



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

Kinan Arnaout

Ingénieur R&D Mobile chez Intellicore



kinan.arnaout@intellicore.net

Intellicore

www.intellicore.net

www.intellicore.tv

The image displays a vertical timeline of Intellicore's work from 2011 to 2014. On the left, four rows of tablet screens show various user interfaces, including sports data, library services, and product catalogs. On the right, a white panel lists client logos: FIBA Basketball World Cup Spain 2014, EVE, ZON, numericable, COX, QuickPlay MEDIA, OLGIN DUCASSE, TELUS, CISCO, NDS, and LIBERTY GLOBAL.

2014

2013

2012

2011

FIBA Basketball World Cup Spain 2014

EVE

ZON

numericable

COX

QuickPlay MEDIA

OLGIN DUCASSE

TELUS

CISCO

NDS

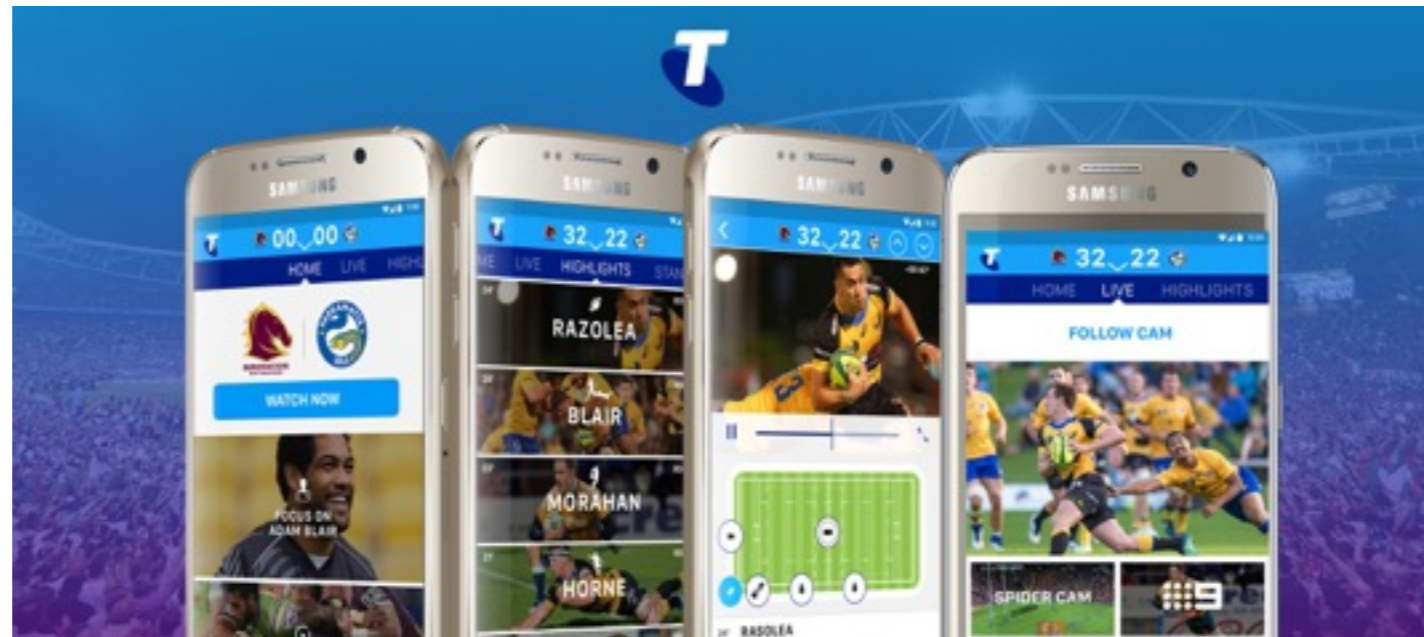
LIBERTY GLOBAL

Intellicore

Playrz



Applications stadium



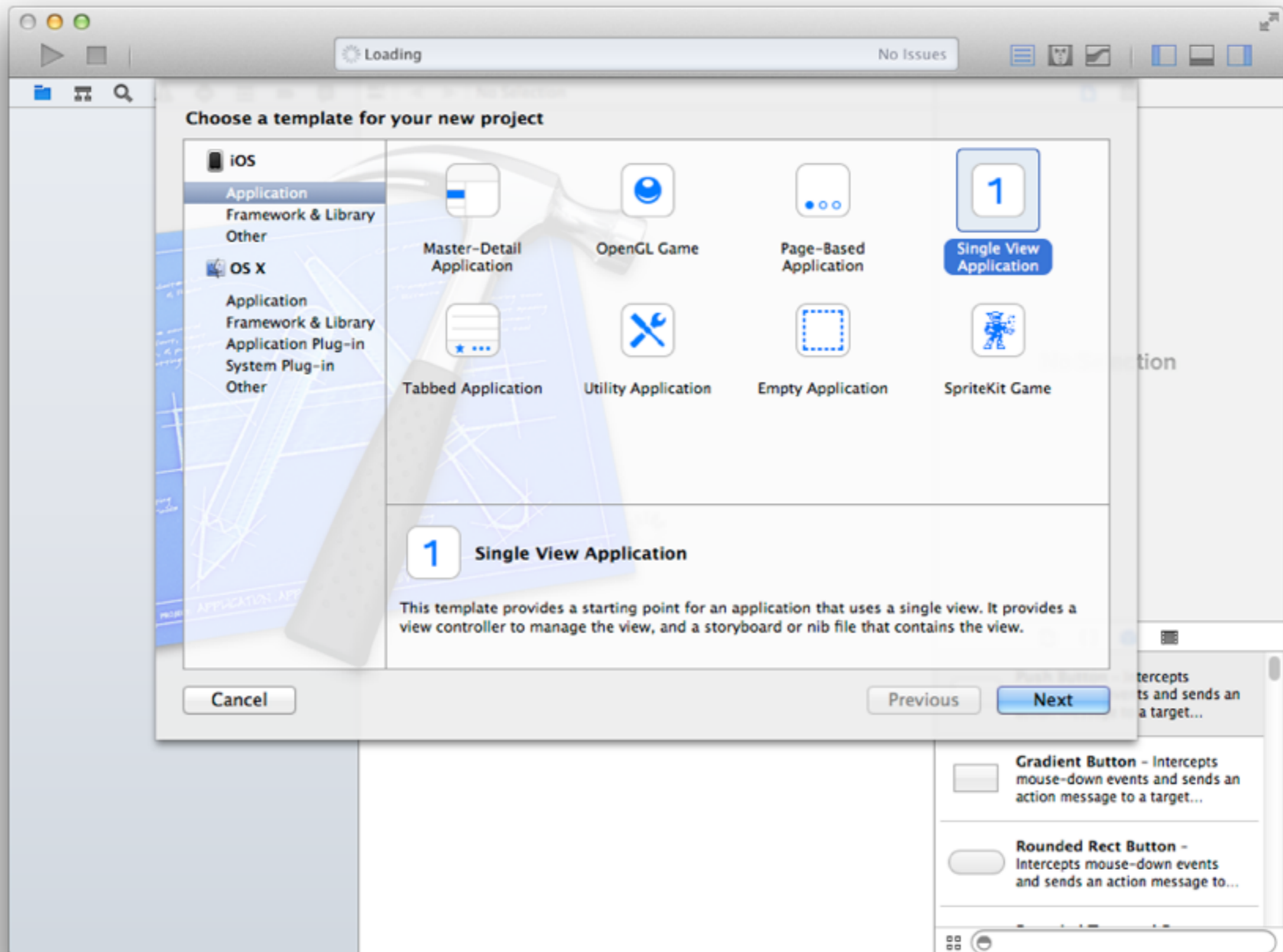


Contraintes mobiles?

