

Développement iOS

Capteurs & Listes

Sommaire

- Rappels
- L'accès aux capteurs
- Créer des listes
- Utilisation avancée des Segue
- TP

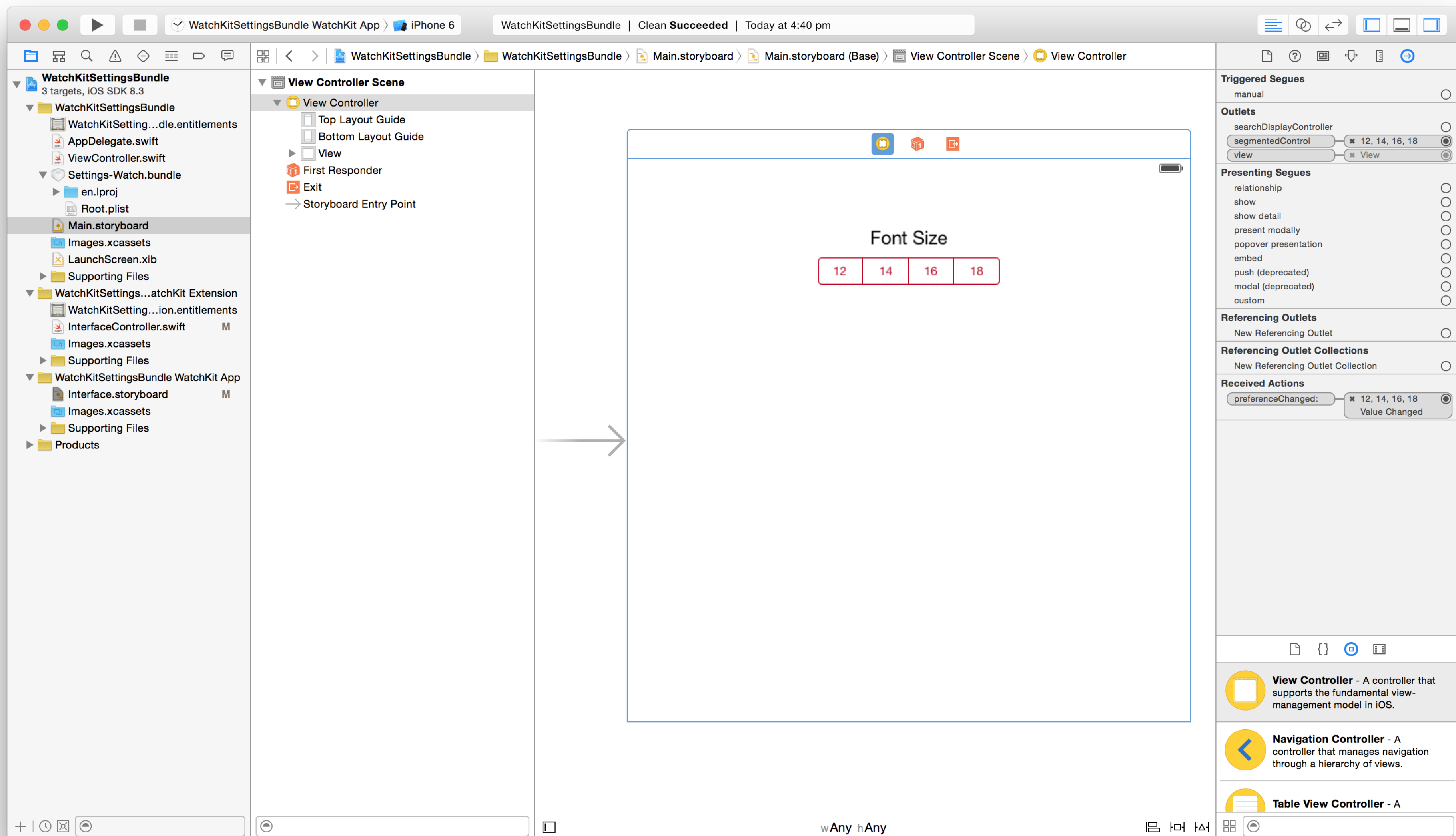


Rappels

Clavier : Aide-mémoire

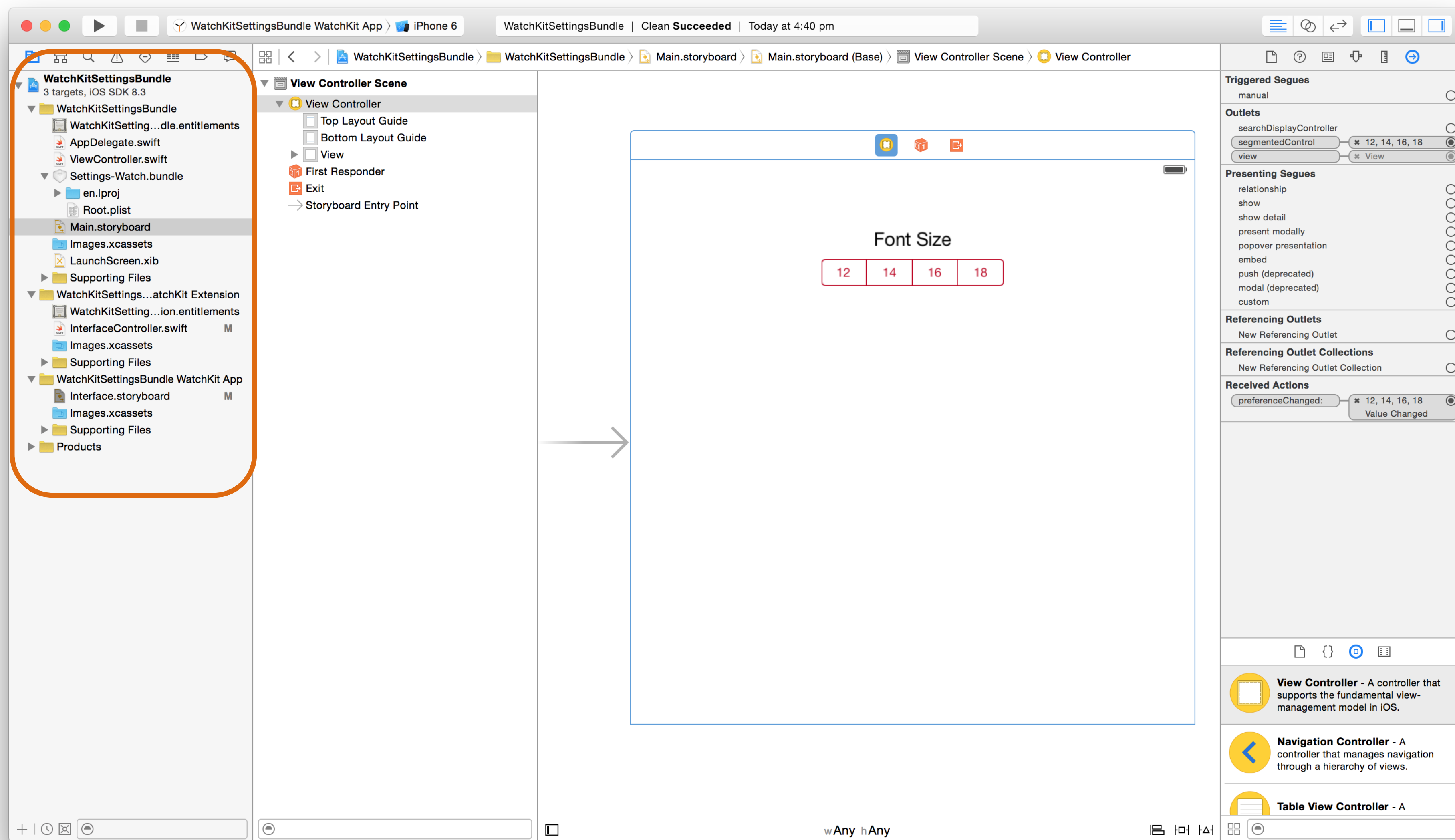
- Les accolades : $\backslash + ($ (et $\backslash +)$)
- Les crochets : $\backslash + \uparrow + ($ (et $\backslash + \uparrow +)$)
- Le pipe : $\backslash + \uparrow + |$
- Le backslash : $\backslash + \uparrow + /$
- Le tilde : $\backslash + n$

