

«SplitMe»

Fabrice.Kordon@lip6.fr



Goal of the example

Play with a UISplitView

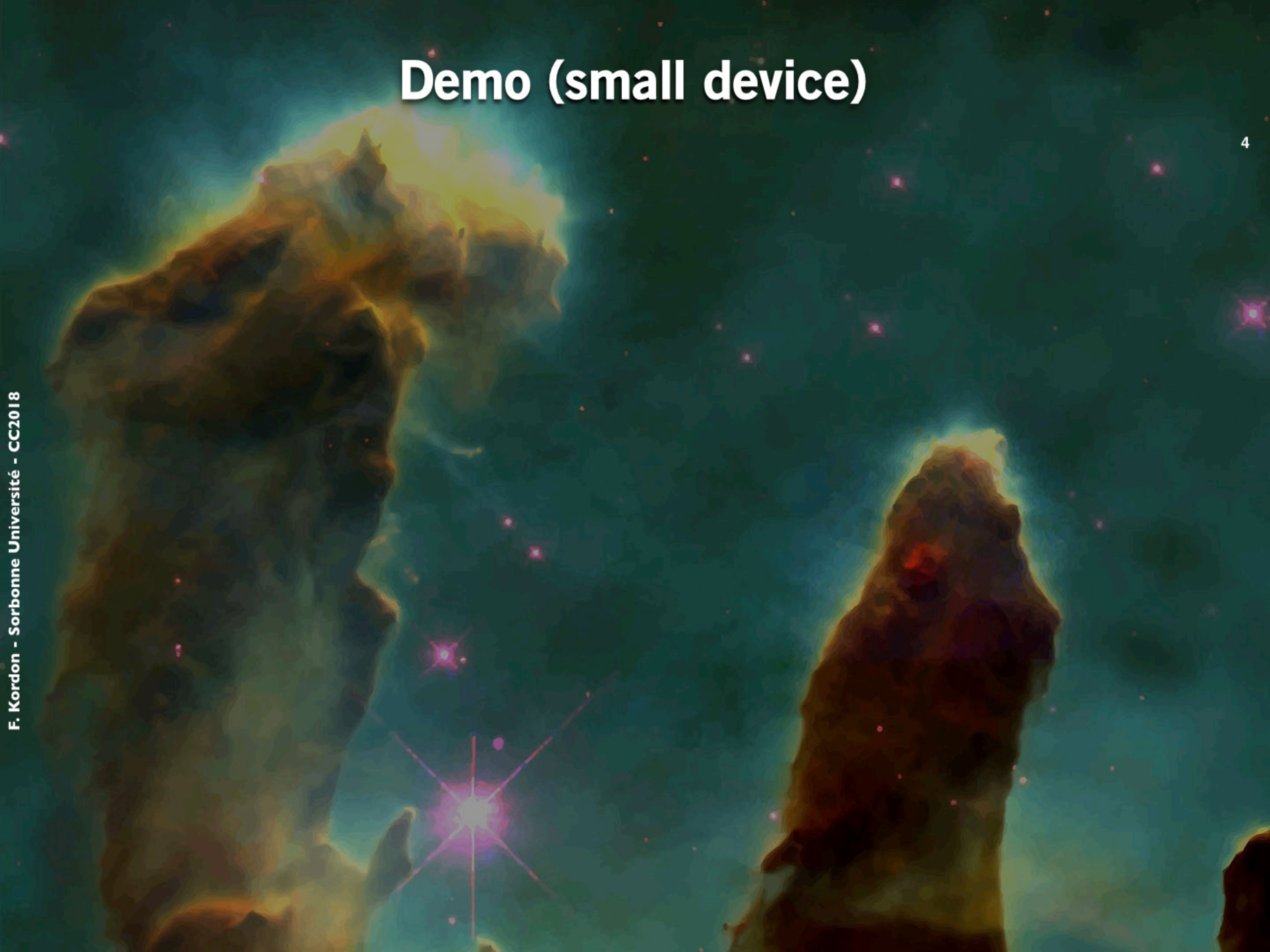
- On large device
- On small device
- Independently of a UITableViewController
 - ▶ These two mechanisms are independent
 - ▶ You will practice this later



