

# Interpreting Movement

Core Motion on iOS 8

PDX CocoaHeads  
February 25th, 2015

# Broad Overview

Factors to Consider

# Sensors & Hardware

# API History

# iOS 8 Demos

# iOS 8 Demos

- Motion Activity
- Real-Time Pedometer
- Historical Pedometer
- Altimeter
- CLFloor

Apple Watch



More Resources

Factors to Consider

Raw v. Interpreted

Real-time v. Historical

# Real-time: Pace of Updates

- Continuous/Sub-second
- Seconds or Minutes

# Push v. Pull

- Do you ask some kind of manager object to send updates/notifications at a pace defined by iOS?
- Does the update interval depend on the device?
- Or can you request data at a moment of your choosing?

# Historical

Limited Availability

# Device Support



# Device Support

Devices that support the version of iOS you're targeting,  
but don't support the APIs you'd like to use.

# Human Factors

# Human Factors

- Required actions

# Human Factors

- Required actions: start, stop, configure, calibrate

# Human Factors

- Required actions: start, stop, configure, calibrate
- Possible disruptions (to real-time data collection)

Interpretations

# Interpretations

- Altitude relative to what?

# Interpretations

- Altitude relative to what?
- What does it mean to ascend or descend a floor?



# Interpretations

- Altitude relative to what?
- What does it mean to ascend or descend a floor?
- Does the data your app presents match user expectations?

# Sensors & Hardware

# Accelerometer

# Accelerometer

- Measures acceleration in three dimensions
- Introduced with original iPhone
- Drives rotation and screen orientation

Magnetometer

# Magnetometer

- Strength and direction of the magnetic field
- Added in the iPhone 3GS
- Powers the Compass app
- Sometimes requires calibration (figure-8 motion)

Gyroscope

# Gyroscope

- Orientation/attitude
- Introduced in the iPhone 4



# M7

An energy-efficient, motion co-processing chip

# M7

Collects, processes, and stores data from motion sensor  
to free up the CPU for other tasks

# M7

Energy efficient?

# M7

“How efficient is our motion processing system? You get 24 hours of motion activity pedometer for about three minutes of a FaceTime call.”

Source: WWDC 2014 Session 612: Motion Tracking with the Core Motion Framework

<http://asciiwwdc.com/2014/sessions/612>

# M7

iPhone 5s

# M7

iPhone 5s

iPad Air\*, iPad Mini 2\*, iPad Mini 3\*

# M8

An upgrade of the M7

# M8

iPhone 6 and 6 Plus



# M8

iPhone 6 and 6 Plus

iPad Air 2\*

# Altimeter

Barometer-based, processed by M8

# The History of Motion on iOS

iOS 2

# iOS 2

UIDevice interprets the accelerometer as:

```
@property(nonatomic, readonly) UIDeviceOrientation  
orientation
```

# iOS 2

UIDevice interprets the accelerometer as:

```
@property(nonatomic, readonly) UIDeviceOrientation  
orientation
```

Or you can ask it to send notifications:

- (void)beginGeneratingDeviceOrientationNotifications
- (void)endGeneratingDeviceOrientationNotifications

```
UIDeviceOrientationDidChangeNotification
```

iOS 3

# iOS 3

**UIResponder** interprets the accelerometer and sends motion events:

- `(void)motionBegan:(UIEventSubtype)motion  
withEvent:(UIEvent *)event`
- `(void)motionEnded:(UIEventSubtype)motion  
withEvent:(UIEvent *)event`
- `(void)motionCancelled:(UIEventSubtype)motion  
withEvent:(UIEvent *)event`



# iOS 3

What event types are supported?

`UIEventTypeMotion`

`UIEventSubtypeMotionShake`

iOS 4

# iOS 4

Motion gets a framework! And a manager!

# iOS 4

`CMMotionManager` provides access to data from two of the three motion sensors:

- Accelerometer
- Gyroscope
- ~~Magnetometer~~

# iOS 4

`CMMotionManager` also establishes some patterns in the design of the motion APIs.

