

Développement iOS

Introduction

Sommaire

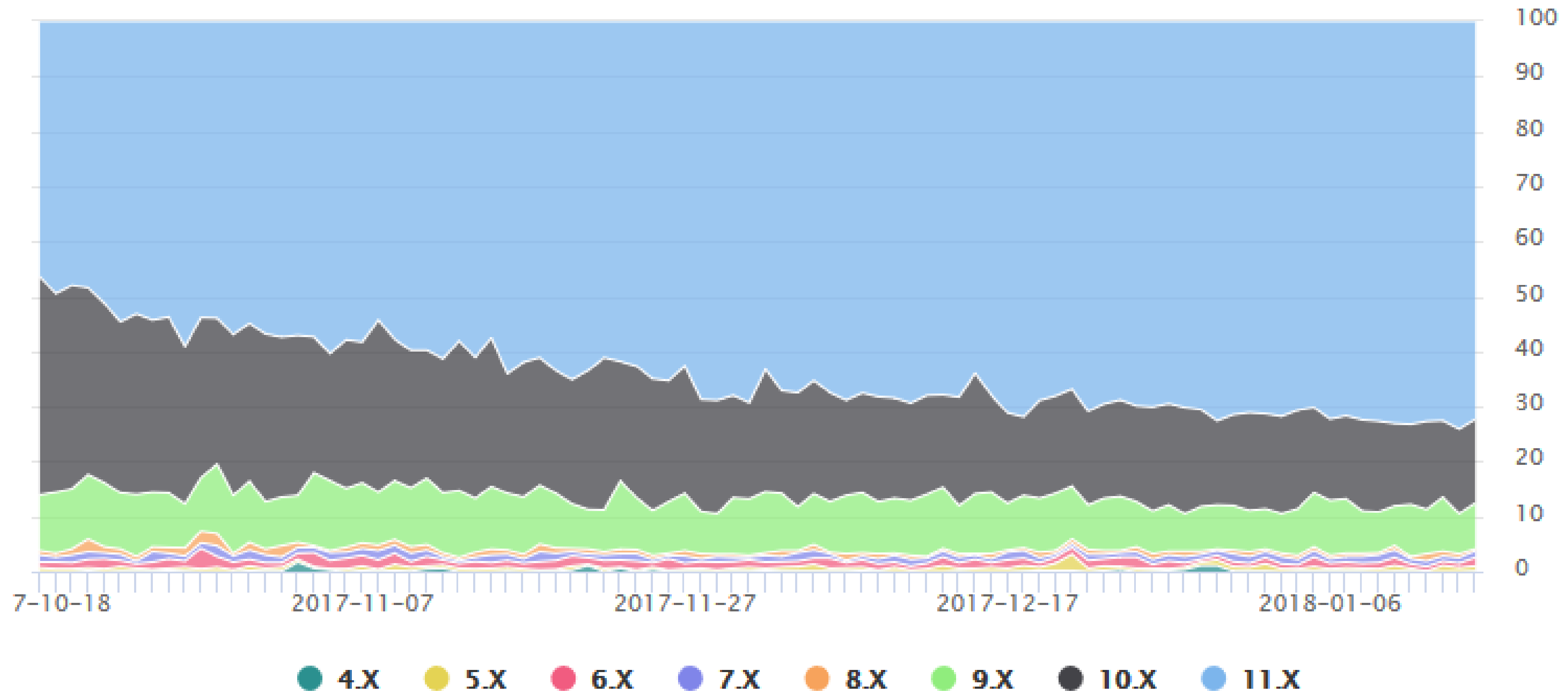
- Introduction
- Les outils de développement
- Objective-C à Swift
- TP



