

Mobile Applications Development 2 (MAD2)

Diploma in IT

Teaching Team:
Dr Joel Yang
Mr Charles Keck

MAD2 Oct 2018

Chapter 4

iOS User Interface II



Objectives

To be able to understand:

- Storyboard Segue
- UITableView
 - Delegate
 - DataSource



Storyboard

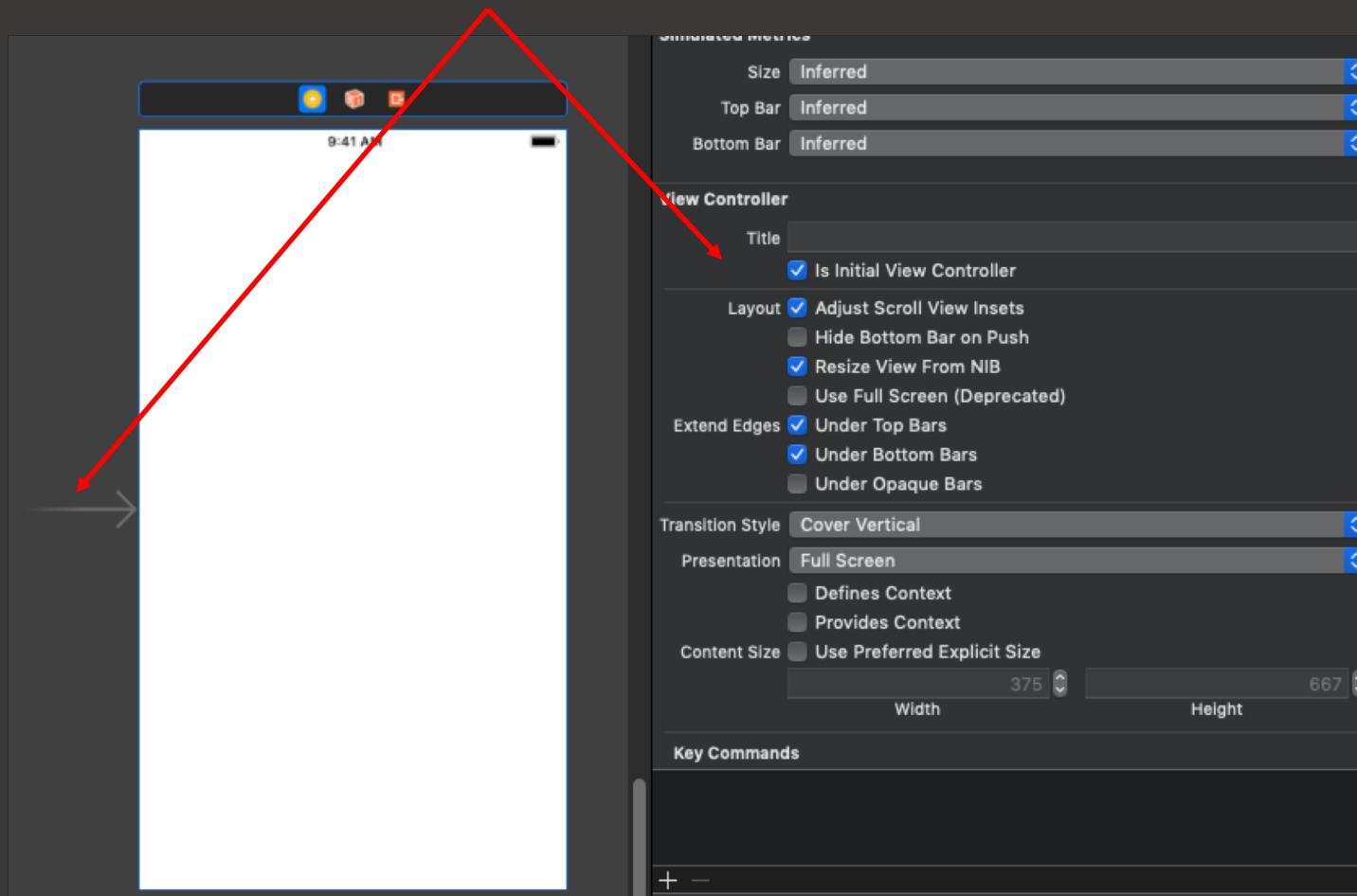
Storyboard

- You can **visually layout** all your view controllers in “**scenes**” and describe the connections between them. With a storyboard you’ve a better conceptual overview of all the scenes in your app.
- Describe the **transitions** between the various scenes. These transitions are called “**segues**” and you create them by connecting your view controllers in the storyboard. Thanks to segues you need less code to take care of your UI.

