



Introduction au développement iOS avec Swift

Cours 1 ~ Dev stack, UI



Aujourd'hui

- Environnement de développement sur iOS
 - *Swift, Xcode & Tools*
- User Interface



L'environnement de développement iOS

#



Objective-C

- Crée par Brad Cox et Tom Love (80's)
- Surcouche au langage C
- Utilisé et propagé par NeXT puis Apple

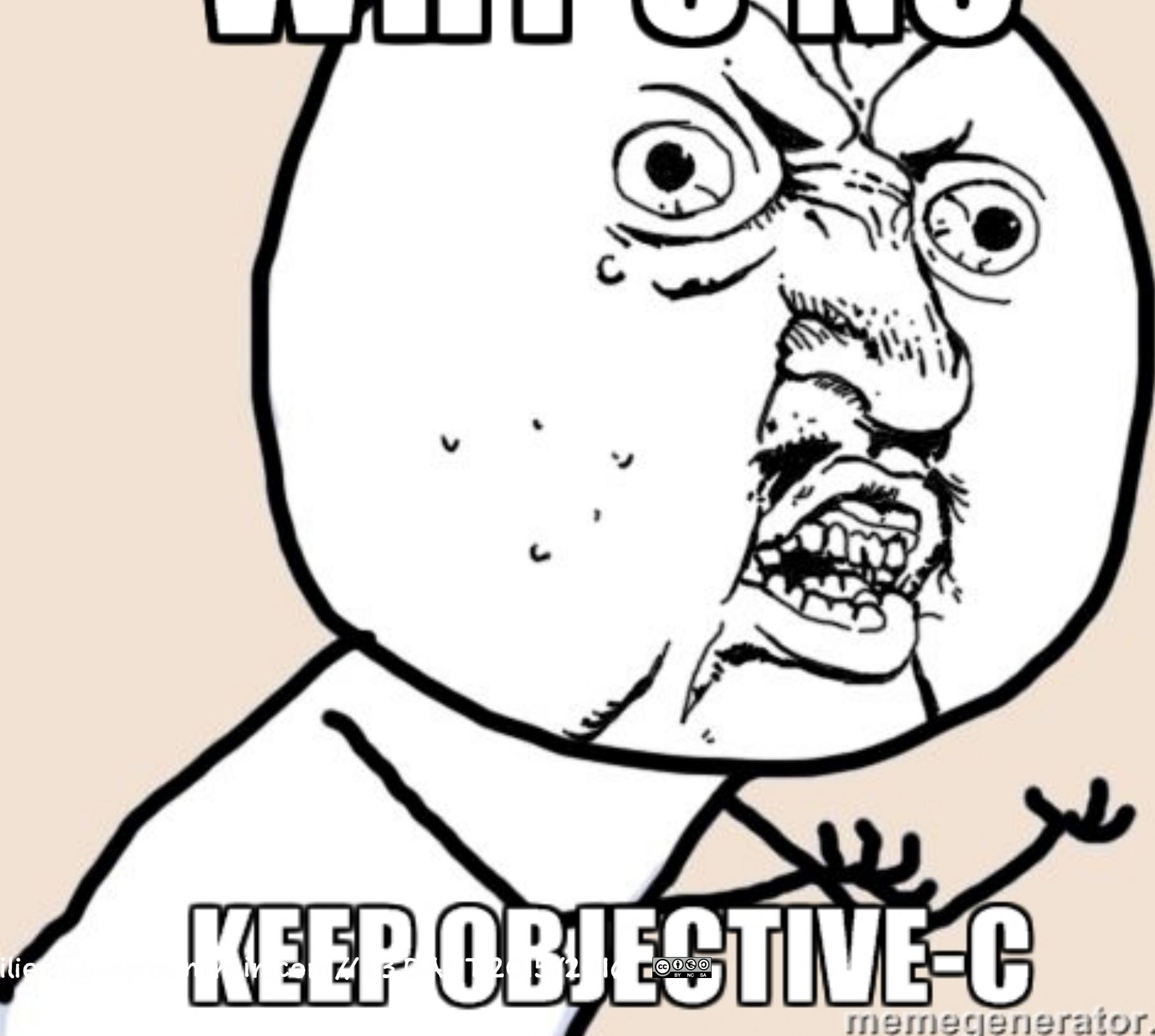




Swift

- Présenté en juin 2014 par Apple
- Crée par Chris Lattner
- Open-source
- Inspiré d'un grand nombre de langages de programmation moderne

WHY UNO



KEEP OBJECTIVE-C

memegenerator.net

Swift

- Débarassé de l'héritage du C
- Moderne
- Robuste
- Concis
- Rapide

Swift

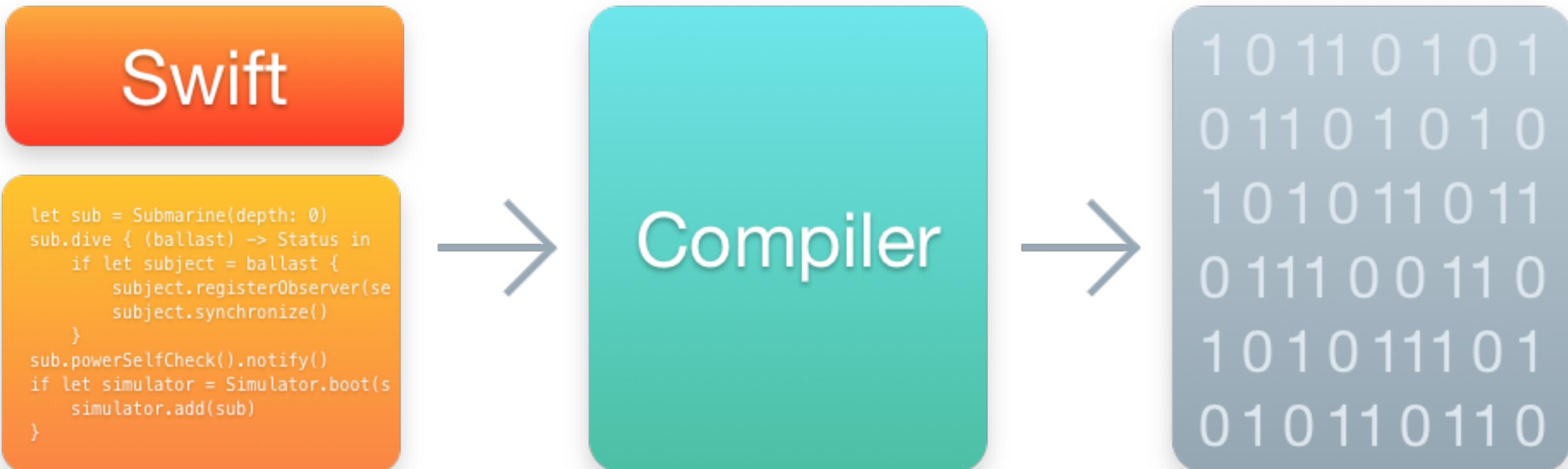
The Swift Programming Language greatly benefited from the experiences hard-won by many other languages in the field, drawing ideas from Objective-C, Rust, Haskell, Ruby, Python, C#, CLU, and far too many others to list.

