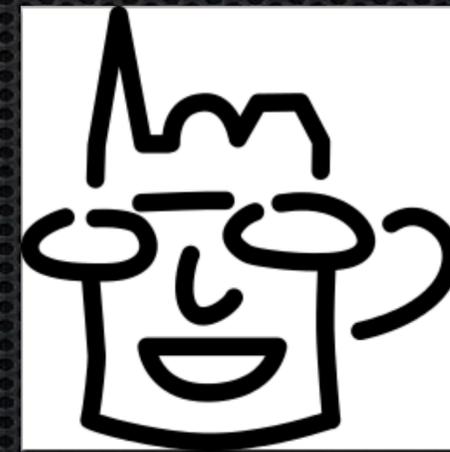


ARKit: @ CocoaHeads



Talk at CocoaHeads Aachen
Werner Lonsing

2017-07-27



ARKit: made by Apple

Introducing ARKit made by Apple,
beta and incomplete, but already a
milestone in terms of
the integration of hardware and software



AR on PowerBook OS X with OpenGL in 2004

What took you so long?



AR on iPhone 3GS with OpenGL ES in 2009

ARKit: Overview

Visual Inertial Odometry

ARKit uses Visual Inertial Odometry (VIO) to accurately track the world around it. VIO fuses camera sensor data with CoreMotion data. These two inputs allow the device to sense how it moves within a room with a high degree of accuracy, and without any additional calibration.

Scene Understanding and Lighting Estimation

With ARKit, iPhone and iPad can analyze the scene presented by the camera view, detect horizontal planes like tables and floors in the room, and can track and place objects on smaller feature points as well. ARKit also makes use of the camera sensor to estimate the total amount of light available in a scene and applies the correct amount of lighting to virtual objects.

High Performance Hardware and Rendering Optimizations

ARKit runs on the Apple A9 and A10 processors.

<https://developer.apple.com/arkit/>

ARKit: Visual Inertial Odometry

A term to remember: VIO

Visual inertial odometry describes the complete machinery deployed by Apple to provide their technology related to Augmented Reality.

VIO is working almost completely in the background, and there is (so far) no need to interfere with it.

Example usage in the iOS 11 beta 3 Release Notes, ARKit, Known Issues

- Continuing from a breakpoint while debugging an ARSession may result in VIO breaking. Any visual objects placed in the world/anchor are not visible. (31561202)

ARKit: Definitions (2004)

Augmented Reality (AR) — reality combined with some virtual elements e used for group working, public participation in planning, project consultations. Lonsing (2004) proposes the definition: ‘(.) **an augmented reality system generates a composite view in real time. The composition is a combination of a real scene viewed by a user and a virtual scene generated by a computer, where the real scene is submerged with additional information in order to enhance the perception of the user.**’

Hanzl, 2007

ARKit: Definitions

Augmented Reality is a technology which superimposes digital objects over one's perception of the real world. Thus, captivating user experiences can be realized, where digital artifacts can interact with real-world objects.

Jan Kaßel, Aachen (HCI) 2016

<https://hci.rwth-aachen.de/materials/publications/kassel2016a.pdf>

ARKit: Definitions

The basic requirement for any AR experience—and the defining feature of ARKit—is the ability to create and track a correspondence between the real-world space the user inhabits and a virtual space where you can model visual content. When your app displays that content together with a live camera image, the user experiences augmented reality: the illusion that your virtual content is part of the real world.

Apple, 2017

https://developer.apple.com/documentation/arkit/understanding_augmented_reality

ARKit: Augmented Reality

The promise of AR:



The reality of AR



Devices and their handling

ARKit: Task

Content

When and where to display what?

Usability

To whom and how?

ARKit: Classes

ARSession

- A shared object that manages the device camera and motion processing needed for augmented reality experiences.

Standard Views

ARSCNView

- A view for displaying AR experiences that augment the camera view with 3D SceneKit content.