Introduction

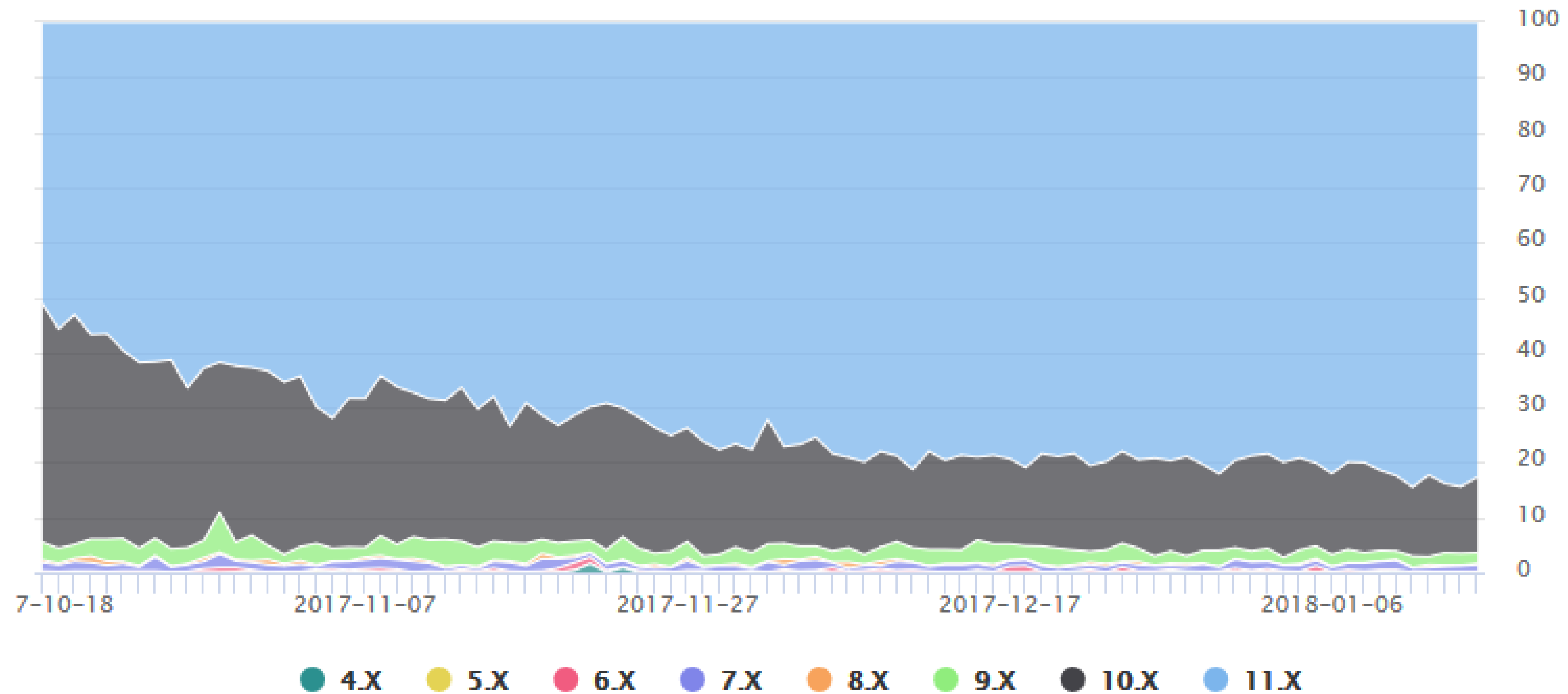
iOS

- Début en 2007
- Objective-C depuis le début
- Swift depuis 2014



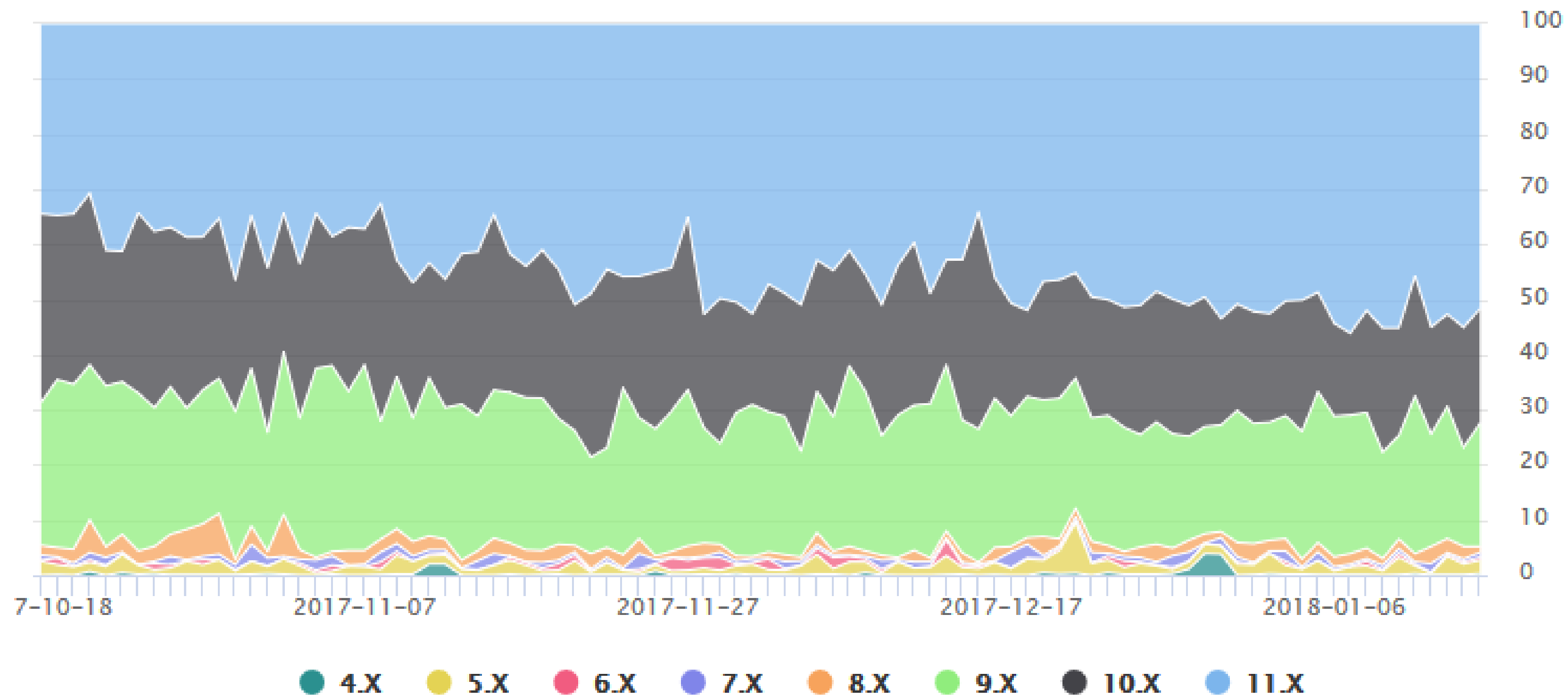
iOS : Segmentation

Une application doit être réalisée pour les 2 dernières versions de l'OS



iPhone : segmentation

Une application iPhone doit se concentrer sur les dernières versions de l'OS



iPad : segmentation

La dernière version iOS est moins présente sur iPad, mais les 3 dernières versions représentent plus de 90% des versions

App Store

2.2 millions d'applications

140 milliards de téléchargements depuis
le début

Deux tiers des applications gratuites





Les outils de développement

Environnement matériel

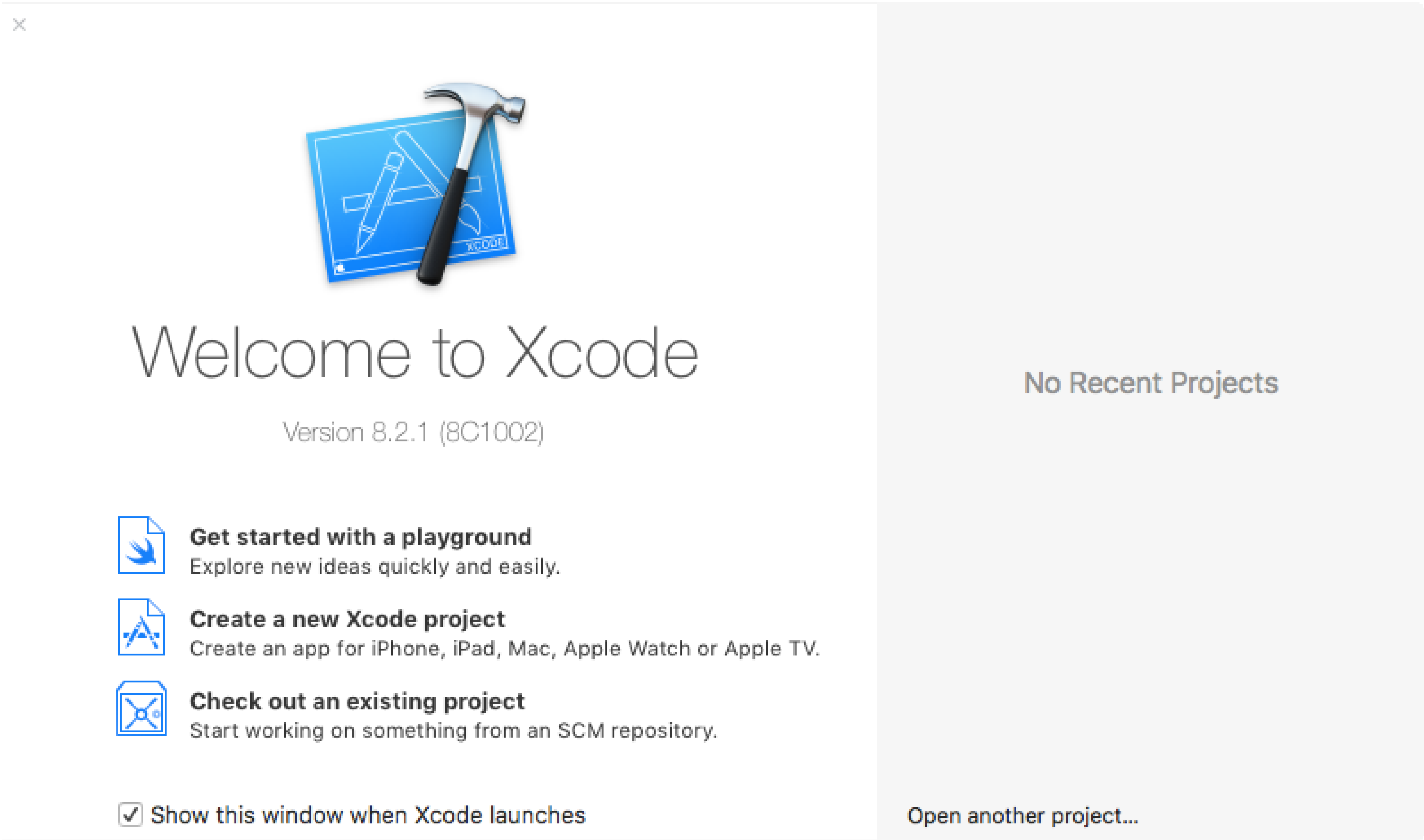
- L'environnement de développement n'existe que sur MacOS
- Nécessite un Mac, ainsi qu'un device iOS
- La dernière version de MacOS est généralement indispensable
- Attention aux changements pour le passage au monde Apple

