

iOS Guide

Kuixi Song

目录

- 工具
- Swift 入门
- Xcode 入门
- 更多资源

工具

- 硬件
 - MacBook
- 软件
 - Xcode
 - Sketch

20 分钟 Swift 入门

变量 & 常量

// 常量

```
let constant: String = "1"
```

```
constant = "2"    // 试图更改常量会报错
```

// 变量

```
var variable = "3"
```

```
variable = "4"
```

类型 & 类型转换

// 声明变量类型

```
let typeString: String = "5"
```

```
var noTypeString = "6" // 自动推断类型
```

```
let float: Float = 1.0
```

```
let double: Double = Double(float) // 类型转换
```

字符串插值

```
let lastName = "Kuixi"  
let firstName = "Song"  
let newString = "My name is \(lastName +  
firstName)"
```

数组 & 字典

// 数组

```
var fruits = ["Apple", "Orange", "Banana"]  
fruits[0]  
fruits.append("Strawberry")  
fruits.remove(at: 0)
```

// 字典

```
var fruitNumber: [String: Int] = ["Apple" : 1,  
"Orange" : 2, "Banana" : 3, "Watermelon" : 4]  
fruitNumber["Apple"] = 5
```


条件控制

```
// guard
let number = 0
func squareRoot(of number: Int)
{
    guard number >= 0 else {
        print("Negative
Number")
        return
    }
    // Do more..
}

// if
if number == 0 {
    print("Zero")
} else if number == 1 {
    print("One")
} else {
    print("Bigger than One")
}
```

```
// switch
switch number {
case 0:
    print("Zero")
case 1, 2: // 多个值
    print("One or Two")
case 3...5: // 区间
    print("Three to Five")
default:
    print("Bigger than Six")
}
```

循环

```
// for
var forLoop = 0
for i in 0..<3 {
    forLoop += i
}

// while
while forLoop > 1 {
    forLoop -= 1
}

// do-while
repeat {
    forLoop *= 2
} while (forLoop <= 2^10)
```

函数

```
// 函数
// func 函数名(参数列表) -> 返回值 {
//     函数体
// }
func myFunction(parameterA: Int, parameterB: Int) -> Int {
    return parameterA + parameterB
}
myFunction(parameterA: 3, parameterB: 5)

// 参数可省略名字, 且可以有多个返回值
func anotherFunction(_ omittedParameterName: Int) -> (Int, Int, Int) {
    return (omittedParameterName * 2, omittedParameterName + 2,
omittedParameterName - 2)
}
anotherFunction(100)

// 形参和实参名字可不同
func squared(of value: Double) -> Double {
    return value * value
}
squared(of: 20.0)
```

嵌套函数

```
// 嵌套函数
func addTen(to number: Int) -> Int {
    var temp = number
    func addFive(to number: Int) -> Int {
        return number + 5
    }
    temp = addFive(to: temp)
    temp = addFive(to: temp)
    return temp
}
addTen(to: 10)
```

函数是一等公民

```
// 函数是一等公民
func incrementer() -> ((Int) -> Int) {
    func addOne(to number: Int) -> Int {
        return number + 1
    }
    return addOne
}
var increment = incrementer()
increment(19)
```

闭包

```
// 闭包就是匿名函数
[1, 5, 3, 2, 4].sorted(by: { (a, b) -> Bool in
    return a < b
})
[1, 5, 3, 2, 4].filter {
    $0 % 2 == 0
}
[1, 5, 3, 2, 4].map {
    $0 * 2
}
// 算 n 的阶乘
func factorial(of n: Int64) -> Int64 {
    return (1...n).reduce(1, *)
}
factorial(of: 20)

let printMyName: ((_ name: String) -> Void) = { name in
    print("I am \(name)")
}
printMyName("KrayC")
```

可选值

```
// 可选值
var optionalValue: Int? // 默认为 nil
optionalValue = 3
optionalValue! // 强制解包, 尽量避免, 易导致 Crash
// 可选值绑定
if let realValue = optionalValue {
    print("My value is \$(realValue)")
}
// 默认值
print(optionalValue ?? 0)
```