XCode



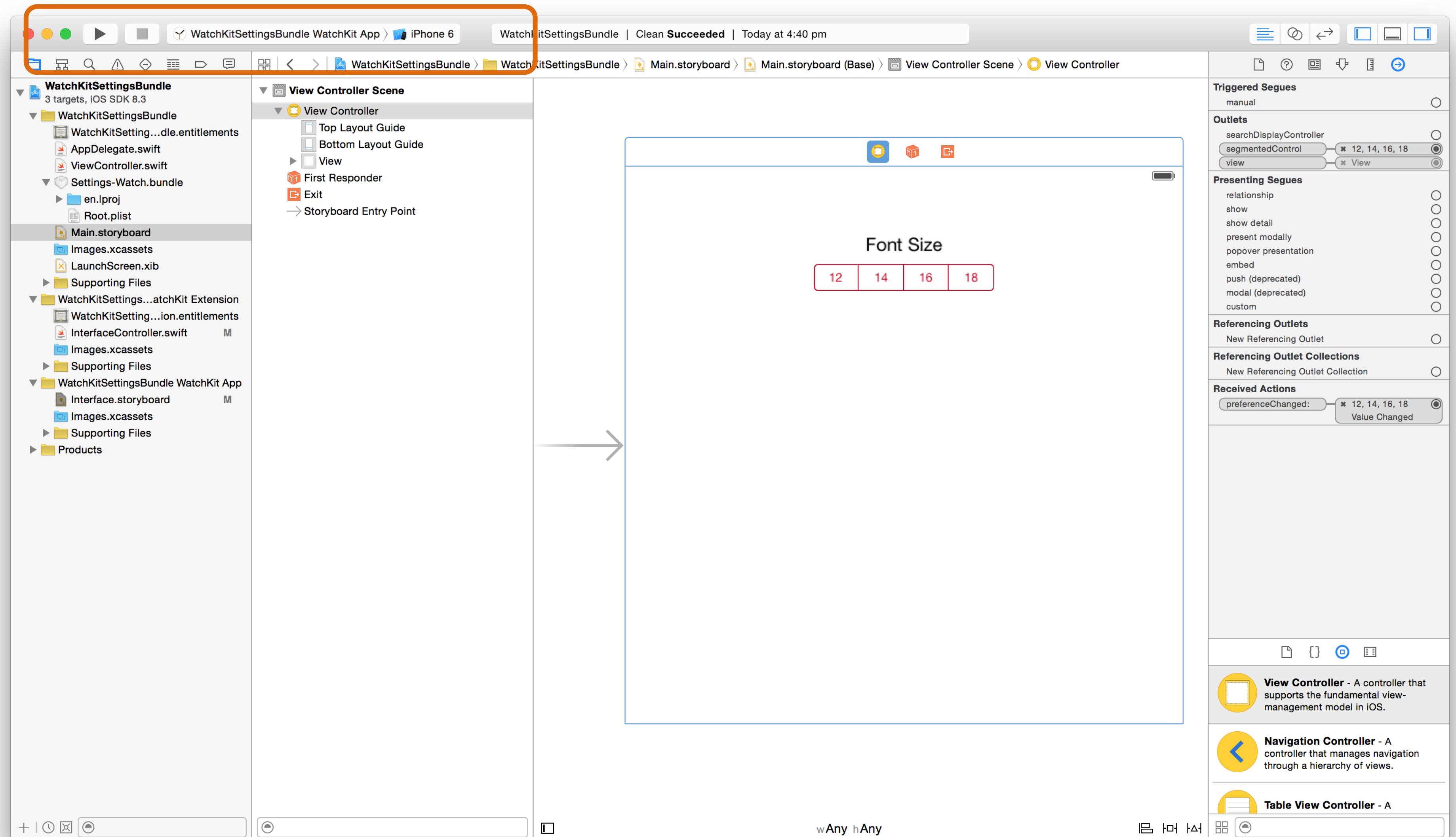
XCode

Arborescence du projet



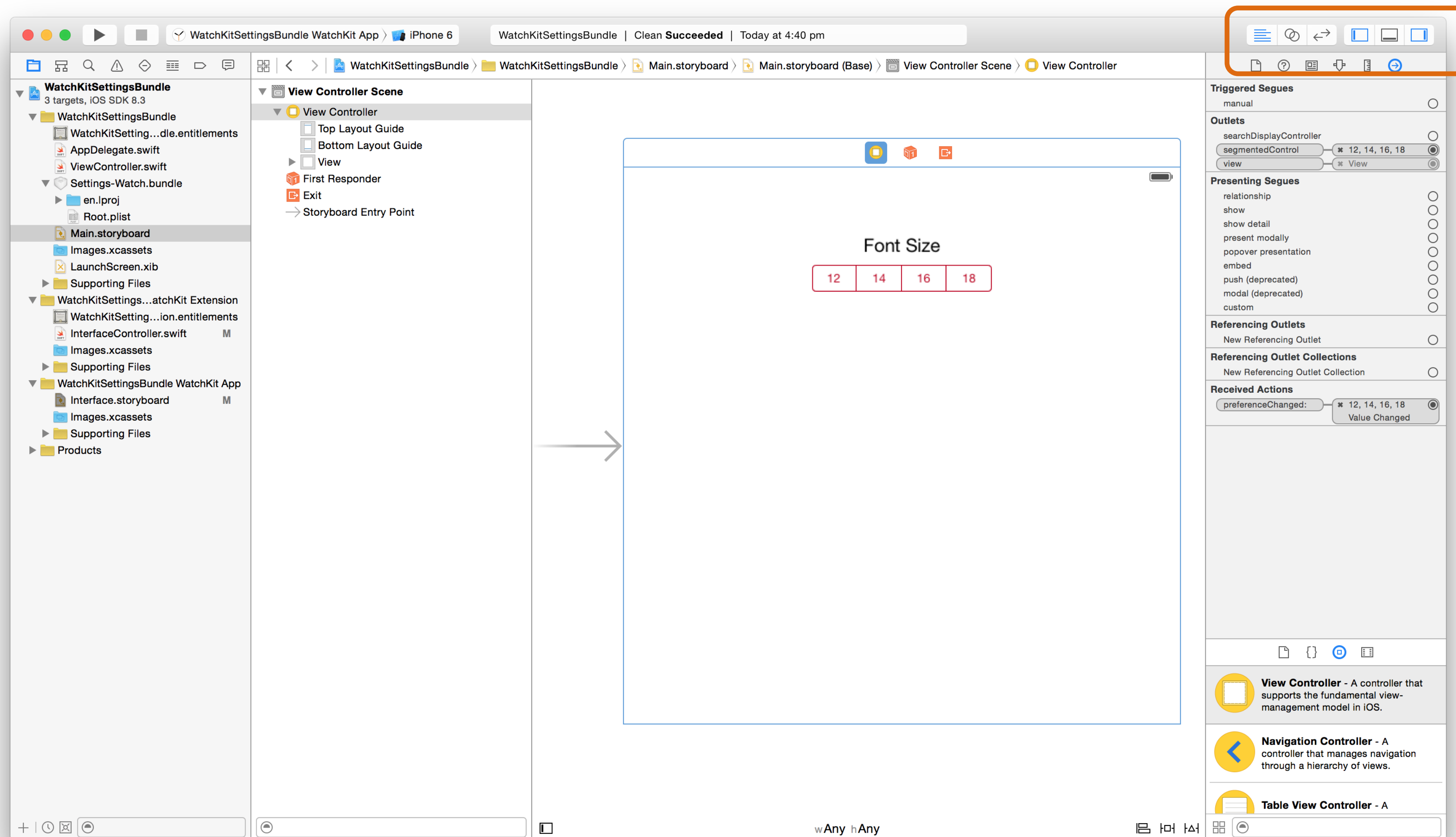
XCode

Exécution du projet

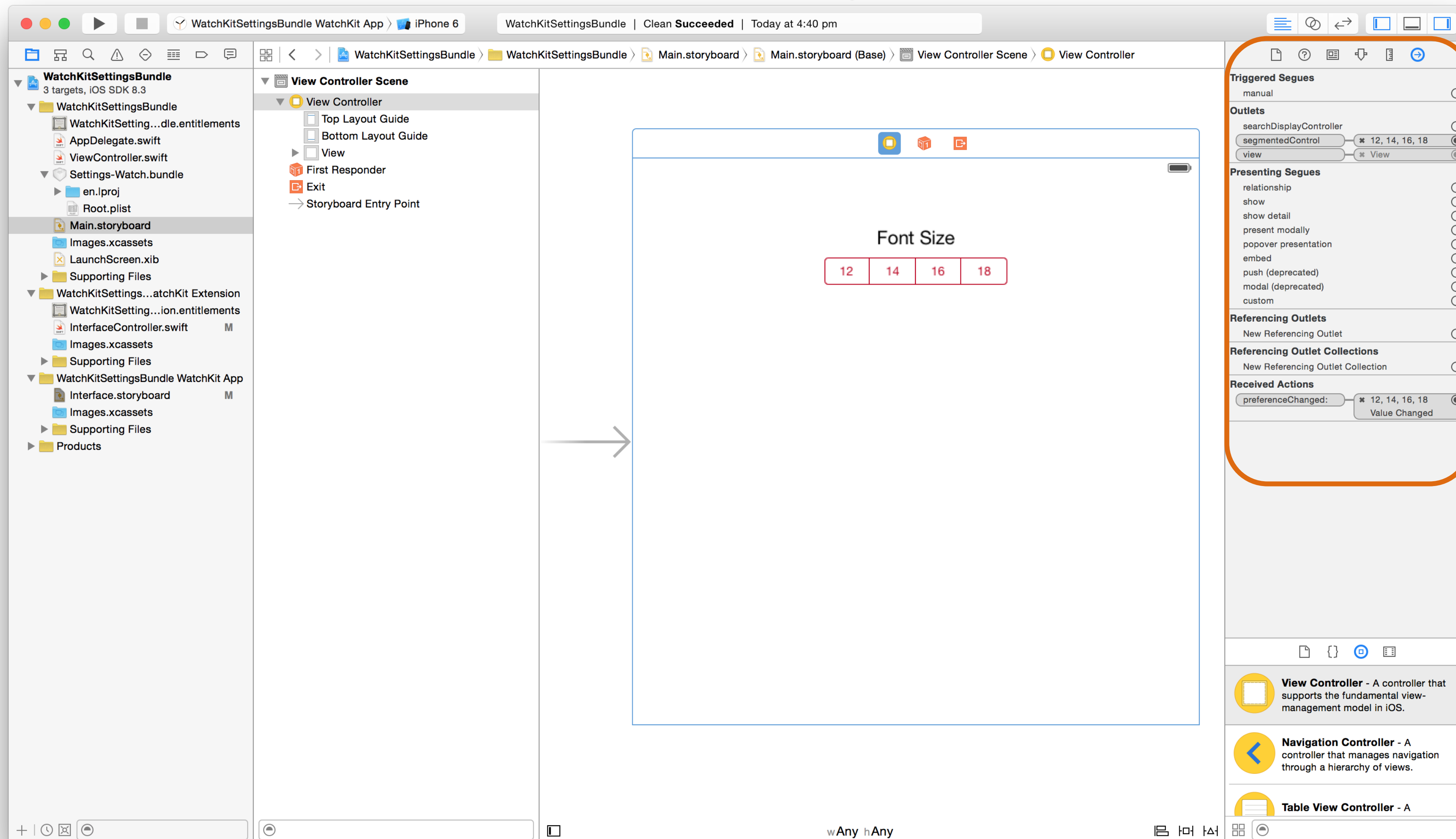


XCode

Affichage

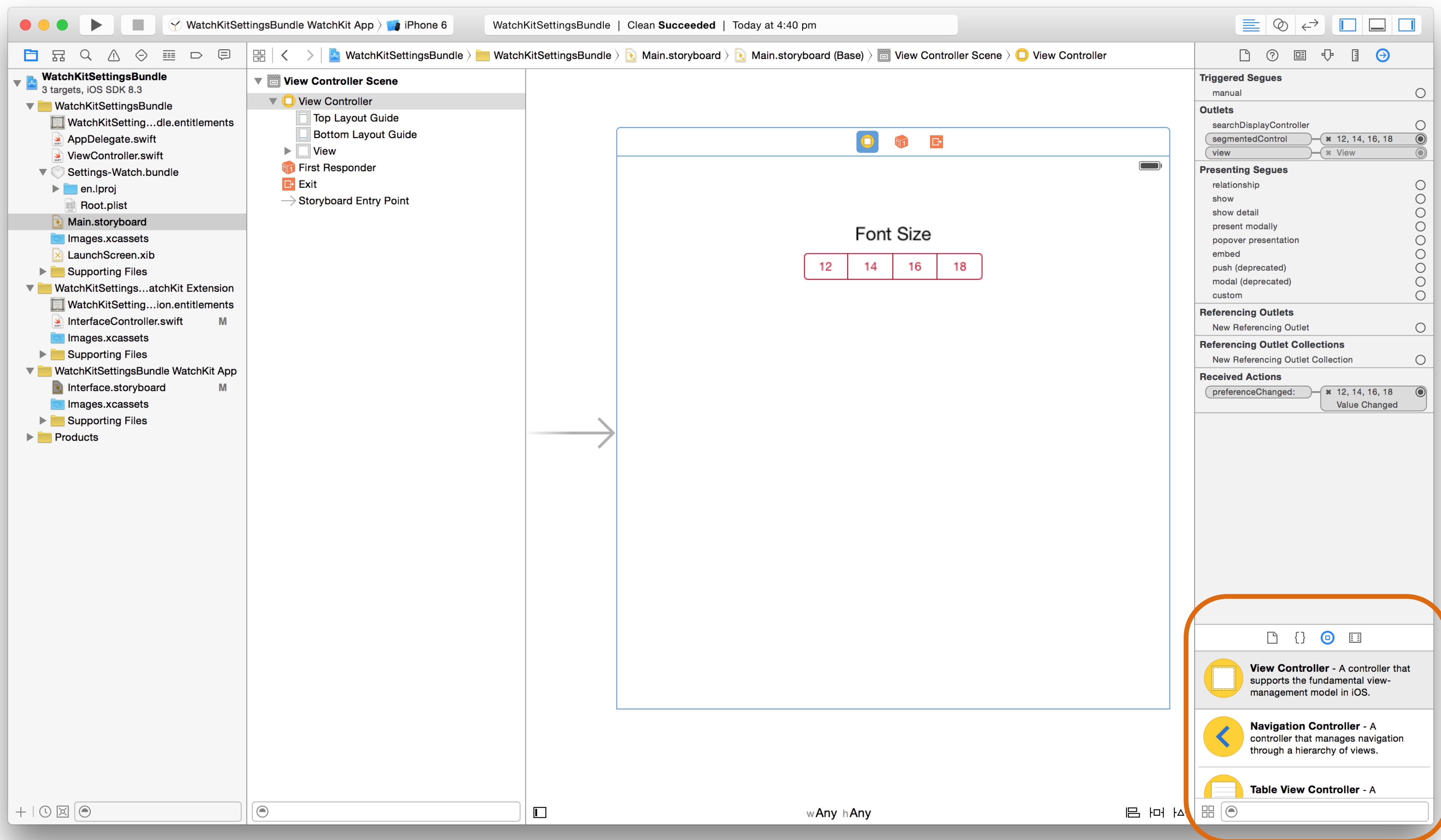


XCode



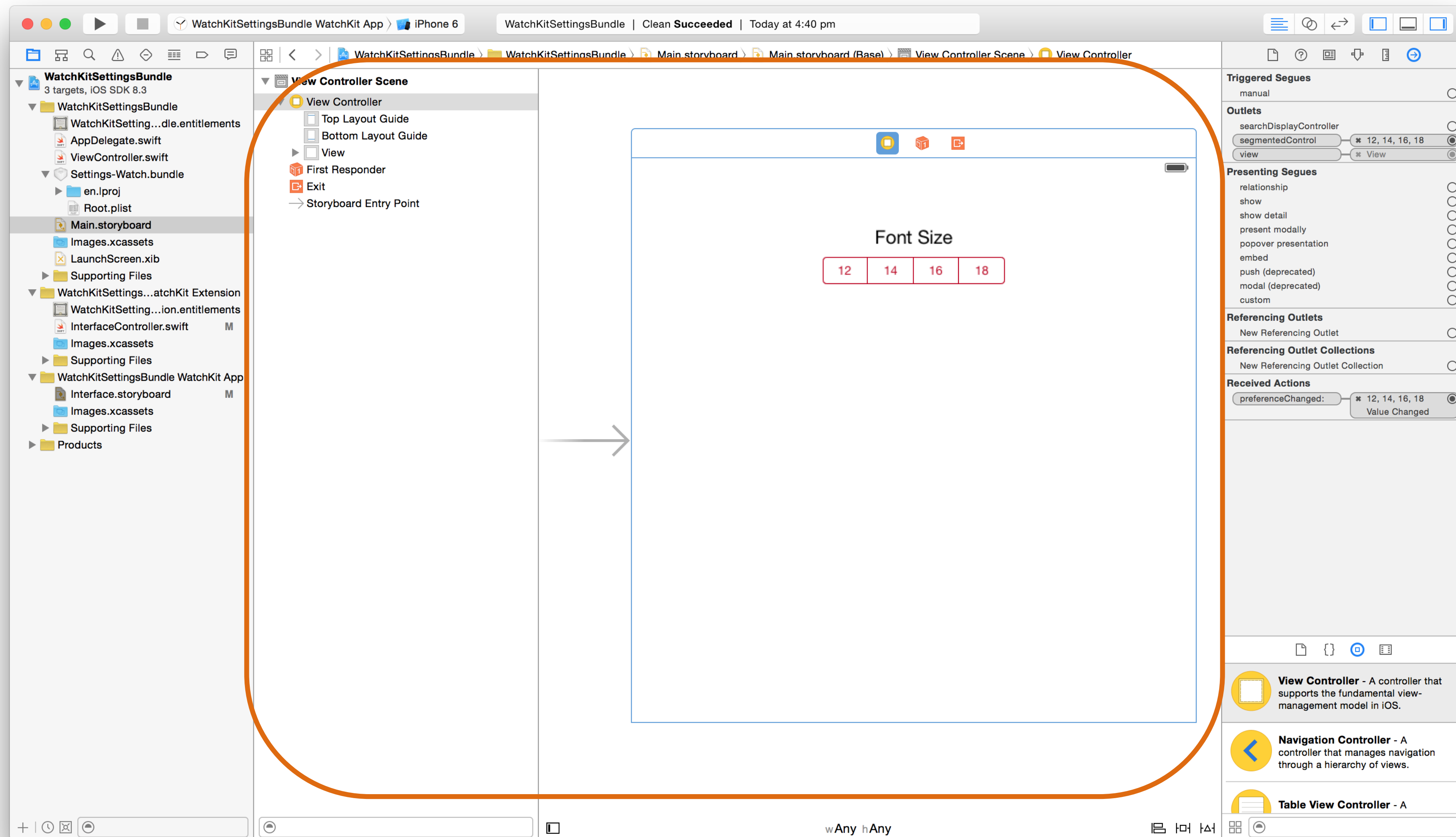
Attributs de l'objet

XCode



Eléments et snippets

XCode



Editeur

iOS & Développement Swift

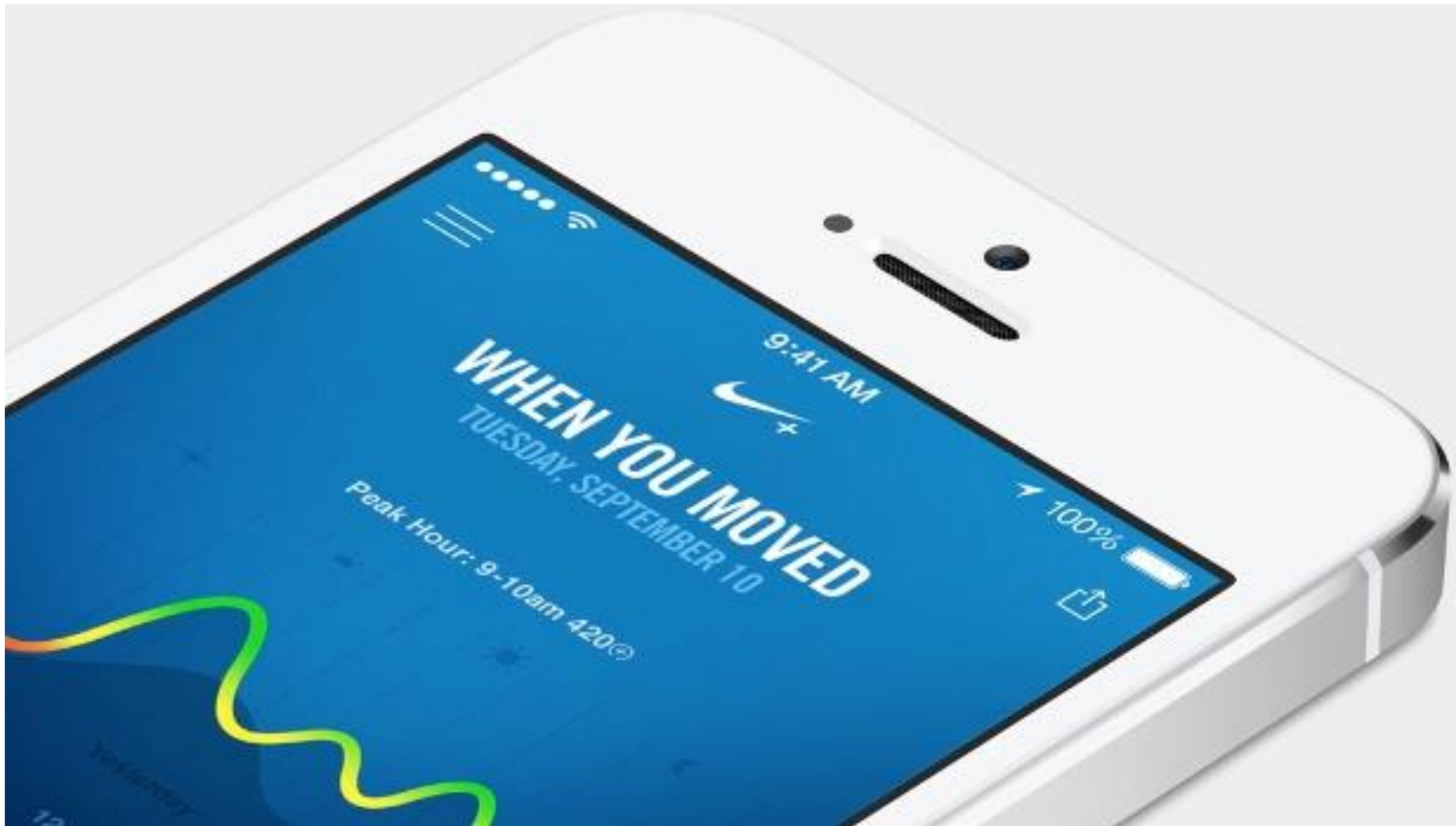
- Liens entre la View (Storyboard) et le Controller :
 - Utilisation d'un **IBAction** pour réaliser une action
