Développement d’applications mobiles iOS

Plan de formation

<u>**Séance 1 (4h)**</u>

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

<u>Séance 2 (4h)</u>

Capteurs mobiles sur iPhone & iPad

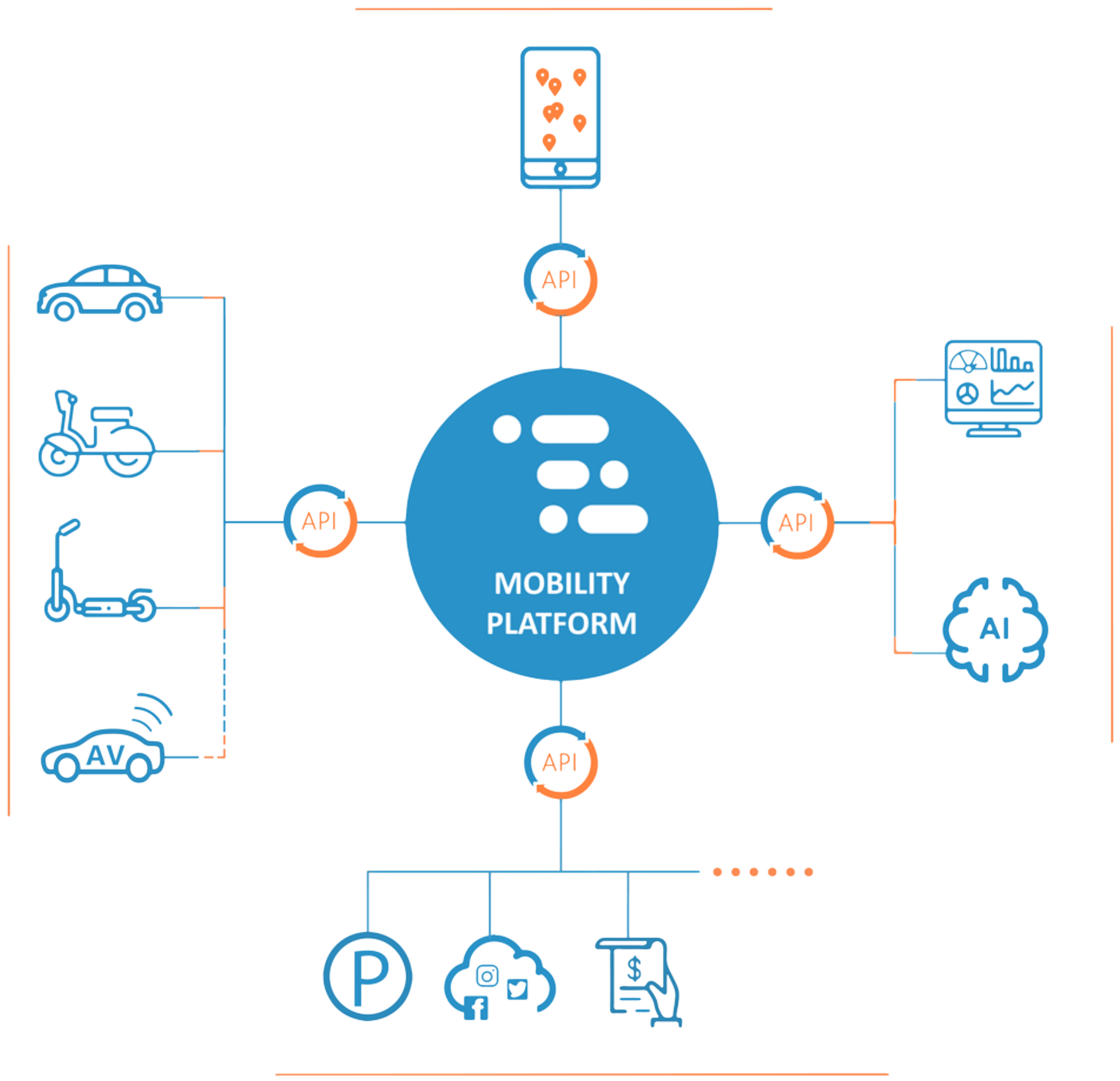
Kinan Arnaout

Mobile apps team leader @ Vulog

karnaout@vulog.com

VULOG

https://www.vulog.com/



VULOG

Clients

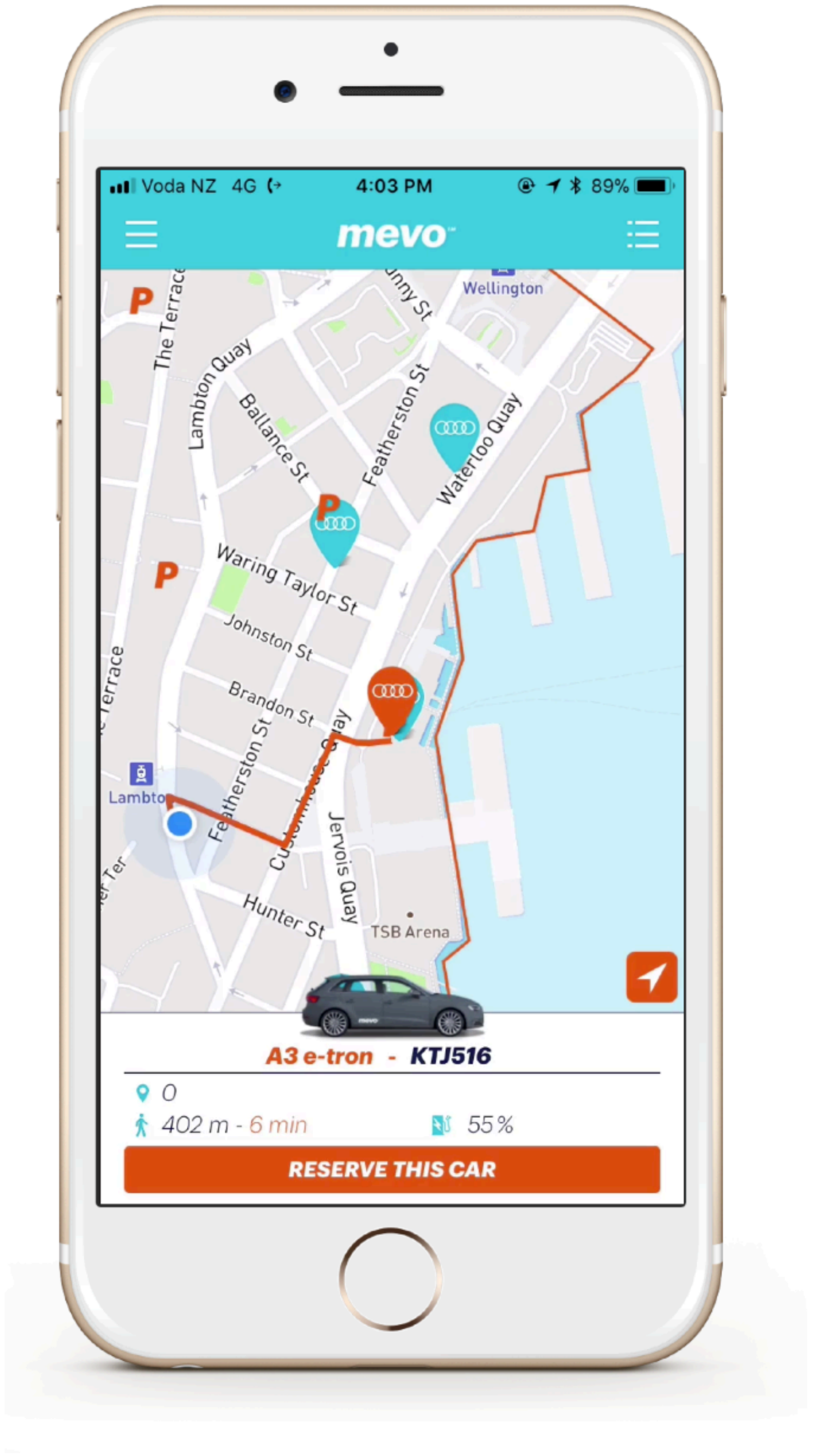
PSA GROUPE
emov
GROUPE RENAULT
wible ve más allá
KIA
POPPY

MOL LIMO
RACV
TROOPY
URBANO
mevo™
GreenMobility® YOUR CITY CAR

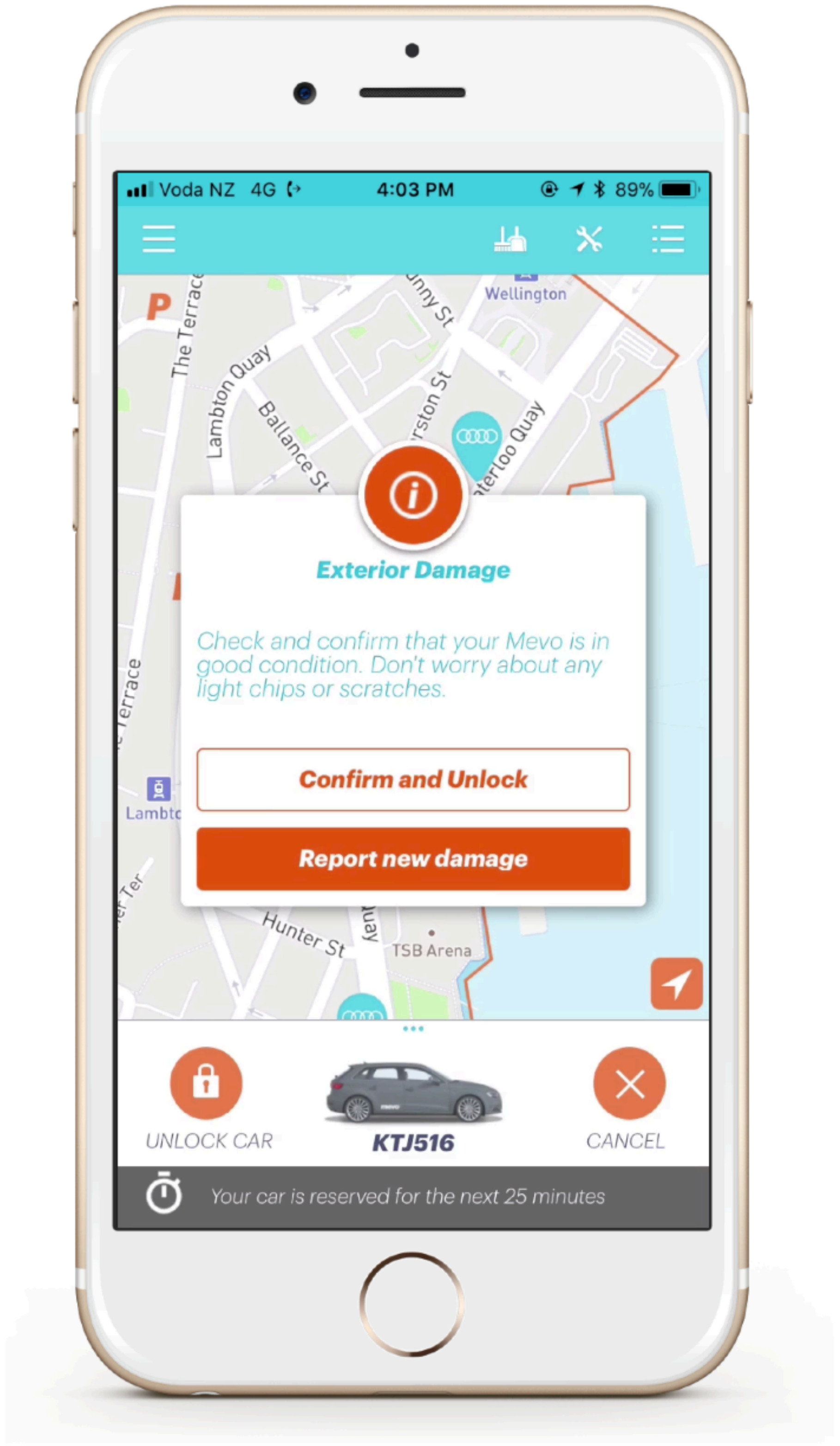
evo™ CAR SHARE
Communauto
transdev
autobleue
MAIRIE DE PARIS
MÉTROPOLE NICE CÔTE D'AZUR

