b5c2ad9cbe7b)

- 转载请注明出处：<http://www.jianshu.com/p/3c7ee58d40f4>
(<https://www.jianshu.com/p/3c7ee58d40f4>)

- 1.1-ARKit框架简介
- 1.2-ARAnchor
- 1.3-ARCamera
- 1.4-ARError
- 1.5-ARFrame
- 1.6-ARHitTestResult
- 1.7-ARLightEstimate
- 1.8-ARPlaneAnchor
- 1.9-ARPointCloud
- 1.10-ARSCNView
- 1.11-ARSession
- 1.12-ARSessionConfiguration
- 1.13-ARSKView

<h2 id="1.1">1.1-ARKit框架简介

- 再上一小节中，相信大家已经对 <ARKit> 框架的使用及原理有了一个全局的认识，为了能够更加深入的研究学习 ARKit 技术，所以笔者本小节主要介绍 <ARKit> 框架中所有的API
 - 本篇主要翻译自苹果官方文档，适当加上一些笔者的见解
- ARKit框架类图





0501.png

<h2 id="1.2">1.2-ARAnchor</h2>

- *ARAnchor*表示一个物体在3D空间的位置和方向（*ARAnchor*通常称为物体的3D锚点，有点像UIKit框架中CALayer的Anchor）
 - *ARFrame*表示的也是物体的位置和方向，但是*ARFrame*通常表示的是AR相机的位置和方向以及追踪相机的时间，还可以捕捉相机的帧图片
 - 也就是说ARFrame用于捕捉相机的移动，其他虚拟物体用ARAnchor

```
@interface ARAnchor : NSObject <NSCopying>

/**
 标识符
 */
@property (nonatomic, readonly) NSUUID *identifier;

/**
 锚点的旋转变换矩阵，定义了锚点的旋转、位置、缩放。是一个4x4的矩阵（读者可以自行科普什么叫4x4矩阵）
 */
@property (nonatomic, readonly) matrix_float4x4 transform;

/**
 构造方法，一般我们无需构造。因为添加一个3D物体时ARKit会有代理告知我们物体的锚点
 */
- (instancetype)initWithTransform:(matrix_float4x4)transform;

@end
```

<h2 id="1.3">1.3-ARCamera</h2>

- *AR相机，该类API较多，并且理解比较困难，将会在后续小节单独介绍
 - 该类非常重要，且API较多，将在后续小节介绍

<h2 id="1.4">1.4-ARError</h2>

- *ARError*是一个描述ARKit错误的类，这个错误来源于几个方面，例如设备不支持，或者当相机常驻后台时ARSession会断开等问题

```
//作用域，一般会表示是哪一个类出现问题
NSString *const ALErrorDomain;

//错误码描述 100: 不支持会话追踪配置，主线由于A9芯片以下的机型会报错 101: 失活状态 102: 传感器故障 200: 追踪失败
typedef NS_ERROR_ENUM(ALErrorDomain, ALErrorCode) {
    /** Unsupported session configuration. */
    ALErrorCodeUnsupportedConfiguration = 100,

    /** A sensor required to run the session is not available. */
    ALErrorCodeSensorUnavailable = 101,

    /** A sensor failed to provide the required input. */
    ALErrorCodeSensorFailed = 102,

    /** World tracking has encountered a fatal error. */
    ALErrorCodeWorldTrackingFailed = 200,
};
```

