Tables

Mobile Application Development in iOS

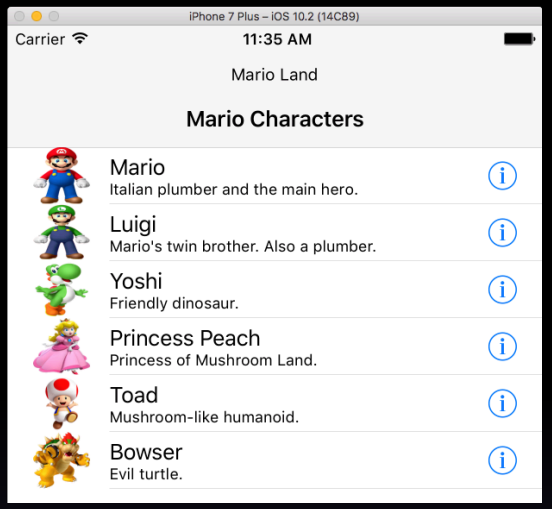
School of EECS

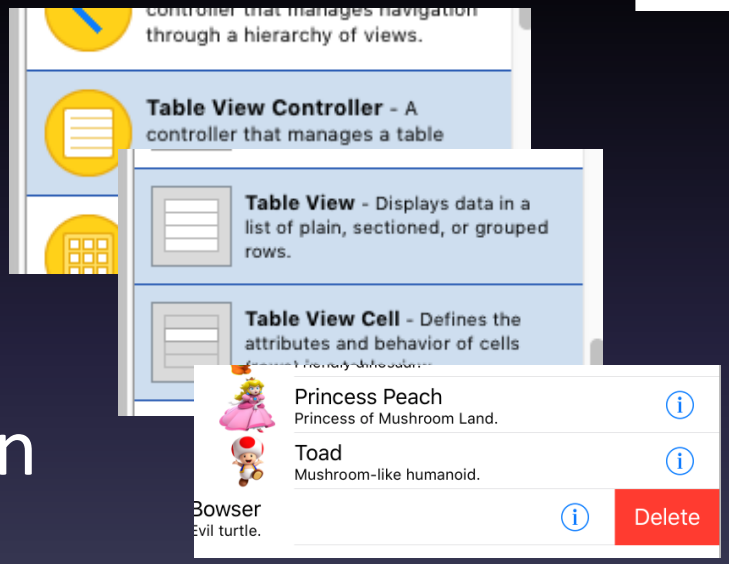
Washington State University

Instructor: Larry Holder

Outline

- Table View Controller
- Table View
- Table Cells
- Cell interaction
- Navigation





