UIControl in Swift

TABLE OF CONTENTS

01

OVERVIEW

- **UIControl** put it simply, a class is the base class for visual elements that convey a specific action or user's intent (drag, button click, etc.).

- Controls implement elements such as buttons and sliders, which your app can use to facilitate navigation, gather user input, or manipulate content.

- Controls use the target-action mechanism to report user interactions to your app.

02

MECHANISM AND TYPES

MECHANISM

- it is a class that gives the user the ability to interact with the view that is presented to the user. UIControluses a mechanism called Target-Action to deliver user actions to the app.

- This Target-Action mechanism is **addTarget**(_:action:for:)implemented using methods. As parameters, it passes the object responsible for the action, the method that defines the action for the action, and for which action (.touchUpInsider, .valueChangedetc.) the method is to be called.

- The **addTarget** method has three parameters, the first parameter is the object that define the event process function, the second parameter is a function name selector, the third parameter is an event name. In below example code, when the touchUpInside event happened on the button, it will invoke current object's (ViewController object) clickButtonThree function.

```swift
class ViewController:UIViewController{
    /*
        code
    */
    override func viewDidLoad(){
        super.viewDidLoad()
        /*
            code
        */
        myButton.addTarget(self, action: #selector(btnClicked),
            for: .touchUpInside)
    }

    @objc func btnClicked(){
        ...
    }
}
```

- From this code, UIButton excutes myButton.addTarget(self, action: #selector(btnClicked), for: .touchUpInside) method which invokes action trough btnClicked()

- As a result, if you add target method, and then action would be invoked when UIcontrol.event .touchUpInside is satisfied.

UIControl.event

- Constants describing the types of events possible for controls.
- Developer can choose which event invoke actions from target.

For example, if you add .touchUpInside(UIControl.event) on addTarget method, action will be invoke when you click inside of button.

```
static var touchUpInside: UIControl.Event
```

A touch-up event in the control where the finger is inside the bounds of the control.

Because there are so many kinds of UIControl.event, it will be better for us to find event which is adapt for your situation in apple document.

UIControl.state

- Constants describing the state of a control.
- UIControl can have multiple states.
- User can set different actions depend on states.

```
button.setTitle("Button1", for: .normal)
button.setTitle("Button2", for: .selected)
```

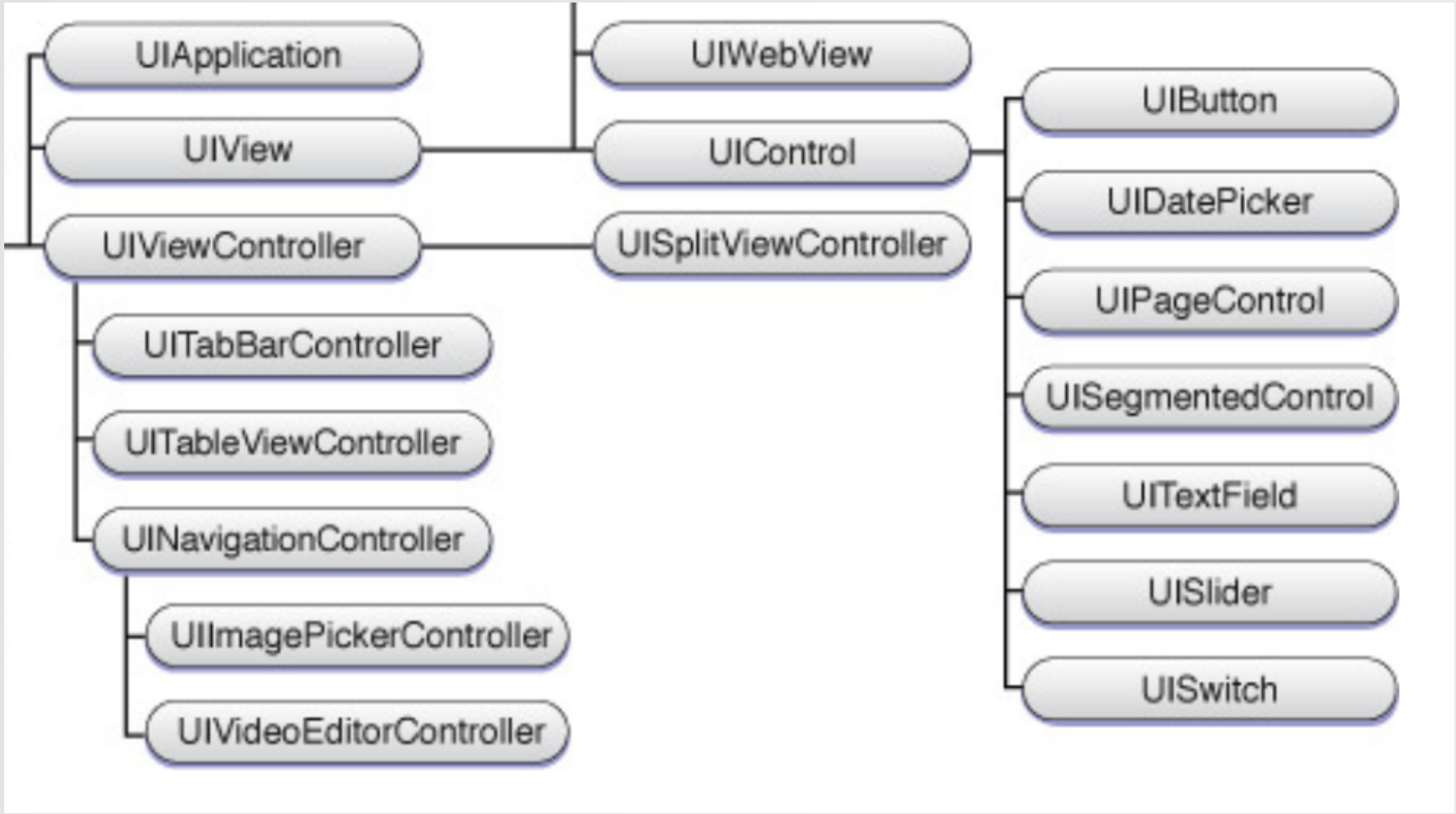
From this code, before select the button it shows “Button1” as title.