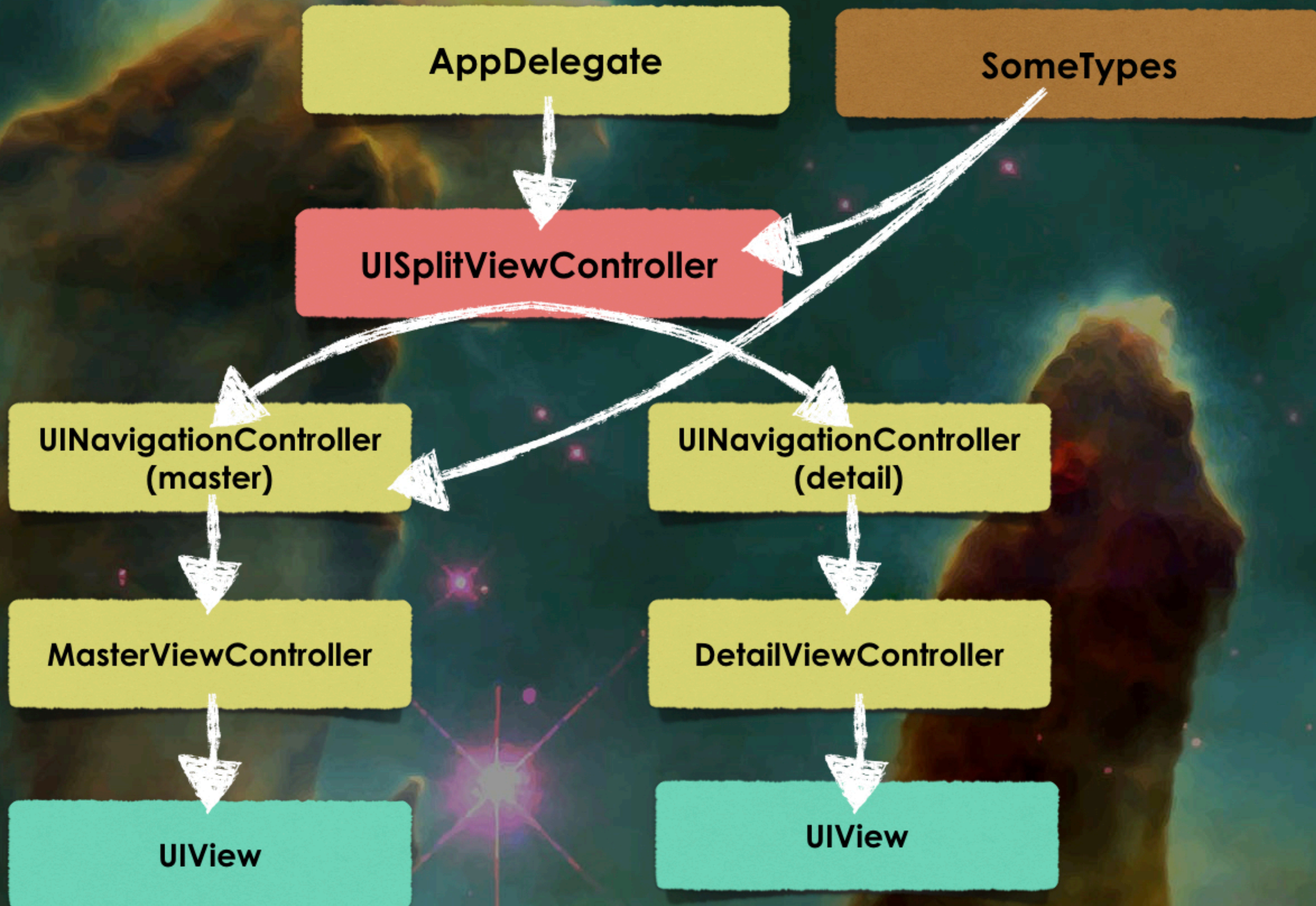
Demo (large device)



Demo (small device)



