



Développement d'applications mobiles iOS

Plan de formation

Séance 1 (4h)

Introduction à iOS, Objective-C / Swift et aux outils de développement

Séance 2 (4h)

Présenter l'information, communiquer, interagir

Suivre la procédure depuis son **mobile** ou sa **tablette** sur le site <http://get.udid.io/>

Enregistrement des **devices** dans le member center d'un compte développeur

Génération d'un **provisioning profile** associé à un projet (**bundle ID**) ou générique (**wildcard**)

Partage du **certificat** et de la **clé privée** permettant de **déployer** directement sur device



#mobile #sensors

Capteurs mobiles

Appareil photo

Microphone

GPS

Magnétomètre

Accéléromètre

Gyroscope

Baromètre



Disponibilité des capteurs

Capteur		iPhone 1st gen	iPhone 3G	iPhone 3GS	iPhone 4	iPhone 4S	iPhone 5	iPhone 5C	iPhone 5S	iPhone 6	iPhone 6+
Microphone		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI
GPS		NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI
Digital Compass		NON	NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI
Came ra	Front	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI
	Back	NON	NON	NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI
Accéléromètre		OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI	OUI
Gyroscope		NON	NON	NON	OUI	OUI	OUI	OUI	OUI	OUI	OUI
Touch ID		NON	NON	NON	NON	NON	NON	NON	OUI	OUI	OUI
Baromètre											



Arrêtés



En cours



Obtenir des données

Design pattern “Delegate”

“La délégation est le fait de confier une tâche à une autre personne, généralement un subordonné. La délégation ne désengage pas le délégant de sa responsabilité.” (wikipedia)



Light

Ambient Light Sensor

Principalement utilisé pour ajuster la luminosité de l'écran

Pas d'API documentée pour accéder aux données...

... mais des solutions

<http://iphonedevwiki.net/index.php/AppleSL29003>

Capteur de proximité

Utilisé par l'iPhone pour éteindre l'écran lors d'un appel

```
// Get device
UIDevice *device = [UIDevice currentDevice];

// Enable proximity monitoring
device.proximityMonitoringEnabled = YES;

// Register to notification
[[NSNotificationCenter defaultCenter] addObserver:self
                                         selector:@selector(proximityChanged:)
                                         name:@"UIDeviceProximityStateDidChangeNotification"
                                         object:device];
```

```
- (void)proximityChanged:(NSNotification *)notification
{
    // Extract device from notification
    UIDevice *device = [notification object];

    // Log information
    NSLog(@"Proximity state: %d", device.proximityState);
}
```

Capture d'images

UIImagePickerController fait parti de UIKit.framework

```
UIImagePickerController *imagePickerController = [[UIImagePickerController alloc] init];  
imagePickerController.allowsEditing = NO;  
imagePickerController.sourceType = UIImagePickerControllerSourceTypeCamera;  
imagePickerController.delegate = self;
```

Méthodes du delegate à implémenter

```
- (void)imagePickerController:(UIImagePickerController *)picker  
didFinishPickingMediaWithInfo:(NSDictionary *)info;  
  
- (void)imagePickerControllerDidCancel:(UIImagePickerController *)picker;
```

https://developer.apple.com/library/ios/documentation/UIKit/Reference/UIImagePickerController_Class/UIImagePickerController/UIImagePickerController.html

Analyse d'image

Accès au flux vidéo en temps réel

Bridges à prévoir entre l'Objective-C et le C

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

- (void)captureOutput:(AVCaptureOutput *)captureOutput
didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer
fromConnection:(AVCaptureConnection *)connection;
```



Sound

Capturer et jouer du son

AVFoundation.framework

- AVAudioRecorder AVAudioRecorderDelegate
- AVAudioPlayer & AVAudioPlayerDelegate

<http://www.appcoda.com/ios-avfoundation-framework-tutorial/>

Analyse du son

Accès au flux audio en temps réel

Bridges à prévoir entre l'Objective-C et le C

