

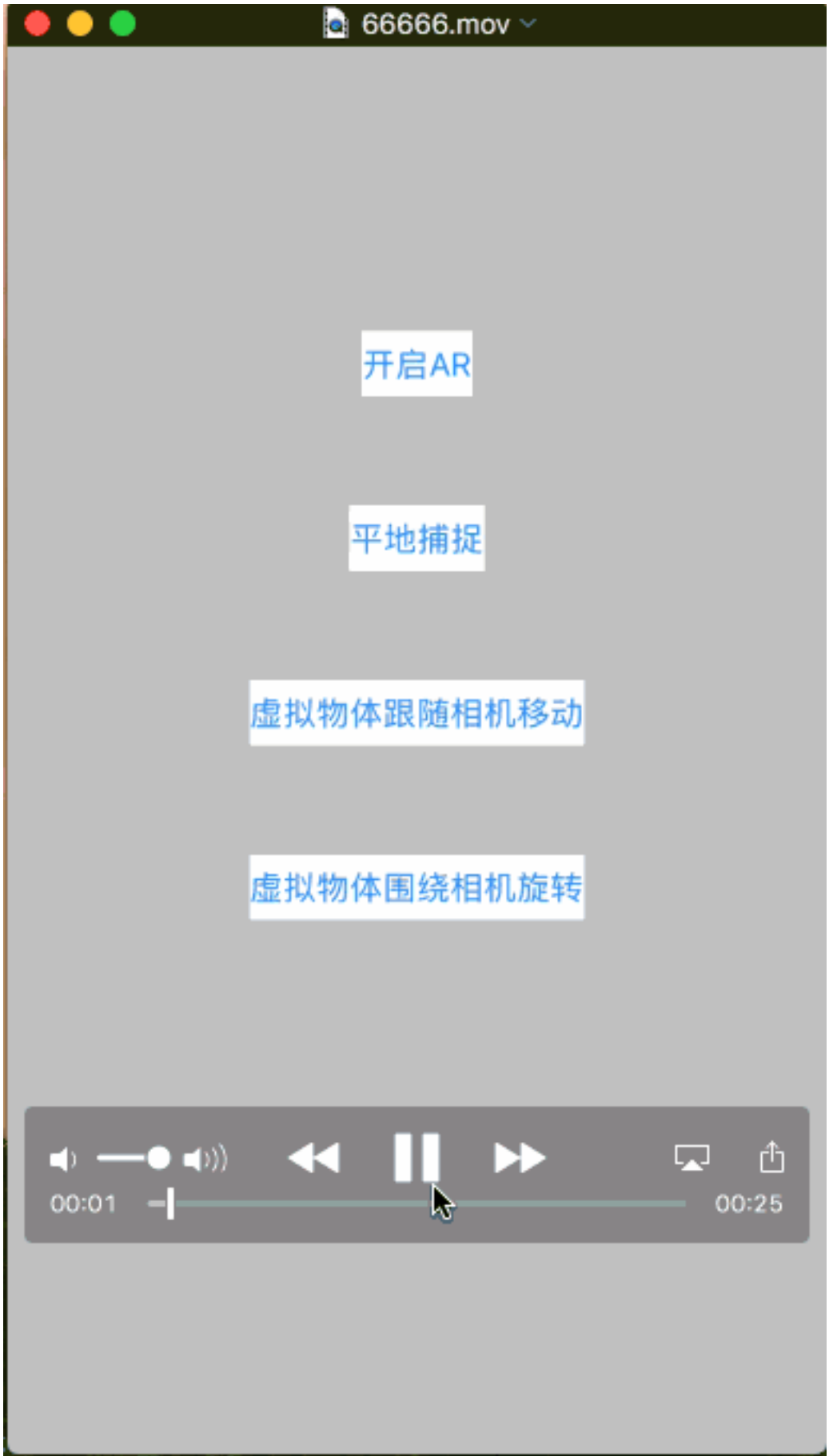
ARKit从入门到精通（8） -ARKit捕捉平地



坤小 (/u/b5c2ad9cbe7b) [+ 关注](#)

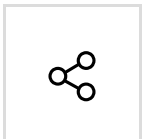
2017.06.13 09:13* 字数 594 阅读 2847 评论 20 喜欢 12 阅读 2847 评论 20 喜欢 12 (/u/b5c2ad9cbe7b)

