

«WatchMe»

Fabrice.Kordon@lip6.fr



Goal of the example

2



ALFA Romeo show room



Display images

- Some nice cars...
- With explanations
- let you manipulate some basics
 - ▶ Storyboard for WatchOS
 - ▶ The main menu
 - ▶ Static and dynamic view handling

Goal of the example



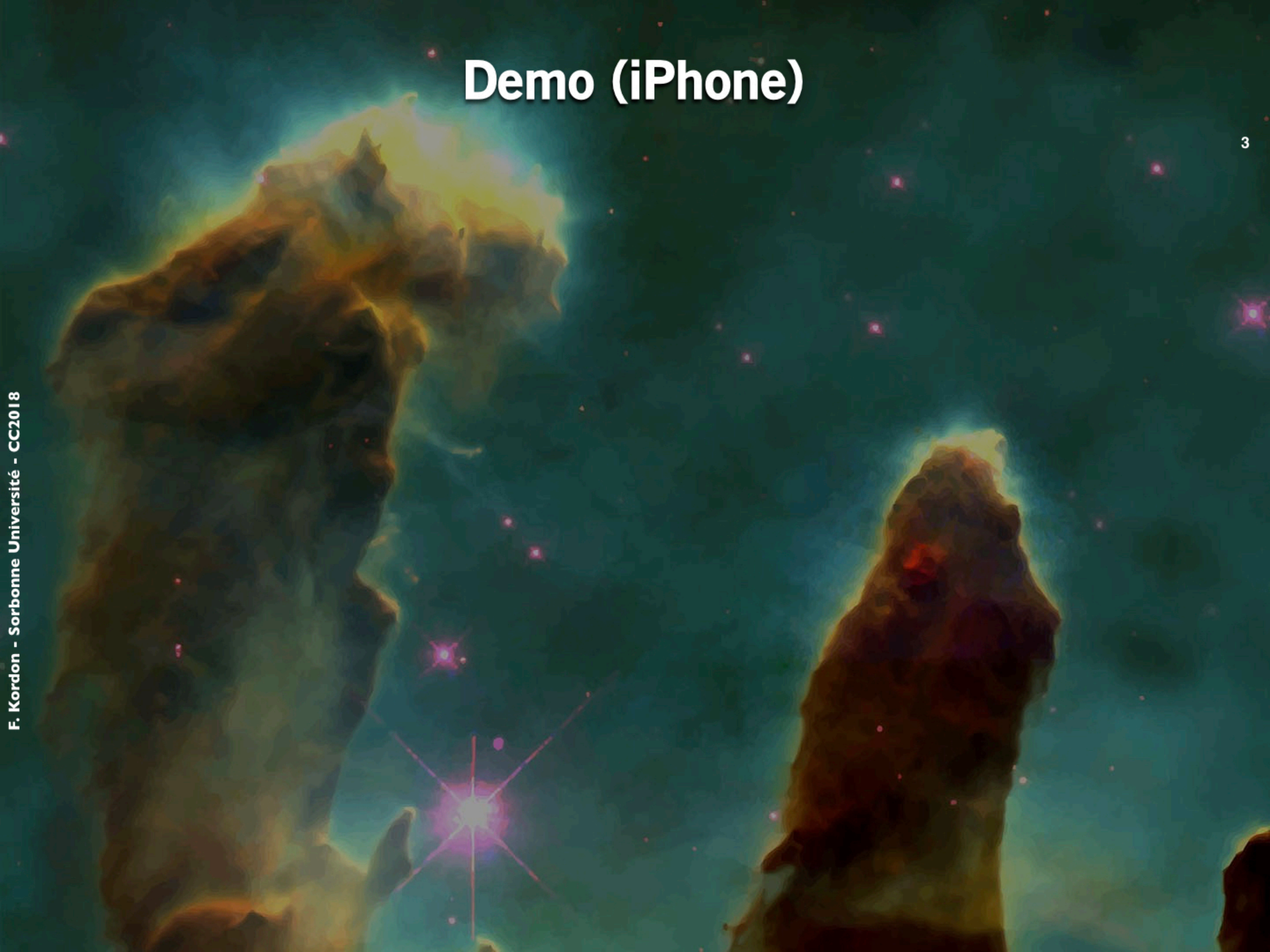
ALFA Romeo show room



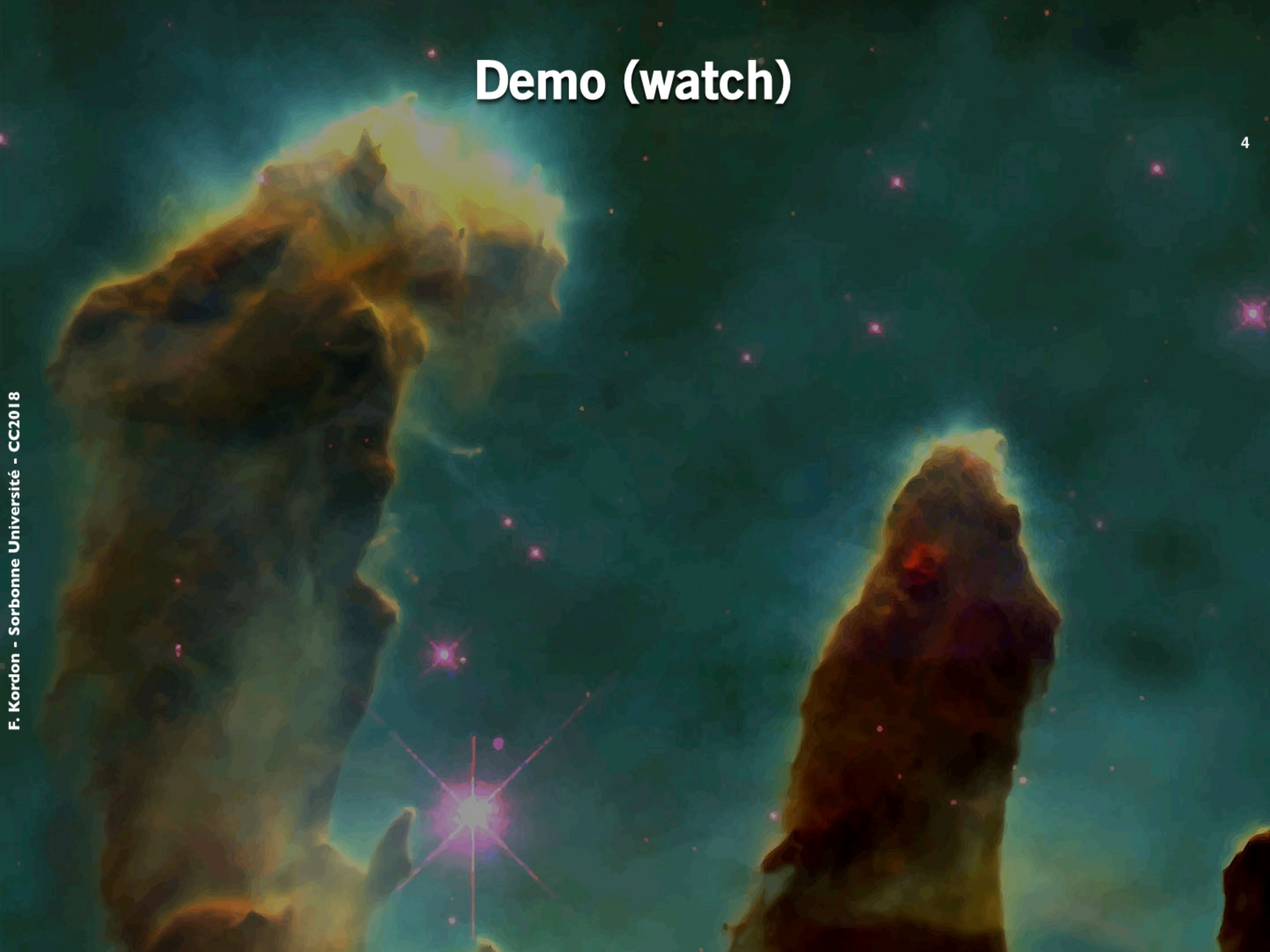
Display images

- Some nice cars...
- With explanations
- let you manipulate some basics
 - ▶ Storyboard for WatchOS
 - ▶ The main menu
 - ▶ Static and dynamic view handling