Storyboard

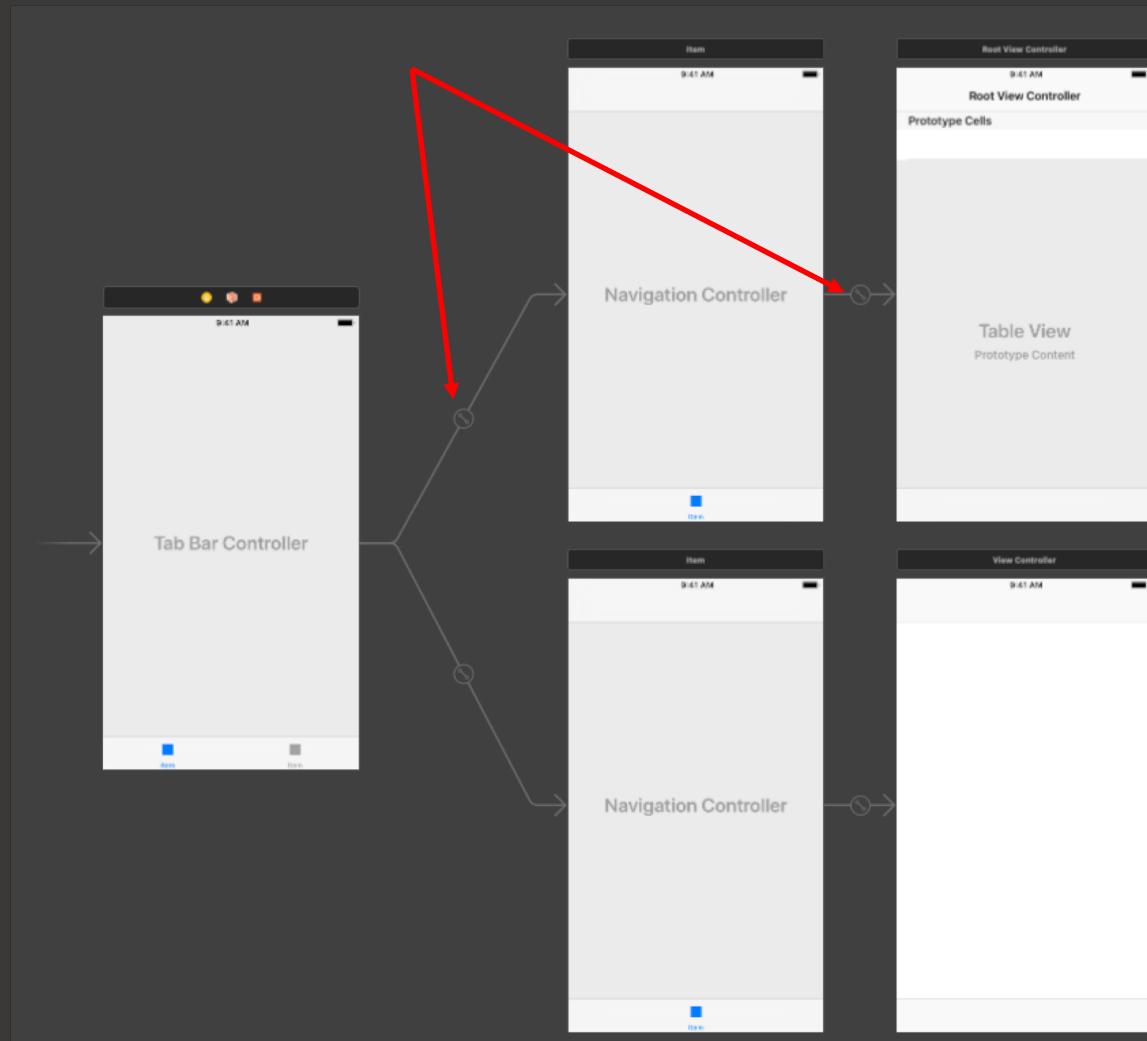
- Make working with table views a lot easier with **prototype** and **static cells** features. You can design your table views almost completely in the storyboard editor, cutting down the amount of code you have to write.
- Make it easier to use **Auto Layout**, a feature that allows you to define **mathematical relationships** between elements defining their position and sizing.