— *Chris Lattner*

Swift open-source

- Plateforme [Swift.org](https://swift.org)
- 100% du développement et des choix d'orientation sont publics
- Tout le monde peut participer

Le Compilateur : LLDB



Frameworks

- Librairies de code
- Séparés de la librairie standard du langage (peuvent provenir d'autres sources/développeurs)
- Font l'essentiel du gros œuvre
- Permettent d'écrire du code de plus haut niveau

Frameworks

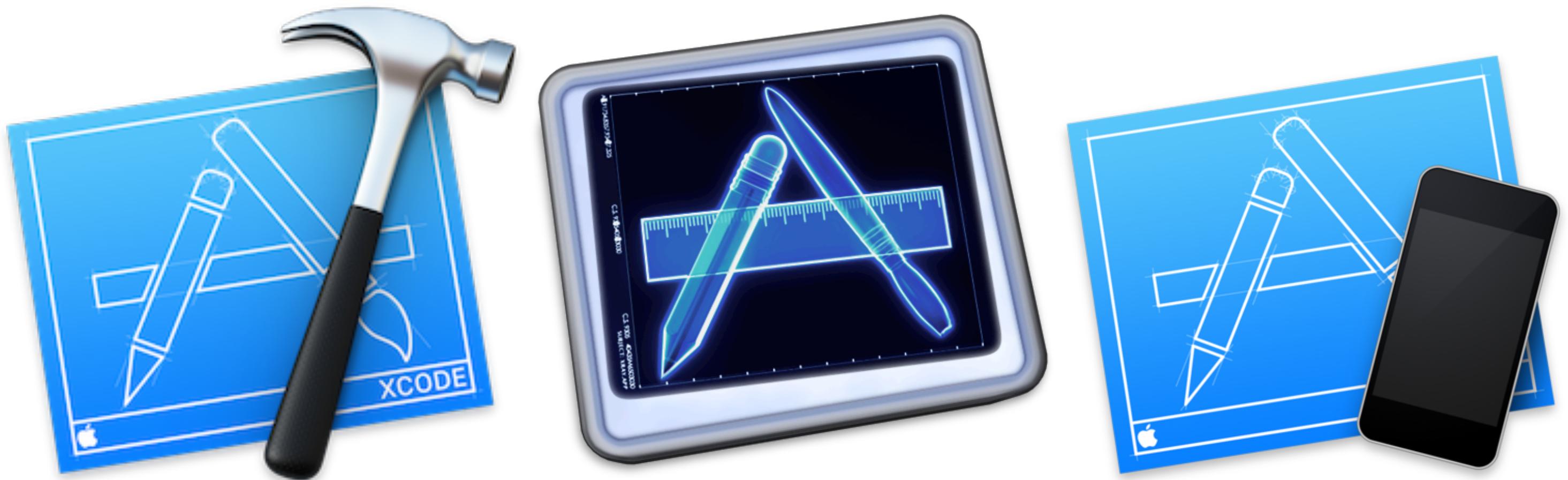
Exemple : Lire de l'audio

- Pas besoin d'écrire tout le code bas niveau pour s'interfacer avec les hauts-parleurs ou le casque.
- Utiliser une classe d'Audio Player d'un des frameworks fournis.

SafariServices ExternalAccessory
CoreImage Video Toolbox AVFoundation
AddressBook Media Accessibility Foundation
AddressBookUI LocalAuthentication Accounts
GameController SpriteKit ImageIO UIKit AVKit CoreVideo
CoreAudio Media Toolbox Twitter GameKit
CFNetwork NotificationCenter SocialCore Graphics
EventKit PushKit NetworkExtension Photos
PhotosUI MobileCore Services GSS
CoreMedia iAd Multipeer Connectivity HomeKit Accelerate
AudioUnit PassKit Metal OpenGLES EventKit UI
CoreMIDI Security NewsstandKit CoreFoundation
AdSupport IOKit OpenAL MapKit CoreData
CoreMotion CloudKit JavaScript CoreSceneKit CoreLocation
CoreText QuartzCore Media Player CoreBluetooth
HealthKit System Configuration Audio Toolbox
AssetsLibrary QuickLook CoreAudioKit CoreTelephony

SafariServices ExternalAccessory
CoreImage VideoToolbox AVFoundation
AddressBook MediaAccessibility Foundation
AddressBookUI LocalAuthentication Accounts
GameController SpriteKit ImageIO UIKit AVKit CoreVideo
CoreAudio MediaToolbox Twitter GameKit
CFNetwork NotificationCenter SocialCoreGraphics
EventKit PushKit NetworkExtension Photos
PhotosUI MobileCoreServices GSS
CoreMedia iAd MultipeerConnectivity HomeKit Accelerate
AudioUnit PassKit Metal OpenGLES EventKitUI
CoreMIDI Security NewsstandKit CoreFoundation
AdSupport IOKit OpenAL MapKit CoreData
CoreMotion CloudKit JavaScript CoreSceneKit CoreLocation
CoreText QuartzCoreMediaPlayer CoreBluetooth
HealthKit SystemConfiguration AudioToolbox
AssetsLibrary QuickLook CoreAudioKit CoreTelephony

Les outils de dev



Xcode

- Integrated Development Environment (IDE)
- Regroupe le code de l'application
- Permet de compiler en un clic
- Inclut Interface Builder
 - Crédit simplifier d'interfaces utilisateur
 - Connection simplifiée des interfaces avec le code

Stopwatch > iPhone 6 Stopwatch: Ready | Today at 12:59 PM

```

Stopwatch
  2 targets, iOS SDK 8.3
    + Stopwatch
      - AppDelegate.swift
      - ViewController.swift
      - Stopwatch.swift
      - Main.storyboard
      - Images.xcassets
      - LaunchScreen.xib
    + Supporting Files
      - Info.plist
  + StopwatchTests
    - StopwatchTests.swift
    - Supporting Files
  + Products

```

```

Copyright (C) 2015 Apple Inc. All Rights Reserved.

/*
import UIKit
class ViewController: UIViewController {
    var stopwatch: Stopwatch?
    @IBOutlet weak var elapsedTimeLabel: UILabel!
    override func viewDidLoad() {
        super.viewDidLoad()
        stopwatch = Stopwatch()
    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
    @IBAction func startButtonTapped(sender: UIButton) {
        NSTimer.scheduledTimerWithTimeInterval(0.1,
            target: self,
            selector: "updateElapsedTimeLabel:",
            userInfo: nil,
            repeats: true)
        stopwatch!.start()
    }
    @IBAction func stopButtonTapped(sender: UIButton) {
        stopwatch!.stop()
    }
    func updateElapsedTimeLabel(timer: NSTimer) {
        if stopwatch!.running {
            elapsedTimeLabel.text = stopwatch!.elapsedTimeAsString()
        } else {
            timer.invalidate()
        }
    }
}

```

```

Copyright (C) 2015 Apple Inc. All Rights Reserved.

/*
import Foundation
class Stopwatch {
    var startTime: NSDate?
    var running: Bool {
        return startTime != nil
    }
    var elapsedTime: NSTimeInterval {
        if let s = startTime {
            return -s.timeIntervalSinceNow
        } else {
            return 0
        }
    }
    func start() {
        startTime = NSDate()
    }
    func stop() {
        startTime = nil
    }
    func elapsedTimeAsString() -> String {
        var minutes = floor(elapsedTime / 60)
        var seconds = floor(elapsedTime - minutes * 60)
        var tenthsOfSecond = (elapsedTime - (minutes * 60 + seconds)) * 10
        return String(format:"%02d:%02d:%d", Int(minutes),
            Int(seconds), Int(tenthsOfSecond))
    }
}