# iOS 4

`UIMotionManager` also establishes some patterns in the design of the motion APIs.

(The following are representative examples, not a complete overview of the API changes.)

# iOS 4

Checking device capability:

```
@property(readonly, nonatomic, getter=isGyroAvailable) BOOL  
gyroAvailable
```

Note that this is a property, *not* a class method.

# iOS 4

Pulling data on demand:

```
@property(readonly) CMAccelerometerData *accelerometerData
```



# iOS 4

Asking the manager to push data:

- `(void)startGyroUpdatesToQueue:(NSOperationQueue *)queue  
withHandler:(CMGyroHandler)handler`

# iOS 4

Asking the manager to push data:

- `(void)startGyroUpdatesToQueue:(NSOperationQueue *)queue  
withHandler:(CMGyroHandler)handler`

But what's the frequency?

# iOS 4

Request a frequency (in seconds):

```
@property(assign, nonatomic) NSTimeInterval gyroUpdateInterval
```

# iOS 4

Request a frequency (in seconds):

```
@property(assign, nonatomic) NSTimeInterval gyroUpdateInterval
```

However...

# iOS 4

Request a frequency (in seconds):

```
@property(assign, nonatomic) NSTimeInterval gyroUpdateInterval
```

From the docs:

*"The value of this property is capped to minimum and maximum values; the maximum value is determined by the maximum frequency supported by the hardware. If your app is sensitive to the intervals of gyroscope data, it should always check the timestamps of the delivered CMGyroData instances to determine the true update interval."*



# iOS 5

`CMMotionManager` provides access to data from **all three** motion sensors:

- Accelerometer
- Gyroscope
- Magnetometer

# iOS 5

Note that this property:

```
@property(readonly) CMMagnetometerData *magnetometerData
```

Does **not** return a cardinal direction, e.g. North.

It returns a “raw” magnetic field value.



# Tangent: CLHeading

CoreLocation (not CoreMotion)

Available since iOS 3.0

# Tangent: CLHeading

`CLLocationDirection` is a double which represents degrees: 0 is North, 90 degrees is East, et cetera:

```
@property(readonly, nonatomic) CLLocationDirection magneticHeading
```

```
@property(readonly, nonatomic) CLLocationDirection trueHeading
```

# Tangent: CLHeading

CLHeading also has "raw heading data" for the x, y, and z-axis.

From the documentation of the x property:

*"The geomagnetic data (measured in microteslas) for the x-axis.*

...

*This value represents the x-axis deviation from the magnetic field lines being tracked by the device."*

# Tangent: CLHeading

You can access magnetometer data via:

- CMMotionManager's CMMagnetometer
- CMDeviceMotion's CMCalibratedMagneticField property
- CLLocationManager's CLHeading

A more detailed exploration:

<http://stackoverflow.com/a/15470571>



nil



# iOS 7

Interpreted Data



# iOS 7

Interpreted/Derived/Processed/Stored by the M7

# CMStepCounter

Steps and Distance

# CMStepCounter

Steps and Distance  
(iPhone 5s only)

# CMStepCounter

How do we know if we can use it?

+ (BOOL)isStepCountingAvailable

# CMStepCounter

How do we know if we can use it?

+ (BOOL)isStepCountingAvailable

Note: this is not a property of an initialized manager, as we saw before. It's a class method.

# CMStepCounter

How do we know if we can use it?

+ (BOOL)isStepCountingAvailable

Note: this is not a property of an initialized manager, as we saw before. It's a class method.

Invoking class methods to determine device capability **does not** trigger “Motion Activity” permission requests!

# CMMotionActivity Manager

# CMMotionActivity Manager

How is the user moving?



