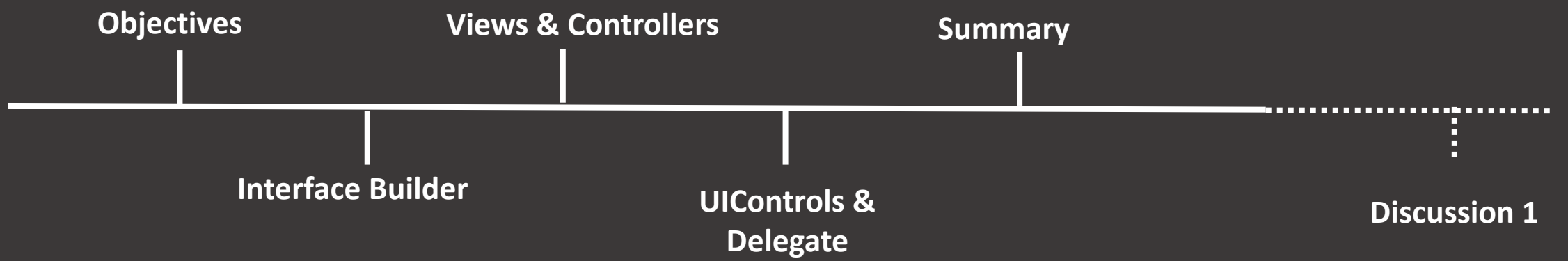
Mobile Applications Development 2 (MAD2)

Diploma in IT

Teaching Team:
Dr Joel Yang
Mr Charles Keck

NGEE ANN
SCHOOL OF INFOCOMM TECHNOLOGY

Chapter 3
iOS User Interface I



Objectives

To be able to understand:

- Interface builder
- Views and controllers
- iOS UI Controls
  - UIButton
  - UITextField
- Delegate

Interface Builder

MAD2 Oct 2018

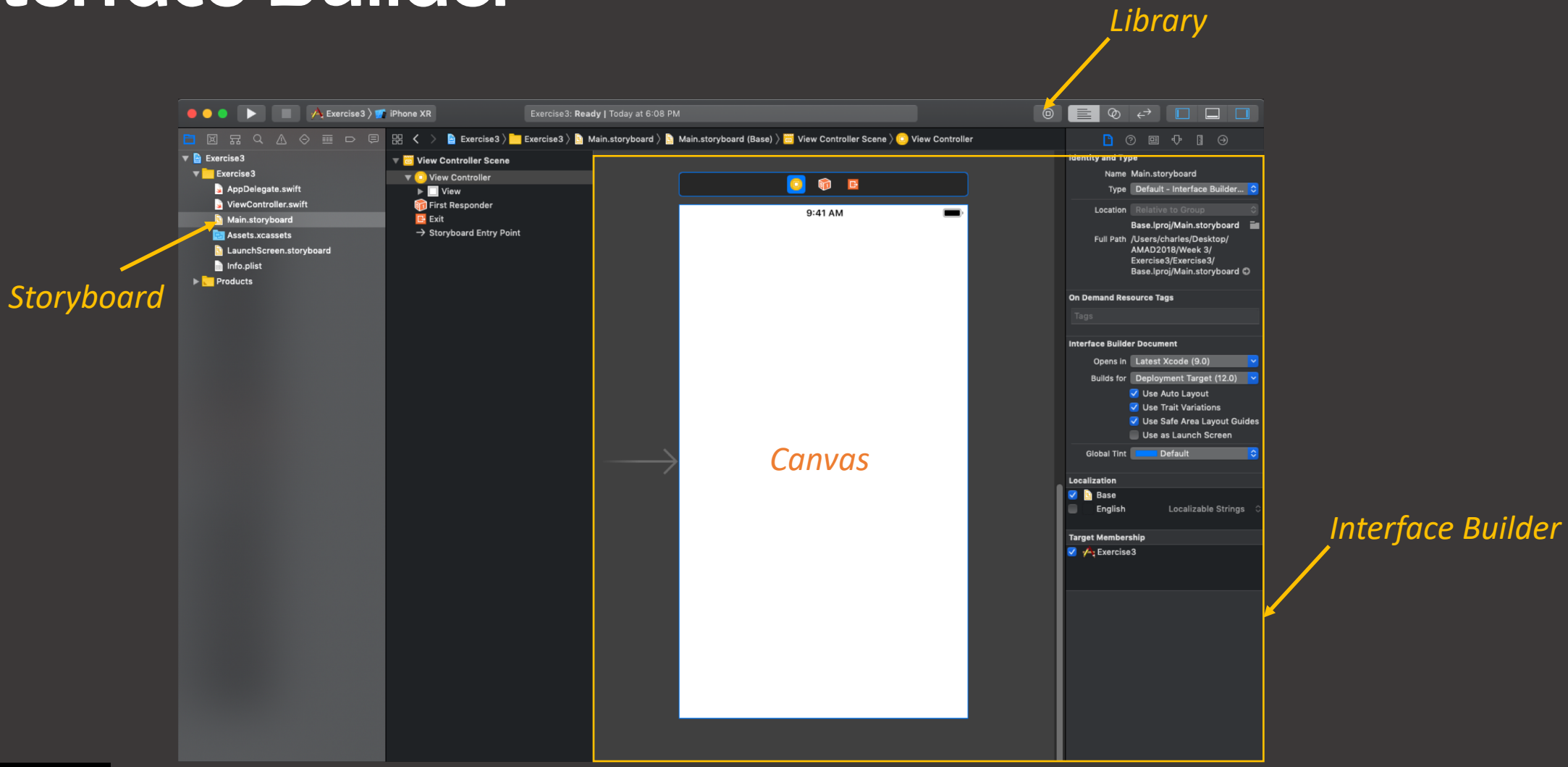
NGEE ANN POLYTECHNIC
SCHOOL OF INFOCOMM TECHNOLOGY

Interface Builder

- A storyboard is a visual representation of the app's user interface, showing screens of content and the transitions between them. You use storyboards to lay out the flow—or story—that drives your app. You see exactly what you're building while you're building it, get immediate feedback about what's working and what's not, and make instantly visible changes to your user interface.