Initial View Controller



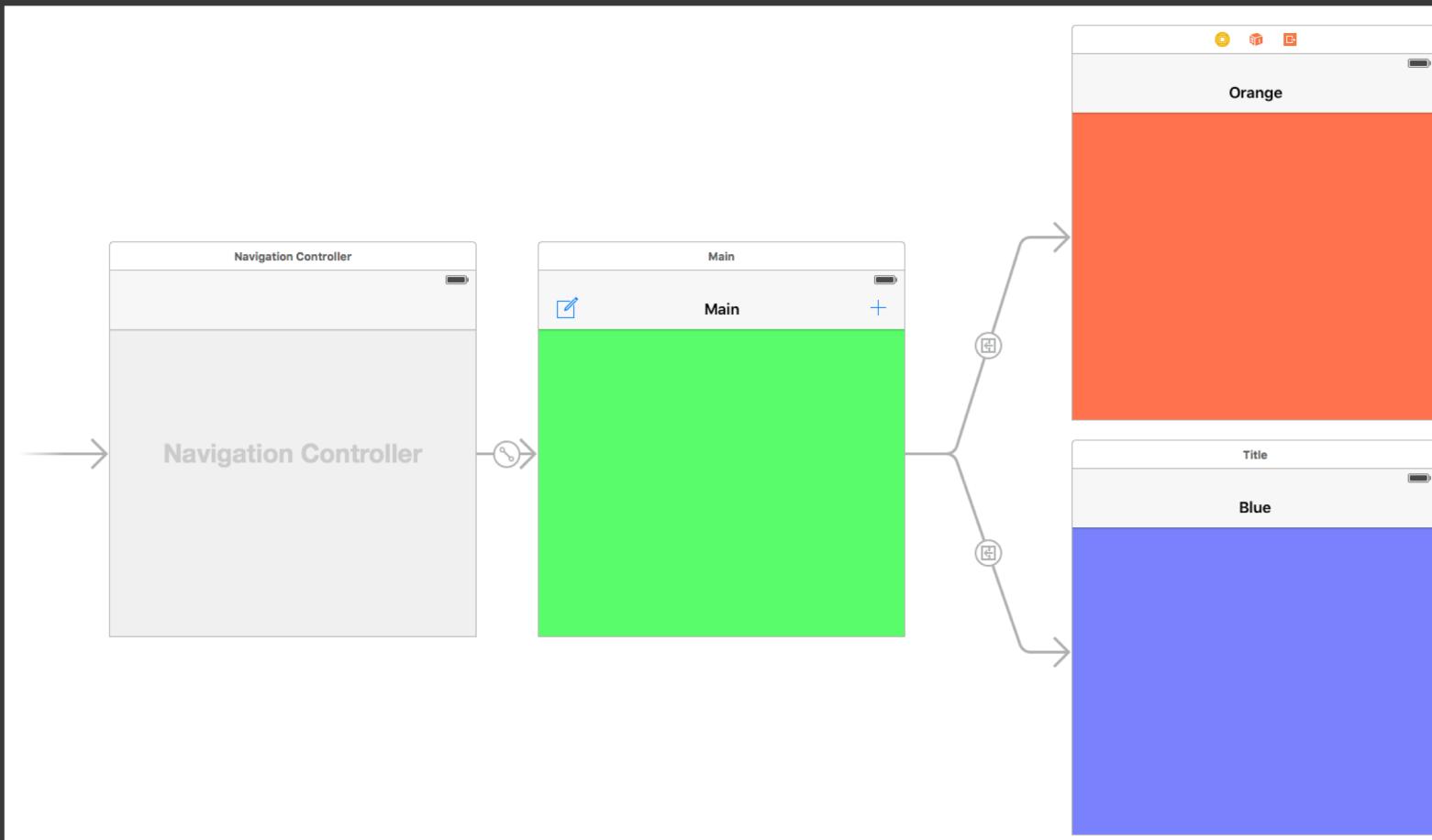
MAD2 Oct 2018

Storyboard Sequ



MAD2 Oct 2018

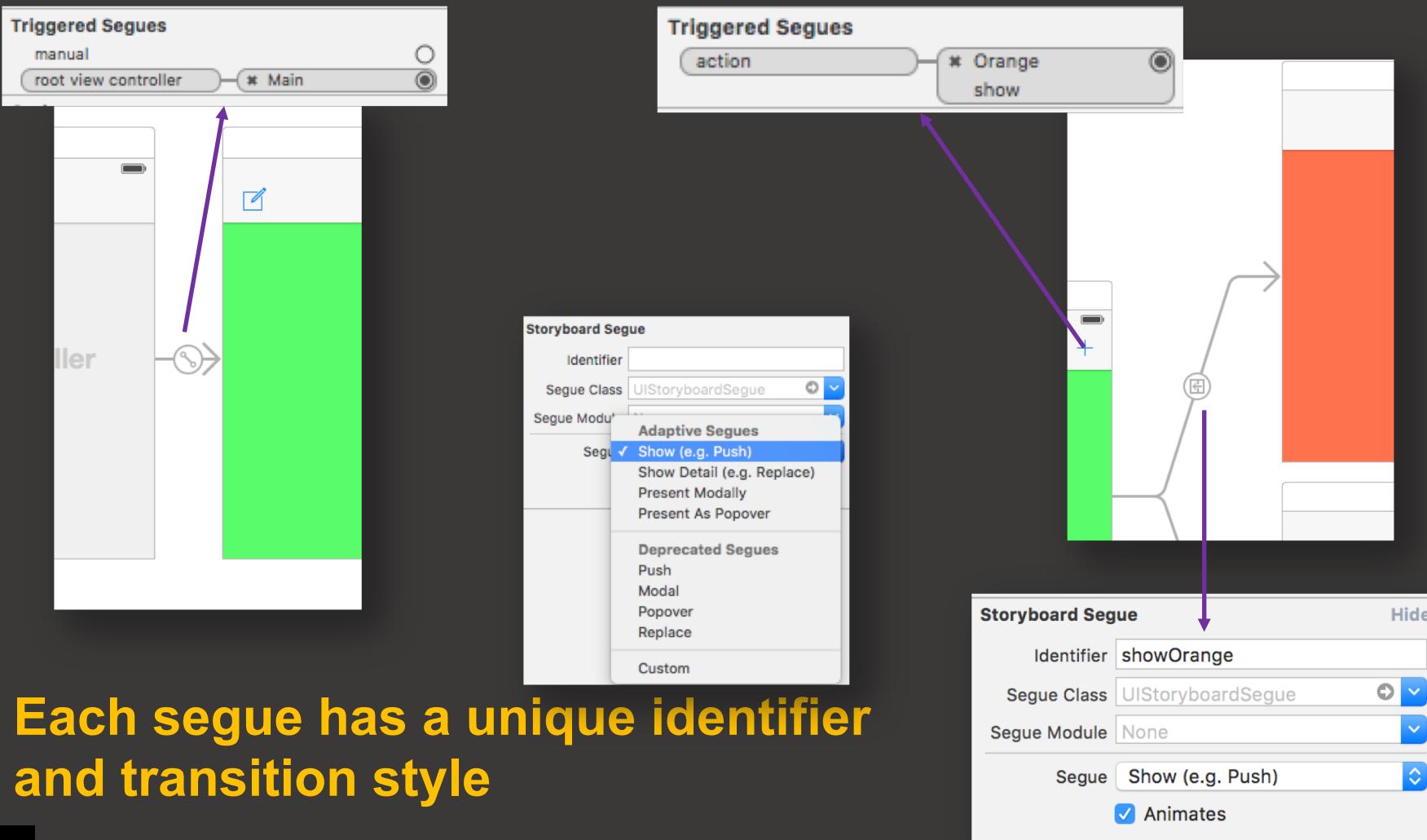
Storyboard Sequencing



Ctrl-click a control and drag to a view controller

MAD2 Oct 2018

Storyboard Seque



Each segue has a unique identifier and transition style

Storyboard Seque

- To pass data between controllers

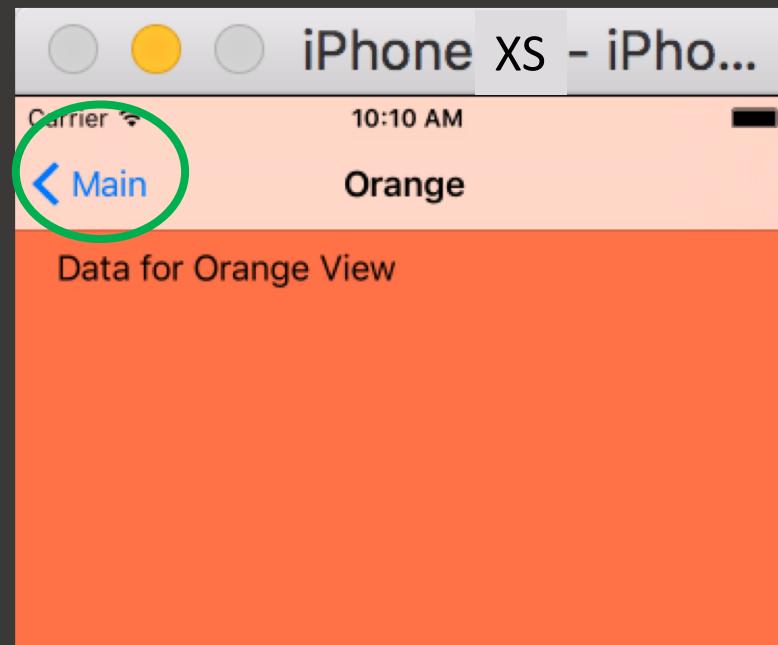
```
override func prepare(for segue: UIStoryboardSegue, sender: Any!) {  
    if (segue.identifier == "showOrange") {  
        (segue.destination as! OrangeViewController).dataOrange  
            = txtValueforOrange.text  
    } else {  
        if (segue.identifier == "showBlue") {  
            (segue.destination as! BlueViewController).dataBlue  
                = txtValueforBlue.text  
        }  
    }  
}
```