- 转载请注明出处:<http://www.jianshu.com/p/fc40ed0080dd>
(<https://www.jianshu.com/p/fc40ed0080dd>)
- 1.1-ARKit捕捉平地实现流程介绍
- 1.2-完整代码
- 1.3-代码下载地址
- 在介绍完 ARKit 详细的工作原理以及所有的API之后，最令人期待的干货终于要来了！
- 废话不多说，先看效果
 - 桌子上的绿萝太孤独了，给它来一个郁金香陪伴一下吧~



0901.gif

- 在椅子上摆瓶花吧~





0902.gif

<h2 id="1.1">1.1-ARKit捕捉平地实现流程介绍</h2>

- 平地捕捉需要一点时间，*ARKit*内部会进行比较复杂的算法，所以有时候可能没有那么快，需要耐心等待。
- 1.搭建自定义ARKit工作环境，详情请见笔者 [ARKit从入门到精通（3）-ARKit自定义实现](#) 这篇文章
- 2.配置 `ARSessionConfiguration` 捕捉平地事件，实现 `ARSCNViewDelegate` 监听捕捉平地回调
- 3.通过 `ARSCNView` 的代理获取平地锚点 `ARPlaneAnchor` 的位置，添加一个用于展示渲染平地的3D模型（上图中一个红色的平地）
 - 在前面小节笔者已经强调过，*ARKit*框架只负责捕捉真实世界的图像，虚拟世界的场景由*SceneKit*框架来加载。所以*ARKit*捕捉到的是一个平地的空间，而这个空间本身是没有东西的（一片空白，只是空气而已），要想让别人能够更加真实的看到这一个平地的空间，需要我们使用一个3D虚拟物体来放入这个空间
- 4.开启延迟线程，在平地的位置添加一个花瓶节点
 - 此处一定要注意：花瓶节点是添加到代理捕捉到的节点中，而不是AR试图的根节点。因为捕捉到的平地锚点是一个本地坐标系，而不是世界坐标系
- 核心代码介绍

```
#pragma mark -搭建ARKit环境

//懒加载会话追踪配置
- (ARSessionConfiguration *)arSessionConfiguration
{
    if (_arSessionConfiguration != nil) {
        return _arSessionConfiguration;
    }
}
```

```
//1.创建世界追踪会话配置（使用ARWorldTrackingSessionConfiguration效果更加好），需要A9芯片支持
ARWorldTrackingSessionConfiguration *configuration = [[ARWorldTrackingSessionConfiguration alloc] init];
//2.设置追踪方向（追踪平面，后面会用到）
configuration.planeDetection = ARPlaneDetectionHorizontal;
_arSessionConfiguration = configuration;
//3.自适应灯光（相机从暗到强光快速过渡效果会平缓一些）
_arSessionConfiguration.lightEstimationEnabled = YES;

return _arSessionConfiguration;

}

#pragma mark -- ARSCNViewDelegate

//添加节点时候调用（当开启平地捕捉模式之后，如果捕捉到平地，ARKit会自动添加一个平地节点）
- (void)renderer:(id <SCNSceneRenderer>)renderer didAddNode:(SCNNode *)node forAnchor:(ARAnchor *)anchor
{

    if(self.arType != ARTypePlane)
    {
        return;
    }

    if ([anchor isKindOfClass:[ARPlaneAnchor class]]) {
        NSLog(@"捕捉到平地");

        //添加一个3D平面模型，ARKit只有捕捉能力，锚点只是一个空间位置，要想更加清楚看到这个空间，我们需要给空间添加一个平地的3D模型来渲染他

        //1.获取捕捉到的平地锚点
        ARPlaneAnchor *planeAnchor = (ARPlaneAnchor *)anchor;
        //2.创建一个3D物体模型（系统捕捉到的平地是一个不规则大小的长方形，这里笔者将其变成一个长方形，并且是否对平地做了一个缩放效果）
        //参数分别是长宽高和圆角
        SCNBox *plane = [SCNBox boxWithWidth:planeAnchor.extent.x*0.3 height:0 length:planeAnchor.extent.x*0.3 chamferRadius:0];
        //3.使用Material渲染3D模型（默认模型是白色的，这里笔者改成红色）
        plane.firstMaterial.diffuse.contents = [UIColor redColor];

        //4.创建一个基于3D物体模型的节点
        SCNNode *planeNode = [SCNNode nodeWithGeometry:plane];
        //5.设置节点的位置为捕捉到的平地的锚点的中心位置 SceneKit框架中节点的位置position是一个基于3D坐标系的矢量坐标SCNVector3Make
        planeNode.position =SCNVector3Make(planeAnchor.center.x, 0, planeAnchor.center.z);

        //self.planeNode = planeNode;
        [node addChildNode:planeNode];

        //2.当捕捉到平时时，2s之后开始在平地上添加一个3D模型

        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(2 * NSEC_PER_SEC)), dispatch_get_main_queue(), ^{
            //1.创建一个花瓶场景
            SCNScene *scene = [SCNScene sceneNamed:@"Models.scnassets/vase/vase.scn"];

            //2.获取花瓶节点（一个场景会有多个节点，此处我们只写，花瓶节点则默认是场景子节点的第一个）

            //所有的场景有且只有一个根节点，其他所有节点都是根节点的子节点
            SCNNode *vaseNode = scene.rootNode.childNodes[0];

            //4.设置花瓶节点的位置为捕捉到的平地的位置，如果不设置，则默认为原点位置，也就是相机位置
            vaseNode.position = SCNVector3Make(planeAnchor.center.x, 0, planeAnchor.center.z);

            //5.将花瓶节点添加到当前屏幕中
            //!!!!此处一定要注意：花瓶节点是添加到代理捕捉到的节点中，而不是AR试图的根节点。因为捕捉到的平地锚点是一个本地坐标系，而不是世界坐标系
            [node addChildNode:vaseNode];
        });
    }
}
```