- Xcode opens the storyboard in Interface Builder—its visual interface editor—in the editor area.

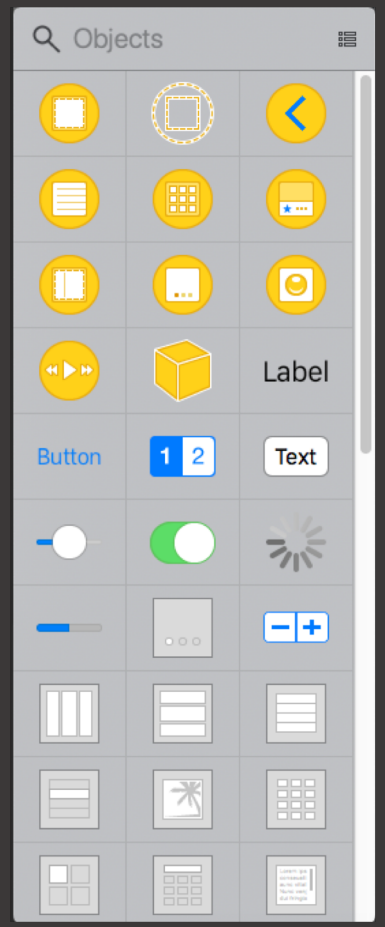
Interface Builder

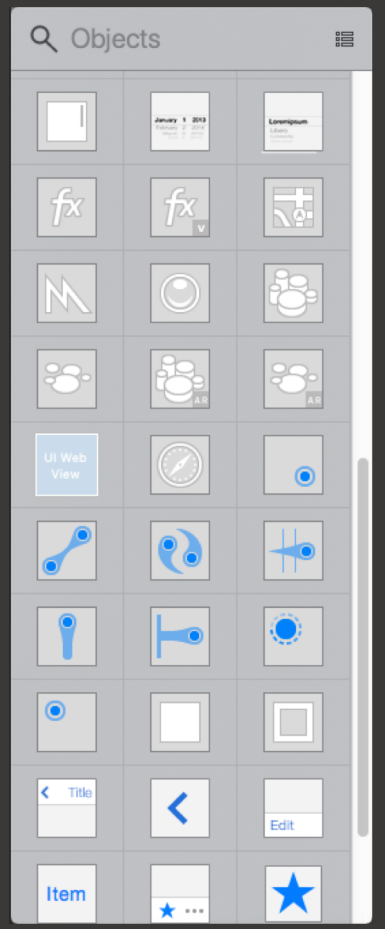


Interface Builder

- Xcode provides a library of objects that you can add to a storyboard file.
- Some of these are elements that appear in the user interface, such as buttons and text fields. Others, such as view controllers and gesture recognizers, define the behavior of your app but don't appear onscreen.

Library


Objects
Label
Button
1 2
Text

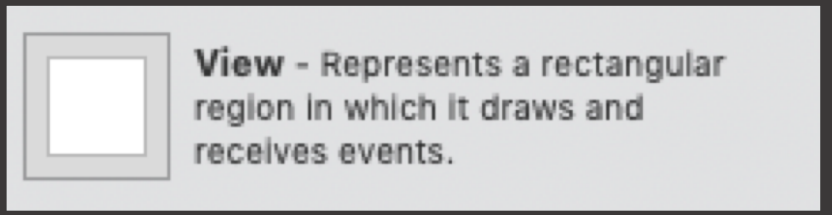

Objects
UI Web View
Title
Edit
Item

Views and Controllers

MAD2 Oct 2018

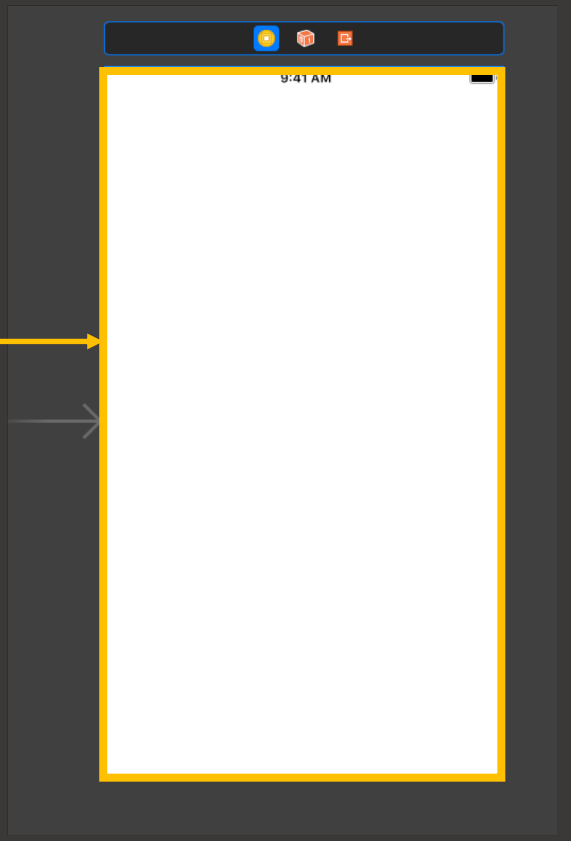
View

- The elements that appear in the user interface are known as views.
- Views display content to the user. They are the building blocks for constructing your user interface and presenting your content in a clear, elegant, and useful way. Views have a variety of useful built-in behaviors, including displaying themselves onscreen and reacting to user input.