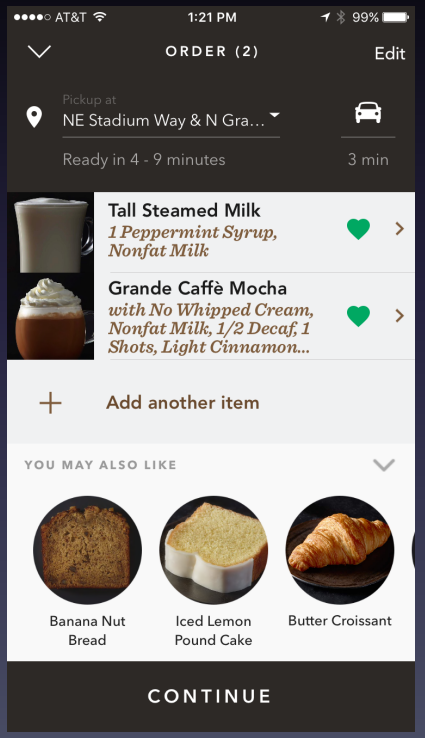


Table View Controller

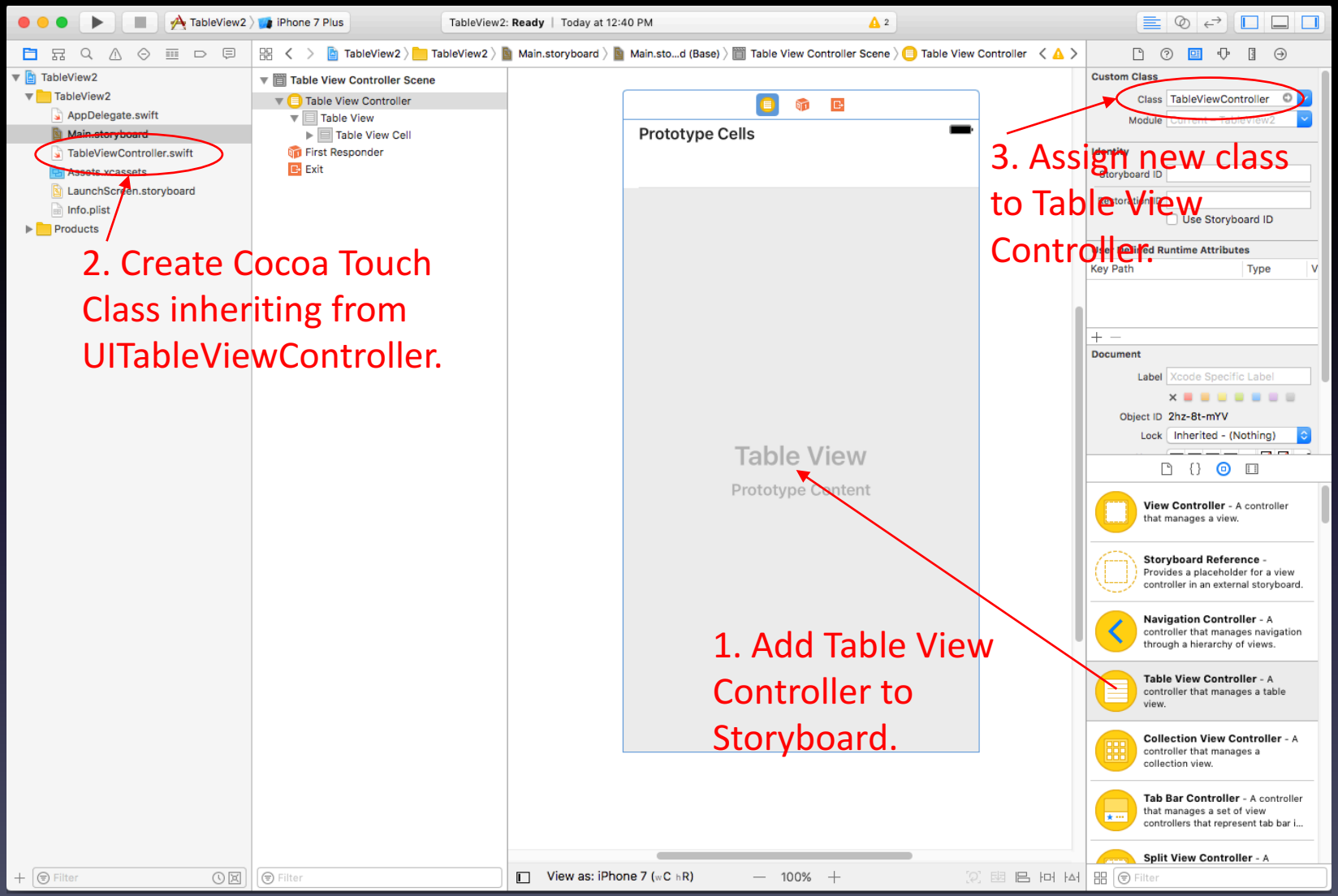
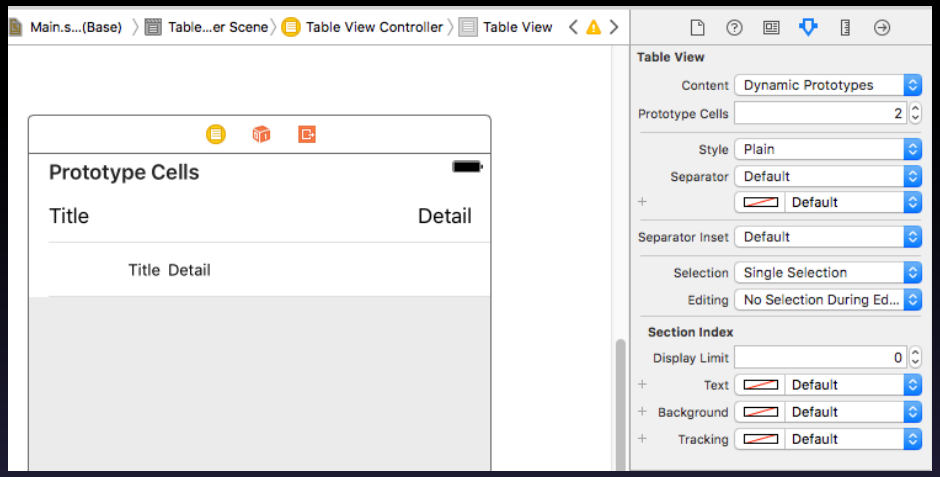


Table View Attributes

- TableView Content
  - Dynamic
    - One section
    - Multiple cell prototypes
    - Variable number of cells
  - Static
    - Multiple sections
    - One cell prototype per section
    - Fixed number of cells
  - Sections Plain or Grouped