```
#import <AVFoundation/AVFoundation.h>  
  
- (void)captureOutput:(AVCaptureOutput *)captureOutput  
  didOutputSampleBuffer:(CMSampleBufferRef)sampleBuffer  
    fromConnection:(AVCaptureConnection *)connection;
```



Location

Positionnement GPS (1/3)

CoreLocation.framework

```
self.locationManager = [[CLLocationManager alloc] init];  
self.locationManager.delegate = self;
```

Lancement manuel des mises à jour

```
// Start/Stop location updates  
[_locationManager stopUpdatingLocation];  
[_locationManager startUpdatingLocation];
```

Design pattern “Delegate”

```
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray  
*)locations;
```

Positionnement GPS (2/3)

```
- (void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations
{
    // Extract last location
    CLLocation *location = [locations lastObject];

    // Extractable data
    location.coordinate.latitude (float)
    location.coordinate.longitude (float)
    location.altitude (float) en mètres
    location.horizontalAccuracy (float) en mètres
    location.verticalAccuracy (float) en mètres
    location.speed (float) vitesse en m/s
    location.course (float) angle en degrés
}
```

Le tableau “locations” contient au moins une position (plus si des appels ont été manqués)

Positionnement GPS (3/3)

```
self.locationManager.desiredAccuracy = kCLLocationAccuracyNearestTenMeters;
```

Possibilité de régler la précision

```
extern const CLLocationAccuracy kCLLocationAccuracyBestForNavigation
__OSX_AVAILABLE_STARTING(__MAC_10_7, __IPHONE_4_0);
extern const CLLocationAccuracy kCLLocationAccuracyBest;
extern const CLLocationAccuracy kCLLocationAccuracyNearestTenMeters;
extern const CLLocationAccuracy kCLLocationAccuracyHundredMeters;
extern const CLLocationAccuracy kCLLocationAccuracyKilometer;
extern const CLLocationAccuracy kCLLocationAccuracyThreeKilometers;
```

À adapter en fonction du besoin tout en pensant à la batterie

Geofencing

Déclaration de périmètres virtuels

Boundary crossing

À partir de CLLocationManagerDelegate

```
- (void)locationManager:(CLLocationManager *)manager didEnterRegion:(CLRegion *)region;  
- (void)locationManager:(CLLocationManager *)manager didExitRegion:(CLRegion *)region;
```

CoreLocation.framework

<http://www.creativebloq.com/ipad/get-started-geofencing-ios-9122867>

Significant Location Changes

SLC pour “*Significant Location Changes*”

À utiliser lorsqu’une position, même grossière est nécessaire

Bon pour préserver la batterie

*Apps can expect a notification as soon as the device moves **500 meters** or more from its previous notification. It should not expect notifications more frequently than **once every five minutes**. If the device is able to retrieve data from the network, the location manager is much more likely to deliver notifications in a timely manner.*

- (void)startMonitoringSignificantLocationChanges;
- (void)stopMonitoringSignificantLocationChanges;

Magnétomètre

Boussole indiquant le Nord

- Nord géographique (aka “True North”)
- Nord magnétique

Information fournie par CLLocationManager

CoreLocation.framework

Altitude

Altitude plus précise sur les iPhone's 6 & 6+

Information fournie par le baromètre

CMAltimeter & CMAltitudeData

CoreMotion.framework



Motion

CoreMotion

Accéléromètre

+

Magnétomètre

+

Baromètre

+

Gyroscope

=

CoreMotion.framework

Accéléromètre

Accélération linéaire du device sur 3 axes

- x
- y
- z

UIKit.framework (UIAccelerometer deprecated) ou CoreMotion.framework

```
// Setup accelerometer delegate
[UIAccelerometer sharedAccelerometer].delegate = self;
```

```
- (void)accelerometer:(UIAccelerometer *)accelerometer didAccelerate:(UIAcceleration *)acceleration
{
    // Log acceleration
    NSLog(@"-- accelerometer:didAccelerate: --");
    NSLog(@"x: %f", acceleration.x);
    NSLog(@"y: %f", acceleration.y);
    NSLog(@"z: %f", acceleration.z);
}
```

Gyroscope

Rotation du device sur 3 axes

- x
- y
- z

CoreMotion.framework