View

- The rectangular area on the screen
  - User interacts with Views
  - Background
  - UI Control
  - Table
  - Textfield
  - Image



- Everything you see and do visually is a View
  - An instance of UIView class in iOS

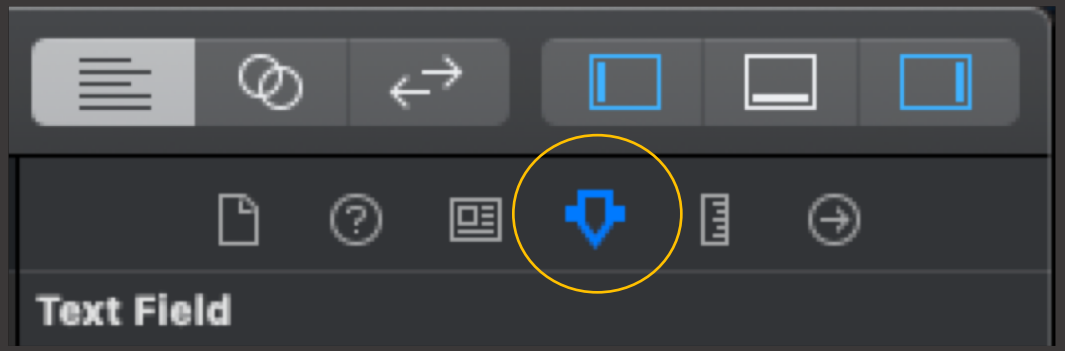
UIView

- You can '**stack**' other UIView objects on top of another UIView
  - addSubview method
  - parent-child relationship
  - superview vs subview

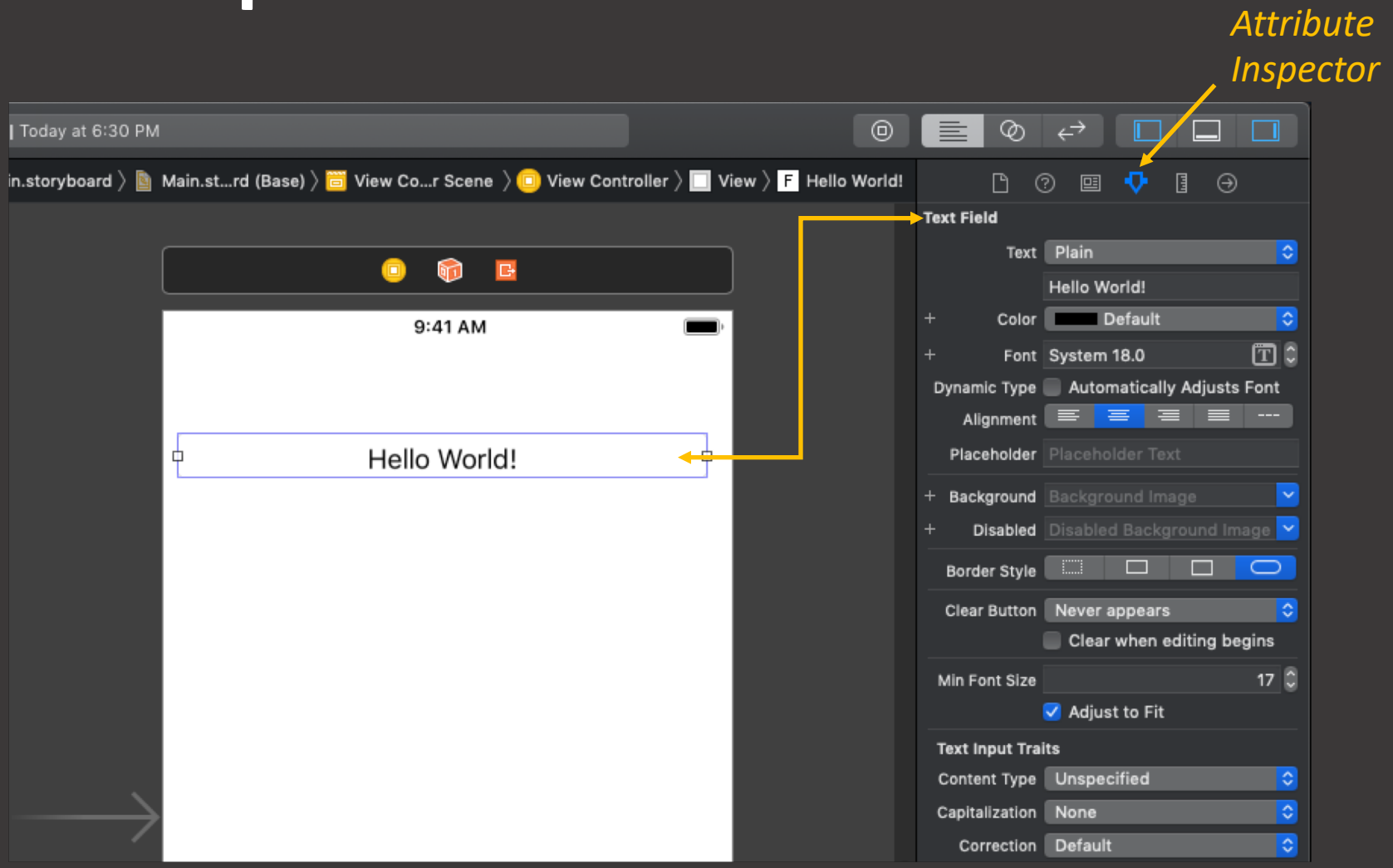
- You can also draw graphics on UIView

Attribute Inspector

- The Attributes inspector appears when you click the fourth button from the left in the inspector selector bar. It lets you edit the properties of an object in your storyboard.