类

```
// 类
class Shape {
    var numberOfSides = 0    // 属性
    var totalInnerAngleSum: Int {    // 计算属性
        get {
            return (numberOfSides - 2) * 180
        }
    }
}

// 初始化
init(sides: Int) {
    self.numberOfSides = sides
}

// 成员方法
func simpleDescription() -> String {
    return "A shape with \(numberOfSides) sides."
}

// 类方法
static func generateShape(numberOfSides: Int) -> Shape {
    return Shape(sides: numberOfSides)
}
}
```


对象

```
let s = Shape(sides: 4)
s.numberOfSides
s.totalInnerAngleSum
s.simpleDescription()
let anotherS = Shape.generateShape(numberOfSides: 3)
anotherS.simpleDescription()
```

父类与子类

```
class Square: Shape {  
    // 新增成员变量及方法  
    var length: Int = 0  
    func area() -> Int {  
        return length ^ 2  
    }  
  
    // 便捷构造器  
    convenience init(sideLength: Int) {  
        self.init(sides: 4)  
        self.length = sideLength  
    }  
  
    // 覆盖成员方法  
    override func simpleDescription() -> String {  
        return "A rectangle with \(numberOfSides) sides."  
    }  
}  
  
let square = Square(sideLength: 10)  
print(square.area())
```