```
// Create a motion manager
CMMotionManager *motionManager = [[CMMotionManager alloc] init];
motionManager.gyroUpdateInterval = 1.0f/60.0f;
```

```
// Start gyro updates
[motionManager startGyroUpdatesToQueue:[NSOperationQueue currentQueue]
withHandler:^(CMGyroData *gyroData, NSError *error) {
    // Log gyro data
    NSLog(@"x: %f", gyroData.rotationRate.x);
    NSLog(@"y: %f", gyroData.rotationRate.y);
    NSLog(@"z: %f", gyroData.rotationRate.z);
}];
```



Réseau

Network

WiFi

Bluetooth

Bonjour



<https://www.youtube.com/watch?v=XYvBQI6zjaQ&>

iBeacons

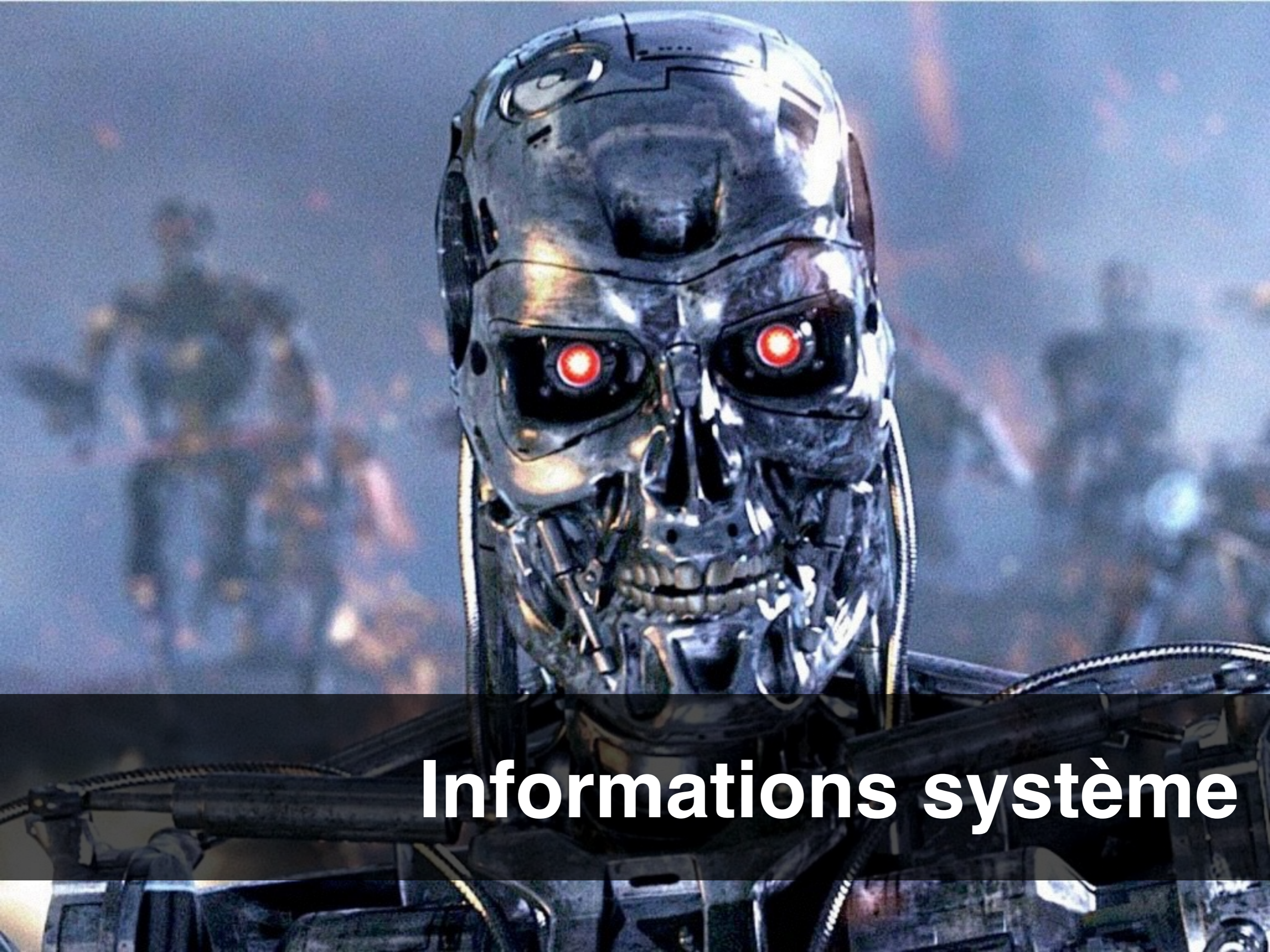
Bluetooth Low Energy (BLE)

Indoor Proximity System :

- immediate (à quelques centimètres)
- near (à quelques mètres)
- far (à plus de 10 mètres, jusque \pm 70m)

Exemples d'utilisation :

- <http://estimote.com/>
- <http://www.lesuperherosduparking.com/>



Informations système

Batterie

Récupérer l'objet UIDevice courant pour extraire des informations