After select button, button shows “Button2” as title.

.**normal** and .**selected** is UIControl.state properties.

Especially, .normal is default property. When the control have .normal property, other properties should be false.

TYPES


UIApplication
UIView
UIViewController
UITabBarController
UITableViewController
UINavigationController
UIImagePickerController
UIVideoEditorController
UIWebView
UIControl
UISplitViewController
UIButton
UIDatePicker
UIPageControl
UISegmentedControl
UITextField
UISlider
UISwitch

03

INITIALIZATION AND USING

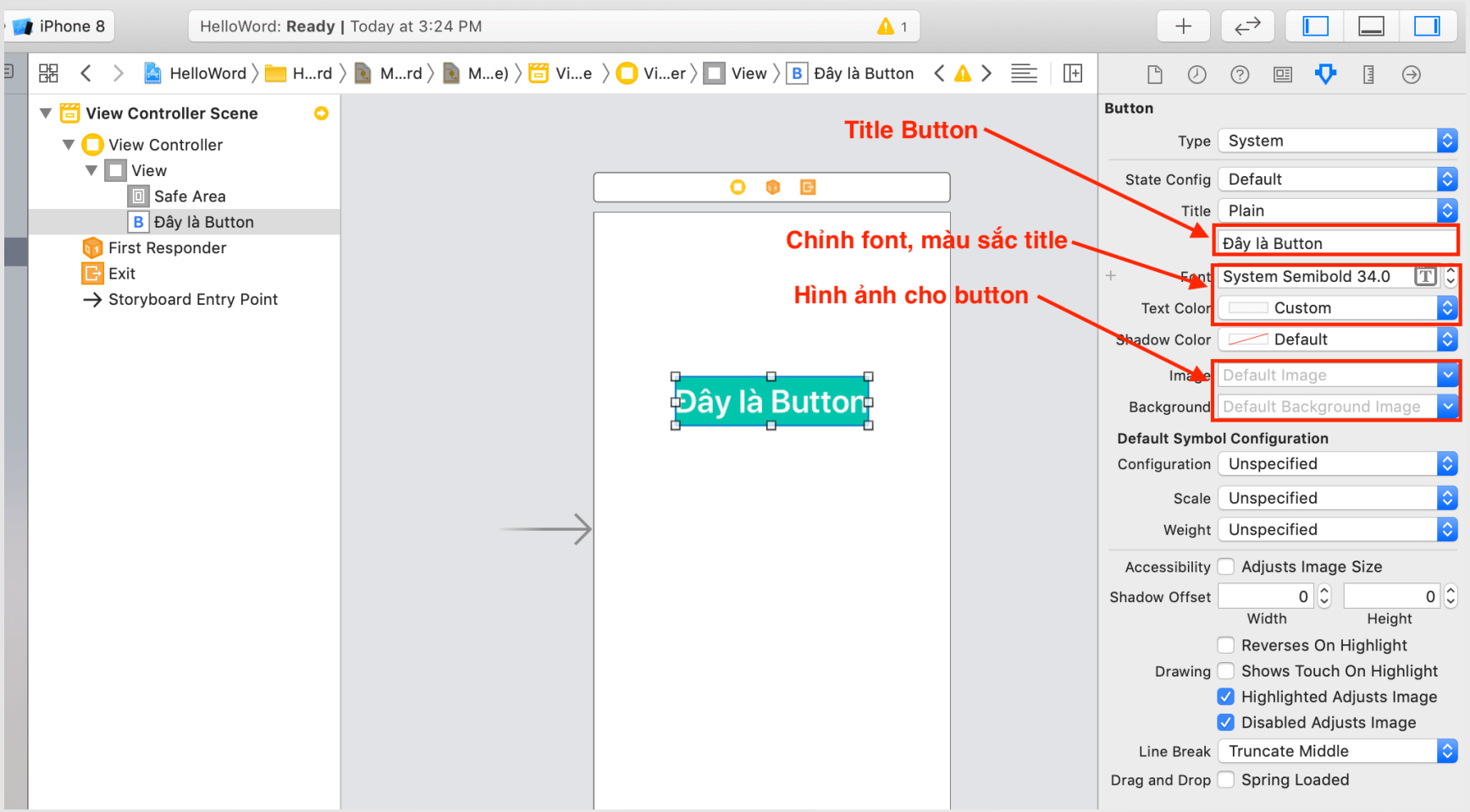
UIButton

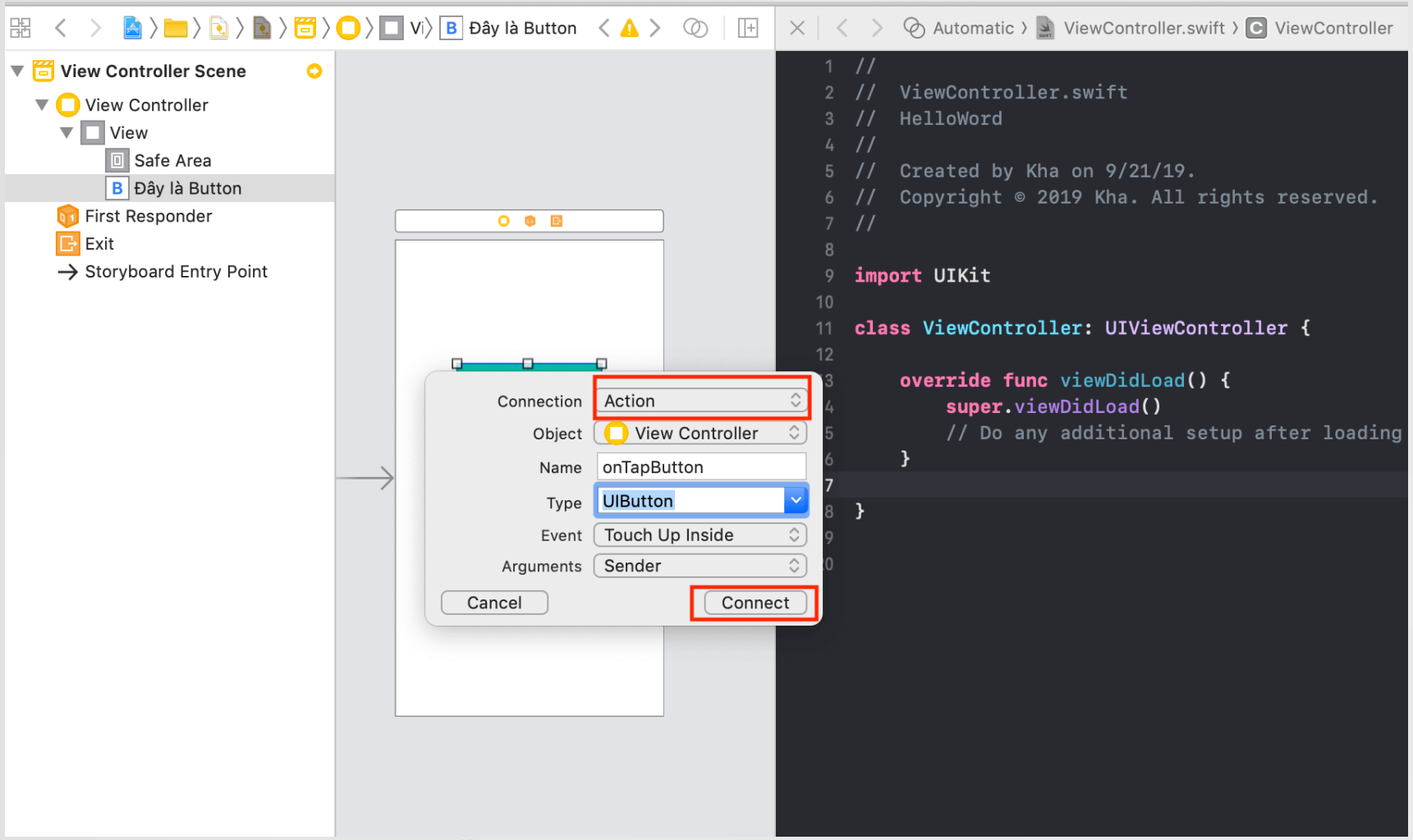
❖ A control that executes your custom code in response to user interactions.