```

Identity and Type

- Name: Stopwatch.swift
- Type: Default - Swift Source
- Location: Relative to Group
- Full Path: /Users/ybakos/Dropbox/Shared/apple/crafting-ios/apps/swift-edition/working/Level 2/Stopwatch/StopwatchEnd/Stopwatch/Stopwatch.swift

Target Membership

- Stopwatch
- StopwatchTests

Text Settings

- Text Encoding: Unicode (UTF-8)
- Line Endings: Default - OS X / Unix (LF)
- Indent Using: Spaces
- Widths: Tab 4 Indent 4
- Wrap lines

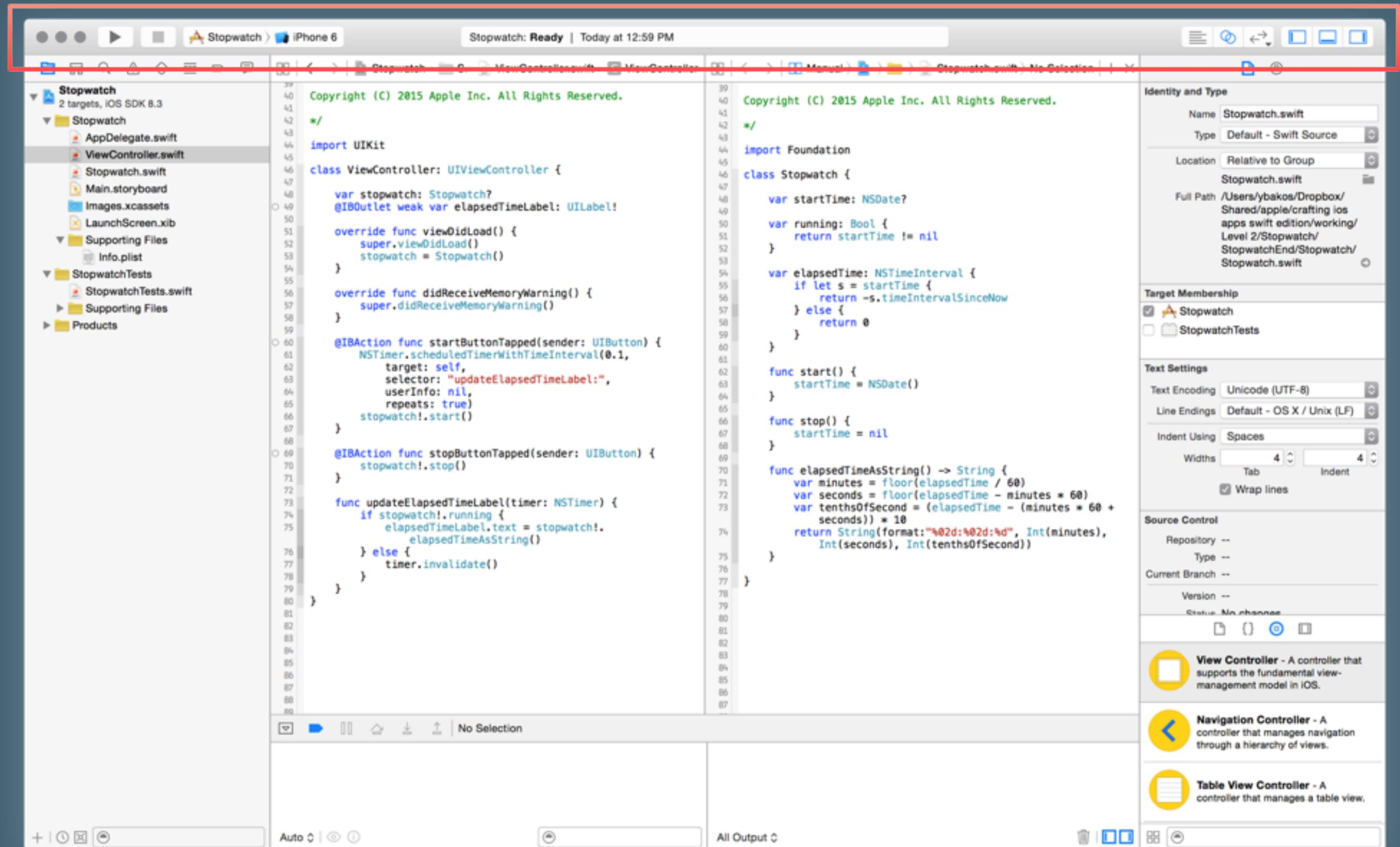
Source Control

- Repository --
- Type --
- Current Branch --
- Version --
- Status: No changes

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

Navigation Controller - A controller that manages navigation through a hierarchy of views.

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



```
Stopwatch > iPhone 6 | Stopwatch: Ready | Today at 12:59 PM
```

```
Copyright (C) 2015 Apple Inc. All Rights Reserved.
```

```
/*
```

```
import UIKit
```

```
class ViewController: UIViewController {
```

```
    var stopwatch: Stopwatch?
```

```
    @IBOutlet weak var elapsedTimeLabel: UILabel!
```

```
    override func viewDidLoad() {
```

```
        super.viewDidLoad()
```

```
        stopwatch = Stopwatch()
```

```
    }
```

```
    override func didReceiveMemoryWarning() {
```

```
        super.didReceiveMemoryWarning()
```

```
    }
```

```
    @IBAction func startButtonTapped(sender: UIButton) {
```

```
        NSTimer.scheduledTimerWithTimeInterval(0.1,
```

```
            target: self,
```

```
            selector: "updateElapsedTimeLabel:",
```

```
            userInfo: nil,
```

```
            repeats: true)
```

```
        stopwatch!.start()
```

```
    }
```

```
    @IBAction func stopButtonTapped(sender: UIButton) {
```

```
        stopwatch!.stop()
```

```
    }
```

```
    func updateElapsedTimeLabel(timer: NSTimer) {
```

```
        if stopwatch!.running {
```

```
            elapsedTimeLabel.text = stopwatch!.
```

```
                elapsedTimeAsString()
```

```
        } else {
```

```
            timer.invalidate()
```

```
        }
```

```
    }
```

```
}
```

```
Copyright (C) 2015 Apple Inc. All Rights Reserved.
```

```
/*
```

```
import Foundation
```

```
class Stopwatch {
```

```
    var startTime: NSDate?
```

```
    var running: Bool {
```

```
        return startTime != nil
```

```
    }
```

```
    var elapsedTime: NSTimeInterval {
```

```
        if let s = startTime {
```

```
            return -s.timeIntervalSinceNow
```

```
        } else {
```

```
            return 0
```

```
        }
```

```
    }
```

```
    func start() {
```

```
        startTime = NSDate()
```

```
    }
```

```
    func stop() {
```

```
        startTime = nil
```

```
    }
```

```
    func elapsedTimeAsString() -> String {
```

```
        var minutes = floor(elapsedTime / 60)
```

```
        var seconds = floor(elapsedTime - minutes * 60)
```

```
        var tenthsOfSecond = (elapsedTime - (minutes * 60 +
```

```
            seconds)) * 10
```

```
        return String(format:"%02d:%02d:%d", Int(minutes),
```

```
            Int(seconds), Int(tenthsOfSecond))
```

```
    }
```