var dataOrange:String?

- Implement the above method in the originating controller

Storyboard Seque

- Button to navigate back

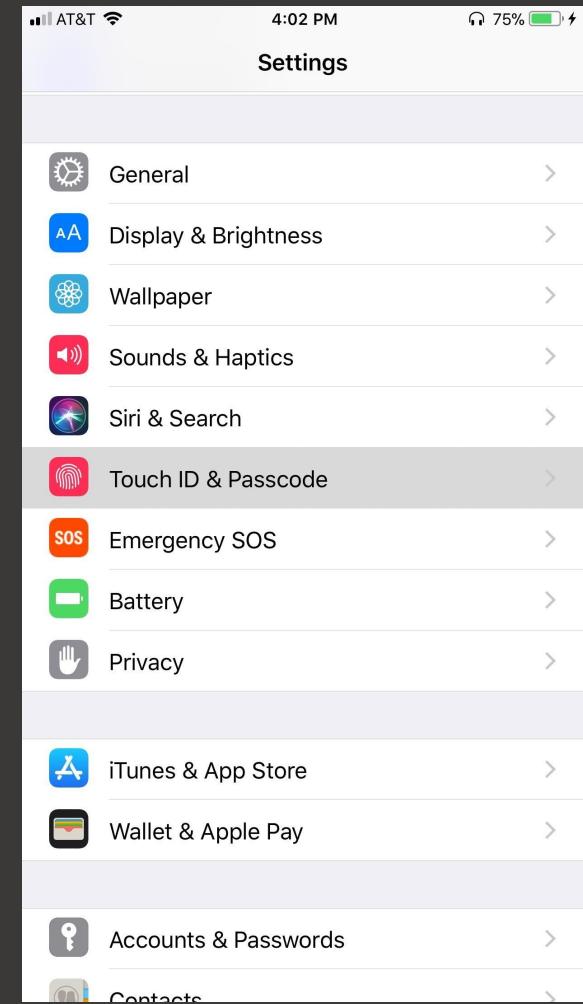




UITableView

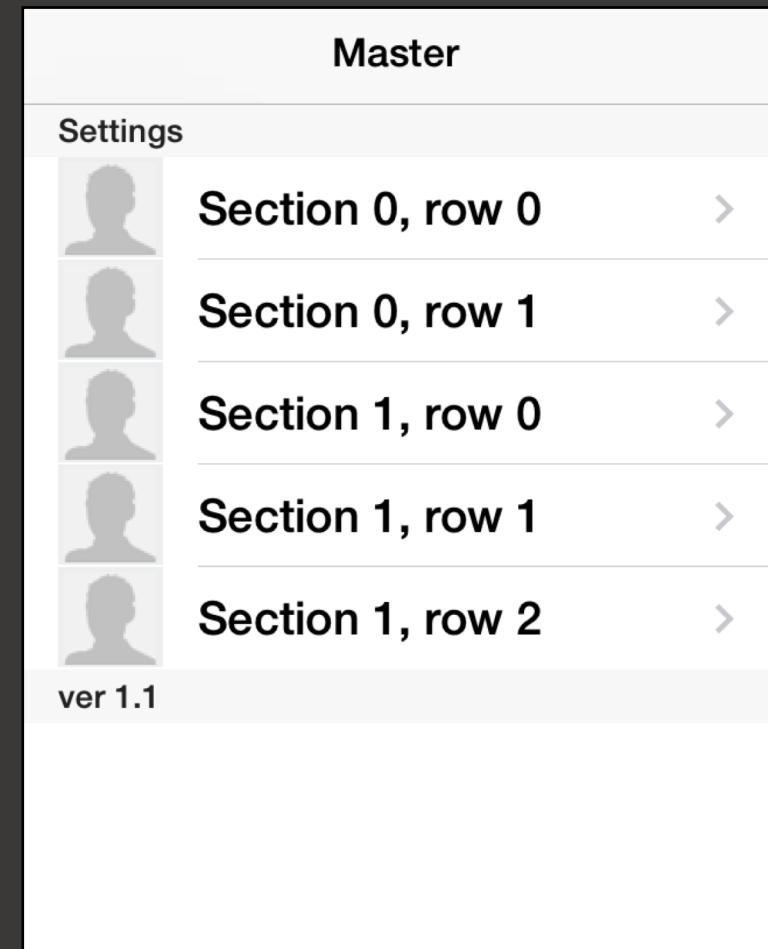
UITableView

- A scrollable list of items that may be divided into sections