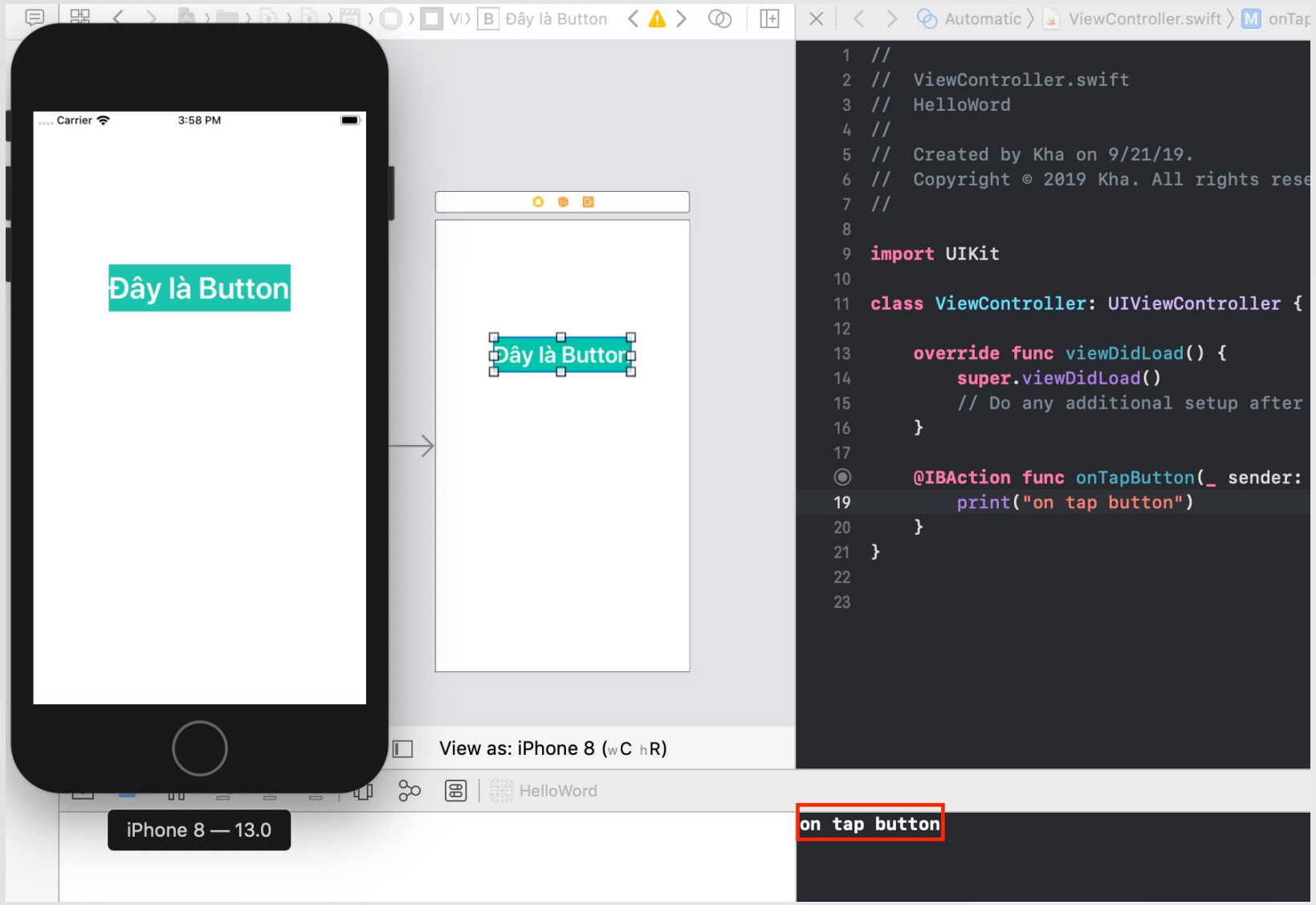
❖ When you tap a button, or select a button that has focus, the button performs any actions attached to it. You communicate the purpose of a button using a text label, an image, or both.

❖ The appearance of buttons is configurable, so you can tint buttons or format titles to match the design of your app.

Create UIButton by drag and drop




Đây là Button
View Controller Scene
View Controller
View
Safe Area
Đây là Button
First Responder
Exit
Storyboard Entry Point
Connection
Action
Object
View Controller
Name
onTapButton
Type
UIButton
Event
Touch Up Inside
Arguments
Sender
Cancel
Connect
Automatic
ViewController.swift
ViewController
//
// ViewController.swift
// HelloWord
//
// Created by Kha on 9/21/19.
// Copyright © 2019 Kha. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading
}
}


Đây là Button
Carrier
3:58 PM
Đây là Button
Đây là Button
View as: iPhone 8 (wC hR)
HelloWord
iPhone 8 — 13.0
Automatic
ViewController.swift
onTap
//
// ViewController.swift
// HelloWord
//
// Created by Kha on 9/21/19.
// Copyright © 2019 Kha. All rights rese
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after
}
@IBAction func onTapButton(_ sender:
print("on tap button")
}
}
on tap button