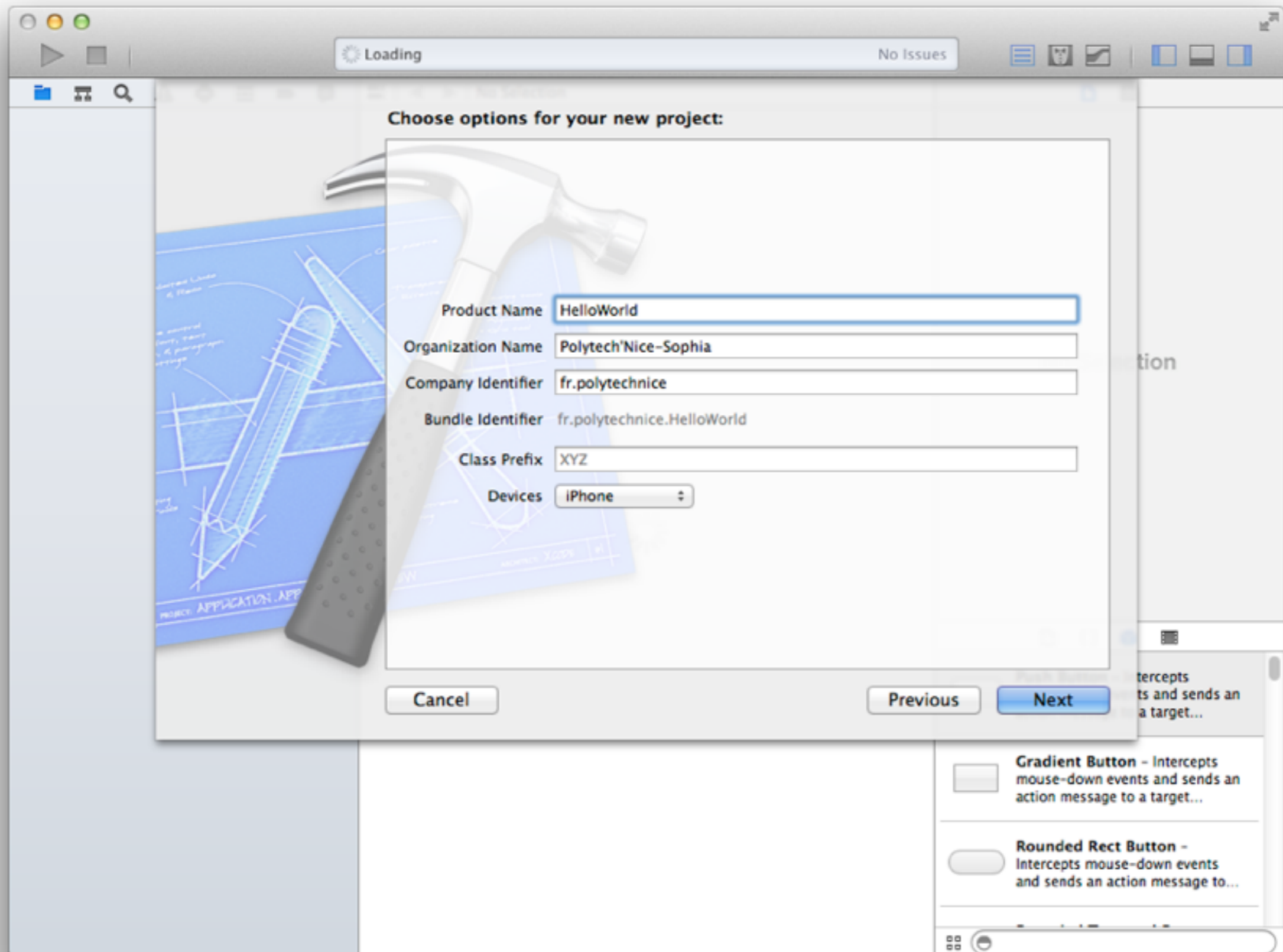


Rappel : premier projet

Création à partir d'un template



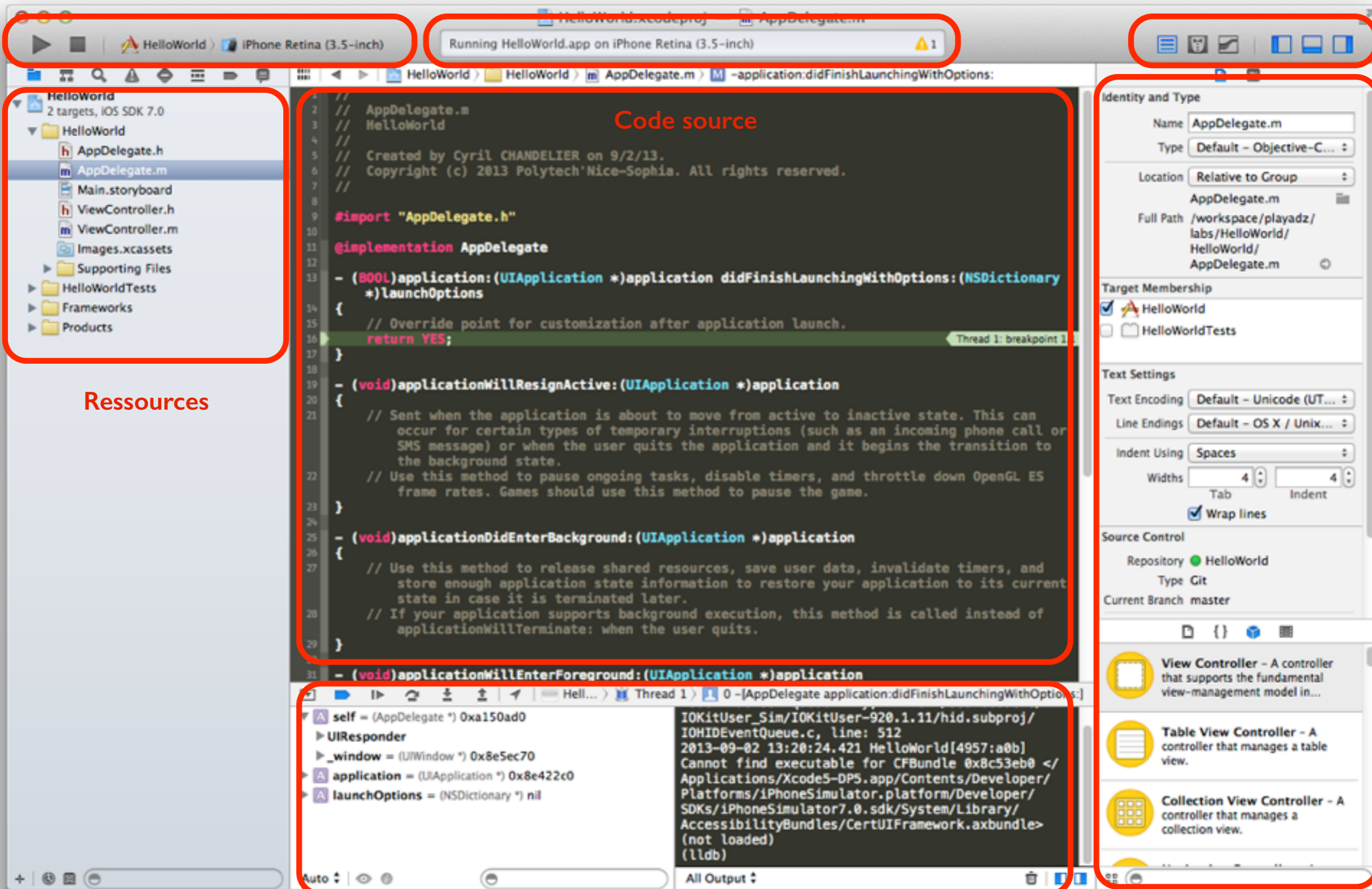
Informations générales



Lancer le projet / Choix de la cible

Indicateur d'activité

Configuration de la fenêtre

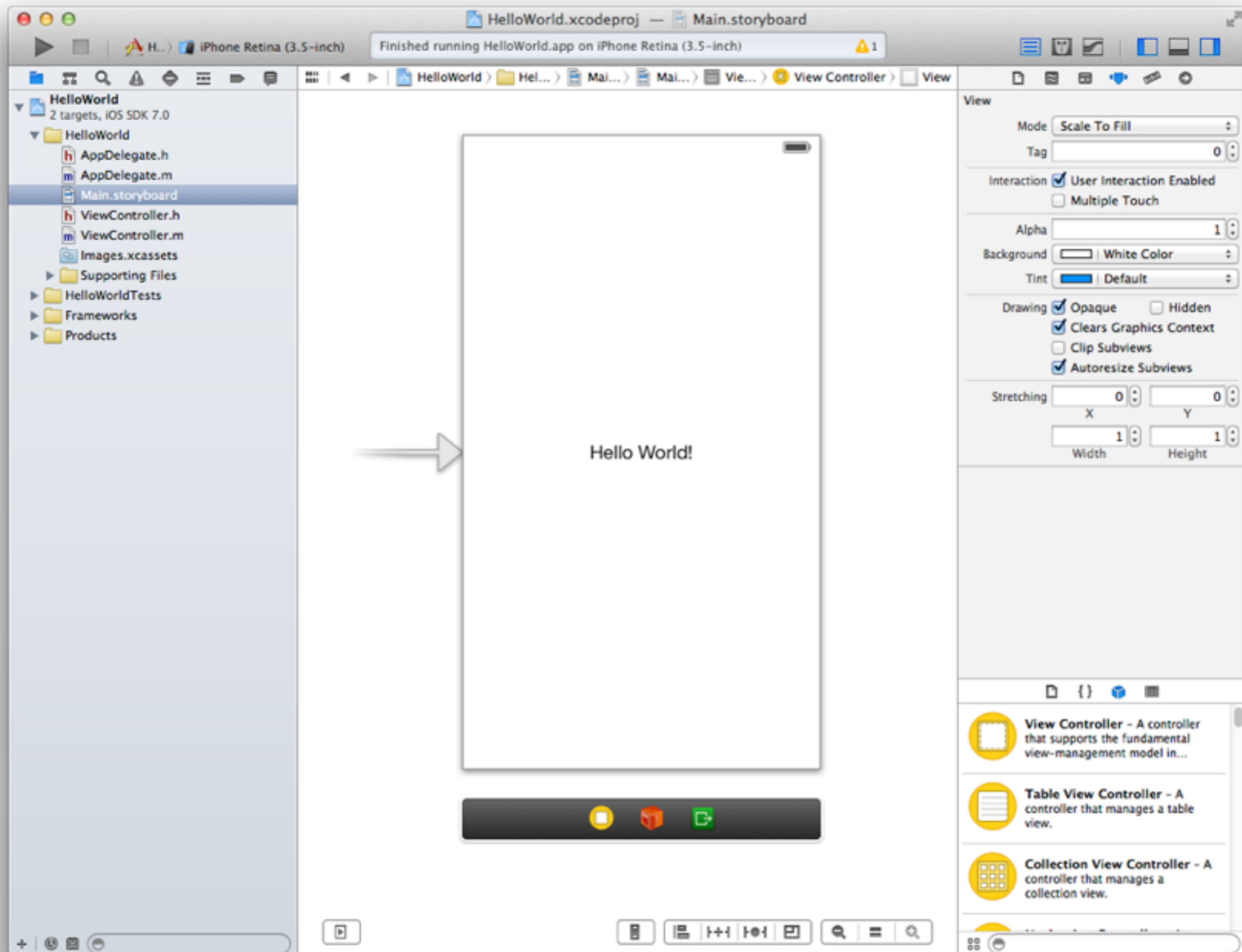


Ressources

Code source

Debugger

Inspecteur



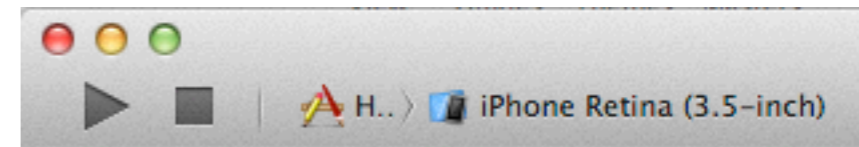
Lancement de l'application

Build and Run

Compilation du projet et des dépendances

Ouverture du simulateur

Premier projet lancé!