Volkswagen

VULOG

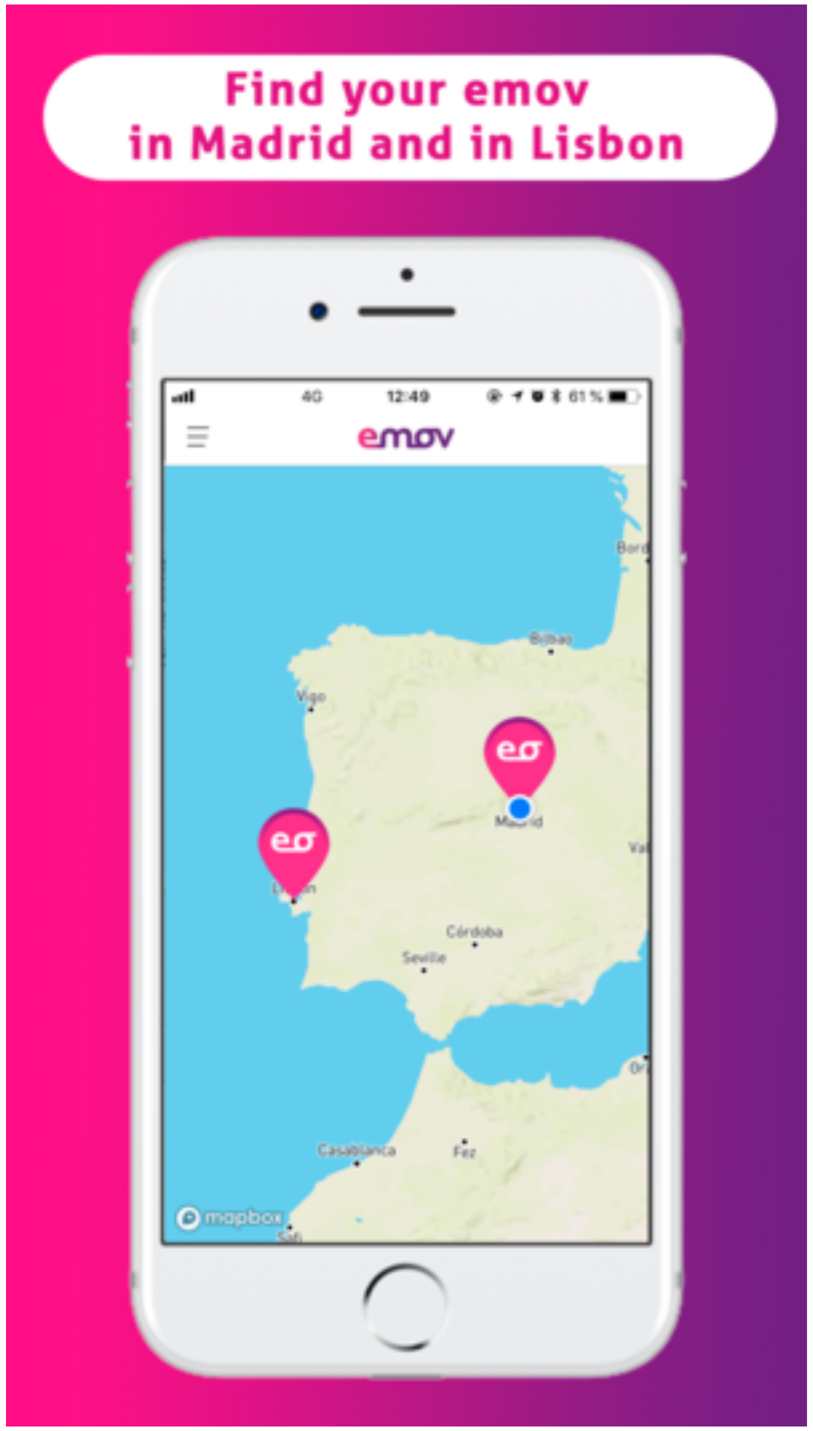


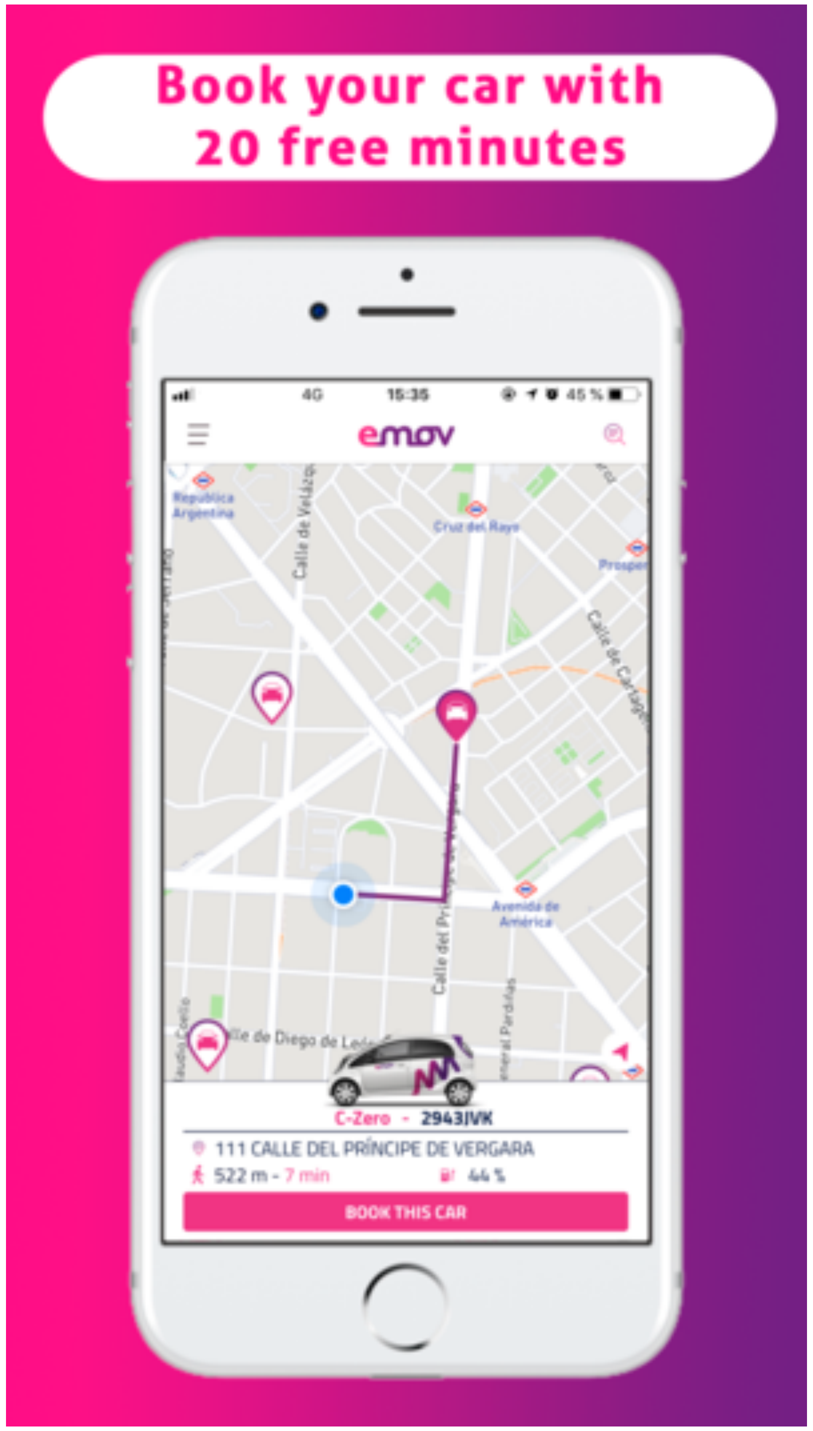
Voda NZ 4G 4:03 PM 89%
mevo
A3 e-tron - KTJ516
0
402 m - 6 min
55%
RESERVE THIS CAR



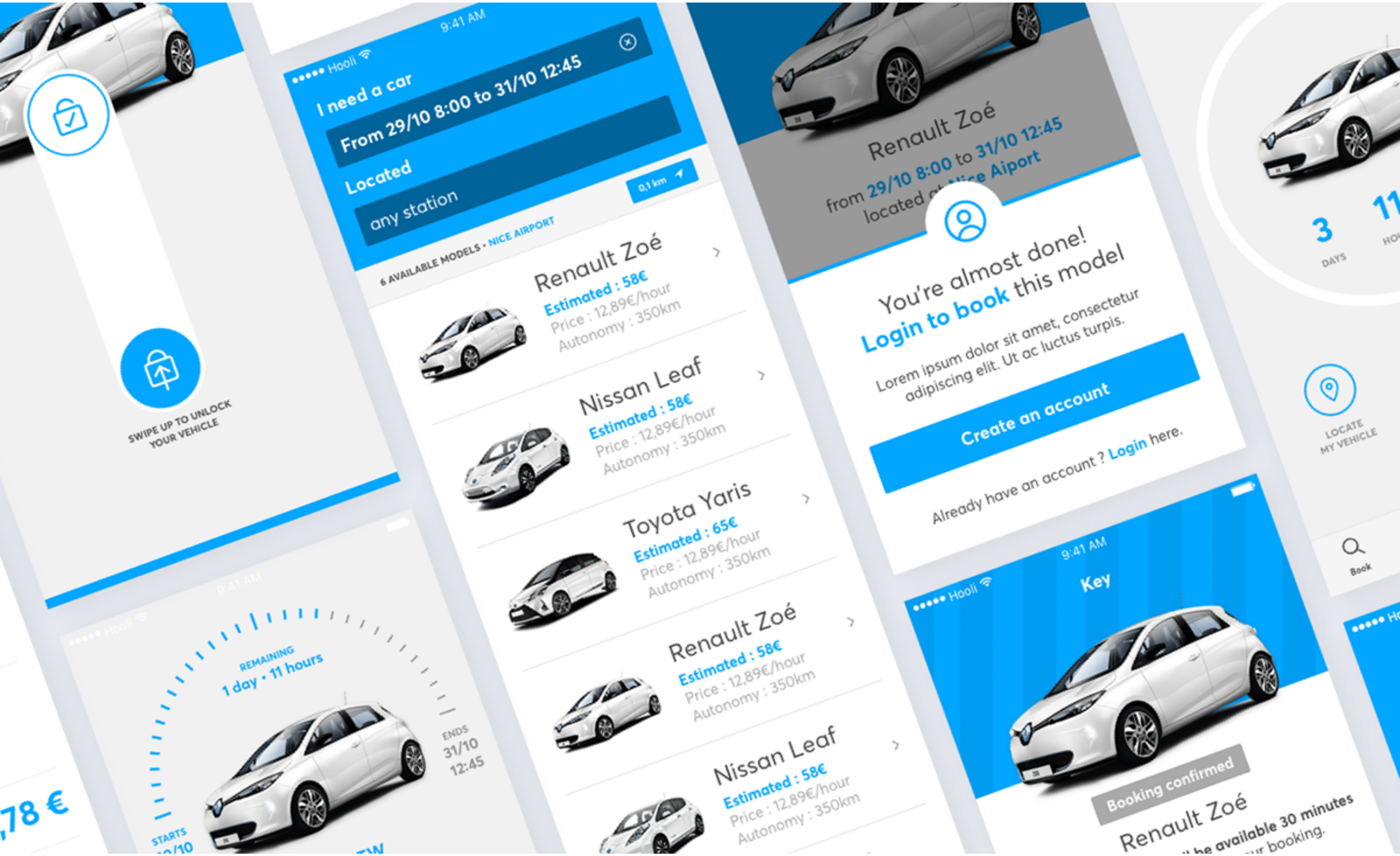
Voda NZ 4G 4:03 PM 89%
Exterior Damage
Check and confirm that your Mevo is in good condition. Don't worry about any light chips or scratches.
Confirm and Unlock
Report new damage
UNLOCK CAR
KTJ516
CANCEL
Your car is reserved for the next 25 minutes

VULOG


Find your emov
in Madrid and in Lisbon
emov
Madrid
Lisbon
Córdoba
Seville
Casablanca
Fez
mapbox


Book your car with
20 free minutes
emov
C-Zero - 2943JVK
111 CALLE DEL PRÍNCIPE DE VERGARA
522 m - 7 min
44 %
BOOK THIS CAR

VULOG



You're almost done!
Login to book this model

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Ut ac luctus turpis.

Already have an account ? **Login** here.

iOS

iPhone

1

3G
3GS

4
4S

5

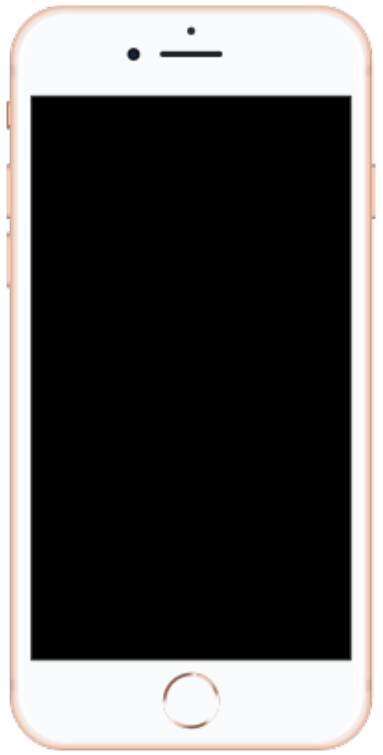
5C

5S

iPhone

6/6S (plus)

