

DÉVELOPPEMENT SOUS IOS

PREMIÈRE APPLICATION



SWIFT

ios

JDEV2015

TI.A09

● Introduction

- TP0 Installation de XCode
- Environnement iOS
- Licences
- Distribution

● Environnement de développement

- TP1 HelloWorld: découverte de XCode

● Swift

- TP2: les classes
- Frameworks
- Gestion de la mémoire
- Construction de vue de façon programmatique
 - TP3



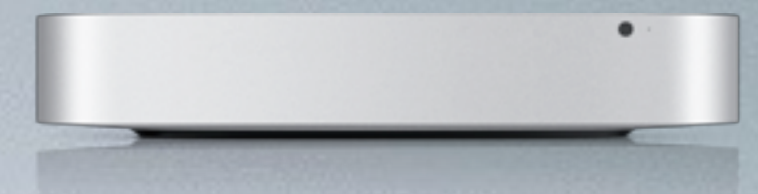
TP0

Installation de XCode



ETAPES

- Ordinateur Apple
- Création d'un Apple ID
- Enregistrement compte développeur
- Chargement et installation de XCode



CRÉATION D'UN APPLE ID (I)

- Web (à préférer)
- iTunes (nécessite la saisie d'une carte pour de futurs potentiels achats)
- Indispensable pour l'utilisation des services et appareils Apple

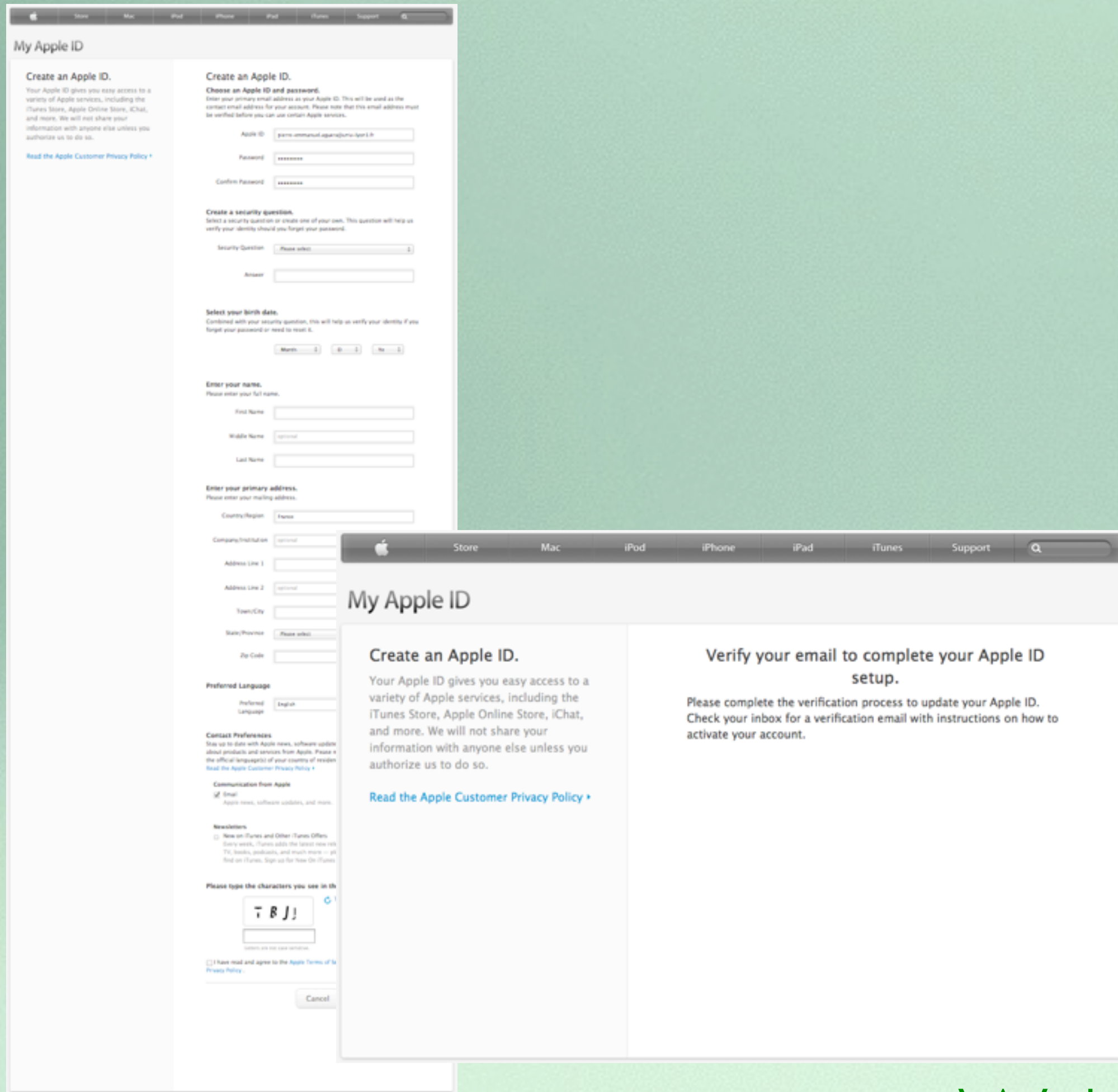


<https://appleid.apple.com>

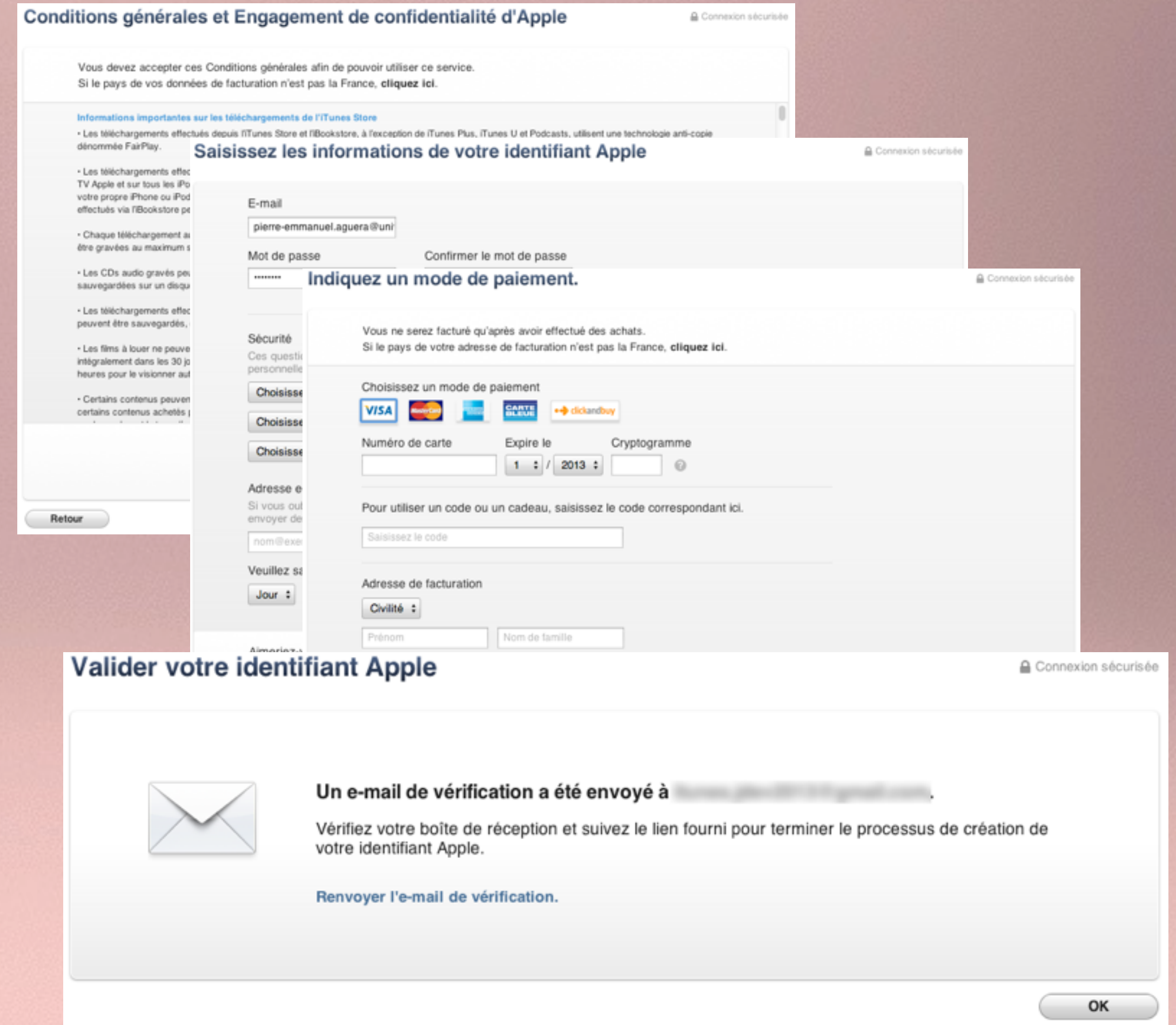


iTunes

CRÉATION D'UN APPLE ID (2)



The screenshot shows the 'My Apple ID' creation process on a web browser. The 'Create an Apple ID' section is active, showing fields for Apple ID, Password, Confirm Password, Security Question, Answer, and Birth Date. Below this, there are fields for First Name, Middle Name, Last Name, and Primary Address. The 'Verify your email to complete your Apple ID setup' section is also visible, indicating that a verification email has been sent to the user's email address.



The screenshot shows the 'Conditions générales et Engagement de confidentialité d'Apple' (Terms and Conditions) screen in the iTunes application. The user is prompted to accept the terms and conditions. The 'Saisissez les informations de votre identifiant Apple' (Enter your Apple ID information) section is active, showing fields for E-mail, Mot de passe (Password), and Confirmer le mot de passe (Confirm password). Below this, the 'Indiquez un mode de paiement' (Indicate a payment method) section is visible, showing options for payment methods like VISA, MasterCard, and Apple Pay. The 'Valider votre identifiant Apple' (Validate your Apple ID) section is also visible, showing a confirmation message that a verification email has been sent to the user's email address.

Web

iTunes



CRÉATION D'UN APPLE ID (3)

Chèr(e) Pierre-Emmanuel Aguera,

Vous avez saisi [redacted] votre identifiant Apple ID. Pour terminer, il s'agit bien de votre adresse électronique et ouvrez une session à l'aide de votre identifiant Apple.

[Vérifiez maintenant >](#)

Pourquoi ce courrier électronique
L'envoi de ce courrier électronique modifie une adresse électronique qui vous concerne pas, ne vous inquiétez pas, il s'agit de votre adresse électronique comme adresse de contact.

Pour plus d'informations, consultez la page d'aide.

Merci,
L'assistance à la clientèle Apple

TM et copyright © 2013 Apple Inc. Apple Sales International
Numéro d'immatriculation de la société : 157143430
[Tous droits réservés](#) / [Tenez-moi informé](#) / [Erreurs](#)

Si vous ne souhaitez plus recevoir d'e-mails de la part d'Apple, cliquez ici.

Store Mac iPod iPhone iPad iTunes Assistance

Mon identifiant Apple


Vérifiez votre adresse électronique.

Veuillez vérifier l'adresse électronique associée à votre identifiant Apple.

Store Mac iPod iPhone iPad iTunes Assistance

Mon identifiant Apple

Vérifiez votre adresse électronique.



Adresse électronique vérifiée.

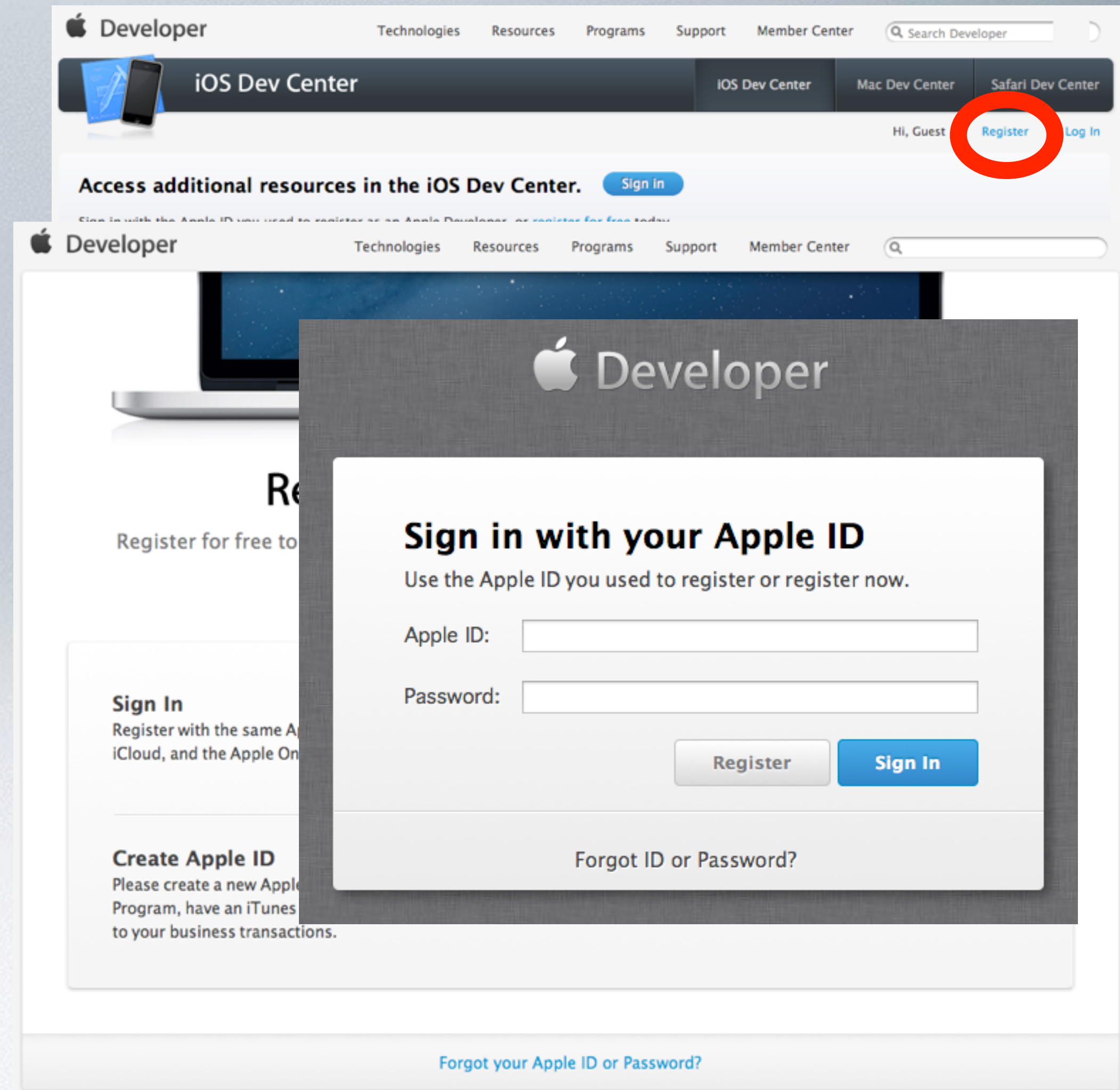
Merci d'avoir vérifié votre adresse. [redacted] est maintenant associé à votre identifiant Apple.

Vérification de l'adresse électronique



CRÉATION D'UN COMPTE DÉVELOPPEUR (I)

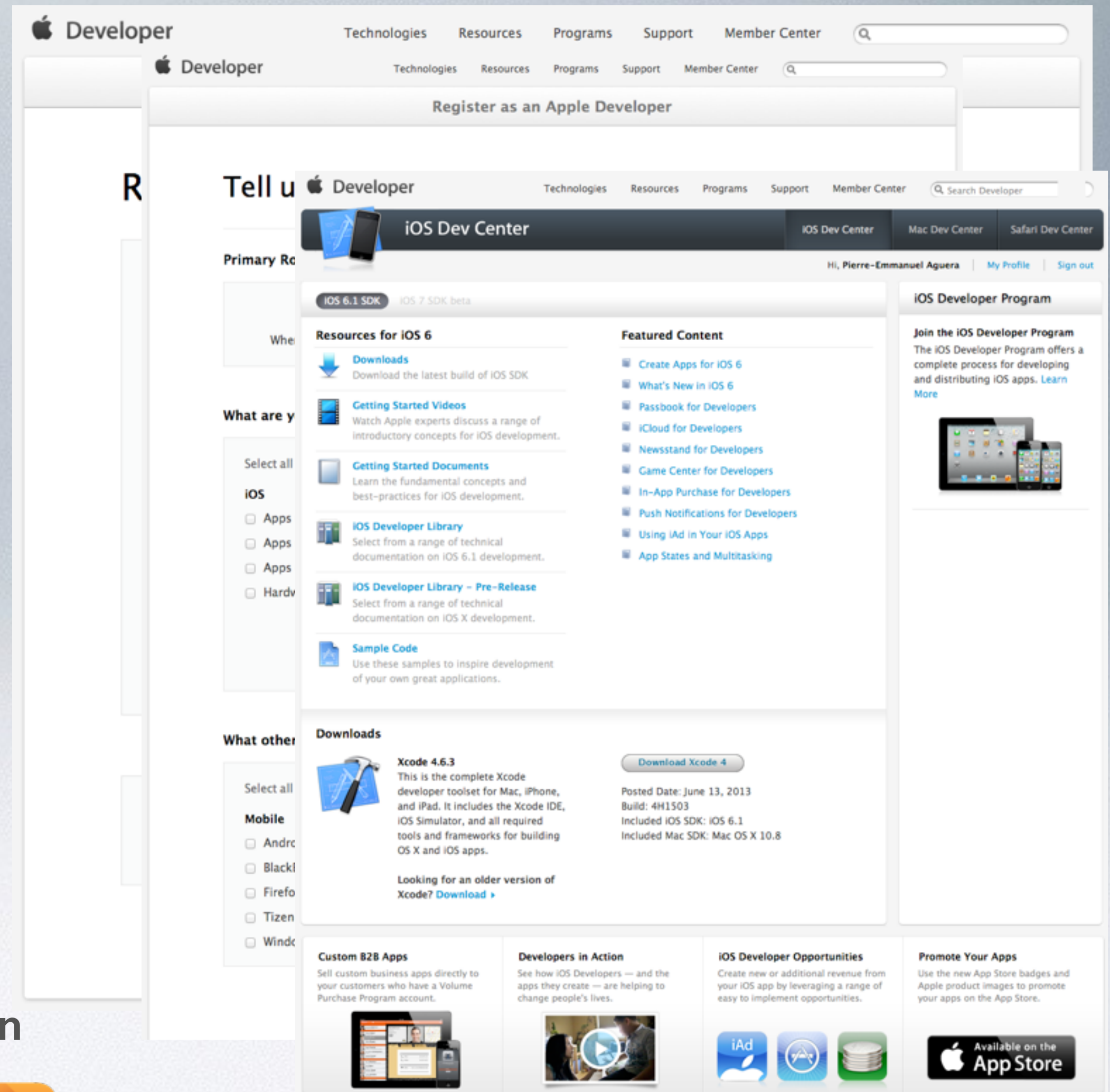
- <https://developer.apple.com/devcenter/ios/index.action>
- Register
- Sign In
- Utilisation de l'AppleID



CRÉATION D'UN COMPTE DÉVELOPPEUR (2)

- Accepter l'Apple Developer Agreement
- Un dernier questionnaire
- Et voilà

<https://developer.apple.com/devcenter/ios/index.action>

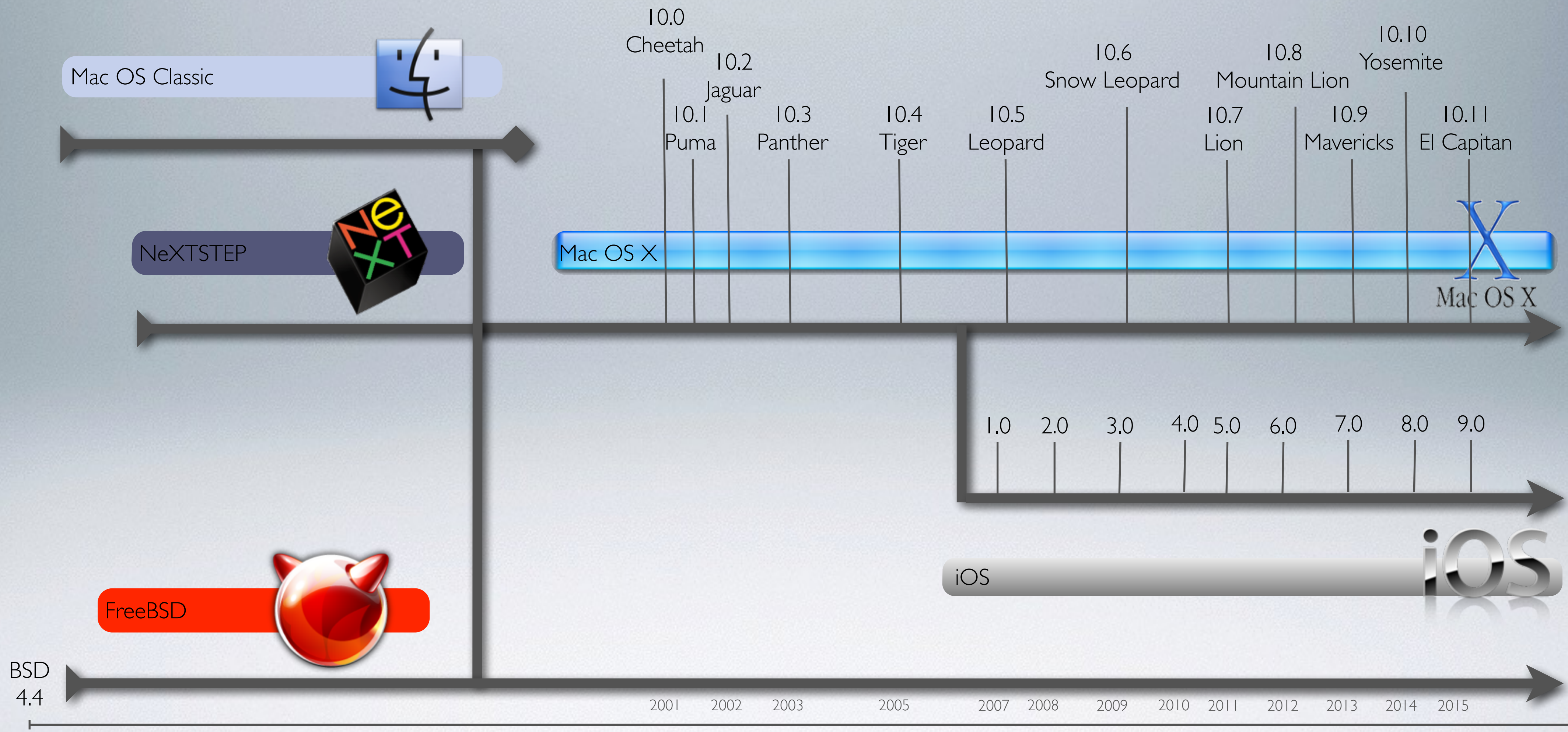


INSTALLATION DE XCODE

- Download XCode
- Renvoi vers Mac App Store
- Ouvrir avec l'App Store
- Cliquer sur Gratuit
- Et installer l'app



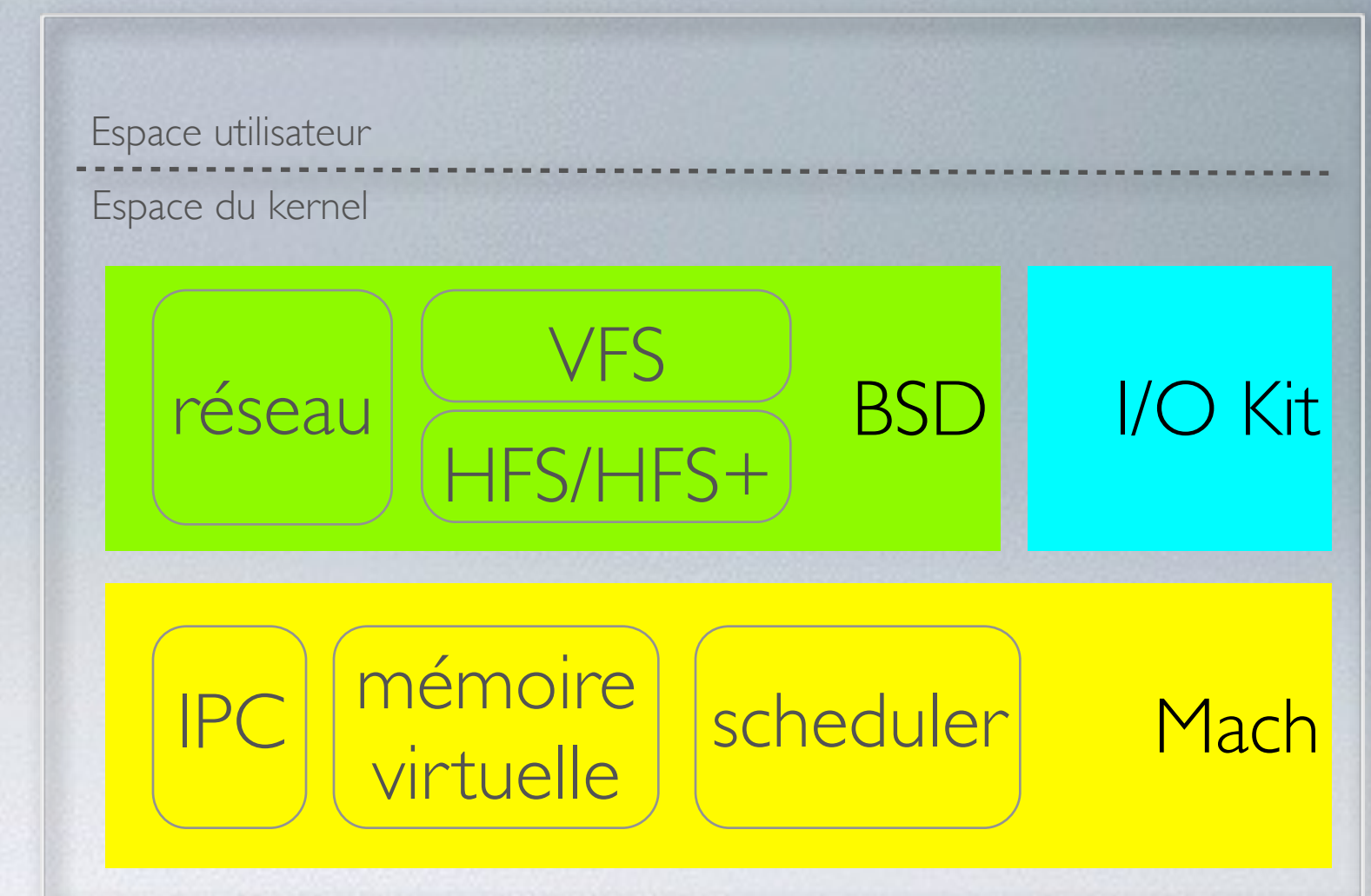
DARWINISME



CARACTÉRISTIQUES LOGICIELLES

Kernel

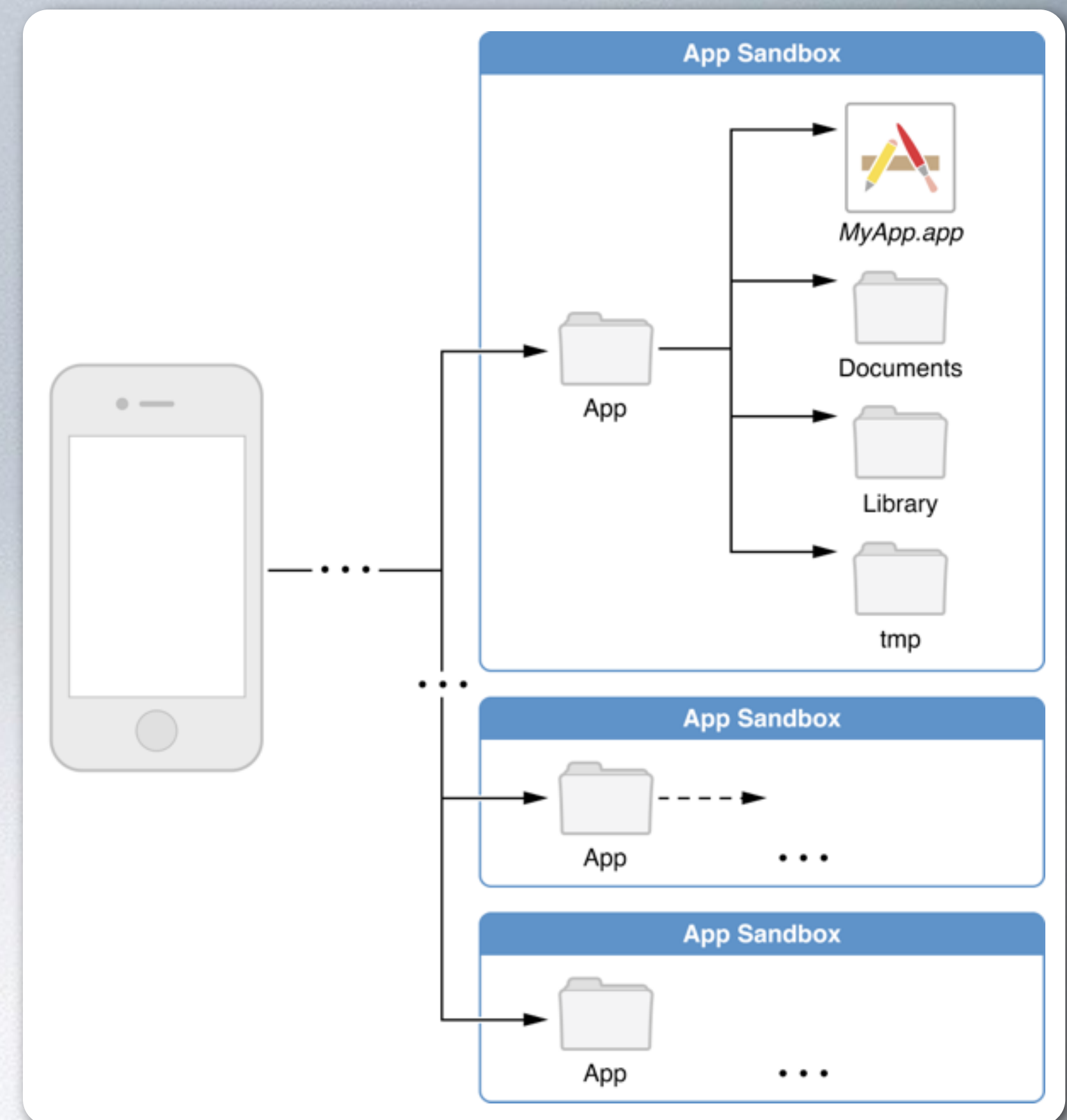
- iOS dérivé de Mac OS X
- Coeur Darwin
- Kernel XNU (X is Not UNIX)
- API spécifiques (SpringBoard, Cocoa Touch, Accelerate framework, External Accessory Framework, ...)