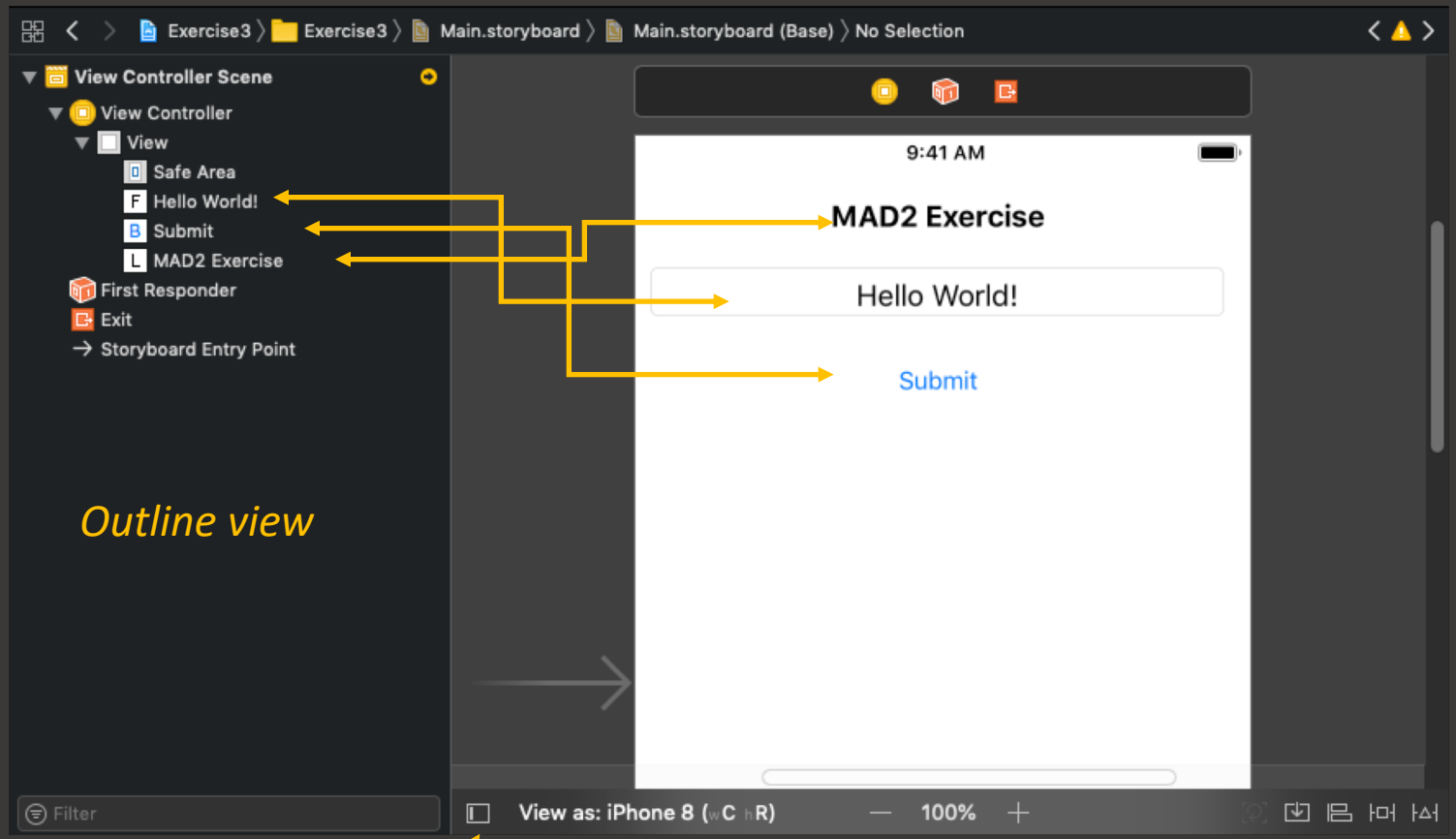
Attribute Inspector



Outline View

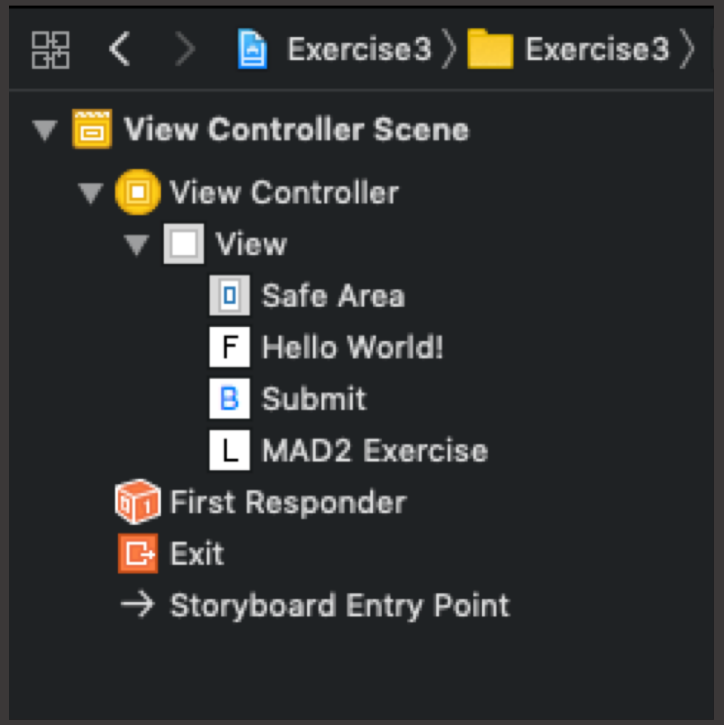
- The outline view, which appears on the left side of the canvas, provides a hierarchical representation of the objects in your storyboard.

Outline View



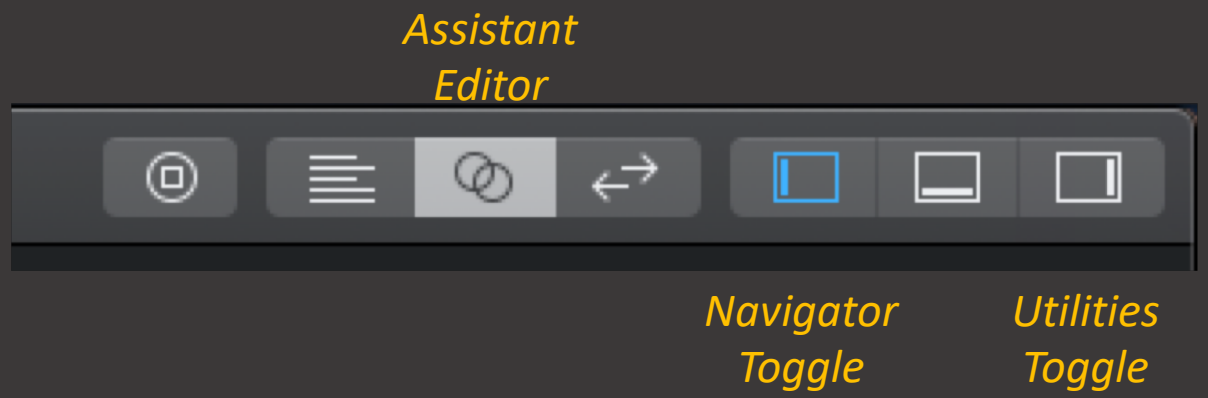
Outline View

- *Why are the user interface elements nested under View?*
- Views not only display themselves onscreen and react to user input, they can serve as containers for other views. Views are arranged in a hierarchical structure called the view hierarchy. The view hierarchy defines the layout of views relative to other views. Within that hierarchy, views enclosed within a view are called subviews, and the parent view that encloses a view is called its superview. A view can have multiple subviews and only one superview.