- User can navigate hierarchically structured data
- Display an indexed list of items
- To display information in visually distinct groupings
- Present a list of options



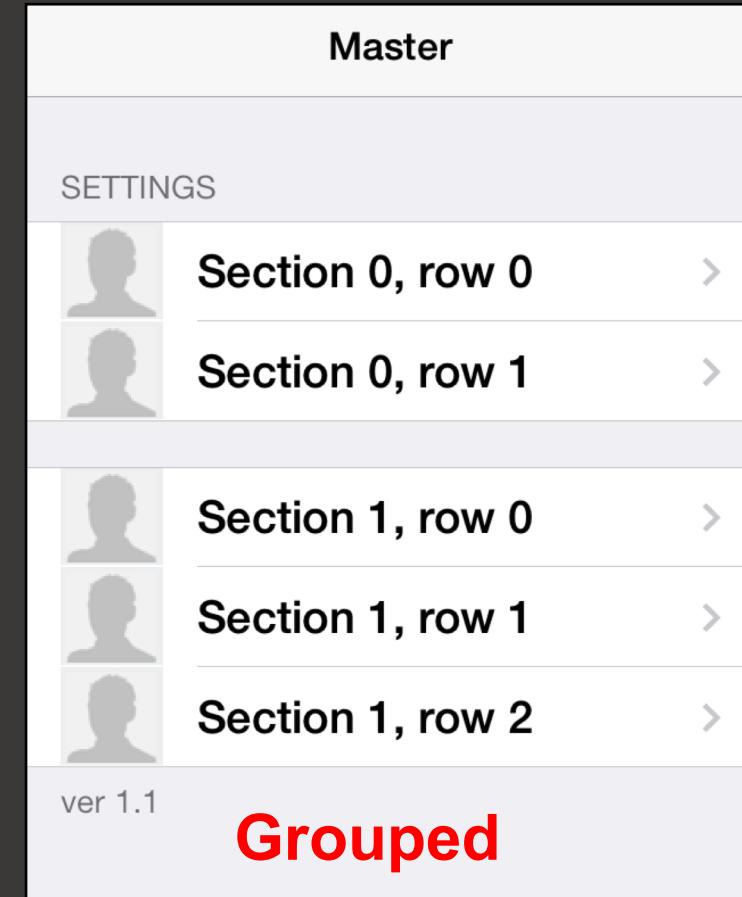
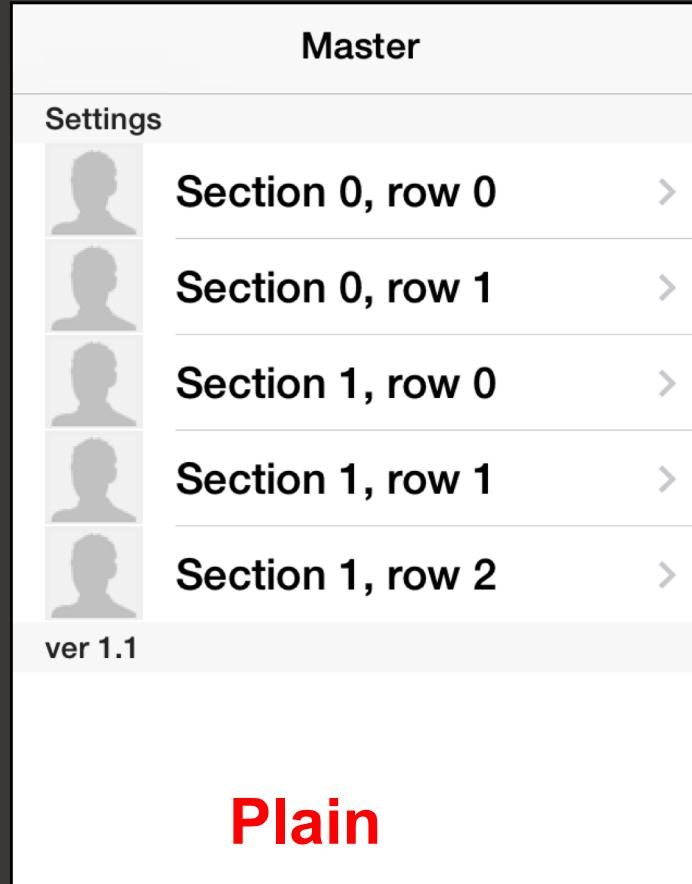
UITableView