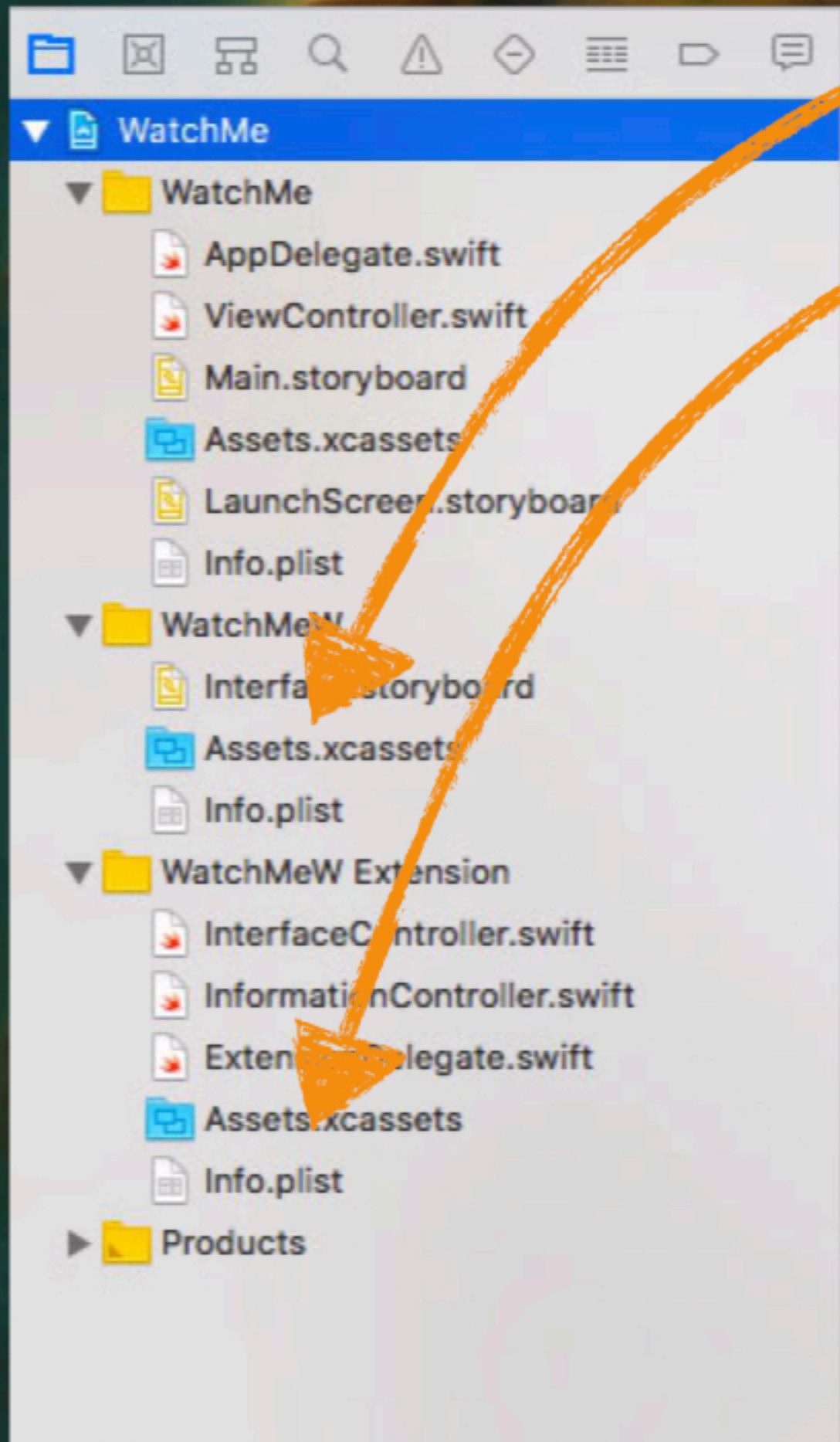
Demo (iPhone)



Demo (watch)