ARSKView

- A view for displaying AR experiences that augment the camera view with 2D SpriteKit content.

(Custom Views; Metal)

- Displaying with Metal, Build a custom AR view by rendering camera images and using position-tracking information to display overlay content.

Configurations

ARSessionConfiguration

- A basic configuration that tracks a device's orientation only.

ARWorldTrackingSessionCon...

- A configuration that tracks a device's orientation and position, and that detects real-world surfaces seen by the device camera.

Camera and Scene Details

ARCamera

- Information about the camera position and imaging characteristics for a captured video frame in an AR session.

ARFrame

- A video image and position tracking information captured as part of an AR session.

ARLightEstimate

- Estimated scene lighting information associated with a captured video frame in an AR session.

Real-World Objects and Positions

ARAnchor

- A real-world position and orientation that can be used for placing objects in an AR scene.

ARPlaneAnchor

- Information about the position and orientation of a real-world flat surface detected in an AR session.

ARHitTestResult

- Information about a real-world surface found by examining a point in the device camera view of an AR session.

ARKit: Protocols

Responding to AR
Updates

ARSessionDelegate
ARSessionObserver

from
ARSessionObserver

ARSCNViewDelegate

ARSKViewDelegate

- Methods you can implement to receive captured video frame images and tracking state from an AR session.
- Methods you can implement to respond to changes in the state of an AR session.

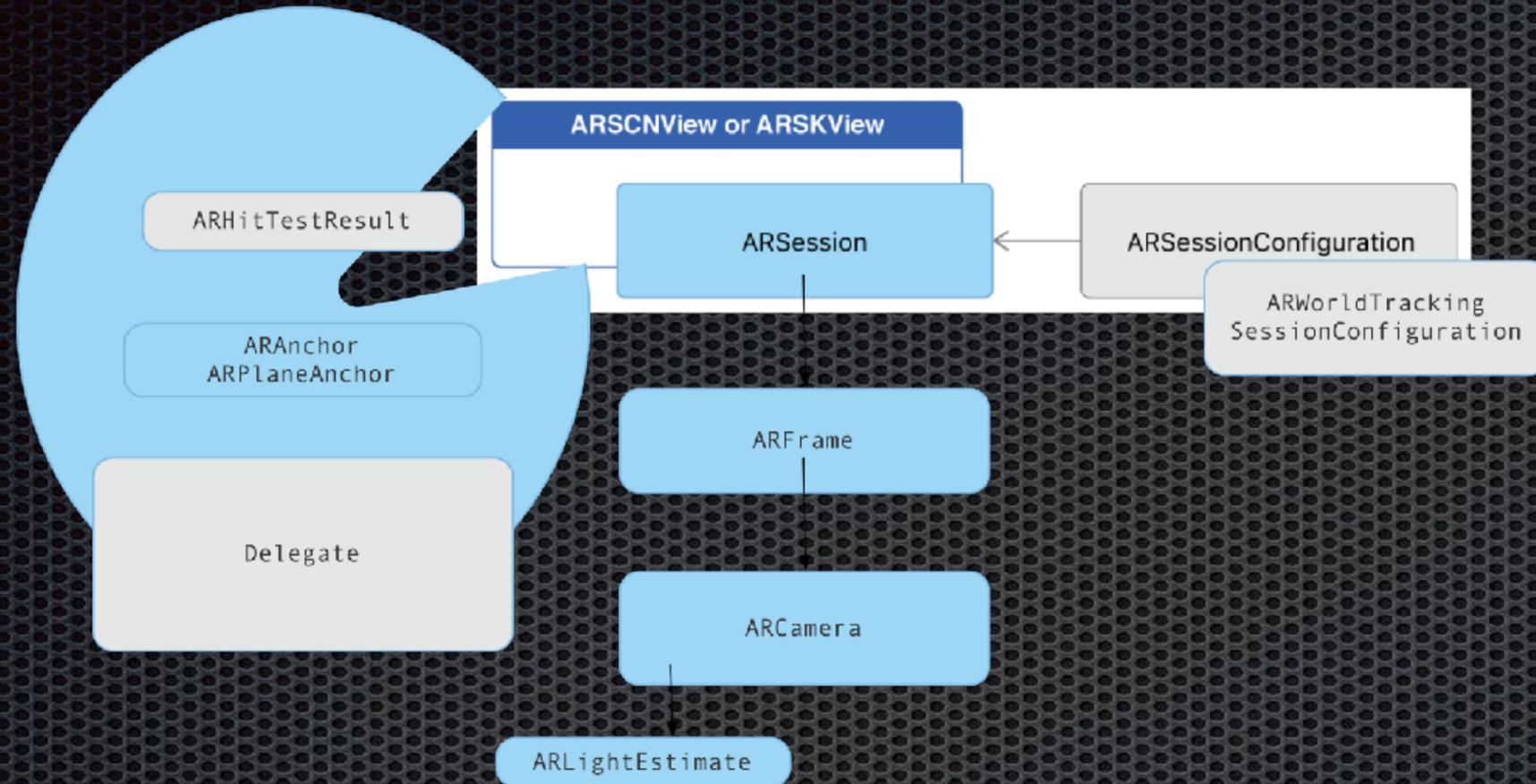
- Methods you can implement to mediate the automatic synchronization of SceneKit content with an AR session.