APPLICATIONS

Limitations

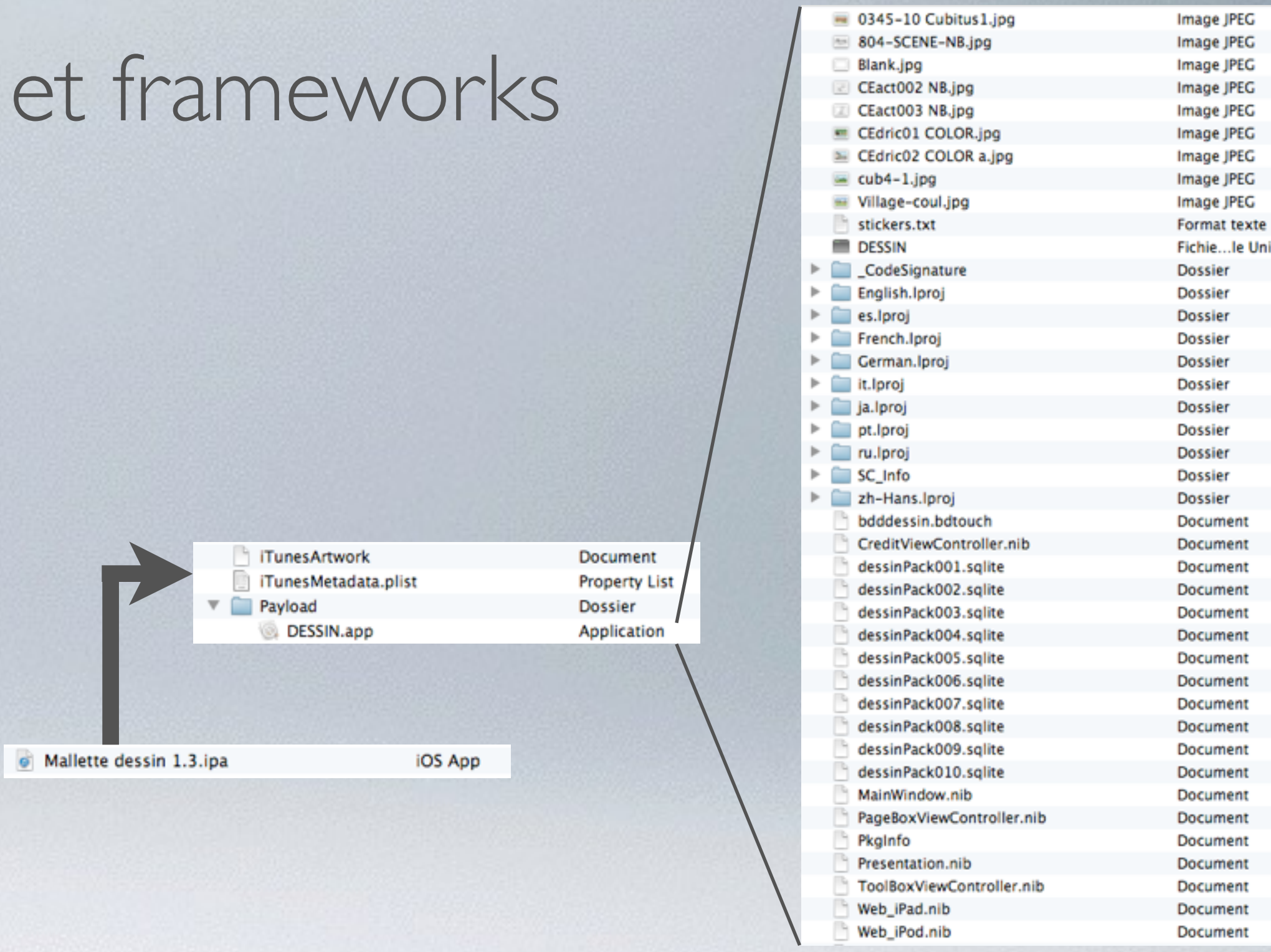
- Mémoire physique limitée
 - ➔ pas de swap
 - ➔ processus de libération de la mémoire
- Gestion de l'énergie :
 - ➔ systèmes de veille
 - ➔ multitâche
- Performances matérielles limitées



CARACTÉRISTIQUES LOGICIELLES

Bundles, applications et frameworks

- Bundle : structure hiérarchique standardisée contenant du code exécutable et les ressources utilisées par ce code
- Application : packagée dans un bundle contenant l'exécutable et ses ressources
- Framework : bundle contenant une ou plusieurs bibliothèques et leurs fichiers



CodeResources/	Symbolic link to Code Signature/CodeResources plist
Headers/	Symbolic link to Miscellaneous .h files provided by this
framework	
Resources/	.nib files (GUI), .lproj files, or other files required by
framework	
Versions/	Subdirectory to allow versioning
A/	Letter directories denoting version of this framework
Current/	Symbolic link to preferred framework version
Framework -name	Symbolic link to framework binary, in preferred version



LICENCES DE DÉVELOPPEMENT

Programmes

	iOS Developer For individuals and organizations creating apps for distribution on the App Store. Learn more ▶	iOS Developer Enterprise For companies and organizations distributing proprietary apps for internal use. Learn more ▶	iOS Developer University For higher education institutions introducing iOS app development into their curriculum. Learn more ▶
iOS SDK	✓	✓	✓
iOS SDK (Pre-release)	✓	✓	N/A
Test apps on iOS devices	✓	✓	✓
Code-level Technical Support	✓	✓	N/A
Ad Hoc Distribution	✓	✓	N/A
App Store Distribution	✓	N/A	N/A
Custom B2B App Distribution	✓	N/A	N/A
iAd Network	✓	N/A	N/A
In-house Distribution	N/A	✓	N/A
Cost	\$99 year	\$299 year	Free
Requirements	If you're enrolling as an organization, a D-U-N-S Number registered to your legal entity is required.	A D-U-N-S Number registered to your legal entity is required.	The University Program is only available to qualified, degree granting, higher education institutions.

LICENCES DE DÉVELOPPEMENT

Rôles

Accès et responsabilité	Agent	Admin	Member
Accès au chargement des logiciels et pre-release	✓	✓	✓
Soumettre des incidents au support technique (en payant ou non)	✓	✓	✓
Accès aux forums développeurs	✓	✓	✓
Inviter d'autres membres et assigner les rôles	✓	✓	✗
Ajouter d'autres programmes de développement (Mac OS, Safari)	✓	✗	✗
Accepter le Developer Program Agreements	✓	✗	✗
Renouveler l'adhésion au(x) programme(s)	✓	✗	✗

iOS Developer Program	Agent	Admin	Member
Charger des profils de développement	✓	✓	✓
Soumettre des requêtes de signature des certificat	✓	✓	✓
Approuver des requêtes de signature des certificats	✓	✓	✗
Créer de profils de développement	✓	✓	✗
Enregistrer, configurer et supprimer des App IDs	✓	✓	✗
Ajouter, supprimer des UDIDs d'appareils	✓	✓	✗
Créer des certificat de distribution et et des profils de distribution	✓	✓	✗
Créer des certificats pour le service Apple Push Notification et Pass Type ID	✓	✓	✗
Soumettre des apps et des mise à jour des apps à iTunes Connect	✓	✗	✗

DISTRIBUTION D'APPLICATION

Modes de distribution

- 4 façons de distribuer les applications
 - ▶ AppStore
 - ▶ B2B
 - ▶ Ad Hoc
 - ▶ In House



LANGAGES



1983

basé sur C et Smalltalk

orienté objet



v1.00 02-06-2014

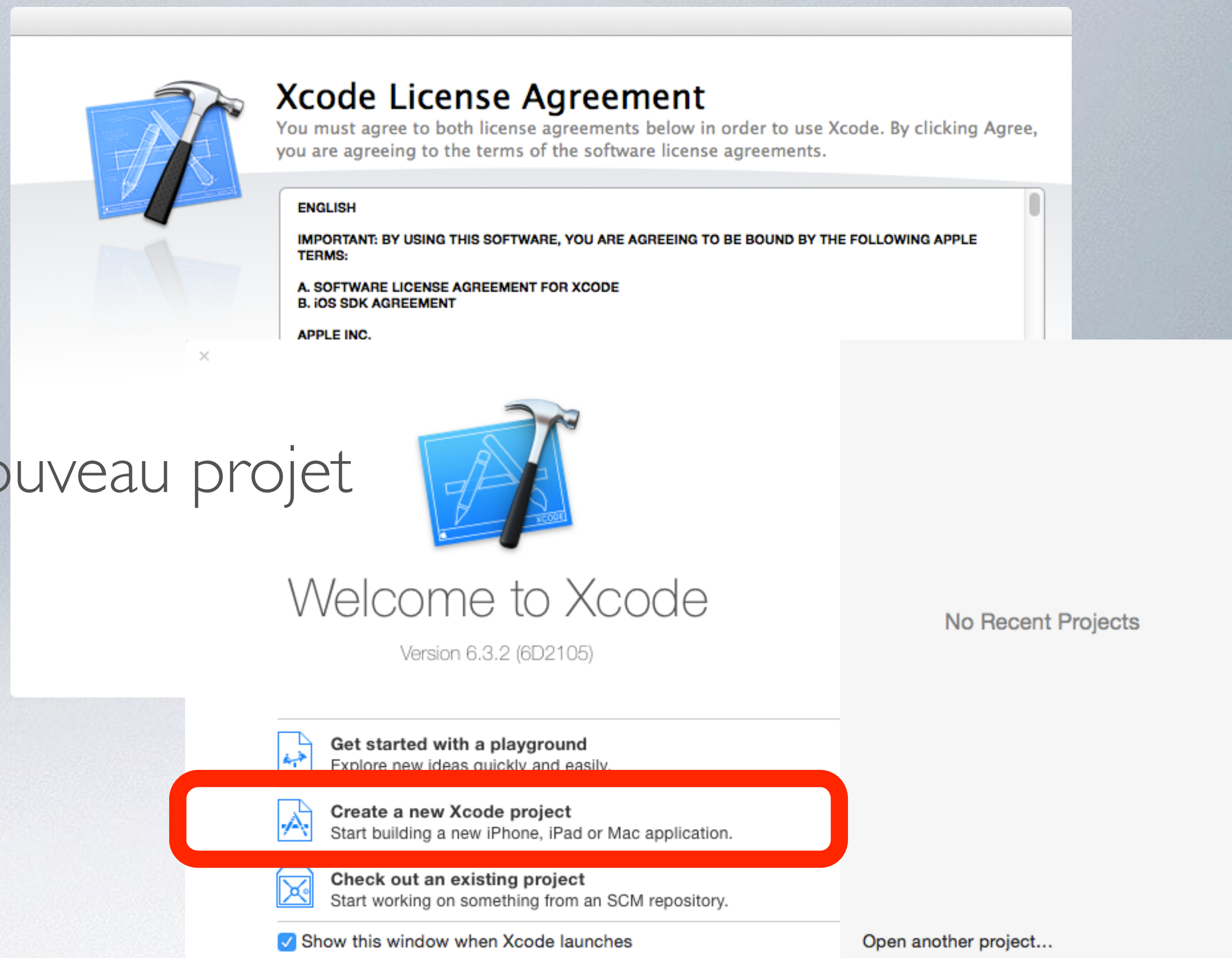
dernière version 1.2 (avril 2015)
version 2.0 en développement