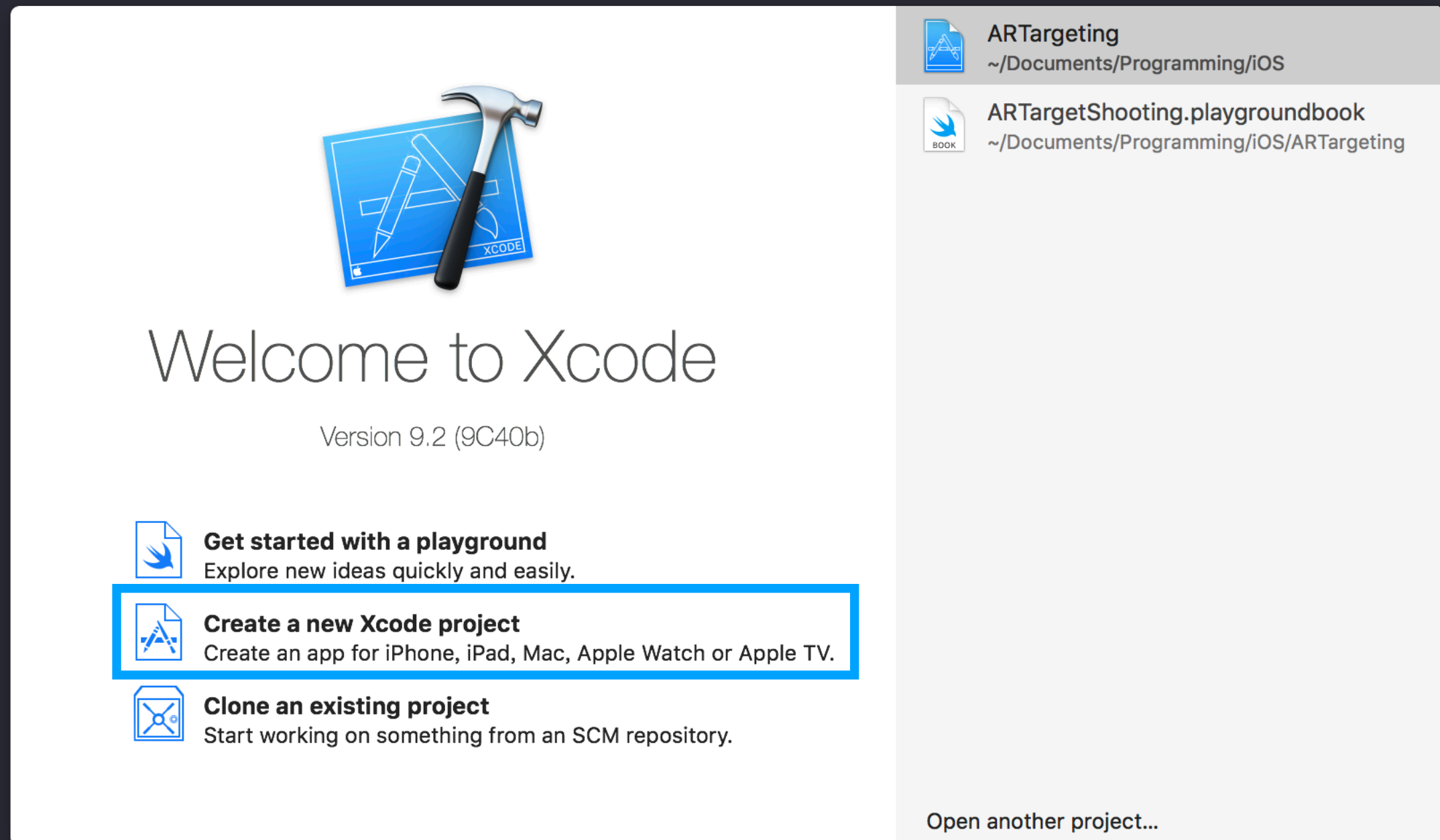
更多主题

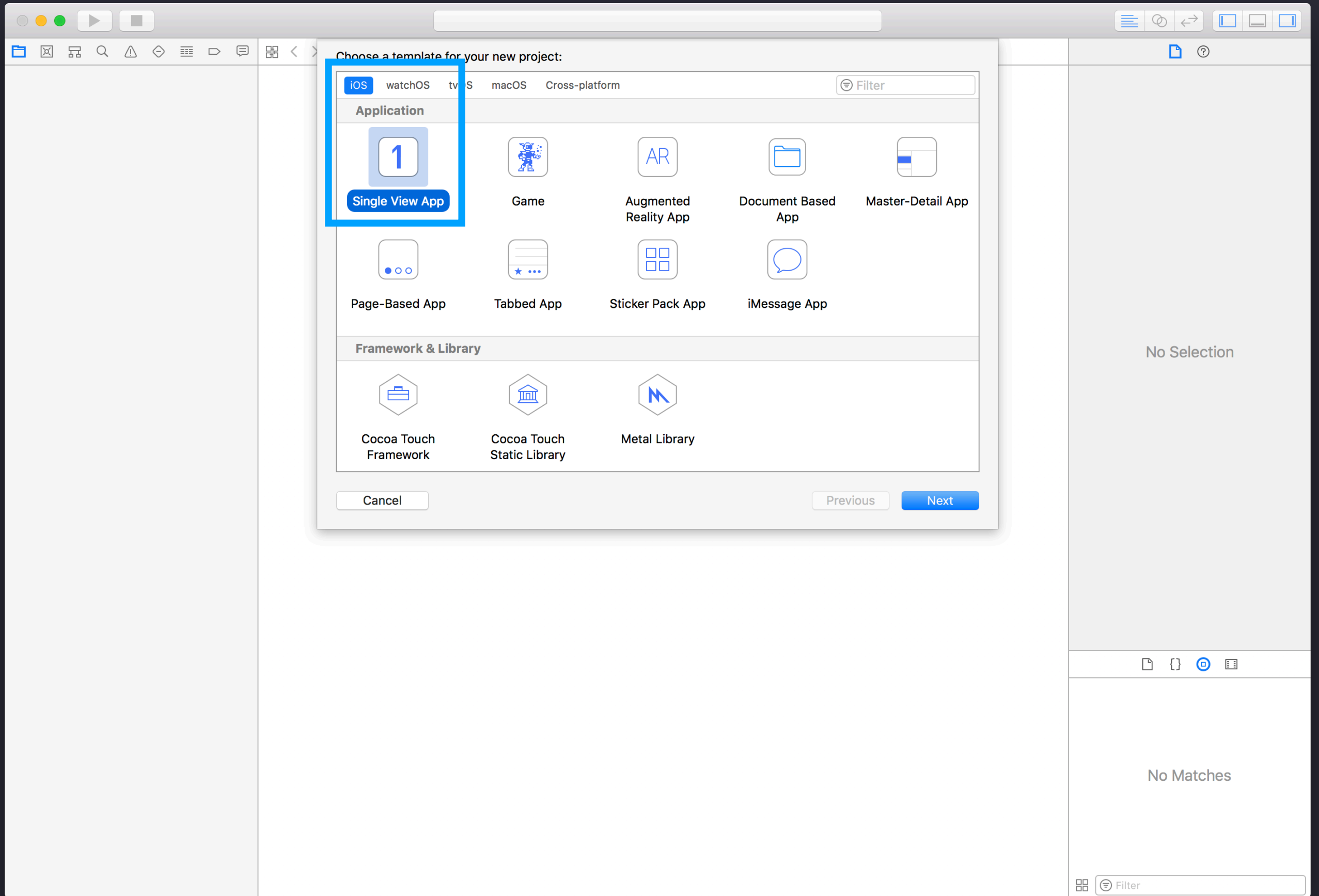
- extension
- protocol
- enumeration
- generic
- struct & class
-