- Key concepts
 - Sections
 - Header
 - Footer
 - Rows in each section
 - Cell in each row
 - Static vs dynamic content
 - Default vs custom layout
 - UITableViewDelegate to handle events
 - UITableViewDataSource protocol to provide data



UITableView

- Different Styles



UITableView

- Specify number of section

```
override func numberOfSections(in tableView: UITableView) -> Int {  
    return 2  
}
```

- Specify number of rows

```
var sectionOneitems:[String] = ["One", "Two", "Three"]  
var sectionTwoitems:[String] = ["Ten", "Twenty"]  
override func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {  
    switch(section) {  
        case 0: return sectionOneitems.count  
        case 1: return sectionTwoitems.count  
        default: return 0  
    }  
}
```

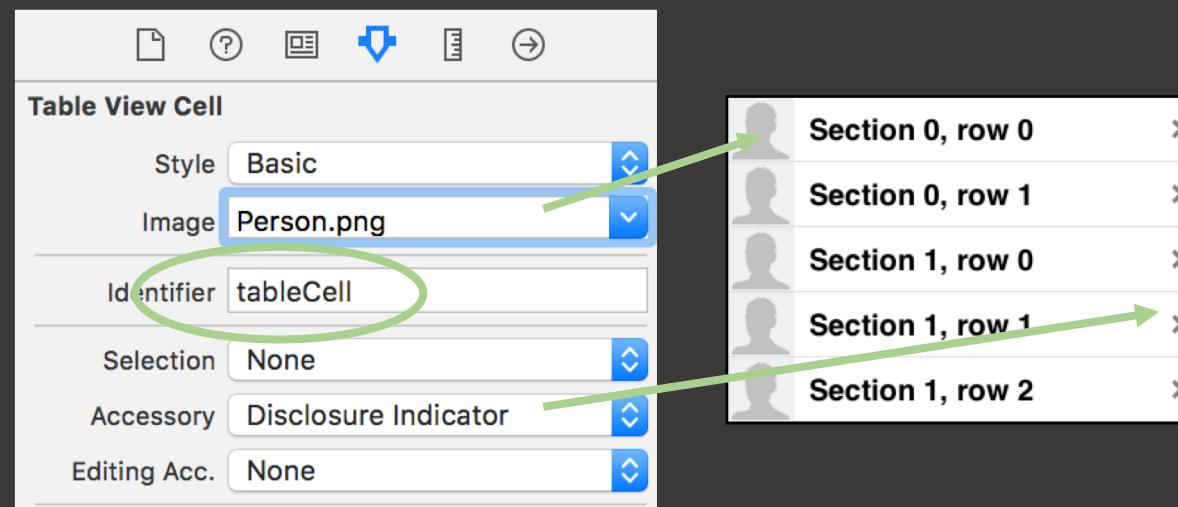
UITableView DataSource Protocol

- Populate each row from with value from DataSource e.g. database, array

```
override func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) -> UITableViewCell {
    let cell = self.tableView.dequeueReusableCell(withIdentifier: "tableCell",
                                                for: indexPath)
    switch ((indexPath as NSIndexPath).section) {
        case 0: cell.textLabel?.text = self.sectionOneitems[(indexPath as
                                                               NSIndexPath).row]
        case 1: cell.textLabel?.text = self.sectionTwoitems[(indexPath as
                                                               NSIndexPath).row]
        default: cell
    }
    return cell
}
```

UITableView DataSource Protocol

- Populate each row from with value from DataSource e.g. database, array



UITableView

- Title in Headers and Footers

```
override func tableView(_ tableView: UITableView, titleForHeaderInSection section: Int) -> String? {
    switch(section) {
        case 0: return String(sections[0])
        case 1: return String(sections[1])
        default: return ""
    }
}

override func tableView(_ tableView: UITableView, titleForFooterInSection section: Int)-> String? {
    return ""
}
```