Objective-C, C#, CLU,
Haskell, Python, Racket,
Ruby, Rust et langage D

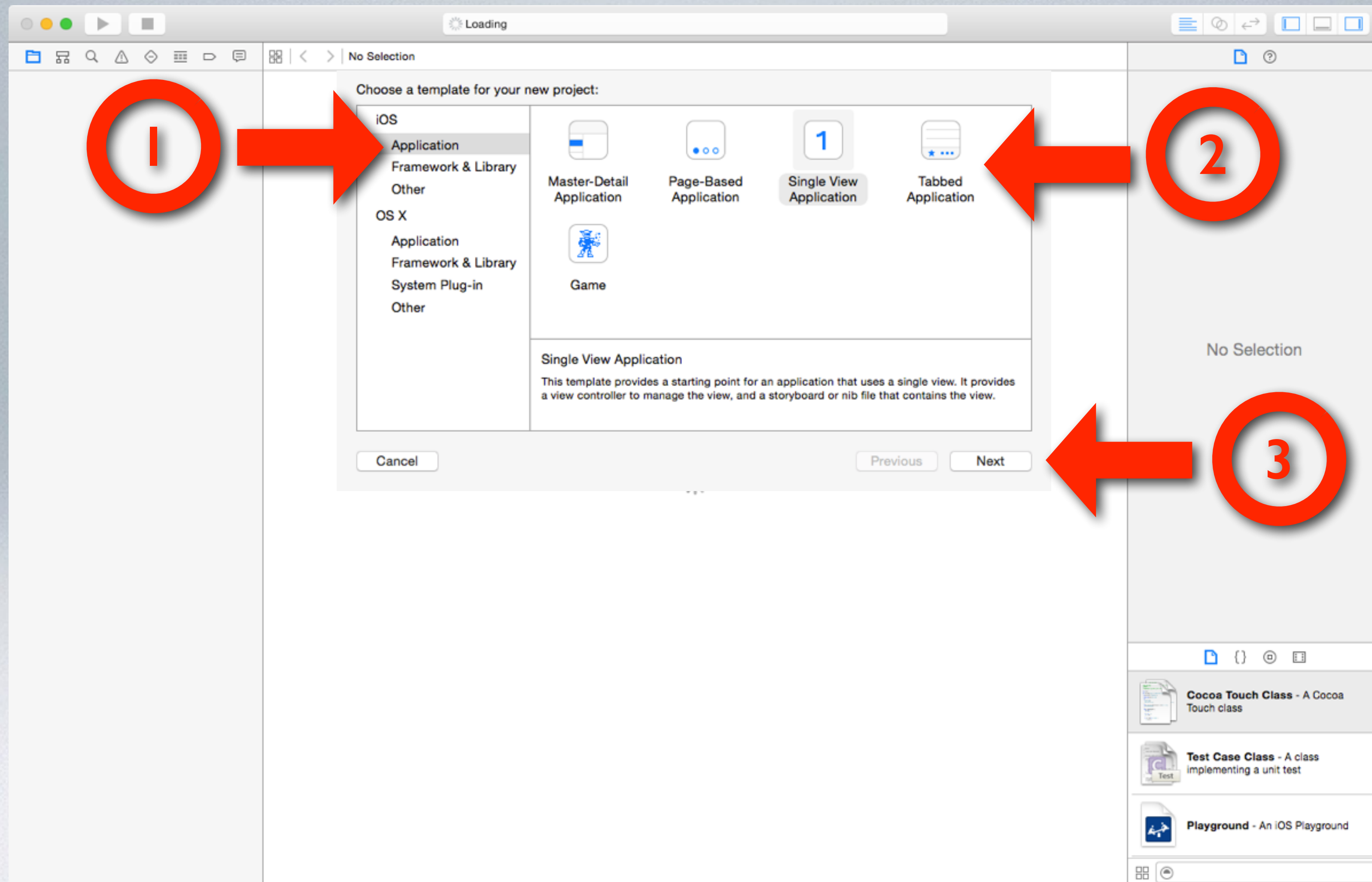
PREMIER LANCEMENT

- Accepter la licence

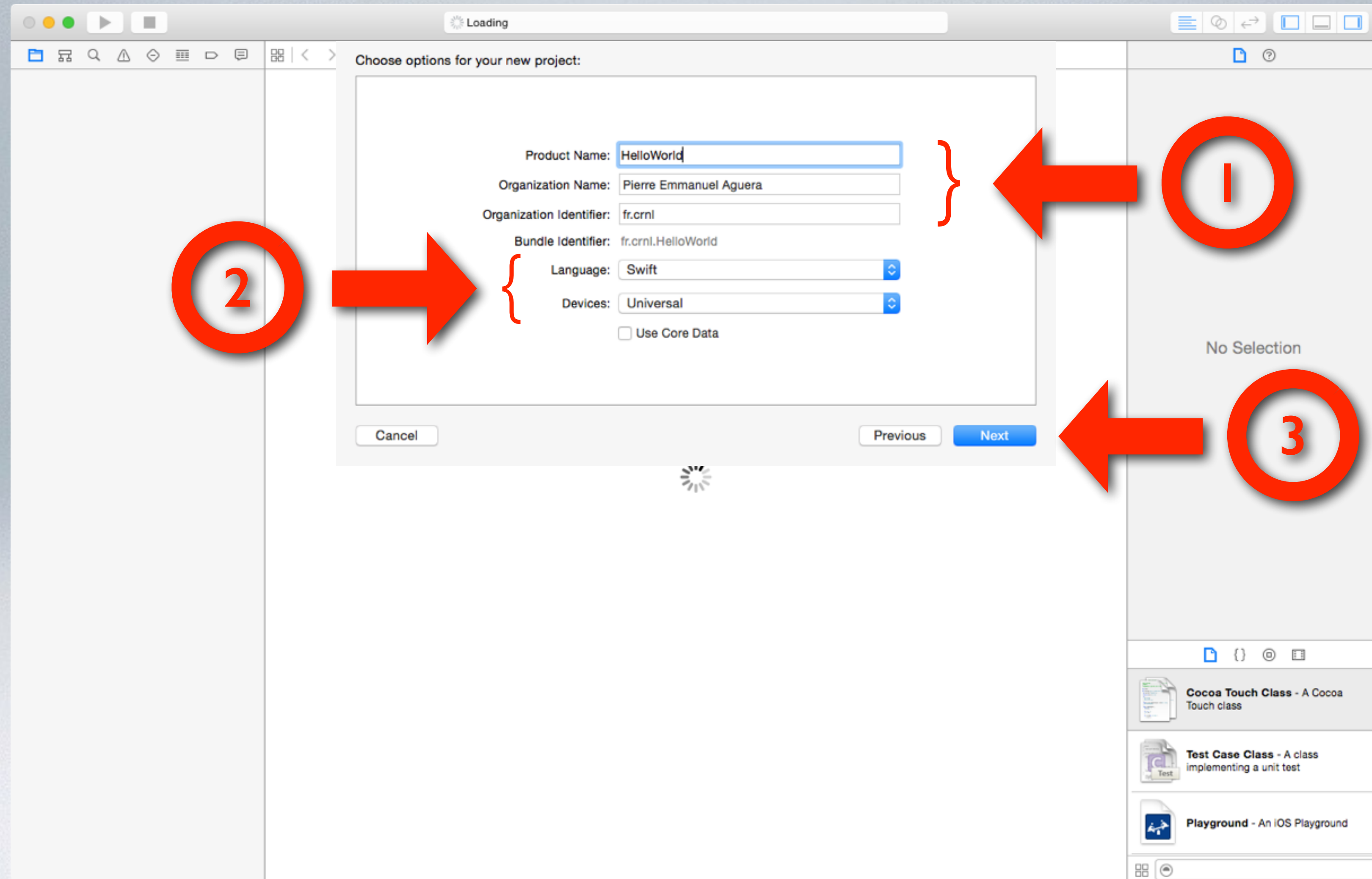
- Créer et ouvrir un nouveau projet



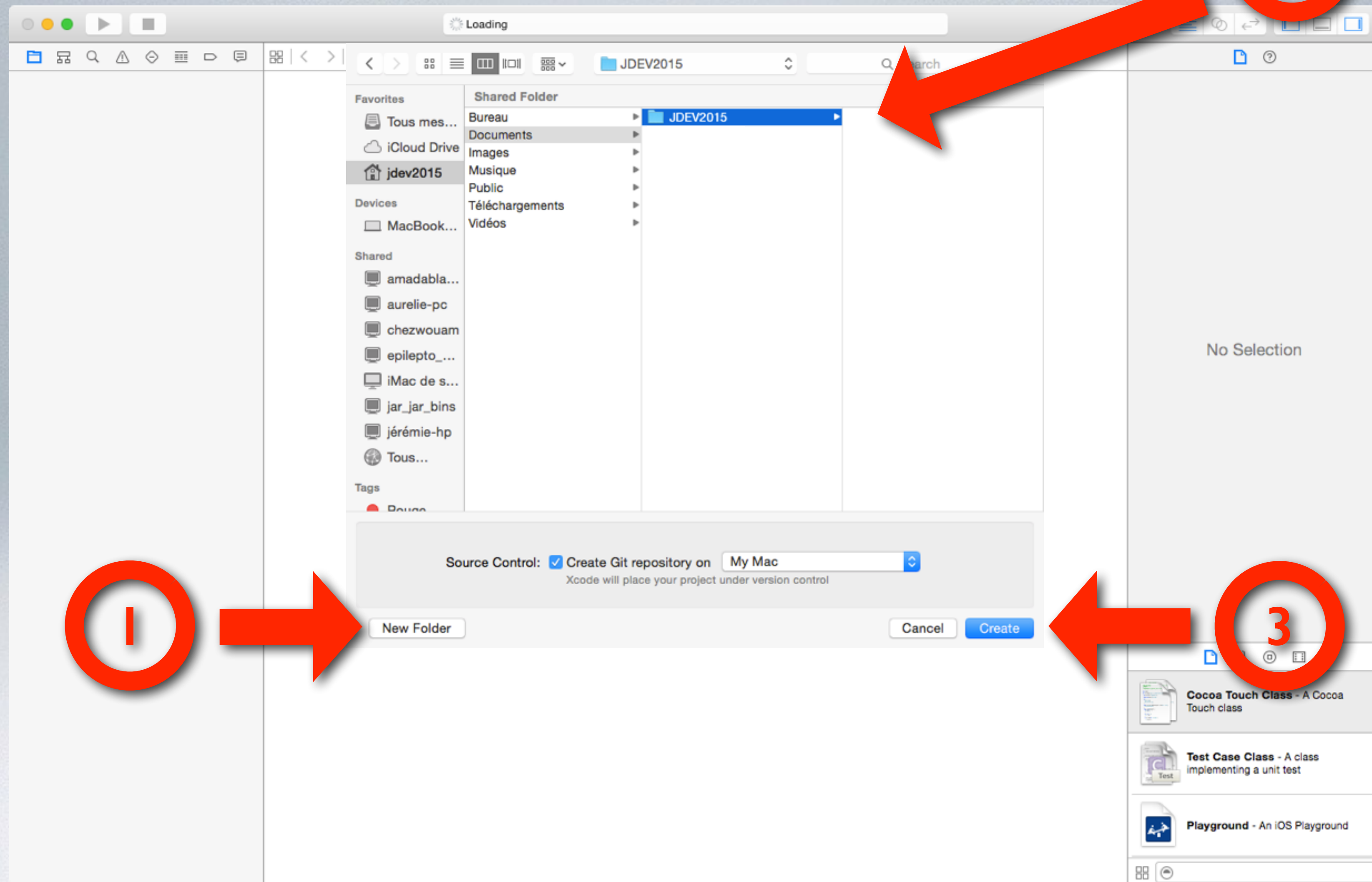
PREMIER PROJET



HELLOWORLD



ENREGISTREMENT



CRÉATION

HelloWorld

2 targets, iOS SDK 8.3

HelloWorld

AppDelegate.swift

ViewController.swift

Main.storyboard

Images.xcassets

LaunchScreen.xib

Supporting Files

HelloWorldTests

Products

General

Capabilities

Info

Build Settings

Build Phases

Build Rules

PROJECT

HelloWorld

TARGETS

HelloWorld

HelloWorldTests

Identity

Bundle Identifier

fr.crnl.HelloWorld

Version

1.0

Build

1

Team

None

Deployment Info

Deployment Target

8.3

Devices

Universal

Main Interface

Main

Device Orientation

Portrait

Upside Down

Landscape Left

Landscape Right

Status Bar Style

Default

Hide status bar

App Icons and Launch Images

App Icons Source

AppIcon

Launch Images Source

Use Assets

Launch Screen File

LaunchScreen

Embedded Binaries

Quick Help

No Quick Help

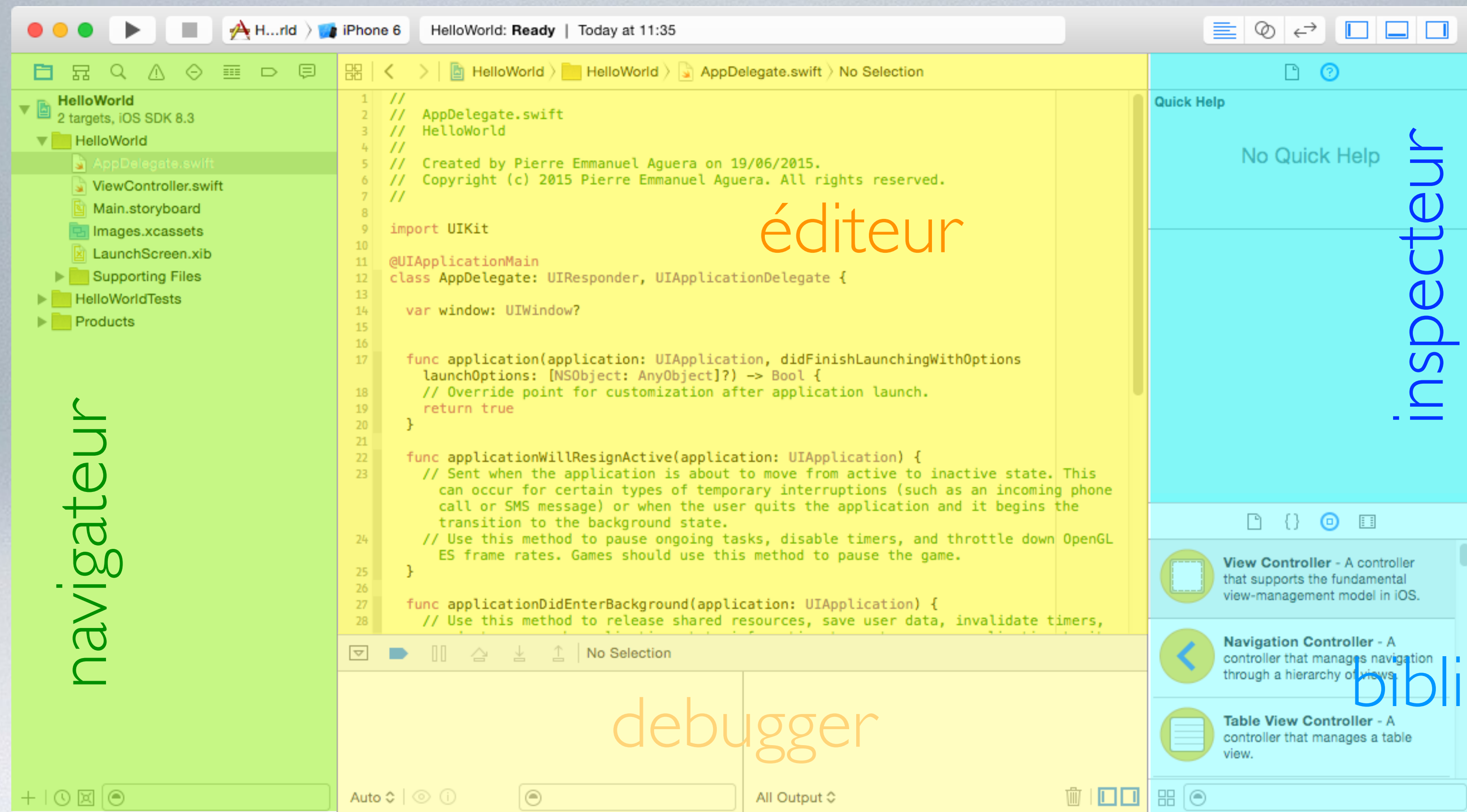
HelloWorld

Rechercher

Nom	Date de modification	Taille	Type
.DS_Store	aujourd'hui 11:36	6 Ko	Document
.git	aujourd'hui 11:36	--	Dossier
HelloWorld	aujourd'hui 11:36	--	Dossier
.DS_Store	aujourd'hui 11:36	6 Ko	Document
AppDelegate.swift	aujourd'hui 11:35	2 Ko	Swift...ce Code
Base.lproj	aujourd'hui 11:35	--	Dossier
LaunchScreen.xib	aujourd'hui 11:35	4 Ko	Texte XML
Main.storyboard	aujourd'hui 11:35	2 Ko	Interfa...ument
Images.xcassets	aujourd'hui 11:36	--	Dossier
.DS_Store	aujourd'hui 11:36	6 Ko	Document
AppIcon.appiconset	aujourd'hui 11:35	--	Dossier
Contents.json	aujourd'hui 11:35	1 Ko	JSON
Info.plist	aujourd'hui 11:35	2 Ko	liste d...priétés
ViewController.swift	aujourd'hui 11:35	509 octets	Swift...ce Code
HelloWorld.xcodeproj	aujourd'hui 11:35	--	Xcode Project
HelloWorldTests	aujourd'hui 11:35	--	Dossier
HelloWorldTests.swift	aujourd'hui 11:35	934 octets	Swift...ce Code
Info.plist	aujourd'hui 11:35	746 octets	liste d...priétés



DESCRIPTION WORKSPACE



bibliothèques

DESCRIPTION APPLICATION (I)

```
// AppDelegate.swift
// HelloWorld
//

import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
        // Override point for customization after application launch.
        return true
    }

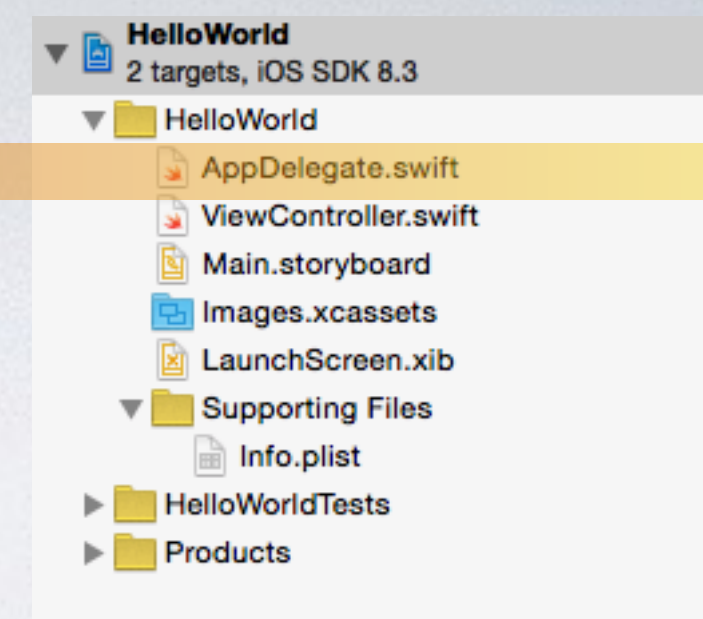
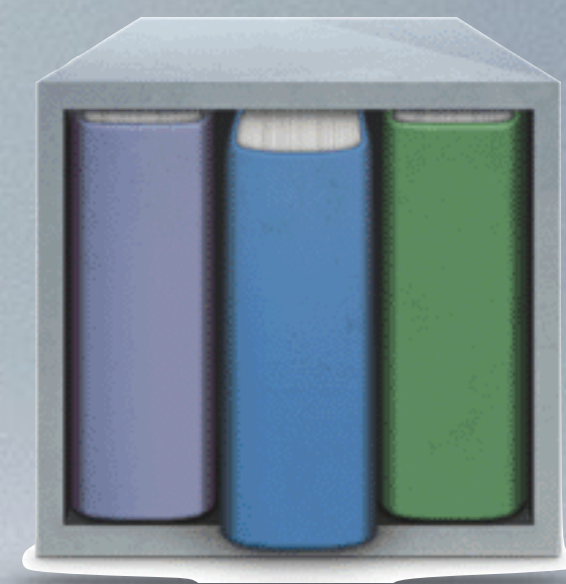
    func applicationWillResignActive(application: UIApplication) {
        // Sent when the application is about to move from active to inactive state. This can occur for certain types of
        // temporary interruptions (such as an incoming phone call or SMS message) or when the user quits the application 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.
    }

    func applicationDidEnterBackground(application: UIApplication) {
        // 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.
    }

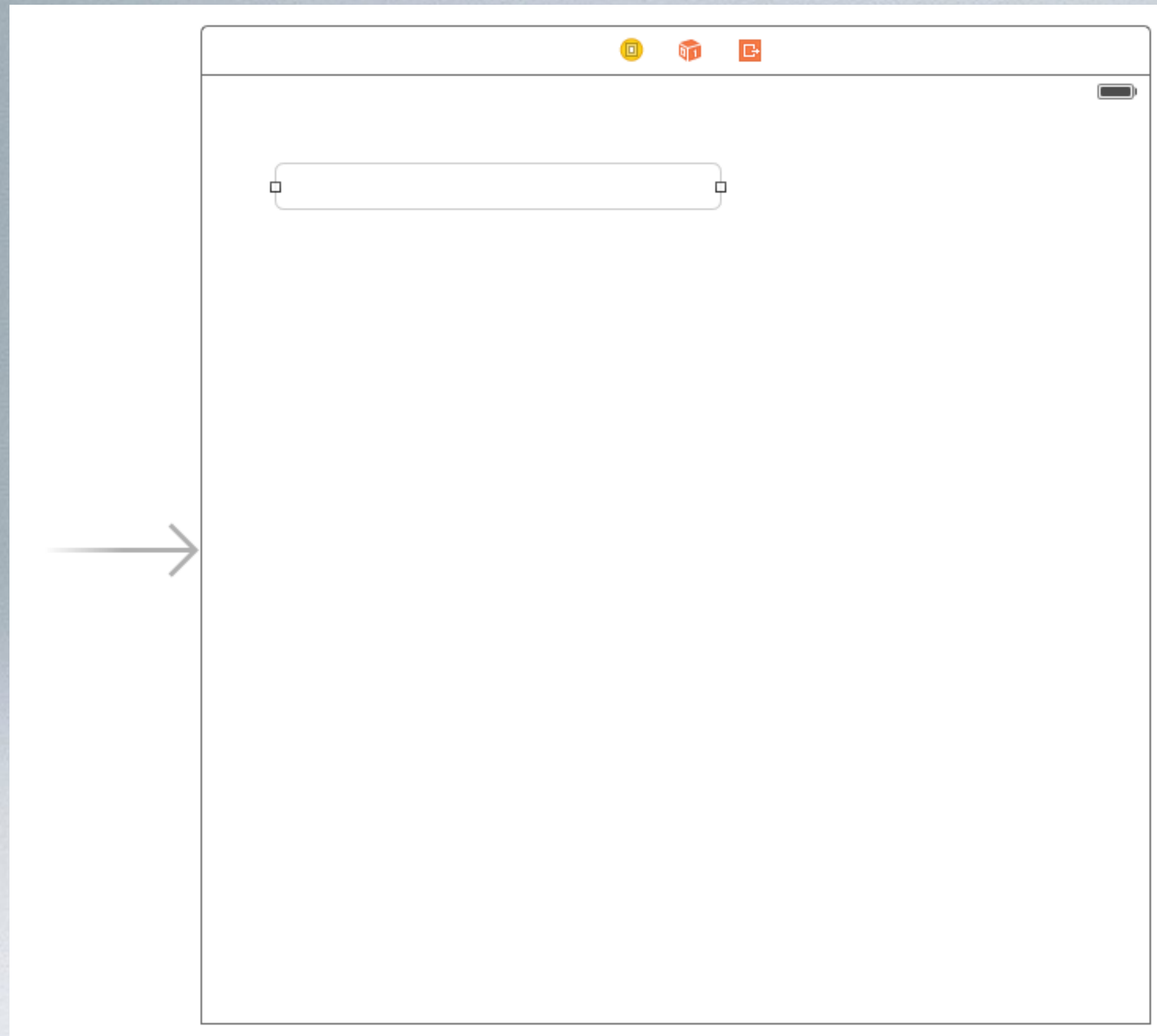
    func applicationWillEnterForeground(application: UIApplication) {
        // Called as part of the transition from the background to the inactive state; here you can undo many of the
        // changes made on entering the background.
    }

    func applicationDidBecomeActive(application: UIApplication) {
        // Restart any tasks that were paused (or not yet started) while the application was inactive. If the application
        // was previously in the background, optionally refresh the user interface.
    }

    func applicationWillTerminate(application: UIApplication) {
        // Called when the application is about to terminate. Save data if appropriate. See also
        applicationDidEnterBackground:.
    }
}
```



LA DÉLÉGATION



▼ UITextFieldDelegate Protocol Reference

Overview

▼ Tasks

▼ Managing Editing

- textFieldShouldBeginEditing:
- textFieldDidBeginEditing:
- textFieldShouldEndEditing:
- textFieldDidEndEditing:

▼ Editing the Text Field's Text

- textField:shouldChangeCharactersInRange:replac...
- textFieldShouldClear:
- textFieldShouldReturn:

Revision History

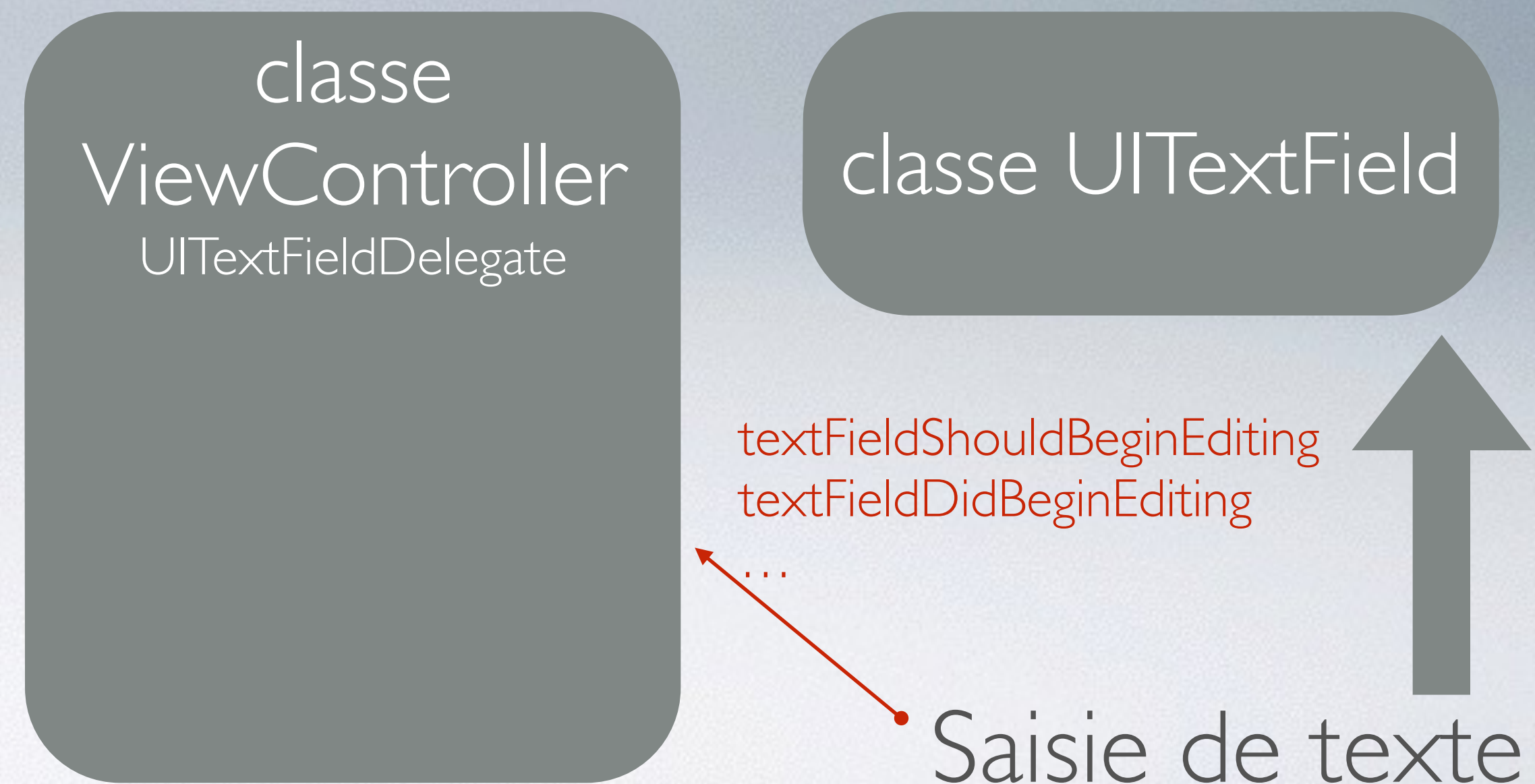
UITextFieldDelegate

Inherits from: None

Conforms to: [NSObject](#)

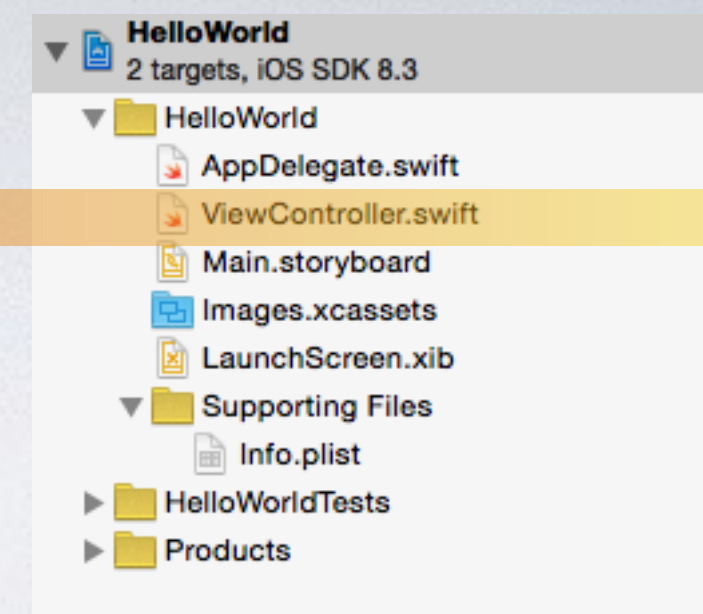
Framework: [UIKit in iOS 2.0 and later.](#) [More related](#)

The UITextFieldDelegate protocol defines the messages sent to a text field delegate as part of the sequence of editing its text. All of the methods of this protocol are optional.

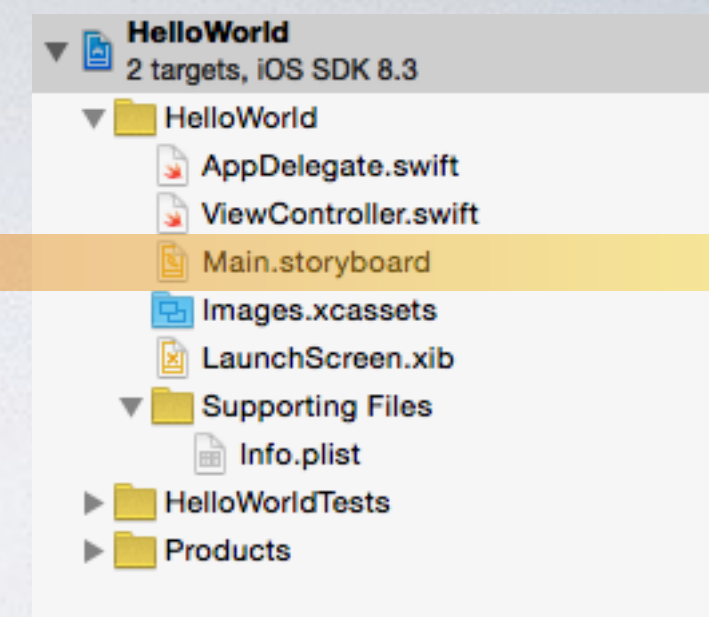
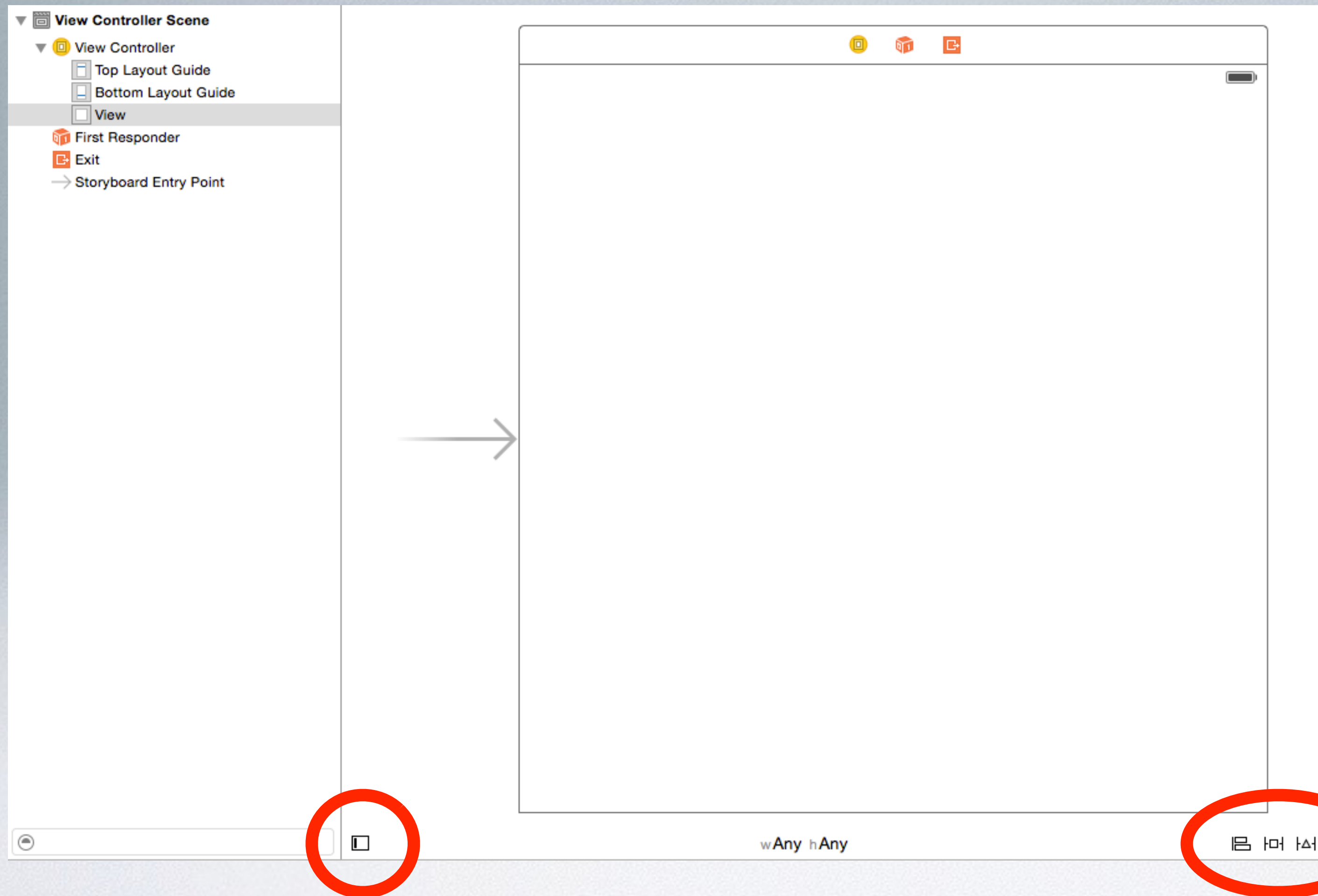


DESCRIPTION APPLICATION (2)

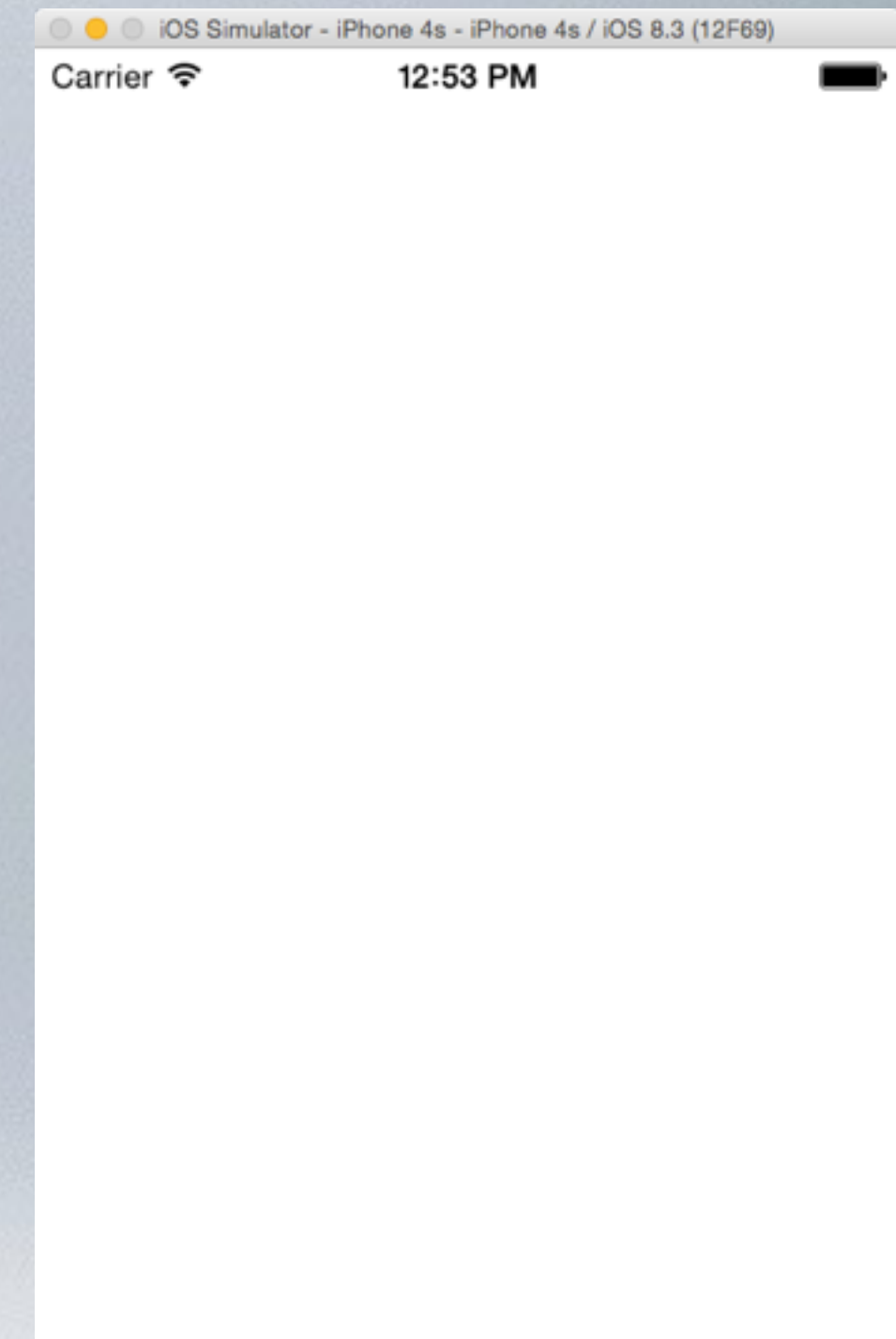
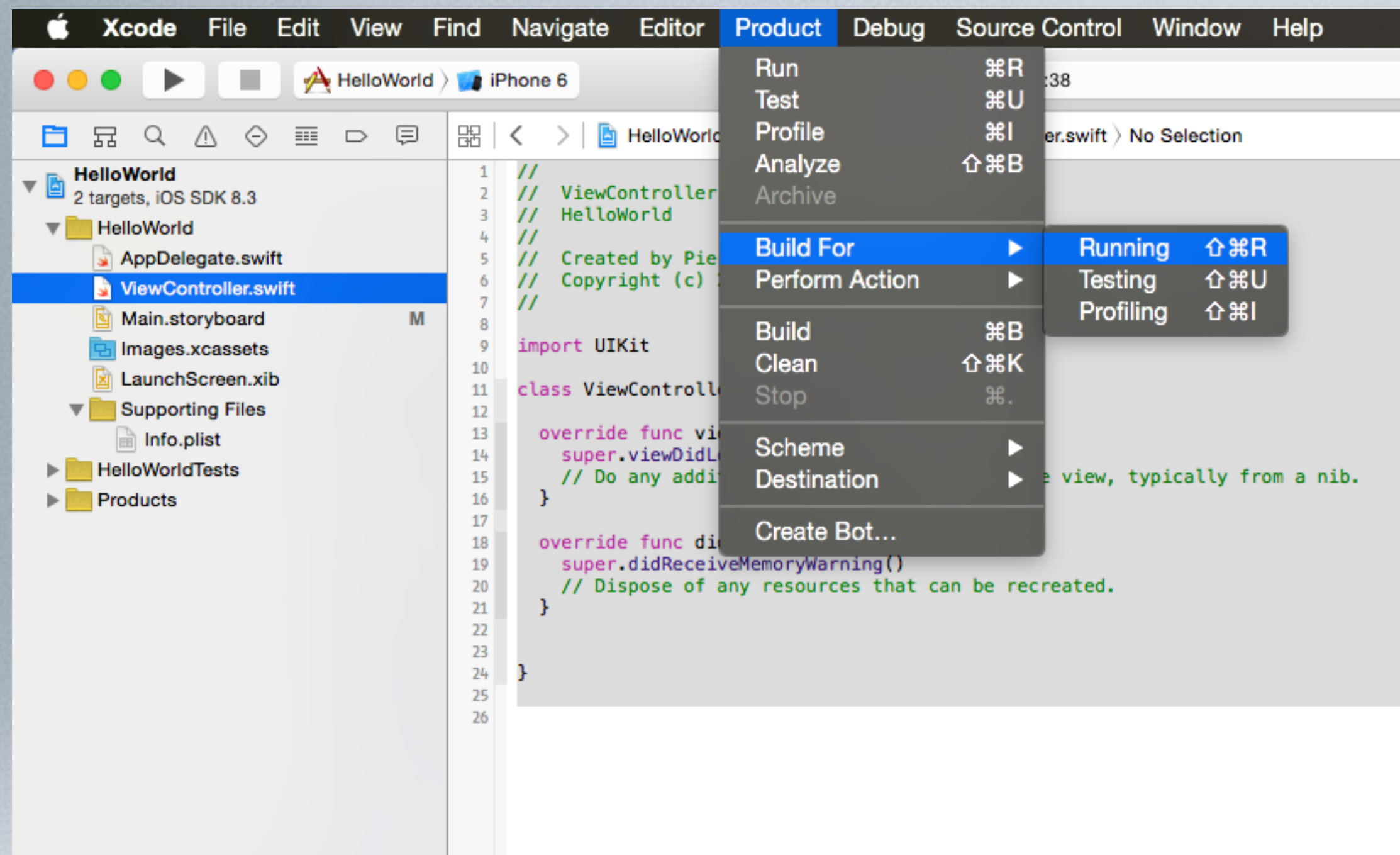
```
//  
// ViewController.swift  
// HelloWorld  
//  
// Created by Pierre Emmanuel Aguera on 19/06/2015.  
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController {  
  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        // Do any additional setup after loading the view, typically from a nib.  
    }  
  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
        // Dispose of any resources that can be recreated.  
    }  
  
}
```