<h2 id="1.2">1.2-完整代码</h2>

```
#import "ARSCNViewViewController.h"

//3D游戏框架
#import <SceneKit/SceneKit.h>
//ARKit框架
#import <ARKit/ARKit.h>

@interface ARSCNViewViewController ()<ARSCNViewDelegate,ARSessionDelegate>

//AR视图：展示3D界面
@property(nonatomic,strong)ARSCNView *arSCNView;

//AR会话，负责管理相机追踪配置及3D相机坐标
@property(nonatomic,strong)ARSession *arSession;

//会话追踪配置：负责追踪相机的运动
@property(nonatomic,strong)ARSessionConfiguration *arSessionConfiguration;

//飞机3D模型(本小节加载多个模型)
@property(nonatomic,strong)SCNNode *planeNode;

@end

@implementation ARSCNViewViewController

- (void)viewDidLoad {
    [super viewDidLoad];

    // Do any additional setup after loading the view.
}

- (void)back:(UIButton *)btn
{
    [self dismissViewControllerAnimated:YES completion:nil];
}

- (void)viewDidAppear:(BOOL)animated
{
    [super viewDidAppear:animated];

    //1.将AR视图添加到当前视图
    [self.view addSubview:self.arSCNView];
    //2.开启AR会话（此时相机开始工作）
    [self.arSession runWithConfiguration:self.arSessionConfiguration];

    //添加返回按钮
    UIButton *btn = [UIButton buttonWithType:UIButtonTypeCustom];
    [btn setTitle:@"返回" forState:UIControlStateNormal];
    btn.frame = CGRectMake(self.view.bounds.size.width/2-50, self.view.bounds.size.height-100, 100, 50);
    btn.backgroundColor = [UIColor greenColor];
    [btn addTarget:self action:@selector(back:) forControlEvents:UIControlEventTouchUpInside];
    [self.view addSubview:btn];
}

#pragma mark -搭建ARKit环境

//懒加载会话追踪配置
- (ARSessionConfiguration *)arSessionConfiguration
{
    if (_arSessionConfiguration != nil) {
        return _arSessionConfiguration;
    }

    //1.创建世界追踪会话配置（使用ARWorldTrackingSessionConfiguration效果更加好），需要A9芯片支持
    ARWorldTrackingSessionConfiguration *configuration = [[ARWorldTrackingSessionConfiguration alloc] init];
    //2.设置追踪方向（追踪平面，后面会用到）
    configuration.planeDetection = ARPlaneDetectionHorizontal;
    _arSessionConfiguration = configuration;
    //3.自适应灯光（相机从暗到强光快速过渡效果会平缓一些）
    _arSessionConfiguration.lightEstimationEnabled = YES;

    return _arSessionConfiguration;
}
}
```

```
//懒加载拍摄会话
- (ARSession *)arSession
{
    if(_arSession != nil)
    {
        return _arSession;
    }
    //1.创建会话
    _arSession = [[ARSession alloc] init];
    _arSession.delegate = self;
    //2返回会话
    return _arSession;
}

//创建AR视图
- (ARSCNView *)arSCNView
{
    if (_arSCNView != nil) {
        return _arSCNView;
    }
    //1.创建AR视图
    _arSCNView = [[ARSCNView alloc] initWithFrame:self.view.bounds];

    //2.设置代理 捕捉到平地会在代理回调中返回
    _arSCNView.delegate = self;

    //2.设置视图会话
    _arSCNView.session = self.arSession;
    //3.自动刷新灯光（3D游戏用到，此处可忽略）
    _arSCNView.automaticallyUpdatesLighting = YES;

    return _arSCNView;
}

#pragma mark -- ARSCNViewDelegate

//添加节点时候调用（当开启平地捕捉模式之后，如果捕捉到平地，ARKit会自动添加一个平地节点）
- (void)renderer:(id <SCNSceneRenderer>)renderer didAddNode:(SCNNode *)node forAnchor:(ARAnchor *)anchor
{
    if(self.arType != ARTypePlane)
    {
        return;
    }

