

Announcement

- **Discuss final project ideas and project deadlines next Wednesday**

Today's Topics

- **SpriteKit**
- **SceneKit**
- **ARKit (iOS 11)**
- **Portions of today's slides and code examples come from the following**
 - <https://developer.apple.com/documentation/scenekit>
 - <https://developer.apple.com/documentation/arkit>
 - <https://github.com/bjarnel/arkit-smb-homage>
 - <https://github.com/farice/ARShooter>
 - Nathan Gitter's Animation Lecture
 - <https://research.engineering.wustl.edu/~todd/animationsAreFun.pdf>
 - Xue (Caroline) Zhang

SpriteKit

- **2D game engine with a simple API and built-in features like sprite rendering, animations, collision detection, and particle effects**
- **A SpriteKit scene (SKScene) can be embedded alongside UIKit views or exist in its own app separate from UIKit**

SKNodes and SKActions

- **SKNode is a game object (similar to a UIView in UIKit)**
- **SKSpriteNode subclass SKNode and renders graphics (like a UIImageView)**
- **SKActions define an animation and can be applied to any SKNode**
- **SKActions can be chained together**

Simple SpriteKit Animations

```
let sprite = SKSpriteNode(imageNamed: "Spaceship")  
let rotate = SKAction.rotate(byAngle: CGFloat.pi, duration: 1)  
let rotateForever = SKAction.repeatForever(rotate)  
sprite.run(rotateForever)
```



SpriteKit Demo

SpriteKit Additional Resources

- **Apple's Website**
 - <https://developer.apple.com/documentation/spritekit>
- **Example App**
 - <https://www.raywenderlich.com/category/apple-game-frameworks>
- **Nathan Gitter's SpriteKit Guide**
 - <http://nathangitter.com/SKGuide.pdf>

SceneKit

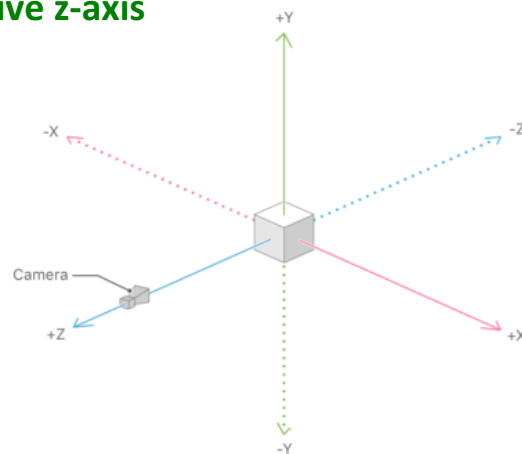
- **Create 3D games and add 3D content to apps using high-level scene descriptions.**
- **Easily add animations, physics simulation, particle effects, and realistic physically based rendering.**
- **A configuration that uses the rear-facing camera, tracks a device's orientation and position, and detects real-world flat surfaces.**

Organizing a Scene with Nodes

- **SceneKit implements content as a hierarchical tree structure of nodes (aka scene graph)**
 - Consists of a root node which defines a coordinate space for the world
- **SceneKit performs the following operations**
 - Displays scene in a view
 - Processes the scene graph
 - Performs animations
 - Renders each frame on the GPU

Coordinate System

- **SceneKit uses a right-handed coordinate system where the direction of the view is along the negative z-axis**



Common Classes in SceneKit

- **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.
- **SCNParticleSystem**
 - Manages the animation and rendering of a system of small image sprites, or particles, using a high-level simulation whose general behavior you specify.
- **SCNPhysicsContact**
 - Detailed information about a contact between two physics bodies in a scene's physics simulation.