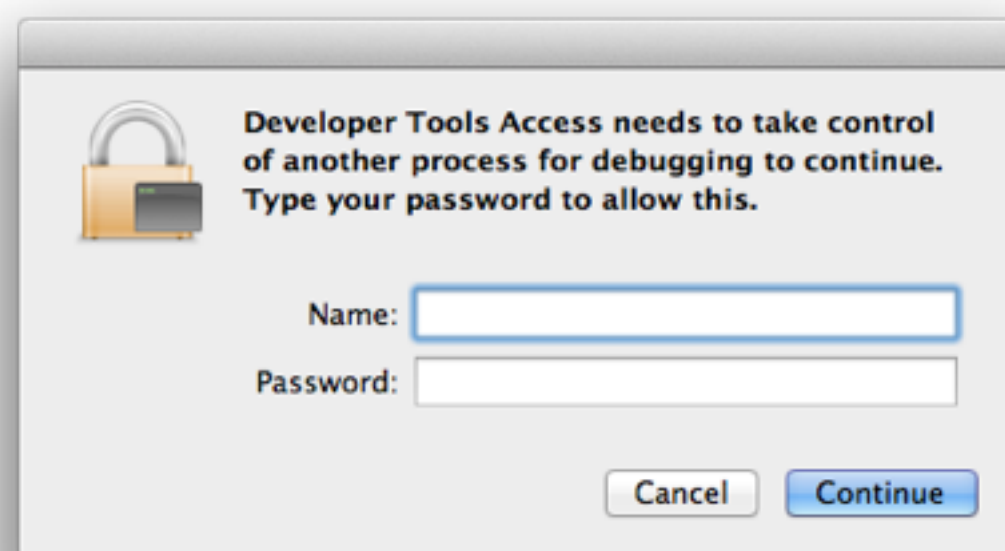
DESCRIPTION APPLICATION (3)



COMPILATION - EXECUTION

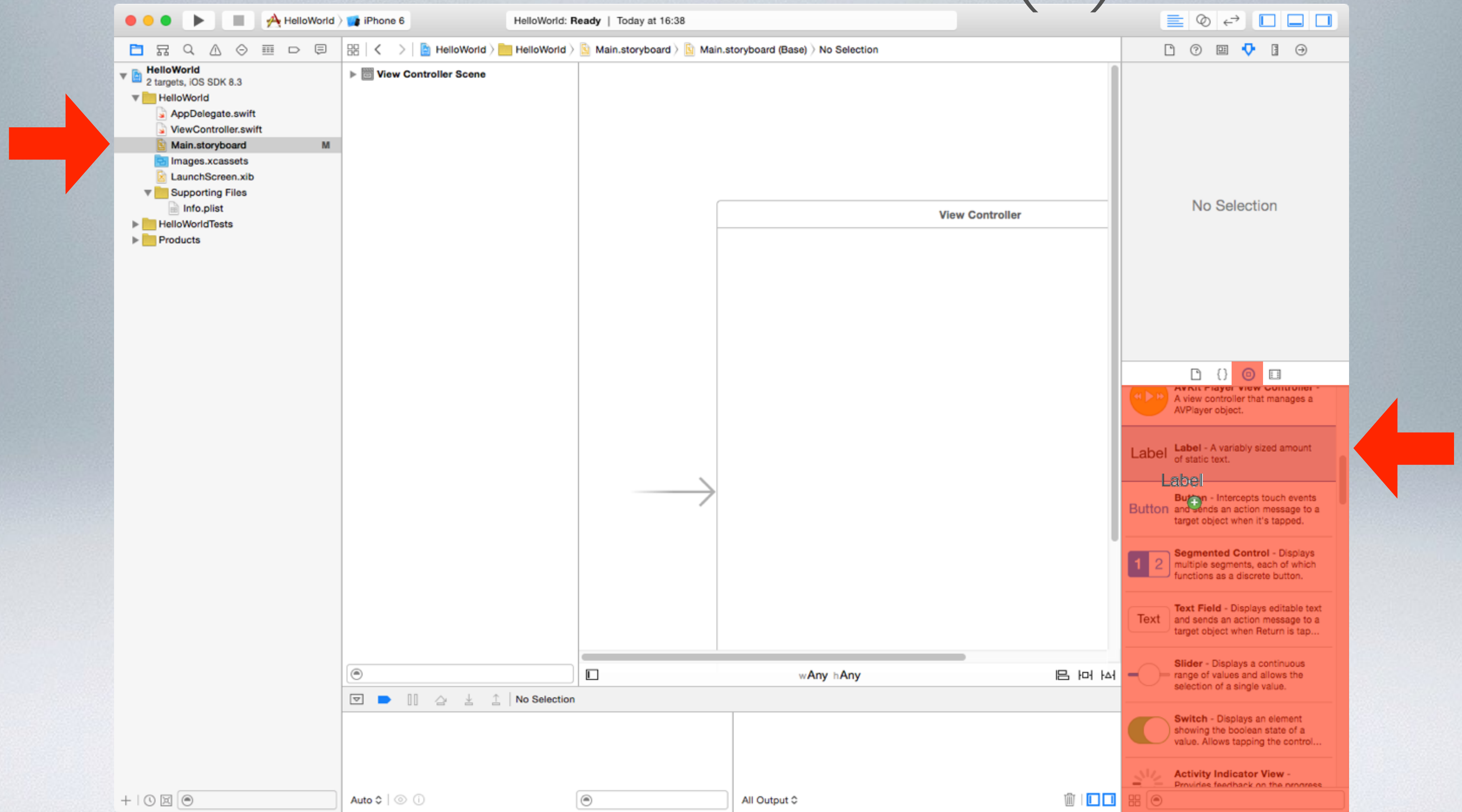


Erreur



```
sudo dscl . append /Groups/_developer GroupMembership <username>
```

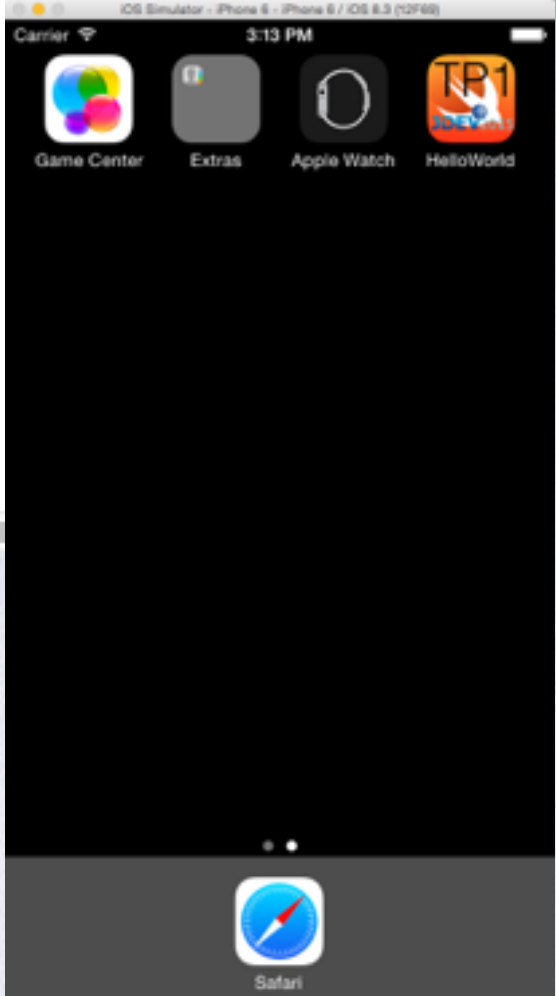
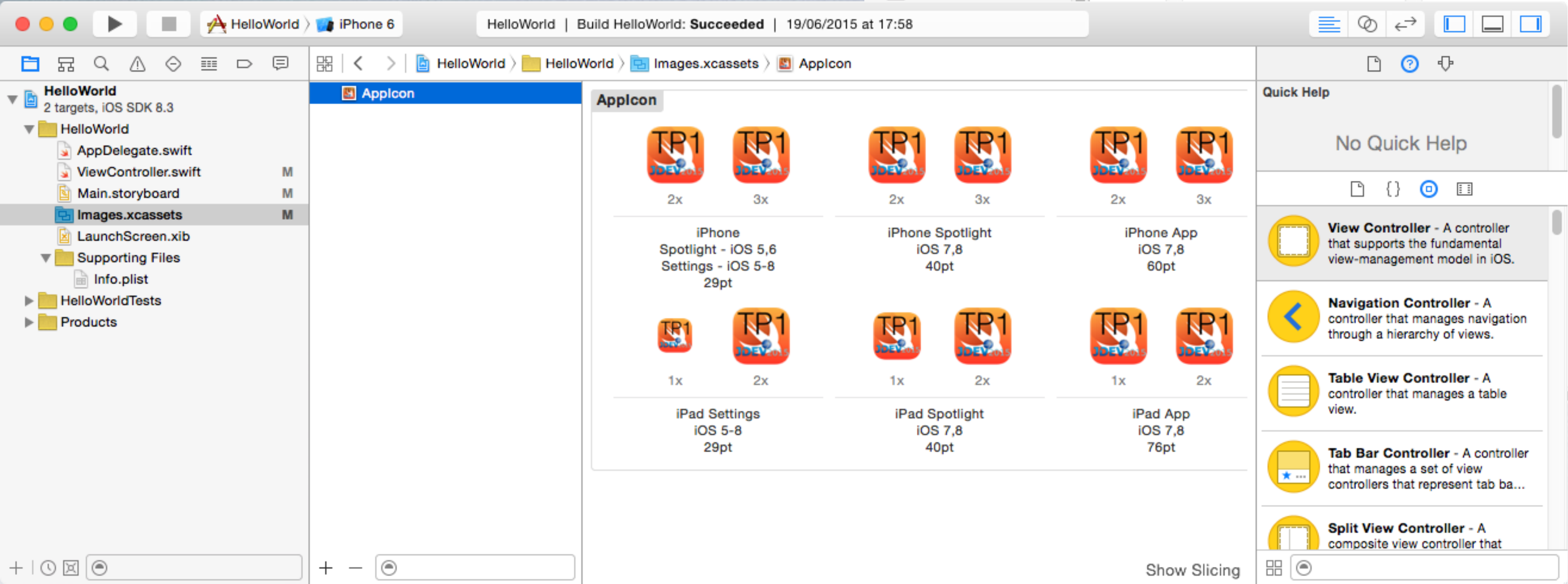
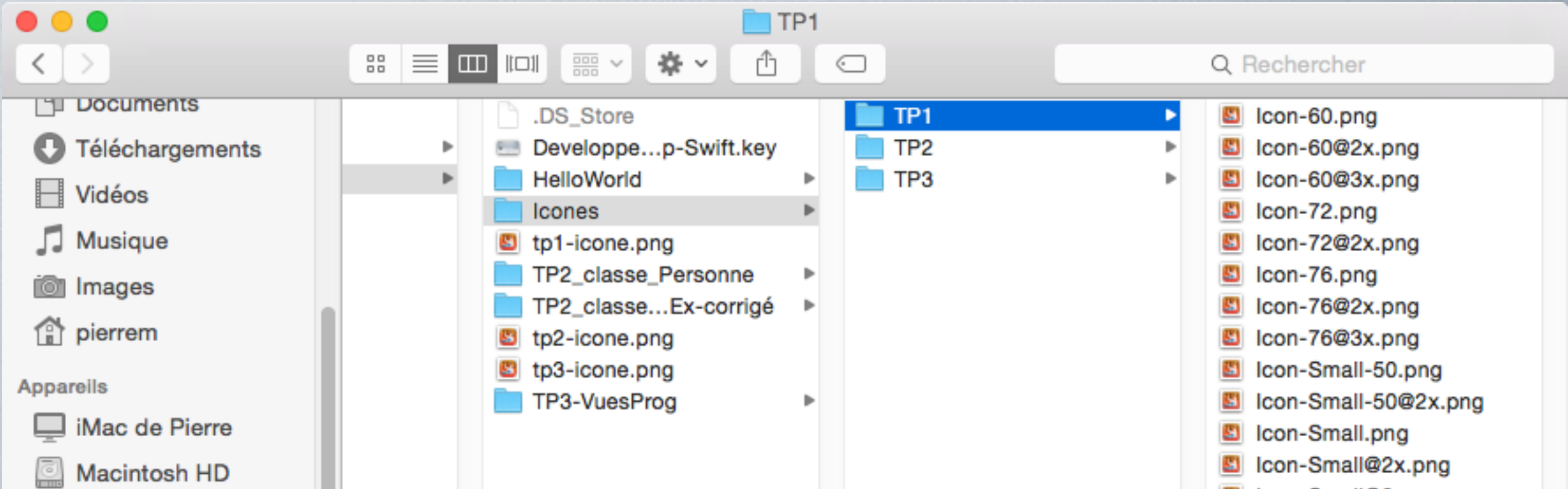

HELLOWORLD (I)



HELLOWORLD (2)



Prepo



INTRODUCTION (I)

- Langage fonctionnel et objet
- Pas de séparateur d'instruction (comme ; en C)
- Types de base: String, Int, Float, Double, Bool
- 2 modes pour les variables: immuable (let) ou modifiable (var)

- `let a=2` `var b=3`

- Déclaration de méthode:
 - `func maMethode(n:Int)->Int`
 - `func maMethode()`

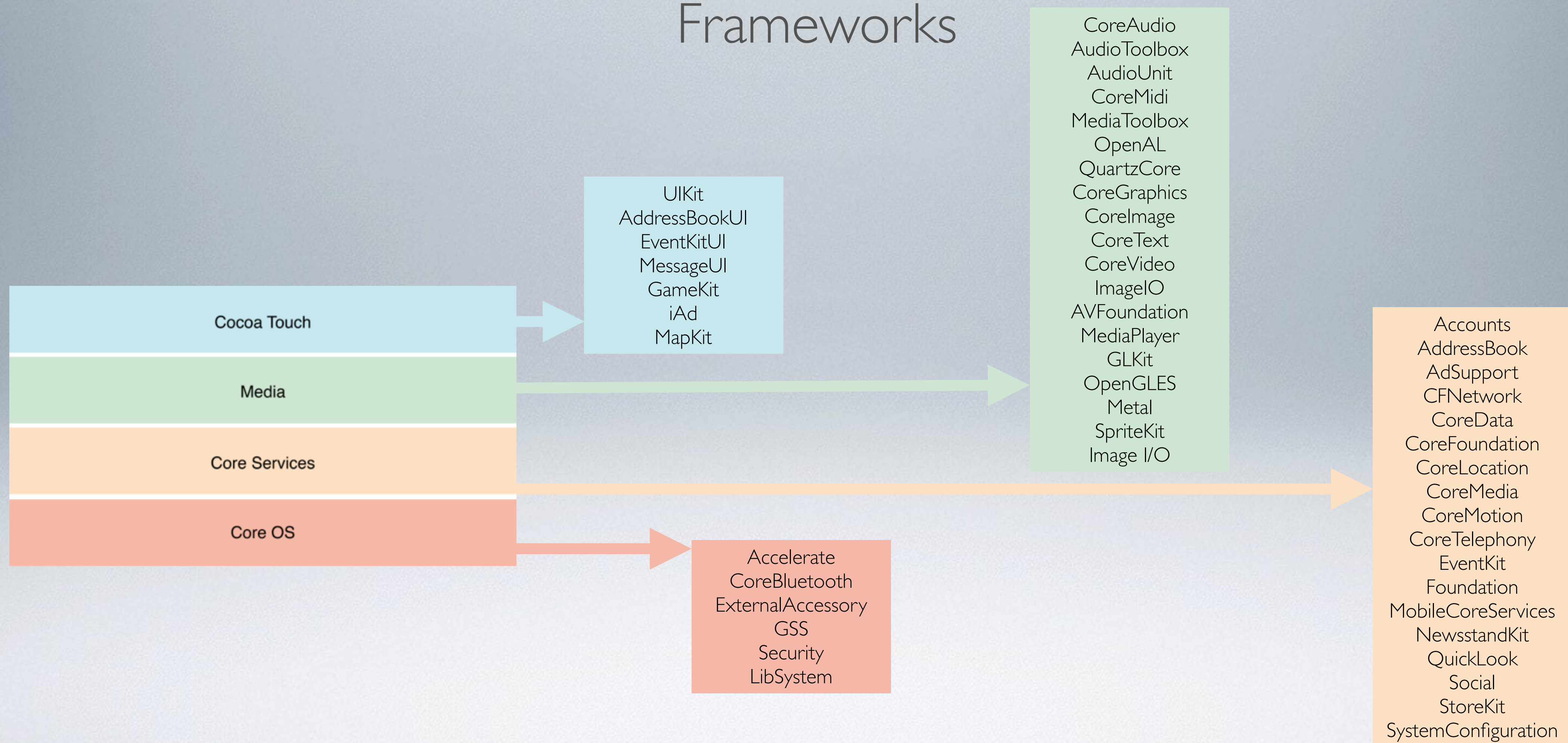
- Appel de méthode:

`monObjet.maMethode(param)`



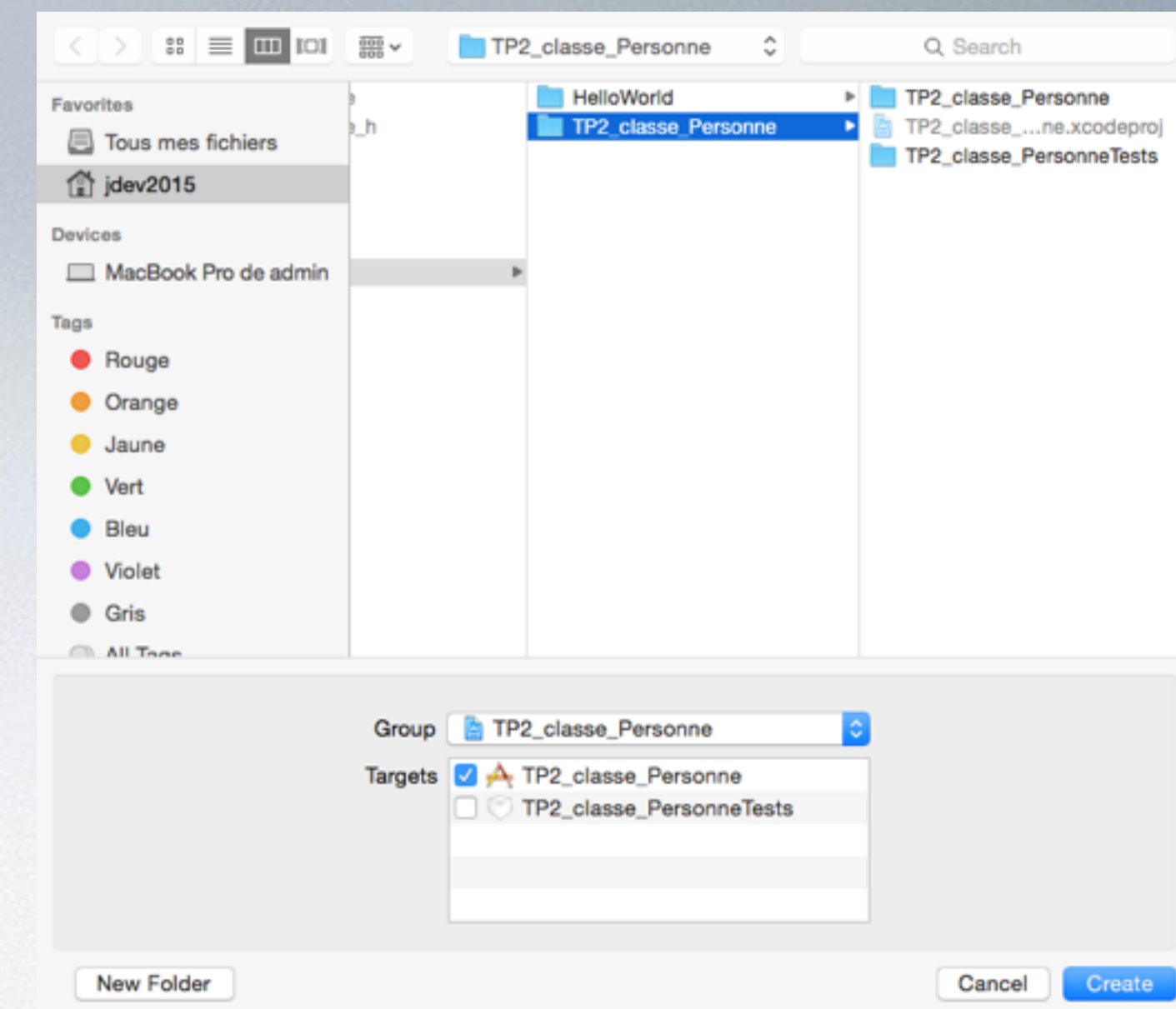
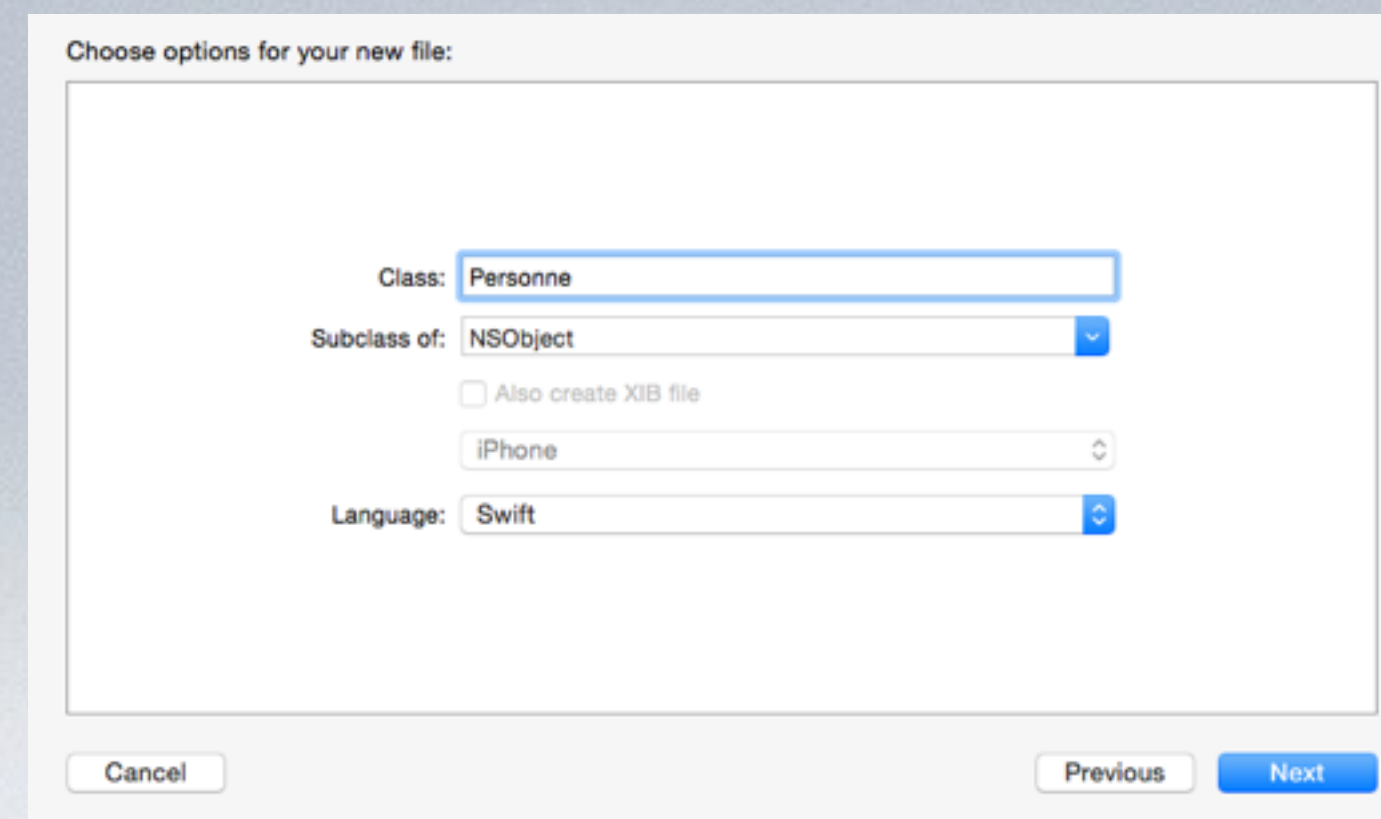
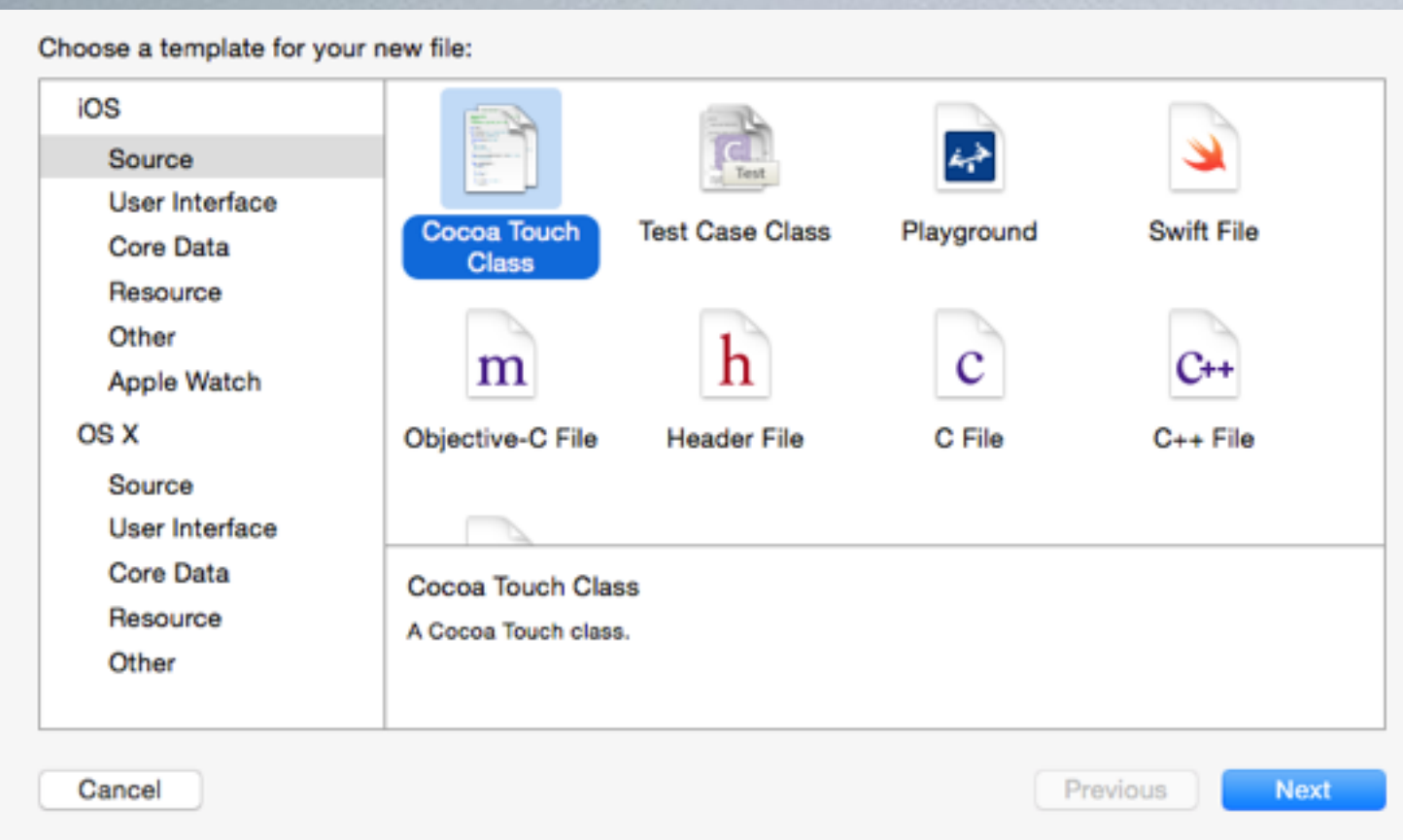
INTRODUCTION (2)