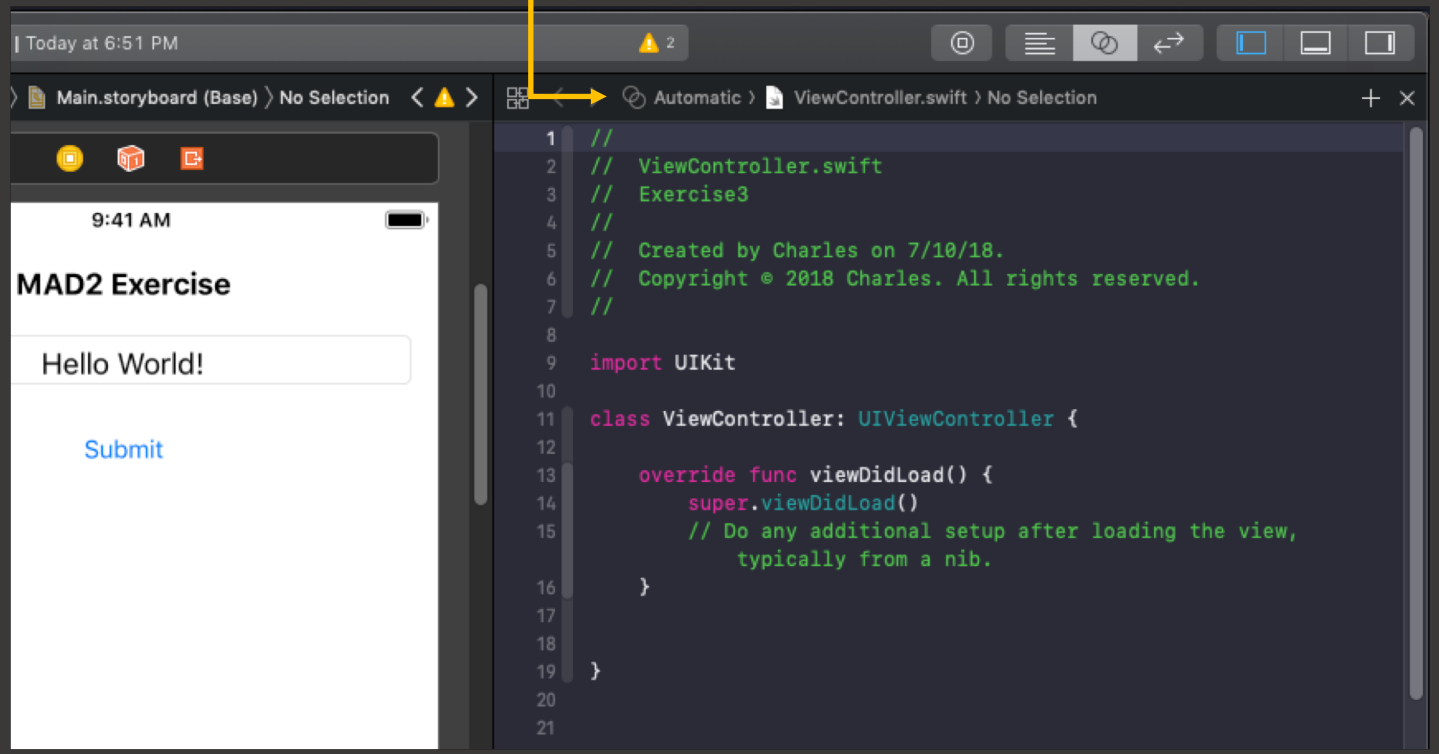
Assistant Editor

- *Developer can preview your app interface using the assistant editor, which displays a secondary editor side-by-side with your main one.*