20 分钟制作一个 App

Xcode

- 必须从苹果开发者网站 / Mac App Store 下载





▶

■

ELGuideProject > Kray.C's iPhone

ELGuideProject > ELGuideProject > Main.storyboard > Main.storyboard (Base) > No Selection

ELGuideProject

ELGuideProject

AppDelegate.swift

ViewController.swift

Main.storyboard

Assets.xcassets

LaunchScreen.storyboard

Info.plist

Products

View Controller Scene

View Controller

Identity and Type

NameMain.storyboard

TypeDefault - Interface Builder...

LocationRelative to Group

Containing directory

Full Path/Users/Kray/Desktop/ELGuideProject/ELGuideProject

Dev Region/Users/Kray/Desktop/ELGuideProject/ELGuideProject/Base.lproj/Main.storyboard

Localization

Base

English

Localizable Strings

Target Membership

ELGuideProject

Text Settings

Indent UsingSpaces

Widths44

TabIndent

Wrap lines

View Controller - A controller that manages a view.

Storyboard Reference - Provides a placeholder for a view controller in an external storyboard.

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

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

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

Tab Bar Controller - A controller that manages a set of view controllers.

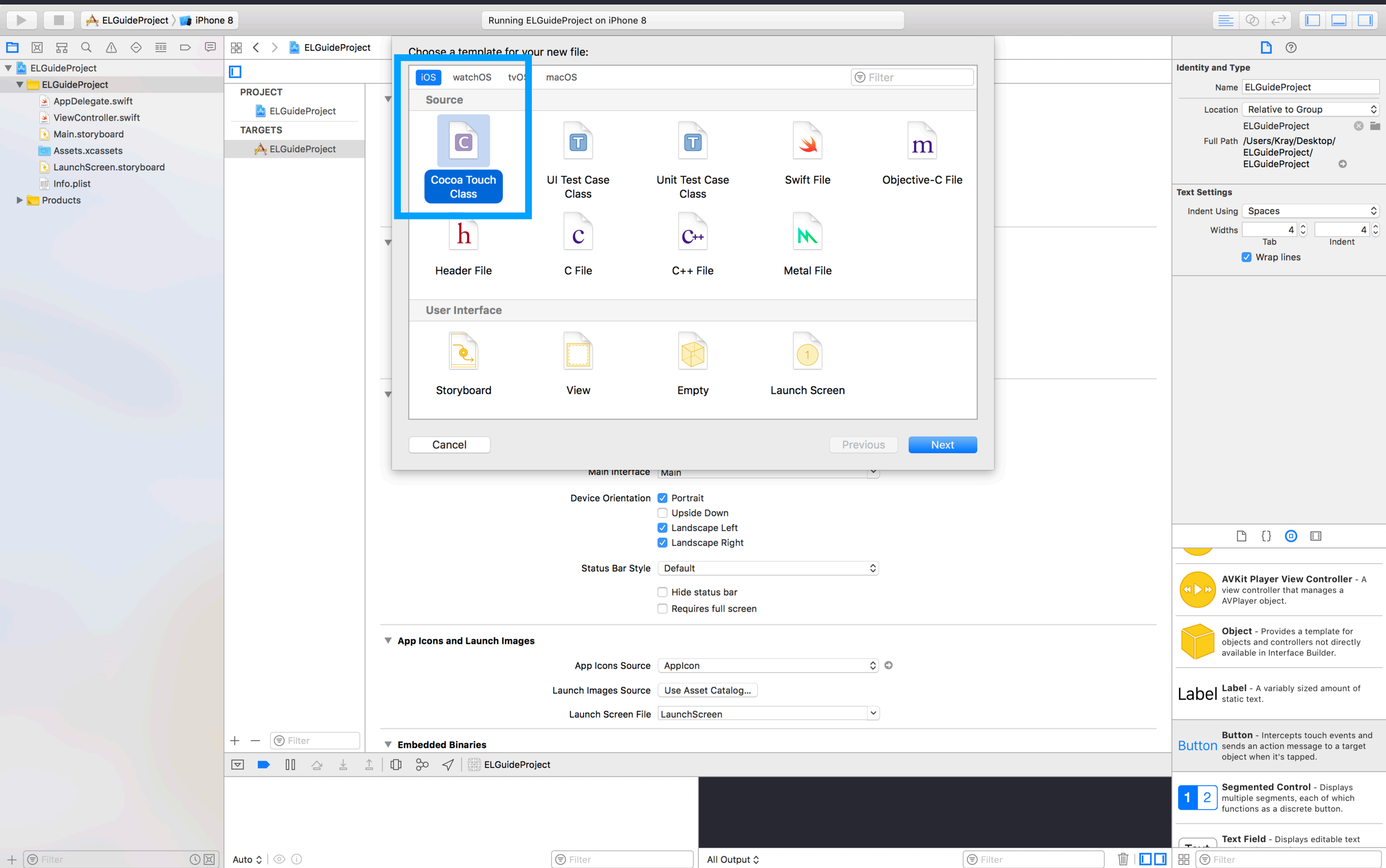
Filter

Filter

View as: iPhone 8 (wC hR)