Les différences Apple

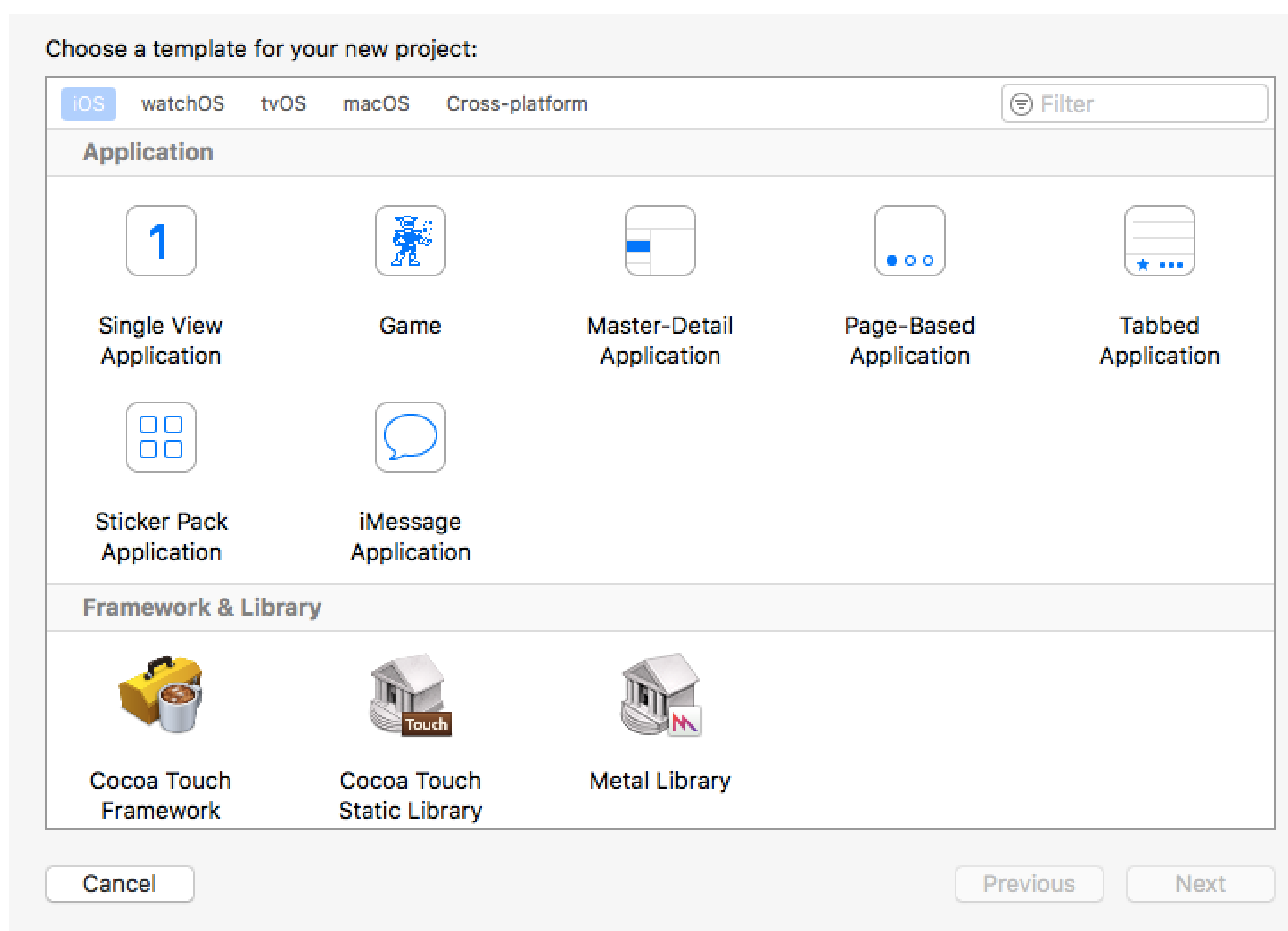
- Layout clavier différent
- Gesture avec plusieurs doigts
- Raccourcis propres à MacOS