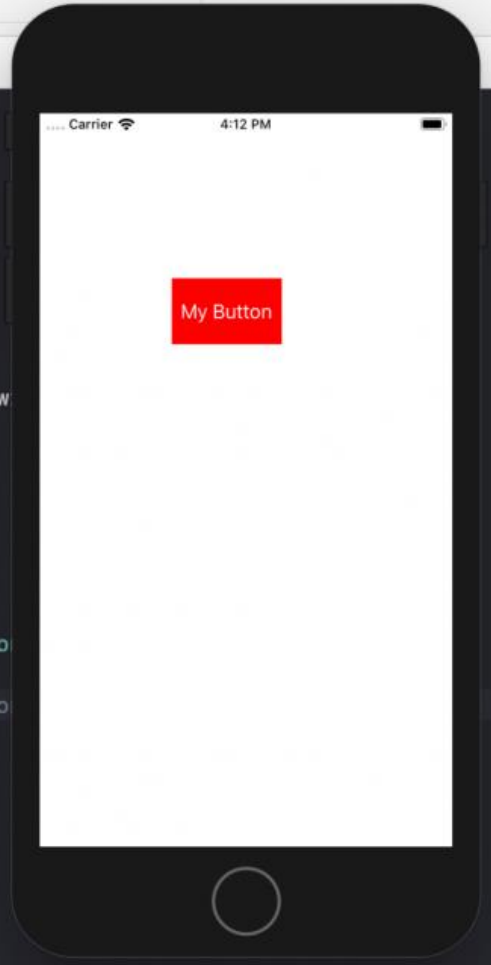
Create UIButton with code

```swift
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        //Khởi tạo và set frame cho myButton
        let myButton = UIButton(frame: CGRect(x: 120, y: 150, width: 100, height: 60))

        //Thay đổi các thuộc tính myButton
        myButton.setTitle("My Button", for: .normal)
        myButton.setTitleColor(.white, for: .normal)
        myButton.backgroundColor = .red

        //Action cho myButton
        myButton.addTarget(self, action: #selector(onTapMyButton), for: .touchUpInside)

        //addSubview myButton nếu không có dòng này thì myButton sẽ không hiển thị
        self.view.addSubview(myButton)
    }

    @objc func onTapMyButton() {
        print("Tapped...")
    }
}
```

iPhone 8 | Running HelloWord on iPhone 8

HelloWord 〉 HelloWord 〉 ViewController.swift 〉 viewDidLoad()

```swift
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        //Khởi tạo và set frame cho myButton
        let myButton = UIButton(frame: CGRect(x: 120, y: 150, w

        //Thay đổi các thuộc tính myButton
        myButton.setTitle("My Button", for: .normal)
        myButton.setTitleColor(.white, for: .normal)
        myButton.backgroundColor = .red

        //Action cho myButton
        myButton.addTarget(self, action: #selector(onTapMyButto

        //addSubview myButton nếu không có dòng này thì myButto
        self.view.addSubview(myButton)
    }

    @objc func onTapMyButton() {
        print("Tapped...")
    }
}
```



iPhone 8 — 13.0

HelloWord

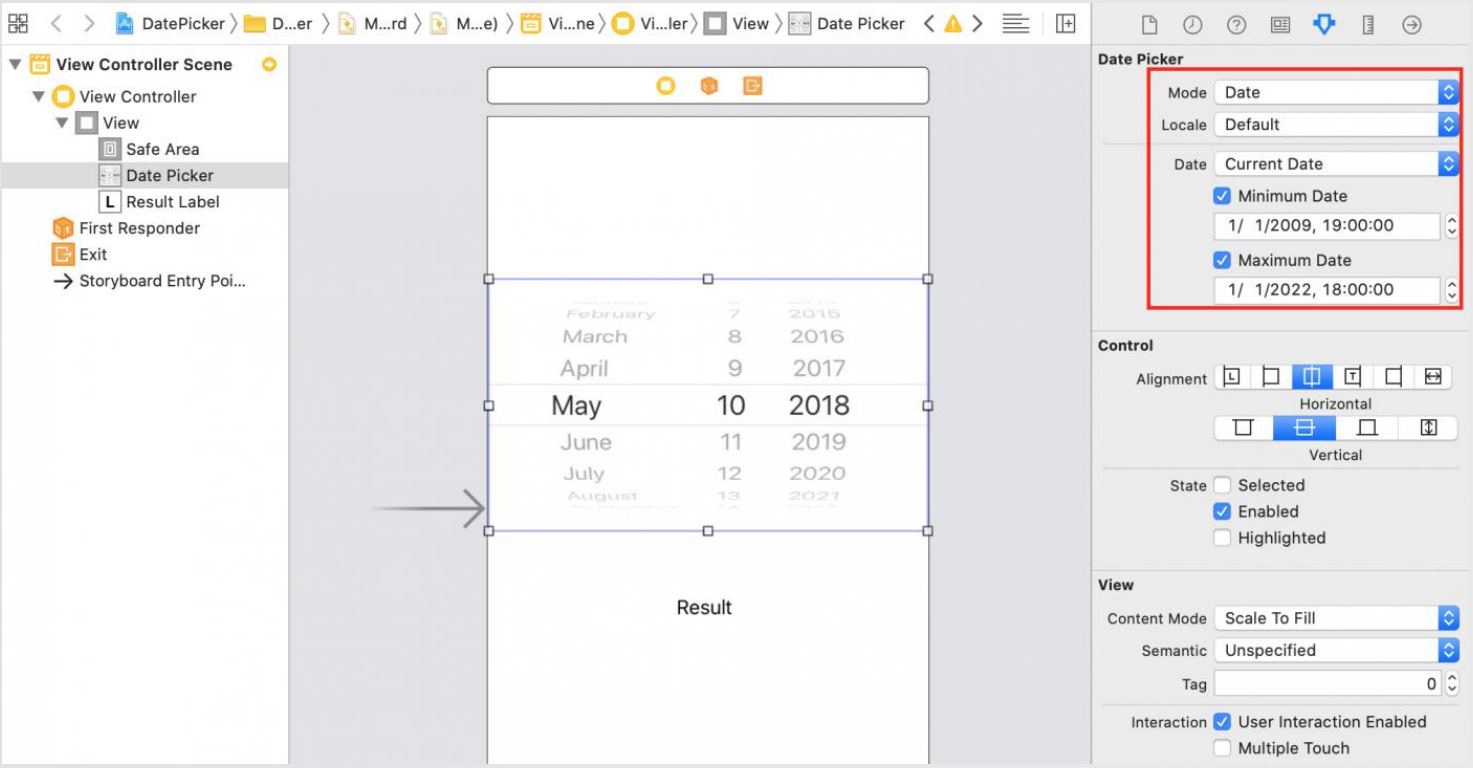
Tapped...