    if ([anchor isKindOfClass:[ARPlaneAnchor class]]) {
        NSLog(@"捕捉到平地");

        //添加一个3D平面模型，ARKit只有捕捉能力，锚点只是一个空间位置，要想更加清楚看到这个空间，我们需要给空间添加一个平地的3D模型来渲染他

        //1.获取捕捉到的平地锚点
        ARPlaneAnchor *planeAnchor = (ARPlaneAnchor *)anchor;
        //2.创建一个3D物体模型 （系统捕捉到的平地是一个不规则大小的长方形，这里笔者将其变成一个长方形，并且是否对平地做了一个缩放效果）
        //参数分别是长宽高和圆角
        SCNBox *plane = [SCNBox boxWithWidth:planeAnchor.extent.x*0.3 height:0 length:planeAnchor.extent.x*0.3 chamferRadius:0];
        //3.使用Material渲染3D模型（默认模型是白色的，这里笔者改成红色）
        plane.firstMaterial.diffuse.contents = [UIColor redColor];

        //4.创建一个基于3D物体模型的节点
        SCNNode *planeNode = [SCNNode nodeWithGeometry:plane];
        //5.设置节点的位置为捕捉到的平地的锚点的中心位置 SceneKit框架中节点的位置position是一个基于3D坐标系的矢量坐标SCNVector3Make
        planeNode.position =SCNVector3Make(planeAnchor.center.x, 0, planeAnchor.center.z);

        //self.planeNode = planeNode;
        [node addChildNode:planeNode];

        //2.当捕捉到平地时，2s之后开始在平地上添加一个3D模型

        dispatch_after(dispatch_time(DISPATCH_TIME_NOW, (int64_t)(2 * NSEC_PER_SEC)), dispatch_get_main_queue()), ^{
            //1.创建一个花瓶场景
            SCNScene *scene = [SCNScene sceneNamed:@"Models.scnassets/vase/vase.scn"];
            //2.获取花瓶节点（一个场景会有多个节点，此处我们只写，花瓶节点则默认是场景子节点的第一个）
            //所有的场景有且只有一个根节点，其他所有节点都是根节点的子节点
```

```
SCNNode *vaseNode = scene.rootNode.childNodes[0];

//4.设置花瓶节点的位置为捕捉到的平地的位置，如果不设置，则默认为原点位置，也就是相机位置
vaseNode.position = SCNVector3Make(planeAnchor.center.x, 0, planeAnchor.center.z);

//5.将花瓶节点添加到当前屏幕中
//!!!此处一定要注意：花瓶节点是添加到代理捕捉到的节点中，而不是AR试图的根节点。因为捕捉到的平地锚点是一个本地坐标系，而不是世界坐标系
[node addChildNode:vaseNode];
});
}
}

//刷新时调用
- (void)renderer:(id <SCNSceneRenderer>)renderer willUpdateNode:(SCNNode *)node forAnchor:(ARAnchor *)anchor
{
    NSLog(@"刷新中");
}

//更新节点时调用
- (void)renderer:(id <SCNSceneRenderer>)renderer didUpdateNode:(SCNNode *)node forAnchor:(ARAnchor *)anchor
{
    NSLog(@"节点更新");
}

//移除节点时调用
- (void)renderer:(id <SCNSceneRenderer>)renderer didRemoveNode:(SCNNode *)node forAnchor:(ARAnchor *)anchor
{
    NSLog(@"节点移除");
}

#pragma mark - ARSessionDelegate

//会话位置更新（监听相机的移动），此代理方法会调用非常频繁，只要相机移动就会调用，如果相机移动过快，会有一定的误差，具体的需要强大的算法去优化，笔者这里就不深入了
- (void)session:(ARSession *)session didUpdateFrame:(ARFrame *)frame
{
    NSLog(@"相机移动");
}

- (void)session:(ARSession *)session didAddAnchors:(NSArray<ARAnchor*>*)anchors
{
    NSLog(@"添加锚点");
}

- (void)session:(ARSession *)session didUpdateAnchors:(NSArray<ARAnchor*>*)anchors
{
    NSLog(@"刷新锚点");
}

- (void)session:(ARSession *)session didRemoveAnchors:(NSArray<ARAnchor*>*)anchors
{
    NSLog(@"移除锚点");
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}


/*
#pragma mark - Navigation

// In a storyboard-based application, you will often want to do a little preparation before navigation
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender {
    // Get the new view controller using [segue destinationViewController].
    // Pass the selected object to the new view controller.
}
*/

@end
```