7/7 plus



8 / 8 plus

X / XS / XS max / Xr

iPad

Ipad Pro

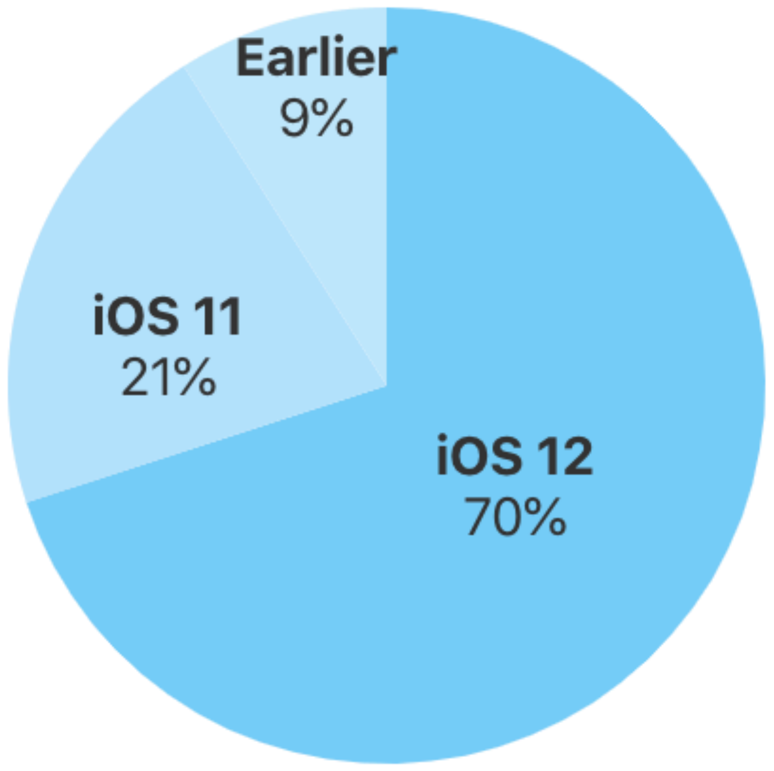
Segmentation du parc

Contrairement à d’autres OS mobiles, le parc n’est pas très fragmenté

Les migrations du parc sont relativement rapides

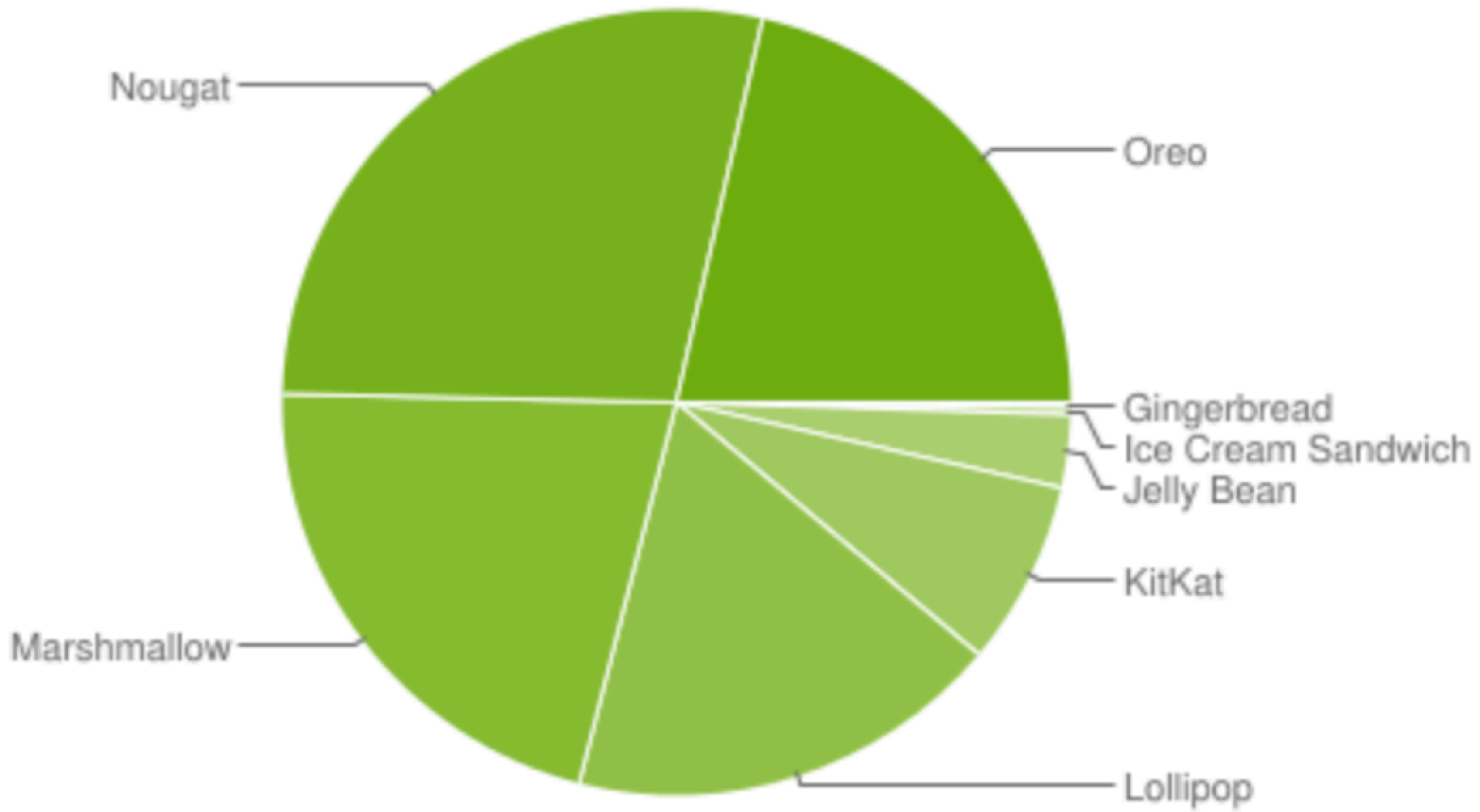
Tout développement se doit de cibler les 2 dernières versions majeures de l’OS

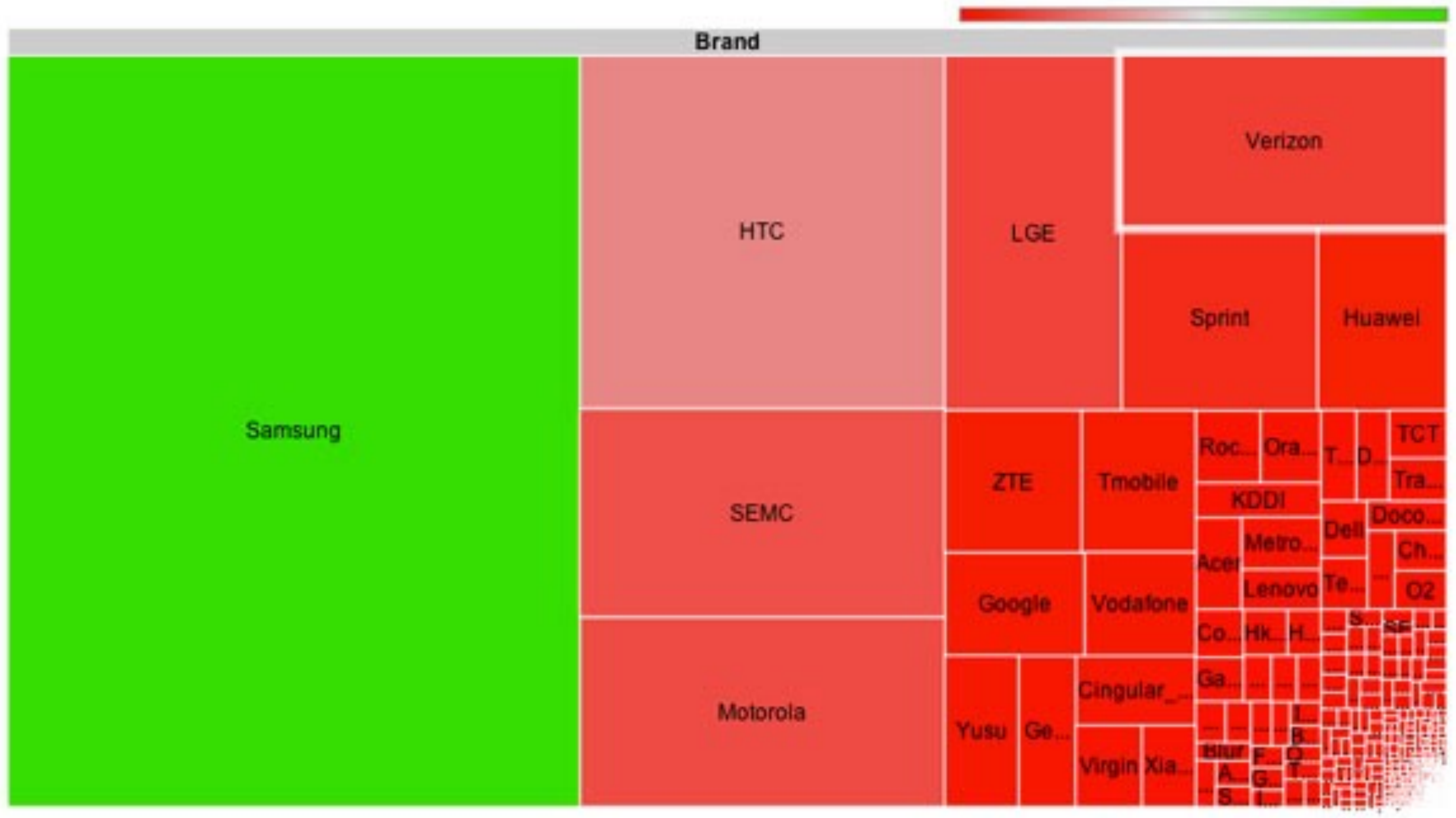
70% of all devices are using iOS 12.



As measured by the App Store on December 3, 2018.

Versus



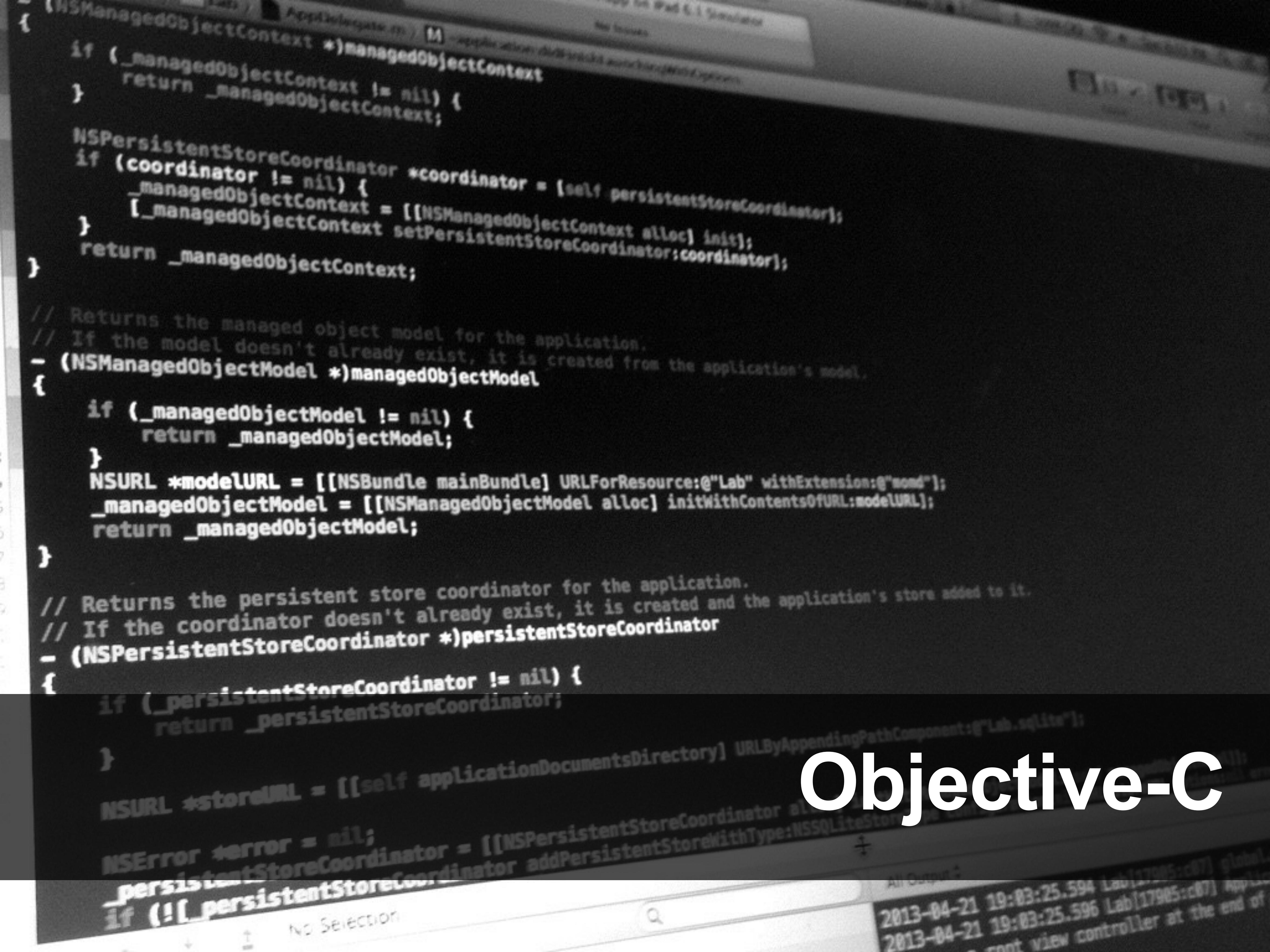


Source: OpenSignal

App Store

> 130.000.000.000
téléchargements

± 2.000.000 applications actives



```
- (NSManagedObjectContext *)managedObjectContext
{
    if (_managedObjectContext != nil) {
        return _managedObjectContext;
    }

    NSPersistentStoreCoordinator *coordinator = [self persistentStoreCoordinator];
    if (coordinator != nil) {
        _managedObjectContext = [[NSManagedObjectContext alloc] init];
        [_managedObjectContext setPersistentStoreCoordinator:coordinator];
    }
    return _managedObjectContext;
}

// Returns the managed object model for the application.
// If the model doesn't already exist, it is created from the application's model.
- (NSManagedObjectModel *)managedObjectModel
{
    if (_managedObjectModel != nil) {
        return _managedObjectModel;
    }
    NSURL *modelURL = [[NSBundle mainBundle] URLForResource:@"Lab" withExtension:@"momd"];
    _managedObjectModel = [[NSManagedObjectModel alloc] initWithContentsOfURL:modelURL];
    return _managedObjectModel;
}

// Returns the persistent store coordinator for the application.
// If the coordinator doesn't already exist, it is created and the application's store added to it.
- (NSPersistentStoreCoordinator *)persistentStoreCoordinator
{
    if (_persistentStoreCoordinator != nil) {
        return _persistentStoreCoordinator;
    }

    NSURL *storeURL = [[self applicationDocumentsDirectory] URLByAppendingPathComponent:@"Lab.sqlite"];

    NSError *error = nil;
    _persistentStoreCoordinator = [[NSPersistentStoreCoordinator al
    if (![_persistentStoreCoordinator addPersistentStoreWithType:NSSQLiteStore
```

```
2013-04-21 19:03:25.594 Lab[17905:c07]
2013-04-21 19:03:25.596 Lab[17905:c07]
```

Objective-C

Avez-vous déjà vu ?

```
// Extrait de la librairie open source AFNetworking

- (AFHTTPRequestOperation *)HTTPRequestOperationWithRequest:(NSURLRequest *)urlRequest
                                                                success:(void (^)(AFHTTPRequestOperation
*operation, id responseObject))success
                                                                failure:(void (^)(AFHTTPRequestOperation
*operation, NSError *error))failure
{
        AFHTTPRequestOperation *operation = nil;

        for (NSString *className in self.registeredHTTPOperationClassNames) {  Class
                operationClass = NSClassFromString(className);
                if (operationClass && [operationClass canProcessRequest:urlRequest]) {
                        operation = [(AFHTTPRequestOperation *)[operationClass alloc] initWithRequest:urlRequest];  break;
                }
        }

        if (!operation) {
                operation = [[AFHTTPRequestOperation alloc] initWithRequest:urlRequest];
        }
        [operation setCompletionBlockWithSuccess:success failure:failure];

        operation.credential = self.defaultCredential;

#ifdef _AFNETWORKING_PIN_SSL_CERTIFICATES_
        operation.SSLPinningMode = self.defaultSSLPinningMode;  #endif
        operation.allowsInvalidSSLCertificate = self.allowsInvalidSSLCertificate;

        return operation;
}
```

Objective-C

Wrapper du langage C

Langage compilé

Dynamique

Envoi de messages à la Smalltalk

Programmation Orientée Objet

Fortement typé

Très verbeux

Typage

PHP

(faiblement typé)

```
$i  =  123;
$s  =  “Hello world!";
```

Java

(fortement typé)

```
int i = 123;
String s = “Hello world!";
```

Objective-C

(fortement typé, pointeurs)

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

Header & implementation

.h

.m

```
@interface Speaker : NSObject

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

// Methods declaration
- (void)sayHello;

@end
```

Speaker.h

```
#import "Speaker.h"

@implementation Speaker

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

@en
d
```

Speaker.m

API publique de la classe

Éléments accessibles depuis l’extérieur

Implémentation des méthodes déclarées dans l’interface

Properties

```
@property (nonatomic, strong)        NSString *lastname;
```

Déclaration des attributs d’une classe

Génération automatique des getters

```
-  (void)whois
{
      NSLog(@"My firstname   is   %@", self.firstname);              //   Getter    via   self.
      NSLog(@"My lastname   is   %@", _lastname);                     //   Getter    via   _
}
```

Génération automatique des setters

```
-  (void)setup
{
      self.firstname       =  @"Cyril";    //   Setter    via   self.
      _lastname    = @"Chandelier";      //   Setter    via   _
}
```