UIDatePicker

❖ A control used for the inputting of date and time values.

❖ You can use a date picker to allow a user to enter either a point in time (calendar date, time value or both) or a time interval (for example for a timer).

❖ The date picker reports interactions to its associated target object.



Mode: The display modes of UIDatePicker have 4 types (Time, Date and Time, Count Down Timer).

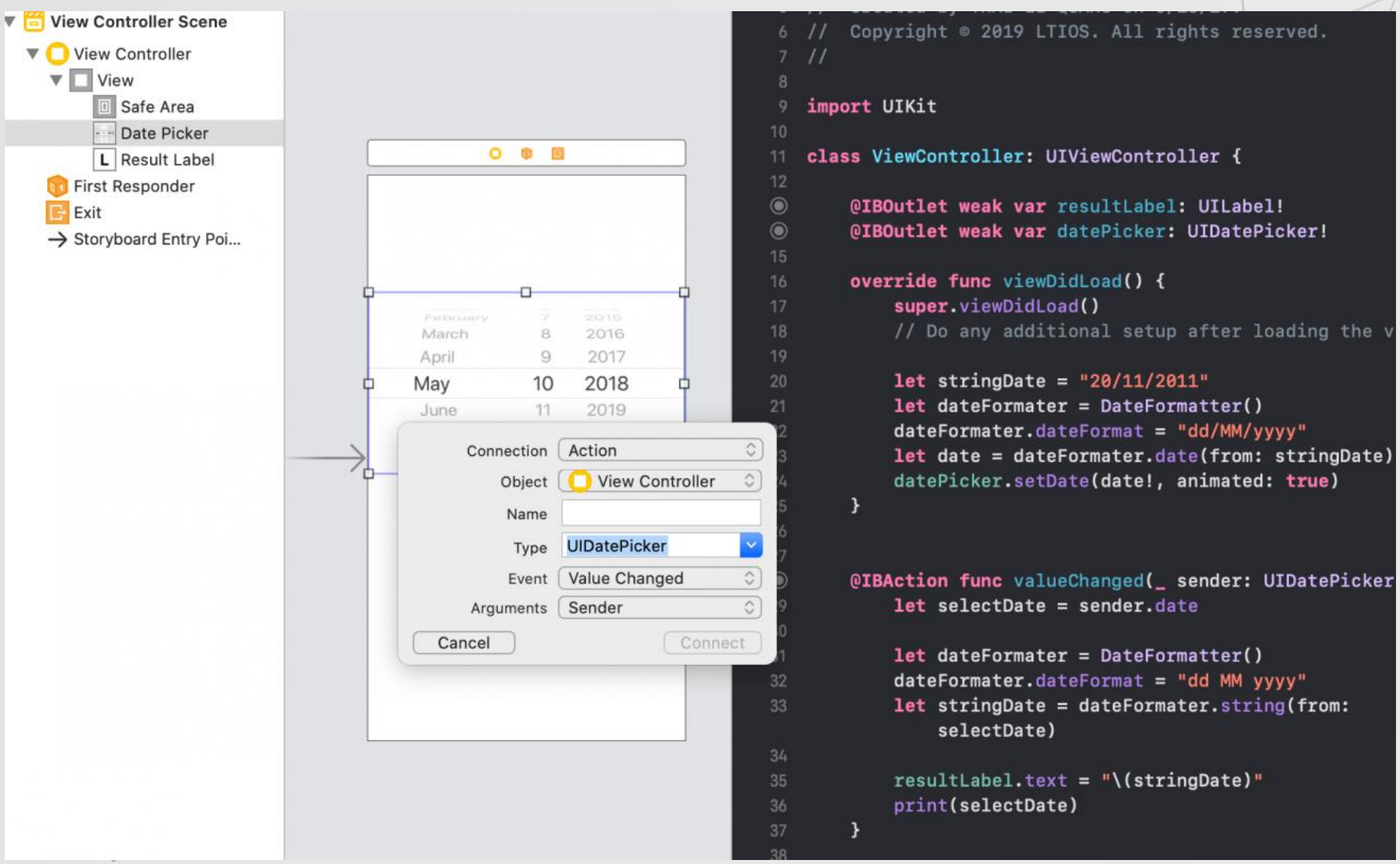
Locale: date and time of the displayed country.