(Swift) **renderer(_:nodeFor:)**
(Objective-C) **renderer:nodeForAnchor:**

- Methods you can implement to mediate the automatic synchronization of SpriteKit content with an AR session.

(Swift) **view(_:nodeFor:)**
(Objective-C) **view:nodeForAnchor:**

ARKit: Objects



So simple: Most classes are already somewhere instantiated, and not of interests.

ARSession

ARSCNView

ARSKView

(Custom Views; Metal)

ARSessionConfiguration

ARWorldTrackingSessionCon...

ARCamera

ARFrame

ARLightEstimate

ARAnchor

ARPlaneAnchor

ARHitTestResult

ARKit: Configuration

A configuration is mandatory to create a session. There are two types of it: **ARSessionConfiguration** and **ARWorldTrackingSessionConfiguration**

The **ARWorldTrackingSessionConfiguration** class tracks the device's movement with six degrees of freedom (6DOF): specifically, the three rotation axes (roll, pitch, and yaw), and three translation axes (movement in x, y, and z).

World tracking is needed for plane detection.

```
// Create a session configuration
let configuration = ARWorldTrackingSessionConfiguration()
configuration.worldAlignment = .gravityAndHeading
configuration.planeDetection = .horizontal // only
// Run the view's session
sceneView.session.run(configuration)
```

```
camera; gravity;
gravityAndHeading [0;1;2]
```

ARKit: Support

To make an app available only on devices supporting ARKit, use the

arkit key in the **UIRequiredDeviceCapabilities** section of your app's **Info.plist**. If augmented reality is a secondary feature of the app, the

isSupported

property is used to determine whether the current device supports the session configuration in use.

ARKit: Anchor

References for virtual locations in the image of the real world.
An anchor contains an identifier and a transformation matrix.

identifier

A unique identifier for the anchor.

transform

A matrix encoding the position, orientation, and scale of the anchor relative to the world coordinate space of the AR session the anchor is placed in.

Anchors can be added manual or automatically.

ARKit: Anchor

Anchors are the objects bridging the virtual and real world scene. They are send to the **delegate** either requesting a renderer (**SceneKit**) or an instance of a view (**SpriteKit**).

ARSCNViewDelegate

`renderer:nodeForAnchor:`

Asks the delegate to provide a SceneKit node corresponding to a newly added anchor.