Litéraux

```
[NSNumber numberWithInt:12];
// ou
@12;

[NSNumber numberWithBool:YES];
// ou
@YES;

NSArray *myArray = [NSArray arrayWithObjects:obj1, obj2, obj3, nil];
// ou
NSArray *myArray = @[  obj1, obj2, obj3 ];

id obj = [myArray objectAtIndex:0];
// ou
id obj = myArray[0];

NSDictionary *dic = [NSDictionary dictionaryWithObjectsAndKeys:lastname, @"lastname", firstname, @"firstname",
[NSDate date], @"registrationDate", nil]];
// ou
NSDictionary *dic =
    @{   @"lastname":
    lastname,  @"firstname":
    firstname,
    @"registrationDate": [NSDate date]
    };

id obj = [dic objectForKey:@"lastname"];
// ou
id obj = dic[@"lastname"];
```

Messages

Envoi de commandes aux objets (équivalent aux appel de méthodes)

Messages analysés au runtime par l’objet

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());
```

Protocols

Déclaration de méthodes à implémenter

Très utilisé pour le design pattern “Délégué”

```
@class Speaker;
@protocol SpeakerDelegate <NSObject>

@required

-   (void)speaker:(Speaker *)aSpeaker said:(NSString *)sentence;

@optional
-   (void)speaker:(Speaker *)aSpeaker ask:(NSString *)question;

@end
```

```
@interface Speaker : NSObject <SpeakerDelegate>

@end
```

Categories

Ajout de fonctionnalités à une classe

```
//    Speaker+Utils.h
#import    "Speaker.h"

@interface Speaker (Utils)

-   (NSString *)fullname;

@end
```

```
//    Speaker+Utils.m
#import "Speaker+Utils.h"

@implementation Speaker (Utils)


-   (NSString *)fullname
{
        return [NSString stringWithFormat:@“%@ %@",  self.firstname, self.lastname];
}

@end
```

Blocks

Bout de code exécutable

Paramètres et types de retours

Très utilisés dans les animations

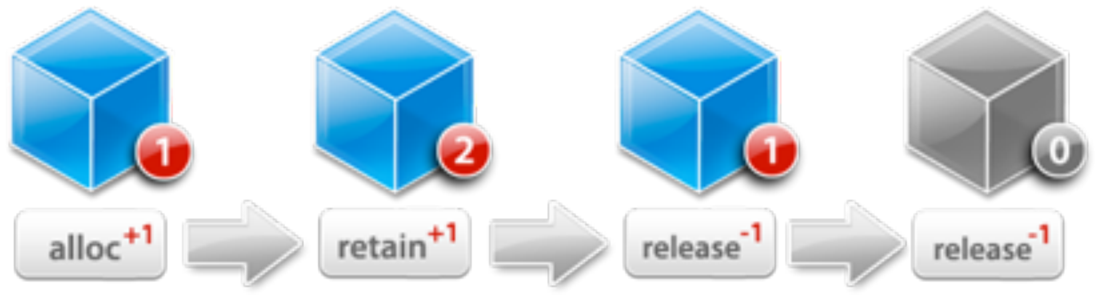
```
// Method with a block as parameter
- (void)doSomethingWithBlock:(void(^)(void))aBlock
{
        aBlock();
}

- (void)iUseBlocks
{
        // Block declaration  void
        (^aBlock)(void) = ^{
                NSLog(@"Hi, i'm a block");
        };

        // Execute it now
        aBlock();

        // Give it to a method
        [self doSomethingWithBlock:aBlock];
}
```

Mémoire

Avant

Gestion de la mémoire de type Reference Couting

- retain
- release
- autorelease



Maintenant

ARC (Automatic Reference Counting)

≠ Garbage Collector

ional patterns
Protocols and extensions on structs
Pattern mat
se syntax
Closures
Generics
Fast ite
e collections
Optional
ator overloading
Object orient
espaces
Tuples
Type infe
Read-Eval-Print-Loop (
mutability syntax
Multiple return types
ctive playground

Swift

“Objective-C without the C”

Introduit durant la WWDC 2014

Présenté comme le successeur de l’Objective-C

Plus rapide (93x) d’aprés des benchmarks réalisé sur des algorithmes complexes de tri et d’encryption

Points communs

Cocoa et Cocoa Touch

Compilateur LLVM

Automatic Reference Counting

Même runtime qu’Objective-C

Principale différences avec Objective-C

Plus besoin de **;** à la fin de chaque instruction

Plus de header (fichiers .h)

Les énumérations peuvent avoir des données associées

(Re)-définition d’opérateurs

Closures

Namespaces

Designed for safety

Les pointeurs ne sont plus exposés

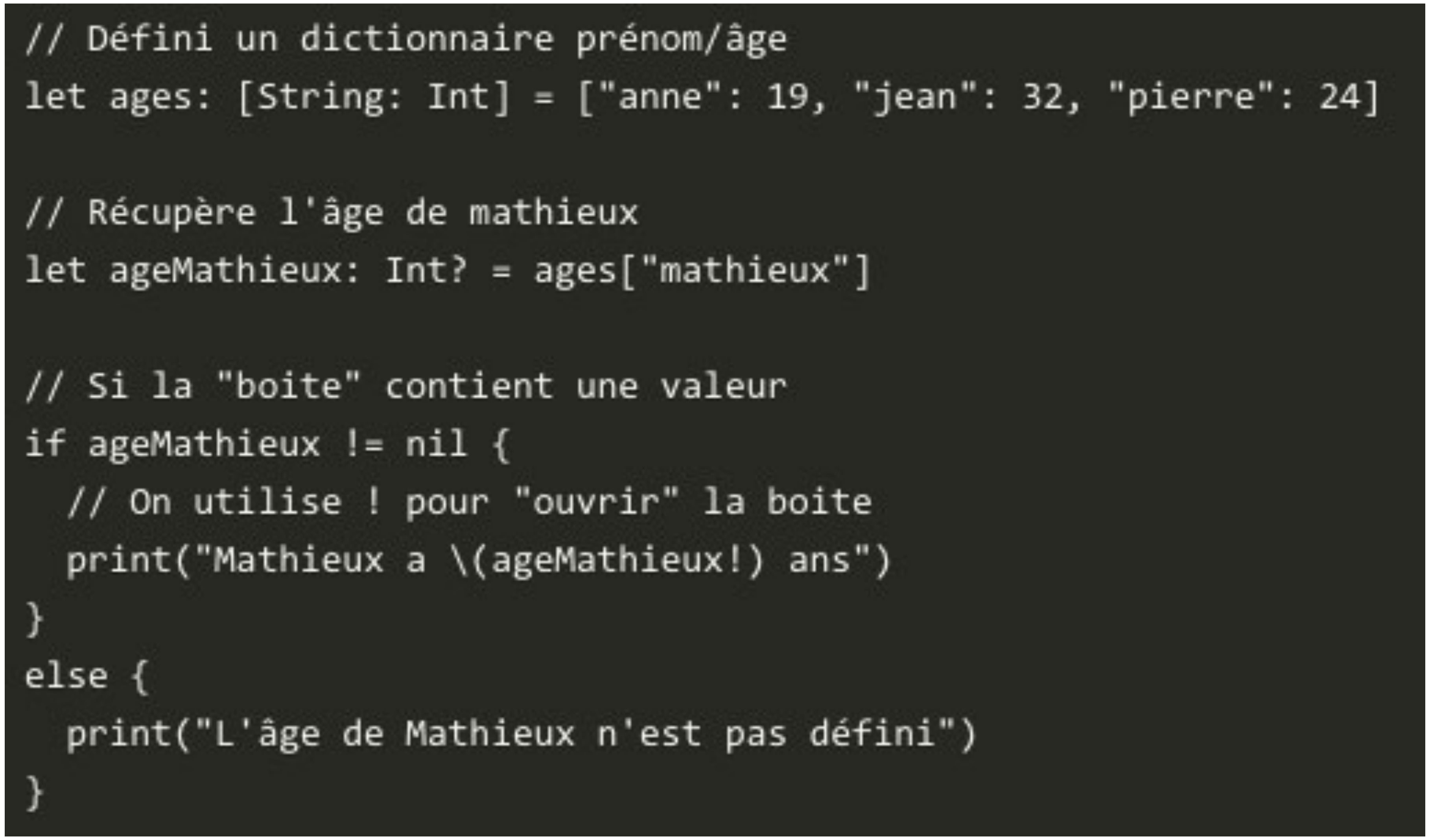
Chaque case d’un switch est terminal

Les variables et les constantes sont toujours initialisées

Le typage est indispensable et contraignant pour faire attention aux algorithmes développés

Optionals

- 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")
}
```

Optionals

```
func cellDidEditTaskContent(cell: TodoCell, newContent: String)
{
      //    Retrieve     index    path
      let  indexPath = self.tableView.indexPathForCell(cell)  if
      (indexPath != nil)
     {
            //    Update   related     task    and  save
              var    task    = self.frc!.objectAtIndexPath(indexPath!) as Task

            //    Update   task
            TaskController.sharedInstance.updateTask(task, content: newContent)
      }
}
```

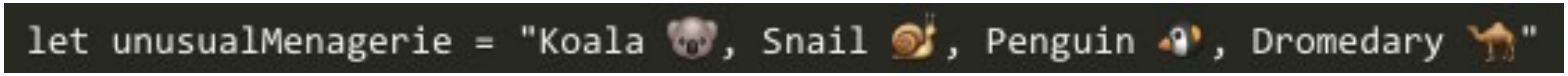
Constantes et variables

```
//    Constant    >  immutable
let  aConstant  = 10

//    Variable    >  mutable
var aVariable   = 10
```

Swift met un point d'honneur a déclarer ses variables avec la mutabilité qui y convient le plus

Il est conseillé de toujours déclarer ses variables en tant que constante, puis de mettre à jour le type si nécessaire uniquement

Tous les caractères unicode sont utilisables



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

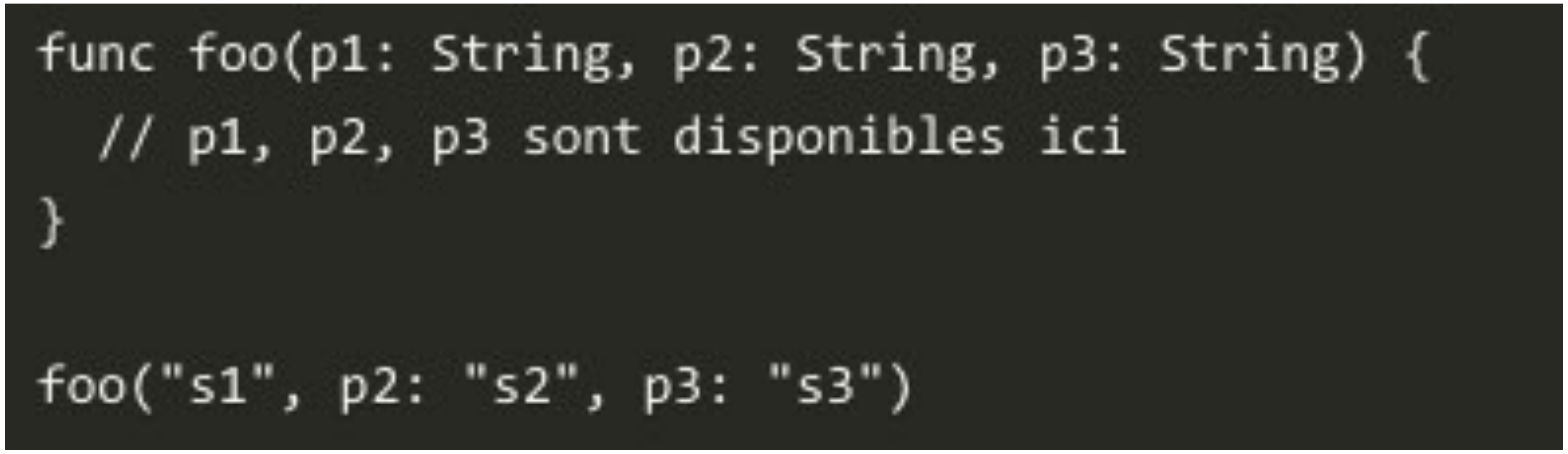
Type Safety & Type Inference

```
let pi = 3.14159
// pi is inferred to be of type Double

let pi : Double = 3.14159
// No type inference
```

Le type des variables et des constantes est "deviné" par le compilateur

Une fois le type attribué, une variable ne peut pas changer de type

Les fonctions

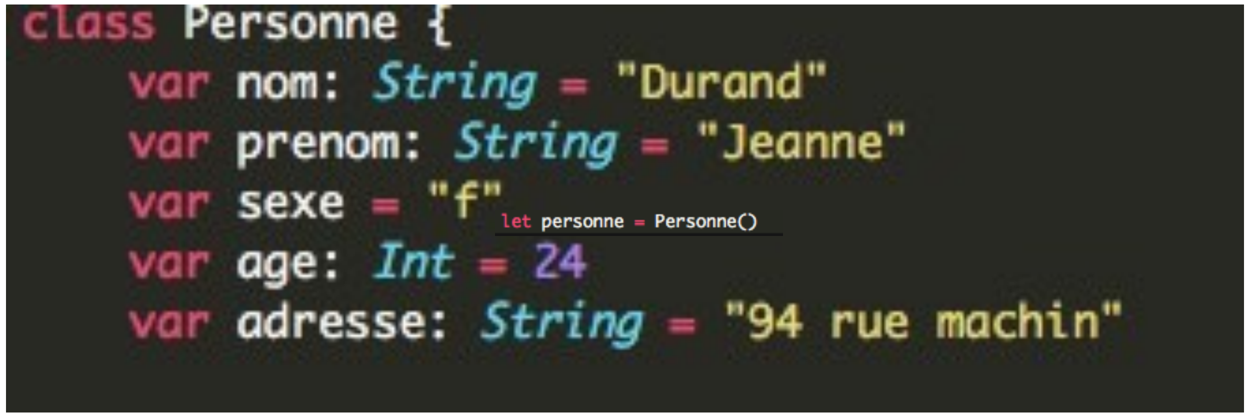


```
func foo(p1: String, p2: String, p3: String) {
  // p1, p2, p3 sont disponibles ici
}

foo("s1", p2: "s2", p3: "s3")
```