<h2 id="1.5">1.5-ARFrame</h2>

- **ARFrame**主要是追踪相机当前的状态，这个状态不仅仅只是位置，还有图像帧及时
间等参数

```
@interface ARFrame : NSObject <NSCopying>

/**
时间戳。
*/
@property (nonatomic, readonly) NSTimeInterval timestamp;

/**
缓冲区图像帧
*/
@property (nonatomic, readonly) CVPixelBufferRef capturedImage;

/**
相机（表示这个ARFrame是哪一个相机的，iPhone7plus有两个摄像机）
*/
@property (nonatomic, copy, readonly) ARCamera *camera;

/**
返回当前相机捕捉到的锚点数据（当一个3D虚拟模型加入到ARKit中时，锚点值得就是这个模型在AR中的位置）
*/
@property (nonatomic, copy, readonly) NSArray<ARAnchor *> *anchors;

/**
灯光，详情可见本章节ARLightEstimate类介绍（指的是灯光强度 一般是0-2000，系统默认1000）
*/
@property (nonatomic, copy, nullable, readonly) ARLightEstimate *lightEstimate;

/**
特征点（应该是捕捉平地或者人脸的，比较苹果有自带的人脸识别功能）
*/
@property (nonatomic, nullable, readonly) ARPointCloud *rawFeaturePoints;

/**
根据2D坐标点搜索3D模型，这个方法通常用于，当我们在手机屏幕点击某一个点的时候，可以捕捉到这一个点所在的3D模型的位置，至于为什么是一个数组非常好理解。手机屏幕一个是长方形，这是一个二维空间。而相机捕捉到的是一个由这个二维空间射出去的长方体，我们点击屏幕一个点可以理解为在这个长方体的边缘射出一条线，这一条线上可能会有多个3D物体模型
point: 2D坐标点（手机屏幕某一点）
ARHitTestResultType: 捕捉类型 点还是面
(NSArray<ARHitTestResult *> *): 追踪结果数组 详情见本章节ARHitTestResult类介绍

*/
- (NSArray<ARHitTestResult *> *)hitTest:(CGPoint)point types:(ARHitTestResultType)types;

/**
相机窗口的的坐标变换（可用于相机横竖屏的旋转适配）
*/
- (CGAffineTransform)displayTransformWithViewportSize:(CGSize)viewportSize orientation:(UIInterfaceOrientation)orientation;

@end
```

<h2 id="1.6">1.6-ARHitTestResult</h2>

- **ARHitTestResult**：点击回调结果，这个类主要用于虚拟增强现实技术（AR技术）中现实世界与3D场景中虚拟物体的交互。比如我们在相机中移动。拖拽3D虚拟物体，都可以通过这个类来获取ARKit所捕捉的结果

```
//捕捉类型枚举
typedef NS_OPTIONS(NSUInteger, ARHitTestResultType) {
    /** 点。*/
    ARHitTestResultTypeFeaturePoint                = (1 << 0),

    /** 水平面 y为0。*/
    ARHitTestResultTypeEstimatedHorizontalPlane    = (1 << 1),

    /** 已结存在的平面。*/
    ARHitTestResultTypeExistingPlane               = (1 << 3),

    /** 已结存在的锚点和平面。*/
    ARHitTestResultTypeExistingPlaneUsingExtent    = (1 << 4),
} NS_SWIFT_NAME(ARHitTestResult.ResultType);

/**
捕捉类型
*/
@property (nonatomic, readonly) ARHitTestResultType type;

/**
3D虚拟物体与相机的距离（单位：米）
*/
@property (nonatomic, readonly) CGFloat distance;

/**
本地坐标矩阵（世界坐标指的是相机为场景原点的坐标，而每一个3D物体自身有一个场景，本地坐标就是相对于这个场景的坐标）类似于frame和bounds的区别
*/
@property (nonatomic, readonly) matrix_float4x4 localTransform;

/**
世界坐标矩阵
*/
@property (nonatomic, readonly) matrix_float4x4 worldTransform;

/**
锚点（3D虚拟物体，在虚拟世界有一个位置，这个位置参数是SceneKit中的SCNVector3：三维矢量），而锚点anchor是这个物体在AR现实场景中的位置，是一个4x4的矩阵
*/
@property (nonatomic, strong, nullable, readonly) ARAnchor *anchor;

@end
```

<h2 id="1.7">1.7-ARLightEstimate</h2>

- **ARLightEstimate**是一个灯光效果，它可以让你的AR场景看起来更加的好

```
@interface ARLightEstimate : NSObject <NSCopying>

/**
灯光强度 范围0-2000 默认1000
*/
@property (nonatomic, readonly) CGFloat ambientIntensity;

@end
```

<h2 id="1.8">1.8-ARPlaneAnchor</h2>

- **ARPlaneAnchor**是ARAnchor的子类，笔者称之为平地基点。ARKit能够自动识别平地，并且会默认添加一个锚点到场景中，当然要想看到真实世界中的平地效果，需要我们自己使用SCNNode来渲染这个锚点
 - 锚点只是一个位置

```
/**
平地类型，目前只有一个，就是水平面
*/
@property (nonatomic, readonly) ARPlaneAnchorAlignment alignment;

/**
3轴矢量结构体，表示平地的中心点  x/y/z
*/
@property (nonatomic, readonly) vector_float3 center;

/**
3轴矢量结构体，表示平地的大小（宽度和高度）  x/y/z
*/
@property (nonatomic, readonly) vector_float3 extent;

@end
```