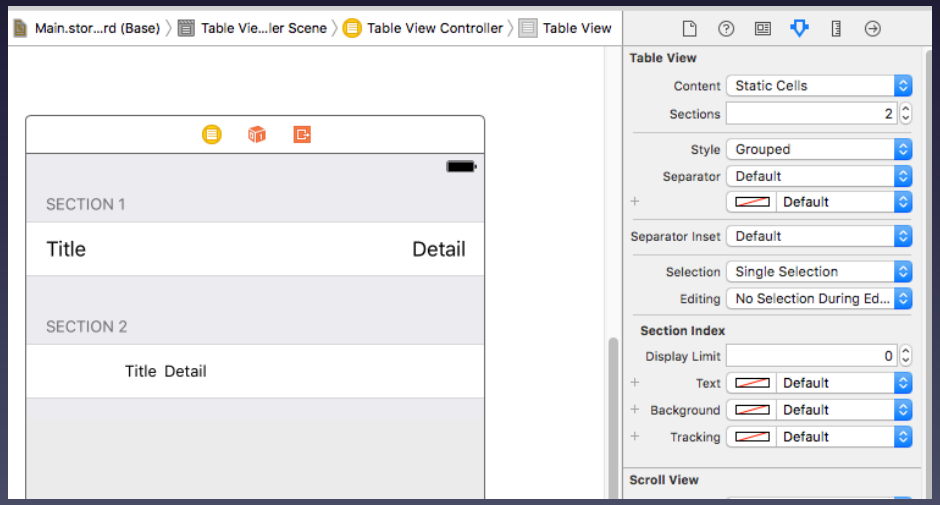
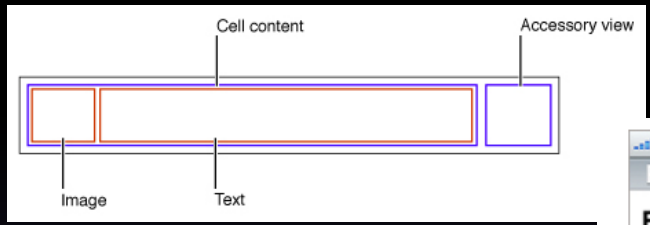
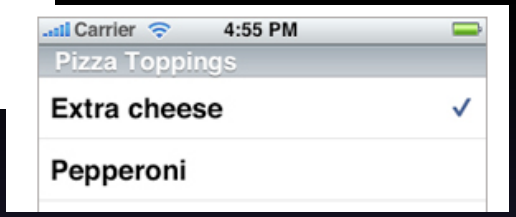
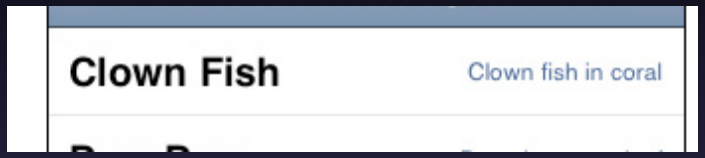


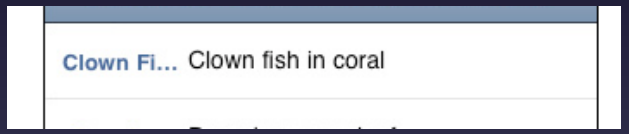
Table Cell Styles

- Table cell styles
  - Basic
  - Right detail
  - Left detail
  - Subtitle
  - Custom



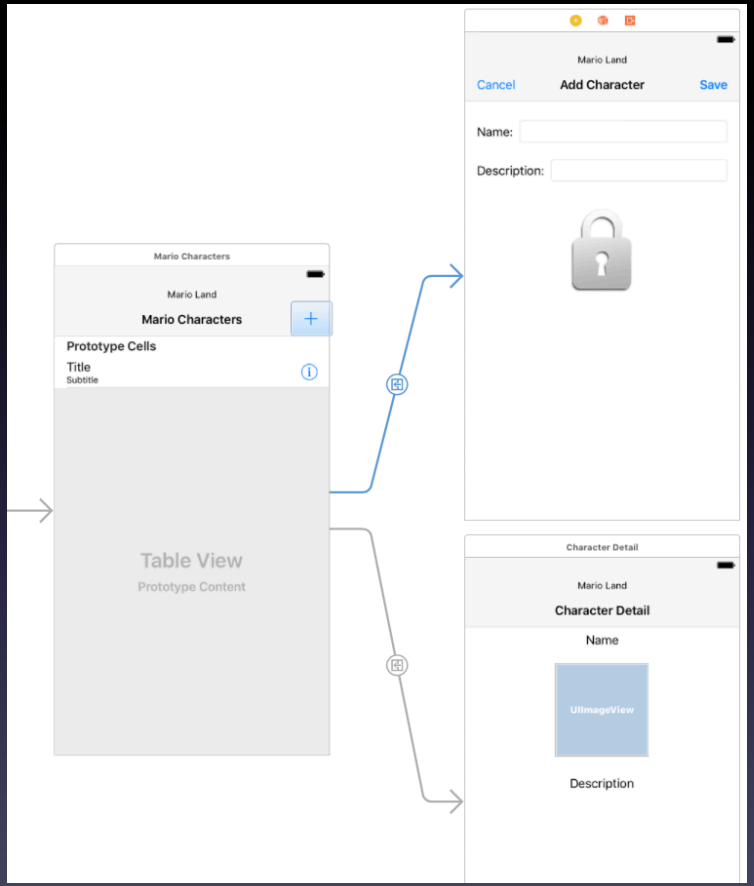






Navigation

- Create views for Details and Add new entries
- Create segue to Detail View
  - Perform when row/accessory selected
- Create Add bar button
- Create segue from Add button to Add View



Cell Interaction: Selection

- Row Selection

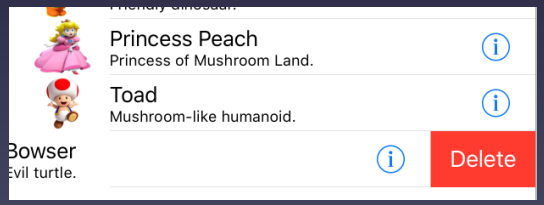
```
override func tableView(_ tableView: UITableView, didSelectRowAt indexPath: IndexPath) {
  self.selectedRow = indexPath.row
  performSegue(withIdentifier: "toDetail", sender: nil)
}
```