Les paramètres sont nommés

Les classes

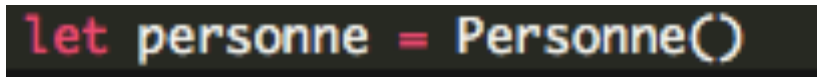
- Très simple à déclarer



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

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

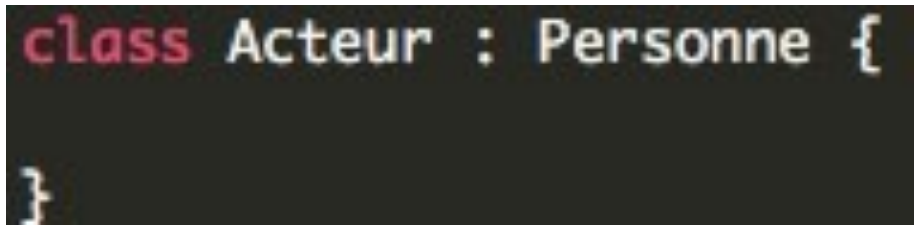
- Et à instancier



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

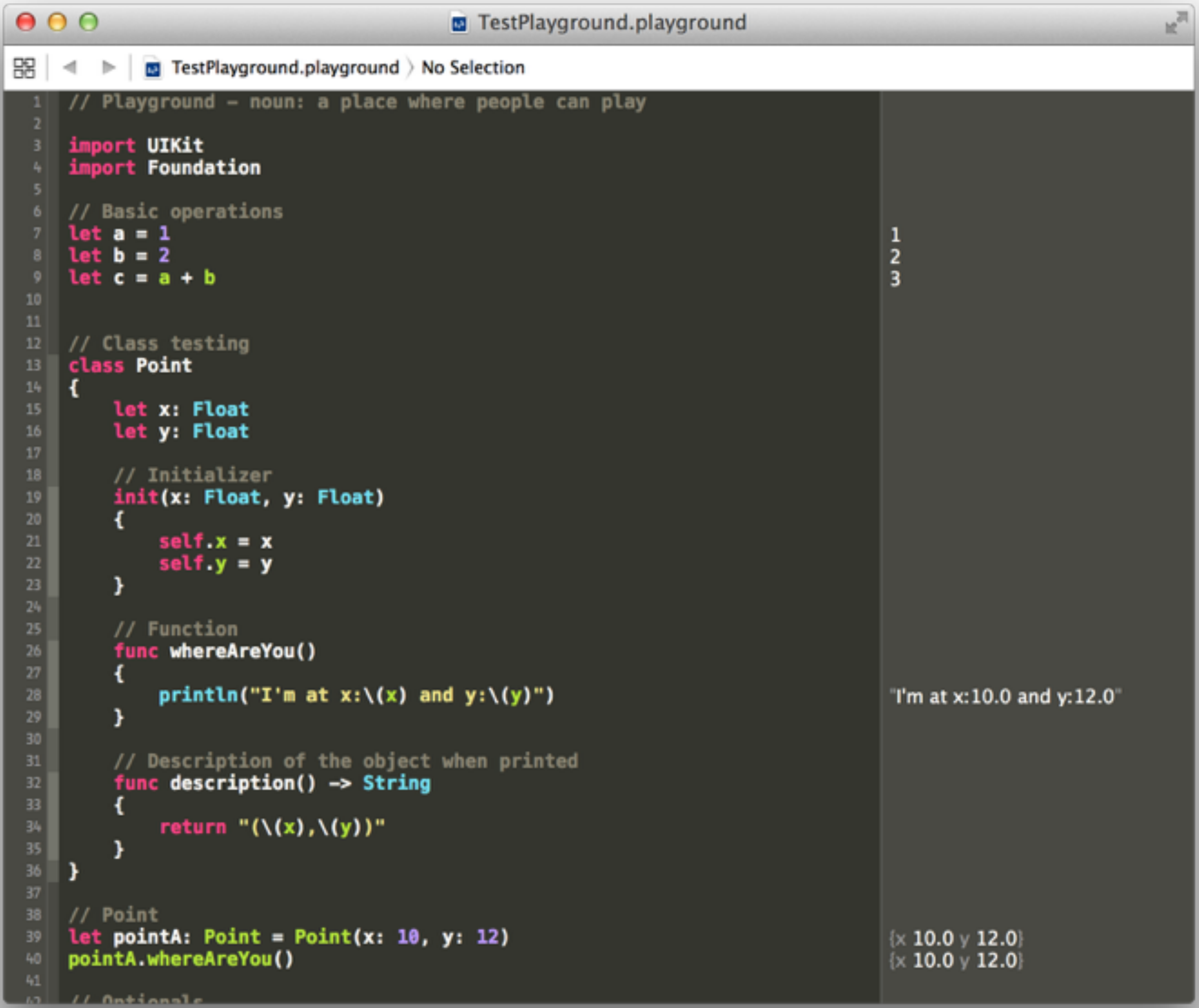
L’héritage

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



```
class Acteur : Personne {

}
```

Playground



TestPlayground.playground

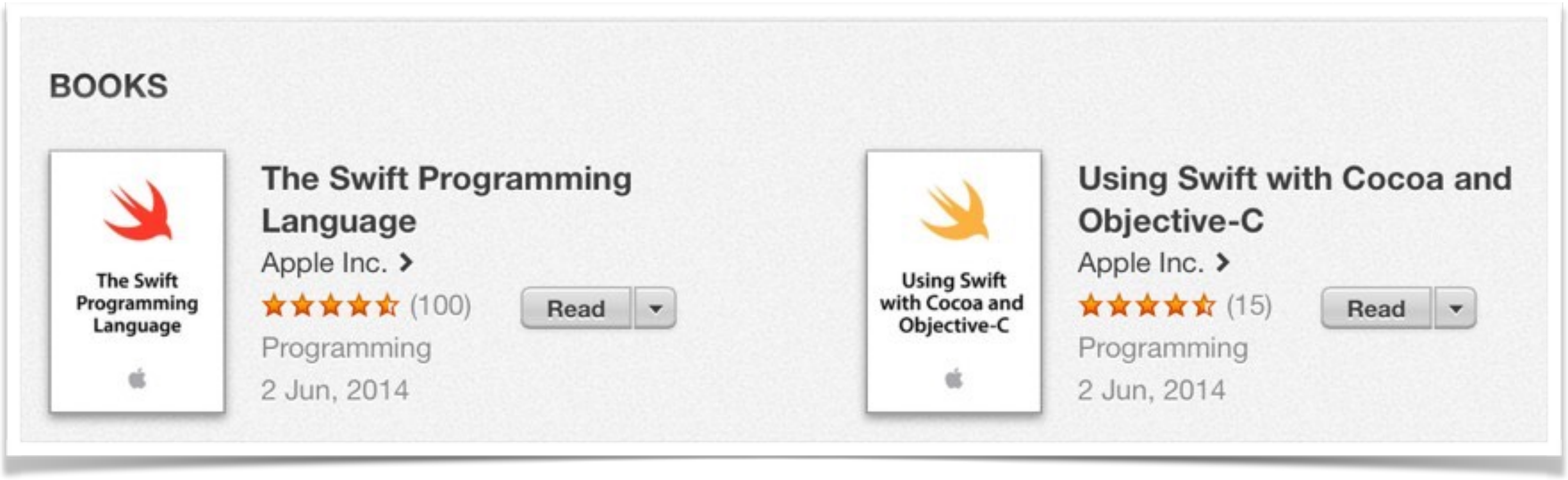
TestPlayground.playground 〉 No Selection

```
// Playground - noun: a place where people can play

import UIKit
import Foundation

// Basic operations
let a = 1
let b = 2
let c = a + b


// Class testing
class Point
{
    let x: Float
    let y: Float

    // Initializer
    init(x: Float, y: Float)
    {
        self.x = x
        self.y = y
    }

    // Function
    func whereAreYou()
    {
        println("I'm at x:\(x) and y:\(y)")
    }

    // Description of the object when printed
    func description() -> String
    {
        return "(\(x),\(y))"
    }
}

// Point
let pointA: Point = Point(x: 10, y: 12)
pointA.whereAreYou()
```

```
1
2
3

"I'm at x:10.0 and y:12.0"

{x 10.0 y 12.0}
{x 10.0 y 12.0}
```

En savoir plus

2 ebooks gratuits, écrits et distribués par Apple



BOOKS

The Swift Programming Language
Apple Inc. ›
★★★★☆ (100)
Programming
2 Jun, 2014

Using Swift with Cocoa and Objective-C
Apple Inc. ›
★★★★☆ (15)
Programming
2 Jun, 2014

The Swift Programming Language
Read
Using Swift with Cocoa and Objective-C
Read

Outils de développement

Get a Mac

Macbook

Macbook Pro

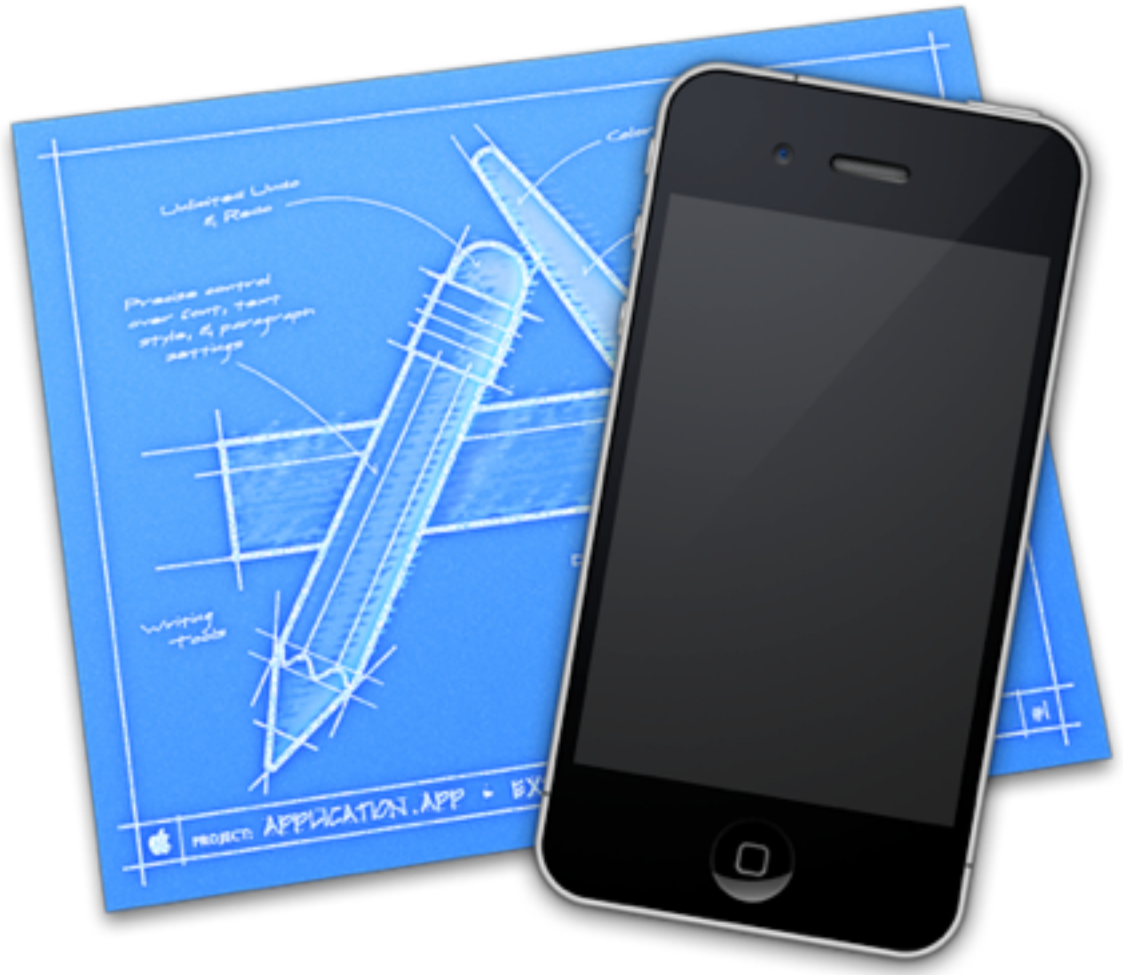
iMac

Mac Pro

Mac Mini

Xcode

Simulateur iPhone & iPad



Programme développeur

$99 / 80€
Valable pendant **1 an** (renouvelable)

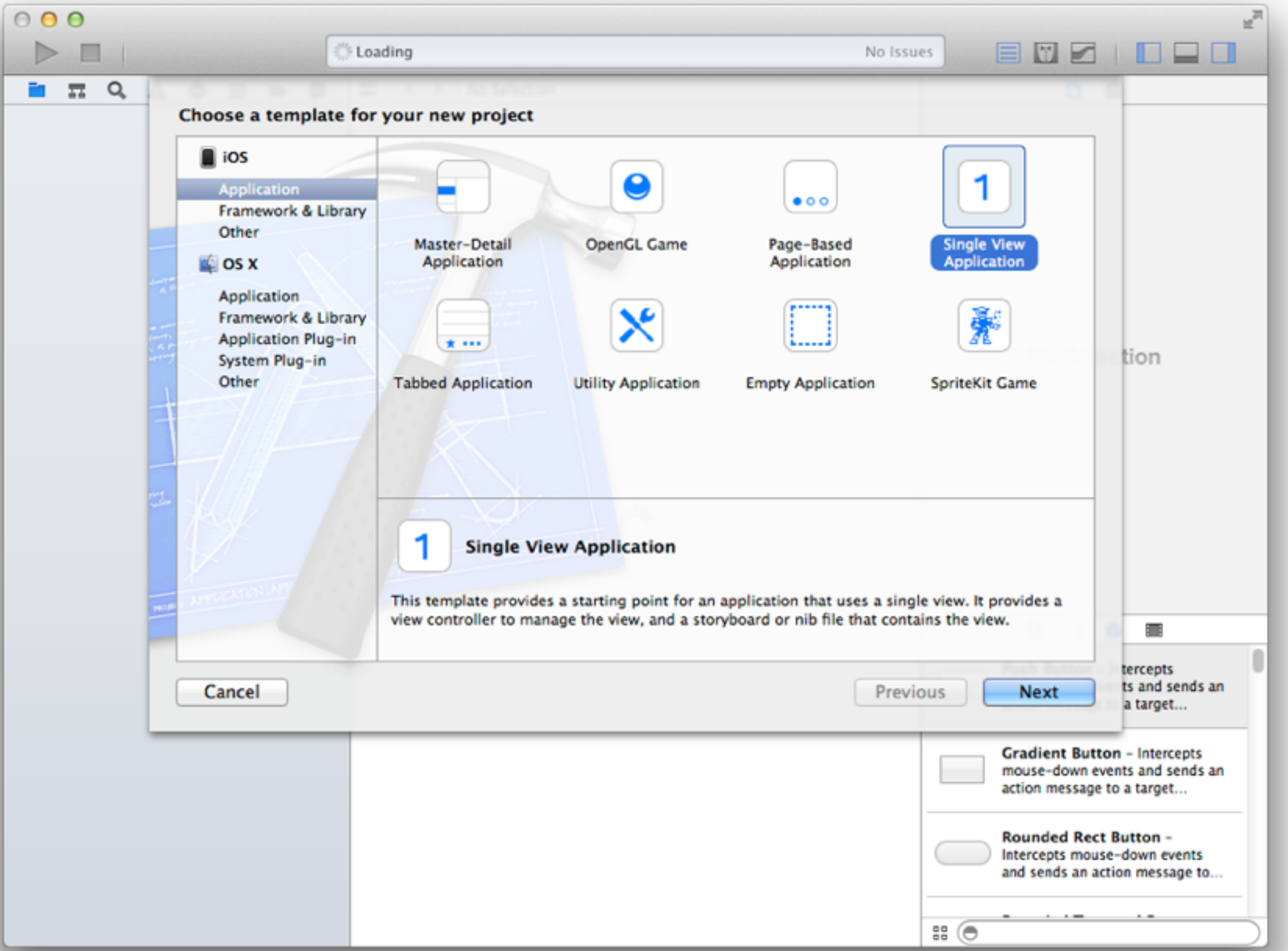
Applications <u>illimitées</u> sur l’App Store