Assistant Editor

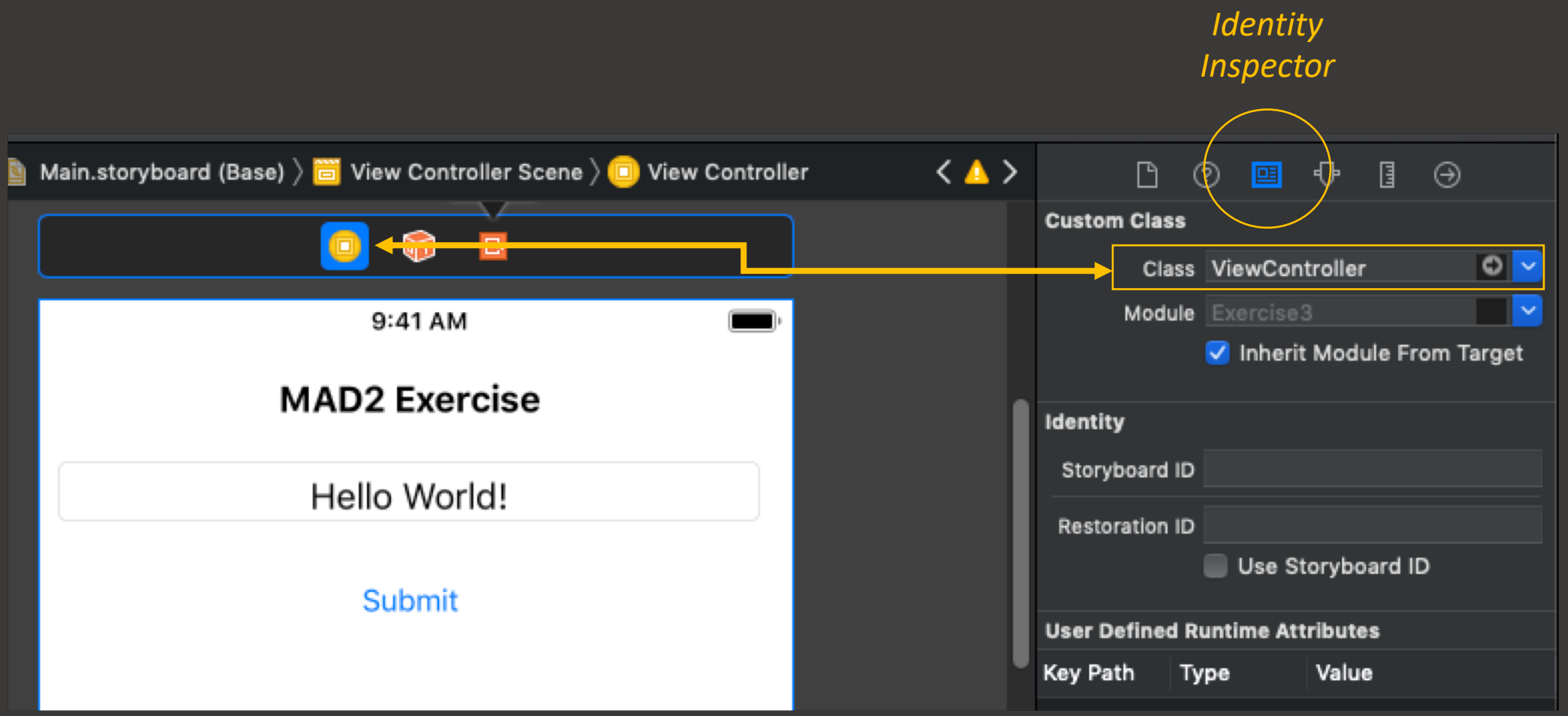
- Automatic



Swift ViewController

- Manages a group of views objects.
  - View controllers usually have a single view with many subviews. The view controller manages the state of these views.
  - A view controller is the glue between the overall application and the screen. It controls the views that it owns according to the logic of the application.

Swift ViewController



Swift ViewController

- Common iOS ViewControllers
  - UIViewController
  - UINavigationController
  - UITableViewController
  - UITabBarController
  - UICollectionViewController
  - UISplitViewController

iOS UI Controls

MAD2 Oct 2018

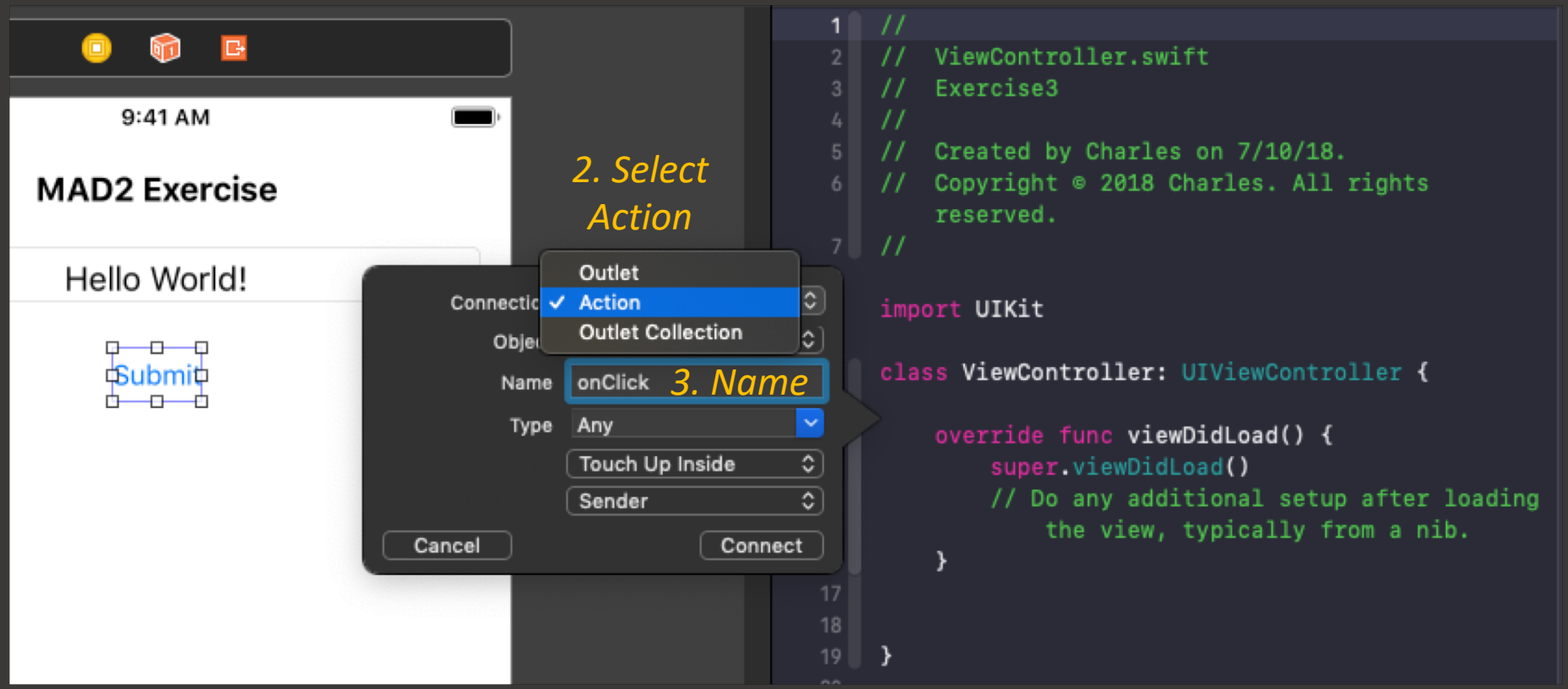
NGEE ANN POLYTECHNIC
SCHOOL OF INFOCOMM TECHNOLOGY