ARSKViewDelegate

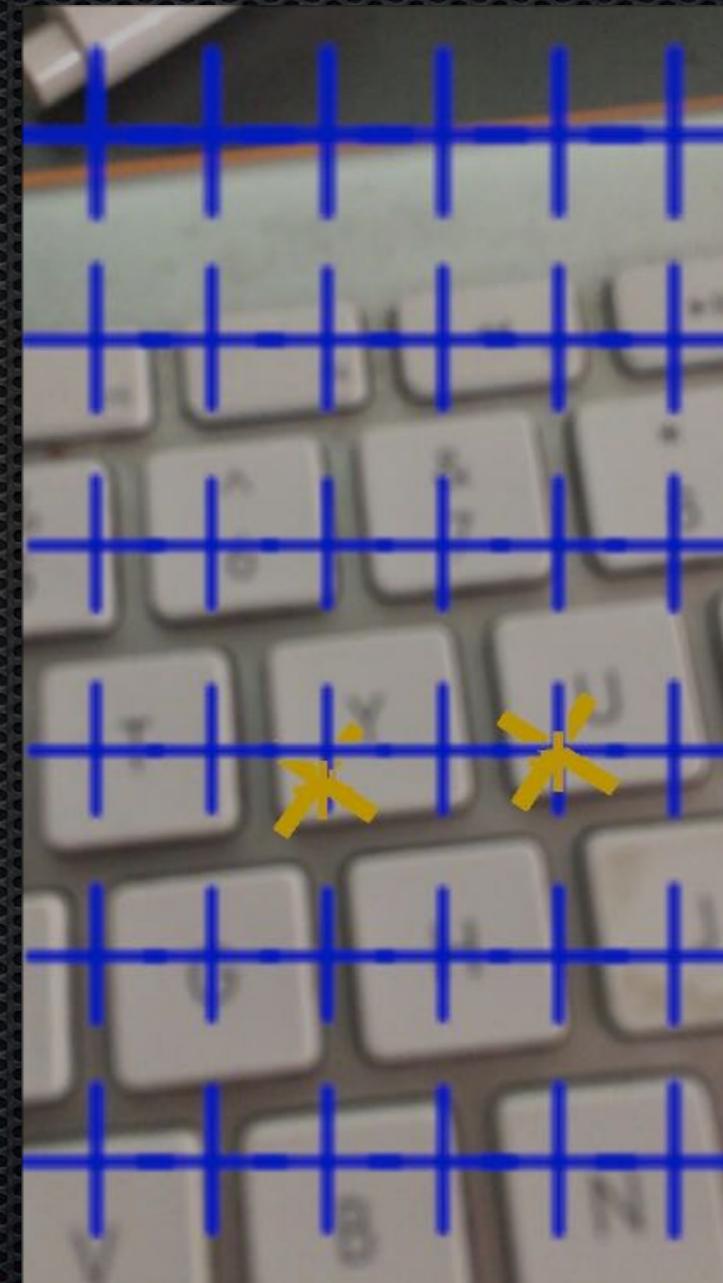
`view:nodeForAnchor:`

Asks the delegate to provide a SpriteKit node corresponding to a newly added anchor.

ARKit: Detection of Features



All kind of features, regardless



Detected features

ARKit: Geometric Calculations

Almost all classes have or can handle three-dimensional informations and calculations like vectors, transformations or the matrix game.

Although this is what AR should all about, its content, it is not of interests here.

The content, the virtual object are only a part of it, and there are already dedicated frameworks in place, namely SpriteKit and SceneKit, but also Metal and some third party frameworks like Unity.

This task is best left to fellow developers, and for future presentations.

ARKit: Example?

For everybody: *Xcode, new project, iOS Augmented Reality App, SpriteKit*

In ViewController.swift

```
in viewWillAppear:  
    configuration.worldAlignment = .gravityAndHeading
```

```
in view(_:nodeFor:)  
    let labelNode = SKLabelNode(text: "🎈")
```

Just change
the Emoji to
"Balloon"

In Scene.swift

```
in touchesBegan(_:with:)  
    let translation.columns.3.z = -0.8
```

ARKit: The End

Thank you.