Frameworks



TP2: LES CLASSES

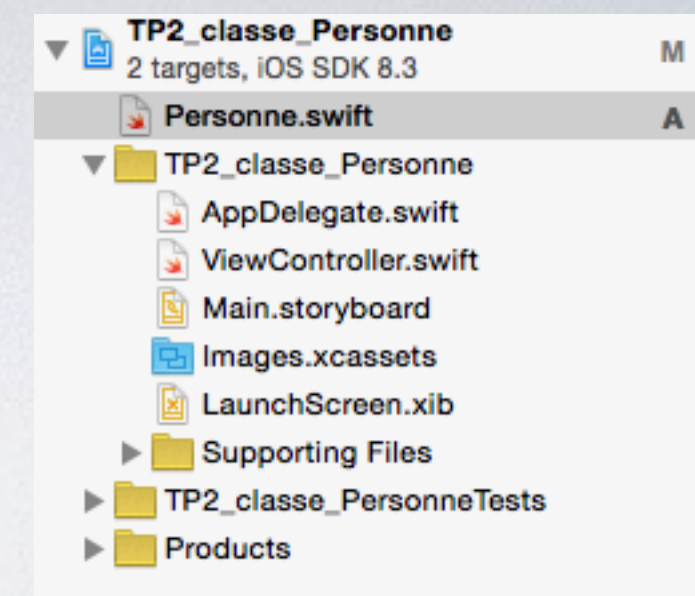
Création d'une classe (I)



TP2: LES CLASSES

Création d'une classe (2)

```
//  
//  Personne.swift  
//  TP2_classe_Personne  
//  
//  Created by Pierre Emmanuel Aguera on 19/06/2015.  
//  Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.  
//  
  
import UIKit  
  
class Personne: NSObject {  
  
}
```



TP2: LES CLASSES

Création d'une classe (3)

```
// Personne.swift
// TP2_classe_Personne
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//
import UIKit

class Personne: NSObject {
    // Attributs publics
    var prenom:String=""
    var nom:String

    // Attribut privé
    private var nomCompleter:String
    private let separateur=" "

    override init() {
        nom = ""
        nomCompleter = ""

        super.init()
    }

    func nomCompleterMeth()->String {
        nomCompleter = "\(prenom)\(separateur)\(nom)"
        println("Nom complet = \(nomCompleter)")
        return "\(prenom)\(separateur)\(nom)"
    }
}
```



TP2: LES CLASSES

Éléments de syntaxe de Swift (I) Types

Enumérés:

```
enum Couleur {case trefle, coeur, carreau, pique}  
let carte1 = Couleur.trefle  
let carte2 = .carreau  
enum Planete:Int {case Mercure=1,Venus,Terre,Mars}  
let a = Planete.fromRaw(2)    // a = Planete.Venus  
let b = Planete.Terre.toRaw() // b = 3
```

Dictionnaires:

```
var dic1 = [String:String]()  
var dic2: [Int:String] = [1:"un", 2:"Deux", 3:"trois"]  
for (cle, val) in dic2  
for cle in dic2.keys  
for val in dic2.values
```

Tableaux:

```
var t1 = [1, 2, 3]  
t.append(4)  
t.removeAtIndex(1)  
var t2 = [Int]()  
var t3 = [Double](count: 3, repeatedValue: 0.0)  
var taille = t1.count  
if t1.isEmpty {  
}  
var t4=t1+t3  
for val in t4 {  
}
```



TP2: LES CLASSES

Éléments de syntaxe de Swift (2)

Optionals

```
var chaine:String?
```

```
var val:Int?
```

```
func maFonction(val:Int)->String?
```

Comparaison:

comparaison de valeurs

==

!=

= affectation

comparaison de références

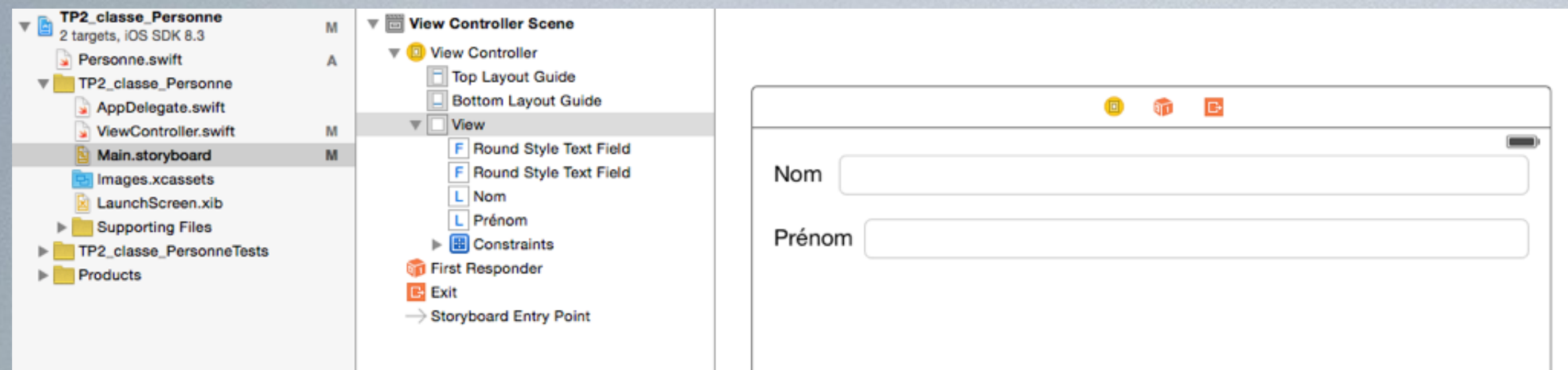
===

!==



TP2: LES CLASSES

Interactions UI/code (I)



IBOutlet : référence aux objets graphiques

```
// ViewController.swift
// TP2_classe_Personne
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var prenomTextField:UITextField?
    @IBOutlet weak var nomTextField:UITextField?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

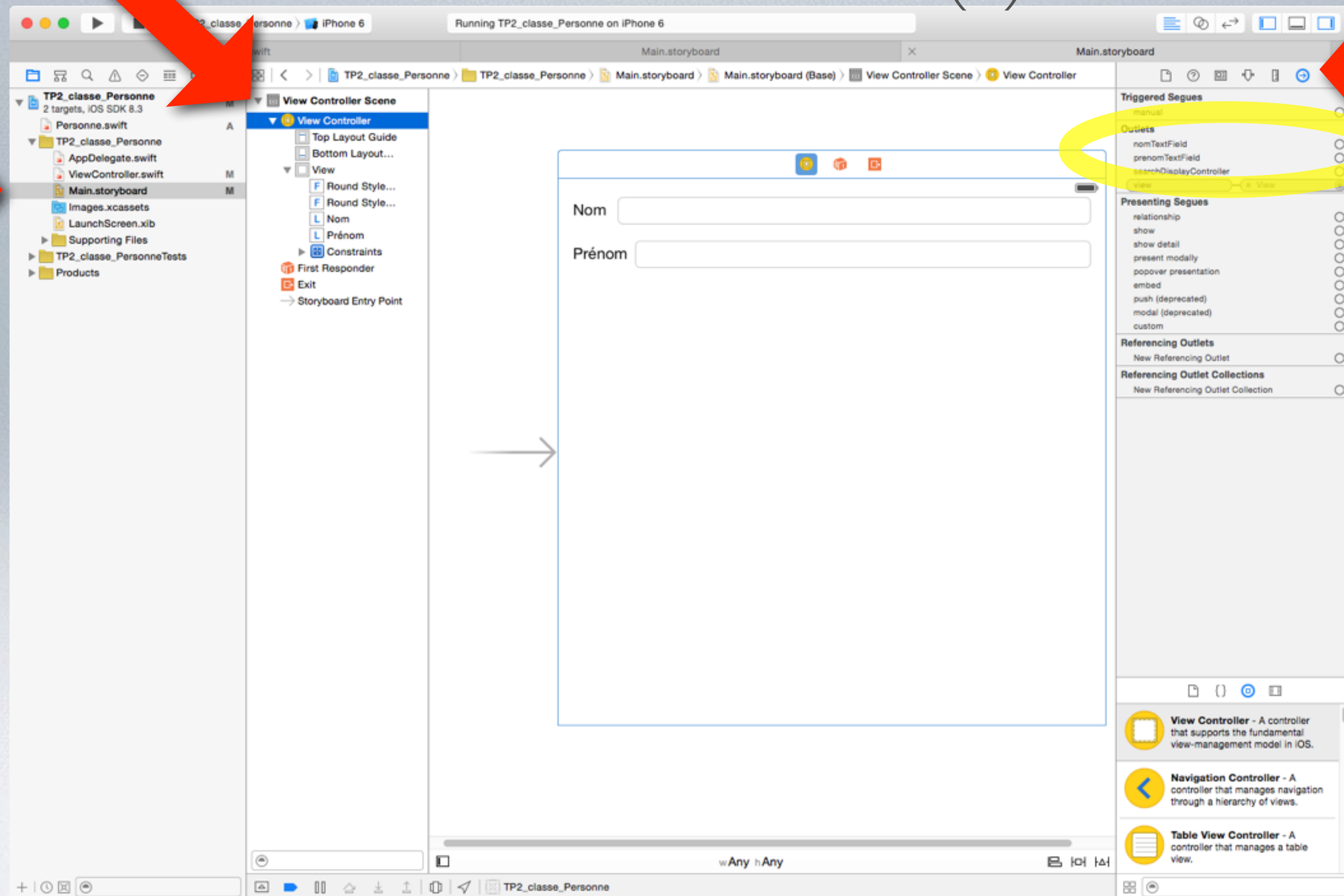
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```


TP2: LES CLASSES

Interactions UI/code (2)

2

1

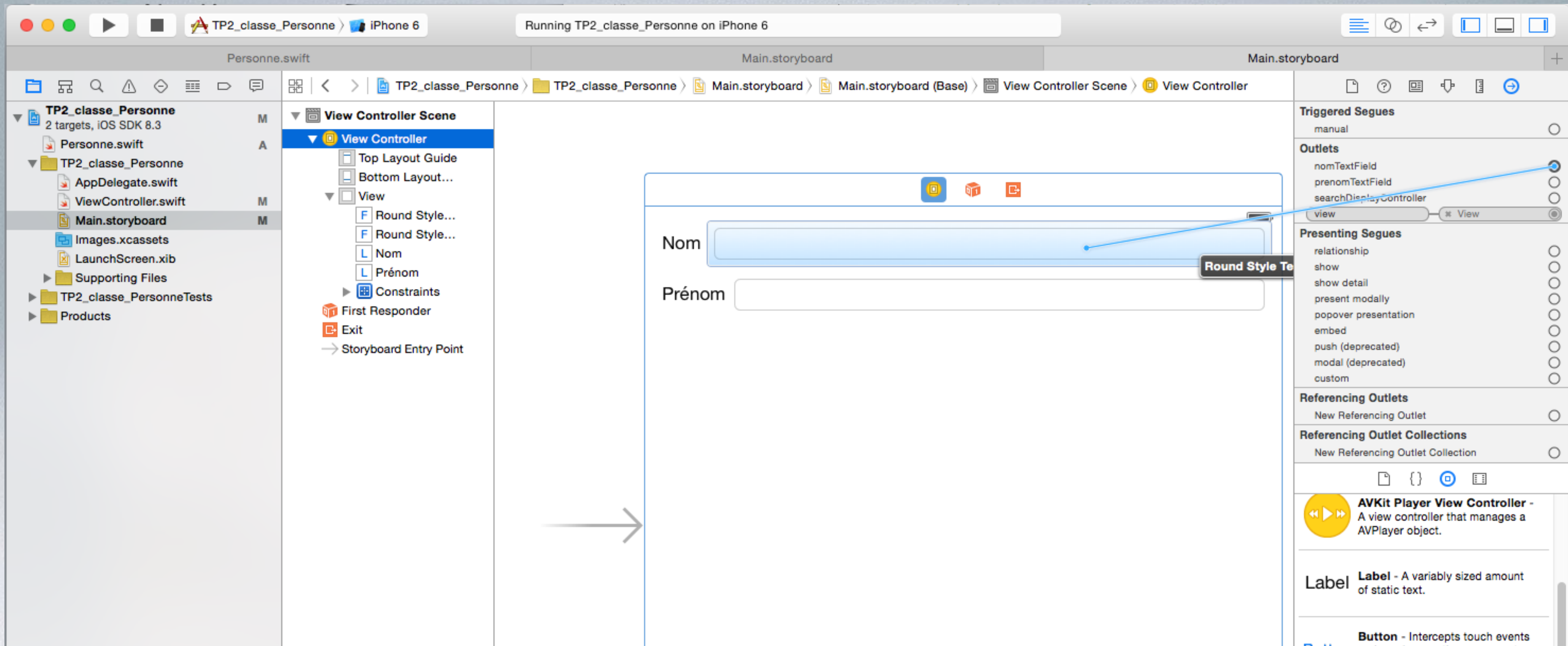


3



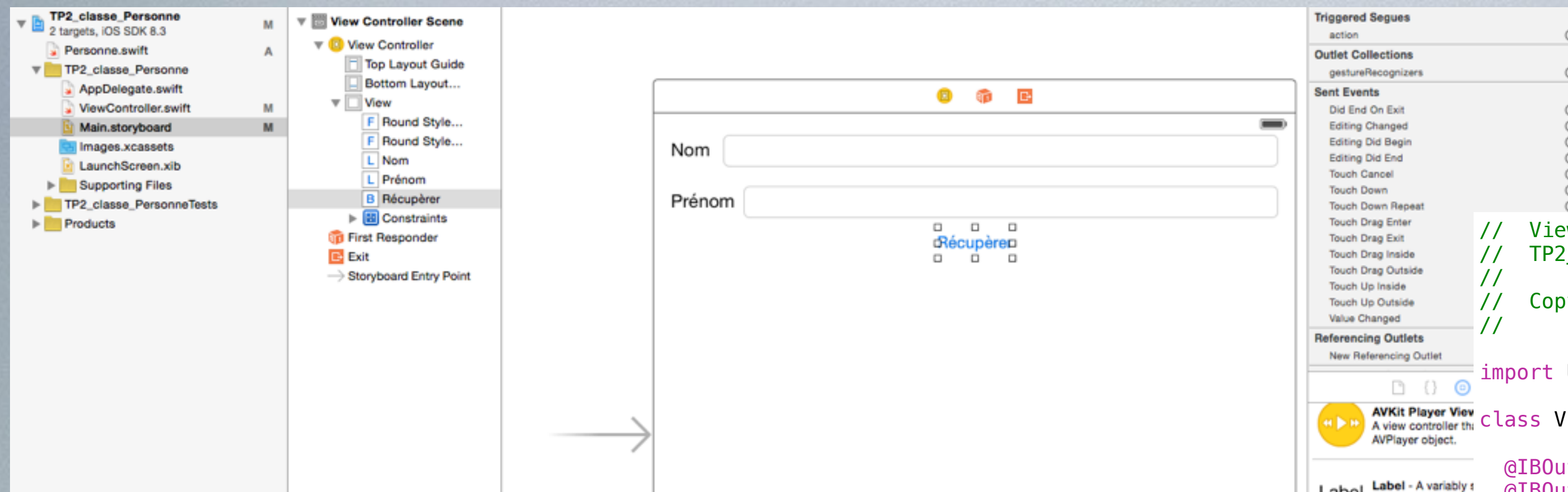
TP2: LES CLASSES

Interactions UI/code (3)



TP2: LES CLASSES

Interactions UI/code (4)



```
// ViewController.swift
// TP2_classe_Personne
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//
```

```
import UIKit
```

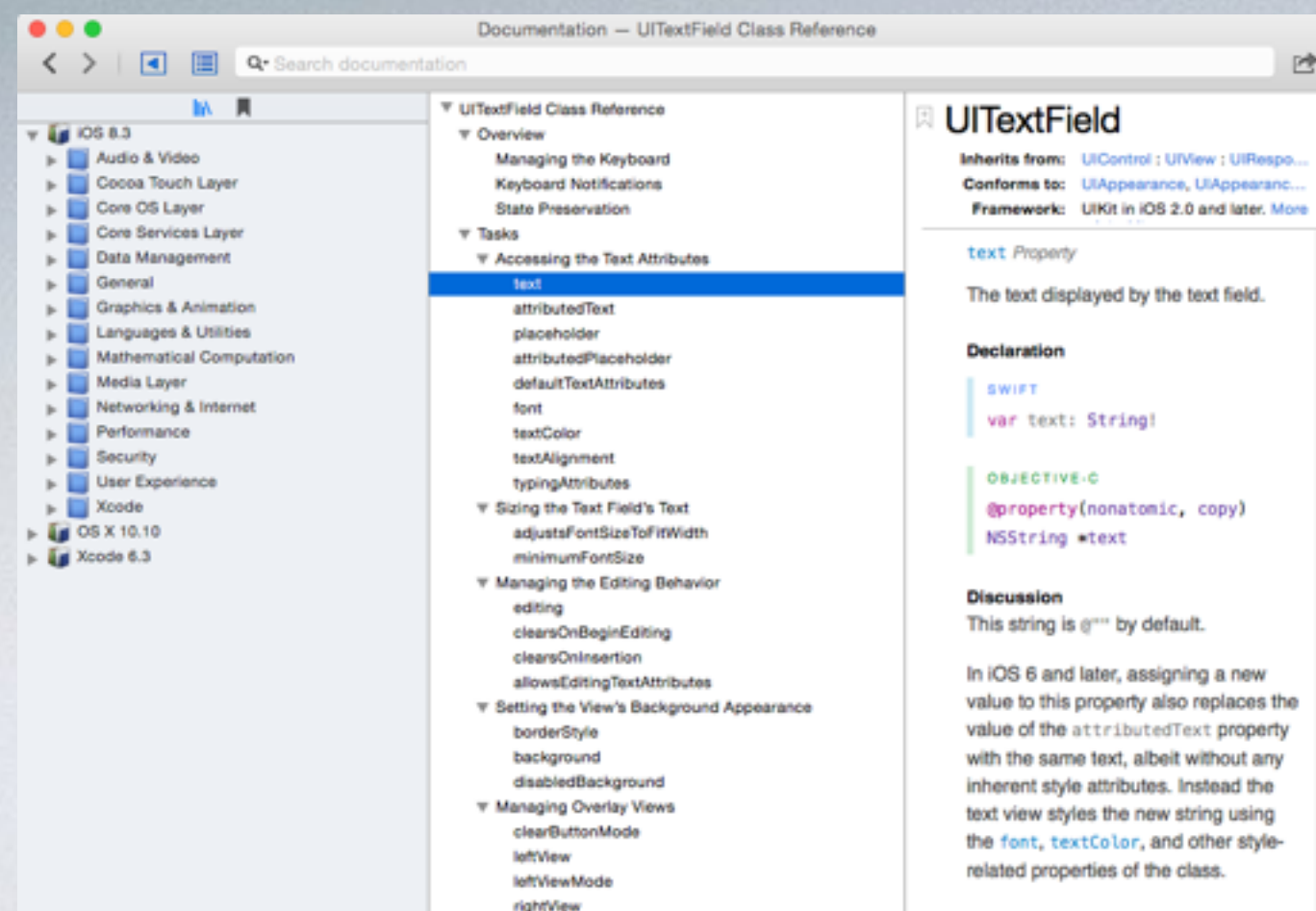
```
class ViewController: UIViewController {
```

```
    @IBOutlet weak var prenomTextField:UITextField?
    @IBOutlet weak var nomTextField:UITextField?
```

