

Sensores

Introducción

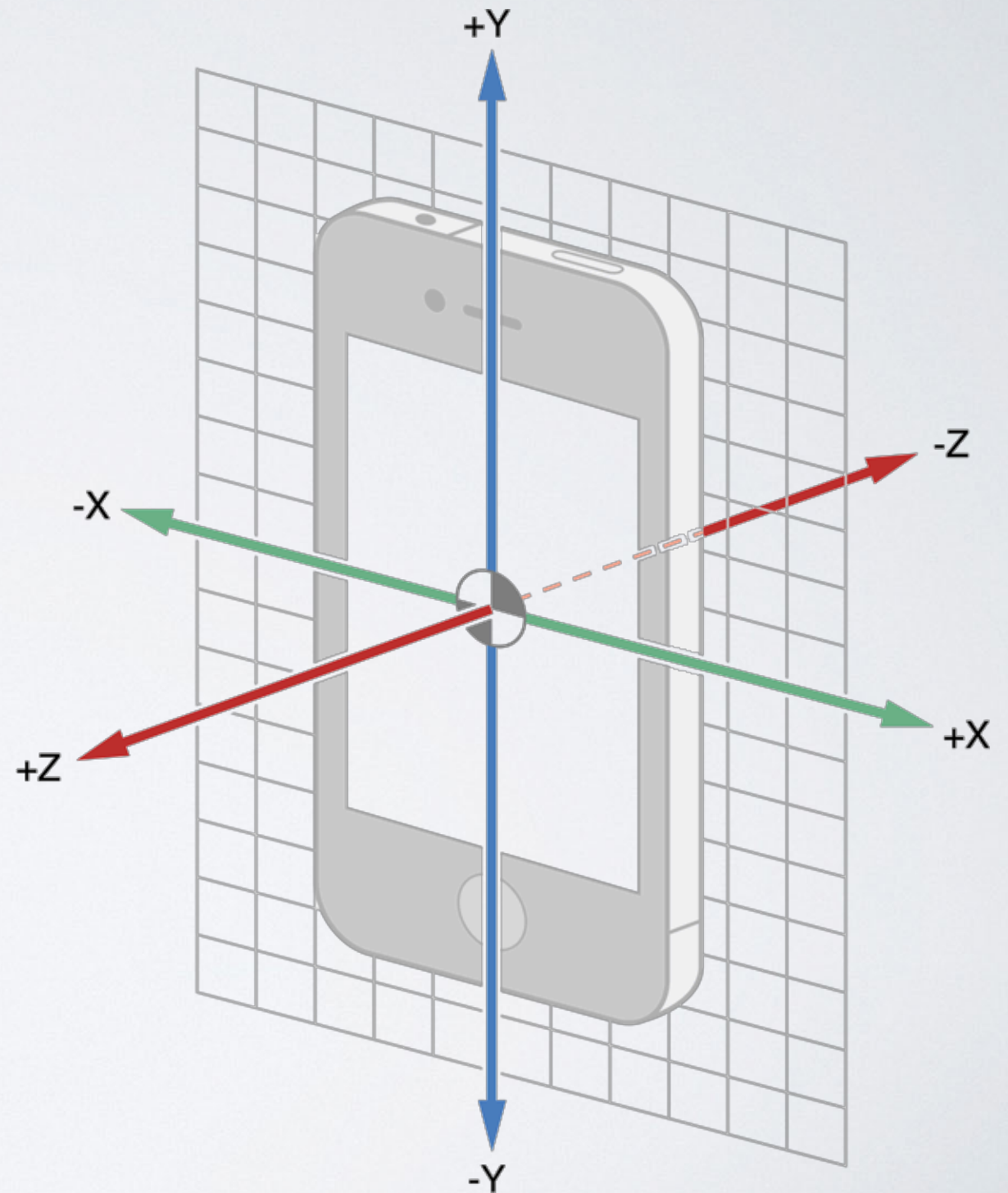
Los dispositivos iOS de última generación disponen de múltiples sensores de distintos tipos que pueden mejorar la experiencia de usuario.

Para probar la mayoría de esas características necesitaremos un dispositivo real, no el simulador de Xcode.