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** On the left, it shows the project structure for "Stopwatch". A red box highlights the "Stopwatch" group, which contains files like AppDelegate.swift, ViewController.swift, and Main.storyboard.
- Editor:** The main editor area displays the content of `ViewController.swift`. The code defines a `ViewController` class that interacts with a `Stopwatch` object to manage elapsed time.
- Utility Navigator:** On the right, the "Identity and Type" section shows the file is named `Stopwatch.swift`, has a type of "Default - Swift Source", and is located relative to the group. It also lists target membership for "Stopwatch" and "StopwatchTests".
- Text Settings:** Includes options for Text Encoding (Unicode (UTF-8)), Line Endings (Default - OS X / Unix (LF)), Indent Using (Spaces), and Widths (Tab width 4, Indent 4).
- Source Control:** Shows repository, type, current branch, version, and status information.
- Document Outline:** At the bottom, it shows "View Controller", "Navigation Controller", and "Table View Controller" with their descriptions.

```
Copyright (C) 2015 Apple Inc. All Rights Reserved.  
*/  
import UIKit  
class ViewController: UIViewController {  
    var stopwatch: Stopwatch?  
    @IBOutlet weak var elapsedTimeLabel: UILabel!  
    override func viewDidLoad() {  
        super.viewDidLoad()  
        stopwatch = Stopwatch()  
    }  
    override func didReceiveMemoryWarning() {  
        super.didReceiveMemoryWarning()  
    }  
    @IBAction func startButtonTapped(sender: UIButton) {  
        NSTimer.scheduledTimerWithTimeInterval(0.1,  
            target: self,  
            selector: "updateElapsedTimeLabel:",  
            userInfo: nil,  
            repeats: true)  
        stopwatch!.start()  
    }  
    @IBAction func stopButtonTapped(sender: UIButton) {  
        stopwatch!.stop()  
    }  
    func updateElapsedTimeLabel(timer: NSTimer) {  
        if stopwatch!.running {  
            elapsedTimeLabel.text = stopwatch!.  
                elapsedTimeAsString()  
        } else {  
            timer.invalidate()  
        }  
    }  
}  
Copyright (C) 2015 Apple Inc. All Rights Reserved.  
*/  
import Foundation  
class Stopwatch {  
    var startTime: NSDate?  
    var running: Bool {  
        return startTime != nil  
    }  
    var elapsedTime: NSTimeInterval {  
        if let s = startTime {  
            return NSDate().timeIntervalSinceDate(s)  
        } else {  
            return 0  
        }  
    }  
    func start() {  
        startTime = NSDate()  
    }  
    func stop() {  
        startTime = nil  
    }  
    func elapsedTimeAsString() -> String {  
        var minutes = floor(elapsedTime / 60)  
        var seconds = floor(elapsedTime - minutes * 60)  
        var tenthsOfSecond = (elapsedTime - (minutes * 60 +  
            seconds)) * 10  
        return String(format:"%02d:%02d:%d", Int(minutes),  
            Int(seconds), Int(tenthsOfSecond))  
    }  
}
```

The screenshot shows the Xcode interface with the following details:

- Project Navigator:** Shows the project structure for "Stopwatch".
- Editor:** Displays the code for `ViewController.swift`. The code defines a `ViewController` class that initializes a `stopwatch` variable and handles button taps to start and stop the timer. It also defines an `updateElapsedTextLabel` function that updates a label with the elapsed time.
- Identity and Type Inspector (highlighted by a red box):** Shows the file's properties:
 - Name:** Stopwatch.swift
 - Type:** Default - Swift Source
 - Location:** Relative to Group
 - Full Path:** /Users/ybakos/Dropbox/Shared/apple/crafting ios apps swift edition/working/Level 2/Stopwatch/StopwatchEnd/Stopwatch/Stopwatch.swift
- Target Membership:** Shows the file is associated with the "Stopwatch" target.
- Text Settings:** Includes options for Text Encoding (Unicode (UTF-8)), Line Endings (Default - OS X / Unix (LF)), Indent Using (Spaces), and Widths (Tab width 4, Indent 4).
- Source Control:** Shows repository, type, and current branch information.
- Documentation:** Provides links to View Controller, Navigation Controller, and Table View Controller documentation.

```
Stopwatch
2 targets, iOS SDK 8.3
Stopwatch
AppDelegate.swift
ViewController.swift
Stopwatch.swift
Main.storyboard
Images.xcassets
LaunchScreen.xib
Supporting Files
Info.plist
StopwatchTests
StopwatchTests.swift
Supporting Files
Products
```

```
Copyright (C) 2015 Apple Inc. All Rights Reserved.
*/
import UIKit
class ViewController: UIViewController {
    var stopwatch: Stopwatch?
    @IBOutlet weak var elapsedTimeLabel: UILabel!
    override func viewDidLoad() {
        super.viewDidLoad()
        stopwatch = Stopwatch()
    }
    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
    }
    @IBAction func startButtonTapped(sender: UIButton) {
        NSTimer.scheduledTimerWithTimeInterval(0.1,
            target: self,
            selector: "updateElapsedTimeLabel:",
            userInfo: nil,
            repeats: true)
        stopwatch!.start()
    }
    @IBAction func stopButtonTapped(sender: UIButton) {
        stopwatch!.stop()
    }
    func updateElapsedTimeLabel(timer: NSTimer) {
        if stopwatch!.running {
            elapsedTimeLabel.text = stopwatch!.elapsedTimeAsString()
        } else {
            timer.invalidate()
        }
    }
}

Copyright (C) 2015 Apple Inc. All Rights Reserved.
*/
import Foundation
class Stopwatch {
    var startTime: NSDate?
    var running: Bool {
        return startTime != nil
    }
    var elapsedTime: NSTimeInterval {
        if let s = startTime {
            return -s.timeIntervalSinceNow
        } else {
            return 0
        }
    }
    func start() {
        startTime = NSDate()
    }
    func stop() {
        startTime = nil
    }
    func elapsedTimeAsString() -> String {
        var minutes = floor(elapsedTime / 60)
        var seconds = floor(elapsedTime - minutes * 60)
        var tenthsOfSecond = (elapsedTime - (minutes * 60 + seconds)) * 10
        return String(format:"%02d:%02d:%d", Int(minutes),
            Int(seconds), Int(tenthsOfSecond))
    }
}
```

Identity and Type
Name: Stopwatch.swift
Type: Default - Swift Source
Location: Relative to Group
Full Path: /Users/ybakos/Dropbox/Shared/apple/crafting-ios/apps/swift/edition/working/Level 2/Stopwatch/StopwatchEnd/Stopwatch/Stopwatch.swift

Target Membership
 Stopwatch
 StopwatchTests

Text Settings
Text Encoding: Unicode (UTF-8)
Line Endings: Default - OS X / Unix (LF)
Indent Using: Spaces
Widths: Tab 4 Indent 4
 Wrap lines