```
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }
```

```
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
```

```
@IBAction func recupTextes(sender: AnyObject) {
    let nom = nomTextField!.text
    let prenom = prenomTextField!.text
    println("Le nom est \(nom) et le prénom est \(prenom)")
}
```

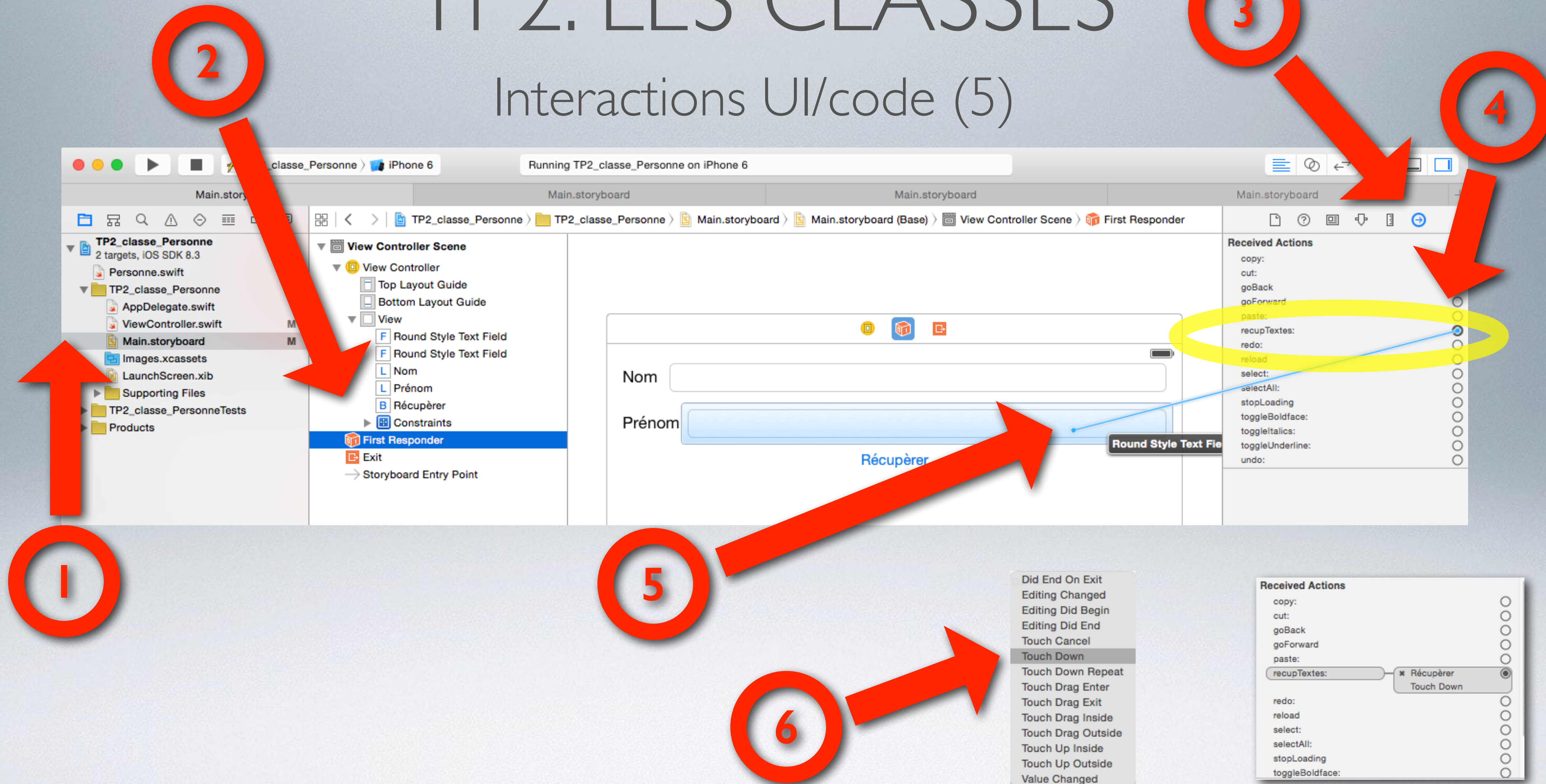


IBAction : callback ou méthode



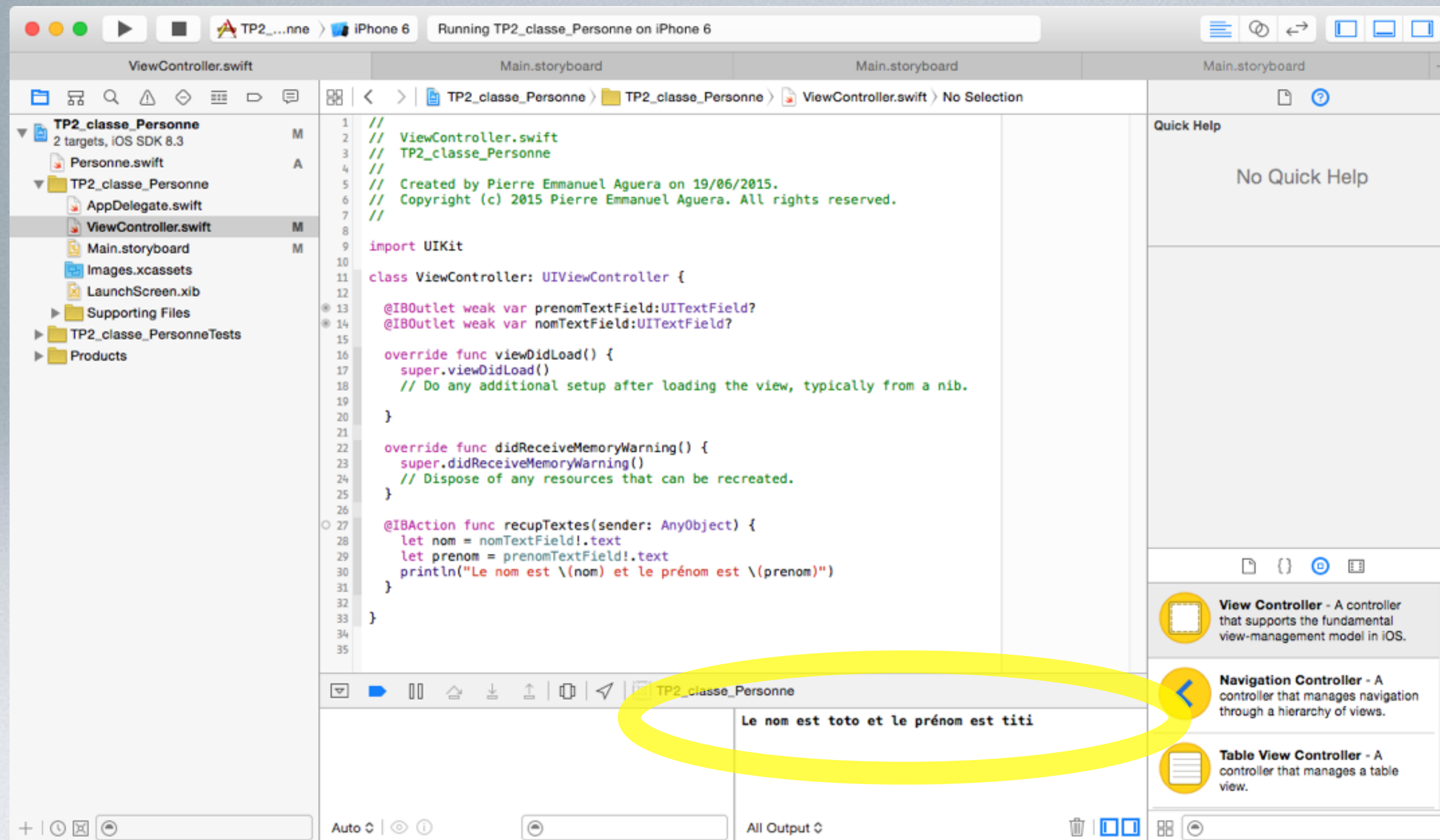
TP2: LES CLASSES

Interactions UI/code (5)



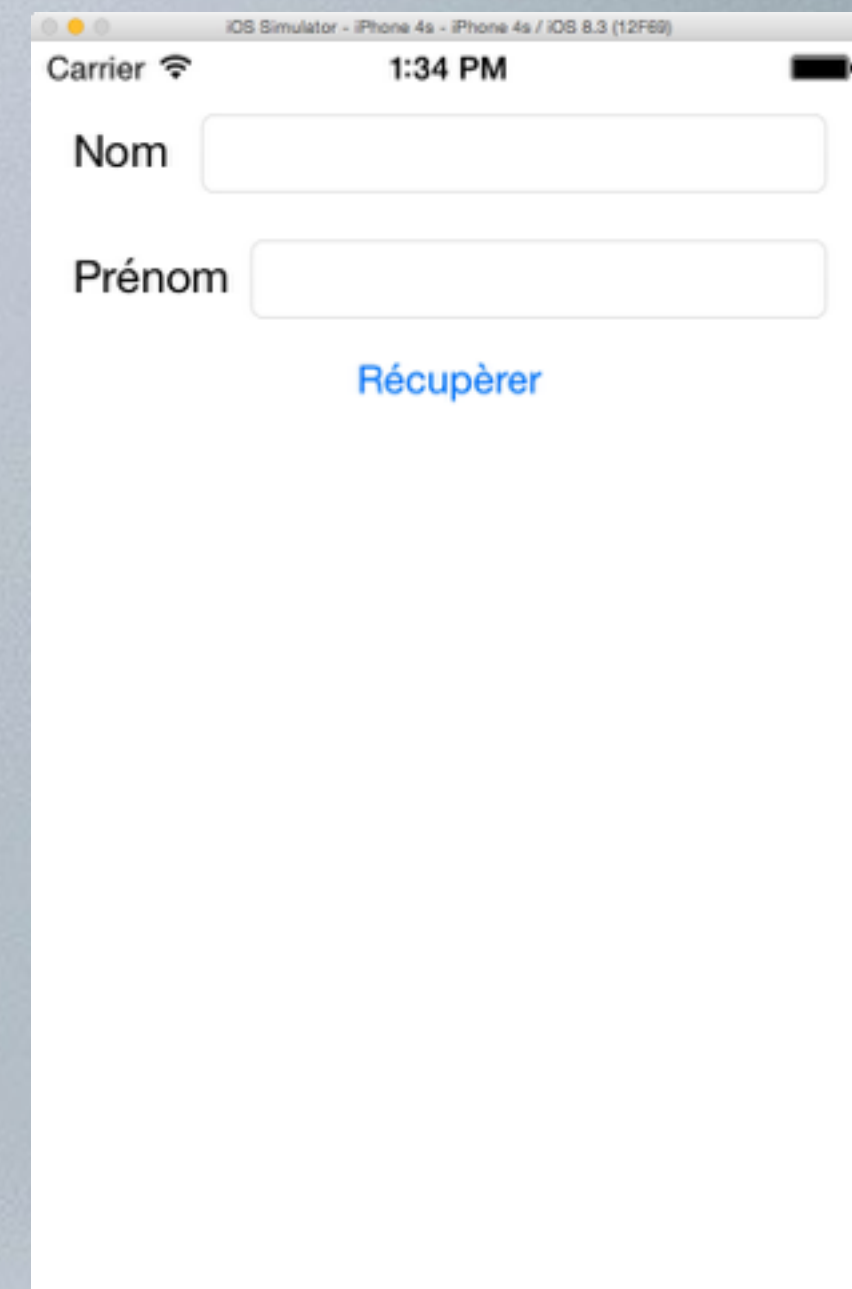
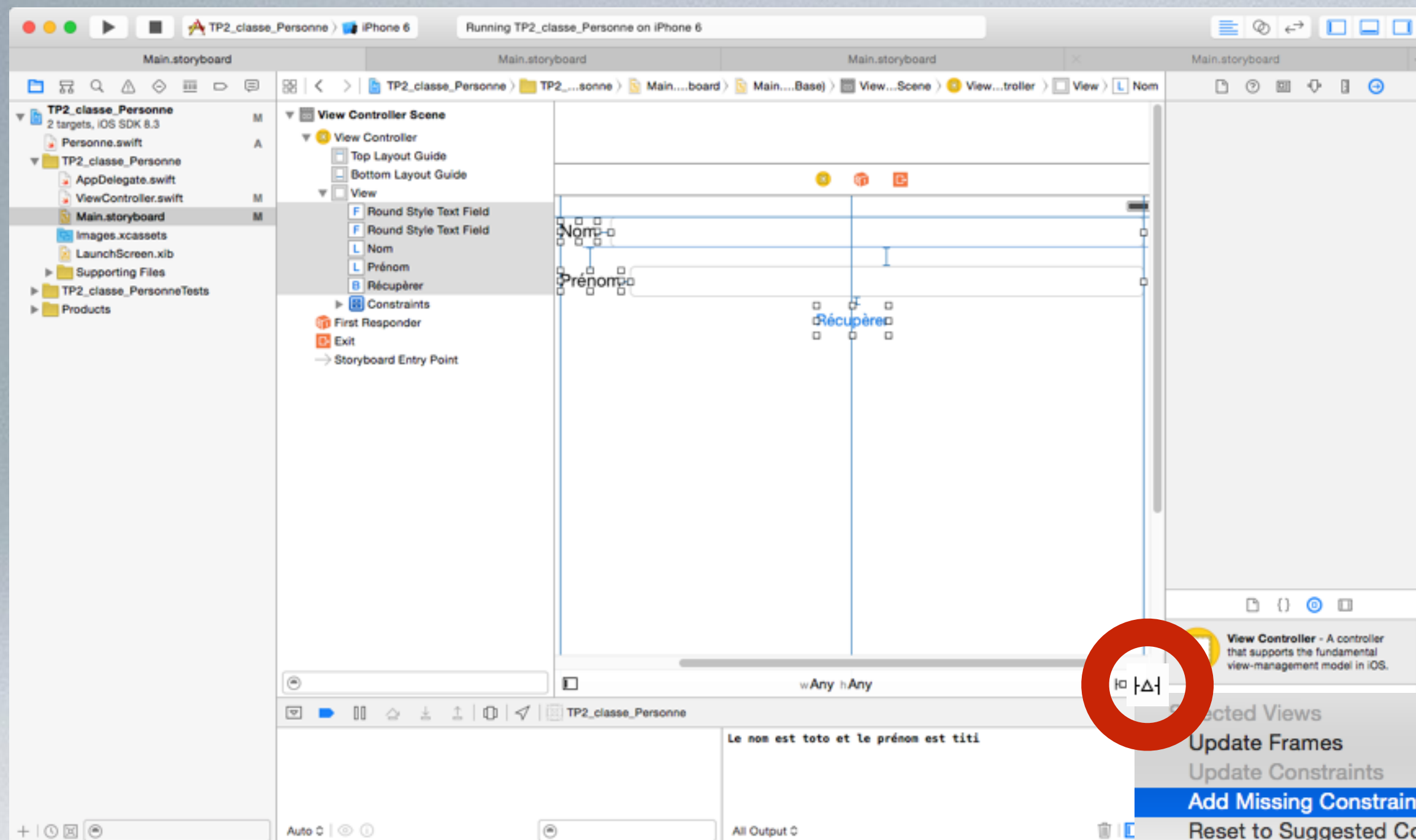
TP2: LES CLASSES

Interactions UI/code (6)



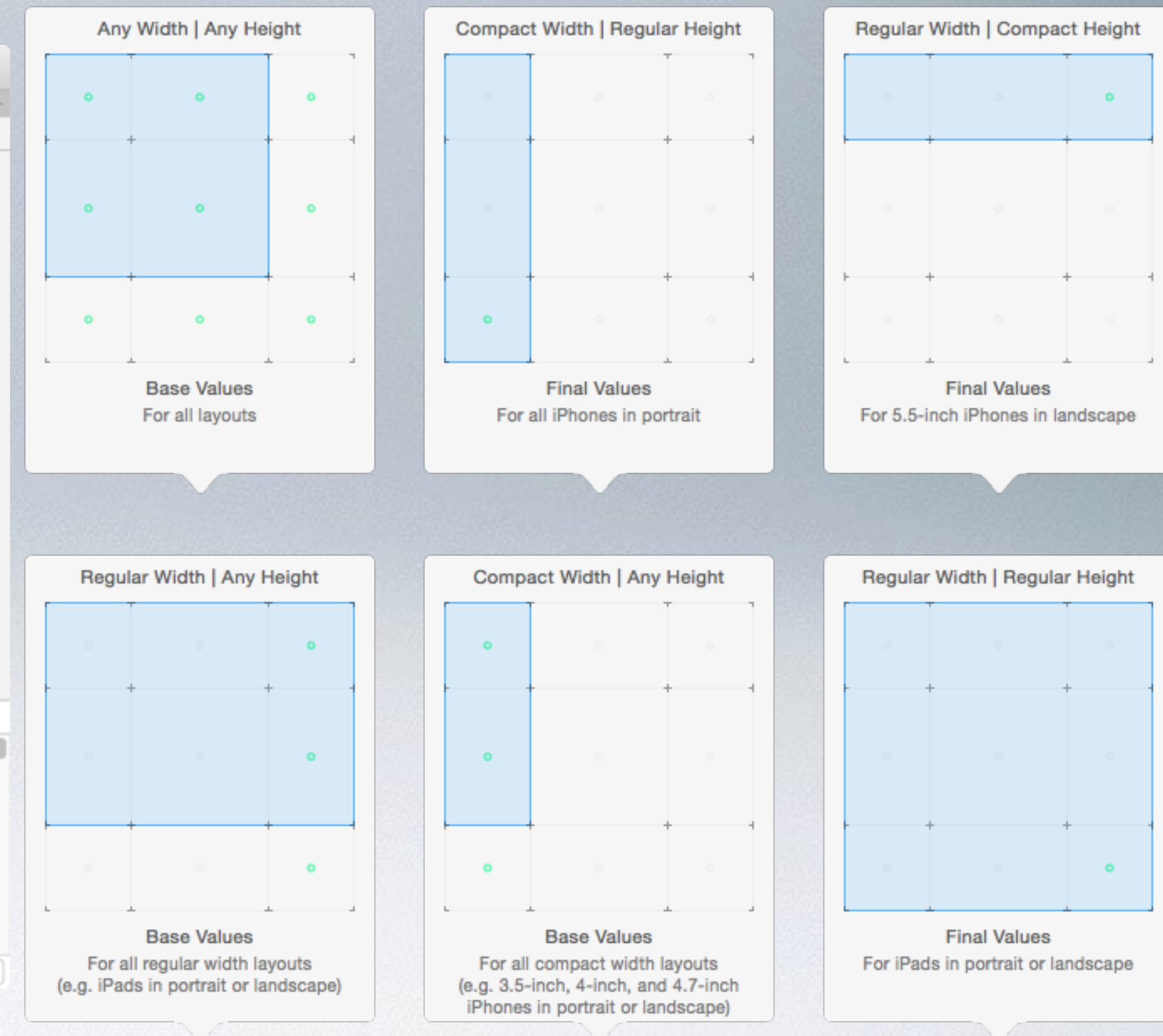
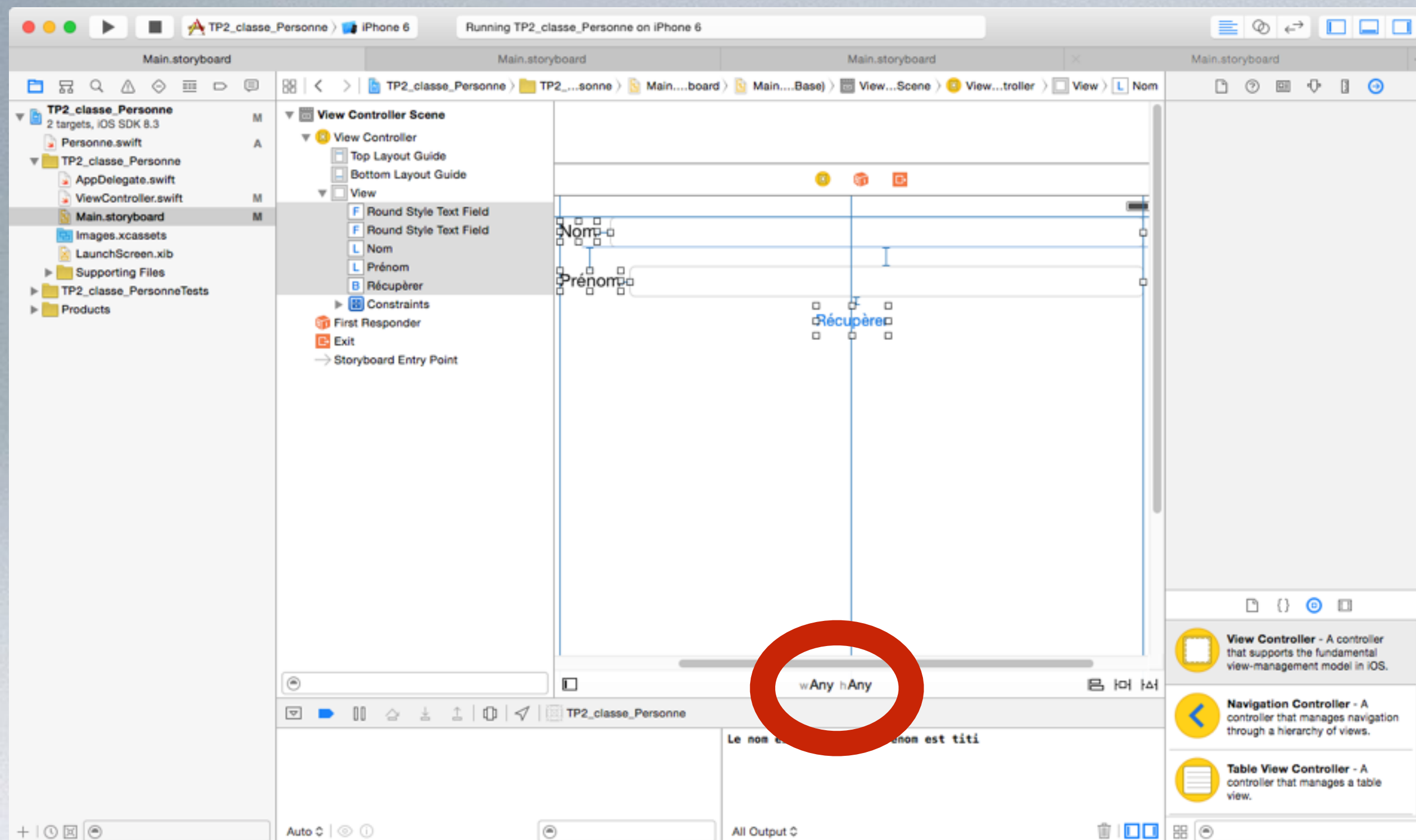
TP2: LES CLASSES

Compléments Storyboard (I)



TP2: LES CLASSES

Compléments Storyboard (2)



TP2: LES CLASSES

Création d'une classe (4)

```
// ViewController.swift
// TP2_classe_Personne
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var prenomTextField:UITextField?
    @IBOutlet weak var nomTextField:UITextField?
    @IBOutlet weak var nomCompleLabel:UILabel?

    private var unePersonne=Personne()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBAction func recupTextes(sender: AnyObject) {
        unePersonne.nom = nomTextField!.text
        unePersonne.prenom = prenomTextField!.text
        nomCompleLabel!.text = unePersonne.nomCompleMeth()
    }

}
```



TP2: LES CLASSES

Utilisation du delegate (I)

```
//  
// ViewController.swift  
// TP2_classe_Personne  
//  
// Created by Pierre Emmanuel Aguera on 19/06/2015.  
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.  
//  
  
import UIKit  
  
class ViewController: UIViewController, UITextFieldDelegate {  
    @IBOutlet weak var prenomTextField:UITextField?  
    @IBOutlet weak var nomTextField:UITextField?  
    @IBOutlet weak var nomCompleLabel:UILabel?  
    ...  
}
```



TP2: LES CLASSES

Utilisation du delegate (2)



UITextFieldDelegate Protocol Reference

Overview

Tasks

Managing Editing

textFieldShouldBeginEditing:

textFieldDidBeginEditing:

textFieldShouldEndEditing:

textFieldDidEndEditing:

Editing the Text Field's Text

textField:shouldChangeCharactersInRange:replac...

textFieldShouldClear:

textFieldShouldReturn:

Revision History

UITextFieldDelegate

Inherits from: None

Conforms to: NSObject

Framework: UIKit in iOS 2.0 and later. [More related](#)

The UITextFieldDelegate protocol defines the messages sent to a text field delegate as part of the sequence of editing its text. All of the methods of this protocol are optional.

UITextFieldDelegate

Inherits from: None

Conforms to: NSObject

Framework: UIKit in iOS 2.0 and later. [More related items...](#)

- textFieldDidEndEditing:

Tells the delegate that editing stopped for the specified text field.

Declaration

SWIFT

```
optional func textFieldDidEndEditing(_ textField: UITextField)
```

OBJECTIVE-C

```
-(void)textFieldDidEndEditing:(UITextField *)textField
```

Parameters

textField The text field for which editing ended.

Discussion

This method is called after the text field resigns its first responder status. You can use this method to update your delegate's state information. For example, you might use this method to hide overlay views that should be visible only while editing.

Implementation of this method by the delegate is optional.

Import Statement

OBJECTIVE-C

```
@import UIKit;
```

SWIFT

```
import UIKit
```

Availability

Available in iOS 2.0 and later.



TP2: LES CLASSES

Utilisation du delegate (3)

```
//
// ViewController.swift
// TP2_classe_Personne
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//
import UIKit

class ViewController: UIViewController, UITextFieldDelegate {

    @IBOutlet weak var prenomTextField:UITextField?
    @IBOutlet weak var nomTextField:UITextField?
    @IBOutlet weak var nomCompleLabel:UILabel?

    private var unePersonne=Personne()

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        nomTextField!.delegate = self
        prenomTextField!.delegate = self
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    ...
}
```



```
...

@IBAction func recupTextes(sender: AnyObject) {
    unePersonne.nom = nomTextField!.text
    unePersonne.prenom = prenomTextField!.text
    nomCompleLabel!.text = unePersonne.nomCompleMeth()
}

func textFieldDidEndEditing(textField: UITextField) {
    if (textField === nomTextField) {
        unePersonne.nom = nomTextField!.text
    } else if (textField === prenomTextField) {
        unePersonne.prenom = prenomTextField!.text
    } else {
        println("UITextField inconnu")
    }

    nomCompleLabel!.text = unePersonne.nomCompleMeth()
}

}
```


TP2: LES CLASSES

Exercice

Ajouter une date de naissance au TP2

Ajouter un élément de sélection de date

Récupérer la date sélectionnée, l'afficher dans un label et la concaténer au label `nomCompletLabel`

Utiliser `UIDatePicker` pour choisir la date.

Utiliser l'événement *Value Changed* de `UIDatePicker` pour une `IBAction` et mise à jour des champs.



Carrier 4:58 PM

Nom aguera

Prénom pierre-emmanuel

[Récupérer](#)

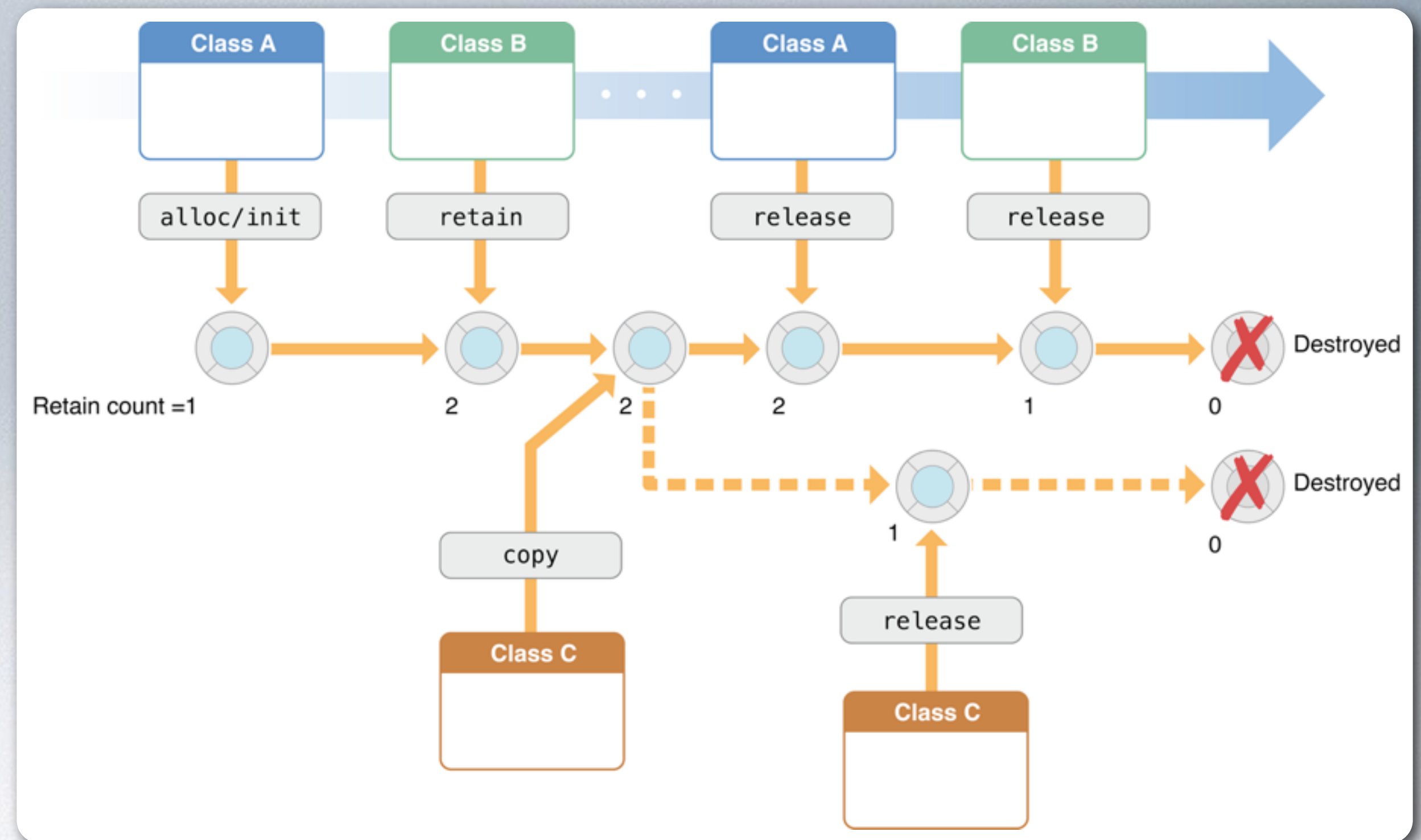
aguera pierre-emmanuel né le 2005-10-

July	15	2002
August	16	2003
September	17	2004
October	18	2005
November	19	2006
December	20	2007
January	21	2008
February	22	2009

2005-10-18 14:57:41 +0000

GESTION DE LA MÉMOIRE

Automatic Reference Counting ARC



GESTION DE LA MÉMOIRE

Automatic Reference Counting ARC (I)

```
class Personne {  
    let nom: String  
    init(nom: String) {  
        self.nom = nom  
        println("\(nom) en cours d'initialisation")  
    }  
    deinit {  
        println("\(nom) en cours de déinitialisation")  
    }  
}
```

```
var reference1: Personne?  
var reference2: Personne?  
var reference3: Personne?
```

```
reference1 = Personne(nom: "Paul Durand")
```

Paul Durand en cours d'initialisation

```
reference2 = reference1  
reference3 = reference1
```

```
reference1 = nil  
reference2 = nil
```

```
reference3 = nil
```

Paul Durand en cours de déinitialisation



GESTION DE LA MÉMOIRE

Automatic Reference Counting ARC (2)

```
class Personne {
    let nom: String
    var logement: Appartement?
    init(nom: String) { self.nom = nom }
    deinit { println("\(nom) en cours de déinitialisation") }
}
```

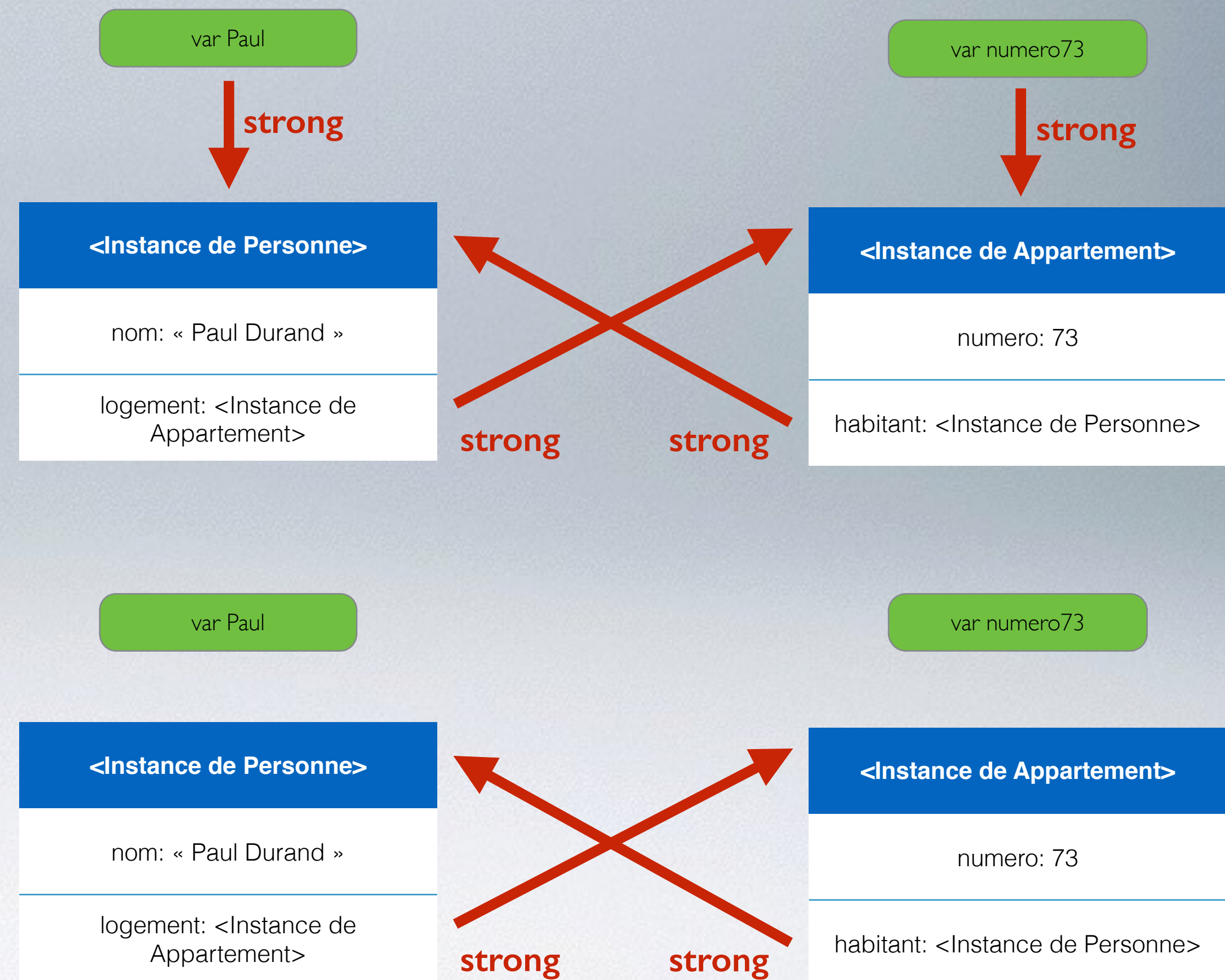
```
class Appartement {
    let numero: Int
    var habitant: Personne?
    init(numero: Int) { self.numero = numero }
    deinit { println("Appartement #\(numero) en cours de déinitialisation") }
}
```

```
var paul: Personne?
var numero73: Appartement?
```

```
paul = Personne2(nom: "Paul Durand")
numero73 = Appartement(numero: 73)
```

```
paul!.logement = numero73
numero73!.habitant = paul
```

```
paul = nil
numero73 = nil
```



GESTION DE LA MÉMOIRE

Automatic Reference Counting ARC (3)

```
class Personne {
    let nom: String
    var logement: Appartement?
    init(nom: String) { self.nom = nom }
    deinit { println("\(nom) en cours de déinitialisation") }
}

class Appartement {
    let numero: Int
    weak var habitant: Personne?
    init(numero: Int) { self.numero = numero }
    deinit { println("Appartement #\(numero) en cours de déinitialisation") }
}

var paul: Personne?
var numero73: Appartement?

