SceneKit

Native 3D for iOS

Jackrabbit

Introduction

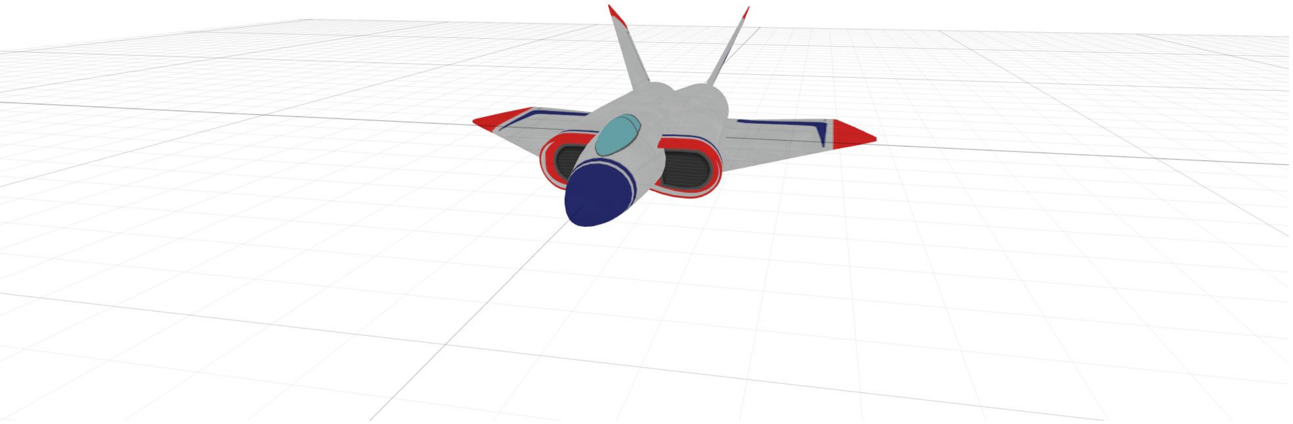
- Who am I?
  - Vincil Bishop, iOS Developer with Jackrabbit Mobile
  - vincil@jackrabbitmobile.com

- What will this presentation cover?
  - SceneKit General Concepts
  - SceneKit Major Types
  - SceneKit Examples
  - Where to learn more...

What is SceneKit

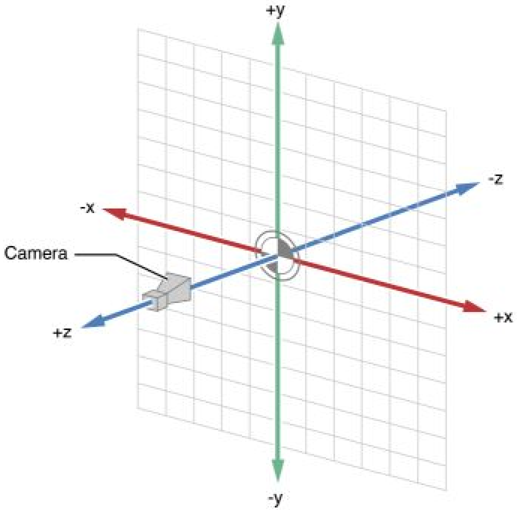
- Apple’s description, see: https://developer.apple.com/scenekit/
- SceneKit is a high-level 3D graphics framework that helps you create 3D animated scenes and effects in your apps.
- It incorporates a physics engine, a particle generator, and easy ways to script the actions of 3D objects so you can describe your scene in terms of its content — geometry, materials, lights, and cameras — then animate it by describing changes to those objects.
- A SceneKit scene is embedded in your app just like a UIView, expect it uses a SCNView.

