CORE MOTION

CoreMotion

- API to access motion sensing hardware
  - Accelerometer
  - Gyroscope
    - iPhone4
    - New generation iPod Touch
- Access using CMMotionManager
  - Only one instance allowed per application
    - Create a method to allow all application components to access pointer to single instance

CMMotionManager

- Check availability of accelerometer using
  - BOOL accelerometerAvailable
- accelerometerData
  - CMAccelerometerData *accelerometerData
    - CMAcceleration acceleration
      - double x // x-axis acceleration in G's
      - double y // y-axis acceleration in G's
      - double z // z-axis acceleration in G's

CMMotionManager

- Check availability of gyroscope using
  - BOOL gyroAvailable
- gyroData
  - CMGyroData *gyroData
    - CMRotationRate rotationRate
      - double x // x-axis rotation in radians/s
      - double y // y-axis rotation in radians/s
      - double z // z-axis rotation in radians/s

CMMotionManager

- Check availability of device-motion using
  - BOOL deviceMotionAvailable
- deviceMotion
  - CMDeviceMotion *deviceMotion
    - CMAttitide *attitude
      - double roll // roll of device in radians (around y-axis)
      - double pitch // pitch of device in radians (around x-axis)
      - double yaw // yaw of device in radians (around z-axis)
      - CMRotationMatrix rotationMatrix
      - CMQuaternion quaternion
    - CMRotationRate *rotationRate
    - CMAcceleration *gravity
    - CMAcceleration *userAcceleration

CoreMotion

- 2 ways to get data
  - Start sampling and poll for values
  - Set update interval and register handler block
- Start sampling to enable polling using

  void start{Accelerometer/Gyro/DeviceMotion}Updates
- Stop sampling (when not needed to save resources) using

  void stop{Accelerometer/Gyro/DeviceMotion}Updates

CoreMotion

- Set update interval by setting
  - NSTimeInterval accelerometerUpdateInterval
  - NSTimeInterval gyroUpdateInterval
  - NSTimeInterval deviceMotionUpdateInterval
- Register for accelerometer updates

```
- (void)startAccelerometerUpdatesToQueue:
        (NSOperationQueue *)queue
        withHandler:(CMAccelerometerHandler)handler;
```

  - Implement handler block of type

```
typedef void (^CMAccelerometerHandler)
        (CMAccelerometerData *accelerometerData,
        NSError *error);
```

CoreMotion

- Register for gyroscope updates

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

  - Implement handler block of type

```
typedef void (^CMGyroHandler)(CMGyroData *gyroData,
                              NSError *error);
```

- Register for device motion updates

```
- (void)startDeviceMotionUpdatesToQueue:
            (NSOperationQueue *)queue
           withHandler:(CMDeviceMotionHandler)handler;
```

  - Implement handler block of type

```
typedef void (^CMDeviceMotionHandler)
            (CMDeviceMotion *motion, NSError *error);
```

CoreMotion

- Use same call to stop interval updates

```
void stop{Accelerometer/Gyro/DeviceMotion}Updates
```

- To use CoreMotion, add to Frameworks folder of project in Xcode
  - Frameworks->Add->Existing Frameworks . . .
- Also, import to any source files that use CoreMotion

```
#import <CoreMotion/CoreMotion.h>
```

Coordinate system