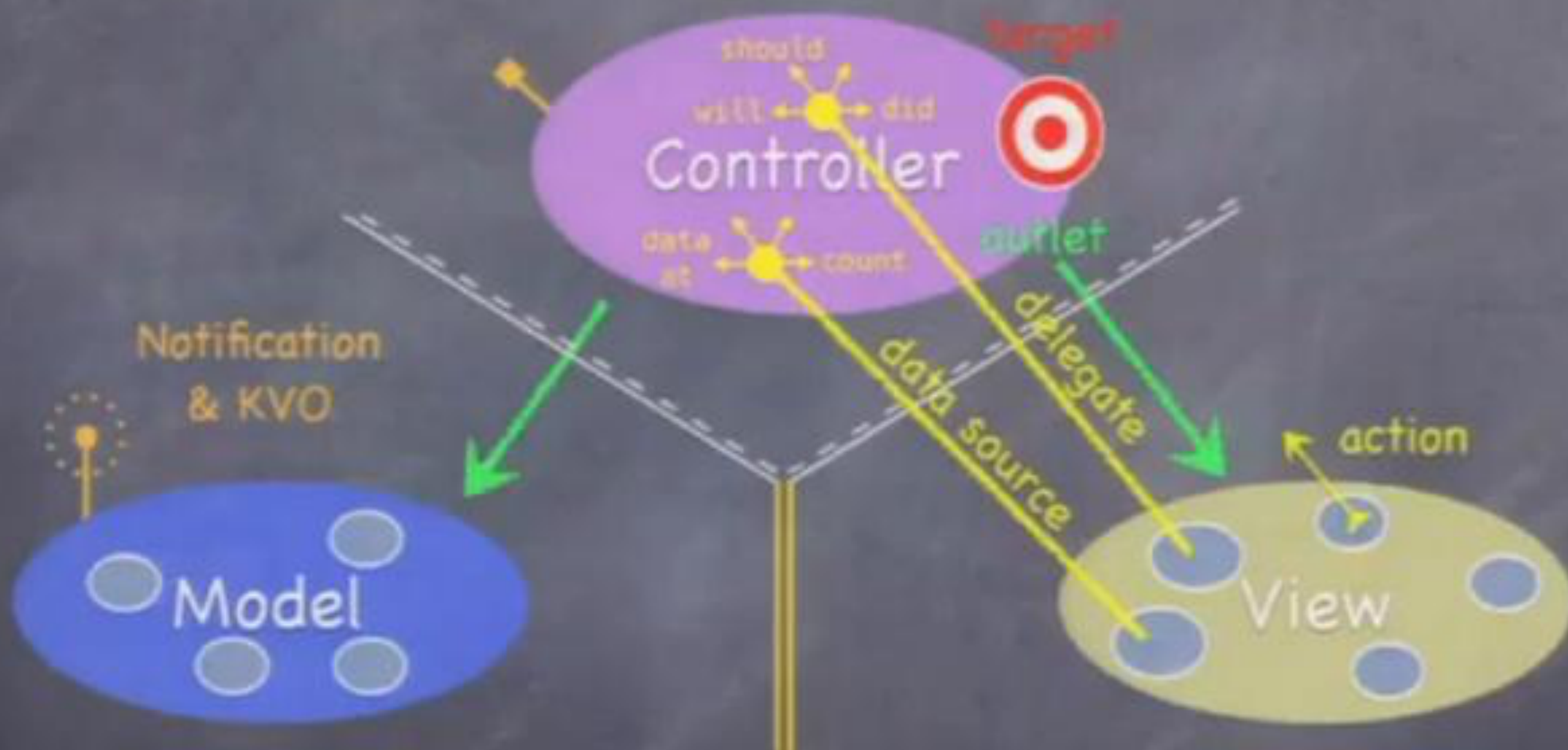
100%

Filter

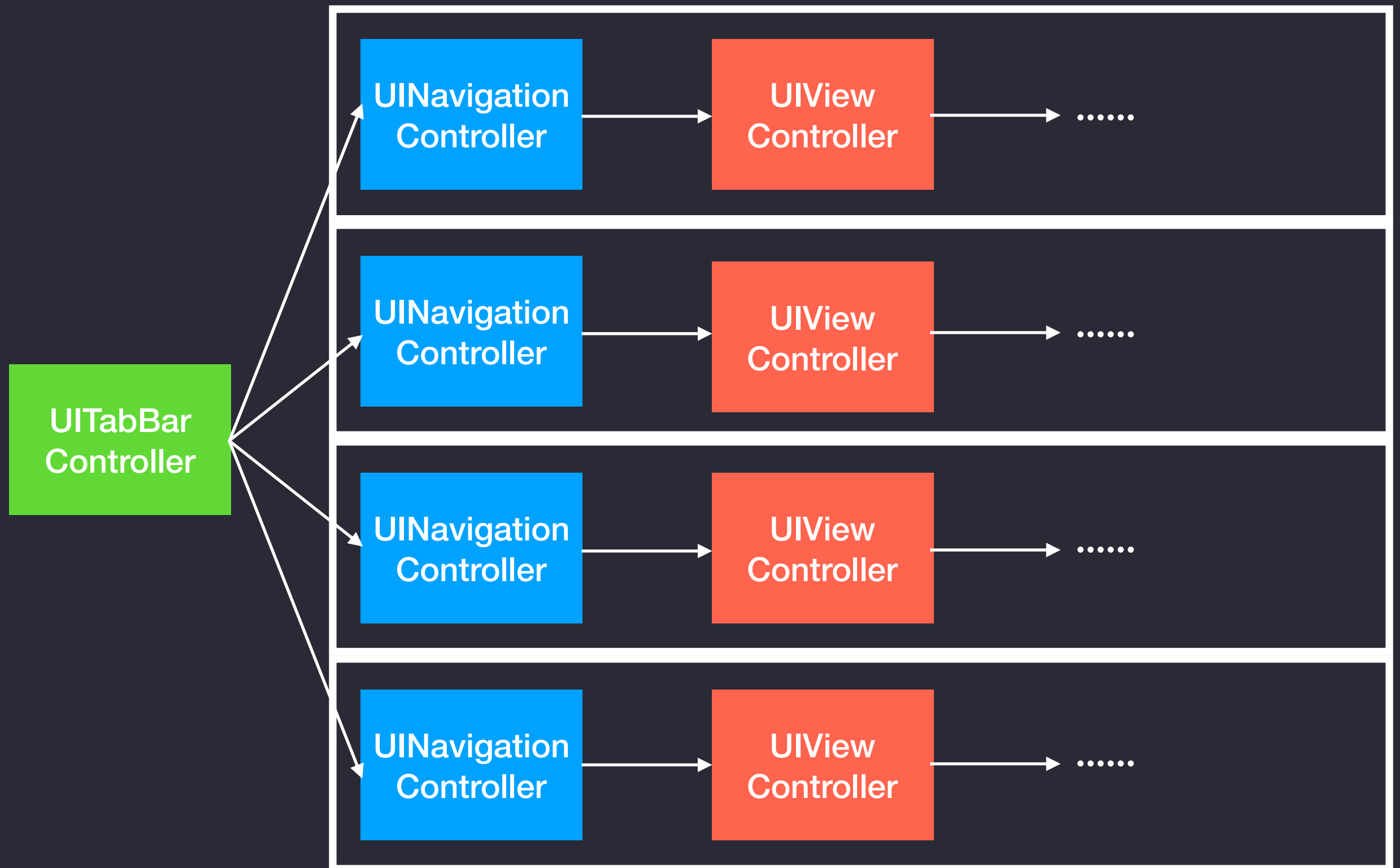


IOS MVC

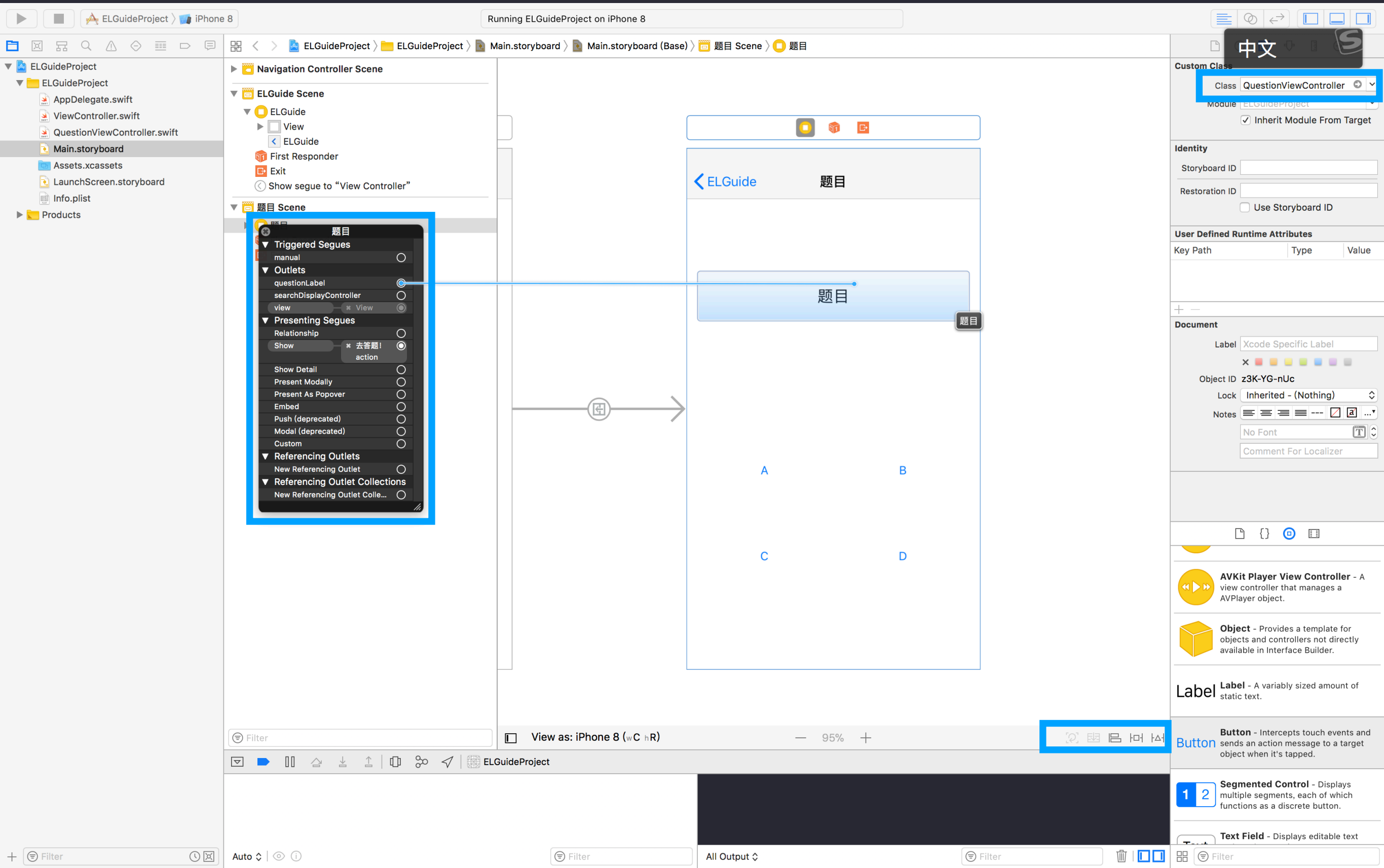
MVC



视图控制器层次结构



```
@IBOutlet weak var questionLabel: UILabel!  
@IBOutlet var choiceButtons: [UIButton]!
```



AutoLayout

Add New Alignment Constraints

<input type="checkbox"/> Leading Edges	
<input type="checkbox"/> Trailing Edges	
<input type="checkbox"/> Top Edges	
<input type="checkbox"/> Bottom Edges	
<hr/>	
<input type="checkbox"/> Horizontal Centers	
<input type="checkbox"/> Vertical Centers	
