

«CustomTable»

Fabrice.Kordon@lip6.fr



As an introduction...

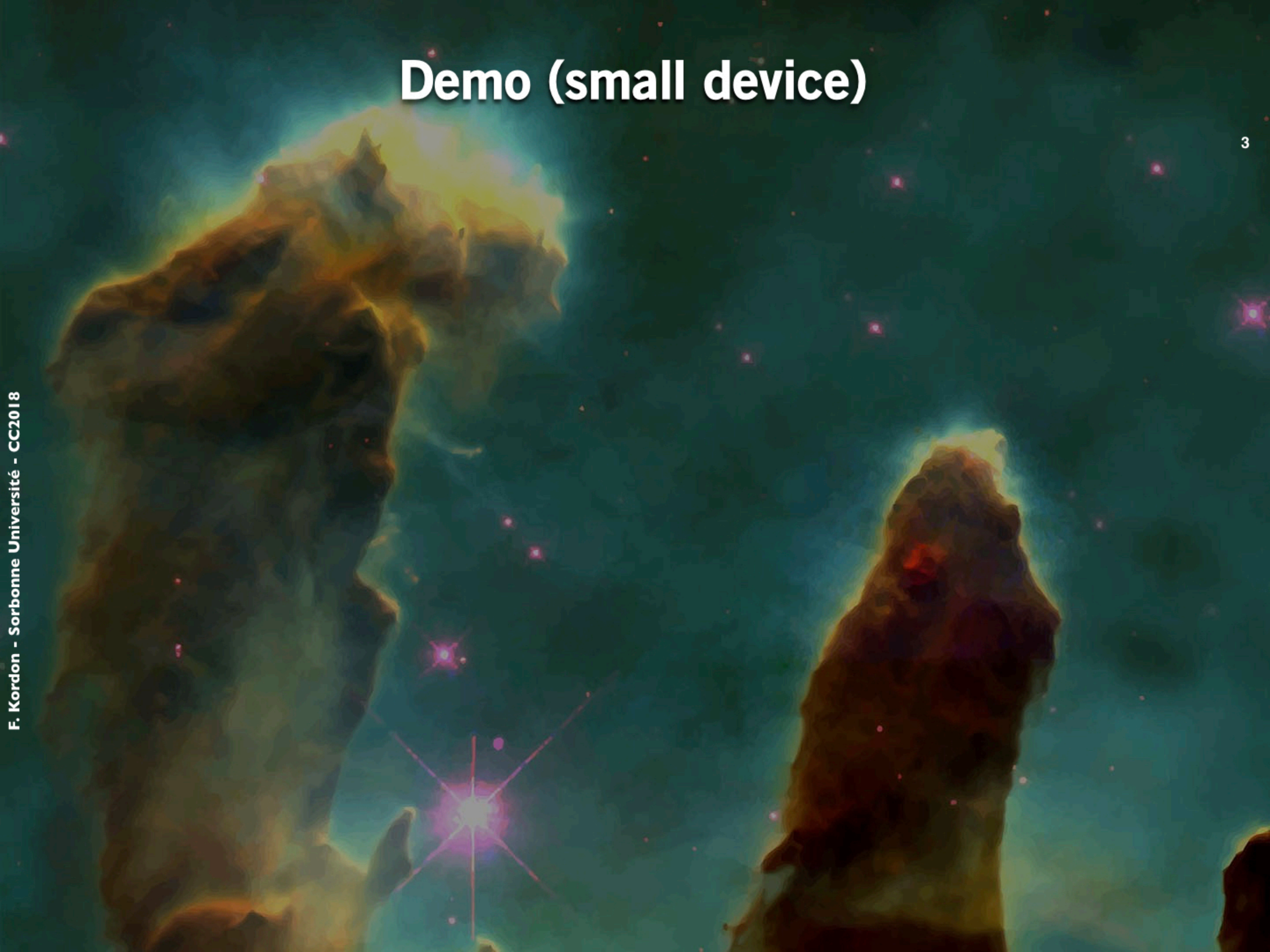
Goals of the example

- Illustrate the customisation of a UITableView
- Illustrate modifications in a UITableView
- Illustrate the use of a UINavigationController
 - ▶ **Push/pop view controllers**