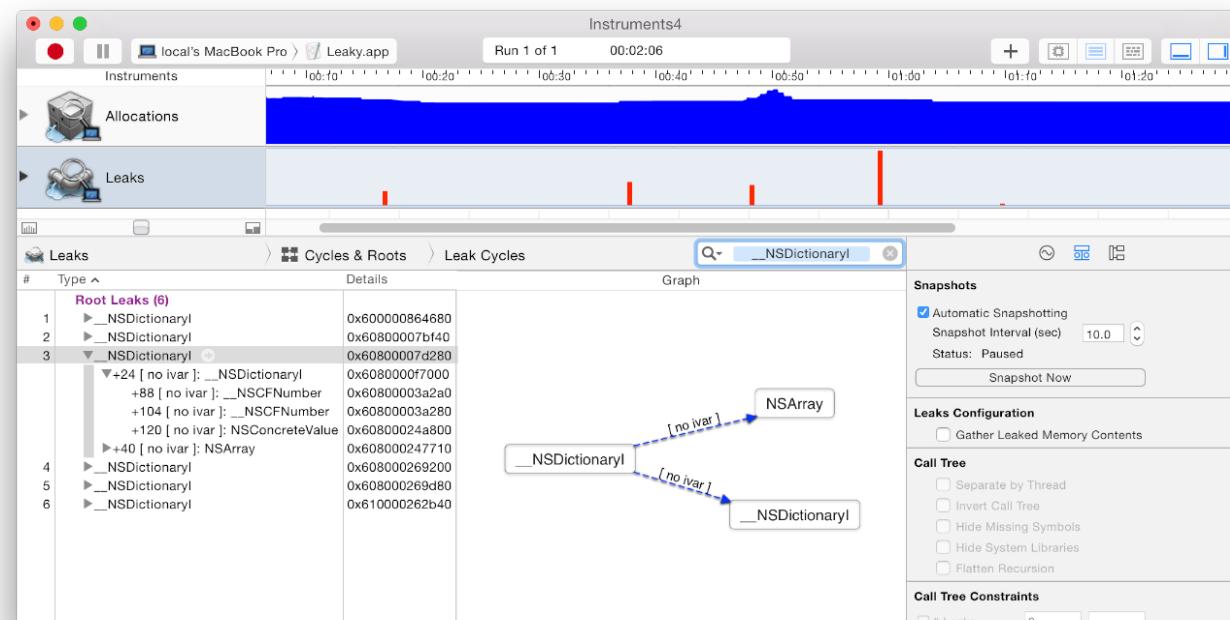
Source Control
Repository --
Type --
Current Branch --
Version --
Status: No changes

No Selection

All Output 0

Instruments

→ Monitoring sur ce qui se passe dans l'application pendant son exécution



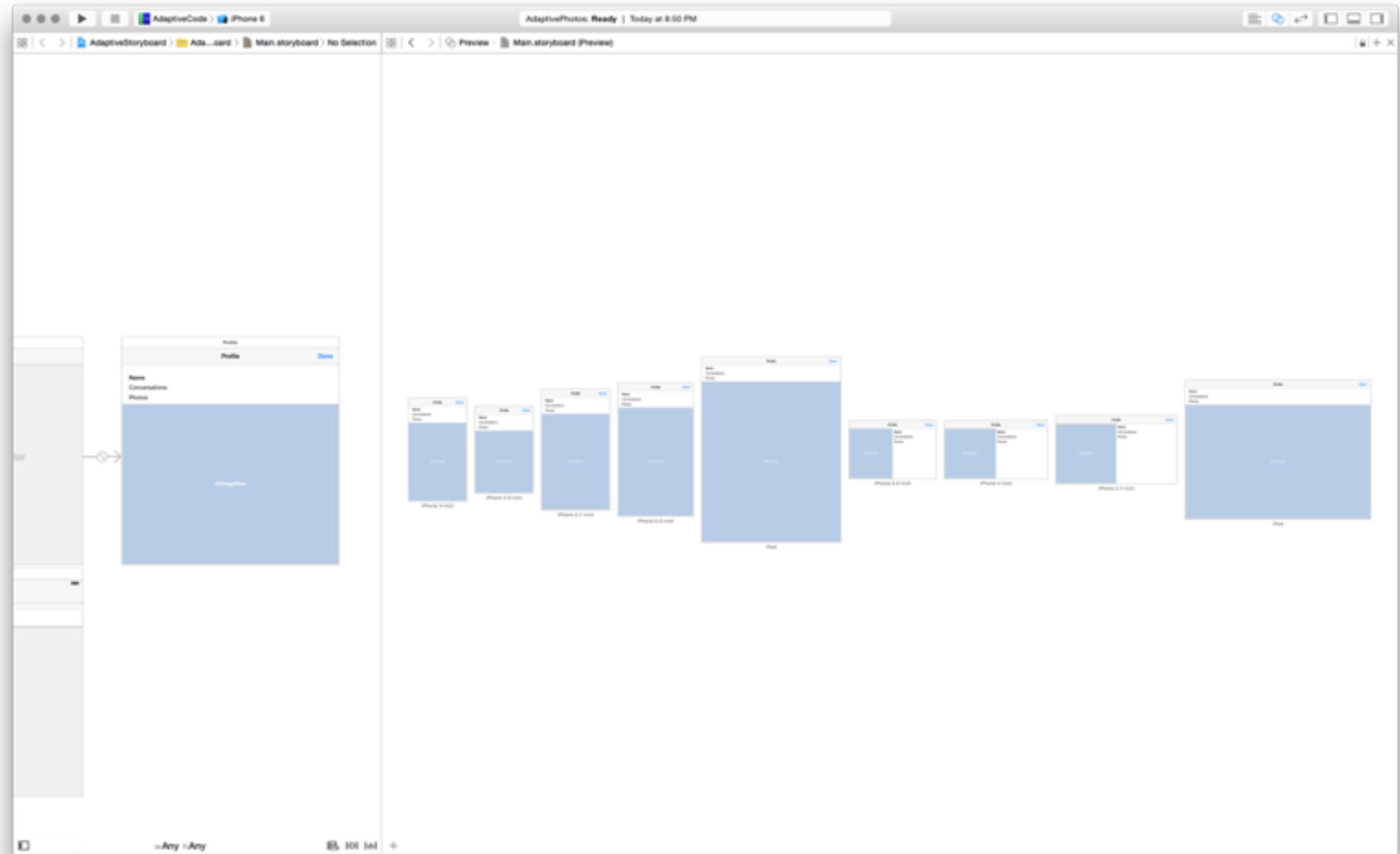
iOS Simulator

- Permet de lancer l'application iOS en cours de développement sur le Mac comme si elle s'exécutait sur un device iOS