UITableView

- Row Selected

```
// when a table row is selected, the following delegate is called

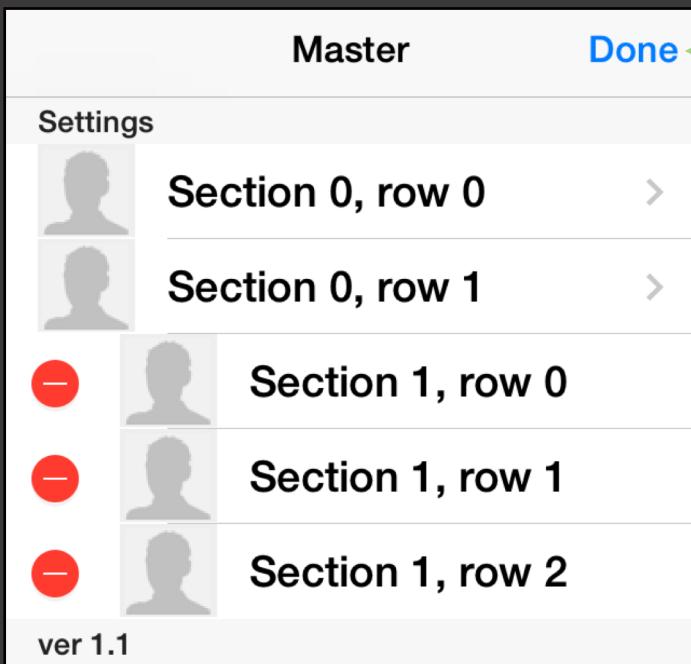
override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath)
{
    let alert = UIAlertController(title: "Item selected",
        message: "You selected section\(indexPath.section) : item \(indexPath.row)", preferredStyle: UIAlertControllerStyle.alert)

    alert.addAction(UIAlertAction(title: "OK", style: UIAlertActionStyle.default,
        handler: { (alert: UIAlertAction!) in print("An alert was tapped!")}))

    self.present(alert, animated: true, completion: nil)
}
```

UITableView

- Editing

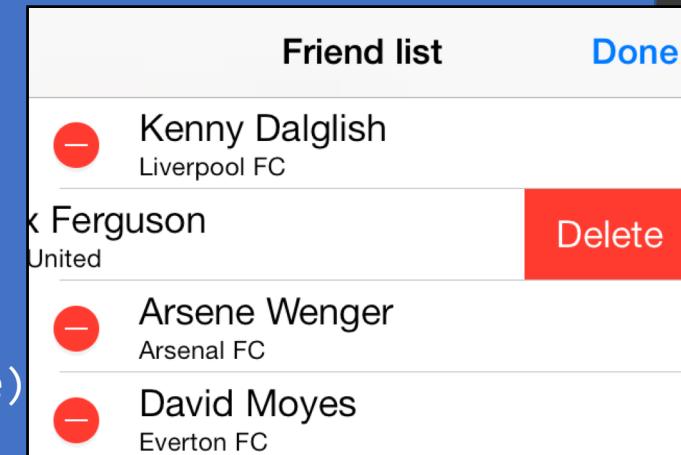


```
override func viewDidLoad() {  
    super.viewDidLoad()  
    self.navigationItem.rightBarButtonItem = self.editButtonItem  
}
```

```
override func tableView(_ tableView: UITableView, canEditRowAt indexPath: IndexPath) -> Bool {  
    if indexPath.section == 0 {  
        return false  
    }  
    else {  
        return true  
    }  
}
```

UITableView

```
override func tableView(_ tableView: UITableView,  
                      commit editingStyle: UITableViewCellEditingStyle,  
                      forRowAt indexPath: IndexPath) {  
  
    if editingStyle == UITableViewCellEditingStyle.delete {  
        if indexPath.section == 0 {  
            sectionOneitems.remove(at: indexPath.row)  
        }  
        else {  
            sectionTwoitems.remove(at: indexPath.row)  
            tableView.deleteRows(at: [indexPath as IndexPath],  
                                 with: UITableViewRowAnimation.fade)  
        }  
    } else if editingStyle == UITableViewCellEditingStyle.insert {  
        // Create a new instance of the appropriate class, insert it into the array,  
        // and add a new row to the table view.  
    }  
}
```



Summary

Understand Storyboard Segue and UITableView