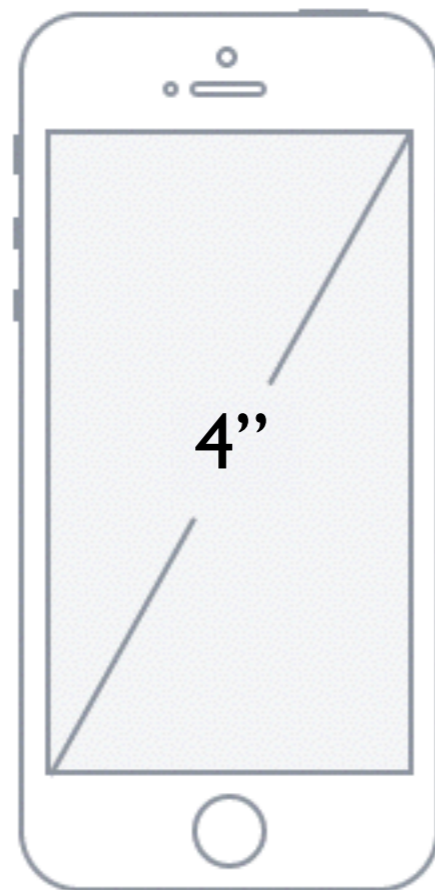
User Interface

Écrans

**iPhone
4 / 4S**



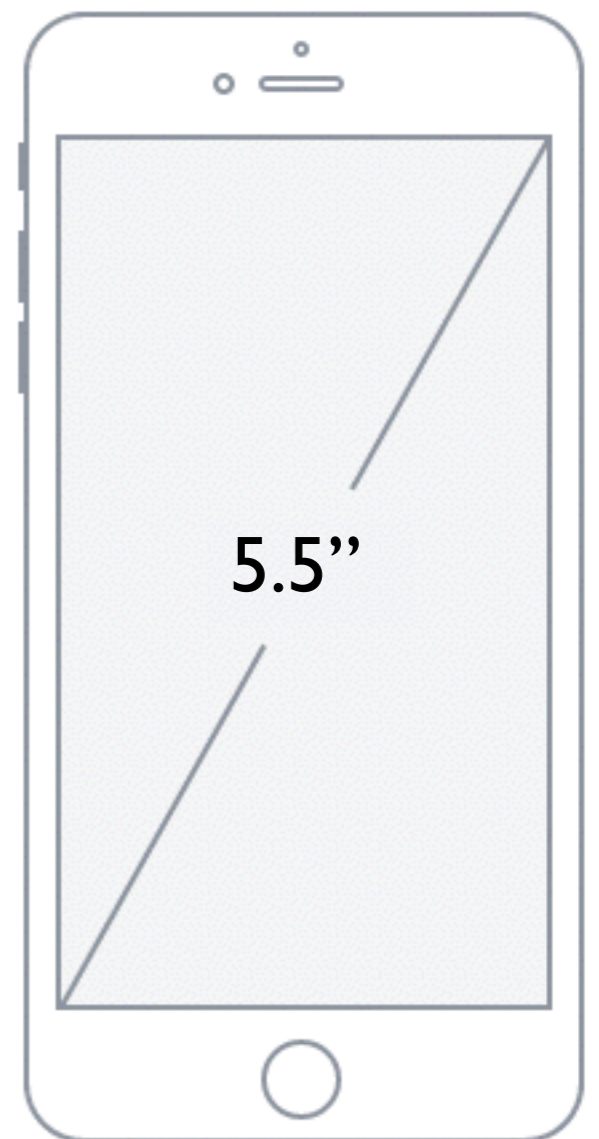
**iPhone
5 / 5C / 5S**



**iPhone
6**

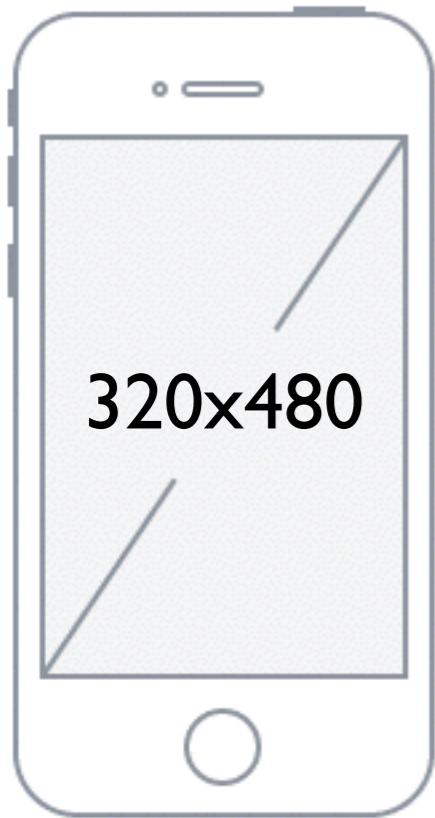


**iPhone
6 Plus**

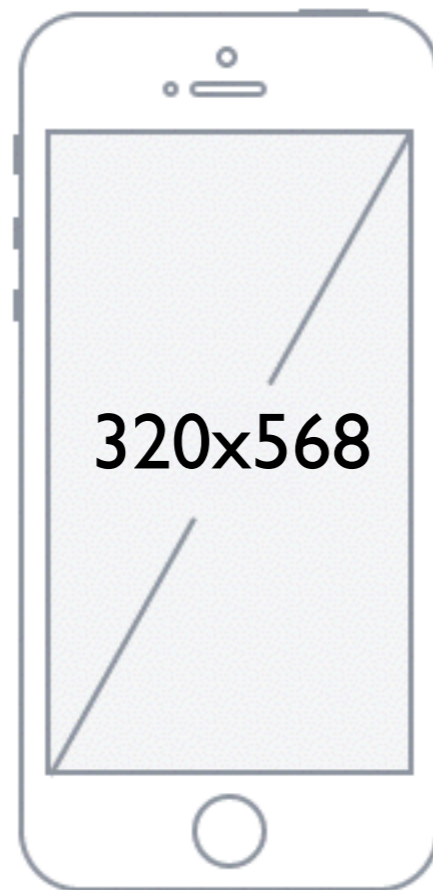


Résolution en points

**iPhone
4 / 4S**



**iPhone
5 / 5C / 5S**



**iPhone
6**



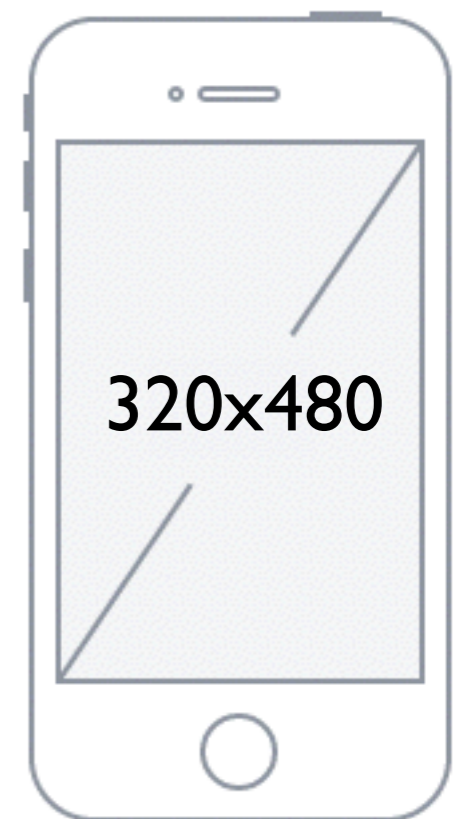
**iPhone
6 Plus**



Résolution en pixels (1/2)

**iPhone
2G / 3G / 3GS**

Sur les premiers iPhone, la résolution en points correspondait aux pixels à l'écran



rendu en 1x

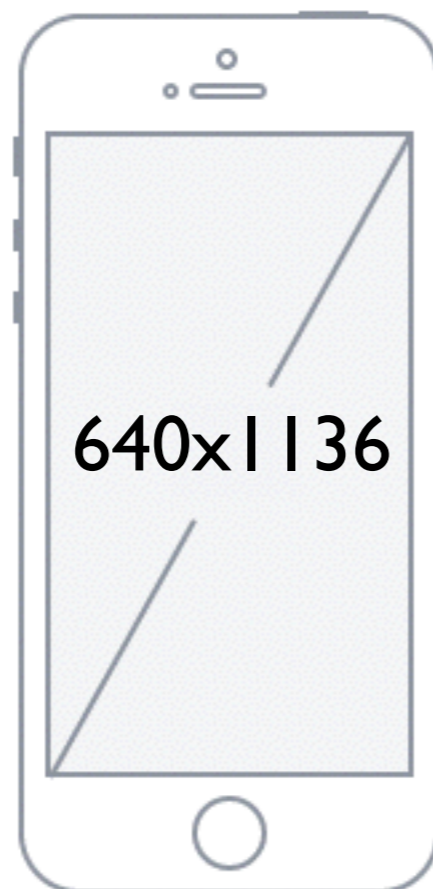
Résolution en pixels (2/2)

iPhone
4 / 4S



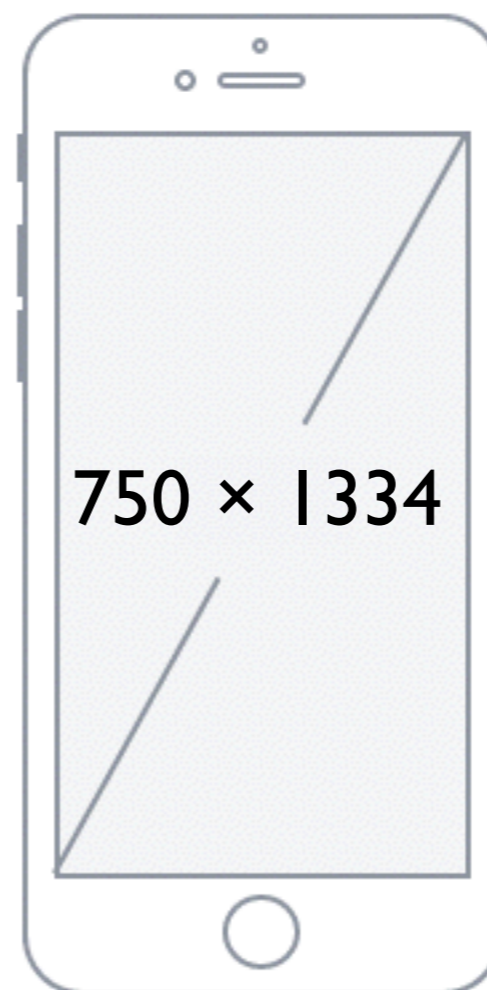
rendu en 2x

iPhone
5 / 5C / 5S



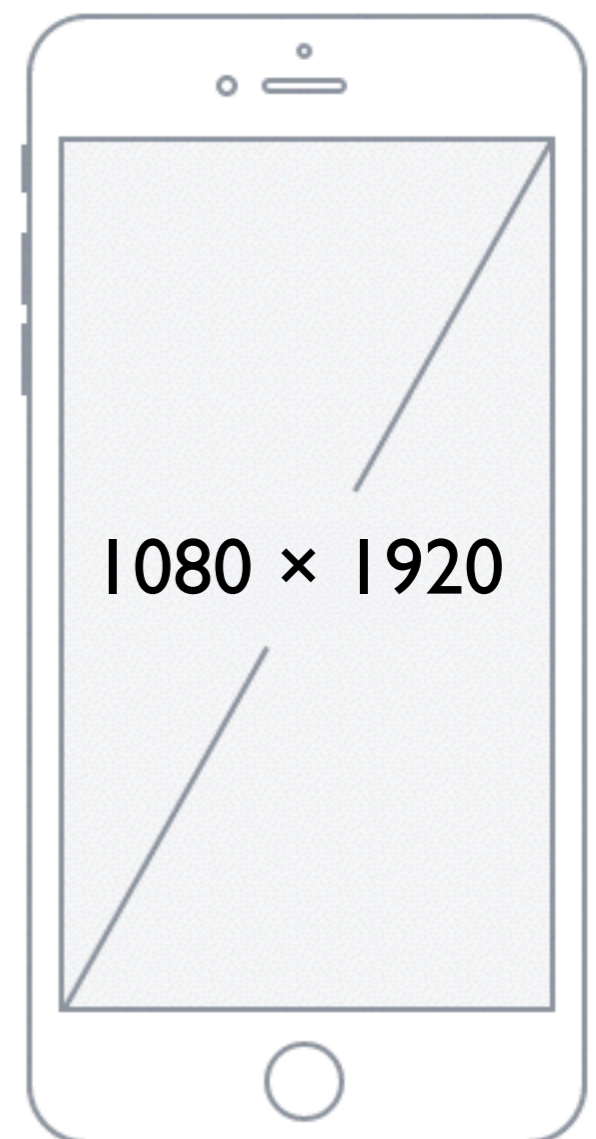
rendu en 2x

iPhone
6



rendu en 2x

iPhone
6 Plus



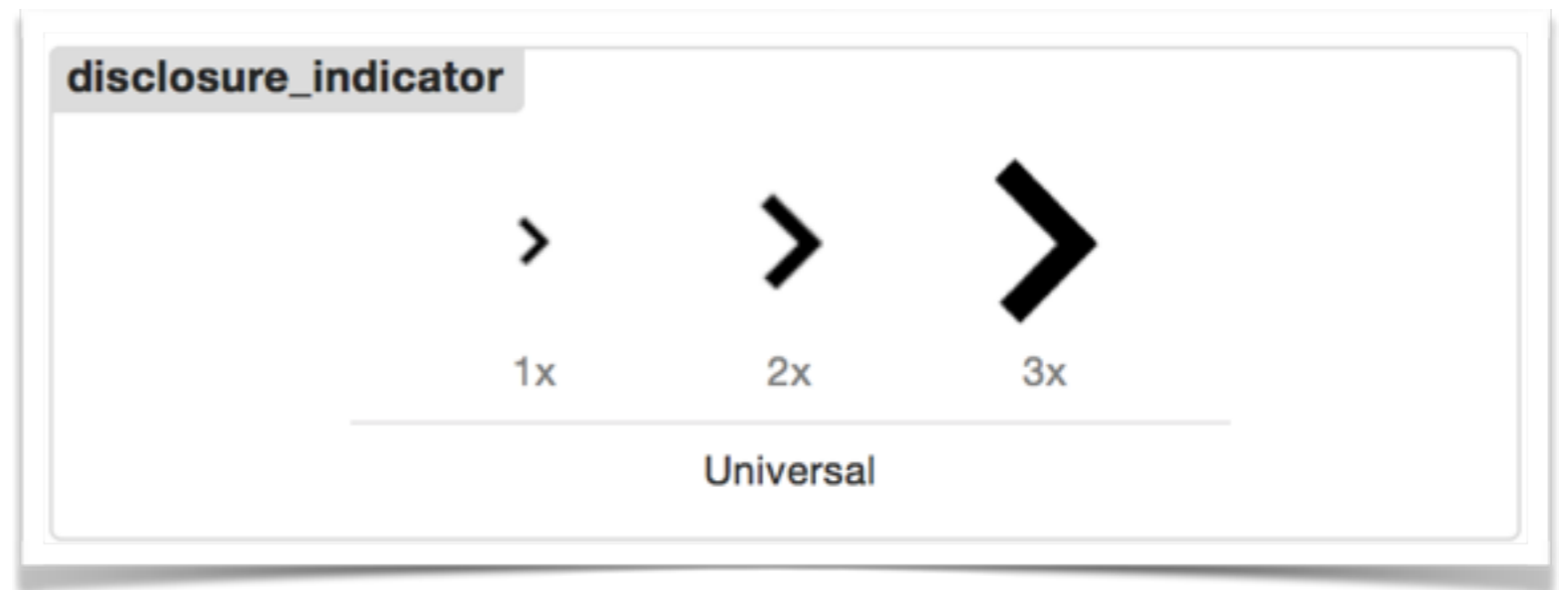
rendu en 3x

Gestion des images

Les images sont gérées soit “en vrac”, soit par un asset catalog

Le projet doit connaître au moins l’une des trois versions :