<input type="checkbox"/> Baselines	
<input checked="" type="checkbox"/> Horizontally in Container	0
<input type="checkbox"/> Vertically in Container	0

Add Constraints

Add New Constraints

100

20 20

153

Spacing to nearest neighbor

☒ Constrain to margins

<input type="checkbox"/> Width	335
<input type="checkbox"/> Height	50
<hr/>	
<input type="checkbox"/> Equal Widths	
<input type="checkbox"/> Equal Heights	
<input type="checkbox"/> Aspect Ratio	
<hr/>	
<input type="checkbox"/> Align	Leading Edges

Add Constraints



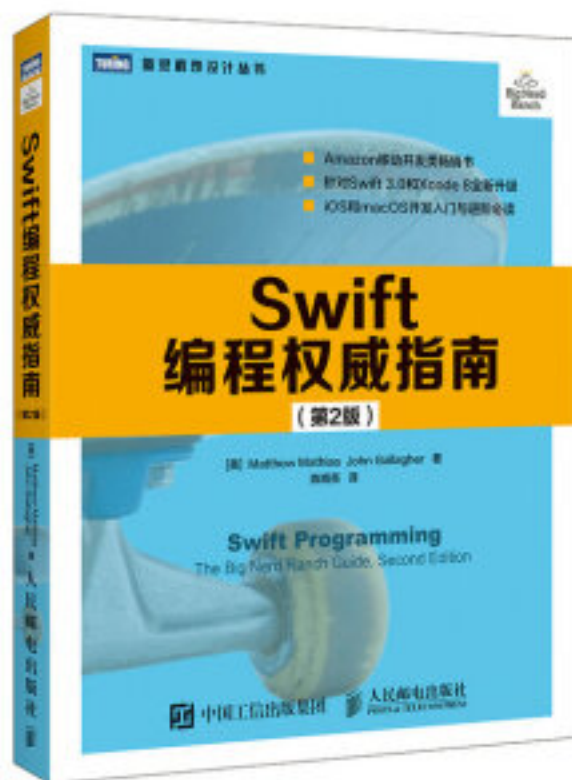
制作 UI 的方式

- Storyboard
- Xib
- 纯代码编写
- 建议：以上三者混用

善用第三方库

- 可以使用，但不要滥用
- 有精力的话，阅读源码，查看原理
- 导入方式
 - CocoaPods
 - Carthage
- 举例
 - 布局：Masonry / SnapKit
 - 网络请求：Alamofire
 - 图片加载：Kingfisher

参考资料



参考资料

- <https://developer.apple.com/documentation/>
- <https://objccn.io>
- <https://wizardforcel.gitbooks.io/w3school-swift/content/index.html>
- <https://github.com>

关于本次 ELGuide

- 源码 + PPT 仓库
 - <https://github.com/songkuixi/ELGuide2018>
- 联系方式
 - 宋奎熹 (WeChat) : krayc425
 - 鄢煜民 (QQ) : 979579683

感谢