# CMMotionActivity

Delivered by CMMotionActivityManager

# CMMotionActivity

BOOL values:

- stationary
  - walking
  - running
  - automotive
  - unknown
- 
- startDate (time of last change)
  - confidence



# iOS 8

(with the new M8 chip)

# iOS 8

(with the new M8 chip)  
(iPhone 6 and 6 Plus only)

# iOS 8

- `CMPPedometer` replaces the now deprecated `CMStepCounter`
- `CMMotionActivity` now has a `bicycling` property
- `CMAltimeter` shares data from the new barometer

Hey, what about the asterisk  
next to the iPads Air and  
their Mini counterparts?!

Hey, what about the asterisk  
next to the iPads Air and  
their Mini counterparts?!

(...which have the new M7/M8 chips, right?!?)



iPads do not support  
pedometer functions.

(...even those that have the new M7/M8 chips.)

Tangent:  
UIRequiredDeviceCapabilities

# UIRequiredDeviceCapabilities

Defined values for this key in Info.plist include:

- accelerometer
- gyroscope
- magnetometer

# UIRequiredDeviceCapabilities

Defined values for this key in Info.plist include:

- accelerometer
- gyroscope
- magnetometer

Defined values **do not** include:

- pedometer
- altimeter

# iOS 8 Demos

<https://github.com/mattblair/CoreMotionDemos>

# Real-Time Motion Activity

# Real-Time Motion Activity

- Running is most accurate, cycling least
- Walking with a phone in hand is harder to detect than when it's in a pocket
- Also supports historical queries, which often need filtering
- WWDC 2014 session #612: Motion Tracking with the Core Motion Framework has lots of details

# Real-Time Pedometer



# Real-Time Pedometer

- The estimates get more accurate over time
- Historical data is more likely to be accurate, since it's better able to assess what is and isn't walking in retrospect
- ***Really difficult*** to test while tethered to your dev machine!

# Historical Pedometer

Tangent: Formatters

# New Formatters in iOS 8

- NSDateIntervalFormatter
- NSLengthFormatter

# Formatters in Action



What does a floor  
mean anyway?

# What does a floor mean anyway?

- Actual stair steps seem accurate to me
- Elevators and escalators are not counted

# What does a floor mean anyway?

But what about:

- Gradual slopes?
- Walking up or down an escalator?
- Do half staircases accumulate over time?



# What does a floor mean anyway?

On a hike I take in Forest Park:

- Altimeter reported about 220 meter gain at peak
- CMPedometer docs say a floor is about 3 meters
- The same hike reported 55 floors ascended

# What does a floor mean anyway?

On a hike I take in Forest Park:

- Altimeter reported about 220 meter gain at peak
- CMPedometer docs say a floor is about 3 meters
- The same hike reported 55 floors ascended
- So floors ascended is about 75% of altitude? No.

# What does a floor mean anyway?

- Floors ascended is cumulative: if I go up 10 floors, down 5, then up 7, I have ascended 17 floors.
- But altitude is relative: in the same situation, it would estimate an altitude delta equivalent to about +12 floors.

# What does a floor mean anyway?

- Beta testers for one of my apps regularly walk a bridge with a very gradual slope – maybe 10-15 degrees – and a max height of 66 feet.

# What does a floor mean anyway?

- Beta testers for one of my apps regularly walk a bridge with a very gradual slope – maybe 10-15 degrees – and a max height of 66 feet.
- On most of their walks, floors ascended/descended reports 0!

Altimeter

# Altimeter Results

- Value is a delta relative to when you start updates
- Forgot to start updates? Oops. This is real-time only.
- Doesn't reset when backgrounded or when the screen is turned off (I've kept it going 2+ hours)
- Approximately 3-6 seconds of latency before values stabilize after foregrounding
- After that 220 m ascent hike, CMAltimeter reported my relative altitude as +3.6 m when I returned to my starting point

CLFloor



# CLFloor

(CL?!?)

# Tangent: CLFloor

Yes, it's CoreLocation

From the CLFloor docs:

*"In places where floor information can be determined, a CLLocation object may include a floor object along with the regular location data."*

If not available for a location, the floor object is nil.