```
// Get device
UIDevice *device = [UIDevice currentDevice];
```

Activer le suivi de la batterie

```
// Enable battery monitoring
device.batteryMonitoringEnabled = YES;
```

Design pattern "Observer"

```
// Register to notifications
[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(batteryStateChanged:)
 name:@"UIDeviceBatteryStateDidChangeNotification"
 object:device];

[[NSNotificationCenter defaultCenter] addObserver:self
 selector:@selector(batteryLevelChanged:)
 name:@"UIDeviceBatteryLevelDidChangeNotification"
 object:device];
```

Évènements système

De nombreux évènements sont observables via les **Darwin Notifications**

```
-(void)registerforDeviceLockNotif
{
    //Screen lock notifications
    CFNotificationCenterAddObserver(CFNotificationCenterGetDarwinNotifyCenter(), //center
                                   NULL, // observer
                                   displayStatusChanged, // callback
                                   CFSTR("com.apple.springboard.lockcomplete"), // event name
                                   NULL, // object
                                   CFNotificationSuspensionBehaviorDeliverImmediately);

    CFNotificationCenterAddObserver(CFNotificationCenterGetDarwinNotifyCenter(), //center
                                   NULL, // observer
                                   displayStatusChanged, // callback
                                   CFSTR("com.apple.springboard.lockstate"), // event name
                                   NULL, // object
                                   CFNotificationSuspensionBehaviorDeliverImmediately);
}
```

Exemple avec le lock/unlock du téléphone

Touch ID

Reconnaissance d'empreintes
digitales

Remplace le mot de passe pour:

- débloquer le téléphone
- payer sur l'App Store

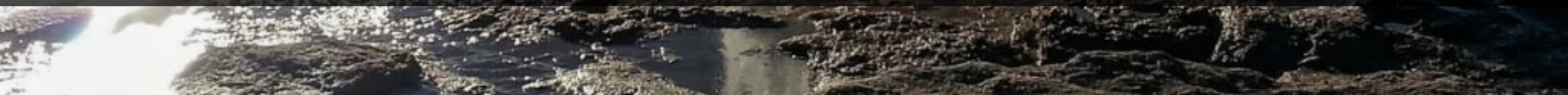
Uniquement depuis les iPhone 5S

**API publique disponible avec
iOS 8**





Divers



Sources

<https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/LocationAwarenessPG/CoreLocation/CoreLocation.html>

<https://developer.apple.com/library/ios/documentation/iPhone/Conceptual/iPhoneOSProgrammingGuide/BackgroundExecution/BackgroundExecution.html>

<http://www.slideshare.net/thomasfankhauser/ios-sensors-15579340>



Questions ?