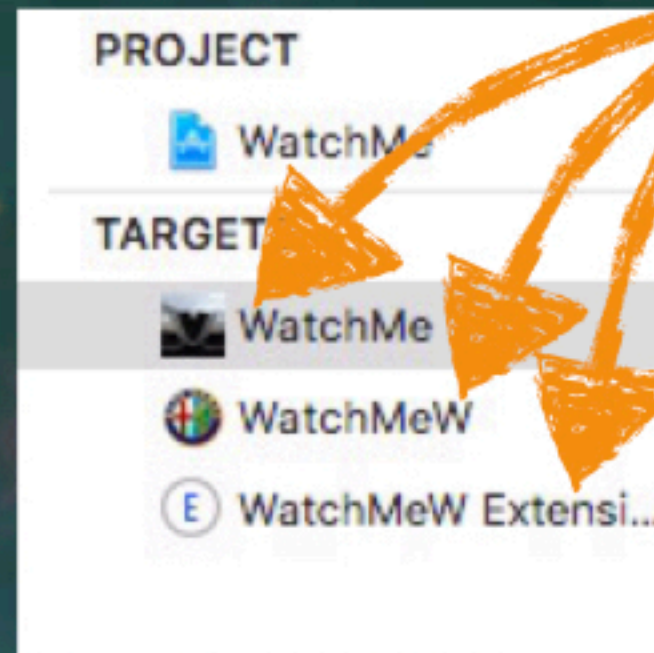
Structure of the project



images & icons in the watch

images in the watch

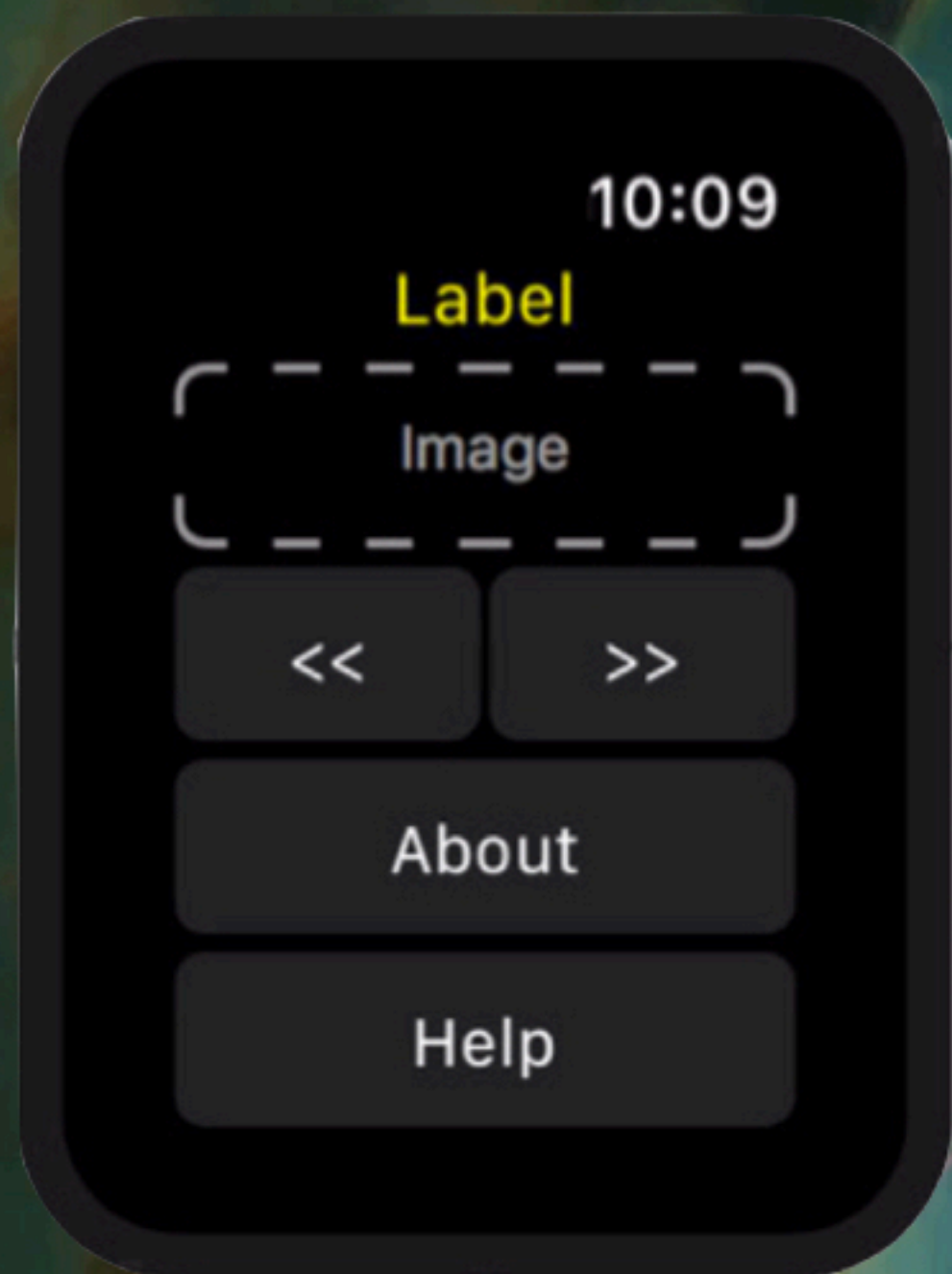
Three targets



The storyboard



The storyboard



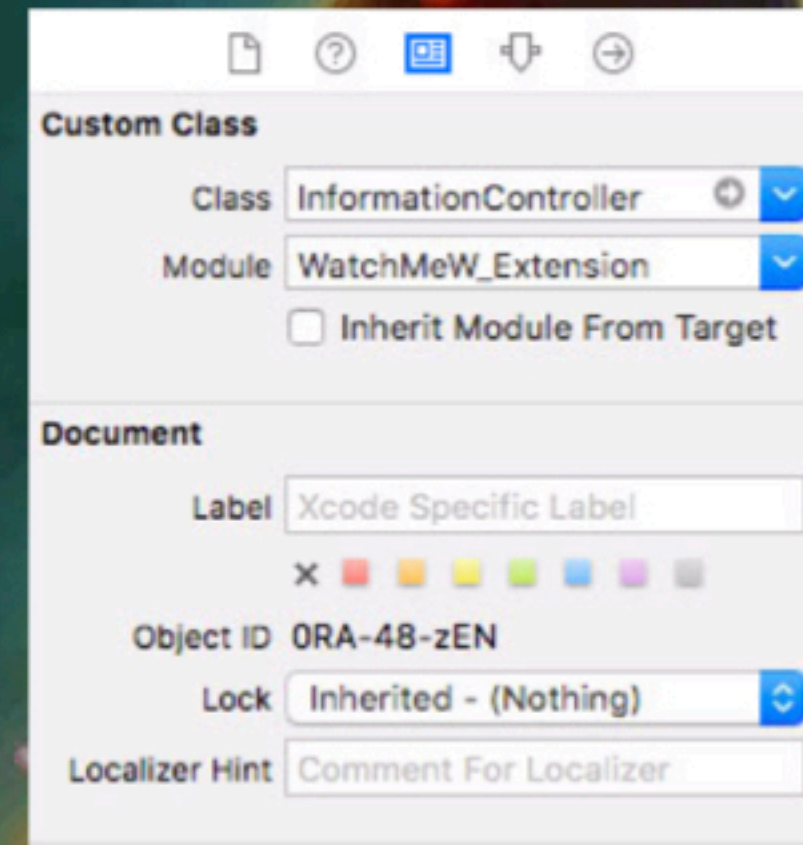
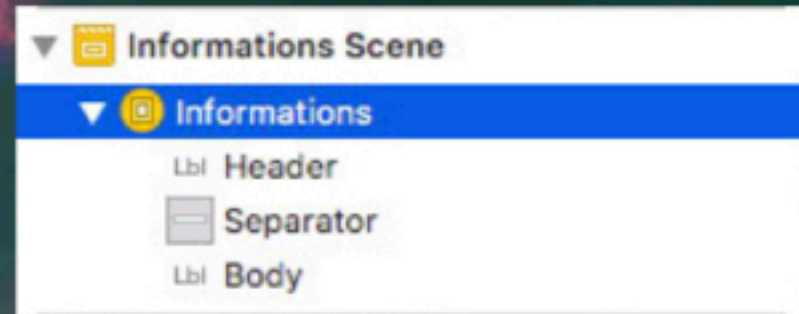
The screenshot shows the 'Interface Controller Scene' hierarchy in Xcode. It includes an 'Interface Controller' with a 'Menu' (containing 'Description', 'Transparent', and 'Opaque') and a 'Group' (containing 'Title Label', 'Car Image', and another 'Group'). The second 'Group' contains buttons for '<<', '>>', 'About', and 'Help'. A legend at the bottom indicates that the arrow icon represents the 'Main Entry Point', the square icon represents 'Modal segue to "/>

The screenshot shows the 'Identity and Type' inspector for the storyboard. The 'Name' is 'Interface.storyboard' and the 'Type' is 'Default - Interface Builder...'. The 'Location' is 'Relative to Group' and the 'Full Path' is '/Users/kitens1/Desktop/WatchMe/WatchMeW/Base.lproj/Interface.storyboard'. The 'On Demand Resource Tags' section is empty. The 'Interface Builder Document' section shows 'Opens in' set to 'Latest Xcode (9.0)', 'Builds for' set to 'Deployment Target (5.0)', 'Global Tint' set to 'Default', and 'Page Direction' set to 'Horizontal'. The 'Localization' section shows 'Base' checked and 'English' unchecked. The 'Target Membership' section shows 'WatchMe' and 'WatchMeW' checked, and 'WatchMeW Extension' unchecked.