- Accessory Selection

```
override func tableView(_ tableView: UITableView,
                        accessoryButtonTappedForRowWith indexPath: IndexPath) {
  self.selectedRow = indexPath.row
  performSegue(withIdentifier: "toDetail", sender: nil)
}
```

Cell Interaction: Deletion

```
// Override to support conditional editing of the table view.
override func tableView(_ tableView: UITableView, canEditRowAt indexPath: IndexPath) ->
Bool {
  // Return false if you do not want the specified item to be editable.
  return true
}

// Override to support editing the table view.
override func tableView(_ tableView: UITableView, commit editingStyle:
UITableViewCellEditingStyle, forRowAt indexPath: IndexPath) {
  if editingStyle == .delete {
    // Delete the row from the data source
    marioCharacters.remove(at: indexPath.row)
    tableView.deleteRows(at: [indexPath], with: .fade)
  }
}
```



Insertion

- In AddViewController.swift
  - Maintain Bool indicating new entry ready
  - Save button sets Bool=true and performs unwind segue

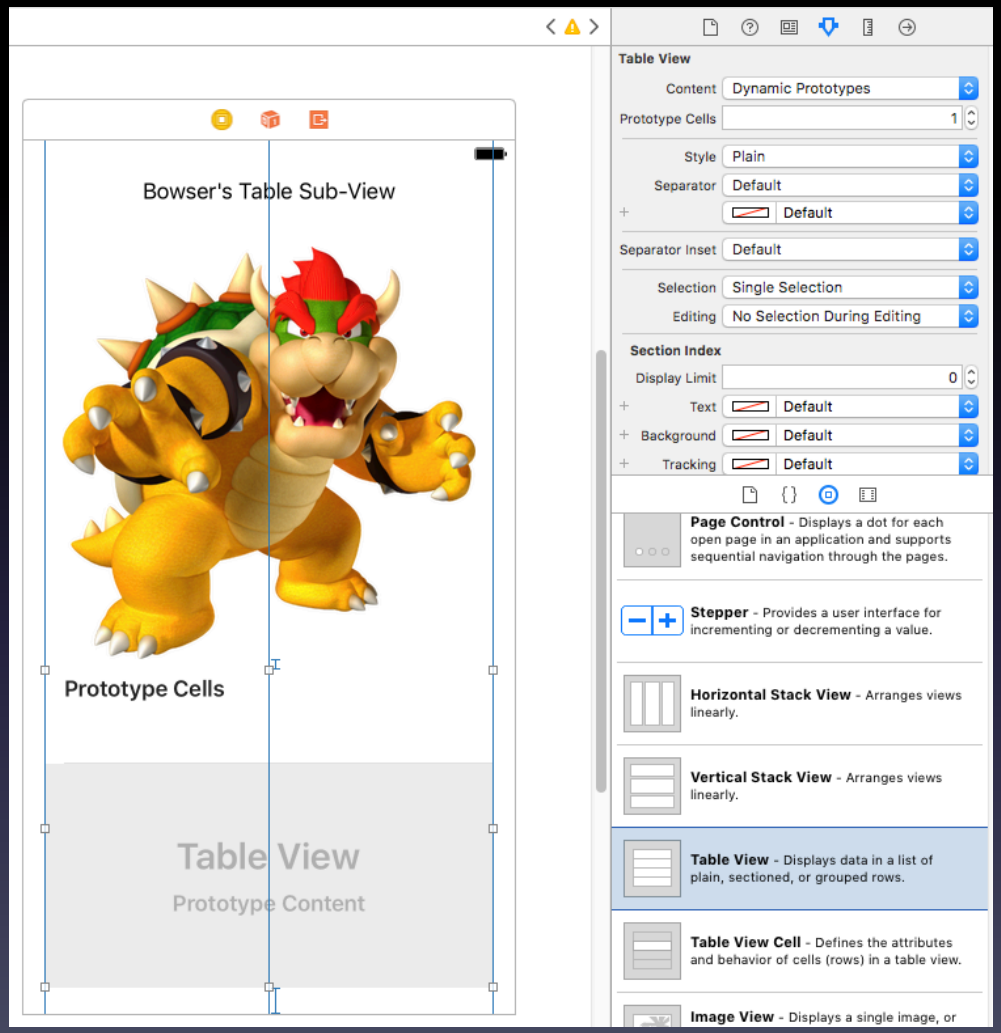
```
class AddViewController: UIViewController, UITextFieldDelegate {

  var newCharacterReady: Bool = false

  @IBOutlet weak var nameTextField: UITextField!
  @IBOutlet weak var descriptionTextField: UITextField!

  @IBAction func saveButton(_ sender: UIBarButtonItem) {
    newCharacterReady = true
    performSegue(withIdentifier: "unwindFromDetail", sender: nil)
  }
// ...
```