UIButton

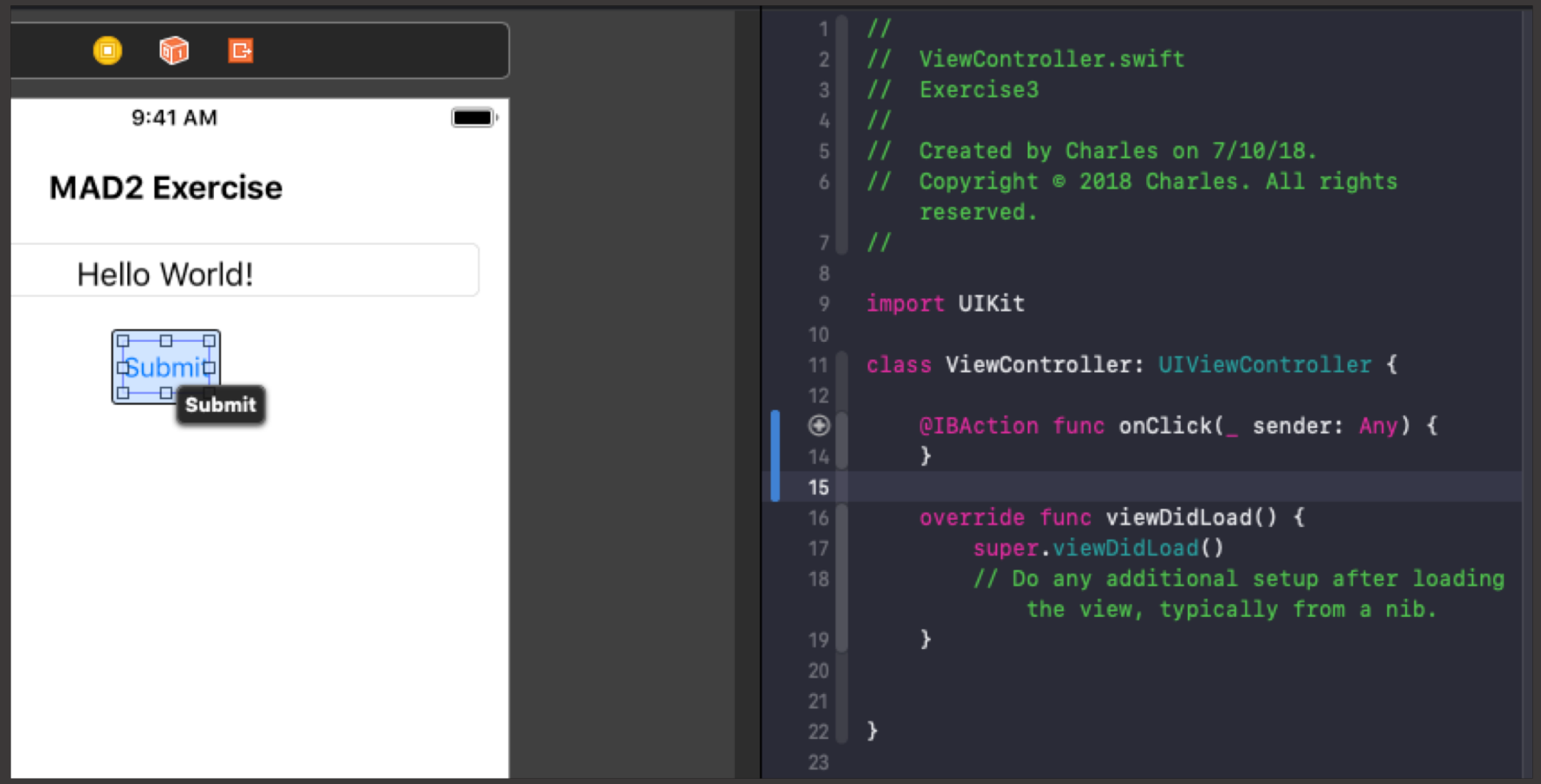
- A control that executes your custom code in response to user interactions.
  - Buttons use the Target-Action design pattern to notify your app when the user taps the button. Rather than handle touch events directly, you assign action methods to the button and designate which events trigger calls to your methods. At runtime, the button handles all incoming touch events and calls your methods in response.

UIButton

1. Right click hold and drag to the ViewController Class



UIButton




```swift
//
//  ViewController.swift
//  Exercise3
//
//  Created by Charles on 7/10/18.
//  Copyright © 2018 Charles. All rights
    reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBAction func onClick(_ sender: Any) {
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading
            the view, typically from a nib.
    }


}
```