SceneKit Default Example



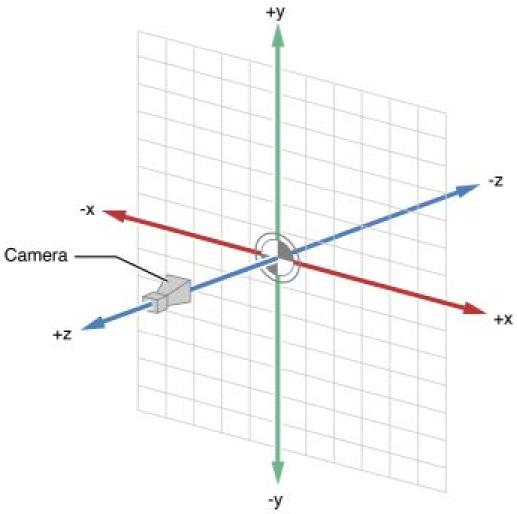
SceneKit General Concepts: The Plane

- SceneKit “nodes” (type: SCNNode) are the major objects that interact with a scene.
- These nodes are organized in a hierarchy and move along a 3 dimensional plane.

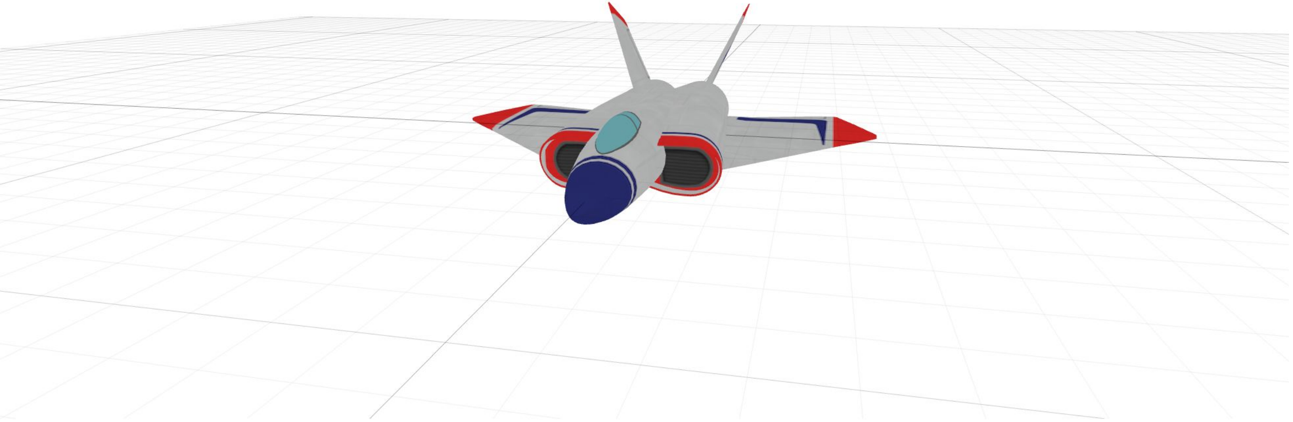


SceneKit General Concepts: Lights, Camera

- A camera node (type: SCNCamera) determines the user's POV.
- One or more light nodes (type: SCNLight) illuminate the scene, and determine brightness, shadows, etc.



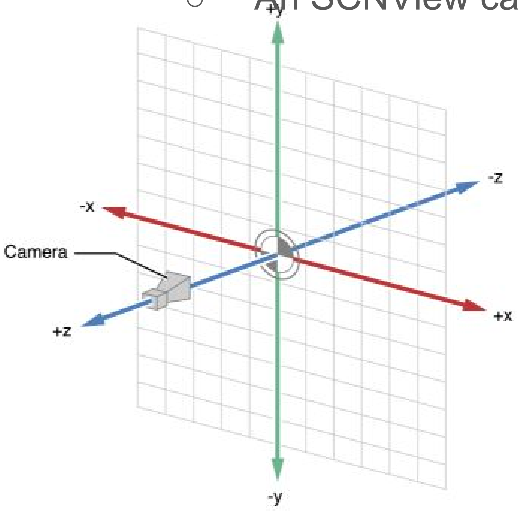
SceneKit Default Example
A Closer Look



SceneKit vs. UIKit

- X,Y, Z 3D Plane coordinates vs. CGRect / CGPoint 2D coordinates
  - `node.position = SCNVector3Make(0, 10, 10);`

- A SceneKit scene is displayed in a SCNView instead of a UIView.
  - An SCNView can be used anywhere a UIView would be used.