Architecture



SomeTypes

```
//  
// SomeTypes.swift  
// SplitMe  
//  
// Created by Fabrice Kordon on 29/10/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//  
  
import UIKit  
  
class SomeTypes: NSObject {  
    enum DeviceType {  
        case undefined // oups ;-)  
        case iphone35 // screen 3.5" = iPhone < 4s (being deprecated)  
        case iphone40 // screen 4" = iPhone 5, 5s  
        case iphone47 // screen 4.7" = iPhone 6, 6s, 7, 8  
        case iphone55 // screen 5.5" = iPhone 6+, 6s+, 7+, 8+  
        case iphone58 // screen 5.8" = iPhone X/XS  
        case iphone6x // screen 6.1" 6.5" & = iPhone XR & XS Max  
        case ipad97 // iPadmini, ipad & ipad pro (9.7")  
        case ipad105 // iPadPro 10.5"  
        case ipad127 // iPad pro (12.7")  
    }  
}
```

AppDelegate

7

```
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?
```

AppDelegate

```
private func identifyDevice () -> SomeTypes.DeviceType {
    var myDevice : SomeTypes.DeviceType?
    let h = UIScreen.main.bounds.size.height
    let w = UIScreen.main.bounds.size.width
    let i = UIDevice.current.userInterfaceIdiom
    // iPhone / iPod touch
    if i == .phone &&
        ((h < 568 && w == 320) || (w < 568 && h == 320)) {
        myDevice = .iphone35
    } else if i == .phone &&
        ((h == 568 && w == 320) || (w == 568 && h == 320)) {
        myDevice = .iphone40
    } else if i == .phone &&
        ((h == 667 && w == 375) || (w == 667 && h == 375)) {
        myDevice = .iphone47
    } else if i == .phone &&
        ((h == 736 && w == 414) || (w == 736 && h == 414)) {
        myDevice = .iphone55
    } else if i == .phone &&
        ((h == 812 && w == 375) || (w == 812 && h == 375)) {
        myDevice = .iphone58
    } else if i == .phone &&
        ((h == 896 && w == 414) || (w == 896 && h == 414)) {
        myDevice = .iphone6x
    }
    // iPads
    } else if i == .pad &&
        ((h == 1024 && w == 768) || (w == 1024 && h == 768)) {
        myDevice = .ipad97
    } else if i == .pad &&
        ((h == 1112 && w == 834) || (w == 1112 && h == 834)) {
        myDevice = .ipad105
    } else if i == .pad &&
        ((h == 1366 && w == 1024) || (w == 1366 && h == 1024)) {
        myDevice = .ipad127
    }
}
return myDevice!
}
```


AppDelegate

7

```
func application(_ application: UIApplication,
                 didFinishLaunchingWithOptions
                 launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
    // Override point for customization after application launch.
    let myDevice = self.identifyDevice()
    let svc = UISplitViewController()
    let dvc = DetailViewController()
    let mvc = MasterViewController(myDevice, detail: dvc, split: svc)
    let mnvc = UINavigationController(rootViewController: mvc) // must be there
    let dnvc = UINavigationController(rootViewController: dvc) // must be there
    svc.viewControllers = [mnvc, dnvc]
    svc.delegate = mvc
    switch myDevice {
    case .iphone35, .iphone40, .iphone47:
        svc.preferredDisplayMode = .automatic
    default:
        svc.preferredDisplayMode = .primaryOverlay
    }

    window?.rootViewController = svc
    window?.makeKeyAndVisible()
    return true
}

// Other methods do not changes
}
```

MasterViewController



Sake of simplicity...