Core Motion

Framework de Apple para leer y procesar los valores de movimiento y orientación de un dispositivo iOS a través de los sensores integrados (acelerómetro, giroscopio y barómetro).

Los datos del acelerómetro y del giroscopio se presentan en función de tres ejes dispuestos como se ve en la imagen de la derecha.



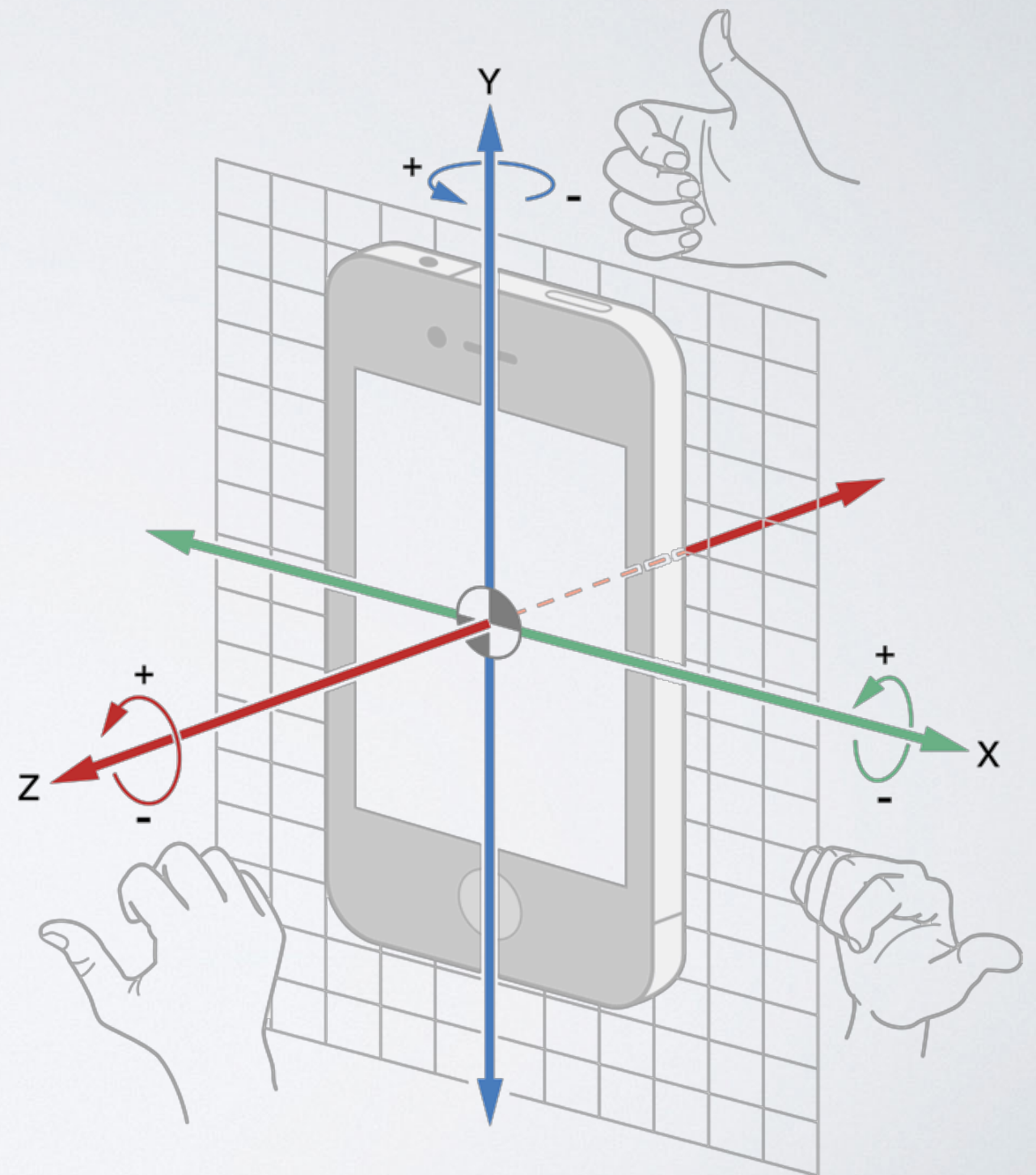
Acelerómetro

Mide la aceleración y la fuerzas inducidas por la gravedad, permitiéndonos detectar el movimiento y el giro de un dispositivo.