- standard
- @2x
- @3x



```
[UIImage imageNamed:@"cell_background_texture_gray"]
```


Applications universelles

Un projet partagé entre une application iPhone et une application iPad

Les fichiers de ressources (.xib, .storyboard, images) peuvent utiliser les modifieurs ~ipad ou ~iphone

Il est fortement recommandé de prévoir une expérience utilisateur différente sur tablettes de celle prévue pour mobile

Fichier d'interface (généralement lié à un UIViewController ou une UIView)

WYSIWYG

Ajouts de composants en glisser/déposer

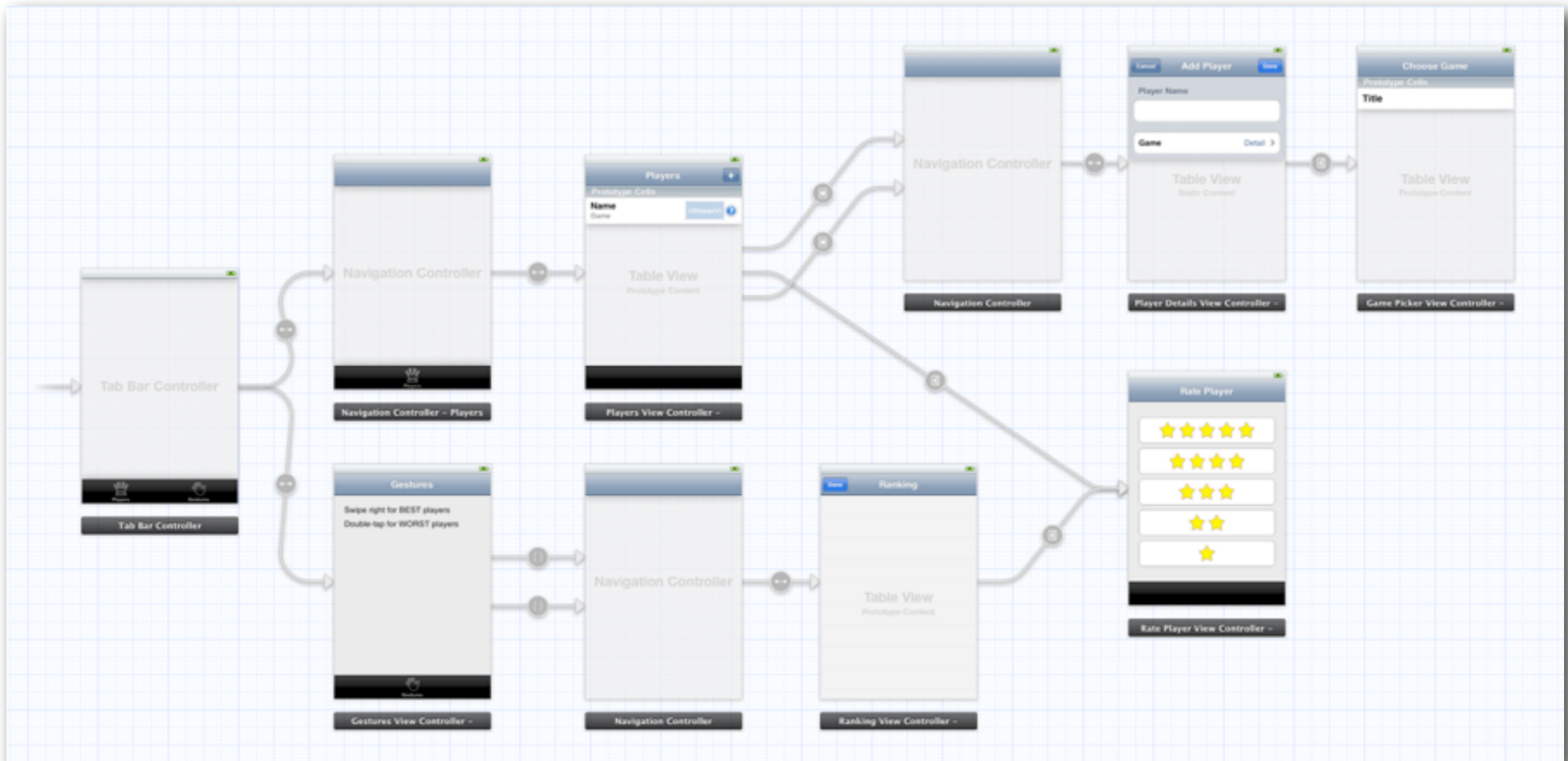
Édition des paramètres via l'inspecteur de propriétés

Storyboard

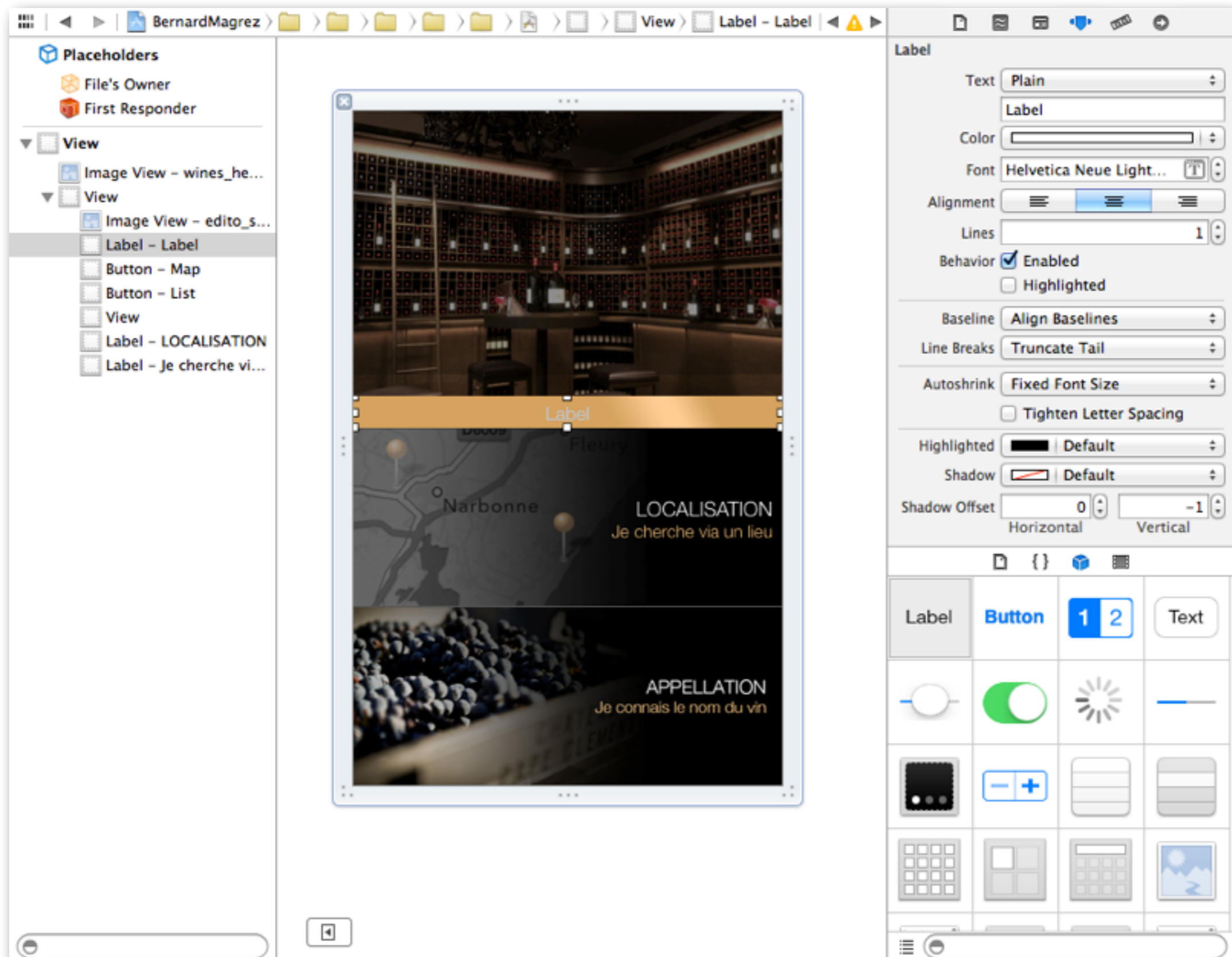
Vision “workflow”

Tout en un (contrairement aux .xib)

Opérations simplifiées (transitions, cellules personnalisées)

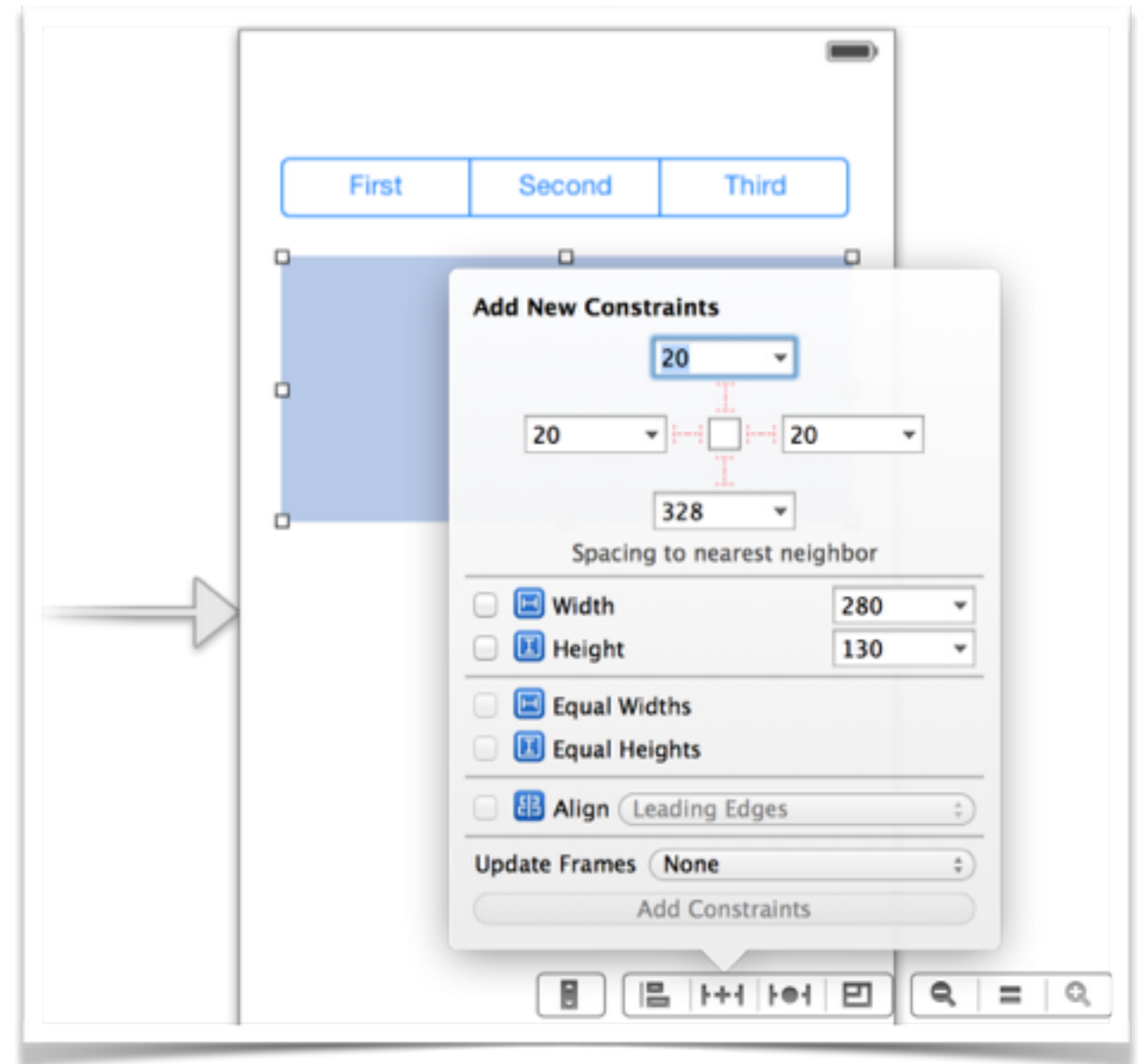


Interface Builder



Auto Layout

*Auto Layout is a system that lets you lay out your app's user interface by creating a **mathematical description of the relationships between the elements**. You define these **relationships** in terms of **constraints** either **on individual elements**, or **between sets of elements**. Using Auto Layout, you can create a dynamic and versatile interface that responds appropriately to changes in screen size, device orientation, and localization.*