 - Utilisation d'un **IBOutlet** pour référencer les éléments

iOS & Développement Swift

- Utilisation d'un **Segue** pour ouvrir une nouvelle vue :
 - Possibilité de la lier à un nouveau Controller
 - Réutilisation du Controller existant

iOS & Développement Swift

- Utilisation des optionnels : wrapper d'éléments *dangereux*
 - Déclaration avec un ?
 - Utilisation avec un !



Accès aux capteurs

Ce qui est accessible

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
Camera	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										OUI	OUI

Permissions

- Depuis iOS 10, le nombre de permissions à demander a augmenté, il concerne maintenant :
- Calendar, Contact, Reminder, Photo , Bluetooth Sharing , Microphone, Camera, Location, Health, HomeKit, Media Library , Motion, CallKit, Speech Recognition, SiriKit, TV Provider.
- Sous la clé **Privacy** dans le fichier info.plist

Permissions

Privacy - Calendars Usage Description	▲ ▼	String	\$(PRODUCT_NAME) calendar events
Privacy - Bluetooth Peripheral Usage Description	▲ ▼	String	\$(PRODUCT_NAME) blueTooth use
Privacy - Health Share Usage Description	▲ ▼	String	\$(PRODUCT_NAME) health share use
Privacy - Health Update Usage Description	▲ ▼	String	\$(PRODUCT_NAME) health use
Privacy - Camera Usage Description	▲ ▼	String	\$(PRODUCT_NAME) camera use
Privacy - Reminders Usage Description	▲ ▼	String	\$(PRODUCT_NAME) reminder use
Privacy - HomeKit Usage Description	▲ ▼	String	\$(PRODUCT_NAME) home use
Privacy - Media Library Usage Description	▲ ▼	String	\$(PRODUCT_NAME) media use
Privacy - Siri Usage Description	▲ ▼	String	\$(PRODUCT_NAME) siri use
Privacy - Speech Recognition Usage Description	▲ ▼	String	\$(PRODUCT_NAME) speech use
Privacy - Motion Usage Description	▲ ▼	String	\$(PRODUCT_NAME) motion use
Privacy - Microphone Usage Description	▲ ▼	String	\$(PRODUCT_NAME) microphone use
Privacy - TV Provider Usage Description	▲ ▼	String	\$(PRODUCT_NAME) tvProvider use
Privacy - Music Usage Description	▲ ▼	String	\$(PRODUCT_NAME) music use
Privacy - Contacts Usage Description	▲ ▼	String	\$(PRODUCT_NAME) contact use

Proximité

- Utilisation de la classe **UIDevice**
- Activation du capteur de monitoring
- Ajout d'un Observer

