SCIENCES SORBONNE UNIVERSITÉ

CMMotionManager

Fabrice.Kordon@lip6.fr

LIP6

CMMotionManager?

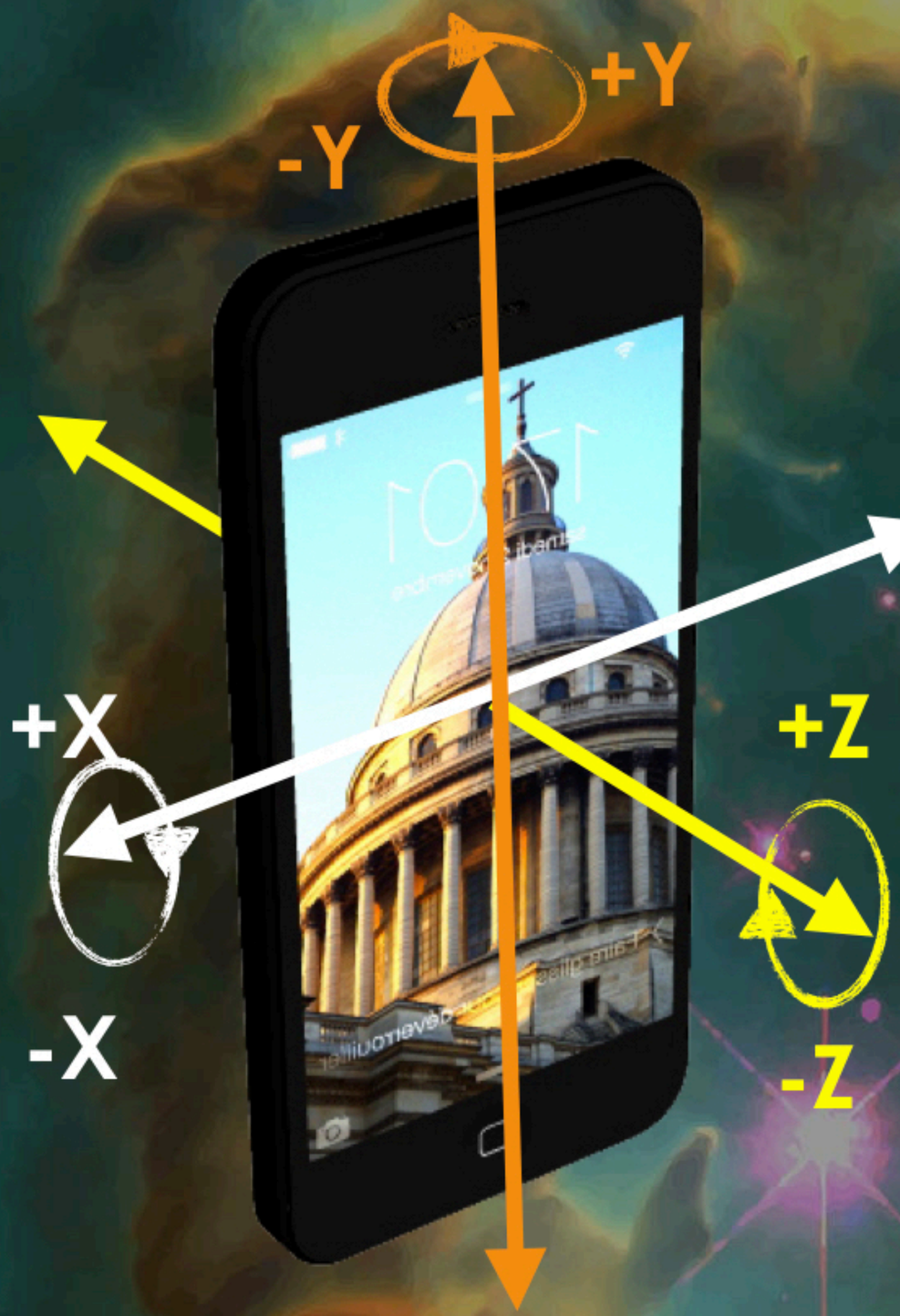
Handles a group of sensors

- Requires to import the CoreMotion framework
  - **Objective-C : CoreMotion/CoreMotion.h**
  - **Swift : CoreMotion**

Data about device's moves

- Accelerometer, gyroscope, magnetometer
  - **Some data may be unavailable on some devices**
- A common approach for sampling
  - **Activation/deactivation, access to data**
- Separated handling of related information