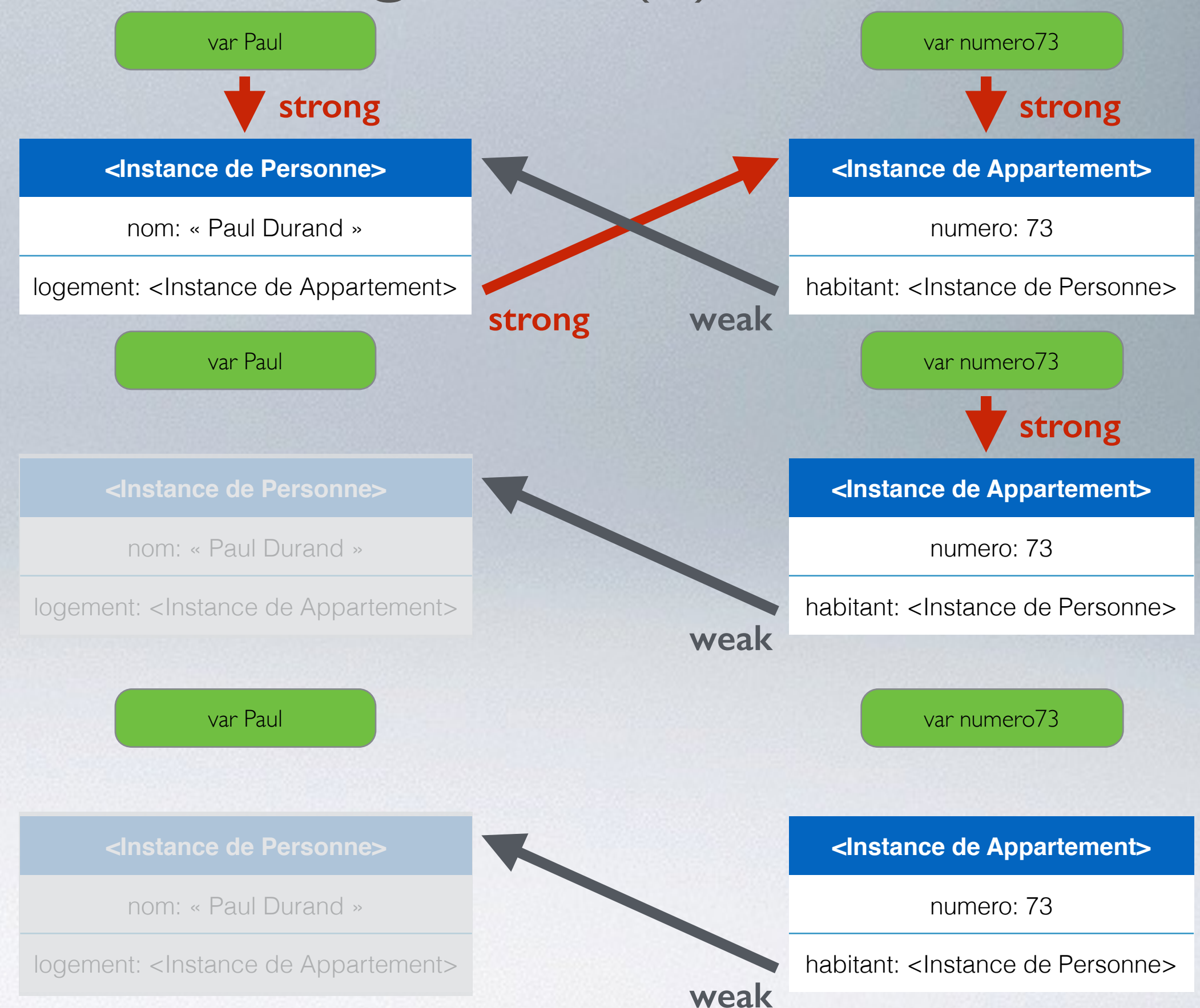
paul = Personne(nom: "Paul Durand")
numero73 = Appartement(numero: 73)

paul!.logement = numero73
numero73!.habitant = paul

paul = nil
numero73 = nil
```

Paul en cours de déinitialisation

Appartement #73 en cours de déinitialisation



GESTION DE LA MÉMOIRE

Automatic Reference Counting ARC (4)

```
class Client {
    let nom: String
    var carte: CarteCredit?
    init(nom: String) {
        self.nom = nom
    }
    deinit { println("\(nom) en cours de déinitialisation") }
}

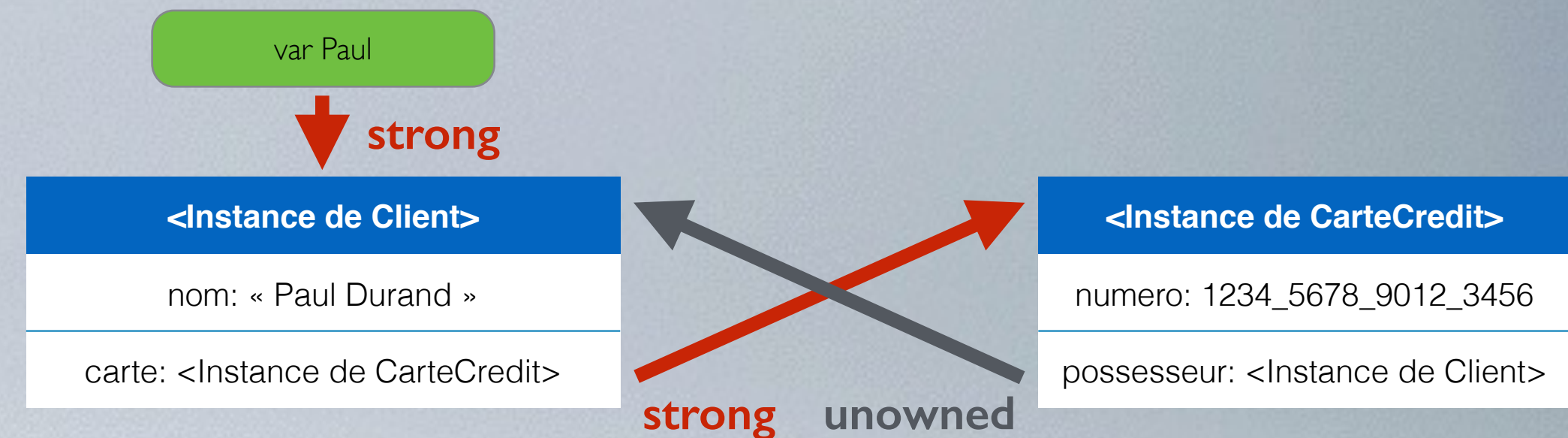
class CarteCredit {
    let numero: UInt64
    unowned let possesseur: Client
    init(numero: UInt64, possesseur: Client) {
        self.numero = numero
        self.possesseur = possesseur
    }
    deinit { println("Carte #\(numero) en cours de déinitialisation") }
}
```

```
var paul: Client?

paul = Client(nom: "Paul Durand")
paul!.carte = CarteCredit(numero: 1234_5678_9012_3456, possesseur: paul!)
```

```
paul = nil
```

Paul Durand en cours de déinitialisation
Carte #1234567890123456 en cours de déinitialisation



CONSTRUCTION DE VUE

Introduction

Éléments de base personnalisables:

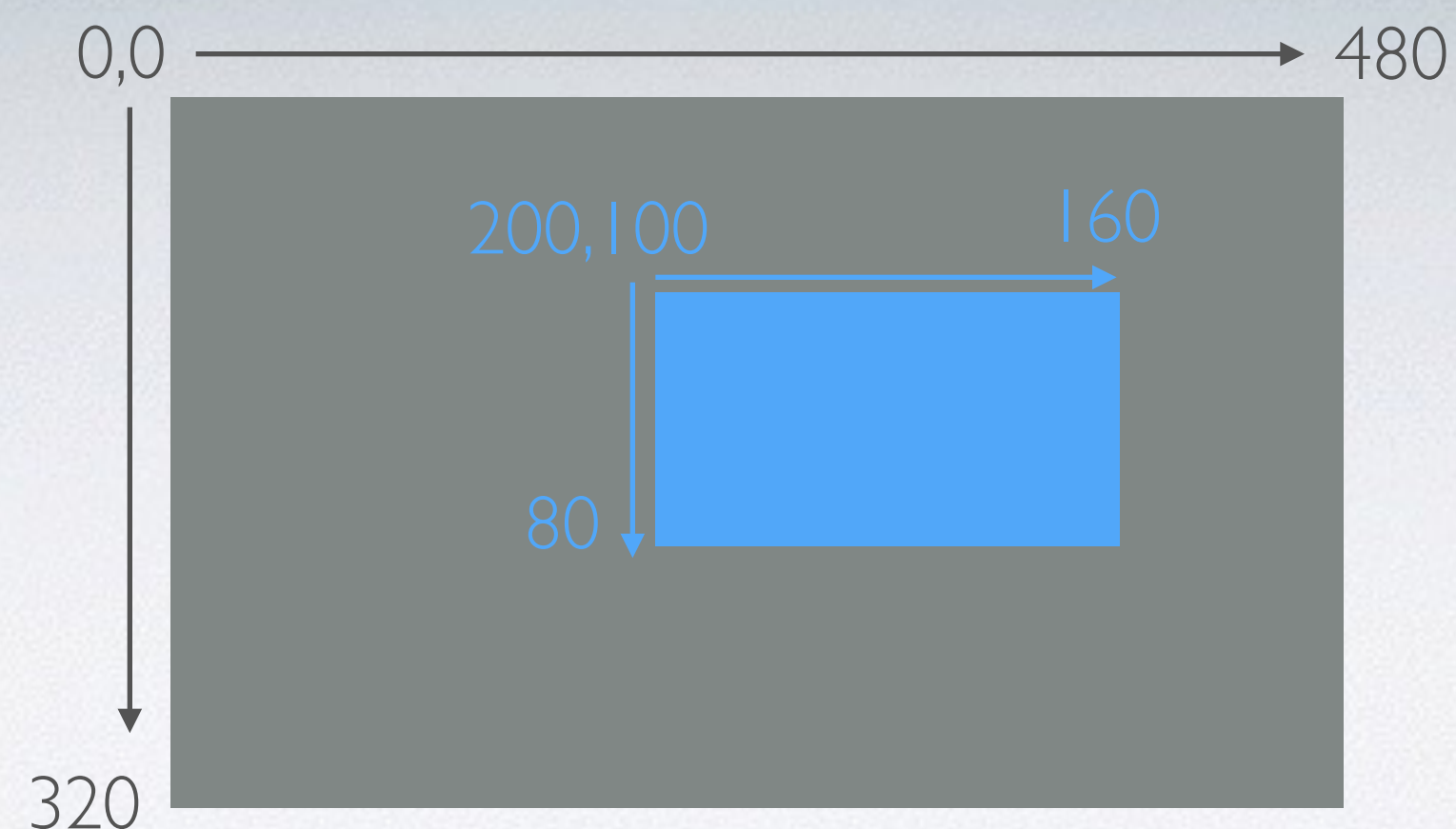
frame: position+coordonnées dans la vue parente

bounds : origine+taille

backgroundColor: couleur de fond (transparent par défaut clearColor)

hidden: booléen indiquant si elle est visible ou non

Les coordonnées sont liées à la vue englobante:



Vue A frame:

origine 0,0 taille 480,320

Vue A bounds:

origine 0,0 taille 480,320

Vue B frame:

origine 200,100 taille 160,80

Vue B bounds:

origine 0,0 taille 160,80



CONSTRUCTION DE VUE

Identification des terminaux

UIDevice

- nom du terminal (name)
- type (userInterfaceIdiom)
- version système (systemVersion)
- modèle (model, localizedModel)
- orientation courante (orientation)
- niveau et état de la batterie (batteryLevel, batteryState)
- existence et état du détecteur de proximité (iPhone)
- ...

UIScreen

- taille de l'écran (bounds)
- densité de l'écran (scale)
- luminosité (brightness)
- bloquer la diminution automatique de luminosité (wantsSoftwareDimming)

```
let terminal = UIDevice.currentDevice()    let screen = UIScreen.mainScreen()
```



CONSTRUCTION DE VUE

Gestion de l'orientation

UIViewController

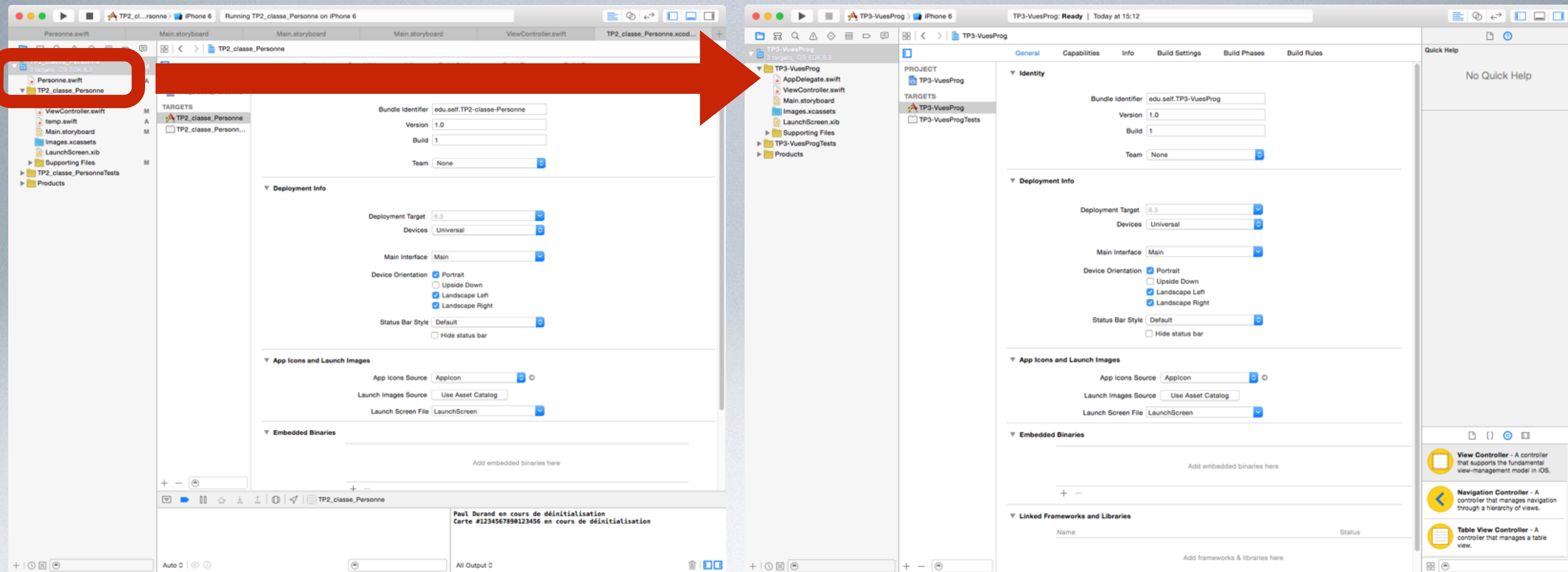
```
func shouldAutorotate() -> Bool
func supportedInterfaceOrientations()-> Int
func viewWillTransitionToSize(size: CGSize, withTransitionCoordinator coordinator: UIViewControllerTransitionCoordinator)

func willRotateToInterfaceOrientation(toInterfaceOrientation: UIInterfaceOrientation,
duration duration: NSTimeInterval) (dépréciée depuis iOS8.0)
```



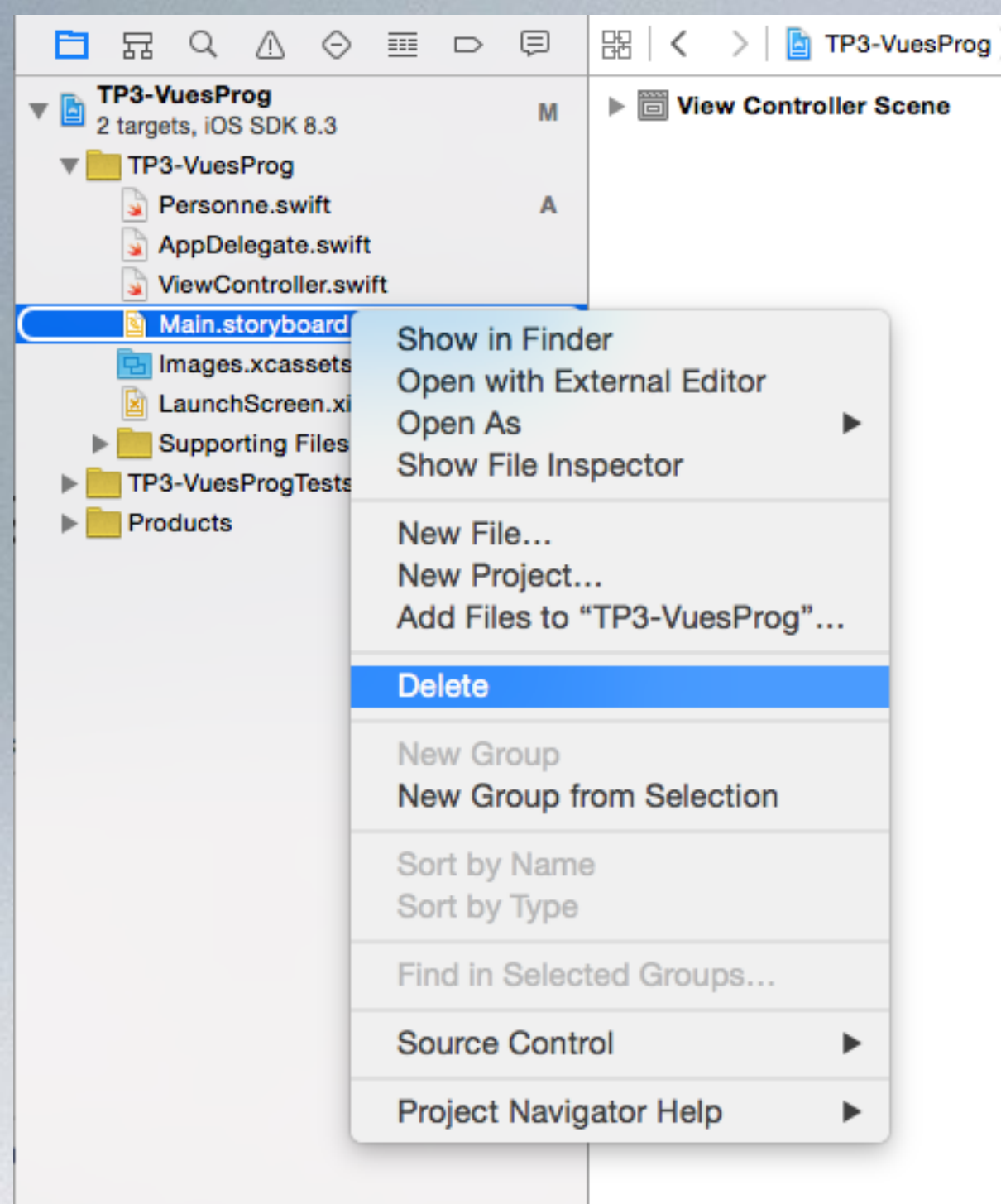
CONSTRUCTION DE VUE

En pratique: TP3 (I)



CONSTRUCTION DE VUE

En pratique: TP3 (2)



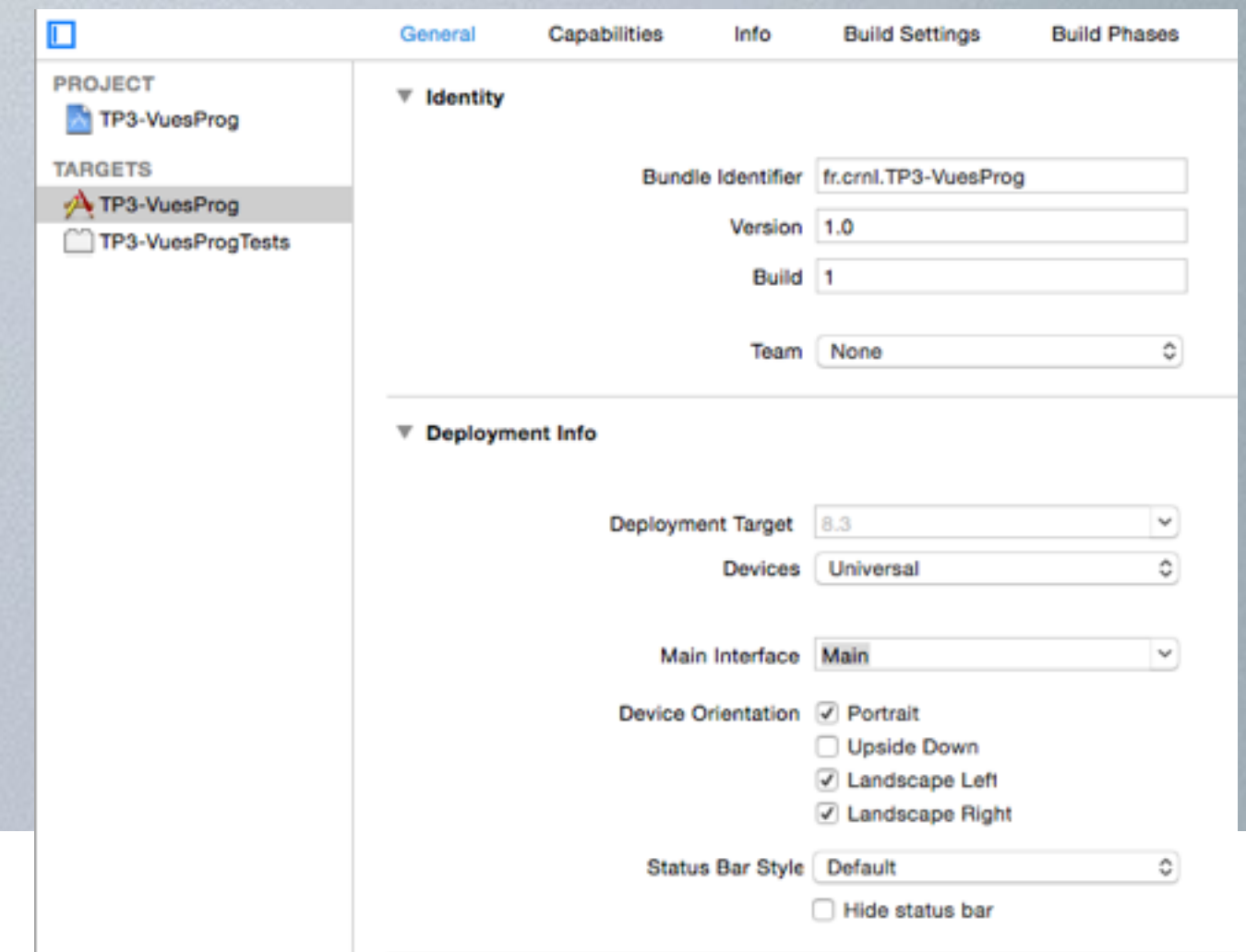
```
// AppDelegate.swift
// TP3-VuesProg
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?


    func application(application: UIApplication, didFinishLaunchingWithOptions launchOptions: [NSObject: AnyObject]?) -> Bool {
        // Override point for customization after application launch.
        window = UIWindow(frame: UIScreen.mainScreen().bounds)
        let VC = ViewController()
        window!.rootViewController = VC
        window!.makeKeyAndVisible()

        return true
    }
}
```



CONSTRUCTION DE VUE

En pratique: TP3 (3)

```
// ViewController.swift
// TP3-VuesProg
//
// Copyright (c) 2015 Pierre Emmanuel Aguera. All rights reserved.
//
import UIKit

class ViewController: UIViewController, UITextFieldDelegate {

    private var nomLabel:UILabel?
    private var prenomLabel:UILabel?
    private var prenomTextField:UITextField?
    private var nomTextField:UITextField?
    private var nomCompletLabel:UILabel?
    private var dateNaissancePicker:UIDatePicker?
    private var dateLabel:UILabel?
    private var recupBouton:UIButton?

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        // Couleur de fond de la vue principale
        view.backgroundColor = UIColor.whiteColor()

        nomLabel = UILabel()
        nomLabel!.text = "Nom"
        view.addSubview(nomLabel!)

        nomTextField = UITextField()
        nomTextField!.text = ""
        nomTextField!.borderStyle = .RoundedRect
        nomTextField!.delegate = self
        view.addSubview(nomTextField!)
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
}
```



CONSTRUCTION DE VUE

En pratique: TP3 (4)

```

override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    // Couleur de fond de la vue principale
    view.backgroundColor = UIColor.whiteColor()

    nomLabel = UILabel()
    nomLabel!.text = "Nom"
    view.addSubview(nomLabel!)