Indispensable pour déployer sur device

iOS Developer
Program

Premier projet

Création à partir d'un template



Choose a template for your new project

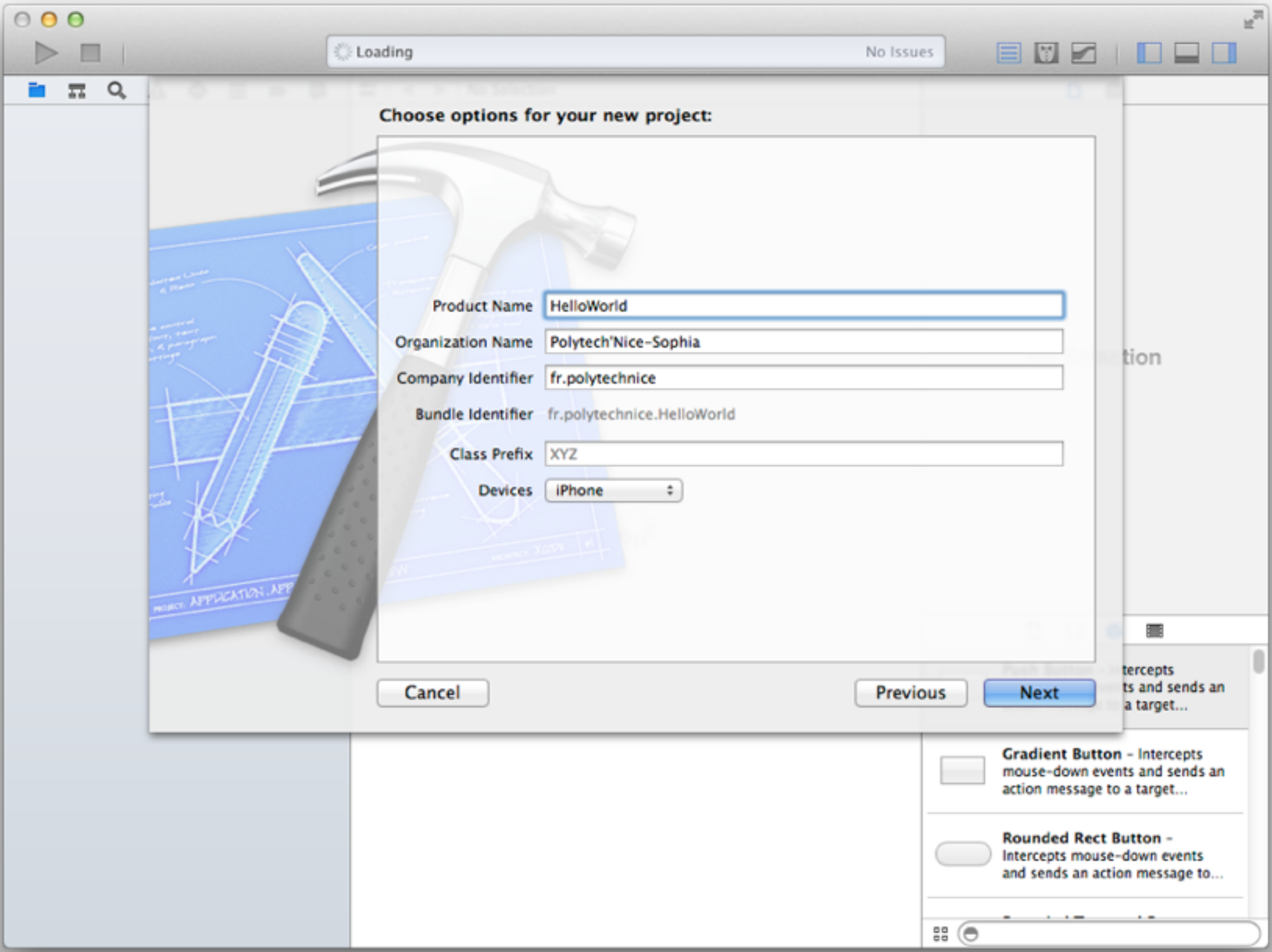
Single View Application

This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a storyboard or nib file that contains the view.

Gradient Button - Intercepts mouse-down events and sends an action message to a target...

Rounded Rect Button - Intercepts mouse-down events and sends an action message to...

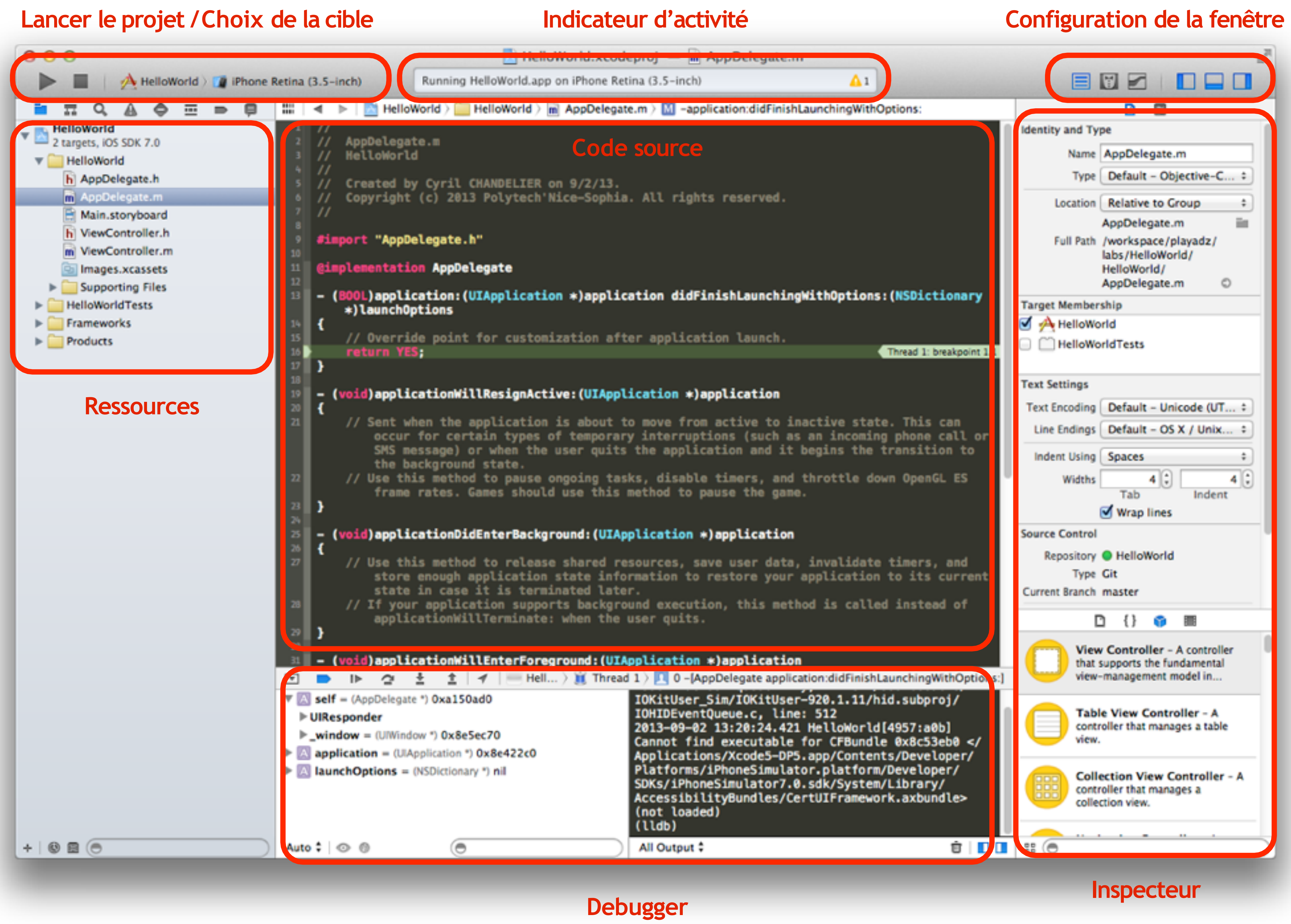
Informations générales



Choose options for your new project:

Product Name	HelloWorld
Organization Name	Polytech'Nice-Sophia
Company Identifier	fr.polytechnice
Bundle Identifier	fr.polytechnice.HelloWorld
Class Prefix	XYZ
Devices	iPhone

Loading
No Issues
Cancel
Previous
Next
Gradient Button - Intercepts mouse-down events and sends an action message to a target...
Rounded Rect Button - Intercepts mouse-down events and sends an action message to...


Lancer le projet / Choix de la cible
Indicateur d'activité
Configuration de la fenêtre
HelloWorld
iPhone Retina (3.5-inch)
Running HelloWorld.app on iPhone Retina (3.5-inch)
HelloWorld.xcodeproj — AppDelegate.m
HelloWorld > HelloWorld > AppDelegate.m > -application:didFinishLaunchingWithOptions:
HelloWorld
2 targets, iOS SDK 7.0
HelloWorld
AppDelegate.h
AppDelegate.m
Main.storyboard
ViewController.h
ViewController.m
Images.xcassets
Supporting Files
HelloWorldTests
Frameworks
Products
Ressources
Code source
Thread 1: breakpoint 1
Hell... > Thread 1 > 0 -[AppDelegate application:didFinishLaunchingWithOptions:]
Auto
All Output
Debugger
Inspecteur

```
//
//  AppDelegate.m
//  HelloWorld
//
//  Created by Cyril CHANDELIER on 9/2/13.
//  Copyright (c) 2013 Polytech'Nice-Sophia. All rights reserved.
//

#import "AppDelegate.h"

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary
    *)launchOptions
{
    // Override point for customization after application launch.
    return YES;
}

- (void)applicationWillResignActive:(UIApplication *)application
{
    // Sent when the application is about to move from active to inactive state. This can
       occur for certain types of temporary interruptions (such as an incoming phone call or
       SMS message) or when the user quits the application and it begins the transition to
       the background state.
    // Use this method to pause ongoing tasks, disable timers, and throttle down OpenGL ES
       frame rates. Games should use this method to pause the game.
}

- (void)applicationDidEnterBackground:(UIApplication *)application
{
    // Use this method to release shared resources, save user data, invalidate timers, and
       store enough application state information to restore your application to its current
       state in case it is terminated later.
    // If your application supports background execution, this method is called instead of
       applicationWillTerminate: when the user quits.
}

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

```
▼ self = (AppDelegate *) 0xa150ad0
  ▶ UIResponder
  ▶ _window = (UIWindow *) 0x8e5ec70
▶ application = (UIApplication *) 0x8e422c0
▶ launchOptions = (NSDictionary *) nil
```