Extension of a previous example

- MyTableView

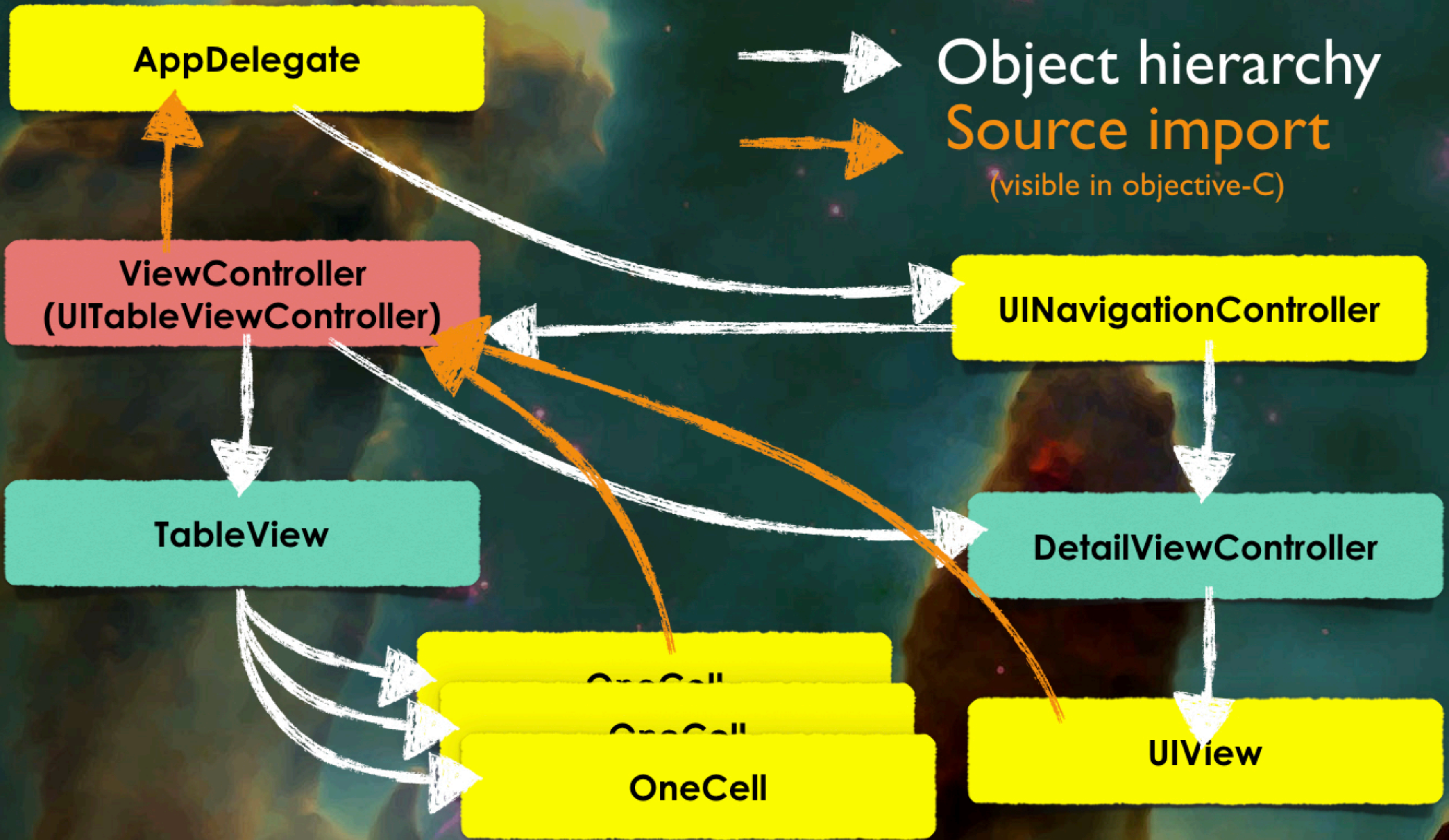
Demo (small device)



Demo (large device)



Architecture



Object hierarchy
Source import
(visible in objective-C)

AppDelegate

```
import UIKit

@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {

    var window: UIWindow?

    func application(_ application: UIApplication,
                    didFinishLaunchingWithOptions
                    launchOptions: [UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        // Override point for customization after application launch.
        let tvc = UITableView()
        let nvc = UINavigationController(rootViewController: tvc)
        window!.rootViewController = nvc
        window!.makeKeyAndVisible()
        return true
    }

    func applicationWillResignActive(_ application: UIApplication) {...}
    func applicationDidEnterBackground(_ application: UIApplication) {...}
    func applicationWillEnterForeground(_ application: UIApplication) {...}
    func applicationDidBecomeActive(_ application: UIApplication) {...}
    func applicationWillTerminate(_ application: UIApplication) {...}
}
```



