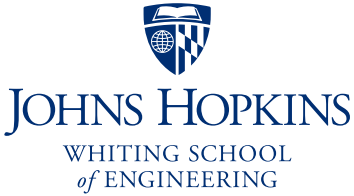
Johns Hopkins Engineering

Location Services and Frameworks

Module 4



CoreMotion Framework

The iPhone Frame

The accelerometer and gyroscope report data along 3 axes of the device. The iPhone is shown in the image to the right. The iPad has the same set of axes.

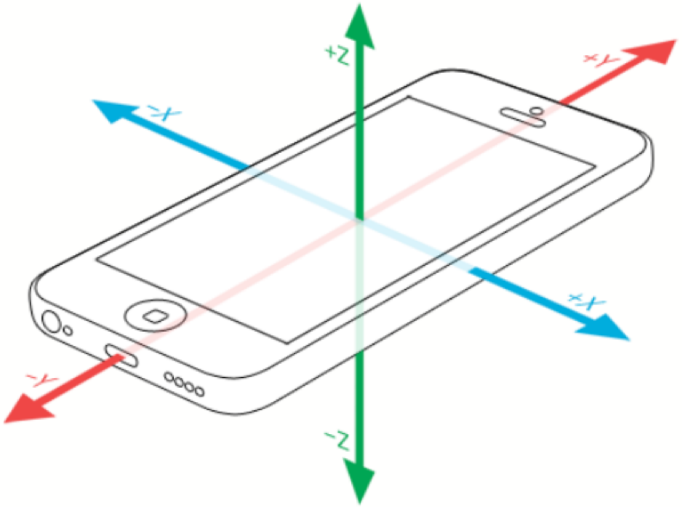


Image from: http://nshipster.com/cmdevicemotion/

CMMotionManager

- Class that provides access to all motion data on the device
- Can either be pulled (given upon request) or pushed (start collection and get notified when it changes)
- Motion types: accelerometer, gyro, magnetometer, deviceMotion

http://nshipster.com/cmdevicemotion/

Reading the Accelerometer

- Always check to see if your motion source is available!
  - `manager.isAccelerometerAvailable`
- Choose an update interval
- You can send updates to a queue (for example, .main)
  - `manager.startAccelerometerUpdates(to: .main) { }`
- x,y,z components give amount of *g* in that direction
- Even sitting still, you get accelerometer readings, because gravity is always acting on it on at least one axis

http://nshipster.com/cmdevicemotion

Reading the Gyroscope

- Can use `isGyroAvailable` for gyro only, or `isDeviceMotionAvailable` for both accelerometer and gyroscope data!

- `startGyroUpdates(to:)` or `startDeviceMotionUpdates(to:)` can be used to get updates sent to a queue

- For all the motion types don't forget to stop getting updates when you are done (e.g. `stopGyroUpdates()`)

http://nshipster.com/cmdevicemotion

Detecting a Shake

- `UIViewControllers` have 3 motion related delegate methods: `motionBegan`, `motionEnded` and `motionCancelled`
- You can check the type of the `motion` object that is passed to this method, and see if it of type `UIEventSubtype.motionShake.`
- If so, you can write your own custom response (e.g. shake to undo)

Reading Altimeter/Altitude

- `CMAltimeter` allows you to detect *changes* in the user's altitude on certain devices (use `isRelativeAltitudeAvailable()`!)
- `startRelativeAltitudeUpdates(to:withHandler:)` to get updates a regular intervals (whether the value has changed or not)
- Events are only delivered while your app is running, either in background or foreground

https://developer.apple.com/documentation/coremotion/cmaltimeter

Reading the Magnetometer

- Set the magnetometerUpdateInterval
- Use startMagnetometerUpdates(to:withHandler:)
- Process the CMMagnetometerData passed into your handler
- Beware – cases with magnetic clasps in them can interfere with your magnetometer readings (especially if you are trying to use the compass)

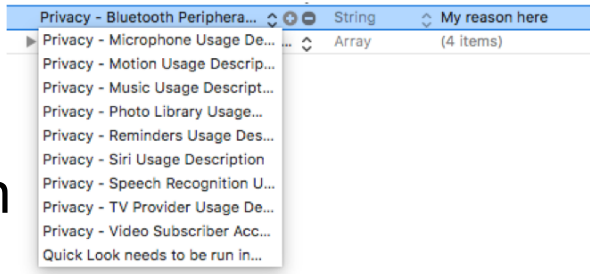
Reading the Pedometer

- Modern devices (with the M7 and later motion co-processor) have the ability to capture steps
- `CMPedometer` allows access to the historical data, such as steps walked, or distance traveled, via the `queryPedometerData(from:to:withHandler:)` method
- You can also get access to live data via `startUpdates(from:withHandler:)`

https://developer.apple.com/documentation/coremotion/cmpedometer

Asking for Permission

- Whenever you want to access data such as motion data (and others such as location, bluetooth, etc), you MUST ask the user for permission. For motion:
  - In your Info.plist file, set the “Privacy - Motion Usage Description” key with a value describing why you want to use this user data in your app



- For other types such as location, you have to call `manager.requestWhenInUseAuthorization()`