Sensors

Mobile Application Development in iOS
School of EECS
Washington State University
Instructor: Larry Holder
Low-level Sensors

- Accelerometer – Movement
- Gyroscope – Rotation and orientation
- Magnetometer – Direction
- GPS – Location
High-level Sensors

- Device orientation
- Shake motion
- Proximity (to user)
- Microphone & cameras
- Battery
- Wifi and cellular radios (but can’t access)
Aggregated Sensors

• Location services
  – Maps, regions (beacon, circular)
  – Geocoders, placemarks

• Motion services
  – CMDeviceMotion: altitude, rotation rate, acceleration
  – Pedometer, step counter
  – Activity: Stationary, walking, running, cycling, driving
Sensor Availability

• Declaring required device capabilities
  – Info.plist: Required device capabilities
  – accelerometer, gps, gyroscope, location-services, magnetometer
  – On every device since iPhone 4, iPad 2
Sensor Availability

- Declaring required device capabilities

<table>
<thead>
<tr>
<th>Key</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Information Property List</td>
<td>Dictionary</td>
<td>(15 items)</td>
</tr>
<tr>
<td>Localization native development re...</td>
<td>String</td>
<td>en</td>
</tr>
<tr>
<td>Executable file</td>
<td>String</td>
<td>$(EXECUTABLE_NAME)</td>
</tr>
<tr>
<td>Bundle identifier</td>
<td>String</td>
<td>$(PRODUCT_BUNDLE_IDENTIFIER)</td>
</tr>
<tr>
<td>InfoDictionary version</td>
<td>String</td>
<td>6.0</td>
</tr>
<tr>
<td>Bundle name</td>
<td>String</td>
<td>$(PRODUCT_NAME)</td>
</tr>
<tr>
<td>Bundle OS Type code</td>
<td>String</td>
<td>APPL</td>
</tr>
<tr>
<td>Bundle versions string, short</td>
<td>String</td>
<td>1.0</td>
</tr>
<tr>
<td>Bundle creator OS Type code</td>
<td>String</td>
<td>???</td>
</tr>
<tr>
<td>Bundle version</td>
<td>String</td>
<td>1</td>
</tr>
<tr>
<td>Application requires iPhone enviro...</td>
<td>Boolean</td>
<td>YES</td>
</tr>
<tr>
<td>Launch screen interface file base...</td>
<td>String</td>
<td>LaunchScreen</td>
</tr>
<tr>
<td>Main storyboard file base name</td>
<td>String</td>
<td>Main</td>
</tr>
<tr>
<td>Required device capabilities</td>
<td>Array</td>
<td>(3 items)</td>
</tr>
<tr>
<td>Item 0</td>
<td>String</td>
<td>accelerometer</td>
</tr>
<tr>
<td>Item 1</td>
<td>String</td>
<td>gyroscope</td>
</tr>
<tr>
<td>Item 2</td>
<td>String</td>
<td>armv7</td>
</tr>
<tr>
<td>Supported interface orientations</td>
<td>Array</td>
<td>(3 items)</td>
</tr>
<tr>
<td>Supported interface orientations (i...</td>
<td>Array</td>
<td>(4 items)</td>
</tr>
</tbody>
</table>
Motion Sensor Availability

• Checking via CMMotionManager
  – accelerometerAvailable, accelerometerActive
  – gyroAvailable, gyroActive
  – magnetometerAvailable, magnetometerActive
  – deviceMotionAvailable, deviceMotionActive
Location Sensor Availability

• Authorization via CLLocationManager
  – requestWhenInUseAuthorization
  – requestAlwaysAuthorization
    • Allows work in background
    • Capabilities ➔ Background Modes ➔ Location updates
  – authorizationStatus
Location Sensor Availability

- Availability via CLLocationManager
  - locationManagersetEnabled
  - deferredLocationUpdatesAvailable
  - significantLocationChangeMonitoringAvailable
  - headingAvailable
  - isRangingAvailable
Accessing Motion Data

• CMMotionManager
  – Add Core Motion framework
  – #import <CoreMotion/CoreMotion.h>

• Two access modes
  – Pull: App requests data when needed
  – Push: App receives data at specified interval

• Turn off sensors when not needed to save battery

• Motion sensors not available in simulator
DeviceMotion

Accelerometer: $x, y, z$ velocity

Gyroscope: $x, y, z$ rotation

DeviceMotion = $f$ (Accelerometer, Gyroscope)
DeviceMotion

• CMDeviceMotion
  – Attitude (yaw, pitch, roll)
  – rotationRate (x,y,z)
  – userAcceleration (minus gravity)
Example: DeviceMotion

```objective-c
#import "CoreMotion/CoreMotion.h"
#import "ViewController.h"

@interface ViewController ()
@property CMMotionManager * motionManager;
@end

@implementation ViewController

- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    if (_motionManager == nil) {
        _motionManager = [[CMMotionManager alloc] init];
    }
}

- (void)viewWillAppear:(BOOL)animated {
    [self startMotionUpdates];
}

- (void)viewWillDisappear:(BOOL)animated {
    [self stopMotionUpdates];
}
```