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 : $\backslash + n$

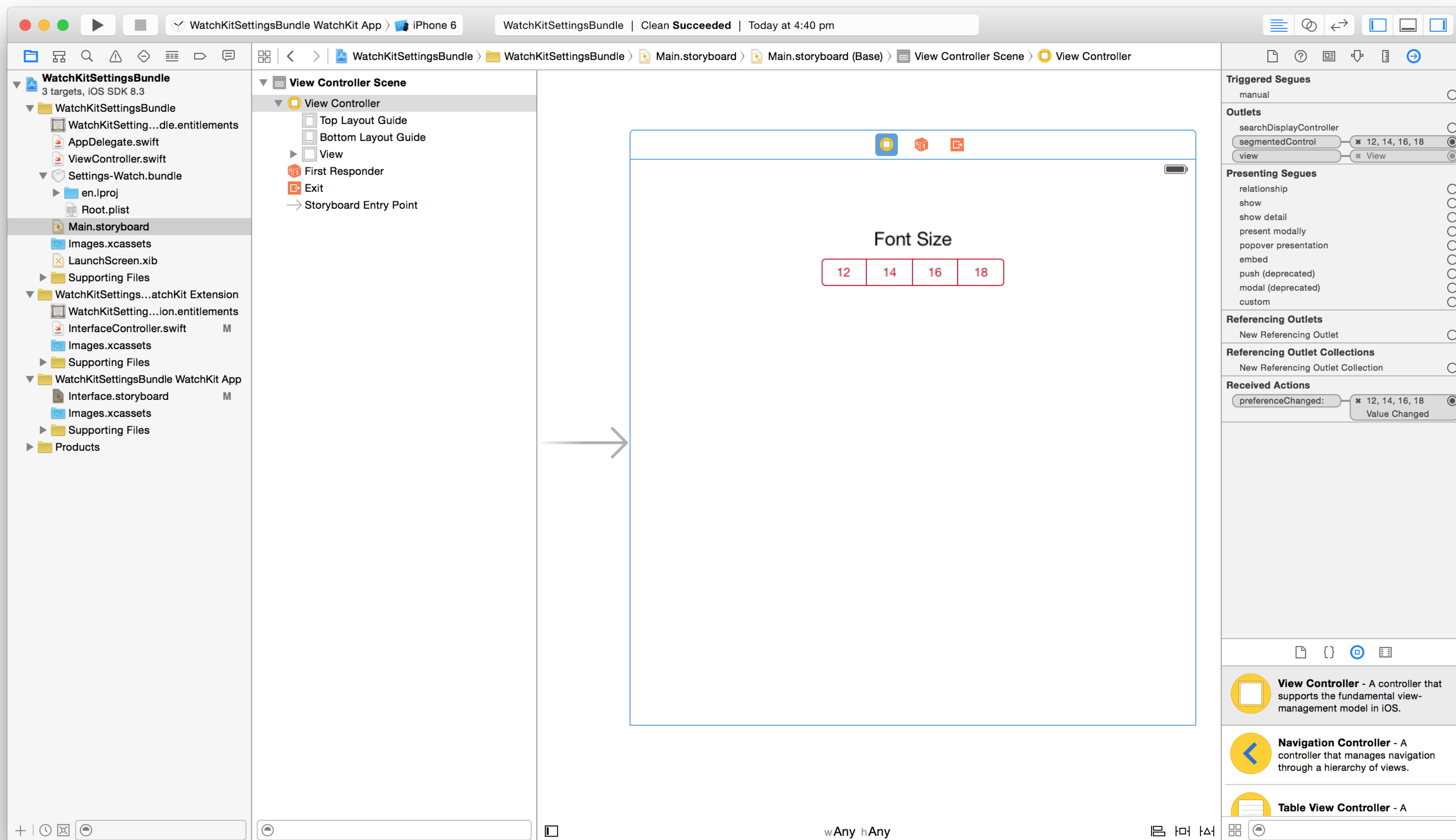


Xcode



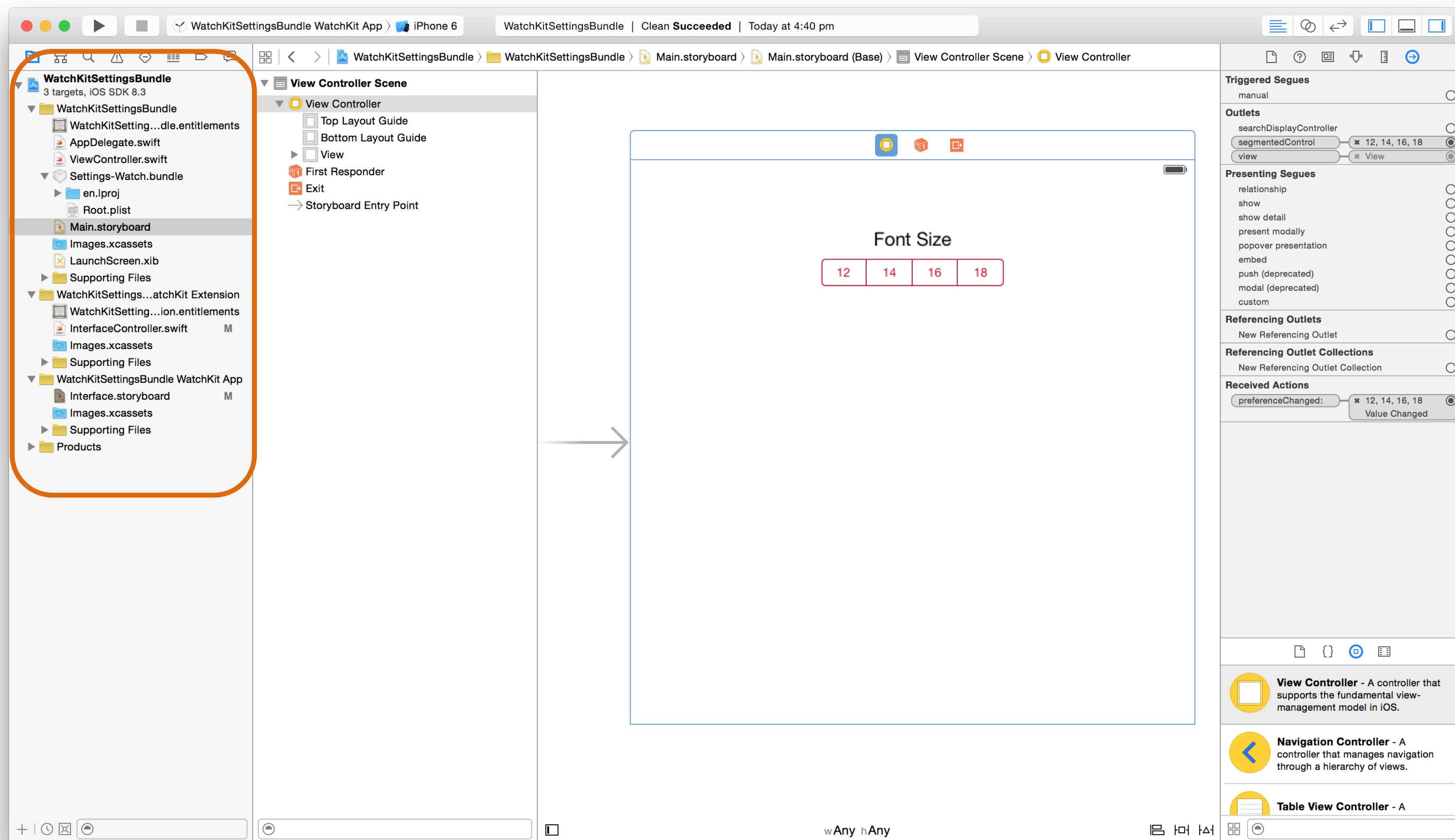
Choix d'un template projet