- Peut également servir à monitorer la batterie

Capture d'images

- Implémenter **UIImagePickerControllerDelegate**
- Implémentation des méthodes nécessaires (takePhoto, UIImagePickerController)
- Ajouter les permissions dans **Info.plist** :
 - NSCameraUsageDescription

Capture de son

- Importer **AVFoundation**
- Implémenter **AVAudioRecorderDelegate**
- Utilisation de l'élément **AVAudioRecorder**
- Ajouter les permissions dans **Info.plist** :
 - **NSMicrophoneUsageDescription**

Jouer un son

- Importer **AVFoundation**
- Implémenter **AVAudioPlayerDelegate**
- Utilisation de l'élément **AVAudioPlayer**

GPS

- Importer **CoreLocation**
- Implémenter **CLLocationManagerDelegate**
- Utilisation de l'élément **CLLocationManager**
- Ajouter les permissions dans **Info.plist** :
 - `NSLocationWhenInUseUsageDescription`
 - `NSLocationAlwaysUsageDescription`

Geofencing

- Permet de savoir si l'utilisateur entre/sort d'une zone.
- Méthode **didEnterRegion** et **didExitRegion**

Significant Location Changes

- Permet une notification pour un changement de position d'au moins 500 mètres
- **startMonitoringSignificantLocationChanges**
- **stopMonitoringSignificantLocationChanges**

Altitude

- Nécessite un device avec baromètre (depuis iPhone 6, sauf SE)
- Utilisation de **CMAltimeter** et **CMAltimeterData**

Accéléromètre

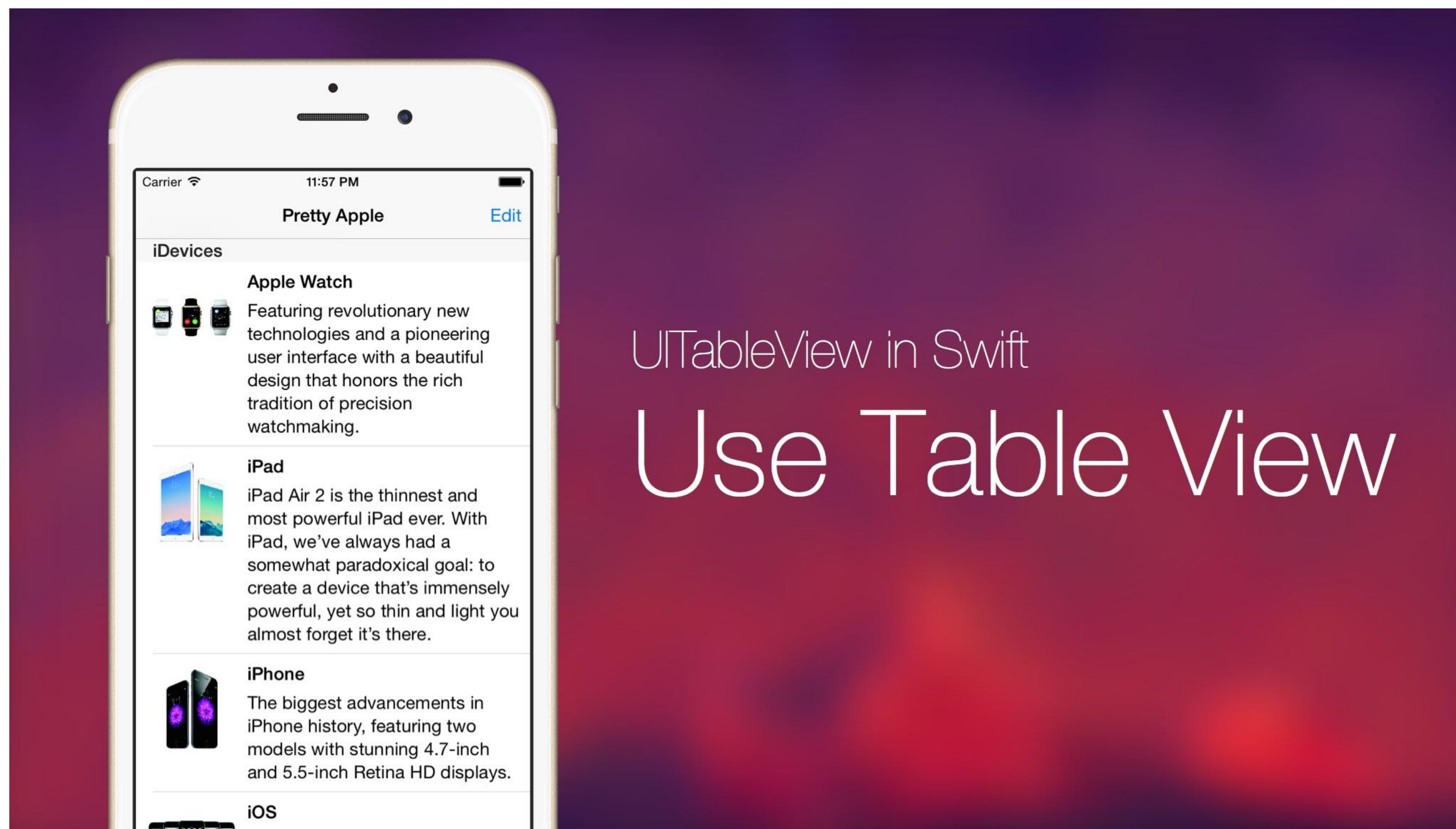
- Import de **CoreMotion**
- Utilisation de **CMMotionManager**

Gyroscope

- Rotation sur les 3 axes
- Utilisation de **CMMotionManager**

Darwin Notifications

- Permet d'écouter des évènements systèmes
- Voir :
<http://iphonedevwiki.net/index.php/SpringBoard.app/Notifications>