```
IOKitUser_Sim/IOKitUser-920.1.11/hid.subproj/
IOHIDEventQueue.c, line: 512
2013-09-02 13:20:24.421 HelloWorld[4957:a0b]
Cannot find executable for CFBundle 0x8c53eb0 </
Applications/Xcode5-DP5.app/Contents/Developer/
Platforms/iPhoneSimulator.platform/Developer/
SDKs/iPhoneSimulator7.0.sdk/System/Library/
AccessibilityBundles/CertUIFramework.axbundle>
(not loaded)
(lldb)
```

Identity and Type

Name: AppDelegate.m
Type: Default - Objective-C...
Location: Relative to Group
AppDelegate.m
Full Path: /workspace/playadz/labs/HelloWorld/HelloWorld/AppDelegate.m

Target Membership

☑ HelloWorld
☐ HelloWorldTests

Text Settings

Text Encoding: Default - Unicode (UT...
Line Endings: Default - OS X / Unix...
Indent Using: Spaces
Widths: 4 (Tab) 4 (Indent)
☑ Wrap lines

Source Control

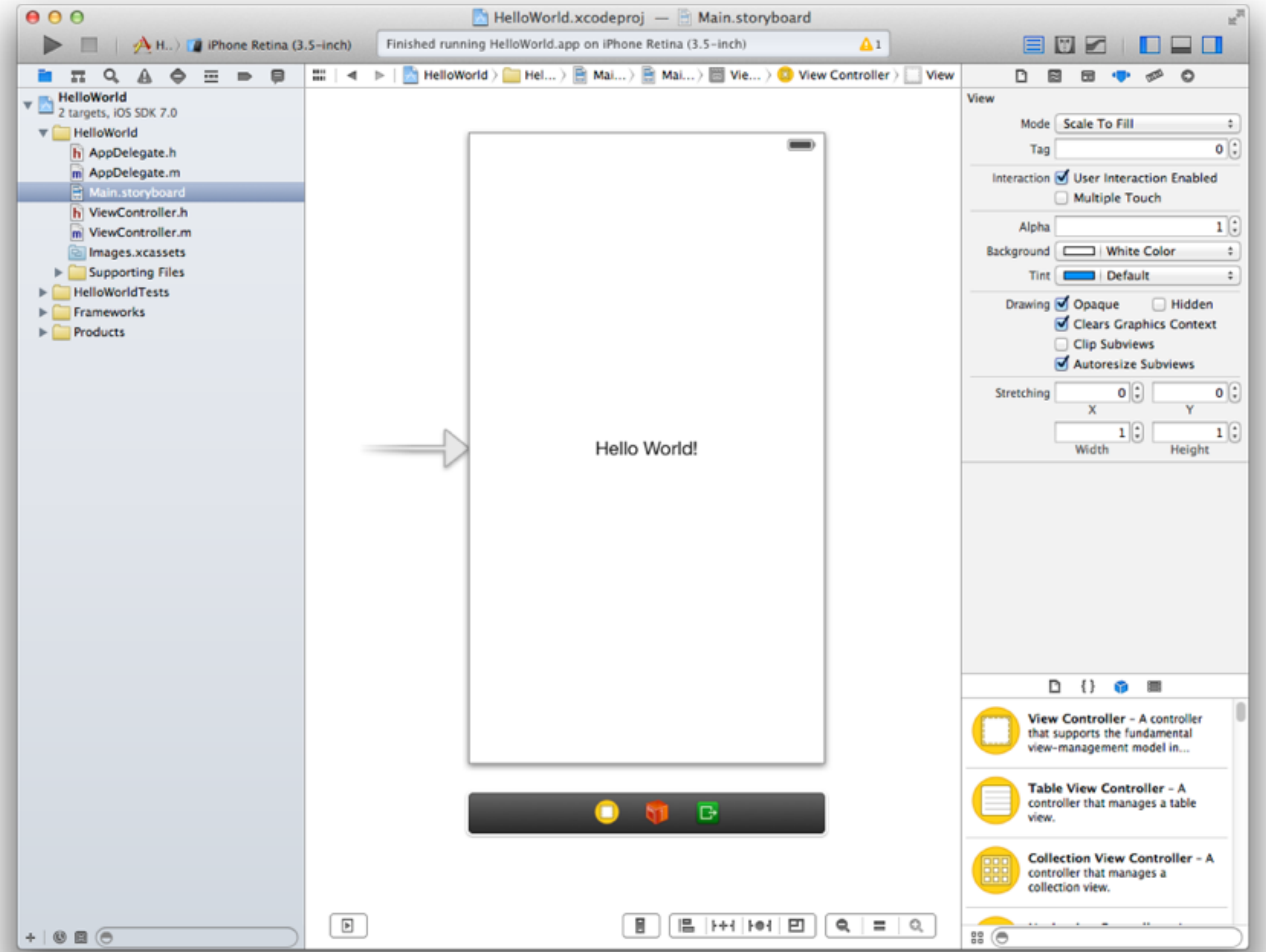
Repository: ● HelloWorld
Type: Git
Current Branch: master

View Controller - A controller that supports the fundamental view-management model in...

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection view.

UI



HelloWorld.xcodeproj — Main.storyboard

Finished running HelloWorld.app on iPhone Retina (3.5-inch)

HelloWorld
2 targets, iOS SDK 7.0
- HelloWorld
  - AppDelegate.h
  - AppDelegate.m
  - Main.storyboard
  - ViewController.h
  - ViewController.m
  - Images.xcassets
  - Supporting Files
- HelloWorldTests
- Frameworks
- Products

HelloWorld › Hel... › Mai... › Mai... › Vie... › View Controller › View

Hello World!

View

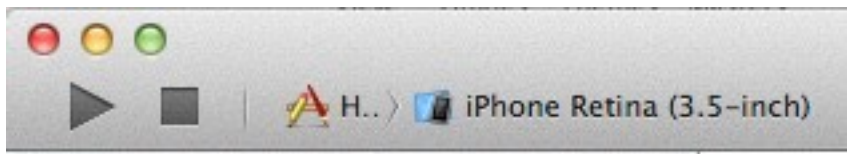
Mode: Scale To Fill
Tag: 0
Interaction: ✓ User Interaction Enabled; Multiple Touch
Alpha: 1
Background: White Color
Tint: Default
Drawing: ✓ Opaque; Hidden; ✓ Clears Graphics Context; Clip Subviews; ✓ Autoresize Subviews
Stretching: X 0, Y 0, Width 1, Height 1

View Controller - A controller that supports the fundamental view-management model in...

Table View Controller - A controller that manages a table view.

Collection View Controller - A controller that manages a collection view.

Lancement de l'application

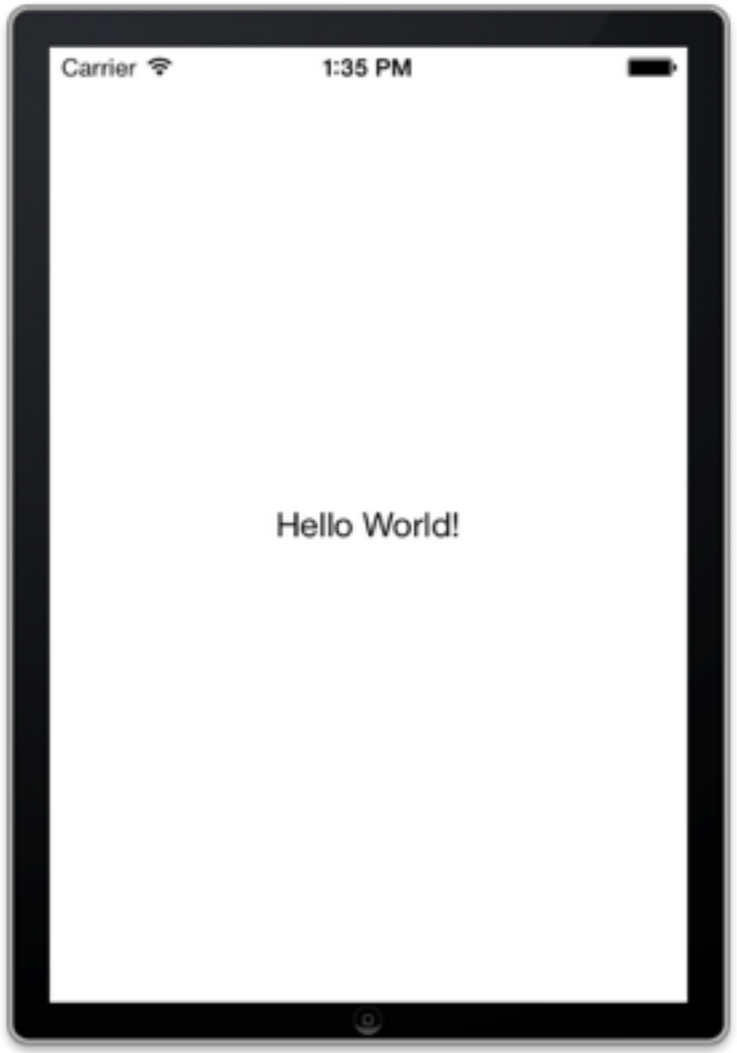


Build and Run

Compilation du projet et des dépendances

Ouverture du simulateur

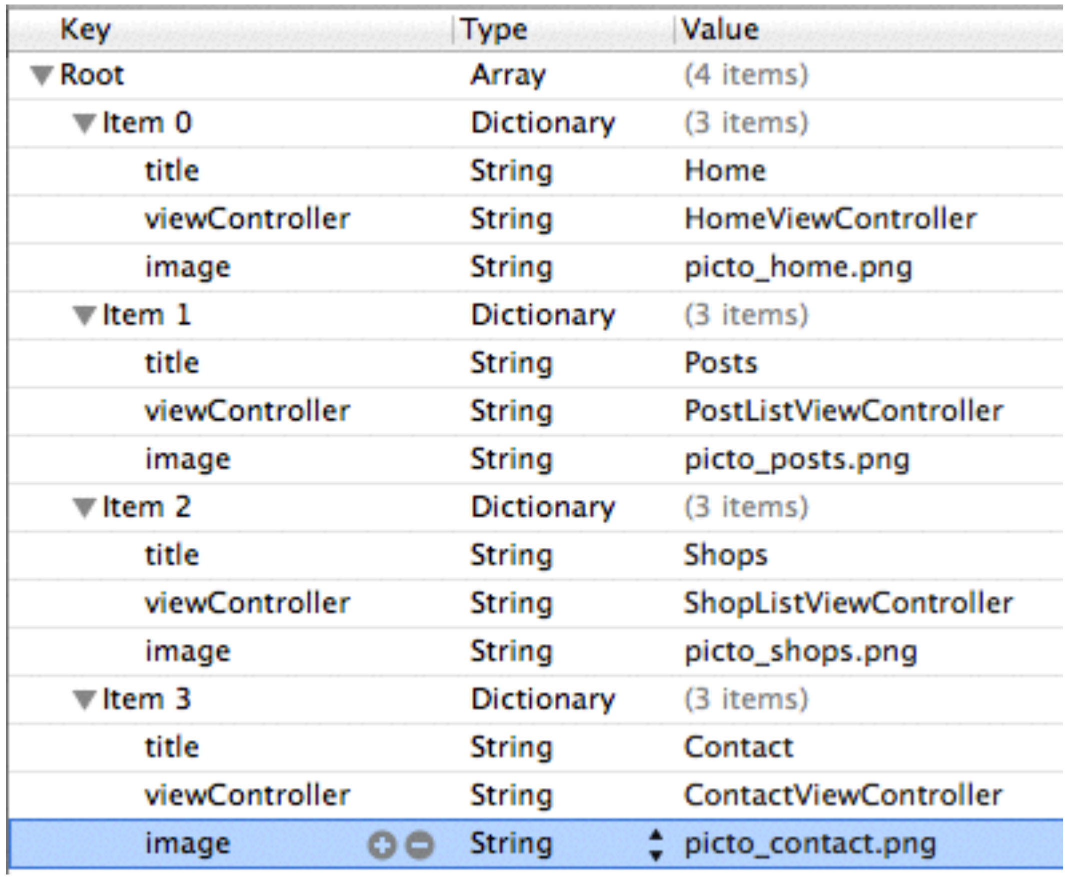
Premier projet lancé!



Persistance des données

PLIST

Sauvegarde de données dans un fichier sur le disque



Key	Type	Value
▼Root	Array	(4 items)
▼Item 0	Dictionary	(3 items)
title	String	Home
viewController	String	HomeViewController
image	String	picto_home.png
▼Item 1	Dictionary	(3 items)
title	String	Posts
viewController	String	PostListViewController
image	String	picto_posts.png
▼Item 2	Dictionary	(3 items)
title	String	Shops
viewController	String	ShopListViewController
image	String	picto_shops.png
▼Item 3	Dictionary	(3 items)
title	String	Contact
viewController	String	ContactViewController
image	String	picto_contact.png

```
// Get saved data
NSBundle *mainBundle = [NSBundle mainBundle];
NSString *filePath = [mainBundle pathForResource:@"Menu" ofType:@"plist"];  NSArray
*menus = [NSArray arrayWithContentsOfFile:filePath];  for(NSDictionary *menu in menus)
{
        // Do something
}
```

```
// Save data
NSArray *arrayToSave = @[ @"A", @"B", @"C" ];  [arrayToSave
writeToFile:@"save.plist" atomically:YES];
```

User Defaults

Sauvegarde des préférences et réglages de l’utilisateur (entre autres)

Gère la sauvegarde d'objets

Gère certains types primitifs (NSInteger, float, double, BOOL)

```
//   User  defaults
NSUserDefaults *userDefaults        = [NSUserDefaults standardUserDefaults];

//   Save  data
[userDefaults      setBool:YES forKey:@"SOUND_ENABLED"];
[userDefaults      synchronize];