XCode



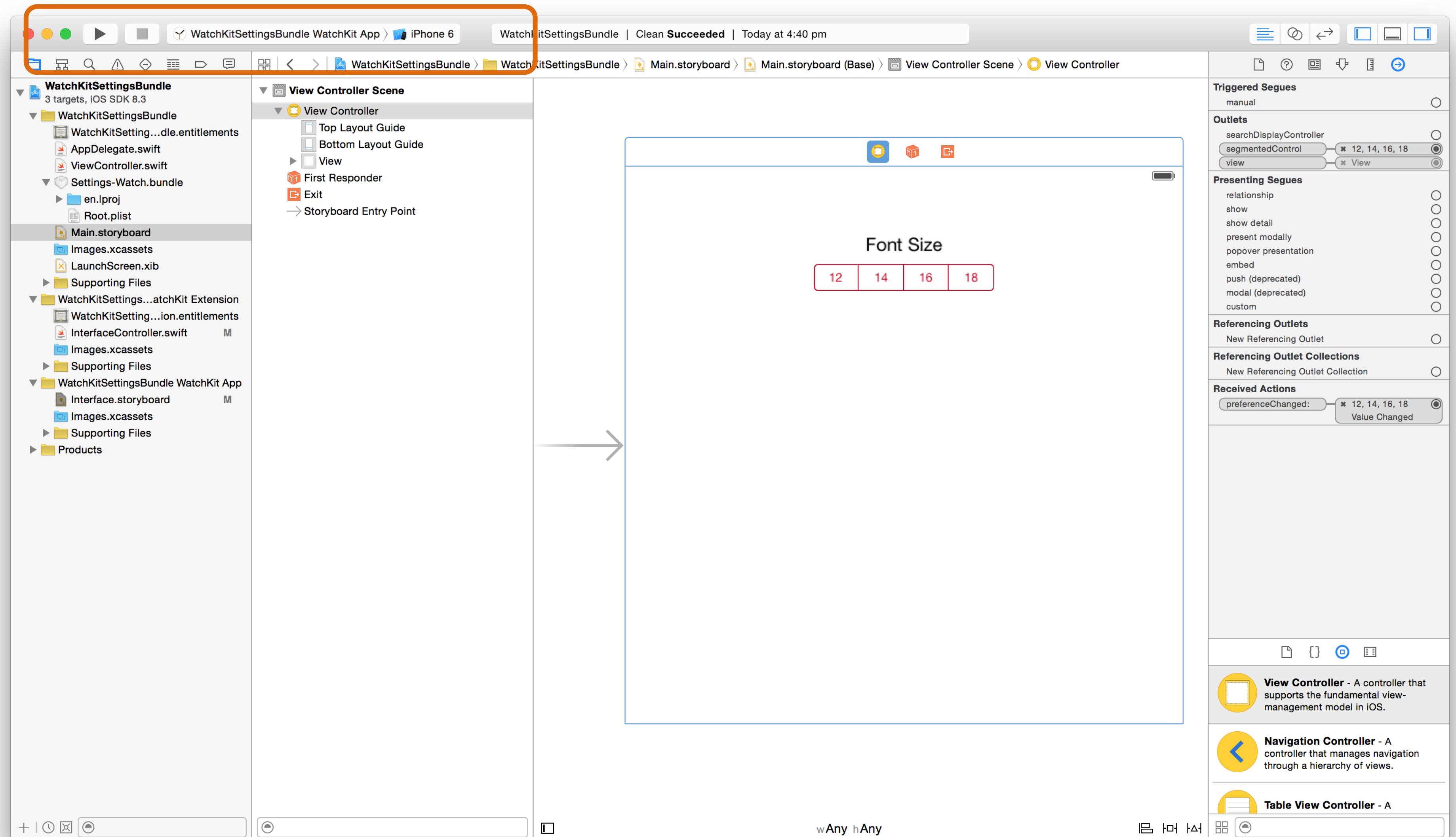
XCode

Arborescence du projet



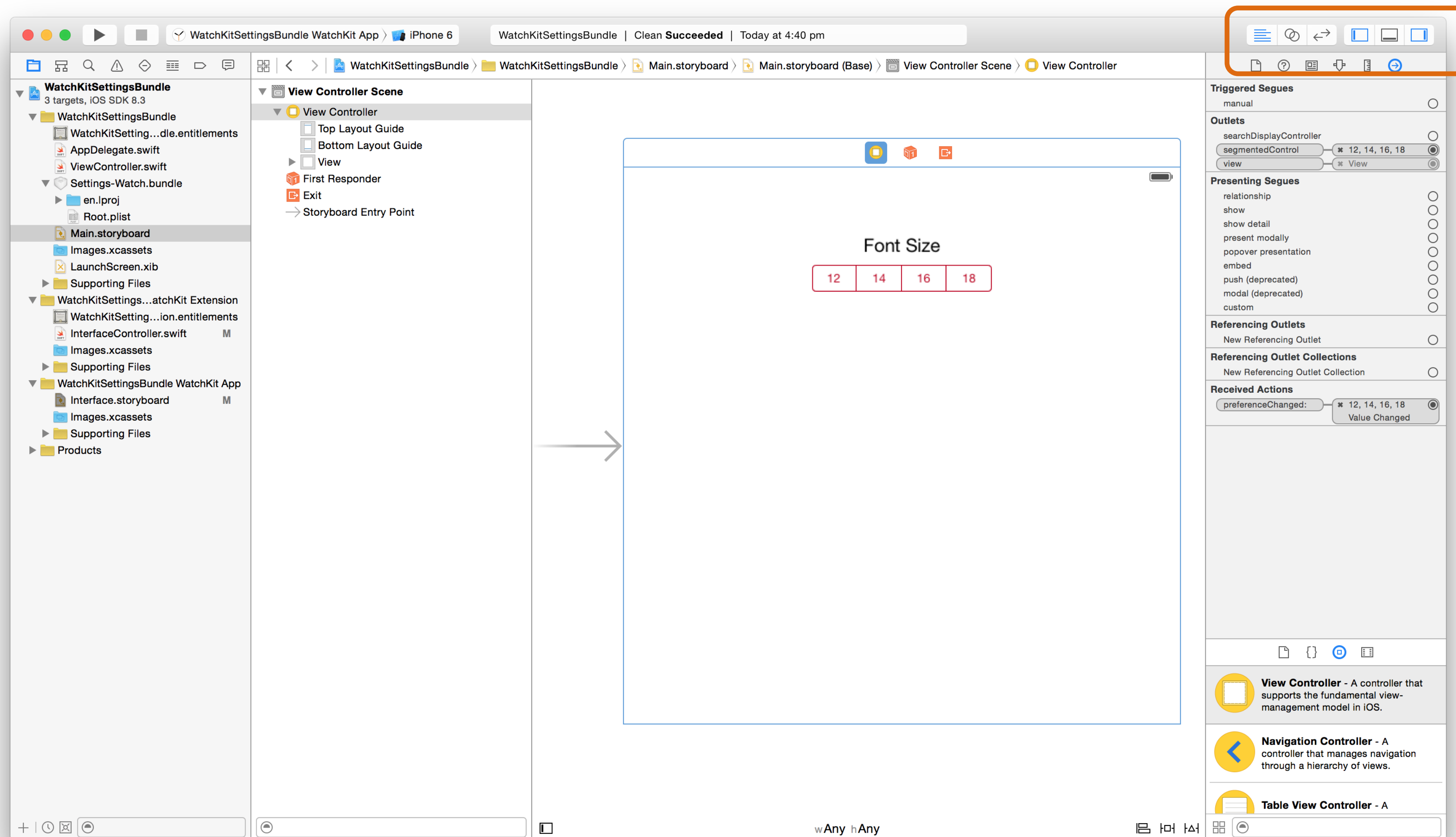
XCode

Exécution du projet

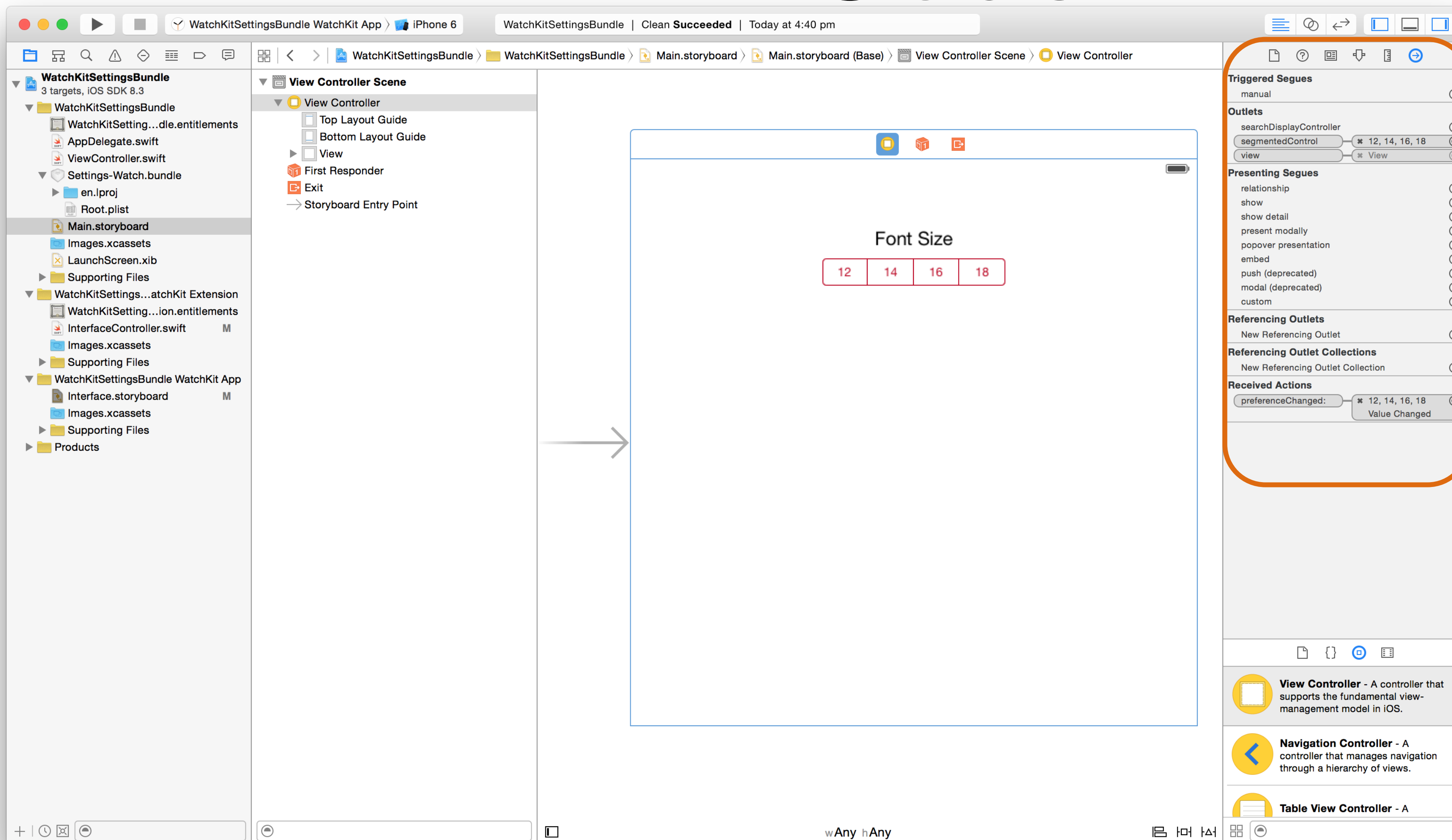


XCode