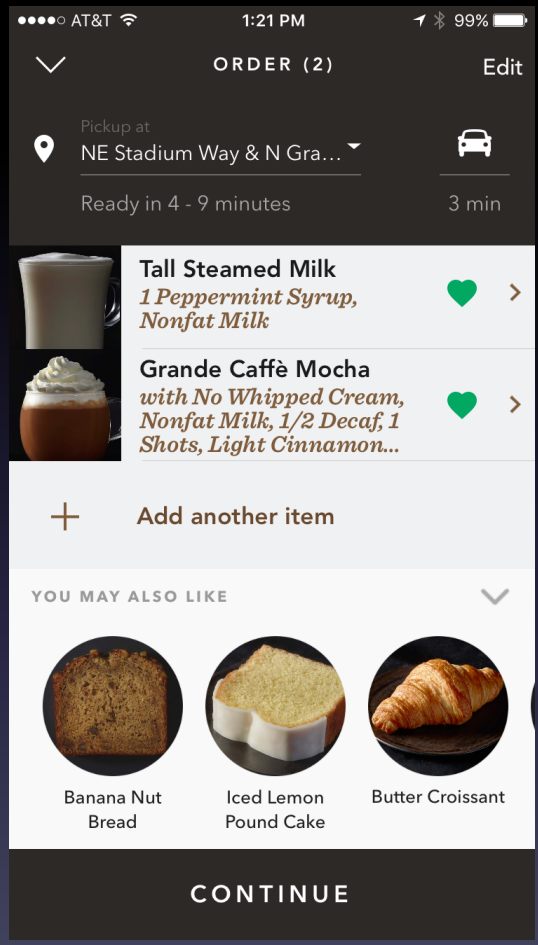
Insertion (cont.)

- In TableViewController.swift
  - In unwind segue
    - Check if new entry ready
    - If so, create new data instance, add to array, and reload data

```
@IBAction func unwindFromDetail (segue: UIStoryboardSegue) {
  let addVC = segue.source as! AddViewController
  if (addVC.newCharacterReady) {
    let name = addVC.nameTextField.text!
    let description = addVC.descriptionTextField.text!
    let newCharacter = MarioCharacter(name, description, "locked-128.png")
    marioCharacters.append(newCharacter)
    self.tableView.reloadData()
  }
}
```

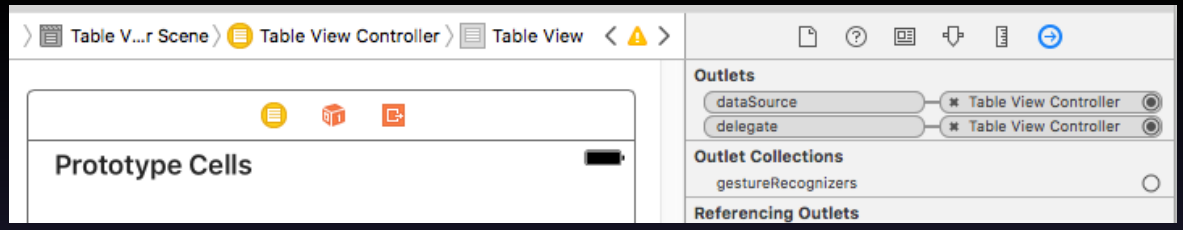
Adding Table View to Existing View





Delegate and Data Source

- Automatic for Table View Controller



- But can setup programmatically for Table sub-View

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

    @IBOutlet weak var bowserTableView: UITableView!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.
        bowserTableView.delegate = self
        bowserTableView.dataSource = self
    }
    // ...
}
```

Delegate and DataSource for Table View

```
class ViewController: UIViewController, UITableViewDelegate, UITableViewDataSource {

  // ...

  func numberOfSections(in tableView: UITableView) -> Int {
    return 1
  }

  func tableView(_ tableView: UITableView, numberOfRowsInSection section: Int) -> Int {
    return 3
  }

  func tableView(_ tableView: UITableView, cellForRowAt indexPath: IndexPath) ->
UITableViewCell {
    let cell = tableView.dequeueReusableCell(withIdentifier: "bowserCell", for: indexPath)

    // Configure the cell...
    cell.textLabel?.text = "Bowser"

    return cell
  }
```

Resources

- Start Developing iOS Apps (good Tables tutorial)
  - https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/
- UITableViewController (documentation)
  - https://developer.apple.com/reference/uikit/uitableviewcontroller

Extras

- Saving data to a file
- Renaming an Xcode project

Saving Data to File

- Class objects to be written must inherit from NSObject and NSCoding
- Class must implement
  - `required init(coder aDecoder: NSCoder)`
  - `func encode(with aCoder: NSCoder)`

Saving Data to File

```
class MarioCharacter: NSObject, NSCoding {
    var name: String
    var health: Int

    init (_ name: String, _ health: Int) {
        self.name = name
        self.health = health
    }

    required init(coder aDecoder: NSCoder) {
        name = aDecoder.decodeObject(forKey: "name") as! String
        health = aDecoder.decodeInteger(forKey: "health")
    }