Available sensors



Accelerometer

- sensor to measure acceleration
  - **Strength in G on 3 axes**

Gyroscope

- Sensor to measure orientation
  - **Rotation on the 3 axes**

Magnetometer

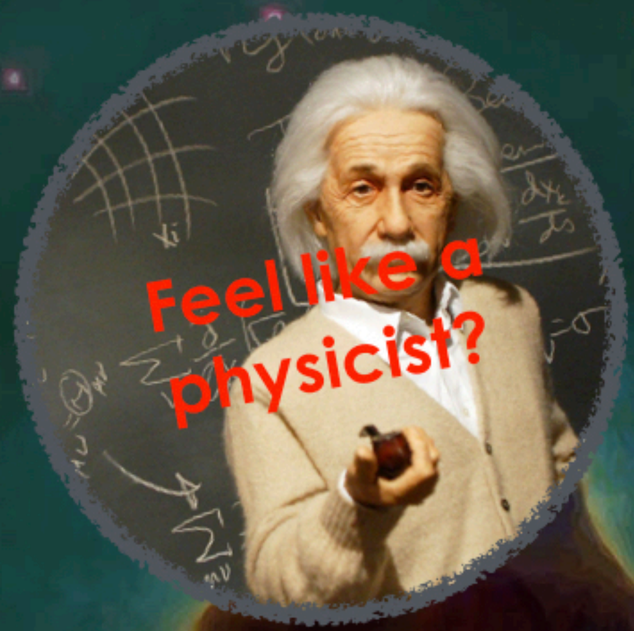
- Sensor to measure magnetic field
  - **Magnetism in µT on the 3 axes**

Available information

- **Property deviceMotion (CMDeviceMotion)**
  - In a CMMotionManager
- **CMDeviceMotion properties**
  - attitude
    - **Device orientation (roll, pitch, yaw)**
  - rotationRate
    - **Raw data from the gyroscope (x, y, z)**
  - gravity
    - **Gravity + device acceleration (x, y, z)**
  - userAcceleration
    - **Device acceleration (x, y, z)**
  - magneticField
    - **Magnetic field (earth + environment) without the bias induced by the device (x, y, z)**

Available information

- **Property deviceMotion (CMDeviceMotion)**
  - In a CMMotionManager
- **CMDeviceMotion properties**
  - attitude
    - **Device orientation (roll, pitch, yaw)**
  - rotationRate
    - **Raw data from the gyroscope (x, y, z)**
  - gravity
    - **Gravity + device acceleration (x, y, z)**
  - userAcceleration
    - **Device acceleration (x, y, z)**
  - magneticField
    - **Magnetic field (earth + environment) without the bias induced by the device (x, y, z)**



Principles

Creation

- As usual
- Recommendation, have only one copy per Application
  - **Careful with accelerometerUpdateInterval (TimeInterval)**

Sampling activation

- startDeviceMotionUpdates, startMagnetometerUpdates, startGyroUpdates

Sampling deactivation

- stopDeviceMotionUpdates, stopAccelerometerUpdates, stopGyroUpdates

Data retrieving

- Two techniques

Retrieving data, technique 1

Thanks to a handler

```
func startDeviceMotionUpdates(to queue: OperationQueue,
                              withHandler handler: @escaping CMDeviceMotionHandler)

func startAccelerometerUpdates(to queue: OperationQueue,
                               withHandler handler: @escaping CMAccelerometerHandler)

func startGyroUpdates(to queue: OperationQueue,
                      withHandler handler: @escaping CMGyroHandler)

func startMagnetometerUpdates(to queue: OperationQueue,
                              withHandler handler: @escaping CMMagnetometerHandler)
```

- Handler's profile
  - (DATA?, Error?) -> void
  - NSError for Objective-C
- OperationQueue?
  - Suggestion 1 : OperationQueue.current
  - Suggestion 2 : OperationQueue.main

Updates?

- Regulated by accelerometerUpdateInterval

Retrieving data, technique 2

Use the CMMotionManager

- If you have a reference to it
- Involved attributes
  - **deviceMotion (CMDeviceMotion)**
  - **accelerometerData (CMAccelerometerData)**
  - **gyroData (CMGyroData)**
  - **magnetometerData (CMMagnetometerData)**

Sampling interval?

- When you need it
- Possible to use a Timer
  - **Which is integrated in the other method**

As a conclusion…

- Let's pretend it is easy

- However…
  - Be careful… such services are high energy
    - Especially on old devices