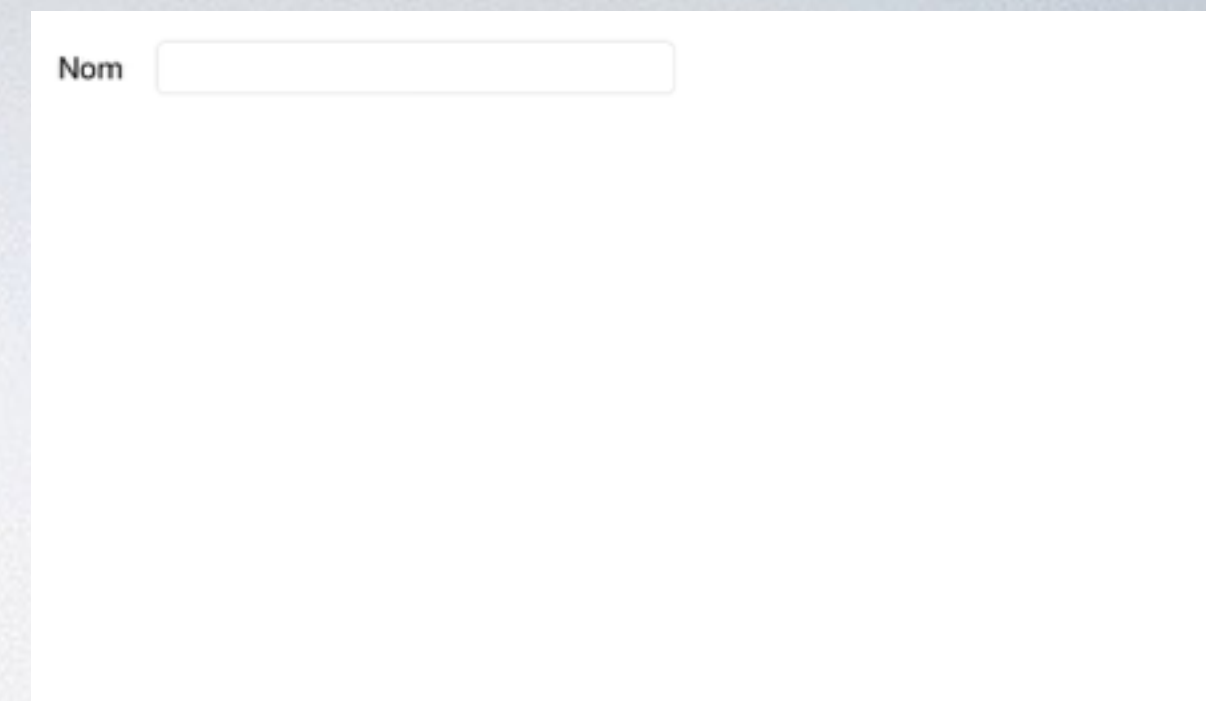
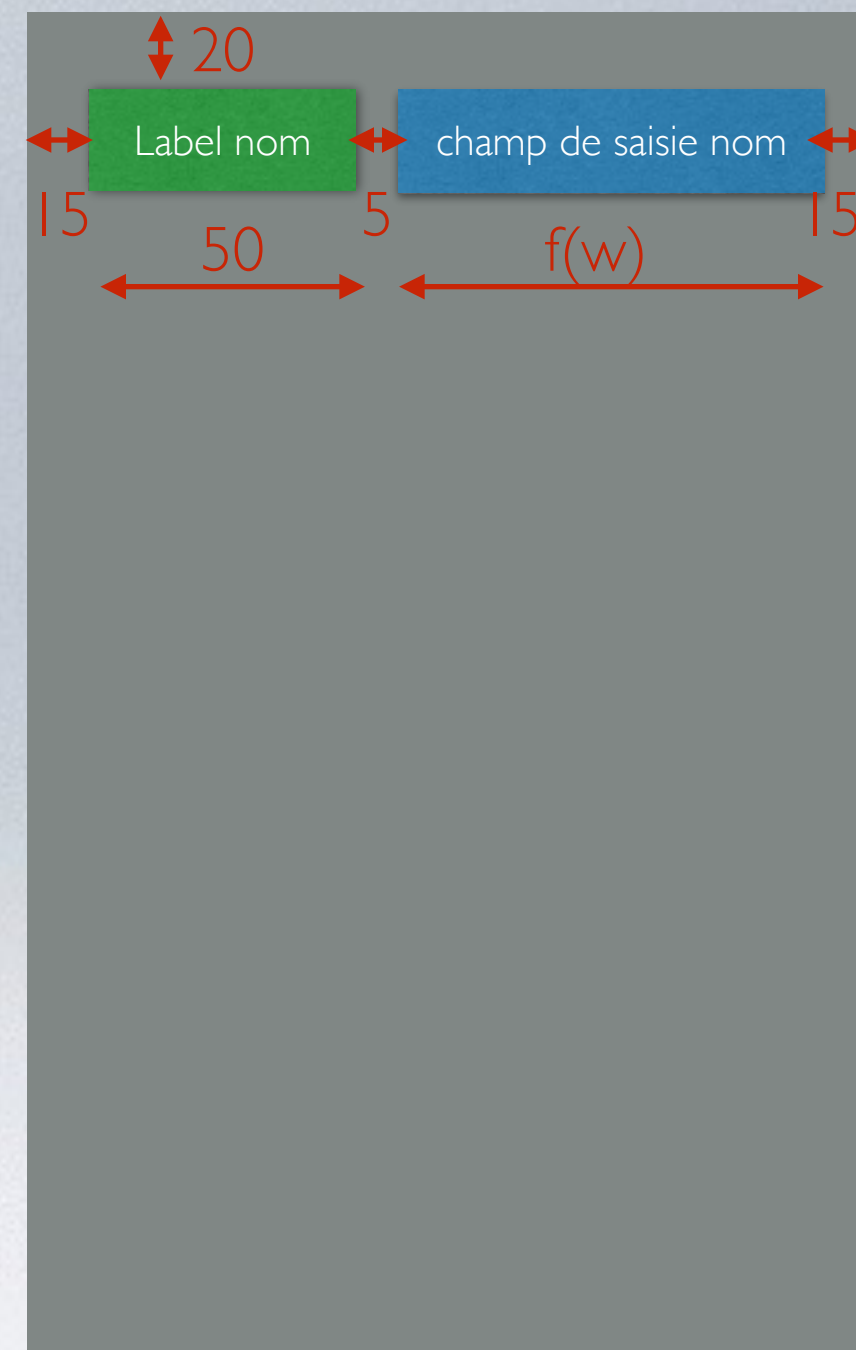
    nomTextField = UITextField()
    nomTextField!.text = ""
    nomTextField!.borderStyle = .RoundedRect
    nomTextField!.delegate = self
    view.addSubview(nomTextField!)

    // Tracé du contenu de la vue
    self.afficheContenu(UIScreen.mainScreen().bounds.size)
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}

// Méthode d'affichage du contenu de la vue
func afficheContenu(size: CGSize) {
    // Création des cadres du label et du champ de saisie du nom
    nomLabel!.frame = CGRectMake(15, 20, 50, 30);
    nomTextField!.frame = CGRectMake(15+50+5, 20, size.width-70-15, 30);
}

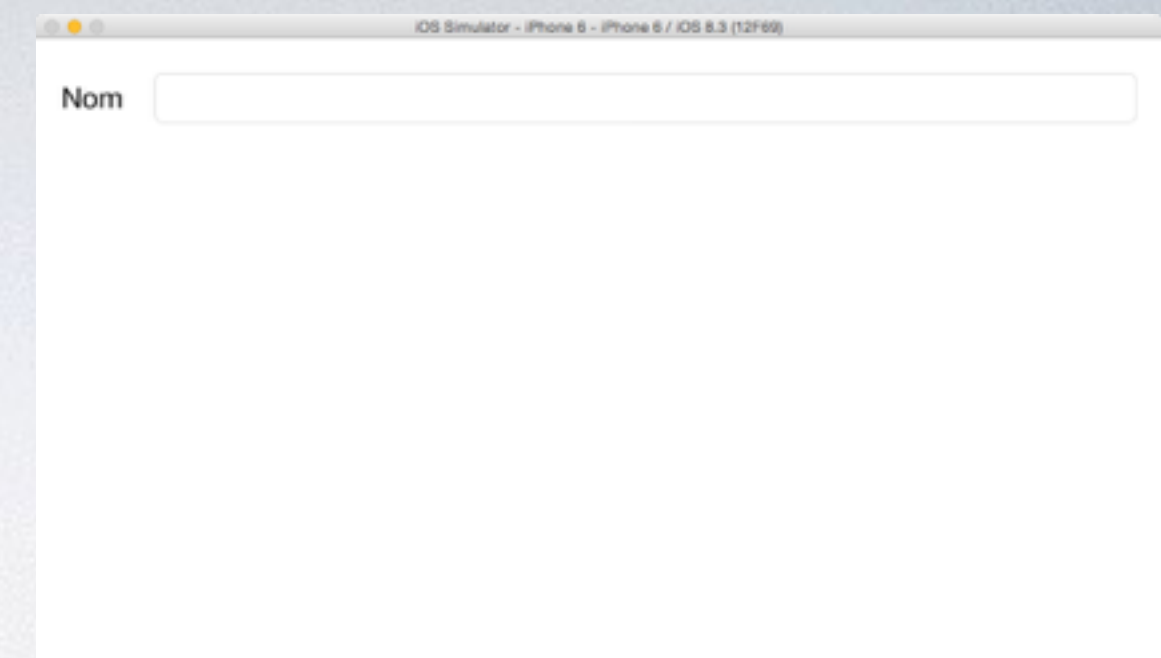
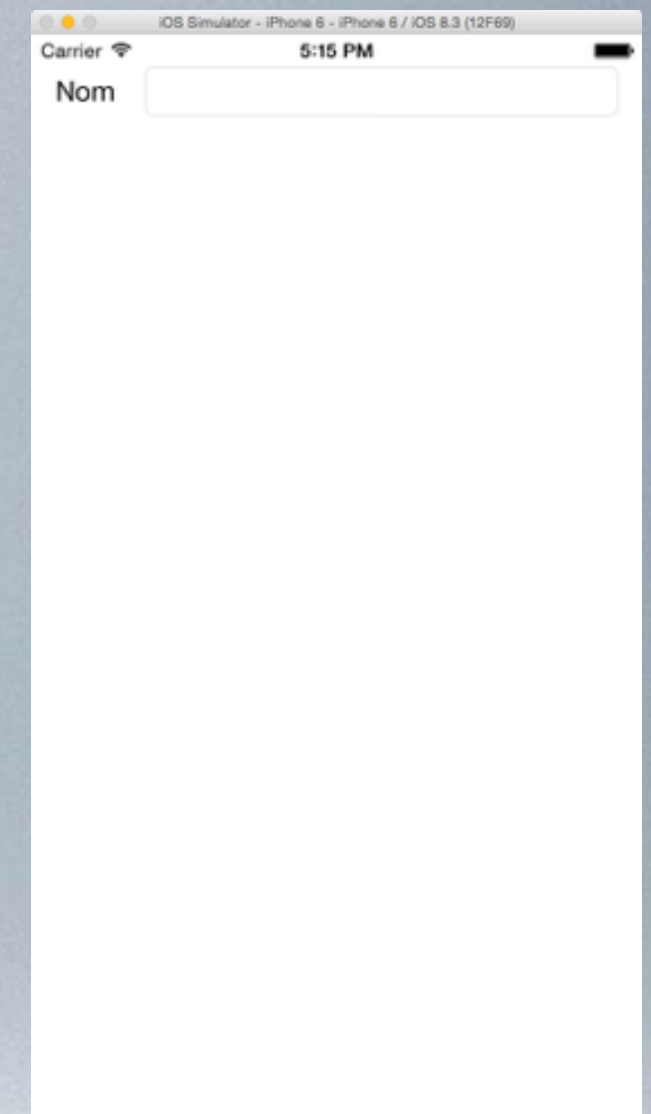
```



CONSTRUCTION DE VUE

En pratique: TP3 (5)

```
...  
  
// Méthode d'affichage du contenu de la vue  
func afficheContenu(size: CGSize) {  
    // Création des cadres du label et du champ de saisie du nom  
    nomLabel!.frame = CGRectMake(15, 20, 50, 30);  
    nomTextField!.frame = CGRectMake(15+50+5, 20, size.width-70-15, 30);  
}  
  
// MARK: Orientations  
override func shouldAutorotate() -> Bool {  
    return true  
}  
  
override func supportedInterfaceOrientations() -> Int {  
    return Int(UIInterfaceOrientationMask.All.rawValue)  
}  
  
override func viewWillTransitionToSize(size: CGSize, withTransitionCoordinator coordinator: UIViewControllerTransitionCoordinator) {  
    self.afficheContenu(size)  
}
```



CONSTRUCTION DE VUE

En pratique: TP3 (6)

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    // Couleur de fond de la vue principale
    view.backgroundColor = UIColor.whiteColor()

    nomLabel = UILabel()
    nomLabel!.text = "Nom"
    view.addSubview(nomLabel!)

    nomTextField = UITextField()
    nomTextField!.text = ""
    nomTextField!.borderStyle = .RoundedRect
    nomTextField!.delegate = self
    view.addSubview(nomTextField!)

    prenomLabel = UILabel()
    prenomLabel!.text = "Prénom"
    view.addSubview(prenomLabel!)

    prenomTextField = UITextField()
    prenomTextField!.text = ""
    prenomTextField!.borderStyle = .RoundedRect
    prenomTextField!.delegate = self
    view.addSubview(prenomTextField!)

    nomCompletLabel = UILabel()
    nomCompletLabel!.text = ""
    view.addSubview(nomCompletLabel!)

    // Tracé du contenu de la vue
    self.afficheContenu(UIScreen.mainScreen().bounds.size)
}

override func didReceiveMemoryWarning() {
    super.didReceiveMemoryWarning()
    // Dispose of any resources that can be recreated.
}

// Méthode d'affichage du contenu de la vue
func afficheContenu(size: CGSize) {
    // Création des cadres du label et du champ de saisie du nom
    nomLabel!.frame = CGRectMake(15, 20, 50, 30);
    nomTextField!.frame = CGRectMake(15+50+5, 20, size.width-70-15, 30);
    // Création des cadres du label et du champ de saisie du prénom
    prenomLabel!.frame = CGRectMake(15, 20+30+5, 70, 30);
    prenomTextField!.frame = CGRectMake(15+70+5, 20+30+5, size.width-90-15, 30);
    // Création des cadres du label du nom complet
    nomCompletLabel!.frame = CGRectMake(15, 55+30+50, size.width-2*15, 30);
}
```

```
class ViewController: UIViewController, UITextFieldDelegate {
```

```
    private var nomLabel:UILabel?
    private var prenomLabel:UILabel?
    private var prenomTextField:UITextField?
    private var nomTextField:UITextField?
    private var nomCompletLabel:UILabel?
    private var dateNaissancePicker:UIDatePicker?
    private var dateLabel:UILabel?
    private var recupBouton:UIButton?
```

```
    private var unePersonne:Personne?
```

```
    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view, typically from a nib.
        unePersonne = Personne()
    }
```

```
    // Couleur de fond de la vue principale
    view.backgroundColor = UIColor.whiteColor()
}
```

```
...
```

```
...
// MARK: UITextFieldDelegate
func textFieldDidEndEditing(textField: UITextField) {
    if (textField === nomTextField!) {
        unePersonne!.nom = textField.text
    } else if (textField === prenomTextField!) {
        unePersonne!.prenom = textField.text
    } else {
        println("ERREUR: textfield inconnu")
    }

    nomCompletLabel!.text = unePersonne!.nomCompletMeth()
}
...
```



CONSTRUCTION DE VUE

En pratique: TP3 (7)

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    unePersonne = Personne()

    ...

    recupBouton = UIButton.buttonWithType(UIButtonType.System) as? UIButton
    recupBouton!.setTitle("Récupérer", forState: .Normal)
    view.addSubview(recupBouton!)
    recupBouton!.addTarget(self, action: "recupTextes:", forControlEvents: .TouchDown)

    // Tracé du contenu de la vue
    self.afficheContenu(UIScreen.mainScreen().bounds.size)
}

...

// Méthode d'affichage du contenu de la vue
func afficheContenu(size: CGSize) {
    // Création des cadres du label et du champ de saisie du nom
    nomLabel!.frame = CGRectMake(15, 20, 50, 30);
    nomTextField!.frame = CGRectMake(15+50+5, 20, size.width-70-15, 30);
    // Création des cadres du label et du champ de saisie du prénom
    prenomLabel!.frame = CGRectMake(15, 20+30+5, 70, 30);
    prenomTextField!.frame = CGRectMake(15+70+5, 20+30+5, size.width-90-15, 30);
    // Création des cadres du label du nom complet
    nomCompletLabel!.frame = CGRectMake(15, 55+30+5, size.width-2*15, 30);
    // Création du cadre du bouton Récupérer
    recupBouton!.frame = CGRectMake((size.width-100)*0.5, 85+30+5, 100, 40)
}

// Méthode de récupération des textes saisis par l'appui sur le bouton Récupérer
func recupTextes(sender: AnyObject) {
    unePersonne!.nom = nomTextField!.text
    unePersonne!.prenom = prenomTextField!.text
    nomCompletLabel!.text = unePersonne!.nomCompletMeth()
}
```



CONSTRUCTION DE VUE

```

super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
unePersonne = Personne()

// Couleur de fond de la vue principale
view.backgroundColor = UIColor.whiteColor()

...
dateNaissancePicker = UIDatePicker()
dateNaissancePicker!.datePickerMode = .Date
view.addSubview(dateNaissancePicker!)
dateNaissancePicker!.addTarget(self, action: "changeDate:", forControlEvents: .ValueChanged)

dateLabel = UILabel()
dateLabel!.text = ""
view.addSubview(dateLabel!)

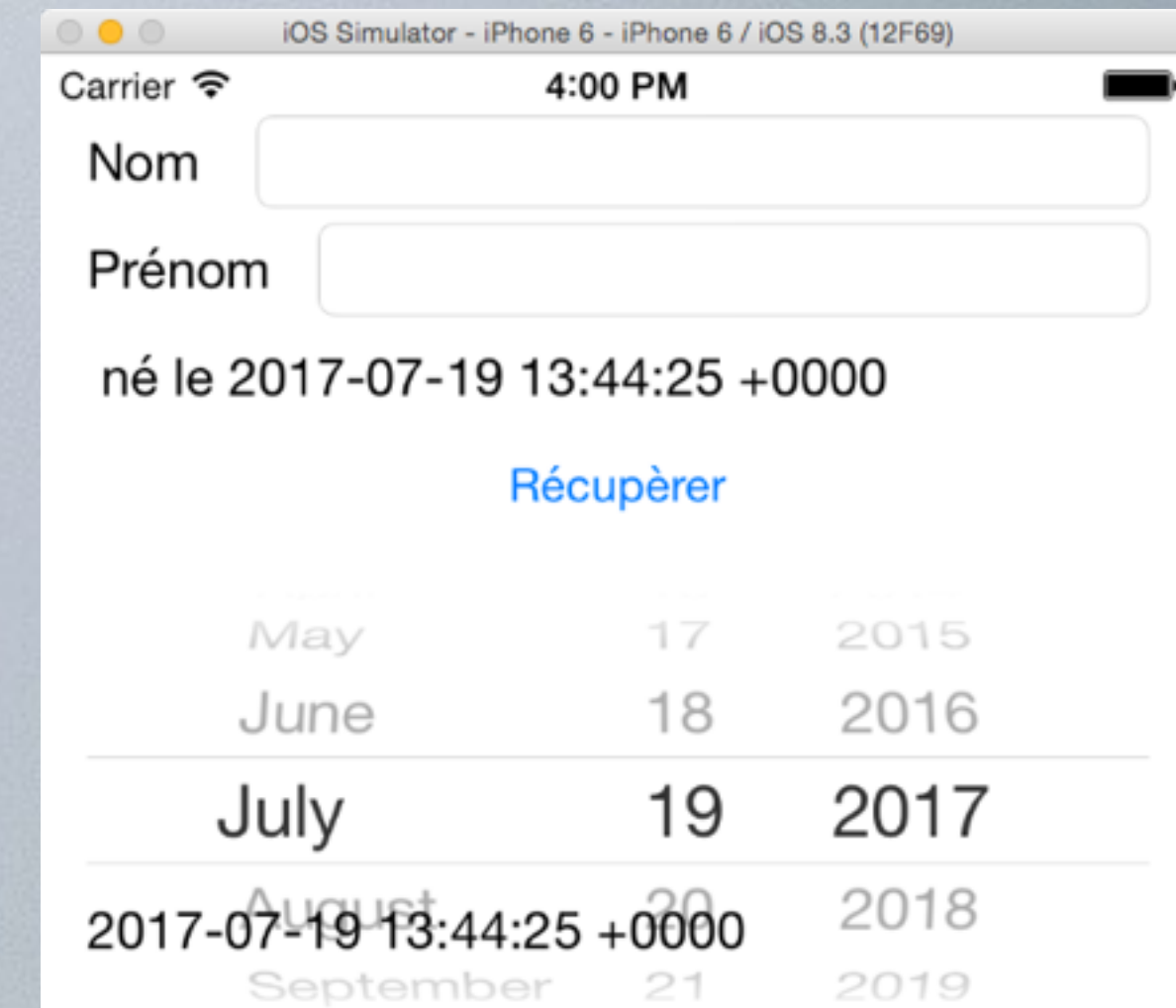
// Tracé du contenu de la vue
self.afficheContenu(UIScreen.mainScreen().bounds.size)
}

...
// Méthode d'affichage du contenu de la vue
func afficheContenu(size:CGSize) {
...
// Création du cadre du DatePicker
dateNaissancePicker!.frame = CGRectMake(15, 160+5, size.width-30, 100)
// Création du cadre du label de date
dateLabel!.frame = CGRectMake(15, 265+5, size.width-30, 30)
}

// Méthode de récupération des textes saisis par l'appui sur le bouton Récupérer
func recupTextes(sender:AnyObject) {
    unePersonne!.nom = nomTextField!.text
    unePersonne!.prenom = prenomTextField!.text
    nomCompleetLabel!.text = "\(unePersonne!.nomCompleetMeth()) né le \(dateNaissancePicker!.date)"
}

// Méthode de récupération de la date du UIDatePicker (appelée lors du changement de valeur)
func changeDate(sender:AnyObject) {
    dateLabel!.text = "\(dateNaissancePicker!.date)"
    // Mise à jour label nom complet avec date
    nomCompleetLabel!.text = "\(unePersonne!.nomCompleetMeth()) né le \(dateNaissancePicker!.date)"
}

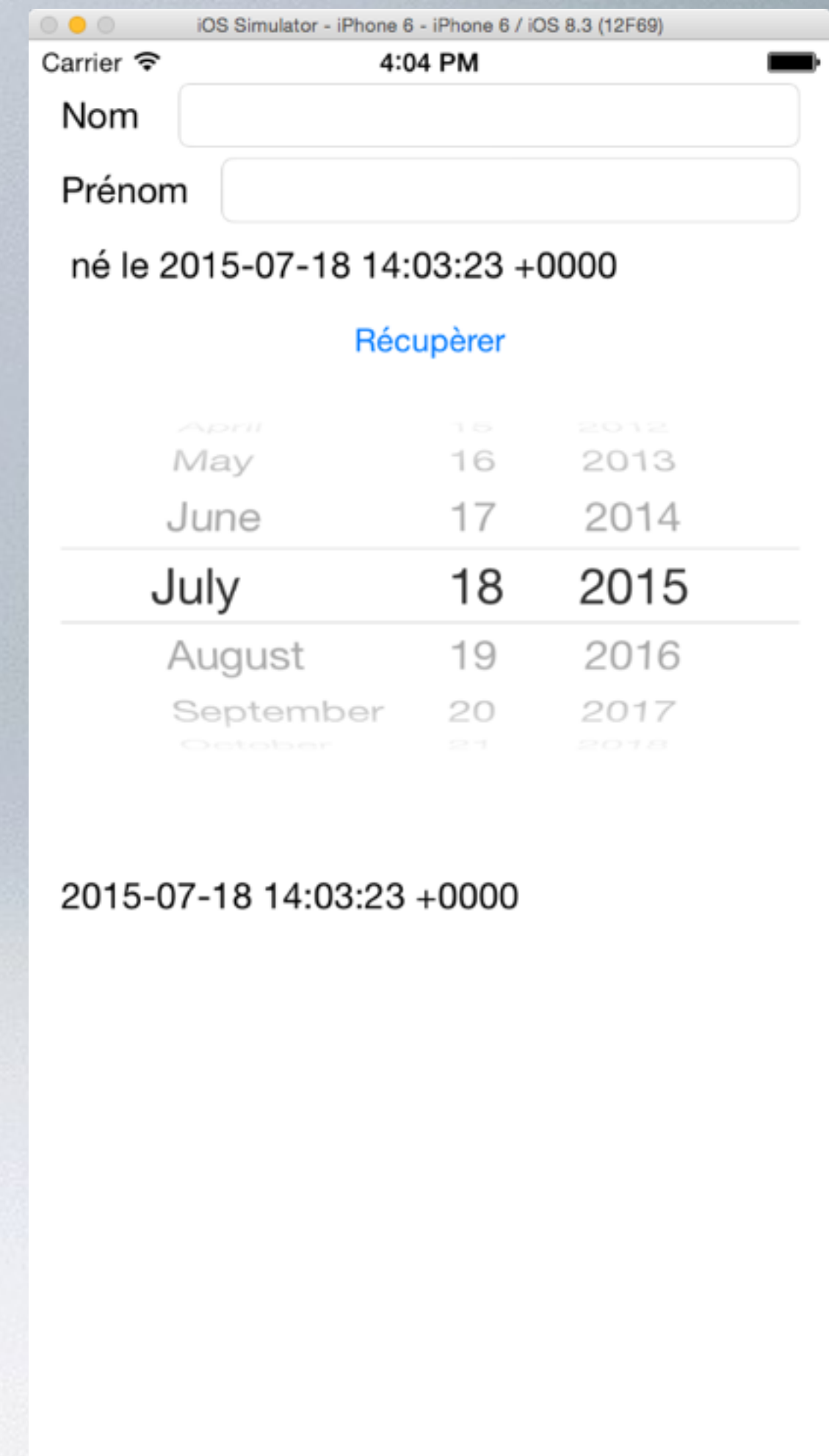
```



CONSTRUCTION DE VUE

En pratique: TP3 (9)

```
// Méthode d'affichage du contenu de la vue
func afficheContenu(size:CGSize) {
    // Création des cadres du label et du champ de saisie du nom
    nomLabel!.frame = CGRectMake(15, 20, 50, 30);
    nomTextField!.frame = CGRectMake(15+50+5, 20, size.width-70-15, 30);
    // Création des cadres du label et du champ de saisie du prénom
    prenomLabel!.frame = CGRectMake(15, 20+30+5, 70, 30);
    prenomTextField!.frame = CGRectMake(15+70+5, 20+30+5, size.width-90-15, 30);
    // Création des cadres du label du nom complet
    nomCompletLabel!.frame = CGRectMake(15, 55+30+5, size.width-2*15, 30);
    // Création du cadre du bouton Récupérer
    recupBouton!.frame = CGRectMake((size.width-100)*0.5, 85+30+5, 100, 40)
    // Création du cadre du DatePicker
    dateNaissancePicker!.frame = CGRectMake(15, 160+5, size.width-30, 215)
    // Création du cadre du label de date
    dateLabel!.frame = CGRectMake(15, 380+5, size.width-30, 30)
}
```



Carrier 4:04 PM

Nom

Prénom

né le 2015-07-18 14:03:23 +0000

[Récupérer](#)

April	15	2012
May	16	2013
June	17	2014
July	18	2015
August	19	2016
September	20	2017
October	21	2018

2015-07-18 14:03:23 +0000

RESSOURCES

Documentation Apple:

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

<https://developer.apple.com/library/ios/navigation/>

<https://developer.apple.com/swift/resources/>

MOOC France Université Numérique:

Programmation sur iPhone et iPad (parties I et II)

iTunes U:

Programmation sur plateforme mobile: application à iOS (UPMC)



MERCI