    func encode(with aCoder: NSCoder) {
        aCoder.encode(name, forKey: "name")
        aCoder.encode(health, forKey: "health")
    }
}
```

Saving Data to File

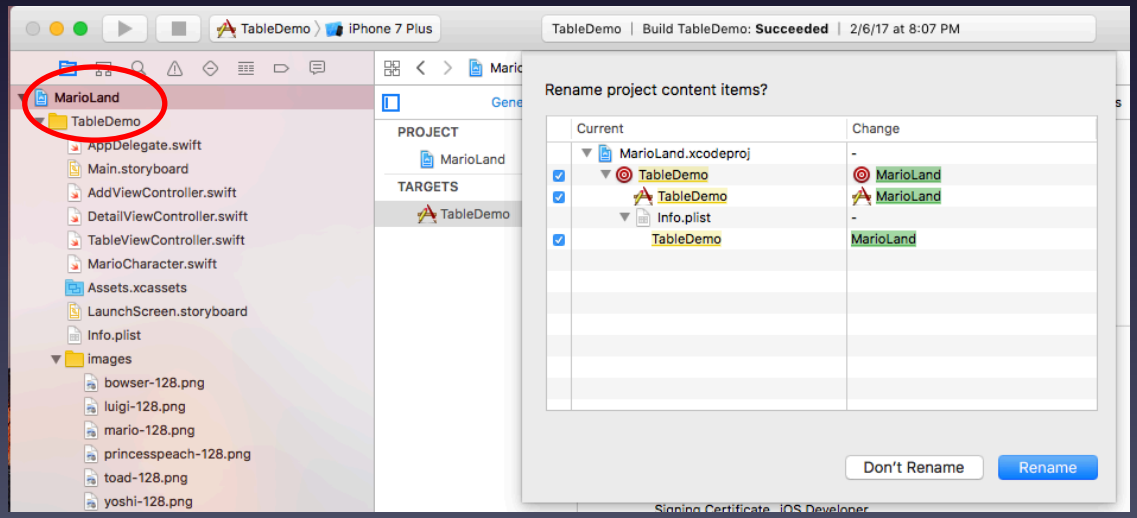
- Get document directory
- Create URL to file
- Use NSKeyedArchiver to write
- Use NSKeyedUnarchiver to read

Saving Data to File

```
let marioCharactersFile = "MarioCharactersFile"

func readFromFile () { // call from initial view controller's viewDidLoad
  let fileDir = FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first!
  let fileURL = fileDir.appendingPathComponent(marioCharactersFile)
  if FileManager.default.fileExists(atPath: fileURL.path) {
    marioCharacters = NSKeyedUnarchiver.unarchiveObject(withFile: fileURL.path)
        as! [MarioCharacter]
  }
}