Mobile Application Development in iOS
Example: DeviceMotion

```swift
-(void)startMotionUpdates {
    // Set device motion update interval (in seconds)
    _motionManager.deviceMotionUpdateInterval = 0.1;

    [_motionManager startDeviceMotionUpdatesToQueue:[NSOperationQueue currentQueue]
        withHandler:^{CMDeviceMotion *motion, NSError *error) {
            [self motionHandler:motion];
        }];
}

-(void)stopMotionUpdates {
    [_motionManager stopDeviceMotionUpdates];
}

-(void)motionHandler:(CMDeviceMotion *)motion
{
    _yawLabel.text = [NSString stringWithFormat:@"Yaw: %f", motion.attitude.yaw];
    _pitchLabel.text = [NSString stringWithFormat:@"Pitch: %f", motion.attitude.pitch];
    _rollLabel.text = [NSString stringWithFormat:@"Roll: %f", motion.attitude.roll];
}
```
Accessing Location Data

- CLLocationManager
  - Add Core Location framework
  - #import <CoreMotion/CoreMotion.h>

- Turn off sensors when not needed to save battery

- Location data IS available in simulator
Location

- CLLocation
  - Coordinate (latitude, longitude)
  - Altitude
  - Speed
  - Course
Example: Location

```swift
#import <CoreLocation/CoreLocation.h>
#import "ViewController.h"

@interface ViewController () <CLLocationManagerDelegate>
@property (strong) CLLocationManager* locationManager;
@end

@implementation ViewController

-(void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
}

-(void)viewDidLoad:(BOOL)animated {
    [self startLocationUpdates];
}

-(void)viewDidLoad:(BOOL)animated {
    [self stopLocationUpdates];
}
```
Example: Location

```
-(void)startLocationUpdates {
    CLAuthorizationStatus clAuthStatus = [CLLocationManager authorizationStatus];
    if ([CLLocationManager locationServicesEnabled] &&
        (clAuthStatus != kCLAuthorizationStatusDenied) &&
        (clAuthStatus != kCLAuthorizationStatusRestricted))
    {
        if (_locationManager == nil) {
            _locationManager = [[CLLocationManager alloc] init];
            _locationManager.delegate = self;
            // Set movement accuracy
            _locationManager.desiredAccuracy = kCLLocationAccuracyBest; // meters
            // Set a movement threshold for new events.
            _locationManager.distanceFilter = kCLDistanceFilterNone; // meters
            [_locationManager requestWhenInUseAuthorization];
            _locationManager.allowsBackgroundLocationUpdates = YES;
        }
        [[_locationManager startUpdatingLocation];
    }
}
```

```
-(void)stopLocationUpdates {
    [_locationManager stopUpdatingLocation];
}
```
Example: Location

```objective-c
// Delegate method from the CLLocationManagerDelegate protocol.
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations
{
    CLLocation* location = [locations lastObject];
    _latitudeLabel.text = [NSString stringWithFormat:@"Latitude: %f", location.coordinate.latitude];
    _longitudeLabel.text = [NSString stringWithFormat:@"Longitude: %f", location.coordinate.longitude];
}
```
Example: Location
Placemarks

1. Create a CLGeocoder
2. Get a location
3. Make request
4. Process Placemarks array

```objective-c
@interface ViewController () <CLLocationManagerDelegate>
@property (strong) CLLocationManager* locationManager;
@property (strong) CLLocation* globalLocation;
@property (strong) CLGeocoder* globalGeoCoder;
@end

// Delegate method from the CLLocationManagerDelegate protocol.
-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations
{
    CLLocation* location = [locations lastObject];
    _globalLocation = location;
    _latitudeLabel.text = [NSString stringWithFormat:@"Latitude: \%+f", location.coordinate.latitude];
    _longitudeLabel.text = [NSString stringWithFormat:@"Longitude: \%+f", location.coordinate.longitude];
}

// Get placemark data for current location
if (_globalGeoCoder == nil)
    _globalGeoCoder = [[CLGeocoder alloc] init];

[_globalGeoCoder reverseGeocodeLocation:_globalLocation completionHandler:
^(NSArray* placemarks, NSError* error){
    if ([placemarks count] > 0)
    {
        CLPlacemark* placemark = [placemarks objectAtIndex:0];
        _regionLabel.text = [NSString stringWithFormat:@"Region = \%@", placemark.postalCode];
    }
}];
```
MapKit

• Add MapKit framework

• Add Map Kit View to View Controller
  – Set delegate
  – Select Shows: User Location
Resources

• Device Compatibility

• Motion Event Handling

• Location and Maps