Créer des listes

Utilisation des listes

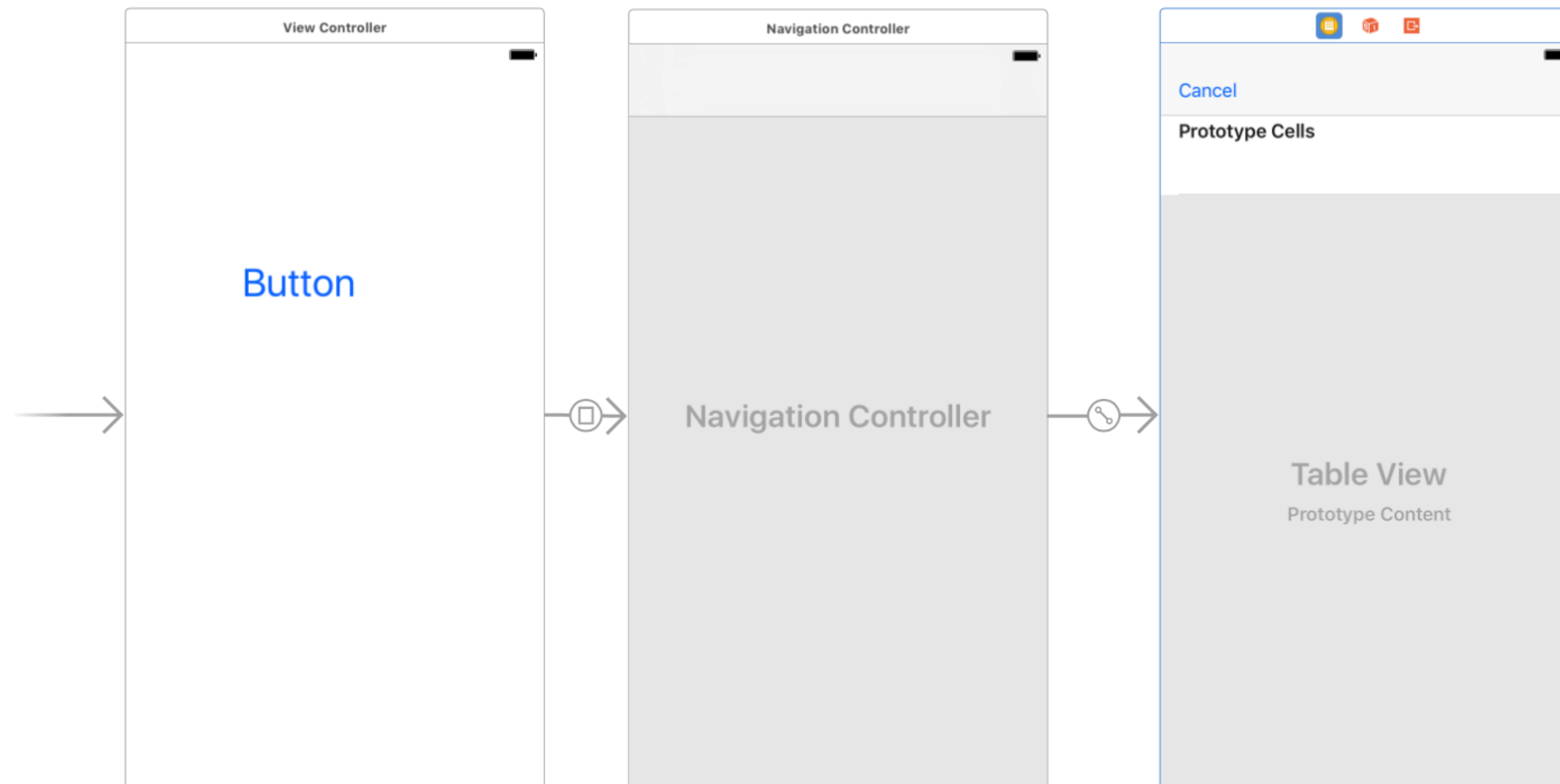
- L'utilisation des listes nécessite l'implémentation de 2 protocoles :
 - **UITableViewDataSource,**
 - **UITableViewDelegate**
- Dans le Storyboard, il faut ajouter une TableView
- Référencer le delegate et le datasource de l'objet TableView

Utilisation des listes

- Implémenter les méthodes demandés par les protocoles
 - Pour le DataSource, au minimum :
 - numberOfSectionsInTableView:
 - tableView:numberOfRowsInSection:
 - tableView:cellForRowAtIndexPath:
 - Pour le Delegate, aucune n'est obligatoire, mais souvent :
 - tableView:didSelectRowAtIndexPath

Utilisation des listes

- Le principe est le même pour les éléments comme UIPickerView
- Exemple d'implémentation: <http://www.codingexplorer.com/getting-started-uitableview-swift/>



Utilisation avancée des Segue

Passage de paramètres

- Utilisation de la fonction prepareForSegue

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?) {  
    if segue.identifier == "Segue"  
    {  
        let secondViewController = segue.destinationViewController as! SecondViewController  
        secondViewController.message = self.Text_Field.text!  
    }  
}
```



TP

TP1

- Créer une application de type « Todo List » :
 - Une liste dans laquelle on peut ajouter/supprimer un élément avec un label
 - Ajouter la notion de priorité pour la tâche. Classer la liste en fonction des priorités
 - Changer la couleur d'un élément en fonction de sa priorité

TP2

- Création d'un browser
 - Un champ texte de saisie de l'URL
 - Ouverture de la page dans une WebView
 - Si on secoue le téléphone, la page est rechargée

TP3

- Utilisation des capteurs :
 - Faire une liste de capteurs
 - Au clic sur un élément de la liste, on affiche le détail du capteur

Une application du type Master-Detail peut être utilisée, profitez-en pour observer les éléments créés (Navigation Controller, Table View Controller ...)