func writeToFile () { // call whenever marioCharacters array changed
  let fileDir = FileManager.default.urls(for: .documentDirectory, in: .userDomainMask).first!
  let fileURL = fileDir.appendingPathComponent(marioCharactersFile)
  NSKeyedArchiver.archiveRootObject(marioCharacters, toFile: fileURL.path)
}
```

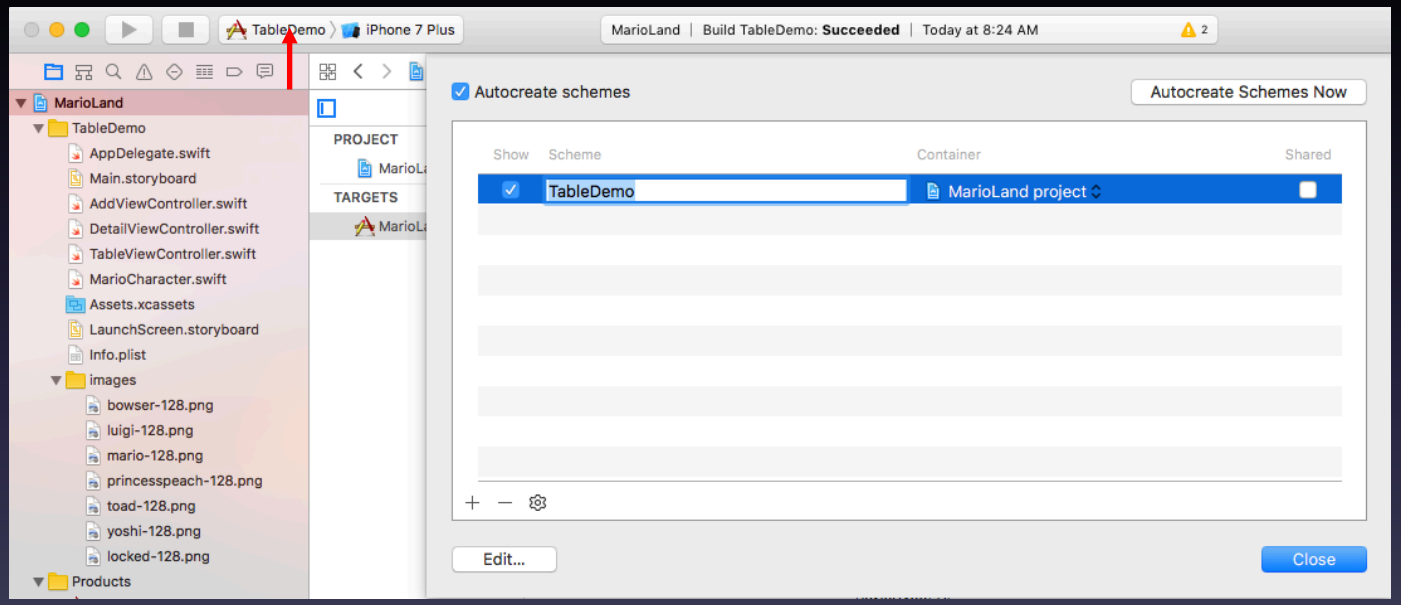
Renaming Xcode Project

- Step 1: Change project name in upper left of file hierarchy
  - Rename project content items



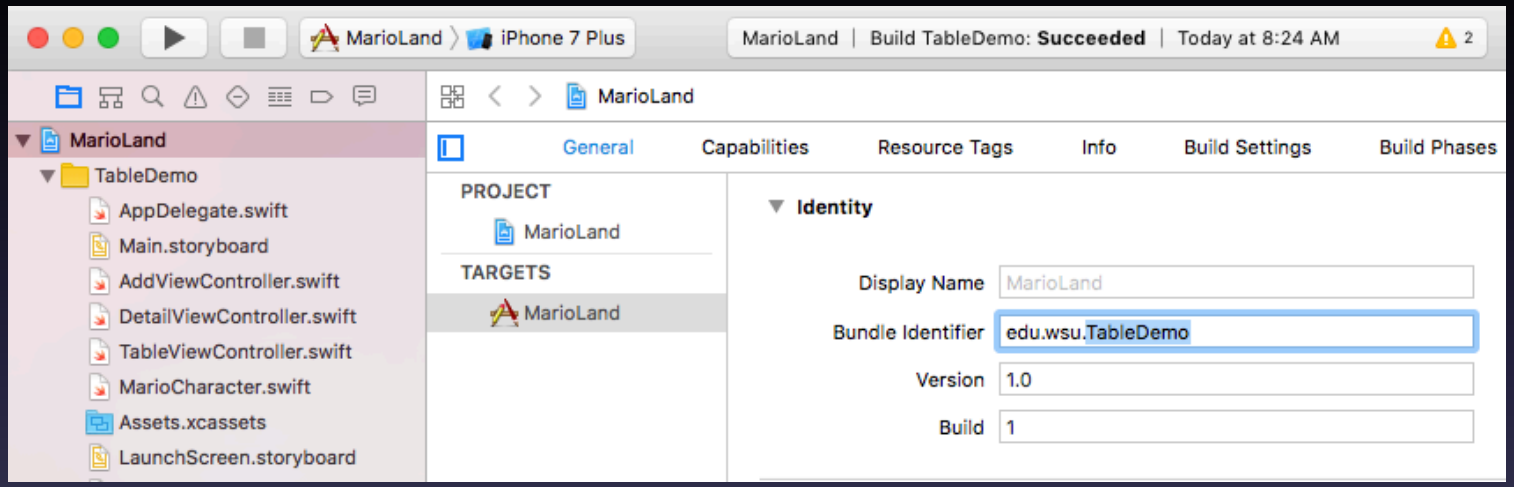