No change
But you understand the code now

OneCell

7

```
//  
// OneCell.swift  
// MyTableView  
//  
// Created by Fabrice Kordon on 28/10/2018.  
// Copyright © 2018 Sorbonne Université. All rights reserved.  
//
```

```
import UIKit
```

```
class OneCell: NSObject {
```

```
    var label = ""
```

```
    var detail = ""
```

```
    init(l : String, d : String) {
```

```
        label = l
```

```
        detail = d
```

```
    }
```

```
}
```



No change too

No need to detail

DetailViewController

```
import UIKit

class DetailViewController: UIViewController {

    private var myValue = ""
    private let l = UILabel()
    private let i = UIImageView()

    convenience init(str: String, img : UIImage) {
        self.init()
        myValue = str
        i.image = img
        if UIDevice.current.userInterfaceIdiom == .phone {
            i.bounds.size = CGSize(width: 250, height: 250)
        } else {
            i.bounds.size = CGSize(width: 500, height: 500)
        }
    }
}
```


DetailViewController

```
override func viewDidLoad() {
    self.view = UIView(frame: UIScreen.main.bounds)
    self.view.backgroundColor = UIColor.white
    l.text = "Detail view for "+myValue
    l.textAlignment = .center
    self.view.addSubview(l)
    self.view.addSubview(i)
    self.navigationItem.title = "Detail (" + myValue + ")"
    self.displayInSize(UIScreen.main.bounds.size)
}

func displayInSize(_ s : CGSize) {
    if s.width > s.height &&
        UIDevice.current.userInterfaceIdiom == .phone {
        l.frame = CGRect(x: 10, y: s.height / 4, width: s.width / 2 - 20, height: 40)
        i.frame = CGRect(x: s.width / 4 * 3 - i.frame.size.width / 2,
            y: s.height / 2 - i.frame.size.height / 2,
            width: i.frame.size.width,
            height: i.frame.size.height)
    } else {
        l.frame = CGRect(x: 10, y: 100, width: s.width - 20, height: 40)
        i.frame = CGRect(x: s.width / 2 - i.frame.size.width / 2,
            y: s.height / 2 - i.frame.size.height / 2,
            width: i.frame.size.width,
            height: i.frame.size.height)
    }
}
```

DetailViewController

```
override func viewWillTransition(to size: CGSize,  
    with coordinator: UINavigationControllerTransitionCoordinator) {  
    super.viewWillTransition(to: size, with: coordinator)  
    self.displayInSize(size)  
}  
}
```

ViewController

```
import UIKit

class ViewController: UITableViewController { // Inherits from UITableViewController

    private var count = 1
    private var content = [[OneCell]]()

    override init(style: UITableView.Style) {
        super.init(style:style)
        self.tableView.separatorColor = UIColor.clear
        for _ in 1...3 { // _ because index unused
            var inSection = [OneCell]()
            for _ in 1...10 { // _ because index unused
                inSection += [OneCell(l: "Cell #\(count)",
                                     d: "detail #\(count)")]
                count += 1
            }
            content += [inSection]
        }
        self.tableView.dataSource = self
        self.tableView.delegate = self
        self.title = "My elements"
        // Buttons for update... could be done in viewDidLoad()
        self.clearsSelectionOnViewWillAppear = false
        self.navigationItem.leftBarButtonItem = self.editButtonItem
        self.navigationItem.rightBarButtonItem =
            UIBarButtonItem(barButtonItemSystemItem: .add,
                           target: self, action: #selector(addCell))
    }
}
```

ViewController

```
required init?(coder aDecoder: NSCoder) {
    // There is nothing else to do there in fact
    super.init(coder: aDecoder)
}

@objc func addCell () {
    self.content[0].insert(OneCell(l:"New cell \ (count)",
                                   d:"new detail \ (count)"), at: 0)

    count += 1
    self.tableView.reloadData()
}

// UITableViewDataSource protocol

override func numberOfSections(in tableView: UITableView) -> Int {
    return content.count
}

override func tableView(_ tableView: UITableView,
                        numberOfRowsInSection section: Int) -> Int {
    return content[section].count
}
```