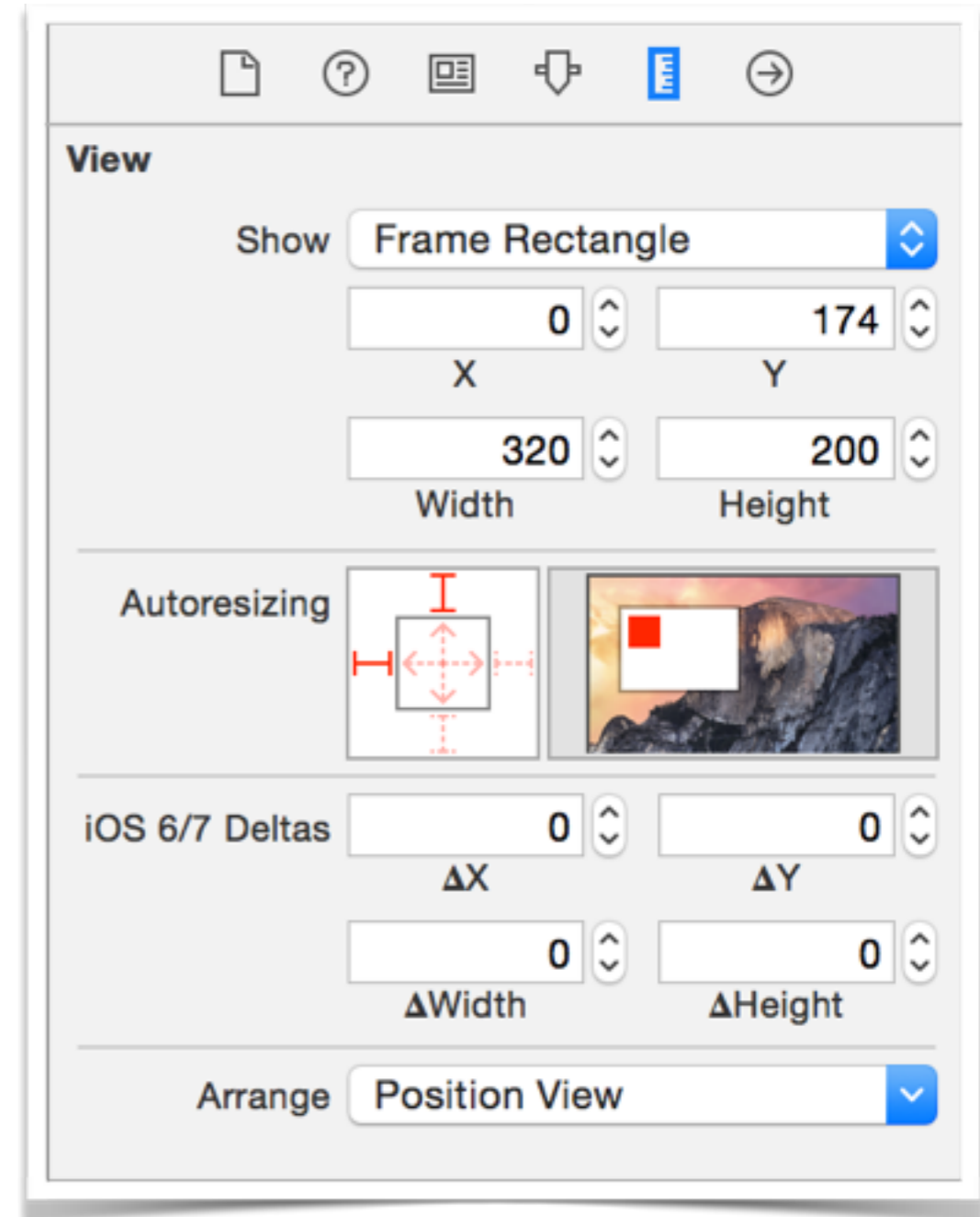


Alternative à Auto Layout

Possibilité de désactiver Auto Layout au besoin, dans une vue, un xib ou le storyboard complet

Les ancres ou *UIViewAutoresizingMask's* s'appliquent

Largeur, hauteur, flexibilité par rapport à la superview



Orientation

Deux orientations à gérer :

- landscape
- portrait

```
- (BOOL)shouldAutorotate
{
    return YES;
}

- (NSUInteger)supportedInterfaceOrientations
{
    return (UIInterfaceOrientationMaskPortrait |
    UIInterfaceOrientationMaskLandscape);
}

- (UIInterfaceOrientation)preferredInterfaceOrientationForPresentation
{
    return UIInterfaceOrientationPortrait;
}
```

Animations

UIKit permet simplement l'animation de vues en :

- manipulant la frame de la vue (largeur, hauteur, origine en x et y, centre)
- appliquant des transformations 2D
- appliquant des transformations 3D

```
[UIView animateWithDuration:1.0f
                        delay:1.0f
                        options:UIViewAnimationOptionBeginFromCurrentState
                        animations:^(
                            // Animations block
                        )
                        completion:^(BOOL finished) {
                            // Completion block
                        }];
```

Interactions avec l'UI (1/2)

Dans une classe liée à un fichier d'interface (généralement un UIViewController ou une UIView), on définit:

- des outlets (IBOutlet)
- des actions (IBAction)

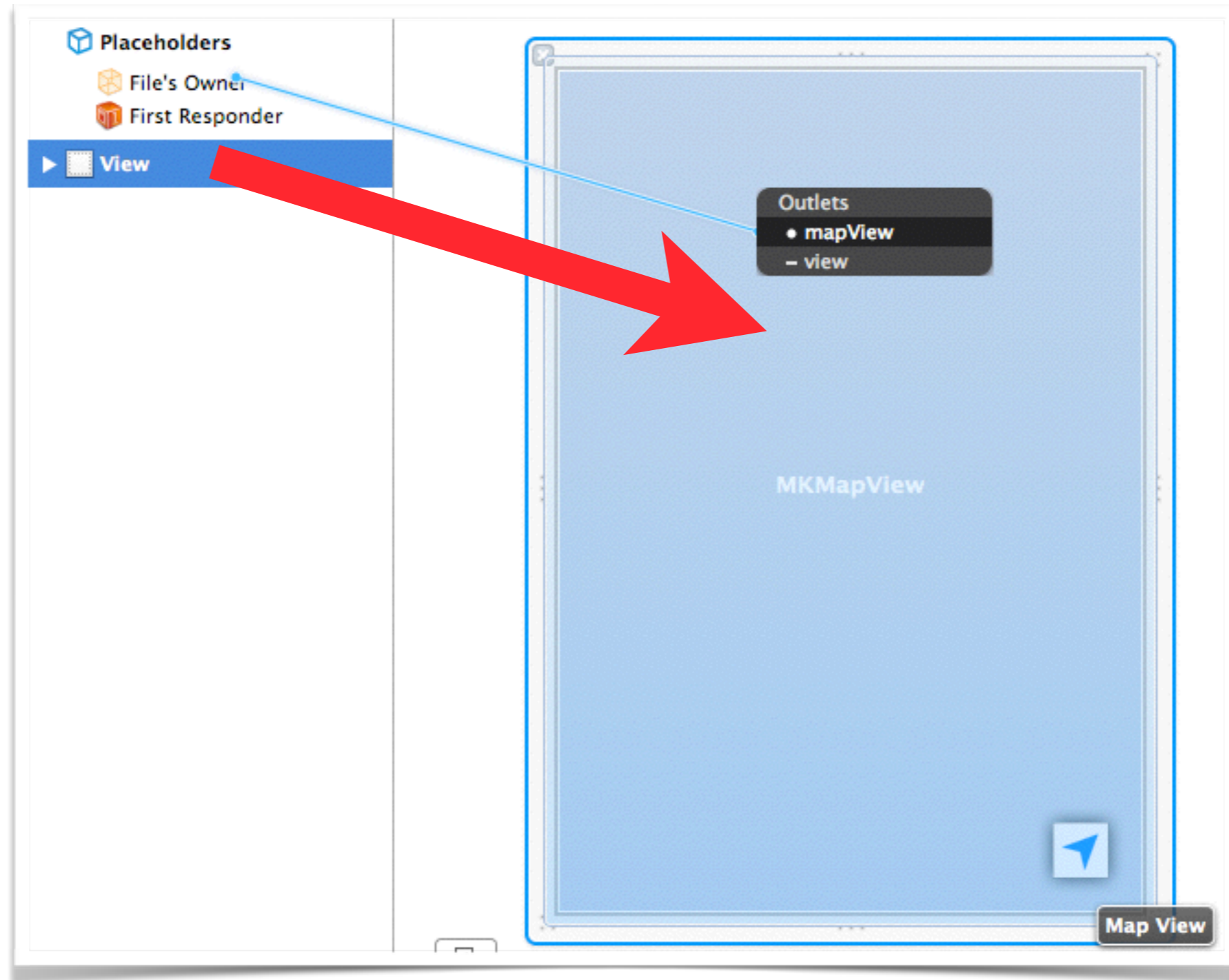
```
@interface MapViewController ()  
  
- (IBAction)focusOnUserLocation;  
  
// Outlets  
@property (nonatomic, weak) IBOutlet MKMapView *mapView;  
  
@end
```


Interactions avec l'UI (2/2)