A distant cousin to CMAltitude, intended for indoor mapping.

Docs:

*"It is erroneous to use the user's level in a building as an estimate of altitude."*

# Who is using CLFloor?

- Westfield SF Mall (?)
- San Jose Airport (?)
- ???

Source: WWDC 2014 Session 708:  
“Taking Core Location Indoors”  
<http://asciiwwdc.com/2014/sessions/708>

How do you register a  
CLFloor?

# How do you register a CLFloor?

According to WWDC 2014 Session 708:

<https://mapsconnect.apple.com/>



“Hmm...maybe I'll register the decks of the Steel Bridge...”



"Portland Steel Bridge with lift span raised - viewed from west" by Steve Morgan - Own work. Licensed under CC BY-SA 3.0 via Wikimedia Commons - [http://commons.wikimedia.org/wiki/File:Portland\\_Steel\\_Bridge\\_with\\_lift\\_span\\_raised\\_-\\_viewed\\_from\\_west.jpg#mediaviewer/File:Portland\\_Steel\\_Bridge\\_with\\_lift\\_span\\_raised\\_-\\_viewed\\_from\\_west.jpg](http://commons.wikimedia.org/wiki/File:Portland_Steel_Bridge_with_lift_span_raised_-_viewed_from_west.jpg#mediaviewer/File:Portland_Steel_Bridge_with_lift_span_raised_-_viewed_from_west.jpg)

## Read Me Before Signing Up

Thank you for your interest in Apple's new indoor positioning technology. We have received an overwhelming response on this service and we are prioritizing our efforts to focus on venues with the following attributes:

---

- Accessible to the general public
  - Annual visitors in excess of 1 million per year
  - Availability of complete, accurate, and scaled reference maps
  - Enabled with Wi-Fi throughout
  - Associated app is authorized by venue owner
- 

After logging in at:

<https://mapsconnect.apple.com/>

# Expectations & Interpretations

- In our daily experiences, altitude, the floor number of a building, and the number of stairs we've climbed (or avoided) are interconnected in obvious ways
- The values returned by these APIs are quite distinct, disconnected, and may be incongruent with each other, and user expectations

Apple Watch

# Apple Watch

- Has an accelerometer
- Has a gyroscope

# Apple Watch ?

- Maybe it just synchronizes data to the motion activity and pedometer datastore automatically?
- Will we get direct, real-time access to the sensors?
- Three-dimensional gestures?
- Events with gravitational force value we could use as a control parameter for \_\_\_\_?

More Resources

# WWDC 2014 Session 612

“Motion Tracking with the Core Motion Framework”

Does not cover M8-related API, which was not announced at the time:

- CMPedometerData’s floorsAscended, floorsDescended properties
- CMAltimeter

<http://asciiwwdc.com/2014/sessions/612>



# WWDC 2014 Session 612

MotionActivityDemo Source Code

[https://developer.apple.com/wwdc/resources/  
sample-code/](https://developer.apple.com/wwdc/resources/sample-code/)

# NSHipster: CMDeviceMotion

- Accelerometer
- Accelerometer + Gyroscope
- “Gravitational” gestures
- Attitude
- Some code examples are in Swift

<http://nshipster.com/cmdevicemotion/>

# Pure Swift

Sam Davies of Shinobi Controls: iOS 8 Day-by-Day  
#35: CoreMotion

<http://www.shinobicontrols.com/blog/posts/2014/10/21/ios8-day-by-day-day-35-coremotion>

<https://github.com/ShinobiControls/iOS8-day-by-day/tree/master/35-coremotion/LoCoMotion>

# More Resources

Check out the README in the demo source on GitHub

# Contact

Matt Blair

[matt@mattblair.net](mailto:matt@mattblair.net)

<http://elsewiseapps.com>

Demo Source Code:

<https://github.com/mattblair/CoreMotionDemos>