ViewController

```
override func tableView(_ tableView: UITableView,
                        cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    var cellId = ""
    if (indexPath as NSIndexPath).row % 3 == 1 {
        cellId = "one"
    } else if (indexPath as NSIndexPath).row % 3 == 2 {
        cellId = "two"
    } else {
        cellId = "three"
    }
    var cell = tableView.dequeueReusableCell(withIdentifier: cellId)
    if cell === nil {
        var img : UIImage?
        cell = UITableViewCell(style: .subtitle, reuseIdentifier: cellId)
        if (indexPath as NSIndexPath).row % 3 == 1 {
            cell!.textLabel?.textColor = UIColor.red
            img = UIImage(named: "alfa-romeo")
        } else if (indexPath as NSIndexPath).row % 3 == 2 {
            cell!.textLabel?.textColor = UIColor.orange
            img = UIImage(named: "mazda")
        } else {
            cell!.textLabel?.textColor = UIColor.blue
            img = UIImage(named: "facel-vega")
        }
        cell!.imageView?.image = img!
    }
    let cont = content[indexPath.section][indexPath.row]
    cell!.textLabel?.text = cont.label
    cell!.detailTextLabel?.text = cont.detail
    cell!.backgroundView = UIImageView(image: UIImage(named: «bg-tbv-cell»))
    return cell!
}
```

ViewController

```
// Override to support editing the table view.
override func tableView(_ tableView: UITableView,
                        commit editingStyle: UITableViewCell.EditingStyle,
                        forRowAt indexPath: IndexPath) {
    if editingStyle == .delete {
        self.content[indexPath.section].remove(at: indexPath.row)
        tableView.deleteRows(at: [indexPath], with: .fade)
    } else if editingStyle == .insert {
        // You could implement it too (but we did it differently)
    }
    self.tableView.reloadData() // because several types of cells
}

// Override to support conditional rearranging of the table view.
override func tableView(_ tableView: UITableView,
                        canMoveRowAt indexPath: IndexPath) -> Bool {
    return indexPath.row % 3 > 0 // some cell may be unmovable
}

// Override to support rearranging the table view.
override func tableView(_ tableView: UITableView,
                        moveRowAt sourceIndexPath: IndexPath,
                        to destinationIndexPath: IndexPath) {
    let val = content[sourceIndexPath.section][sourceIndexPath.row]
    self.content[sourceIndexPath.section].remove(at: sourceIndexPath.row)
    self.content[destinationIndexPath.section].insert(val,
                                                       at: destinationIndexPath.row)
    self.tableView.reloadData() // because several types of cells
}
```

ViewController

```
override func tableView(_ tableView: UITableView,
                        didSelectRowAt indexPath: IndexPath) {
    // Create the view controller to insert
    let s = self.content[indexPath.section][indexPath.row].label
    var img : UIImage?
    if (indexPath as NSIndexPath).row % 3 == 1 {
        img = UIImage(named: "alfa-romeo")
    } else if (indexPath as NSIndexPath).row % 3 == 2 {
        img = UIImage(named: "mazda")
    } else {
        img = UIImage(named: "facel-vega")
    }

    let detailVC = DetailViewController(str : s,
                                       img : img!)
    self.navigationController?.pushViewController(detailVC, animated: true)
}
```

ViewController

```
// UITableViewDelegate protocol

override func tableView(_ tableView: UITableView,
                        heightForHeaderInSection section: Int) -> CGFloat {
    return 80.0
}

override func tableView(_ tableView: UITableView,
                        viewForHeaderInSection section: Int) -> UIView? {
    let hv = UIView(frame: CGRect(x: 0.0, y: 0.0,
                                   width: UIScreen.main.bounds.size.width,
                                   height: 80.0))
    hv.addSubview(UIImageView(image: UIImage(named: "bg-header")))
    let l = UILabel(frame: CGRect(x: 20.0, y: 30.0,
                                   width: UIScreen.main.bounds.size.width - 30.0,
                                   height: 25.0))

    l.textColor = .yellow
    l.font = UIFont.boldSystemFont(ofSize: 20.0)
    l.text = "Section \(section + 1)"
    hv.addSubview(l)
    return hv
}
}
```


Remark about transitions...

10



You have a large variety of transitions

Objective-C

- ▶ UITableViewRowAnimationFade, UITableViewRowAnimationRight, UITableViewRowAnimationLeft, UITableViewRowAnimationTop, UITableViewRowAnimationBottom, UITableViewRowAnimationNone, UITableViewRowAnimationMiddle, UITableViewRowAnimationAutomatic

Swift

- ▶ fade, right, left, top, bottom, none, middle, automatic

A dedicated code for this

- ▶ Setting up elements in the embedded UIView

```
let transition = CATransition()
transition.duration = 1.0
transition.type = .fade
self.navigationController?.view.layer.add(transition, forKey: nil)
self.navigationController?.pushViewController(detailVC, animated: false)
```

As a conclusion...

You know almost everything...

- Handle structured data
- Present them in a standard and user friendly way
- Stack hierarchies of views

You MUST use this

- Standard
- Known by your users