Write Code

UITextField

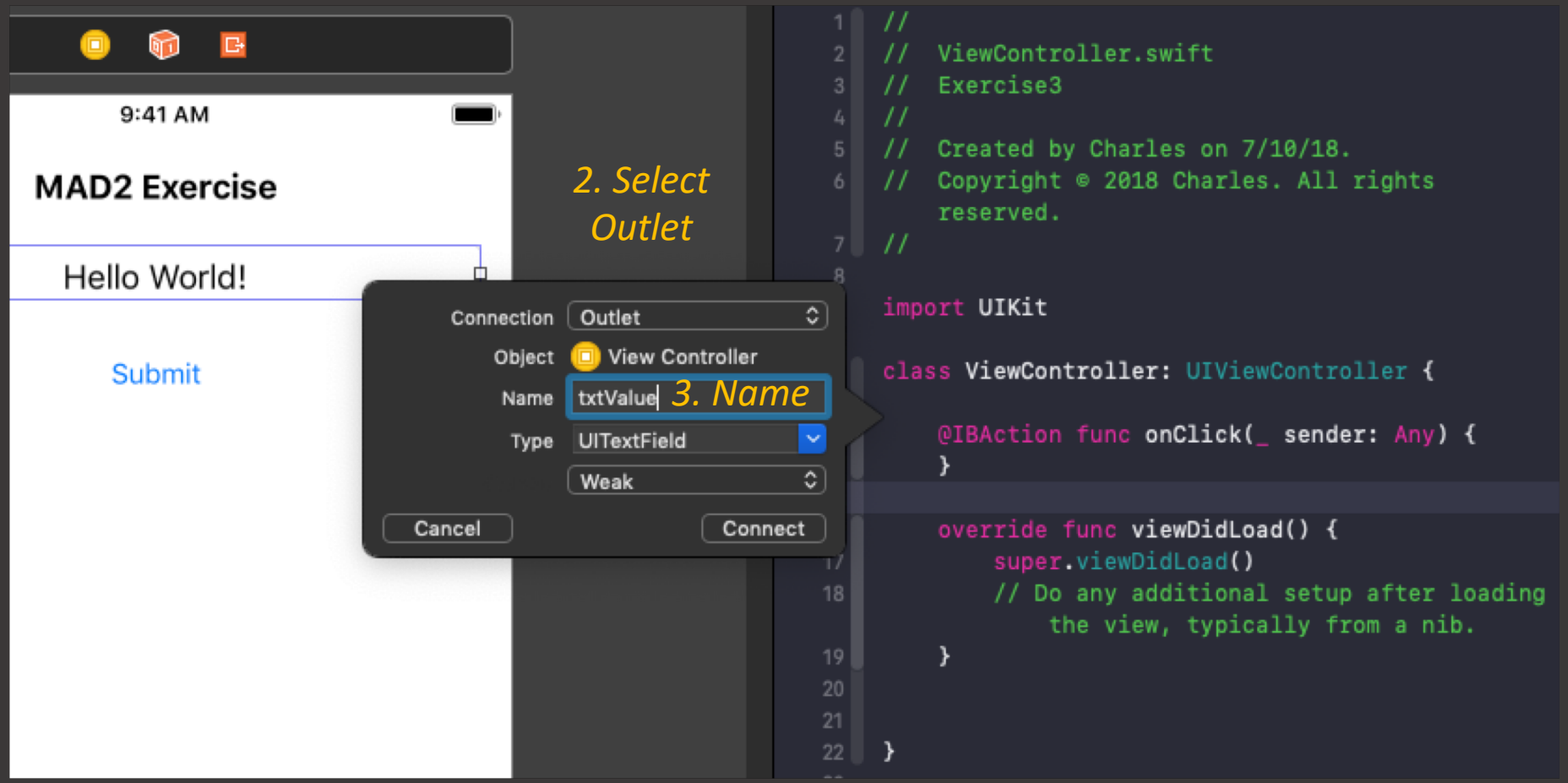
- An object that displays an editable text area in your interface.
  - You use text fields to gather text-based input from the user using the onscreen keyboard. The keyboard is configurable for many different types of input such as plain text, emails, numbers, and so on. Text fields use the target-action mechanism and a delegate object to report changes made during the course of editing.

UITextField

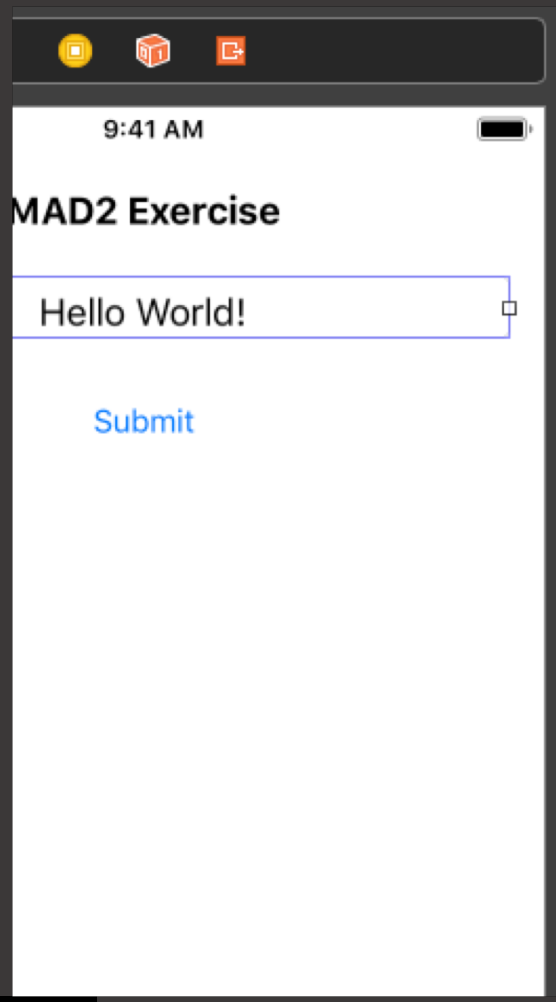
- An object that displays an editable text area in your interface.
  - You use text fields to gather text-based input from the user using the onscreen keyboard. The keyboard is configurable for many different types of input such as plain text, emails, numbers, and so on. Text fields use the target-action mechanism and a delegate object to report changes made during the course of editing.

UITextField

1. Right click hold and drag to the ViewController Class



UITextField




```swift
//
//  ViewController.swift
//  Exercise3
//
//  Created by Charles on 7/10/18.
//  Copyright © 2018 Charles. All rights
    reserved.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var txtValue: UITextField!

    @IBAction func onClick(_ sender: Any) {
        txtValue.text = "I'm not a robot!"
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading
            the view, typically from a nib.
    }


}
```

A UITextField is linked

Populate the txtValue with some strings.

Delegates

Delegate

- Delegation allows an object to notify another object, assigned as delegate,
  - To ask for input or receive an output
  - To notify the delegate that certain event is happening
- ViewController.swift adds a UITextField control
  - ViewController specifies itself as a delegate of UITextField
  - ViewController implements a delegate method of UITextField
  - When user presses RETURN key, the delegate method will be called so that ViewController can follow-up
    - Validate the input
    - Remove the keyboard from screen
    - Change the display of other controls

Example UITextFieldDelegate

- textFieldShouldBeginEditing : this method gets called just before textfield gets active.
- textFieldDidBeginEditing : this method gets called when the textfield active.
- textFieldShouldEndEditing :this method gets called before the textfield becomes inactive.
- textFieldDidEndEditing : this method gets called when the textfield becomes inactive.
- textFieldShouldClear : this method gets called when the clear button pressed.
- textFieldshouldChangeCharactersInRange : this method gets called when while typing every single character before its displayed. if you validate the keyboard for editing certain characters this is the best place to implement.
- textFieldShouldReturn : this method gets called when return key pressed in the keyboard, and it best place to add the dismiss keyboard function.

Self-study: http://sourcefreeze.com/uitextfield-and-uitextfield-delegate-in-swift/

Summary

To be able to understand:

- Views and controllers
- Interface builder
- iOS UI Controls
  - UIButton
  - UITextField
- Delegate