Code located in a
ViewController

MasterViewController

```
import UIKit

class MasterViewController: UIViewController, UISplitViewControllerDelegate {

    private let b1 = UIButton(type: .system)
    private let b2 = UIButton(type: .system)
    private let b3 = UIButton(type: .system)
    var fullWatch = false

    var splitVC : UISplitViewController?
    var myDevice = SomeTypes.DeviceType.undefined
    var detailVC : DetailViewController?

    init(_ devType : SomeTypes.DeviceType,
         detail : DetailViewController,
         split : UISplitViewController) {
        super.init(nibName: nil, bundle: nil)
        splitVC = split
        detailVC = detail
        myDevice = devType
    }

    required init?(coder aDecoder: NSCoder) {
        fatalError("init(coder:) has not been implemented")
    }
}
```

MasterViewController

```
override func viewDidLoad() {
    super.viewDidLoad()
    // Do any additional setup after loading the view, typically from a nib.
    self.navigationItem.title = "Master"
    self.view = UIView()
    self.view.backgroundColor = UIColor.white
    b1.setTitle("Yellow detail", for: .normal)
    b1.addTarget(self, action: #selector(doDetail(sender:)), for: .touchDown)
    self.view.addSubview(b1)
    b2.setTitle("Blue detail", for: .normal)
    b2.addTarget(self, action: #selector(doDetail(sender:)), for: .touchDown)
    self.view.addSubview(b2)
    switch myDevice {
    case .ipad97, .ipad105, .ipad127 :
        b3.setTitle("both together?", for: .normal)
        b3.addTarget(self, action: #selector(flipFlapSeeAll), for: .touchDown)
        self.view.addSubview(b3)
    default: ()
    }
    // fetching the width of the status bar
    let s = CGSize(width: (self.navigationController?.navigationBar.frame.size.width)
                    height: UIScreen.main.bounds.size.height)

    self.displayInFormat(size: s)
}
```

MasterViewController

```
@objc func doDetail(sender: UIButton) {
    if sender === b1 {
        detailVC?.yellowView()
    } else {
        detailVC?.blueView()
    }
    splitVC?.showDetailViewController(detailVC!.navigationController!, sender: self)
}

@objc func flipFlapSeeAll() {
    fullWatch = !fullWatch
    if fullWatch {
        splitVC?.preferredDisplayMode = .allVisible
        b3.setTitle("overlap?", for: .normal)
    } else {
        splitVC?.preferredDisplayMode = .primaryOverlay
        b3.setTitle("both together?", for: .normal)
    }
}

func displayInFormat(size : CGSize) {
    b1.frame = CGRect(x: 10, y: 100, width: 120, height: 30)
    b2.frame = CGRect(x: 10, y: 150, width: 120, height: 30)
    b3.frame = CGRect(x: 10, y: 200, width: 120, height: 30)
    self.view.frame = CGRect(x: 0, y: 0,
                             width: size.width,
                             height: size.height)
}
}
```

DetailViewController



Sake of simplicity...

Code located in a
ViewController

DetailViewController

```
import UIKit

class DetailViewController: UIViewController {

    private let v = UIView()
    private let l = UILabel()

    override func viewDidLoad() {
        super.viewDidLoad()
        l.textAlignment = .left
        l.frame = CGRect(x: 30, y: 100, width: 200, height: 20)
        v.addSubview(l)
        self.view = v
        self.navigationItem.title = "Detail"
    }

    func yellowView() {
        v.backgroundColor = UIColor(red:0.937, green:0.981, blue:0.502, alpha:1)
        l.text = "Secondary view (yellow)"
        self.navigationItem.title = "Yellow detail"
    }

    func blueView() {
        v.backgroundColor = UIColor(red:0.498, green:0.7, blue:0.981, alpha:1)
        l.text = "Secondary view (blue)"
        self.navigationItem.title = "Blue detail"
    }
}
```

DetailViewController

```
override func viewWillTransition(to size: CGSize,  
    with coordinator: UINavigationControllerTransitionCoordinator) {  
    print("size secondary = \(size)")  
    self.view?.frame = CGRect(x: 0, y: 0,  
        width: size.width, height: size.height)  
}  
}
```


As a conclusion...

You know how it works

- The «very pure» mechanisms
- Remind that the master view controllers drive the secondary
 - ▶ Provide appropriate links between the views
 - ▶ Cross references between master and detail?
- Remember, one controller may be a UITableViewController

A remark

- Handling of the UITableViewControllerDelegate protocol
 - ▶ In the master view
 - ▶ In a «main view controller» which inherits from UISplitViewController