Advanced Note:
There are methods to translate between the UIKit and SceneKit coordinate planes...

SceneKit Major Types

SCNView

- A view for displaying 3D SceneKit content.
- Can be used anywhere a UIView would be used.
- Behaves much like a “window” into the SceneKit 3D world.
- Has a reference to a single SCNScene.
- Reference: https://developer.apple.com/reference/scenekit/scnview

SCNScene

- A scene graph—a hierarchy of nodes with attached geometries, lights, cameras and other attributes that together form a displayable 3D scene.
- Contains a “root node” under which all other nodes in the scene are added, much like views in a UIView’s view hierarchy.
- Reference: https://developer.apple.com/reference/scenekit/scnscene

SCNNode

- A structural element of a scene graph, representing a position and transform in a 3D coordinate space, to which you can attach geometry, lights, cameras, or other displayable content.
- Can be moved along the 3D plane by adjusting the .position property, using an SCNVector3 struct.
- Reference: https://developer.apple.com/reference/scenekit/scnnode

SCNGeometry

- A three-dimensional shape (also called a model or mesh) that can be displayed in a scene, with attached materials that define its appearance.
- This is what gives an SCNNode its shape.
- Built in primitive shapes (SCNGeometry subclasses)
    - SCNPlane, SCNBox, SCNSphere, SCNPyramid, SCNCode, etc.
- Reference: https://developer.apple.com/reference/scenekit/scngeometry

SCNMaterial

- A set of shading attributes that define the appearance of a geometry's surface when rendered.
- UIImage, UIColor, SpriteKit scene, etc. can be applied to a SCNMaterial.
- The SCNMaterial is then applied to an SCNGeometery for display on a node.
- Reference: https://developer.apple.com/reference/scenekit/scnmaterial

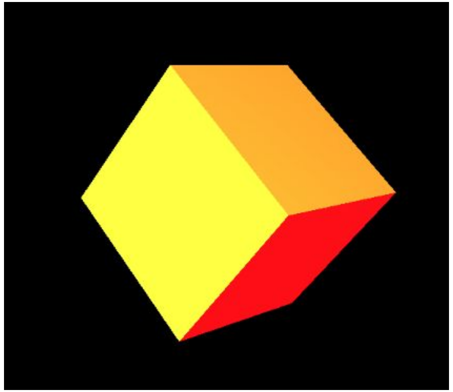
SCNVector3 Struct

- A representation of a three-component vector.
- Comparable to a CGPoint in UIKit.
- Created using `SCNVector3Make(x, y, z);`
- There also exists a SCNVector4, and various utility methods for each.
- Both structures are used for multiple purposes, not just representing a position on the 3D plane.
- Reference: https://developer.apple.com/reference/scenekit/scnvector3

SCNAction

- A simple, reusable animation that changes attributes of any node you attach it to.
- SCNActions are high level, there are other ways to animate SceneKit content.
- There are many default actions: rotate, move, scale, fade, hide, etc.
- Can be grouped, chained, and even custom actions can be defined.
- Reference: https://developer.apple.com/reference/scenekit/scnaction

A Simple Cube Example...



In Conclusion

- Demo Project: https://github.com/BigWorkIndustries/SceneKitDemo
- Apple Reference: https://developer.apple.com/reference/scenekit
- Ray Wenderlich Tutorial: https://www.raywenderlich.com/83748/beginning-scene-kit-tutorial
- My Information: vincil@jackrabbitmobile.com

Any Questions?

Jackrabbit