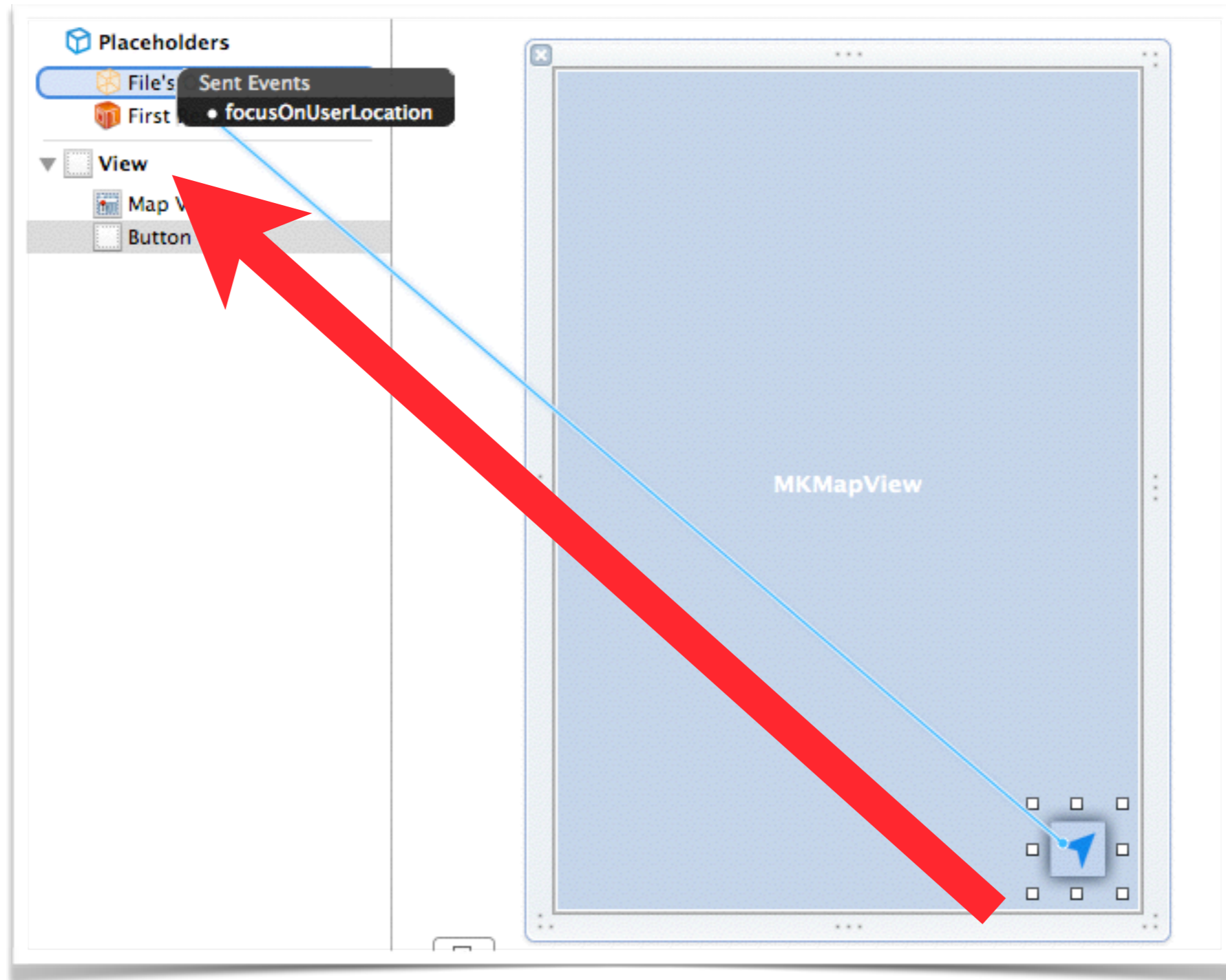
Touche **ctrl** enfoncée pour lier les éléments

- *File's owner* vers les composants d'UI (outlets)
- composants d'UI vers *File's owner* (actions ou outlets pour les delegate/datasource par exemple)

Interactions avec l'UI (2/2)



Interactions avec l'UI (2/2)



Segue

Dans un storyboard, le passage d'une vue à une autre est appelé un segue

On peut attribuer à chaque segue un identifiant depuis le storyboard qui pourra être retrouvé lorsque le segue est déclenché

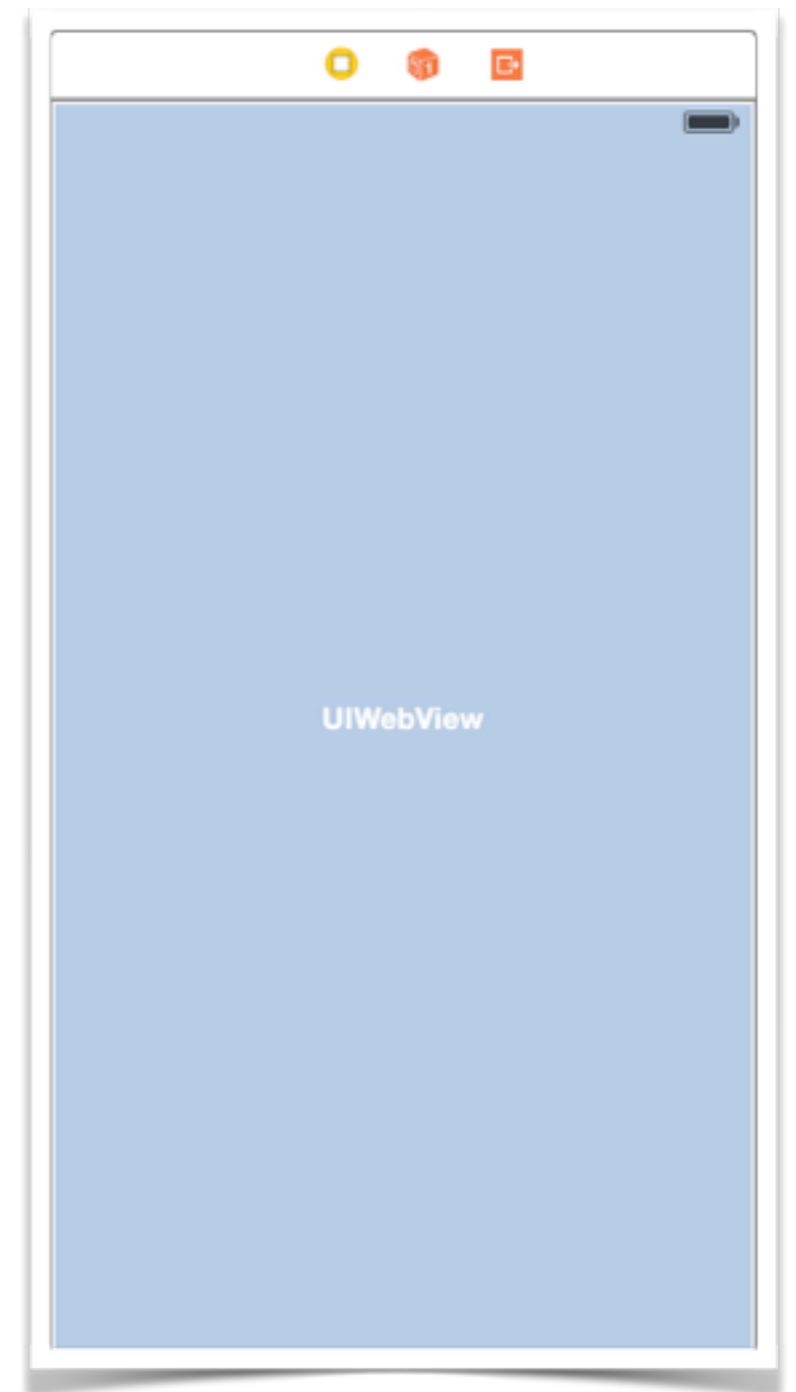
```
- (void)prepareForSegue:(UIStoryboardSegue *)segue sender:(id)sender
{
    if ([segue.identifier isEqualToString:@"pushDetail"])
    {
        id destination = [segue destinationViewController];
        // Configure destination here
    }
}
```

En dernier recours

*You use the **UIWebView** class to embed web content in your application*

Composant à utiliser avec parcimonie

Permet d'afficher une page distante ou du contenu HTML, JavaScript, etc.



UIScrollView: conteneur scrollable

UITableView: liste verticale de cellules

UICollectionView: collection de cellules avec layout

UIButton: boutons cliquables

UILabel & UITextView: affichage de textes, éditables ou non

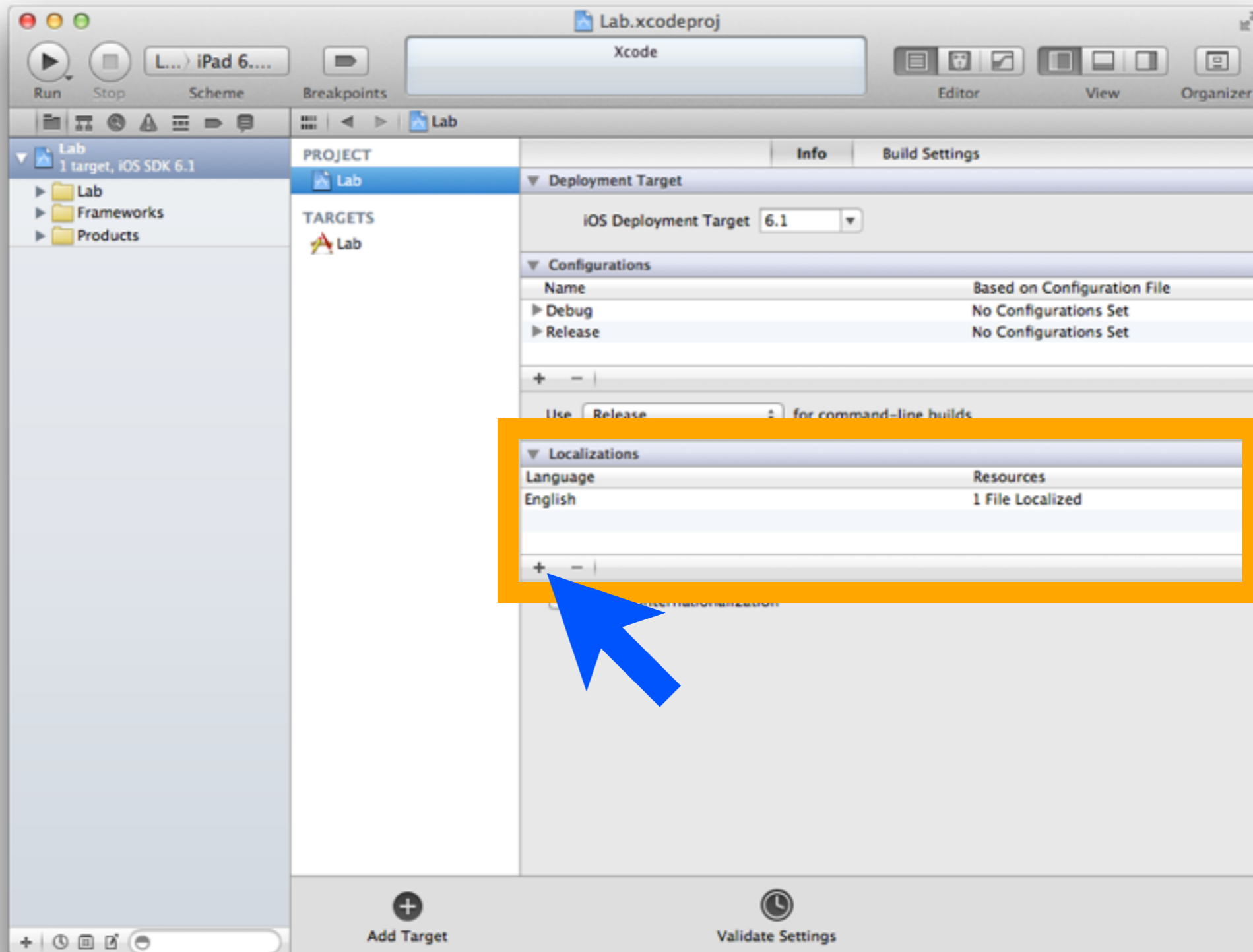
UIImageView: conteneur pour une images

UISwitch: bouton on/off

UIWebView: affichage d'une page web

https://developer.apple.com/library/ios/documentation/uikit/reference/UIKit_Framework/index.html

i18n (1/2)



Systeme clé/valeur dans le fichier Localizable.strings

```
// Global  
"Globals.ok" = "OK";  
"Globals.ko" = "KO";
```

Utilisation de la fonction native

```
label.text = NSLocalizedString(@"global.ok", @"OK");
```

Mise en place d'une macro (pour gagner du temps)

```
#define LOCALIZED_TEXT(key) NSLocalizedString(key, nil)
```

Configuration des labels et boutons dans le code

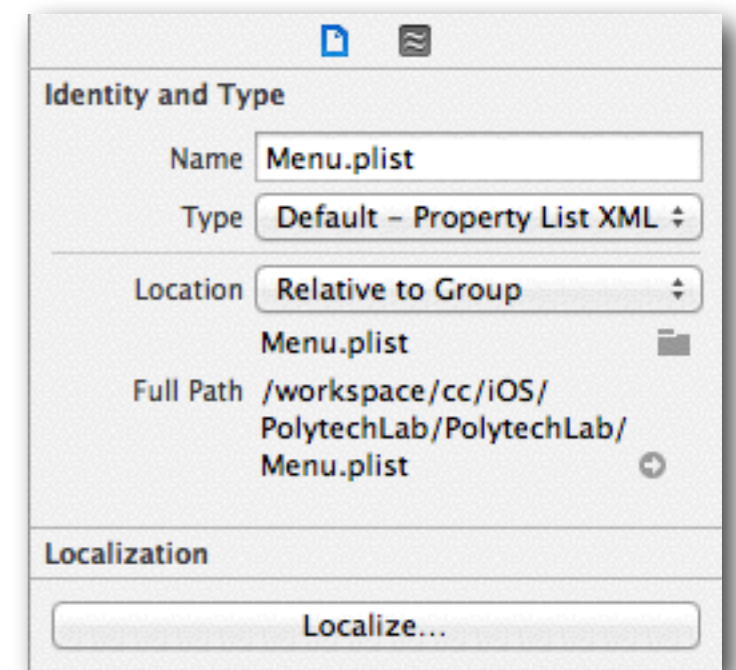
```
label.text = LOCALIZED_STRING(@"global.ok");  
[_backButton setTitle:LOCALIZED_STRING(@"global.ko") forState:UIControlStateNormal];
```

Internationaliser une ressource

Toute ressource est internationalisable: le système trouve le fichier adéquat en fonction de la locale du device

Ressources ?