//   Get  the  saved  data
BOOL soundEnabled  = [userDefaults boolForKey:@"SOUND_ENABLED"];
if(soundEnabled)
{
      //   Do  something
}
```

Core Data

ORM

Manipulation des données via des objets NSFetchRequest/
NSFetchedResultsController

Stockage des données dans une base de donnée sqlite

iCloud

Synchronisation de donnée dans le cloud

Lié à l’**Apple ID** de l’utilisateur (compte Apple)

Synchronisation instantanée entre les devices

<u>Nouveau :</u> CloudKit (iOS 8)

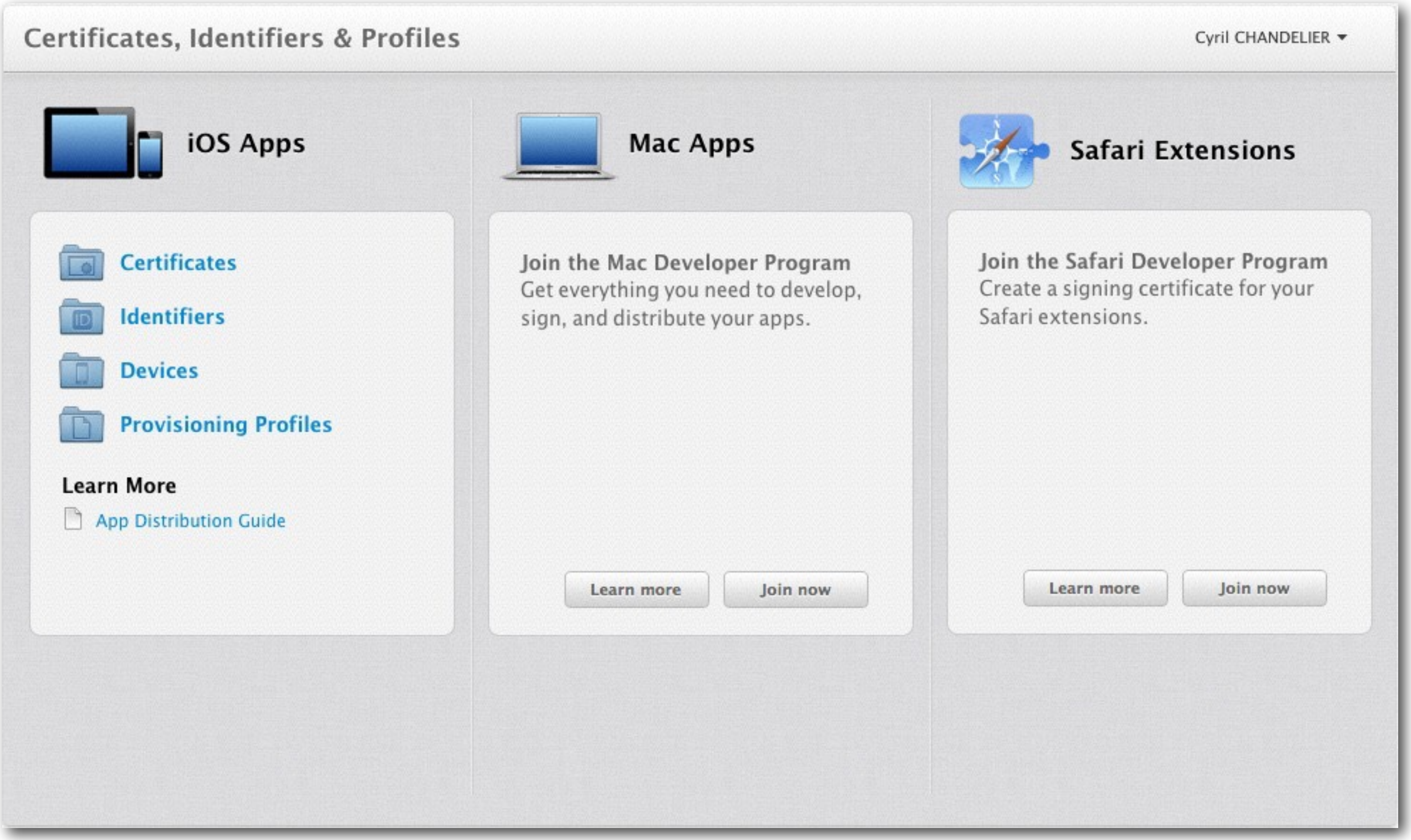
> private / public databases (eq. backend + WS)

> asset storage

> authentification via l’Apple ID (eq. Facebook Connect)

Déploiement

Portail développeur


Certificates, Identifiers & Profiles

Cyril CHANDELIER

iOS Apps

Certificates

Identifiers

Devices

Provisioning Profiles

Learn More

App Distribution Guide

Mac Apps

Join the Mac Developer Program
Get everything you need to develop, sign, and distribute your apps.

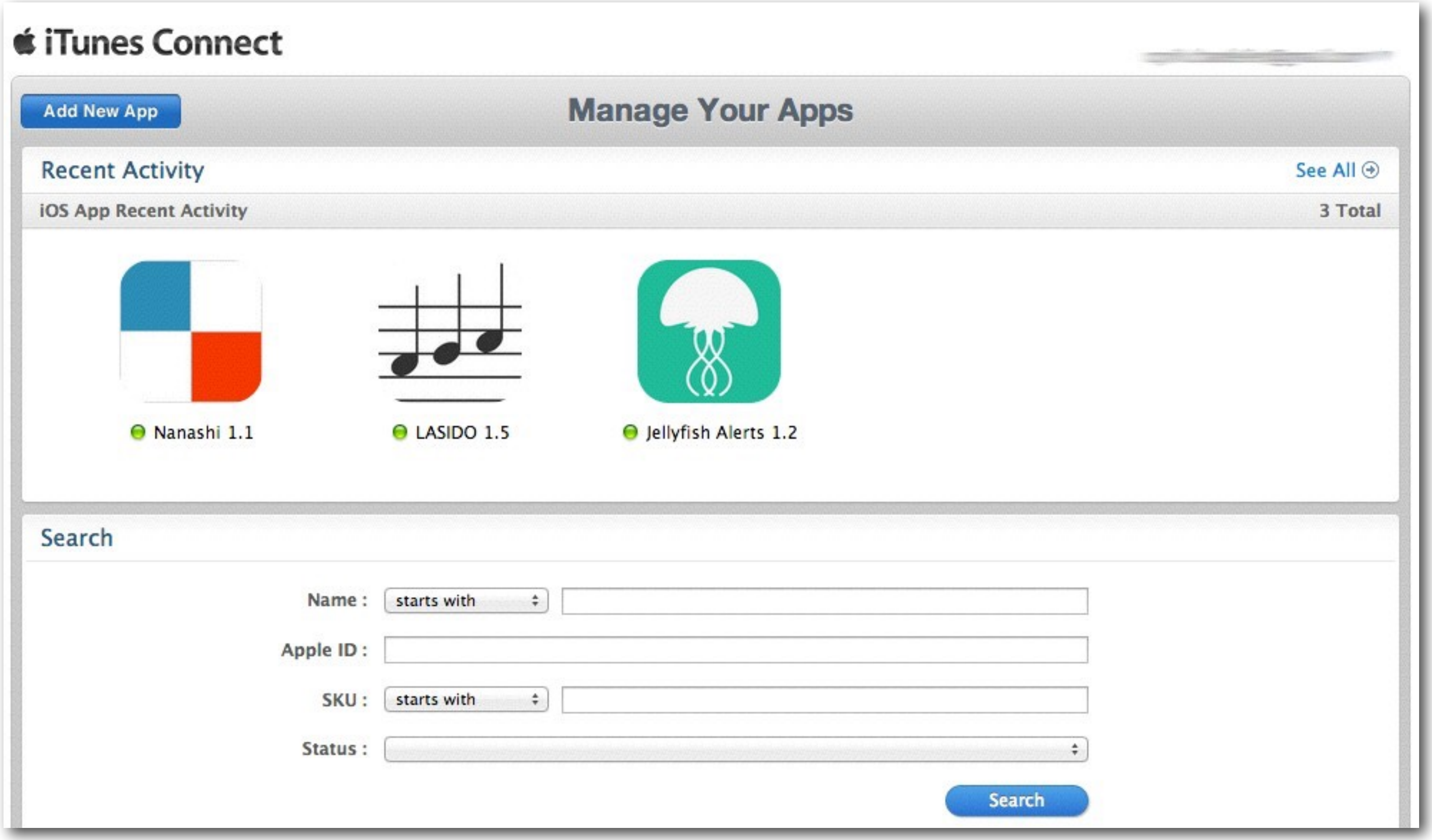
Learn more | Join now

Safari Extensions

Join the Safari Developer Program
Create a signing certificate for your Safari extensions.

Learn more | Join now

Gestion des apps



iTunes Connect

Add New App

Manage Your Apps

Recent Activity — See All

iOS App Recent Activity — 3 Total

- Nanashi 1.1
- LASIDO 1.5
- Jellyfish Alerts 1.2

Search

Name :	starts with
Apple ID :	
SKU :	starts with
Status :	

Search

Testflight

Service racheté par Apple en 2013

Intégré à iTunes Connect depuis Septembre 2014

Gestion de beta testing (utilisateurs, feedback, etc.)

Jusqu’à 2000 beta testers

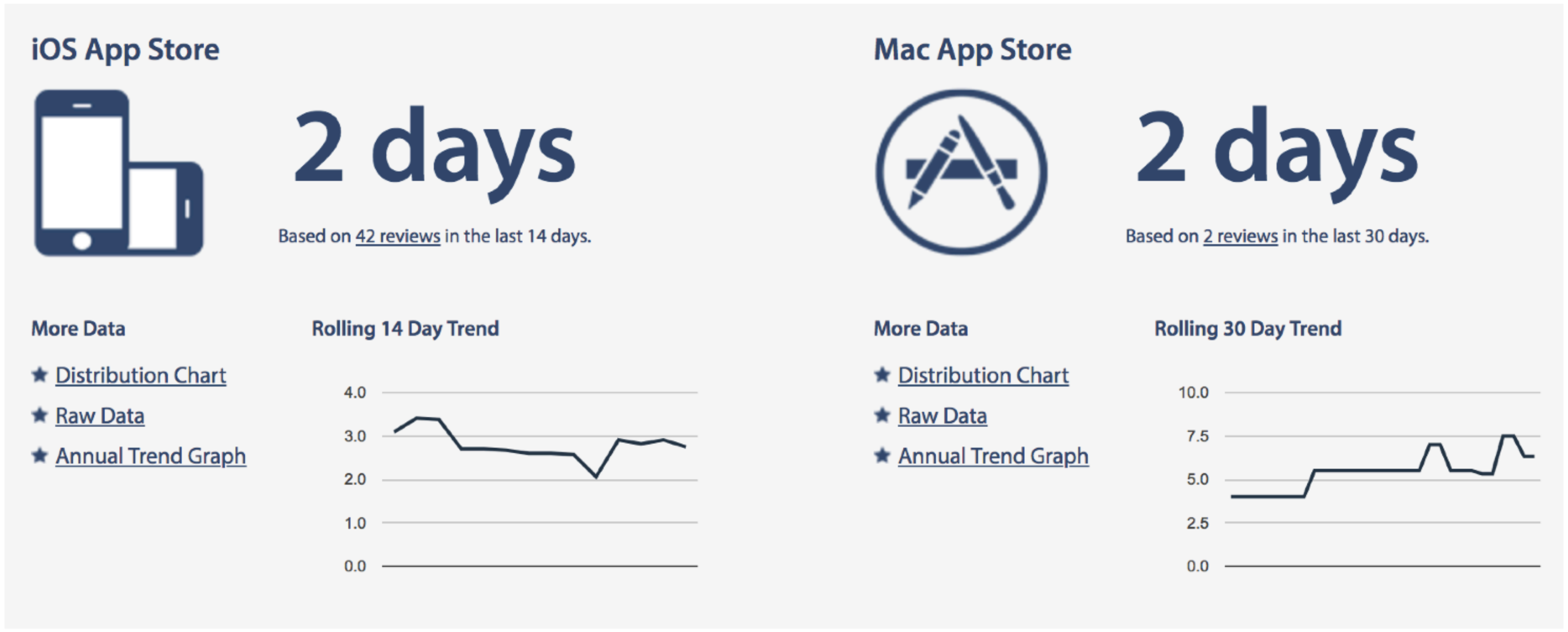
http://www.testflight.com/

Validation

3 à 15 jours (sauf exception)

Ne pas utiliser d'API privées (non documentées) sous peine de rejet

Attention au respect des guidelines Apple



Frameworks majeurs

UIKit

Composants d’UI de base

Accéléromètre

Mouvements du device

Afficher/prendre des photos/vidéos

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

MapKit + CoreLocation

Affichage de cartes

Géo-localisation

Reverse geocoding

Boussole

GameKit

Classements

Scores

Achievements

Joueur contre joueur (tour par tour, temps réel)

P2P

https://developer.apple.com/library/ios/documentation/GameKit/Reference/GameKit_Collection/_index.html

AddressBook / AddressBookUI

Accéder au répertoire

Manipuler les contacts

AddressBookUI fournit les interfaces natives

https://developer.apple.com/library/ios/documentation/ContactData/Conceptual/AddressBookProgrammingGuideforiPhone/Introduction.html

Others

- HomeKit
- ARKit : augmented reality
- Core ML : Machine learning

https://developer.apple.com/documentation

Librairies externes

CocoaPods

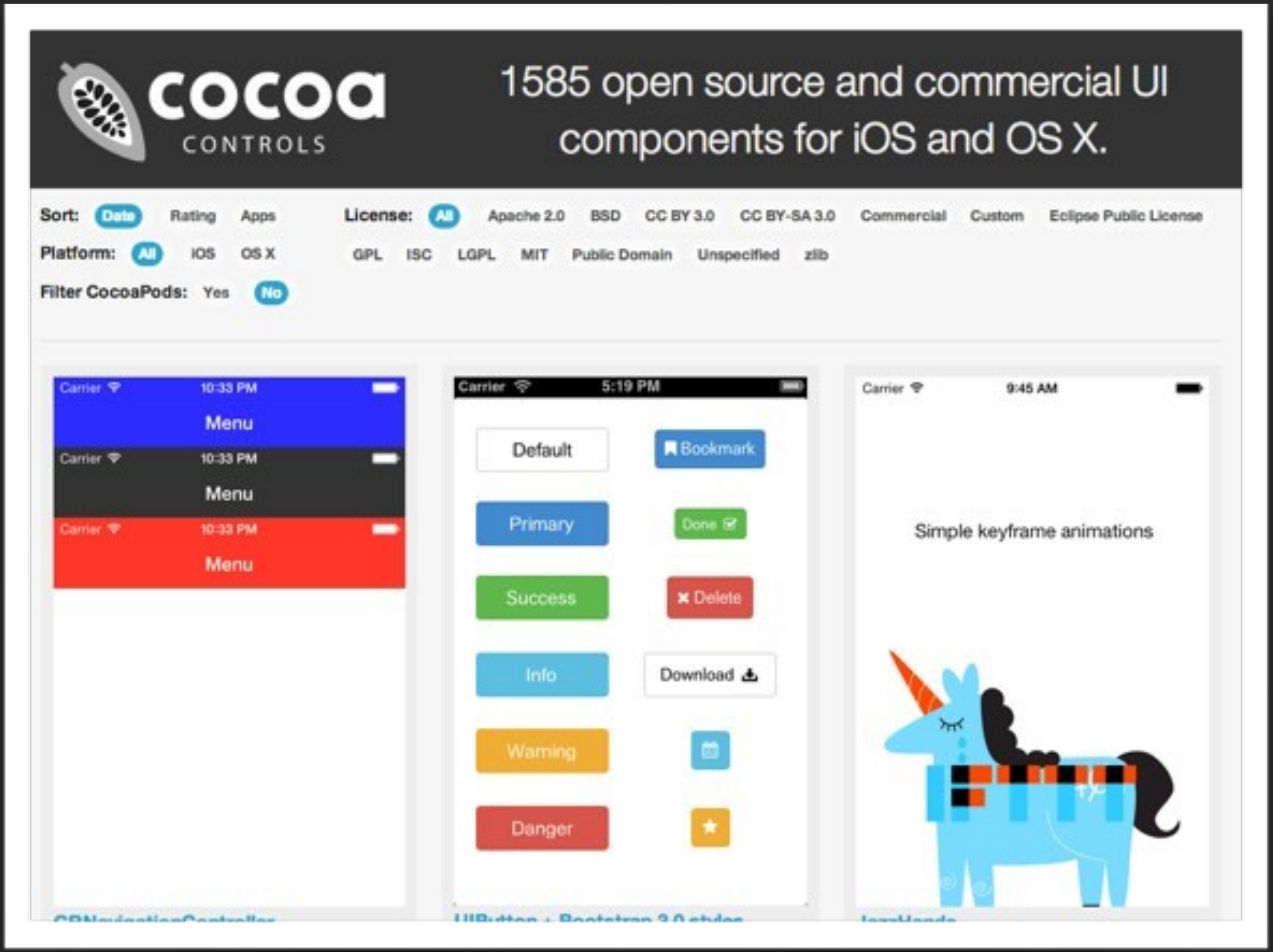
```
$ edit   Podfile  platform
:ios, '6.0'
pod  'JSONKit',              '~>   1.4'
pod  'Reachability',         '~>   3.0.0'

$ pod  install
```

CocoaPods

The best way to manage library dependencies in Objective-C projects.

http://cocoapods.org/

Cocoa Controls



1585 open source and commercial UI components for iOS and OS X.

Sort: [✓ Date] Rating Apps

License: [✓ All] Apache 2.0 BSD CC BY 3.0 CC BY-SA 3.0 Commercial Custom Eclipse Public License GPL ISC LGPL MIT Public Domain Unspecified zlib

Platform: [✓ All] iOS OS X

Filter CocoaPods: Yes [✓ No]

www.cocoacontrols.com

Divers

Documentation

Depuis Xcode

⌘⇧0 ou depuis le menu **Aide**

Depuis un navigateur

http://developer.apple.com/library/ios/

Guidelines

iOS Human Interface Guidelines

https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html

App Review Guidelines

https://developer.apple.com/appstore/resources/approval/guidelines.html

In-App Purchase Guidelines

https://developer.apple.com/in-app-purchase/In-App-Purchase-Guidelines.pdf

Sources

Statistiques http://
www.appexplorer.com/stats/
http://mixpanel.com/trends/

Classement TIOBE

http://www.tiobe.com/index.php/content/
paperinfo/tpci/index.html

Questions ?

TP 1

http://tryobjectivec.codeschool.com

Ne pas hésitez à copier le code dans Xcode sur vos machines pour voir le comportement réel une fois compiler (quelques petites différences)