1.3-代码下载地址

- ***ARKit从入门到精通
Demo:<http://download.csdn.net/detail/u013263917/9868679>
([https://link.jianshu.com?](https://link.jianshu.com?t=http://download.csdn.net/detail/u013263917/9868679)
[t=http://download.csdn.net/detail/u013263917/9868679](http://download.csdn.net/detail/u013263917/9868679))
- 笔者已经将8、9、10三小节的代码合并成一个完整的小demo，供读者交流学习



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)

12



更多分享

(<http://cwb.assets.jianshu.io/notes/images/13410719>



下载简书 App ▶

随时随地发现和创作内容



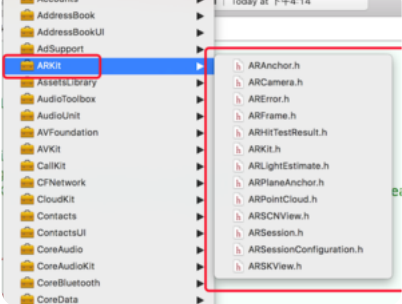
(/apps/download?utm_source=nbc)

被以下专题收入，发现更多相似内容

-  ARKit (/c/770024feab7d?utm_source=desktop&utm_medium=notes-included-collection)
-  AR增强现实 (/c/9d16871ab7ef?utm_source=desktop&utm_medium=notes-included-collection)
-  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)
-  iOSARKit专题 (/c/bf3e8e31d698?utm_source=desktop&utm_medium=notes-included-collection)
-  ARKIT (/c/3682baaef8ae?utm_source=desktop&utm_medium=notes-included-collection)


展开更多 ▾

(/p/3c7ee58d40f4?



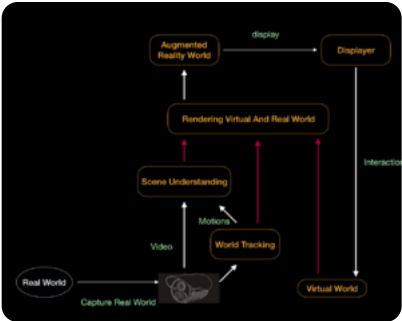
utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
ARKit从入门到精通（4）-ARKit全框架API大全 (/p/3c7ee58d40f4?utm_...

转载请注明出处：http://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-ARL...

 坤小 (/u/b5c2ad9cbe7b?


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

(/p/04a2d44e3ee8?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
ARKit 技术概述 (/p/04a2d44e3ee8?utm_campaign=maleskine&utm_c...