Suele utilizarse en juegos: por ejemplo para conducir con el teléfono a modo de volante o guiar una bola por un laberinto como si fuera una caja real.

Giroscopio

Permiten monitorear y controlar posiciones del dispositivo como la orientación, la dirección, el movimiento angular y la rotación.



Magnetómetro

Se trata de un componente electrónico capaz de medir y cuantificar la cantidad de fuerza magnética de un objeto. O para lo que muchos dispositivos lo usan, como **brújula**, detectando el polo norte magnético.

CMMotionManager

- Nos permite acceder a los datos de movimiento del dispositivo.
- Por cuestiones de rendimiento, Apple recomienda usar una única instancia CMMotionManager en toda la aplicación.
- CMMotionManager proporciona una interfaz uniforme para cada uno de los cuatro tipos de movimiento de datos: acelerómetro, giroscopio, magnetómetro y el movimiento del dispositivo.

Cómo usar CMMotionManager

```
let manager = CMMotionManager()
```

```
manager.gyroUpdateInterval=0.2
```

```
manager.startGyroUpdates()
```

```
manager.stopGyroUpdates()
```


Podómetro

Cuenta el número de pasos que hemos realizado.
Los dispositivos que no cuentan con este
componente usan el acelerómetro.

Disponible a partir de iPhone 5s.

Podómetro

Se maneja con la clase CMPedometer:

```
pedometer.startUpdates(from: Date()) {  
    (data, error) in  
  
        data!.numberOfSteps // numero de pasos  
        data!.distance?.doubleValue // distancia en metros  
        data!.floorsAscended?.doubleValue // numero de plantas ascendidas  
        por escaleras  
    }  
}
```

Altímetro

Da información sobre la altitud del dispositivo a través de la clase `CMAltimeter`:

```
altimeter.startRelativeAltitudeUpdates(to:  
dataProcessingQueue) {  
    (data, error) in  
  
    data!.relativeAltitude.doubleValue // altitud relativa en metros  
}
```


Actividades

A través de la clase `CMMotionActivityManager`, también podemos saber si el usuario está parado, andando, corriendo, montando en bici o en un vehículo.

Actividades

```
activityManager.startActivityUpdates(to: OperationQueue.main) {  
    data in  
  
    if data!.running {  
        // corriendo  
    } else if data!.cycling {  
        // montando en bici  
    } else if data!.walking {  
        // andando  
    } else if data!.automotive {  
        // en un vehículo  
    }  
}
```

GPS

Permite conocer las coordenadas latitud y longitud del dispositivo físico, así como determinar rutas.

El framework que ofrece Apple para tratar con toda esta información se llama Core Location y normalmente va ligado con el framework de mapas Map Kit.

Sensor de proximidad

Es un transductor que detecta objetos o señales que se encuentran cerca del elemento sensor.

Se basan en un LED infrarrojo y también en un receptor IR. Emite una luz infrarroja y si es reflejada de vuelta al receptor, significa que hay algo muy cerca del dispositivo.



Sensor de proximidad

Para usarlo solo tenemos que hacer:

```
UIDevice.current.isProximityMonitoringEnabled = true
```

Y después escuchar la NSNotification

```
UIDeviceProximityStateDidChange.
```

Orientación del dispositivo

- Usando la clase `UIDevice` podemos conocer la orientación del dispositivo.

```
let orientation = UIDevice.current.orientation
```

- Para escuchar cambios solo hay que suscribirse a la `NSNotification UIDeviceOrientationDidChange`

Agitar el dispositivo

- También podemos detectar cuando el usuario agita el dispositivo en subclases de UIResponder (como UIView y UIViewController)
- Para ello sólo hay que implementar el método `motionEnded` y comprobar si el tipo de movimiento es `shake`.

Agitar el dispositivo

```
override func motionEnded(_ motion: UIEventSubtype, with
event: UIEvent?)
{
    if (motion == UIEventSubtype.motionShake)
    {
        print("Shake motion detected!!")
    }
}
```


Otros sensores

- Sensor de huella (Touch ID)
- Sensor de luminosidad (ALS)
- Sensor de reconocimiento facial (Face ID)