Renaming Xcode Project

- Step 2: Rename scheme



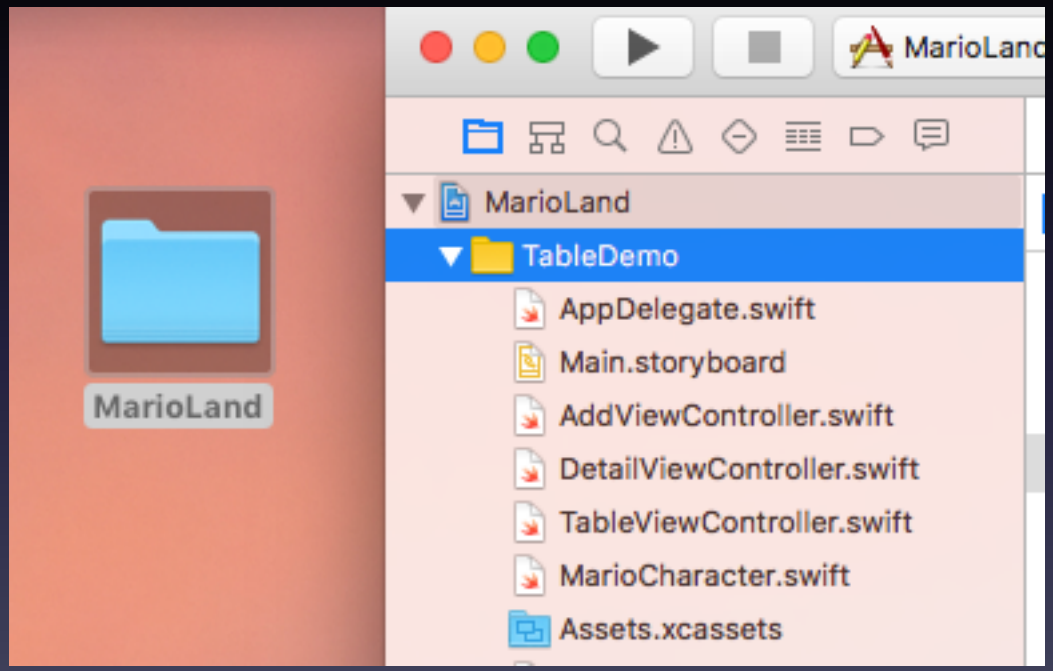
Renaming Xcode Project

- Step 2: Change bundle identifier



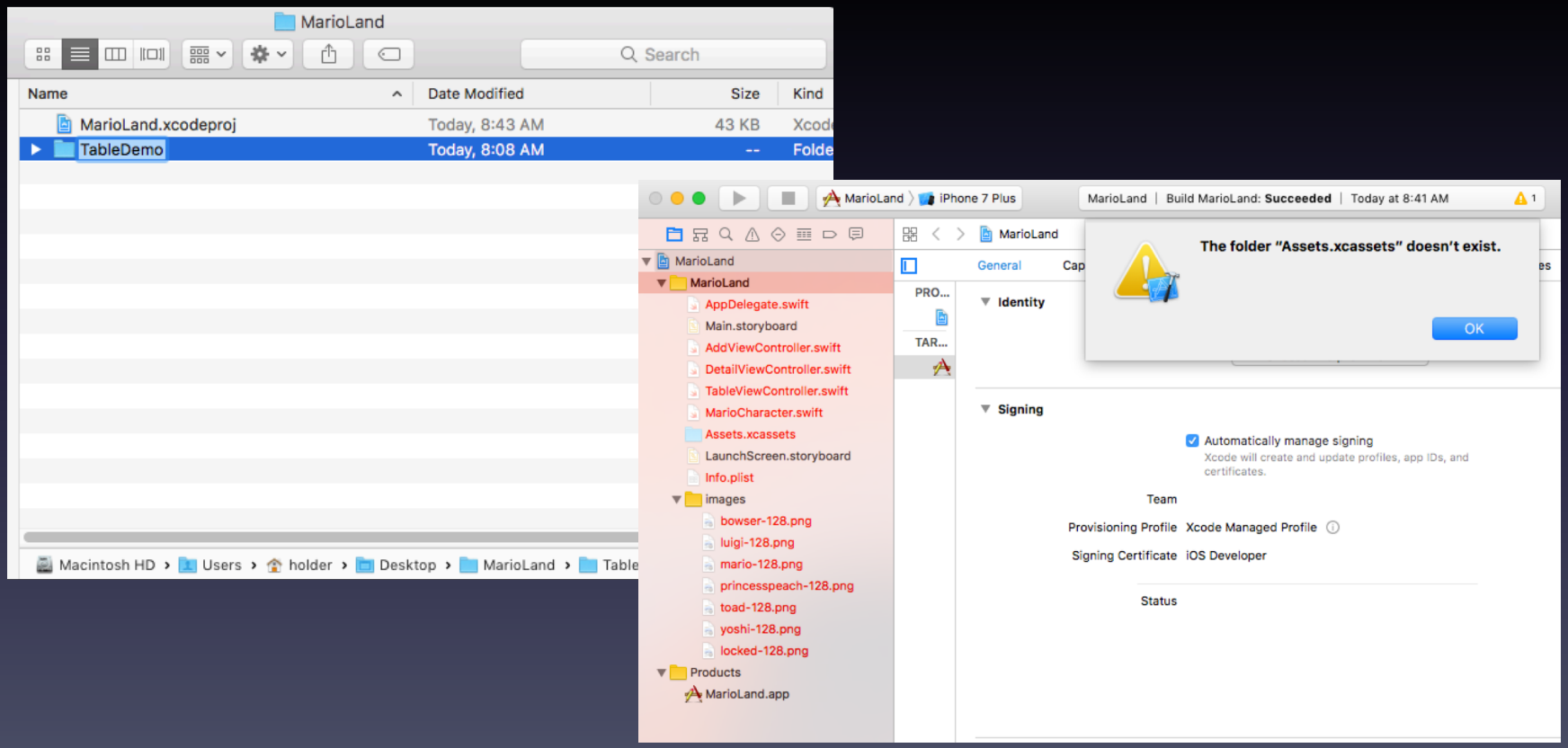
Renaming Xcode Project

- Step 3: Change top-level folder/file names



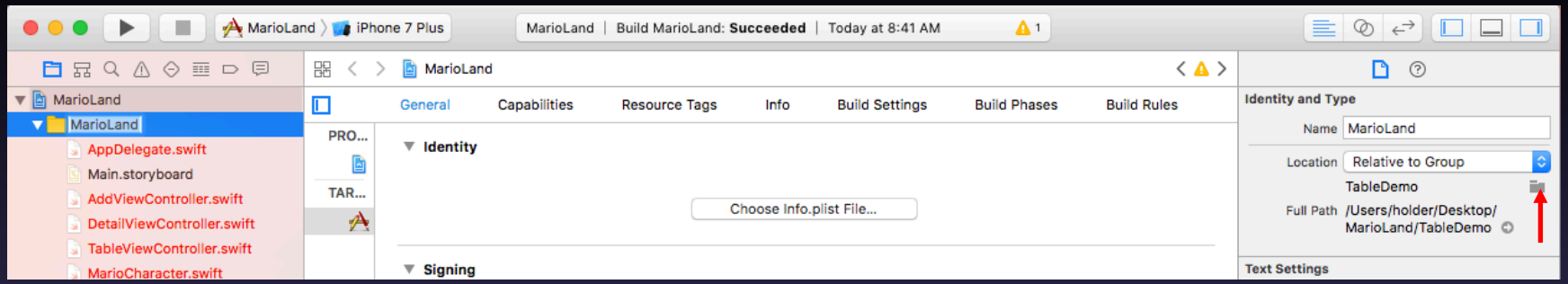
Renaming Xcode Project

- Step 4: Rename source code folder (optional)



Renaming Xcode Project

- Step 4b: Change source code path



Renaming Xcode Project

- Step 4c: Change any build settings