<h2 id="1.9">1.9-ARPointCloud</h2>

- *ARPointCloud*：点状渲染云，主要用于渲染场景

```
@interface ARPointCloud : NSObject <NSCopying>

/**
点的数量
*/
@property (nonatomic, readonly) NSUInteger count;

/**
每一个点的位置的集合（结构体带*表示的是结构体数组）
*/
@property (nonatomic, readonly) const vector_float3 *points;

@end
```

<h2 id="1.10">1.10-ARSCNView</h2>

- *AR视图*，在第一小节笔者介绍过，*ARKit*支持3D的AR场景和2D的AR场景，*ARSCNView*是3D的AR场景视图
 - 该类非常重要，且API较多，将在后续小节介绍
- 该类是整个*ARKit*框架中唯一两个有代理的类其中之一

<h2 id="1.11">1.11-ARSession</h2>

- *AR会话*，它的作用已经在前面小节中介绍，这里不再累述
 - 该类非常重要，且API较多，将在后续小节介绍
- 该类是整个*ARKit*框架中唯一两个有代理的类其中之一

<h2 id="1.12">1.12-ARSessionConfiguration</h2>

- *ARSessionConfiguration*会话追踪配置，主要就是追踪相机的配置
 - 注意：该类还有一个子类：*ARWorldTrackingSessionConfiguration*，它们在同一个API文件中

```
//会话追踪配置类
@interface ARSessionConfiguration : NSObject <NSCopying>

/**
当前设备是否支持，一般A9芯片以下设备不支持
*/
@property(class, nonatomic, readonly) BOOL isSupported;

/**
会话的对其方式，这里的对其指的是3D世界的坐标。枚举值见下方
*/
@property (nonatomic, readwrite) ARWorldAlignment worldAlignment;

/**
是否需要自适应灯光效果，默认是YES
*/
@property (nonatomic, readwrite, getter=isLightEstimationEnabled) BOOL lightEstimationEnabled;

@end


//世界会话追踪配置，苹果建议我们使用这个类，这个子类只有一个属性，也就是可以帮助我们追踪相机捕捉到的平地
@interface ARWorldTrackingSessionConfiguration : ARSessionConfiguration

/**
侦查类型。枚举值见下方（默认侦查平地）
*/
@property (nonatomic, readwrite) ARPlaneDetection planeDetection;

@end
```

```
//追踪对其方式，这个决定了会话的参考坐标系（参照物）
typedef NS_ENUM(NSInteger, ARWorldAlignment) {
/** 相机位置 vector (0, -1, 0) */
ARWorldAlignmentGravity,
```

```
/** 相机位置及方向。 vector (0, -1, 0)
    heading : (0, 0, -1) */
ARWorldAlignmentGravityAndHeading,

/** 相机方向。 */
ARWorldAlignmentCamera
```

```
} NS_SWIFT_NAME(ARSessionConfiguration.WorldAlignment);
```

```
/**
侦查类型
/
API_AVAILABLE(ios(11.0)) API_UNAVAILABLE(macos, watchos, tvos)
typedef NS_OPTIONS(NSUInteger, ARPlaneDetection) {
/* 不侦查. */
ARPlaneDetectionNone = 0,
```

```
/** 平地侦查 */
ARPlaneDetectionHorizontal = (1 << 0),
```

```
} NS_SWIFT_NAME(ARWorldTrackingSessionConfiguration.PlaneDetection);
```

<h2 id="1.13">1.13-ARSKView</h2>

- **ARSKView也是AR视图，只不过他是2D的，由于2D比3D简单很多，并且ARSKView基本与ARSCNView类似，所以这里不做重复介绍。详细内容可参考ARSCNView**



坤小 (/u/b5c2ad9cbe7b) ♂

写了 17710 字, 被 752 人关注, 获得了 279 个喜欢

(/u/b5c2ad9cbe7b) 写了 17710 字, 被 752 人关注, 获得了 279 个喜欢

+ 关注

♡ 喜欢 (/sign_in?utm_source=desktop&utm_medium=not-signed-in-like-button) | 18







更多分享

(http://cwb.assets.jianshu.io/notes/images/1339652



(/apps/download?utm_source=nbc)