Additional SceneKit Classes

- **Define shapes and textures**
 - SCNGeometry
 - SCNCylinder
 - SCNMaterial
- **Physics**
 - SCNPhysicsBody
- **Animations**
 - SCNAction
- **Vectors and Matrices**
 - SCNVector3
 - SCNMatrix4

SceneKit Demo

SceneKit Additional Resources

- **Apple's Developer Website**
 - <https://developer.apple.com/documentation/scenekit>
 - <https://developer.apple.com/library/content/samplecode/SceneKitVehicle/Introduction/Intro.html>
 - <https://developer.apple.com/library/content/samplecode/Fox/Introduction/Intro.html>
 - <https://developer.apple.com/videos/play/wwdc2017/605/>
- **Ray Wenderlich Tutorials**
 - <https://www.raywenderlich.com/146175/scene-kit-tutorial-swift-part-1-getting-started>
 - <https://www.raywenderlich.com/138712/make-game-like-can-knockdown>

SceneKit Examples

Augmented Reality (AR)

- **User experiences that add 2D or 3D elements to the live view from a device's camera**
- **Those elements appear to inhabit the real world**
- **ARKit**
 - Integrate iOS camera and motion features to produce augmented reality experiences in your app or game

Examples

Parts of ARKit

- Tracking
- Scene Understanding
- Rendering

Tracking

- **World Tracking**
 - Obtain relative position in physical environment
- **Visual inertial odometry**
 - Camera images and motion data to obtain view
- **No external setup**
 - No additional sensors needed

Scene Understanding

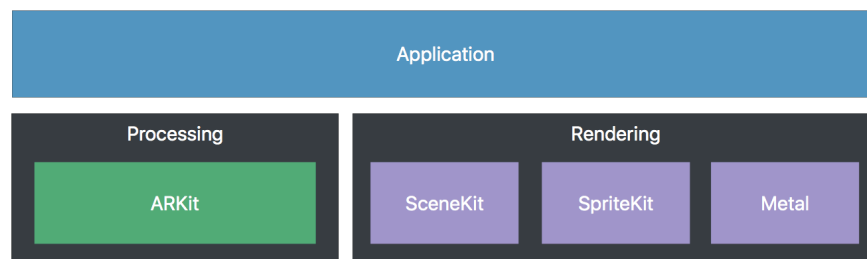
- **Plane Detection**
 - Discover surfaces in physical environment
- **Hit-testing**
 - Place virtual objects in physical world
- **Light estimation**
 - Correctly light virtual content in physical world

Rendering

- **Easy integration into any render**
- **AR views**
 - Available in SceneKit and SpriteKit
- **Custom rendering**

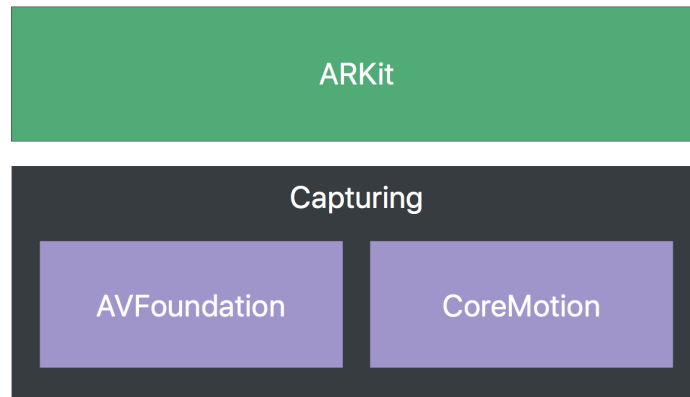
Using ARKit in Application

- **ARKit handles all processing in your AR app**
 - Provides information necessary to render a scene



Using ARKit

- **Capturing of data involves camera and CoreMotion sensors**



Components in ARKit

- **ARSession**
 - A shared object that manages the device camera and motion processing needed for augmented reality experiences.
- **ARWorldTrackingConfiguration**
 - A configuration that uses the rear-facing camera, tracks a device's orientation and position, and detects real-world flat surfaces.

Additional Components in ARKit

- **ARSceneView**
 - A view for displaying AR experiences that augment the camera view with 3D SceneKit content.
- **ARSCNViewDelegate**
 - Methods you can implement to mediate the automatic synchronization of SceneKit content with an AR session.

ARKit Demo