Date: the date that UIDatePicker is selecting (if Current Date is selected, the current date will be obtained).

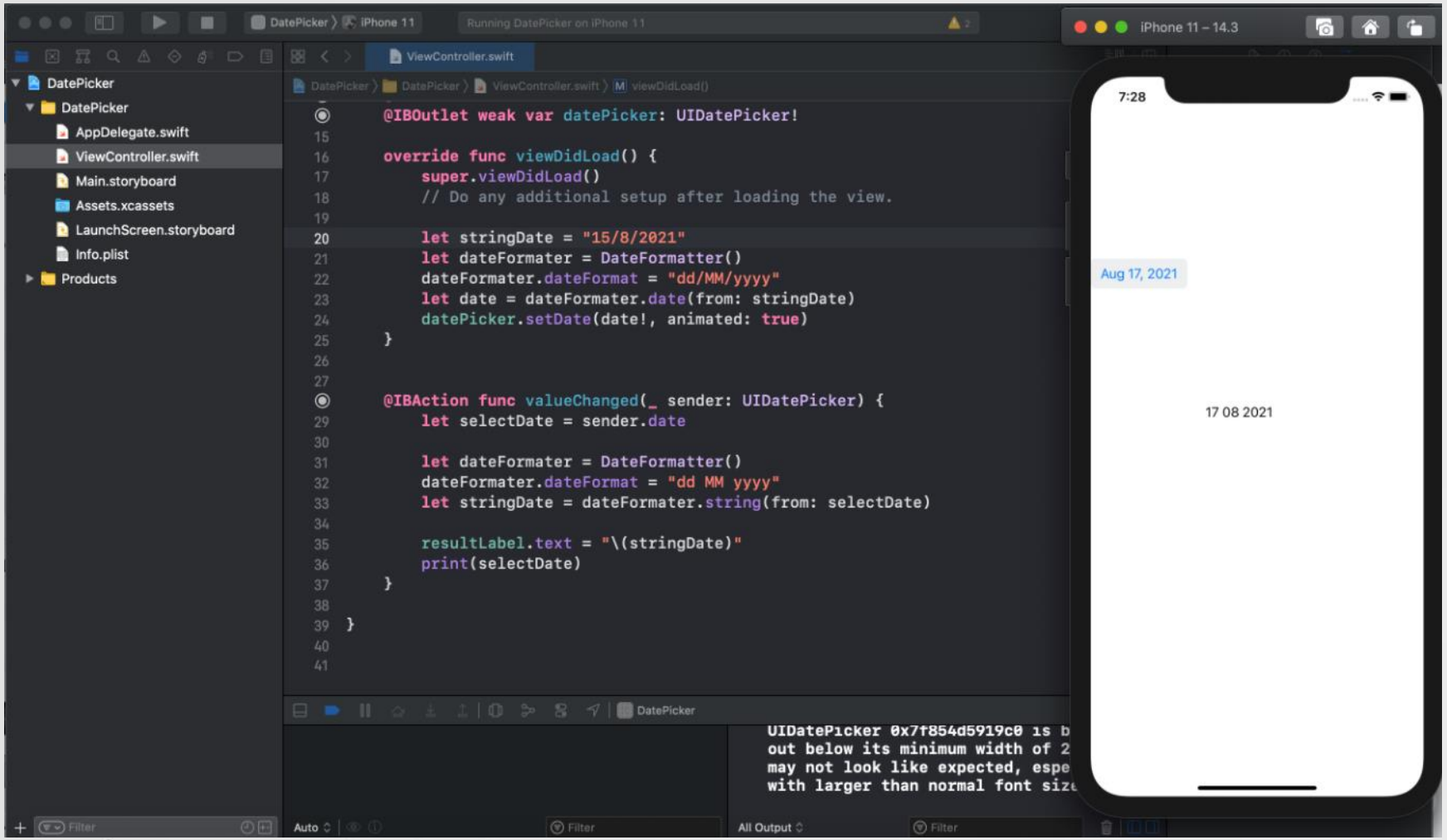
Minimum Date: The maximum date displayed.

Maximum Date: the smallest date displayed.