The storyboard



What about the «main app»?
No need to implement something!



InterfaceController

7

```
import WatchKit
import Foundation
```

```
class InterfaceController: WKInterfaceController {
```

```
    private let tabTitle = ["Alfetta GTV", "Spider Duetto",
                            "Giulia Spider", "6C 2500", "Giulia GT", "1900 SSZ",
                            "Montréal", "GT", "Alfasud Sprint"]
```

```
    private let tabExplanation = [
```

```
        "This car was built between 1974 and 1987 (production: 140000 units). The photo shows the second revision (after 1980). TH 2.5 V6 liter was a must.",
```

```
        "This car was built between 1966 and 1993. The photo shows the second revision (1969-1983). This car became famous because it was the one of Dusting Hoffman in «the laureate».",
```

```
        "This car was built between 1955 and 1965. It is a must of the 1950's, symbol of the dolce vita.",
```

```
        "This car is one of the last Alfa 6C in version 2500 (manufactured after WW2 based on the model of the late 1930's). It was sold until 1951.",
```

```
        "This car was built between 1955 and 1965. The «Junior» version was very successful. It was a GTA version, lighter and also more powerful, dedicated to races (production: 124 units).",
```

```
        "This car is a small special series designed by Zagato in 1956 based on an Alfa 1900",
```

```
        "This is the public version derived from a «dream car» presented in a Montréal international exhibition in 1967. Only 3925 units were produced between 1971 and 1974 (but sold until 1977). Embedded with a 1.6 liter engine, it is a victim of the the first petrol shock.",
```

```
        "This car was first presented in the motor show of Geneva in 2003 and then produced in a limited edition. It was very successful (production: 80832 units). It was also popularized in several video games like «Test Drive Unlimited».",
```

```
        "This car is a muscled version of the Alfasud TI manufactured in the Pomigliano d'Arco area, south of Italy. Elle was produced between 1976 and 1989." ]
```

```
    private var displayed = 1
```

```
    @IBOutlet weak var titleLabel: WKInterfaceLabel!
    @IBOutlet weak var carImage: WKInterfaceImage!
```



InterfaceController

```
override func awake(withContext context: Any?) {
    super.awake(withContext: context)
    // Configure interface objects here.
    self.display()
}

func display () {
    titleLabel.setText(tabTitle[displayed-1])
    // image on the Phone
    //carImage?.setImage(UIImage(String(format:"alfa-%u", displayed)))
    // image on the watch
    carImage?.setImageNamed(String(format:"alfa-%u", displayed))
}

override func willActivate() {
    super.willActivate()
}

override func didDeactivate() {
    super.didDeactivate()
}
```


InterfaceController

// Local buttons handled programmatically

```
@IBAction func next() {  
    if displayed == tabPage.count {  
        displayed = 1  
    } else {  
        displayed += 1  
    }  
    self.display()  
}
```

```
@IBAction func previous() {  
    if displayed == 1 {  
        displayed = tabPage.count  
    } else {  
        displayed -= 1  
    }  
    self.display()  
}
```



InterfaceController

```
// Menu buttons

@IBAction func describe() {
    self.presentController(withName: "Information",
        context: [tabTitle[displayed-1], tabExplanation[displayed-1]])
}

@IBAction func setTransparency() {
    carImage.setAlpha(0.5)
}

@IBAction func setOpaque() {
    carImage.setAlpha(1.0)
}
```



InformationController

```
import WatchKit
import Foundation

class InformationController: WKInterfaceController {

    @IBOutlet weak var header: WKInterfaceLabel!
    @IBOutlet weak var body: WKInterfaceLabel!

    override func awake(withContext context: Any?) {
        super.awake(withContext: context)
        // Configure interface objects here.
        let ctx = context! as! [String]
        header.setText(ctx[0])
        body.setText(ctx[1])
    }

    override func willActivate() {
        super.willActivate()
    }

    override func didDeactivate() {
        super.didDeactivate()
    }
}
```



As a conclusion...



Easy is'n't it?

- Nice add-on to many applications
- Huge planned development for health Apps
 - ▶ Nice from a medical perspective
 - ▶ But raises numerous societal problems



Data exchange between watch and phone?

- It is possible
- Mentioned in a next video