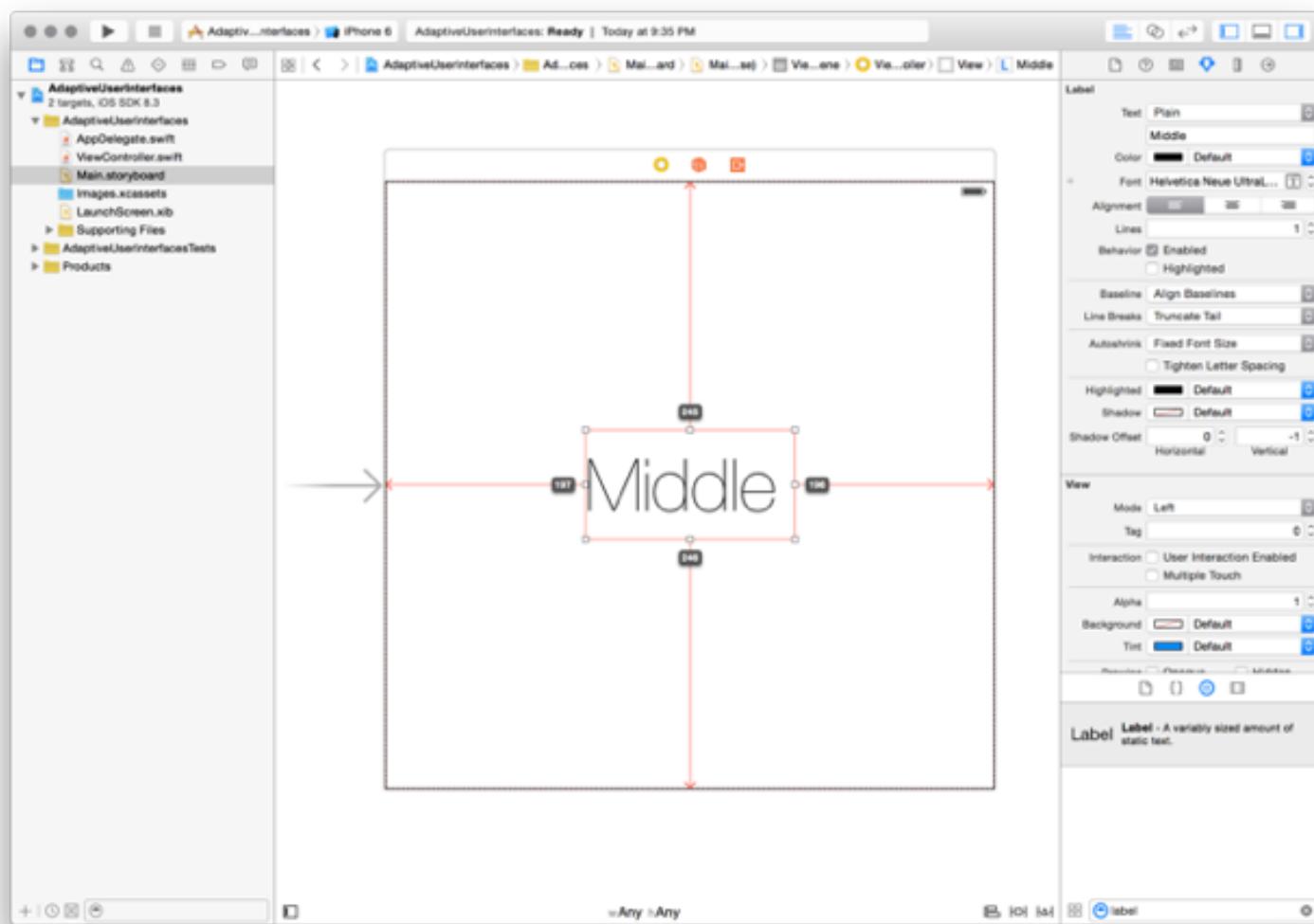


#

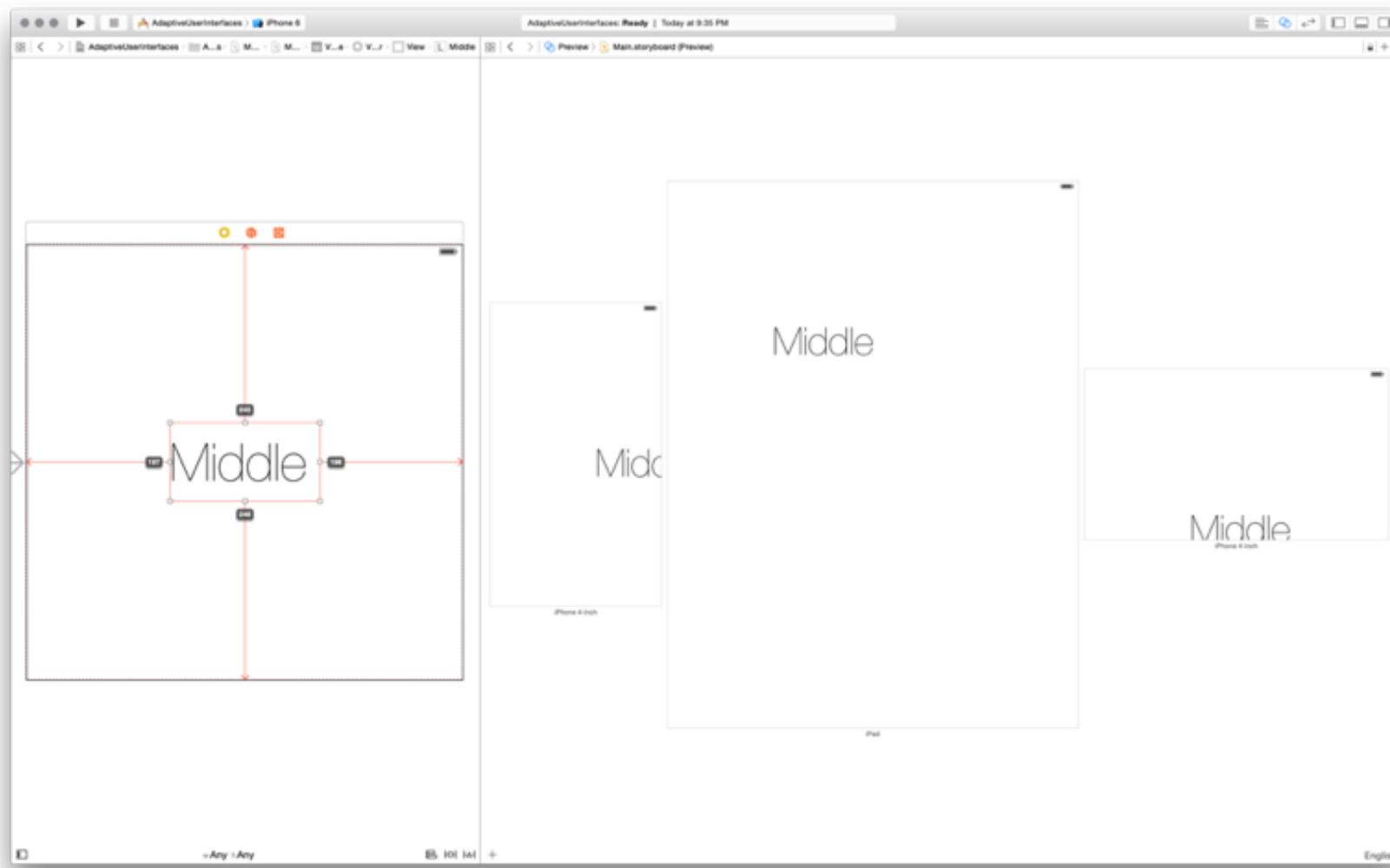
User Interfaces



Taille et position

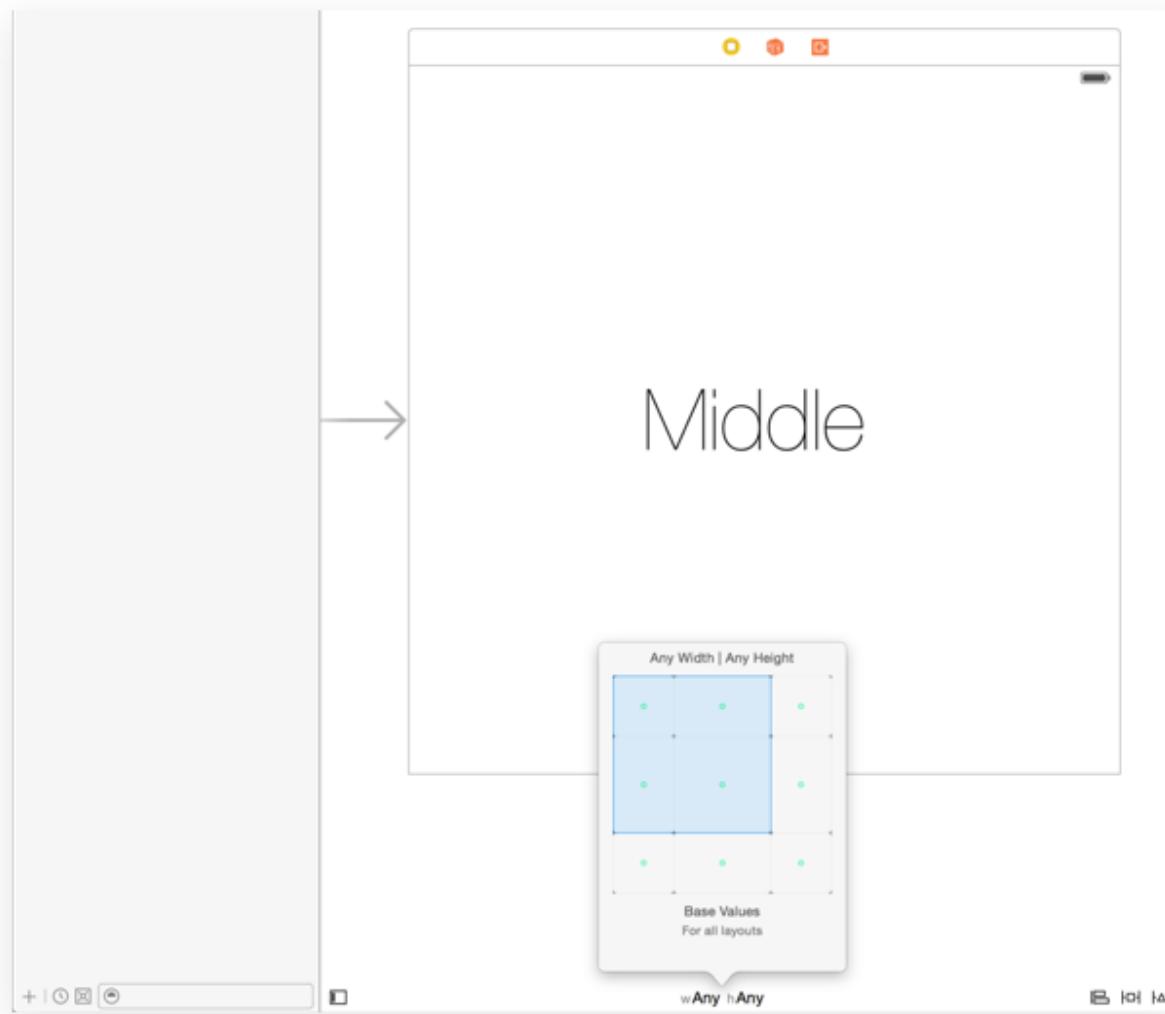


Taille et position

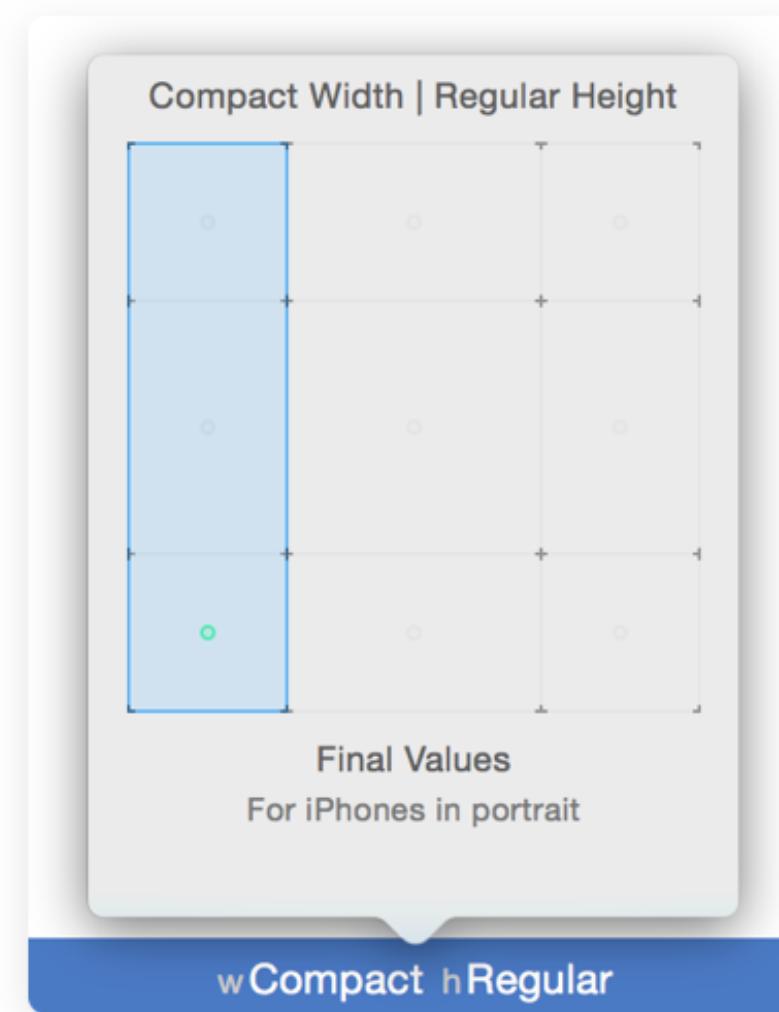


- Multiplicité des tailles d'écran
 - Différentes orientations
 - Mode fenêtré sur iPad (largeur variable)
 - Impossible d'estimer les tailles en points
-
- Comment faire ?

Abstraction (Size classes)



