iPhone Programming
Core Motion

Tim Gegg-Harrison

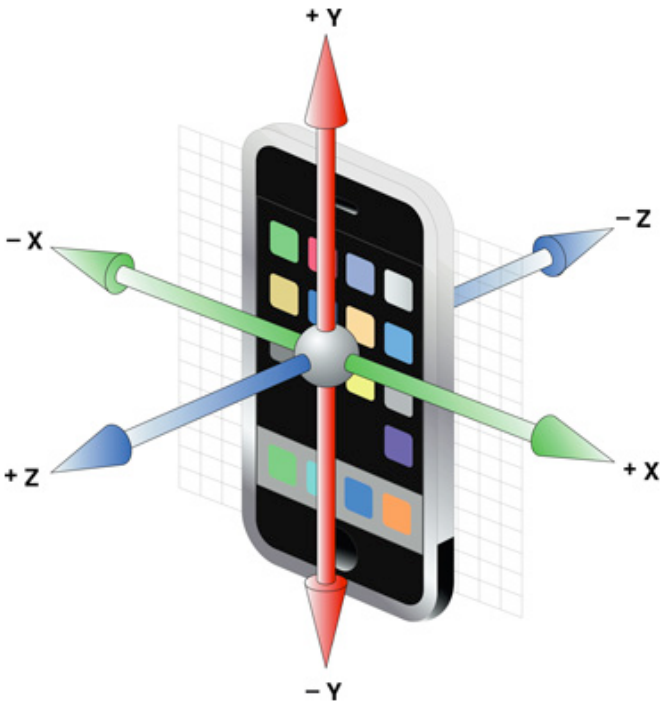
Motion Events

- UIEvent
- UIEventType.touches
  - touching the device
    - touchesBegan
    - touchesMoved
    - touchesEnded
- UIEventType.motion
  - shaking the device
    - motionBegan
    - motionEnded

Core Motion

- Core Motion allows a developer to observe and respond to the motion and orientation of an iOS device by inspecting the raw and processed data from a combination of built-in sensors, including the accelerometer, gyroscope, and magnetometer.
- Both accelerometer and gyroscope data are presented in terms of three axes that run through the iOS device.
  - For an iPhone held in portrait orientation, the X-axis runs through the device from left (negative values) to right (positive values), the Y-axis through the device from bottom (—) to top (+), and the Z-axis runs perpendicularly through the screen from the back (—) to the front (+).

Core Motion …



Core Motion …

- The CoreMotionManager class provides access to all the motion data on an iOS device.
- Core Motion provides both "pull" and "push" access to motion data.
  - To "pull" motion data, you can access the current status of any sensor or the composited data as read-only properties of CoreMotionManager.
  - To receive "pushed" data, you start the collection of your desired data with a block or closure that receives updates at a specified interval.
- CoreMotionManager provides a consistent interface for each of the four motion data types: accelerometer, gyro, magnetometer, and deviceMotion.

Core Motion …

- Check for availability:

```
let manager = CMMotionManager()
if manager.accelerometerAvailable {
     // ...
}
```

Core Motion …

- Define the update interval:

```
manager.accelerometerUpdateInterval = 0.01
```

Core Motion …

- Start updates:
  - "Pull" data (**`manager.accelerometerData`** is accessible at any time with the devices' current accelerometer data):

```
manager.startAccelerometerUpdates()
```

  - "Push" data (handler closure will be called at the frequency given by the update interval):

```
manager.startAccelerometerUpdates(
    to: OperationQueue.main) withHandler: {
    (data: CMDeviceMotion?, error: Error?) in
    // …
}
```

Core Motion …

- Stop updates:

```
manager.stopAccelerometerUpdates()
```