苹果在 WWDC2017 中推出了 ARKit，通过这个新框架可以看出苹果未来会在 AR 方向不断发展，本着学习兴趣，对此项新技术进行了学习，并在团队进行了一次分享，利用业余时间把几周前分享的内容整理成文档

 壹米玖坤VR讲师 (/u/63cebe43d0cc?

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

ARKit让飞机跟着镜头飞起来(转) (/p/48f009b252b1?utm_campaign=ma...


1.1-ARKit物体跟随相机移动流程介绍 1.2-完整代码1.3-代码下载地址 废话不多说，先看效果 1001.gif 1.1-ARKit物体跟随相机移动流程介绍 1.点击屏幕添加物体，已经在第三小节ARKit从入门到精通（3）-ARKit自

 树上的cat_ee3c (/u/43f7b4175b56?

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

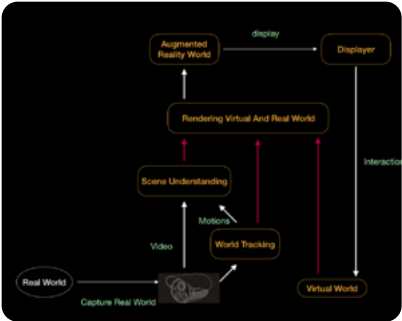
ARKit从入门到重新入门(二)--每个类的含义及食用方法 (/p/ca6f5e1b633a...

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/7faa4a3af589?



utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation)
直击苹果 ARKit 技术 (/p/7faa4a3af589?utm_campaign=maleskine&ut...


苹果在 WWDC2017 中推出了 ARKit，通过这个新框架可以看出苹果未来会在 AR 方向不断发展，本着学习兴趣，对此项新技术进行了学习，并在团队进行了一次分享，利用业余时间把几周前分享的内容整理成文档

 程序员钙片吃多了 (/u/1235761e64a4?

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

Day3_保持良好的心态 (/p/a7984330340a?utm_campaign=maleskine&...

这学期开学后经常因为大宝不能按时完成作业变得暴躁易怒，经常爱批评他，很小的事情都会让我内心很受挫。造成这样的原因大致有两点，一是日常生活没有奶奶的参与，所有的家务都全在我身上，无形中少了很

 秋小聿 (/u/b54a4b2e4fee?

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

风和游子 (/p/4e2a322ff506?utm_campaign=maleskine&utm_content=...

风来自北方，也来自南方 就像一个游子的母亲 走过了河西走廊，也去过故宫天坛 游子是一粒种子 带着寄托和希望 种在黄土地，长在汗水里 风啊，你是我生长的乳汁 给了我关怀 可到头来尽是泪眼模糊的分聚离愁 那

 千云雨 (/u/a258be502e07?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

六岁男孩妈妈白雪微谈家庭教育---小顶妈首访记 (/p/d41fd8788f1a?utm_c...

小顶妈小序： 白雪今年三十出头，北京出生，北京长大。在她十几岁、大约初中的时候我认识了她。说认识，其实也没有说过几句话，主要是认识她美丽、睿智给予我们很多提携帮助的妈妈。记忆中，她寡言、多

 魏凯 (/u/63009ad16f6a?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/0c715f45fcab?



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

非诚勿扰（特别版） (/p/0c715f45fcab?utm_campaign=maleskine&utm...

娇艳的外表，丰满的果肉， 一咬满口琼浆， 再高冷的嘴，也要多吃几口； 再挑剔的舌头，也会被征服。甜得过初恋，甜得你恩宠。 它就是我们本期特别嘉宾 金都一号 红心火龙果 爱之初体验 它源自台湾，产自海

 卖果果的小胖墩儿 (/u/af6d2c571b88?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)

(/p/9aaf22394163?



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

瑶瘦瘦的一次黔东南“变形计”之旅|国庆中秋小长不看人头之旅（内含攻略） ...

如果有一天我的理想被风雨淋湿,你是否愿意回头扶我一把？ 如果有一天我无力前行，你是否愿意陪我一个温暖的午后？ 如果我问你什么，你是否想到妈妈梦中的惊起？ 如果那是一个你不熟悉的家，你会不会把善良当

 巨鹿猫 (/u/326fe636f276?utm_campaign=maleskine&utm_content=user&utm_medium=seo_notes&utm_source=recommendation)