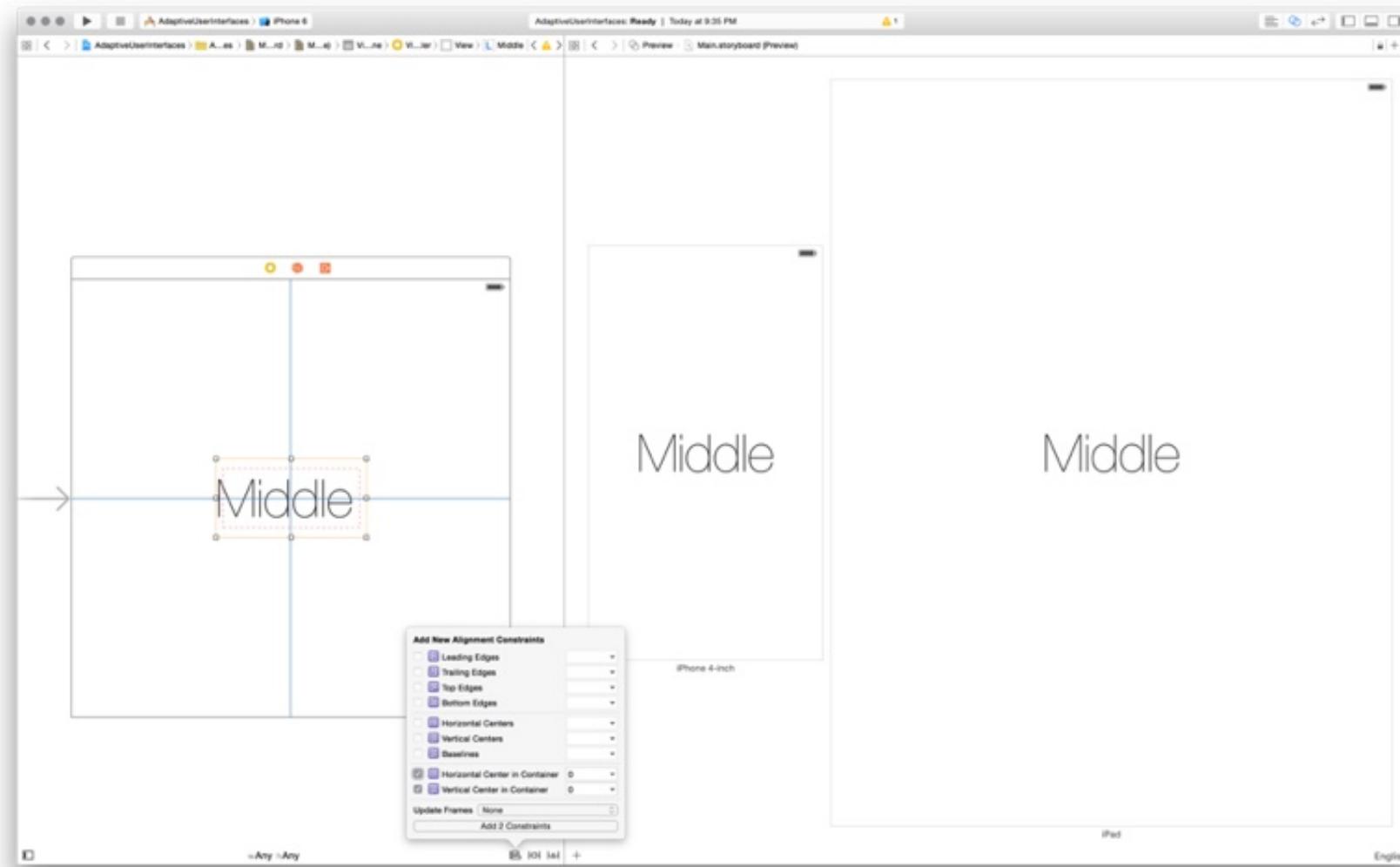
Abstraction (Size classes)



Relativité (Constraints)

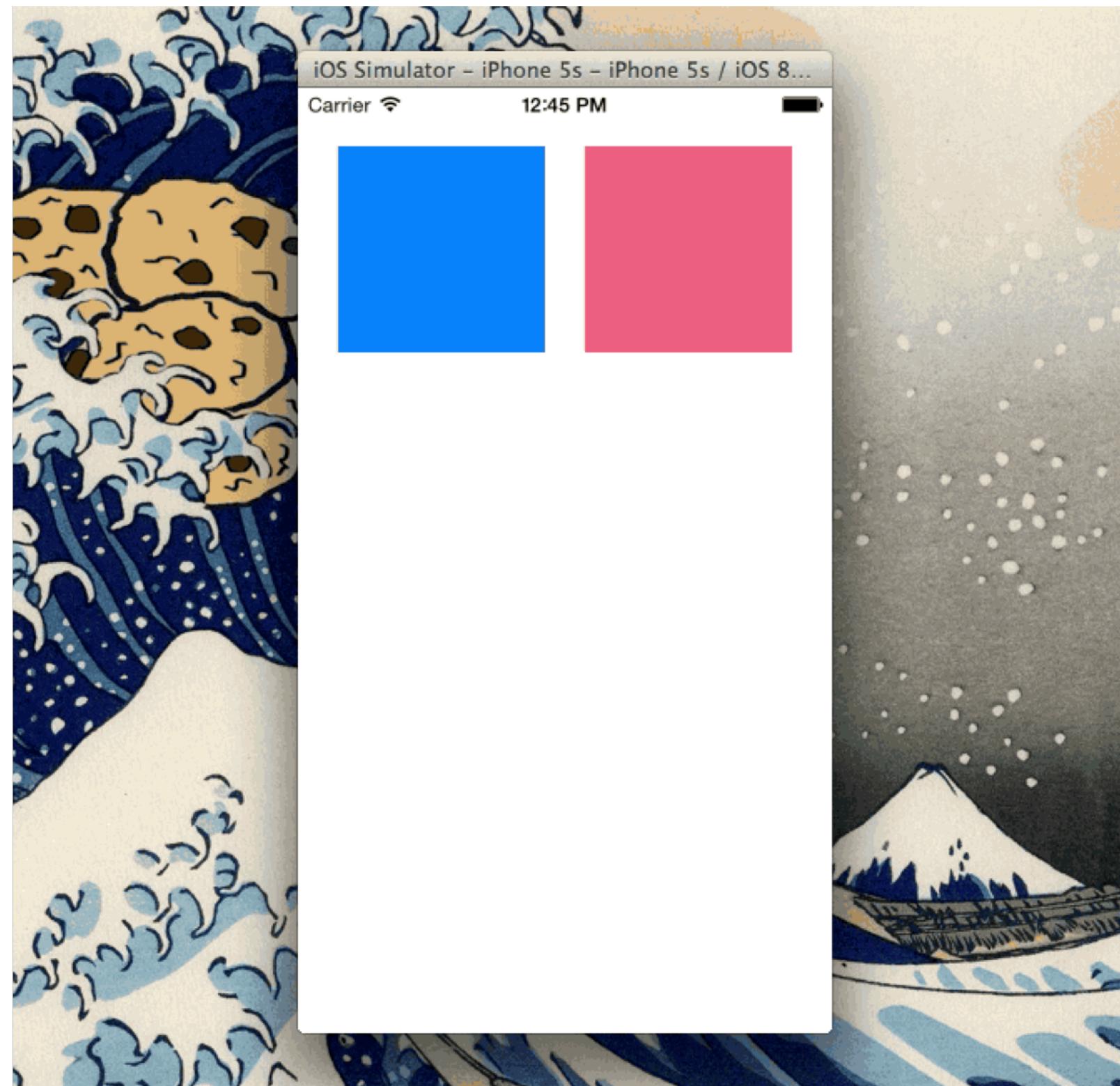
- Les contraintes sont la solution à tout ça
- Définissent comment les éléments sont positionnés par rapport aux autres
- “**TextView is relative to the vertical and horizontal center of the screen**”

Relativité (Constraints)



Relativité (Constraints)

```
let constraint = NSLayoutConstraint(item: view,  
                                    attribute: .TrailingMargin,  
                                    relatedBy: .Equal,  
                                    toItem: subView,  
                                    attribute: .TrailingMargin,  
                                    multiplier: 1,  
                                    constant: 0)  
  
self.addConstraint(constraint)
```



The end

Salle Mac

- Ne jamais éteindre les macs
- Fermer la session en partant
- Ne pas laisser de fichiers (sessions partagées)

Salle accessible en libre service,
tous les jours de Xh à XXh (si pas de cours)

Matériel à disposition

- 3 iPod touch 5
- 3 iPad mini 1
- 3 iPad mini 2 (écran retina)

Disponibles en TP et empruntables au secrétariat

Salle Mac

- Login : m2sar
- Password : sarm2