被以下专题收入, 发现更多相似内容




iOS (/c/f30dbbf00b59?utm_source=desktop&utm_medium=notes-included-collection)




ARKit (/c/20e4f0253137?utm_source=desktop&utm_medium=notes-included-collection)




ARKit (/c/b4ab693c3049?utm_source=desktop&utm_medium=notes-included-collection)




iOS ARKit专题 (/c/bf3e8e31d698?utm_source=desktop&utm_medium=notes-included-collection)



ARKit (/c/3682baaef8ae?utm_source=desktop&utm_medium=notes-included-collection)



Unity技术... (/c/d860bab59d8f?utm_source=desktop&utm_medium=notes-included-collection)



AR/VR/MR (/c/eb7b05e795cb?utm_source=desktop&utm_medium=notes-included-collection)

展开更多 ▾

推荐阅读

更多精彩内容 > (/)

WatchOS系统开发大全（5）-WKInterfaceController... (/p/5ff63f6452a7?utm_campaign=maleskine&utm_content=note&utm_source=desktop&utm_medium=notes-included-collection)

1.1-简介 WKInterfaceController是我们开发Watch App的核心类，它的地位和之前使用的UIViewController一样。 每一个Watch App构建时，至少需要在AppDelegate中实现WKInterfaceControllerDelegate协议。 坤小 (/u/b5c2ad9cbe7b?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

WatchOS系统开发大全（7）-WKInterfacelImage (/p/163822116dff?utm_campaign=maleskine&utm_content=note&utm_source=desktop&utm_medium=notes-included-collection)

WKInterfacelImage 1.1-简介 WKInterfacelImage与iOS的UIKit框架中UIImageView一样，是一个显示图片的控件。 1.2-API介绍 WKInterfacelImage的API只有四个方法。 1.3-使用示例 效果

坤小 (/u/b5c2ad9cbe7b)

utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

价格的真相：用户嫌贵，并不是真的贵了。 (/p/25a7e8d253f8?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

99%的marketing，都很可能遇到过这些问题：“客户说我们产品太贵了，怎么办？”“A公司又开始降价促销了，怎么办？”“B公司的低价我们做不到，怎么办？”

康熙师爷 (/u/d6884f19b3e1?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

谈月收入，就是在伤害90后的感情 (/p/1608c719aa84?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

离2018年只有一个月了，但90后们在经过秃头、脱发、油腻、颓、丧、拼等等各种关键词轮番洗礼后，还是没能躲过12月到来之前的最后一次被推上风口浪

谈心社 (/u/50e0da62c77d?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

我们准备了千元现金，请你做这件很久没做了的事..... (/p/8bd378de66a8?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

你有多久，没提笔写字了？握着虚拟键盘打字如飞的手，一提起笔来却像磁带卡了壳。九宫格都能熟捻于心，可明明很熟悉的字，却记不起它的横竖撇捺。互

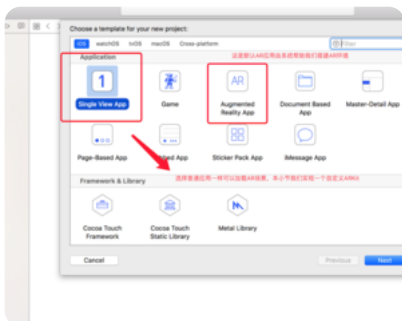
简书活动精选 (/u/cd73ae789321?utm_campaign=maleskine&utm_content=user&utm_medium=pc_all_hots&utm_source=recommendation)

ARKit从入门到重新入门(二)--每个类的含义及食用方法 (/p/ca6f5e1b633a?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

笔者在上篇文章中对ARKit进行了简单介绍，在本篇文章中主要介绍ARKit中经常用到的API及使用方法。了解这些API的含义以方便我们后边的深入学习。1.1－ARAnchor 用于在AR场景中放置物体的一个现实世界的位

 未明一二 (/u/3dde14779860?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/bcb579feeea3?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

ARKit从入门到精通三部曲之三 (/p/bcb579feeea3?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