- Fichiers d'interfaces (.xib ou .storyboard)
- Images (splashscreen, icônes ou autre)
- Vidéos
- Sons
- Fichiers PDF
- Fichiers de configurations (PLIST ou autre)
- etc.



Custom Font

Ajout d'une police au projet dans le fichier Info.plist

▼ Fonts provided by application	⌵ ⌴ ⌵	Array	⌵ (3 items)
Item 0		String	BellGothicStd-Black.otf
Item 1		String	BellGothicStd-Bold.otf
Item 2		String	BellGothicStd-Light.otf

Mise en place d'une macro dans le .pch

```
#define BOLD_FONT_OF_SIZE(s) [UIFont fontWithName:@"BellGothicStd-Bold" size:s]
```

Configuration des labels et boutons dans le code

```
label.font = BOLD_FONT_OF_SIZE(15.0f);  
button.titleLabel.font = BOLD_FONT_OF_SIZE(15.0f);
```

A large, bright white cumulus cloud with a soft, rounded top, set against a clear, deep blue sky. The cloud is the central focus of the image, with its base resting on a dark grey horizontal band.

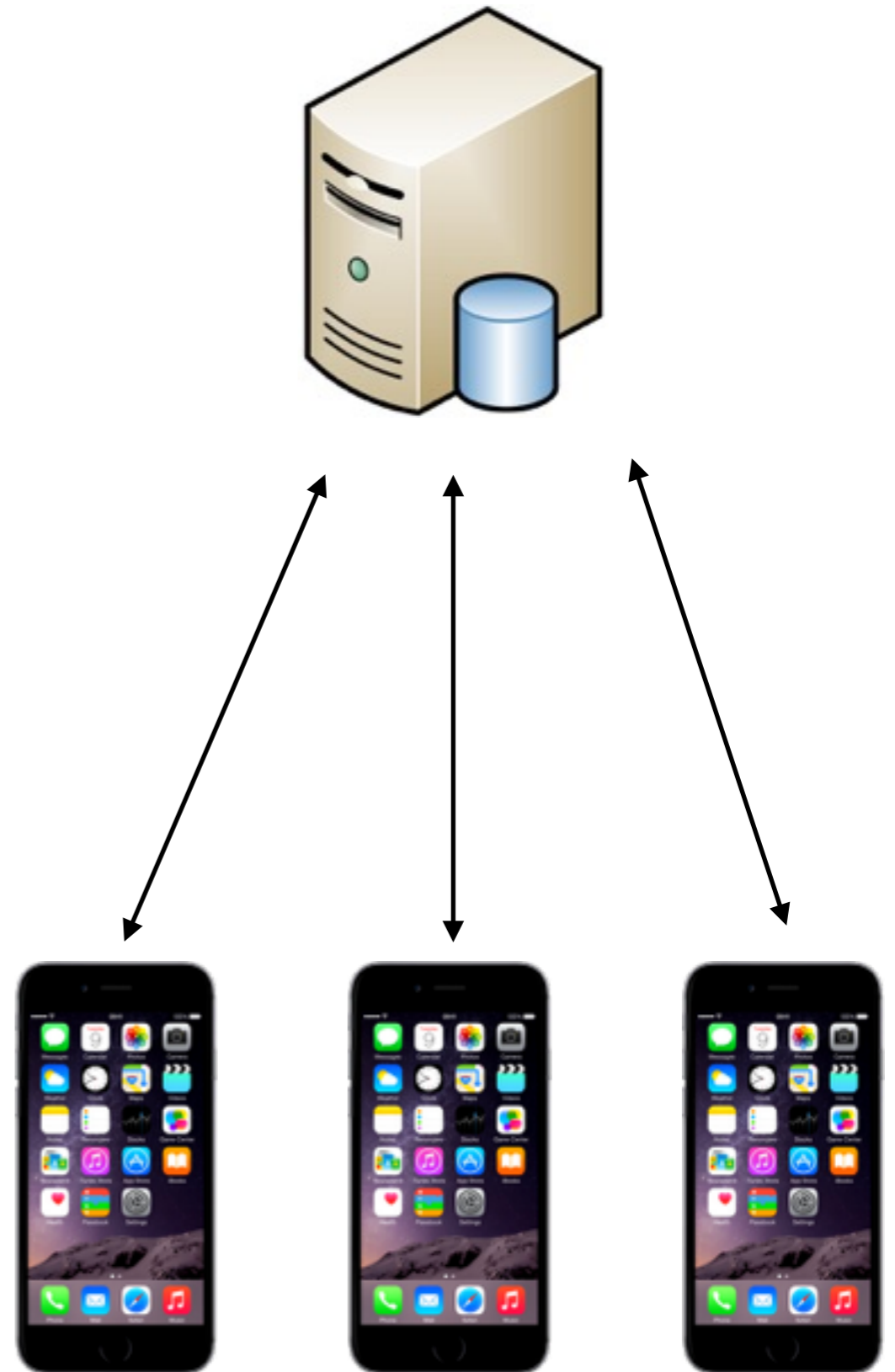
Communiquer

Modèles de communication client /

Envoi / réception de données
via socket

Envoi / réception de données
via web services

Réception de données via
notifications push



Sockets

Connexion réseau directe entre deux points (deux devices, un device et un serveur, etc.)

À la base de tout, mais bas niveau

Intéressant pour le temps réel

<https://developer.apple.com/library/ios/documentation/NetworkingInternet/Conceptual/NetworkingTopics/Articles/UsingSocketsandSocketStreams.html>

Web Sockets

Conçus pour être implémentée par des browsers Web, des portages ont été mis en place pour iOS pour communiquer sur cette technologies standardisée

<https://github.com/square/SocketRocket>

Web Services

API REST / SOAP ou simple contenu sur page distante

Privilégier le format JSON

Compresser au maximum les échanges client / serveur (gzip)

De l'intérêt d'un "backend"

Le backend centralise les données de l'application, pour tout ces utilisateurs

Sert généralement ces données aux clients via une API (des web services) à l'envoi à la réception

L'API rend l'accès aux données universel, intéressant pour le mobile mais aussi pour une éventuelle web app...

AFNetworking

Librairie tierce spécialisée dans l'envoi de requêtes réseau

Wrapper des API natives et complexes de réseau

Envoi de requête synchrones ou asynchrones

Développée par *Matt Thompson* et portée par la communauté

<https://github.com/AFNetworking/AFNetworking>

Intégration via CocoaPods

Dans un fichier Podfile, on ajoute les lignes suivantes

```
platform :ios, '7.0'  
pod 'AFNetworking', '~> 2.0'
```

On exécute ensuite la commande suivante dans son terminal

```
pod install
```

Les dépendances sont téléchargées et configurées avec le projet automatiquement par CocoaPods

Il faut alors utiliser le .xcworkspace au lieu du .xcodeproj (sous peine de fatal error à la compilation)

Utilisation

```
AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager
manager];
[manager GET:@"http://example.com/resources.json" parameters:nil
success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"JSON: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];
```

```
AFHTTPRequestOperationManager *manager = [AFHTTPRequestOperationManager
manager];
NSDictionary *parameters = @{@"foo": @"bar"};
[manager POST:@"http://example.com/resources.json" parameters:parameters
success:^(AFHTTPRequestOperation *operation, id responseObject) {
    NSLog(@"JSON: %@", responseObject);
} failure:^(AFHTTPRequestOperation *operation, NSError *error) {
    NSLog(@"Error: %@", error);
}];
```

Attention au threading

Attention au traitement des données dans le main thread

Les appels réseaux sont le plus souvent fait (AFNetworking) dans un thread background, mais les callbacks sont généralement redirigés sur le thread principal

Le traitement des données, notamment lorsqu'écriture des données en base (et donc sur disque) est très couteux et peut causer des ralentissements / freeze de l'interface utilisateur

Apple Push Notification Service

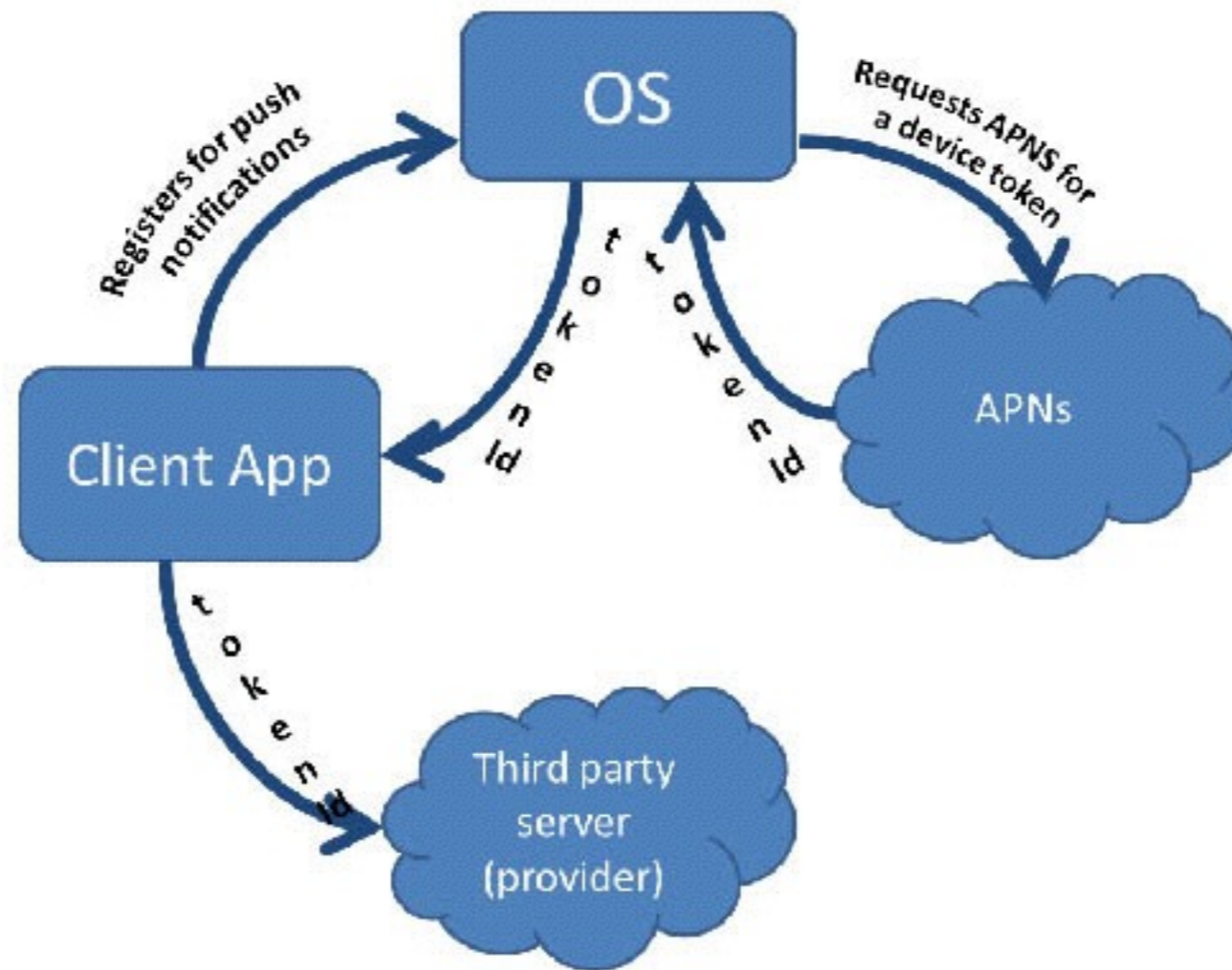
Envoi d'alertes visibles par l'utilisateur sous trois formes:

- message texte
- son
- badge sur l'application

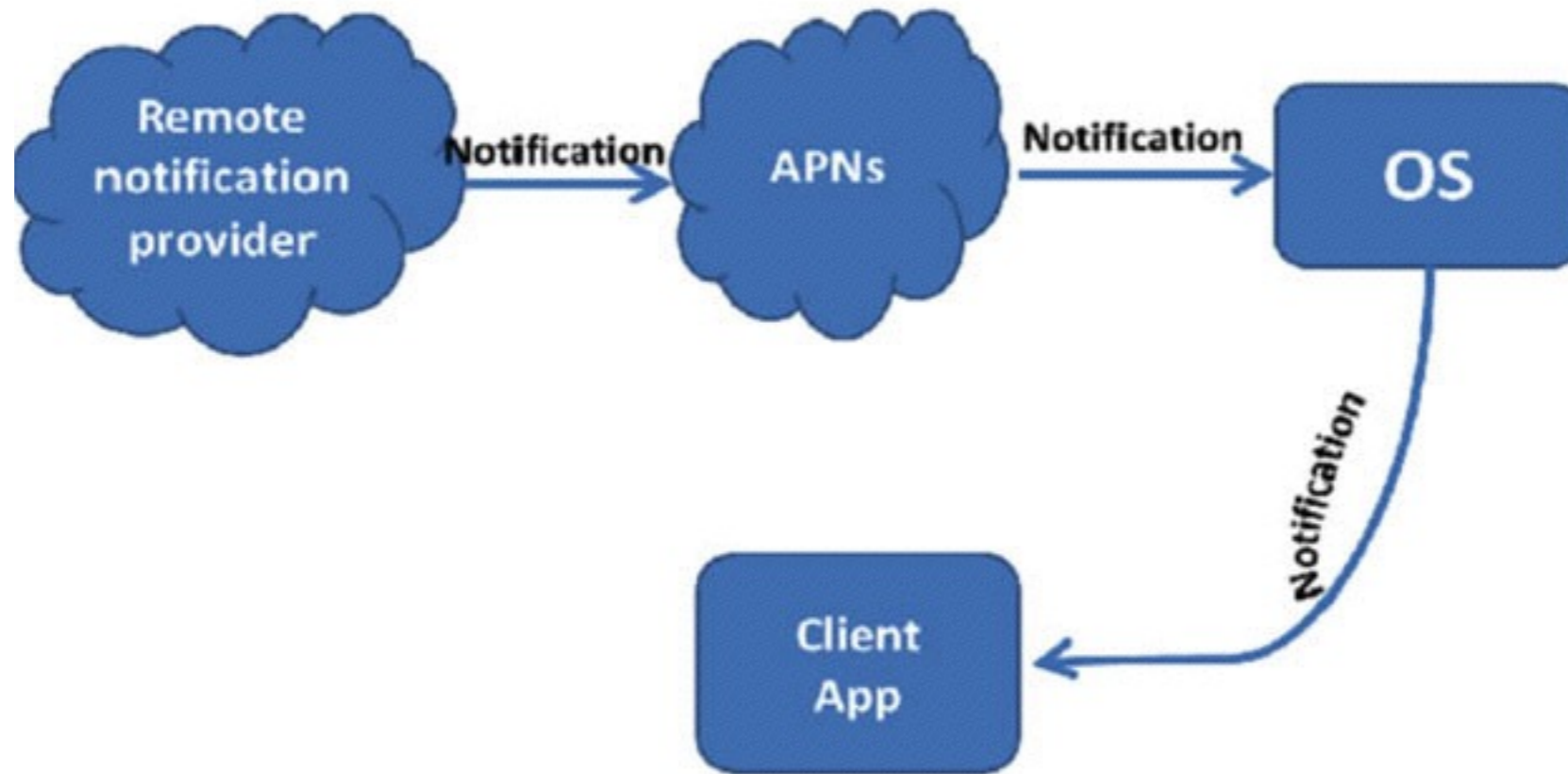
Possibilités d'envoyer des notifications silencieuses pour "réveiller" l'application et l'inviter à télécharger du nouveau contenu

/!\ Nécessite un serveur tier

APNS part 1 - Enregistration



APNS part 2 - Envoi de notifications



Outils indispensables

Client REST

- Postman : <http://www.getpostman.com/>
- RESTed : <https://itunes.apple.com/fr/app/rested-simple-http-requests/id421879749>
- Paw HTTP Client : <https://itunes.apple.com/fr/app/paw-http-client/id584653203>

Proxy HTTP

- Charles : <http://www.charlesproxy.com/>

10 minutes

C'est la durée de vie d'une application en background

Le système récupère de la mémoire partout où il le peut

La notion de "service" et de "background" durable n'existe pas

```
- (void)applicationDidEnterBackground:(UIApplication *)application;  
- (void)applicationWillEnterForeground:(UIApplication *)application;
```

Background modes

▼  **Background Modes**

ON

- Modes:
- Audio and AirPlay
 - Location updates
 - Voice over IP
 - Newsstand downloads
 - External accessory communication
 - Uses Bluetooth LE accessories
 - Acts as a Bluetooth LE accessory
 - Background fetch
 - Remote notifications

Steps: ✓ Add the "Required Background Modes" key to your info plist file

Introduit dans iOS 8

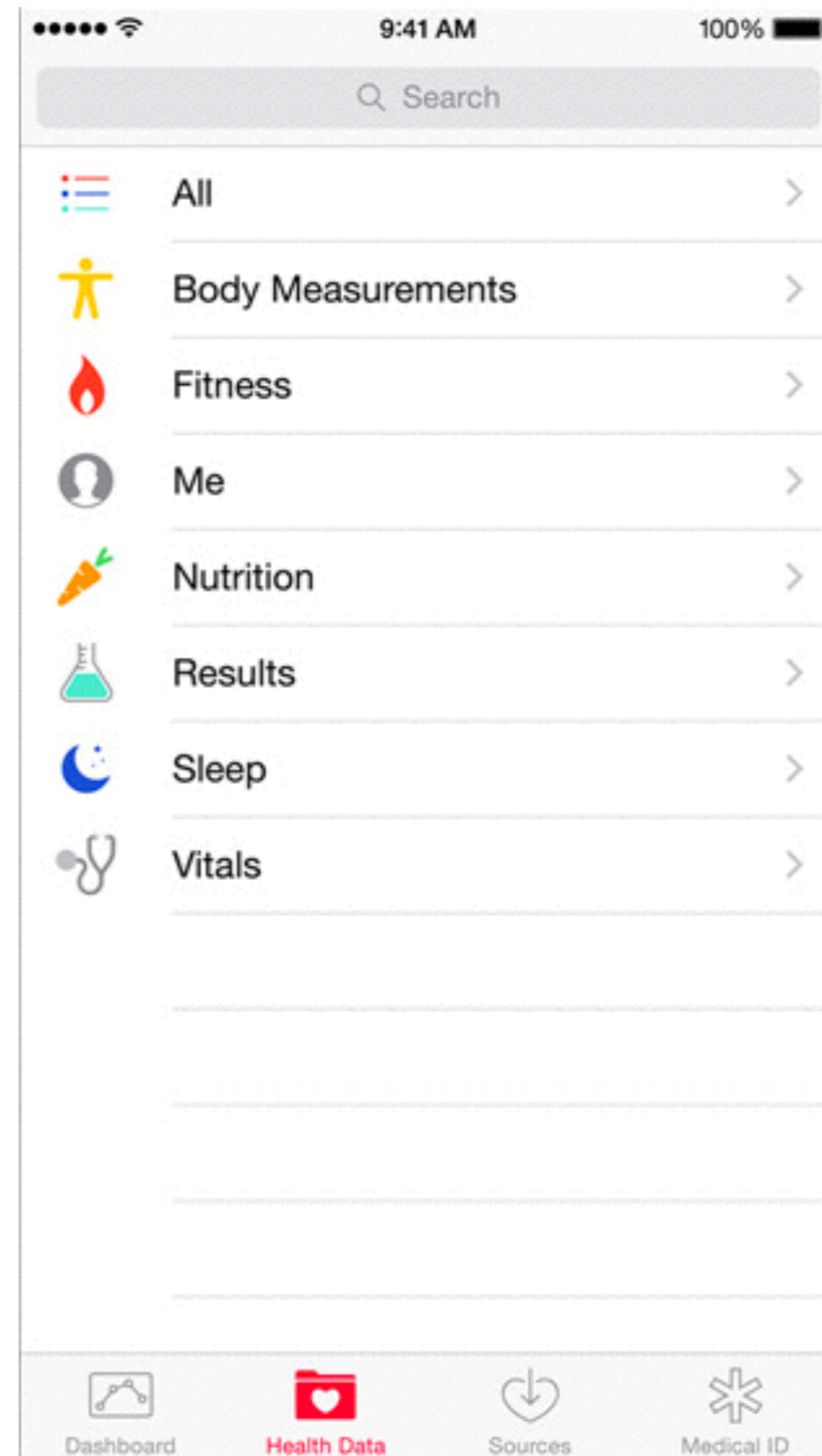
Permet aux applications liées à la santé et au fitness de partager des données entre elles

Métriques natives ou personnalisées

Données centralisées et sécurisées

L'utilisateur décide des données à partager par application (système de permissions)

Health



HomeKit

Centralise le contrôle d'objets connectés dans la maison

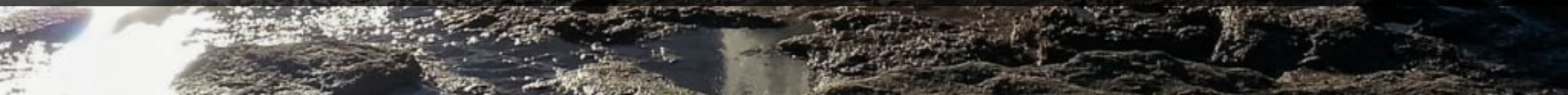
Apple's Home Automation Protocol

- Homes
- Rooms
- Accessories
- Services
- Zones

HomeKit.framework



Divers



Sources

<http://www.paintcodeapp.com/news/ultimate-guide-to-iphone-resolutions>

<https://github.com/AFNetworking/AFNetworking>

<http://www.adobe.com/devnet/air/articles/ios-push-notifications.html>

<https://developer.apple.com/notifications/>



Questions ?