场景SCNScene

sceneView.scene = scene

SCNBox()//最后一个参数：圆角半径

SCNMeterial//渲染器 .diffuse.contents=UIColor.red box.meterials=[meterial]

SCNNode(SCNBox)立方体形状的节点

boxNode.position = SCNVector3(0,0,-0.2)

Scene.rootNode.addChildNode(boxNode)

从一般UIView创建：#import <SceneKit/SceneKit.h> #import<ARKit/ARKit.h>

<ARSCNViewDelegate>

ARSCNView ARSession ARConfiguration

ARWorldTrackingConfiguration session.runwithConfigration

用到图片资源的时候最好从文件夹拖，保证addtotarget

//行星demo (第2次课)关键点:将月球饶地球的轨道节点添加到地月组合节点

节点几何渲染上图\_sunNode.geometry.firstMaterial.multiply.contents = “…jpg”//镶嵌，把整张图片拉伸，之后会变淡

.firstMaterial.diffuse.contents=”…jpg”//平均扩散到物件表面，并且光滑透亮，注:这里直接是图片路径，而不是UIImage

.multiply.intensity=0.5在渲染上的强度

.firtstMaterial.lightingModelName=SCNLigtingModelConstant光源

.multiply.wrapS= ... .diffuse.wrapS=.mutiply.wrapS=.diffuse.wrapT=SCNWrapModeRepeat //设置纹理过滤模式

[CABasicAnimation animationWithKeyPath:@"contentsTransform"]//纹理变形动画

animation.fromValue=[NSValue valueWithCATransform3D: CATransform3DConcat(CATransform3DMakeTranslation(0,0,0),CATransform3DMakeScale(3,3,3))]//拉伸纹理，使其看起来旋转

夜光图(不受光的一面).firstMaterial.emission.content=”.jpg”

镜面图.firstMaterial.specular.contents=”.jpg”

.firstMaterial.shininess=0.1//光泽 .firstMaterial.specular.intensity=0.5//反射多少光出去 .firstMaterial.specular.contens//反光的样子

[node runAction:[SCNAction repeatActionForever:[SCNAction rotateByX:0 y:2 z:0]]]//地球自转

光node.light=[SCNLigtht light] ligtht.color light.type=SCNLightOmni

light.attenuationStartDistance=1.0 light.attenuationEndDistance=20.0近端和远端位置

//光晕.firstMaterial.lightingModeName=SCNLightingModelConstant不要光, .firstMaterial.writesToDepthBuffer=NO不要深度

session.run(configuration, options:[.resetTracking, .removeExistingAnchors])//全局追踪 重新设置锚点 以免找不到位置

SCNView results=hitTest(postion, type:[.featurePoint])//搜寻现实世界中的锚点，用在相机与物件之间的距离。搜索ARSession检测到的锚点还有真实世界的对象，而不是view里面的内容。假设说你今天要找的是SceneKit里面的内容，那么请使用(\_ :, options: )

Results.first.worldTransform世界坐标系的位置和方向

let source = SCNGeometrySource(vertices: [self,vector])//顶点

画线: let element=SCNGeometryElement(indices:[Int],primitiveType: .line)

SCNGeometry(sources: [source], elements:[element])

: Equatable 协议 实现static func == (lhs: SCNVector3, rhs:SCNVector3)->Bool

firstMaterial.lightingModel = .constant//不产生阴影，环境光

firstMaterial.isDoubleSided = true//两面都产生光亮

约束：SCNLookAtConstraint(target:sceneView.pointOfView)//注视sceneView中心点，让其始终面向使用者

constraint.isGimbalLockEnabled=true//环绕默认轴，默认是false

face1调到0，其它几个face调到0.01, 不要一起调//坑

3D模型网站：turbosquid.com free3d.com 下载.obj用blender(blender.org)转为.dae

spriteKit: SKView().ignoresSiblingOrder = true//代码设置和ska文件设置两者可以并存

spriteKit: SKNode().anchorPosition: 设置锚点位置，默认是中心

记得注释掉view.ignoresSiblingOrder或设置成false，否则没有层次 //坑

//page 10.1