此文章为转载文章:ARKit从入门到精通（3）-ARKit自定义实现 在上一小节中ARKit从入门到精通（2）-ARKit工作原理及流程介绍，我们完整的介绍了ARKit的工作原理，那么本小节我们就通过对ARKit工作原理

 泥孩儿0107 (/u/39afa65f213a?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/c753272f9917?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

ARkit知多少---从青铜到王者 (/p/c753272f9917?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

Introducing ARKit iOS 11 引入 ARKit，这是个全新的框架，允许开发者轻松地 iPhone 和 iPad 创建 与 伦比的增强现实体验。通过将虚拟对象和虚拟信息同 户周围的环境相互融合，ARKit 使得应用跳出屏幕的限

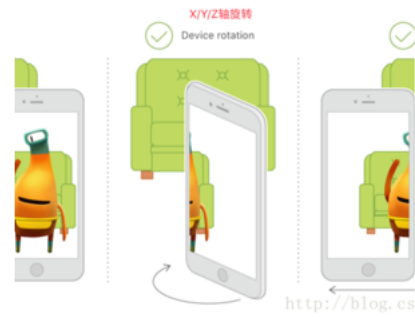
 没八阿哥的 (/u/ad200a77aa6?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

ARKit从入门到精通 (/p/373ce8f58ab3?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)

ARKit从入门到精通（1）-ARKit初体验标签： ARKit2017-06-12 17:33 1772人阅读 评论(0) 收藏 举报 分类： ARKit初探（3） 目录(?)[+]ARKit从入门到精通（1）-ARKit初体验转载请标注出处:http://blog.cs...

 零度_不结冰 (/u/52ae6e2e7db5?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/76c6d28f01dc?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
ARKit从入门到精通三部曲 (/p/76c6d28f01dc?utm_campaign=maleskin...

此文章为转载文章 ARKit从入门到精通（1）-ARKit初体验 1.1-AR技术简介 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，

 泥孩儿0107 (/u/39afa65f213a?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

沉潜技法 (/p/6400aaa70b3a?utm_campaign=maleskine&utm_content...

曾经的我也自认一名文艺青年，因为开工作室的关系，认识了很多艺术高中的朋友。某天我整了把毕加索，自觉大快淋漓（无非是一身颜料而已）一个很从事美术专业的兄弟来我家，我说你看看，这玩意怎么样？他

 马凯伦 (/u/88ec9ad02ca4?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

音乐和文字 (/p/daed25c9f850?utm_campaign=maleskine&utm_conte...

喜欢沉醉在音乐的文字中， 聆听音乐碰撞灵魂的共鸣与对白。音乐文字之韵， 美的给人以诗的意境、 画的景色、 情的馨香。 它能触动心底隐藏的情感， 是拨动灵魂那根最敏感琴弦。 潇洒的文字 随着流淌的音乐盘

 独望夕晖般的凄凉 (/u/91616e958544?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

祝 (/p/8159bbb80344?utm_campaign=maleskine&utm_content=note...

我见过流星我许了愿. 我拔过睫毛我许了愿. 我望过月亮我许了愿. 我折过纸船我许了愿. 我听过飞机我许了愿. 我吃过蛋糕我许了愿. 我看过孔明灯我许了愿. 我抓过氢气球我许了愿. 我对田边的青蛙说. 我对飞舞的萤火虫

 Daney呀 (/u/565ec0648e98?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

惠普与瑞士钟表制造商合作推出Bold Motion智能手表 (/p/50dcb424d51f?...

科技传媒网讯 当我们看到惠普设计的智能手表的时候，你会感叹其在智能手表的设计上可谓是真正的反潮流。惠普去年推出的MB Chronowing由Michael Bastian设计，它有一个非接触式的单色显示器，但是却塞满

 奥芯软件 (/u/8dc40b1df75d?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/23bd7e48b532?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
再往下挖一点 (/p/23bd7e48b532?utm_campaign=maleskine&utm_con...

“李哥，那消息你知道了么？”小张把老李拉到没人的地方低声问道。“什么消息？”老李心猛的一跳，按理来说小张不应该知道啊。“得了吧李哥，您就别瞒我，就是那些宝贝的下落啊。”老李暗道:完了，他知道了，想一

 梦回初 (/u/122459aeb577?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)