Affichage

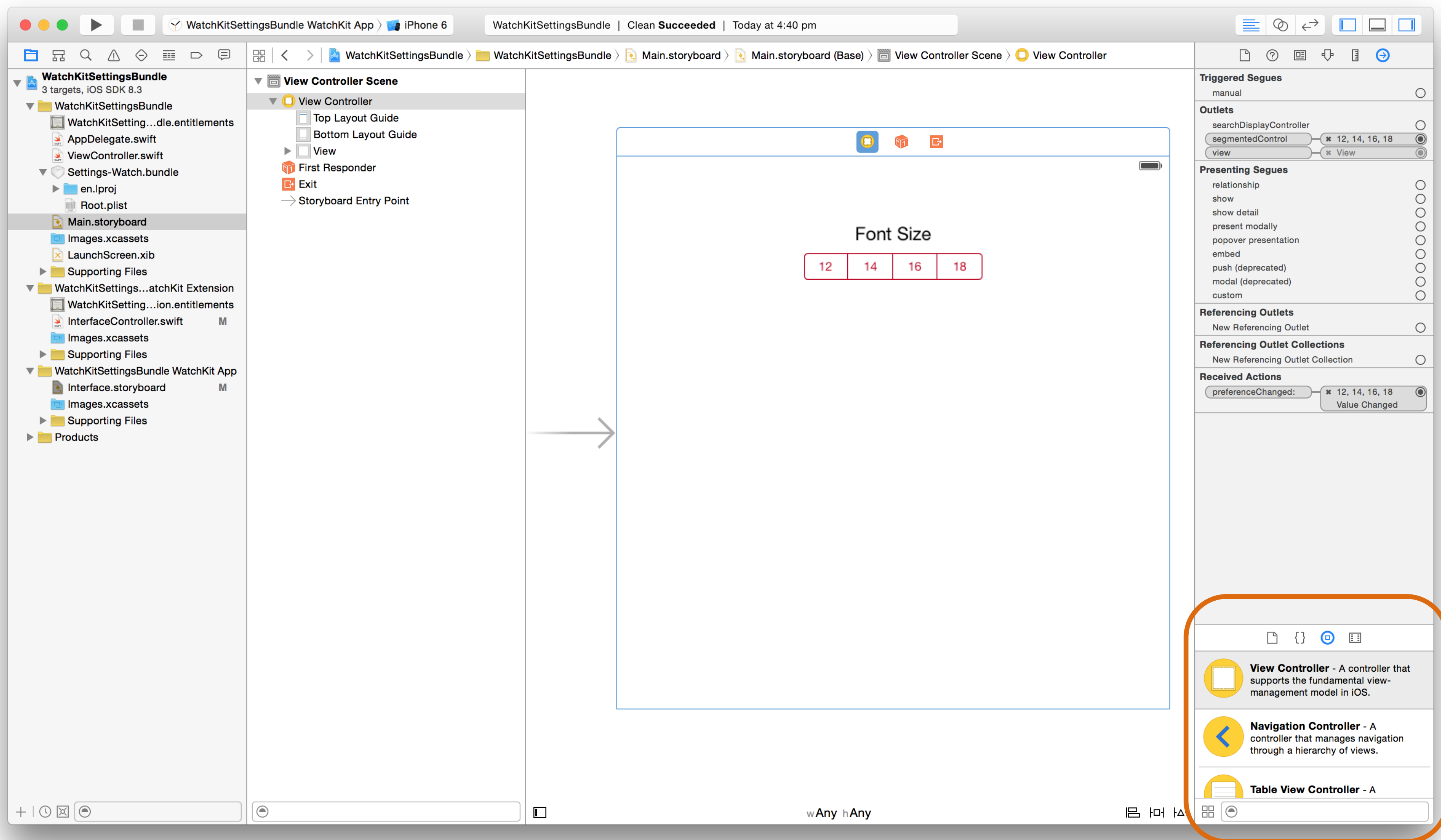


XCode



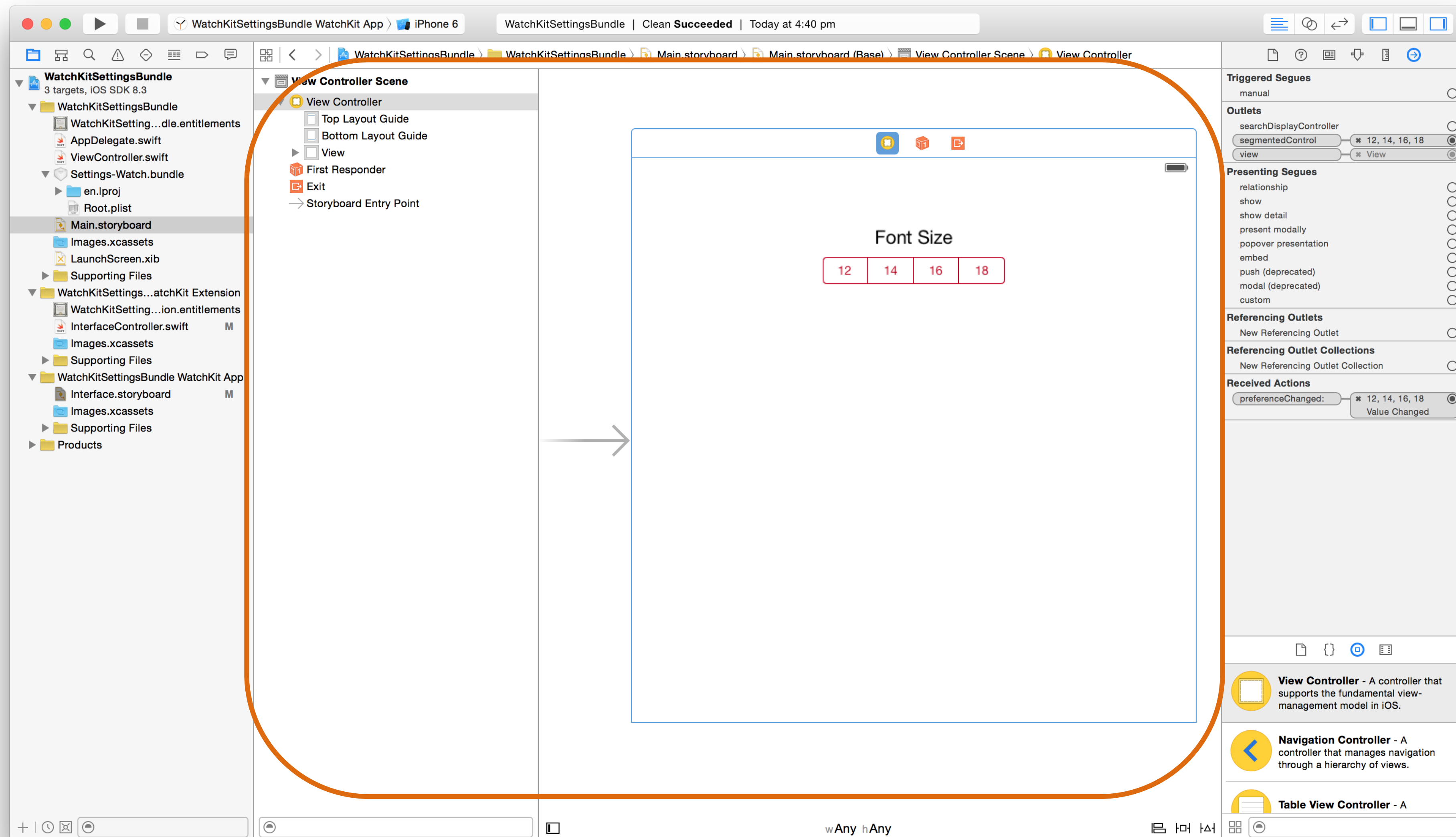
Attributs de l'objet

XCode

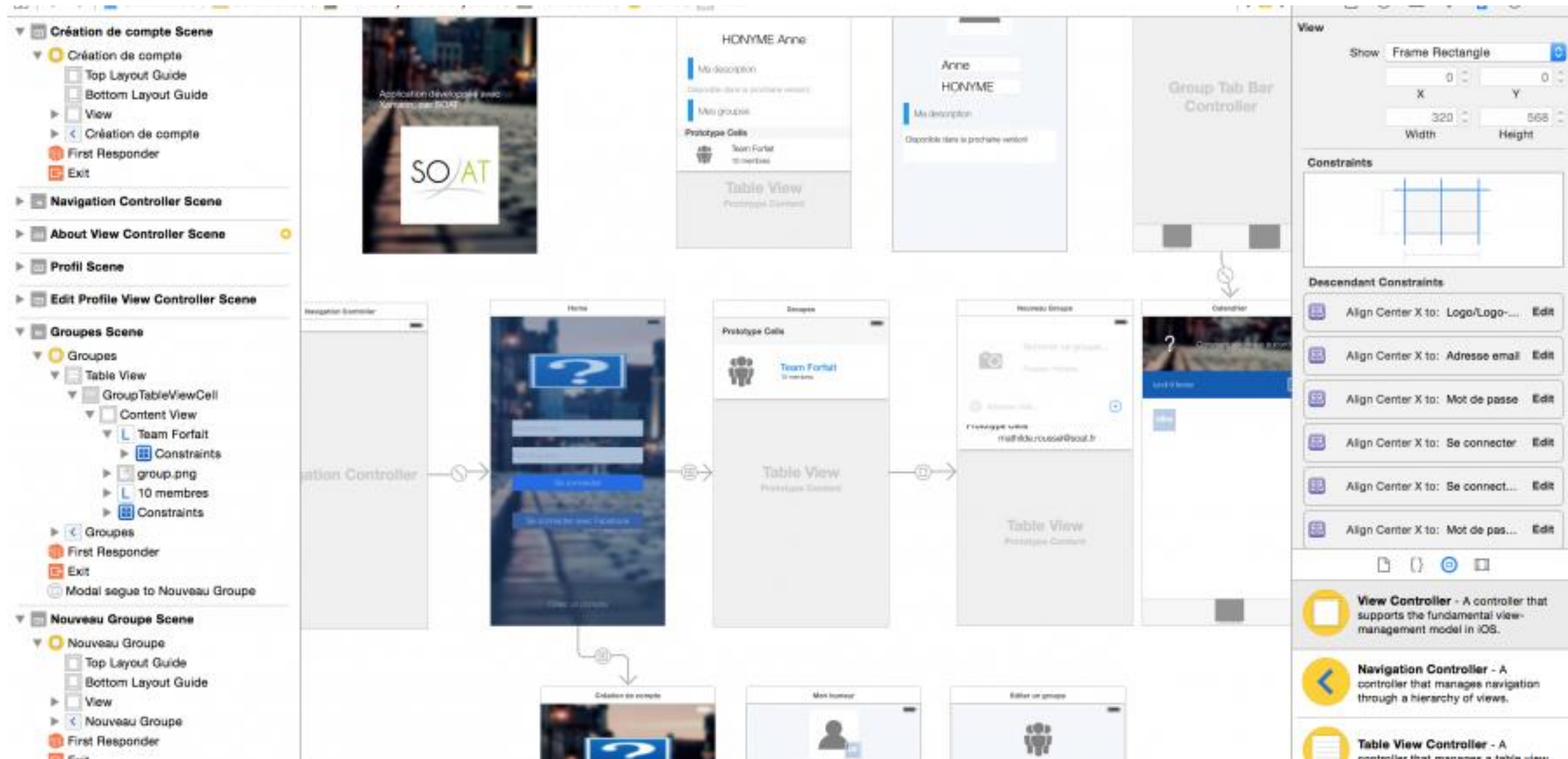


Eléments et snippets

XCode



Editeur



Le storyboard

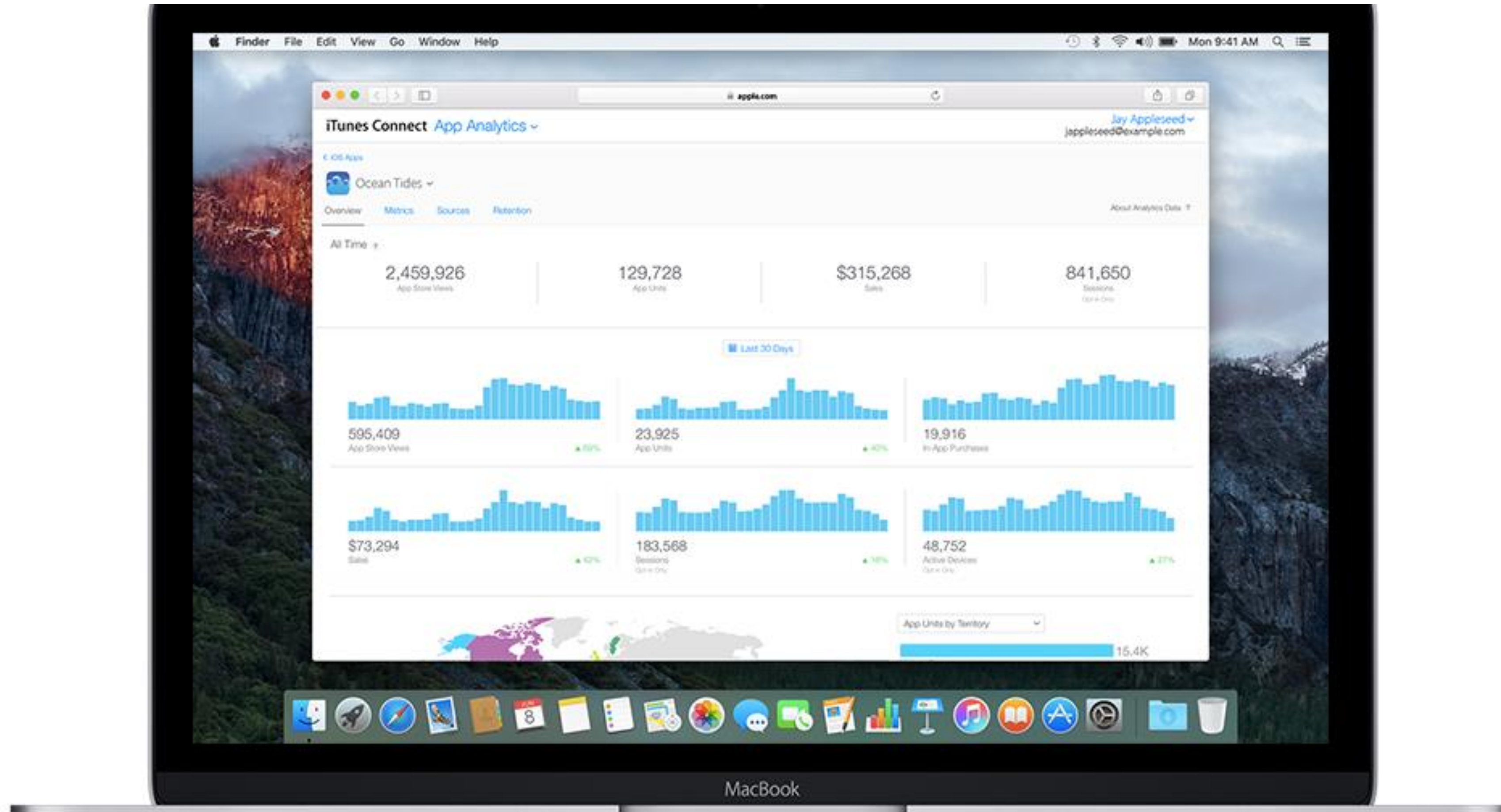
Décrit le cycle de l'application par l'intermédiaire de ses écrans
Attention, sur iOS, tout se passe dans un seul fichier.

Developer program

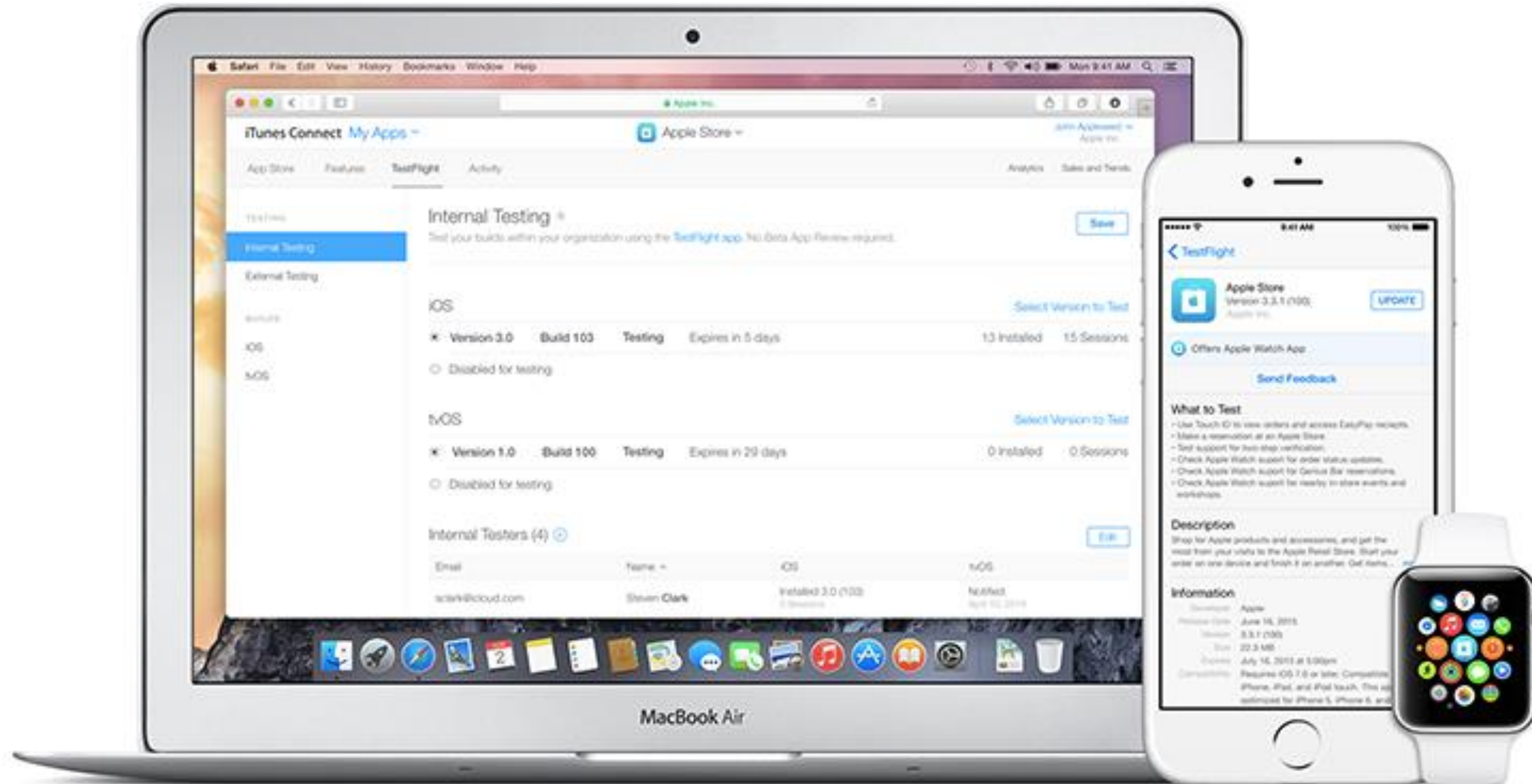
Un programme Developer de qualité

99\$ de souscription annuelle

Inscription pas obligatoire si pas de publication de l'application



App Analytics

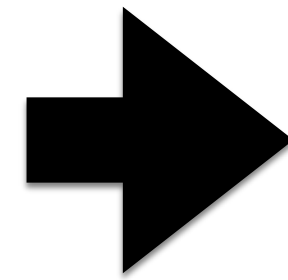


TestFlight

Inviter jusqu'à 2000 personnes pour les beta tests



Objective-C



Objective-C à Swift

Objective-C

- Wrapper de C
- Orienté Objet
- Fortement typé
- Très verbeux
- Basé sur de l'envoi de messages
- Pas d'héritage multiple



Objective-C

```
NSInteger i = 123;  
NSString *s = @"Hello world";
```

Fortement typé, utilisation de pointeurs



```
@interface Speaker : NSObject

// Properties
@property (nonatomic, strong) NSString *firstname;
@property (nonatomic, strong) NSString *lastname;

// Methods declaration
- (void)sayHello;

@end
```

```
#import "Speaker.h"

@implementation Speaker

- (void)sayHello
{
    NSLog(@"Hello, my name is %@ %@", _firstname,
        _lastname);
}

@end
```

Utilisation de headers (.h)

Comme en C, les .h permettent d'exposer la partie accessible depuis l'extérieur de la classe.
Les .m correspondent à l'implémentation.

Objective-C

```
// Speaker objects
Speaker *speaker1 = [[Speaker alloc] init];
Speaker *speaker2 = [[Speaker alloc] init];

// Simple, no argument
[speaker1 sayHello];

// Single argument
[speaker1 say:@"Hello"];

// Multiple argument
[speaker1 say:@"Hello" to:@"students"];

// Nested messages
[speaker1 say:@"Hello" to:[speaker2 fullName]];
```

Java

```
// Speaker objects
Speaker speaker1 = new Speaker();
Speaker speaker2 = new Speaker();

// Simple, no argument
speaker1.sayHello();

// Single argument
speaker1.say("Hello");

// Multiple argument
speaker1.say("Hello", "students");

// Nested messages
speaker1.say("Hello", speaker2.getFullName());
```

Les messages

A rapprocher à un appel à une méthode en Java

A noter

- Les Selectors : pointeur sur des fonctions
- Les Properties : Permet de déclarer des variables en simplifiant l'écriture des getter/setter
- Les Protocoles : équivalent aux interfaces en Java

Swift

- Introduit par Apple en 2014
- Langage Open source depuis 2015
- Jusqu'à 93 fois plus rapide qu'Objective-C
- Garde le même runtime qu'Objective-C



Ce qui ne change pas

- Cocoa Touch reste l'implémentation de l'UI
- Le gestionnaire des références est inchangé (ARC)
- Runtime identique

Quoi de neuf dans Swift ?

- Grande simplification de la syntaxe
- Plus de header (.h)
- Ajout des namespace
- Le ; à la fin des instructions devient optionnel
- Les pointeurs ne sont plus exposés
- Initialisation automatique

Déclaration de variables

Utilisation de let pour les constantes

```
let uneChaine = "Une valeur littérale de chaine"
```

Utilisation de var pour les variables

```
// 2 chaîne équivalente  
var chaineVide = "" // littéral vide  
var autreChaineVide = String() // syntaxe initializer
```

Tous les caractères unicode sont utilisables

```
let unusualMenagerie = "Koala 🐨, Snail 🐌, Penguin 🐧, Dromedary 🐪"
```

Les types optionnels

- Permet de définir un type qui contient, ou pas, une valeur

```
// Défini un dictionnaire prénom/âge
let ages: [String: Int] = ["anne": 19, "jean": 32, "pierre": 24]

// Récupère l'âge de mathieux
let ageMathieux: Int? = ages["mathieux"]

// Si la "boite" contient une valeur
if ageMathieux != nil {
    // On utilise ! pour "ouvrir" la boite
    print("Mathieux a \$(ageMathieux!) ans")
}
else {
    print("L'âge de Mathieux n'est pas défini")
}
```

Les fonctions

```
func foo(p1: String, p2: String, p3: String) {  
    // p1, p2, p3 sont disponibles ici  
}  
  
foo("s1", p2: "s2", p3: "s3")
```

Les paramètres sont nommés

Les classes

- Très simple à déclarer

```
class Personne {  
    var nom: String = "Durand"  
    var prenom: String = "Jeanne"  
    var sexe = "f" let personne = Personne()  
    var age: Int = 24  
    var adresse: String = "94 rue machin"
```

- Et à instancier

```
let personne = Personne()
```

L'héritage

- Attention, comme en Java, on hérite d'une seule classe

```
class Acteur : Personne {  
}
```

Les protocoles

- Equivalent des interfaces en Java

Les énumérations

- Proche du Java ...

```
enum Langage {  
    case Swift  
    case Java  
    case Php  
    case C  
}
```

TP : initiation à Swift

<https://swift.sandbox.bluemix.net>

- Un nombre est dit de Armstrong (ou narcissique) un entier naturel qui est égal à la somme des cubes des chiffres qui le composent.
- Par exemple 153 ($153 = 1^3 + 5^3 + 3^3$) est un nombre de Armstrong.
- Combien de nombre de Armstrong sont compris entre 0 et 10.000 ?
- Pour le clavier : Pomme => Préférences système => Clavier => Méthode de saisie : ajouter « Français – PC »

TP : initiation

- Créer une application composée d'un bouton et d'un label rempli par un nombre, au clic sur le bouton, un nombre aléatoire est généré dans ce champ
- Créer un générateur de mot de passe prenant en paramètre le nombre de caractères à entrer puis les types de caractères acceptés
- Ajouter un indicateur de qualité de mot de passe selon les critères choisis par l'utilisateur
- Pour le clavier : Pomme => Préférences système => Clavier => Méthode de saisie : ajouter « Français – PC »