Action and display date format



```swift
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var resultLabel: UILabel!
    @IBOutlet weak var datePicker: UIDatePicker!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view.

        let stringDate = "15/8/2021"
        let dateFormater = DateFormatter()
        dateFormater.dateFormat = "dd/MM/yyyy"
        let date = dateFormater.date(from: stringDate)
        datePicker.setDate(date!, animated: true)
    }


    @IBAction func valueChanged(_ sender: UIDatePicker) {
        let selectDate = sender.date

        let dateFormater = DateFormatter()
        dateFormater.dateFormat = "dd MM yyyy"
        let stringDate = dateFormater.string(from: selectDate)

        resultLabel.text = "\(stringDate)"
        print(selectDate)
    }

}
```


DatePicker
iPhone 11
Running DatePicker on iPhone 11
ViewController.swift
DatePicker
DatePicker
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
DatePicker > DatePicker > ViewController.swift > viewDidLoad()
@IBOutlet weak var datePicker: UIDatePicker!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
let stringDate = "15/8/2021"
let dateFormater = DateFormatter()
dateFormater.dateFormat = "dd/MM/yyyy"
let date = dateFormater.date(from: stringDate)
datePicker.setDate(date!, animated: true)
}
@IBAction func valueChanged(_ sender: UIDatePicker) {
let selectDate = sender.date
let dateFormater = DateFormatter()
dateFormater.dateFormat = "dd MM yyyy"
let stringDate = dateFormater.string(from: selectDate)
resultLabel.text = "\(stringDate)"
print(selectDate)
}
}
UIDatePicker 0x7f854d5919c0 is b
out below its minimum width of 2
may not look like expected, espe
with larger than normal font size
iPhone 11 – 14.3
7:28
Aug 17, 2021
17 08 2021

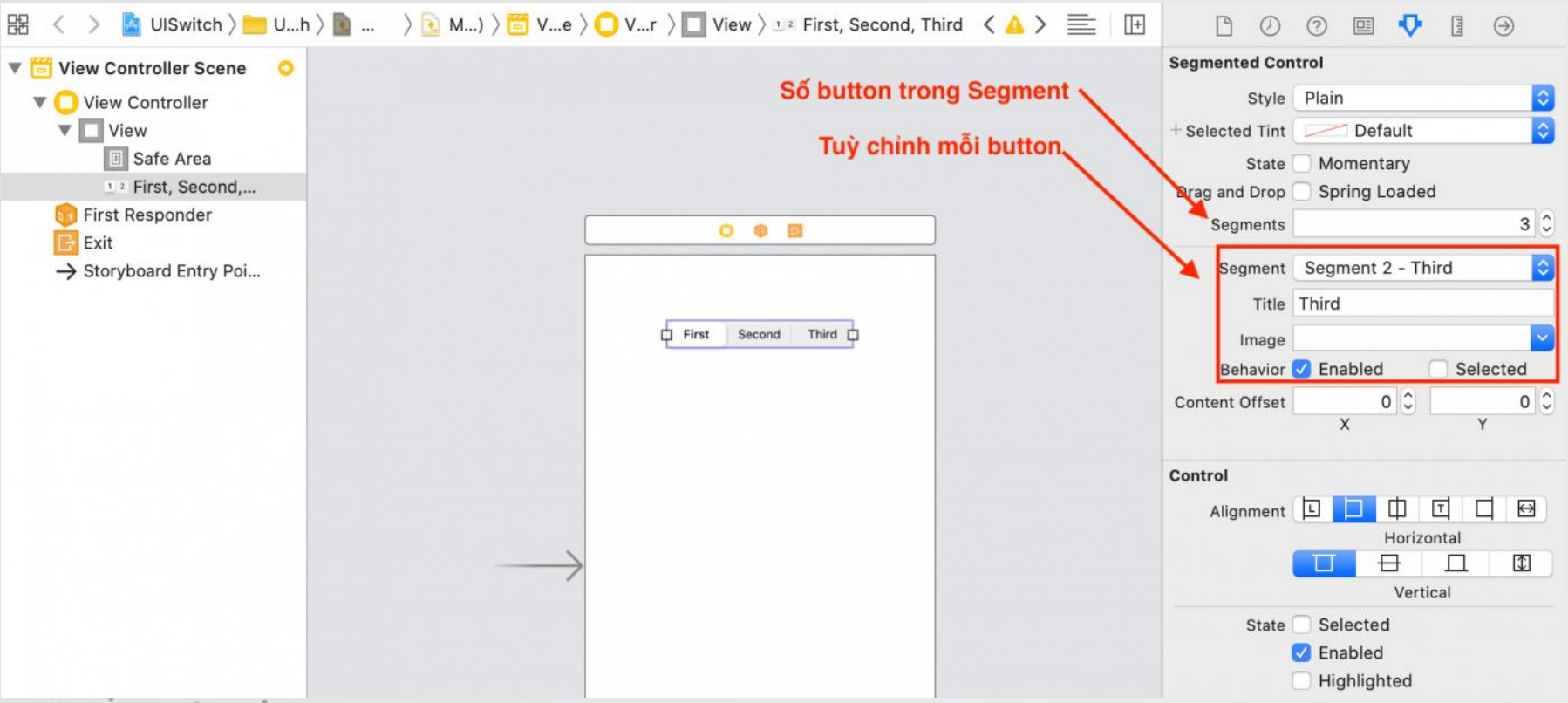
UISegmentedControl

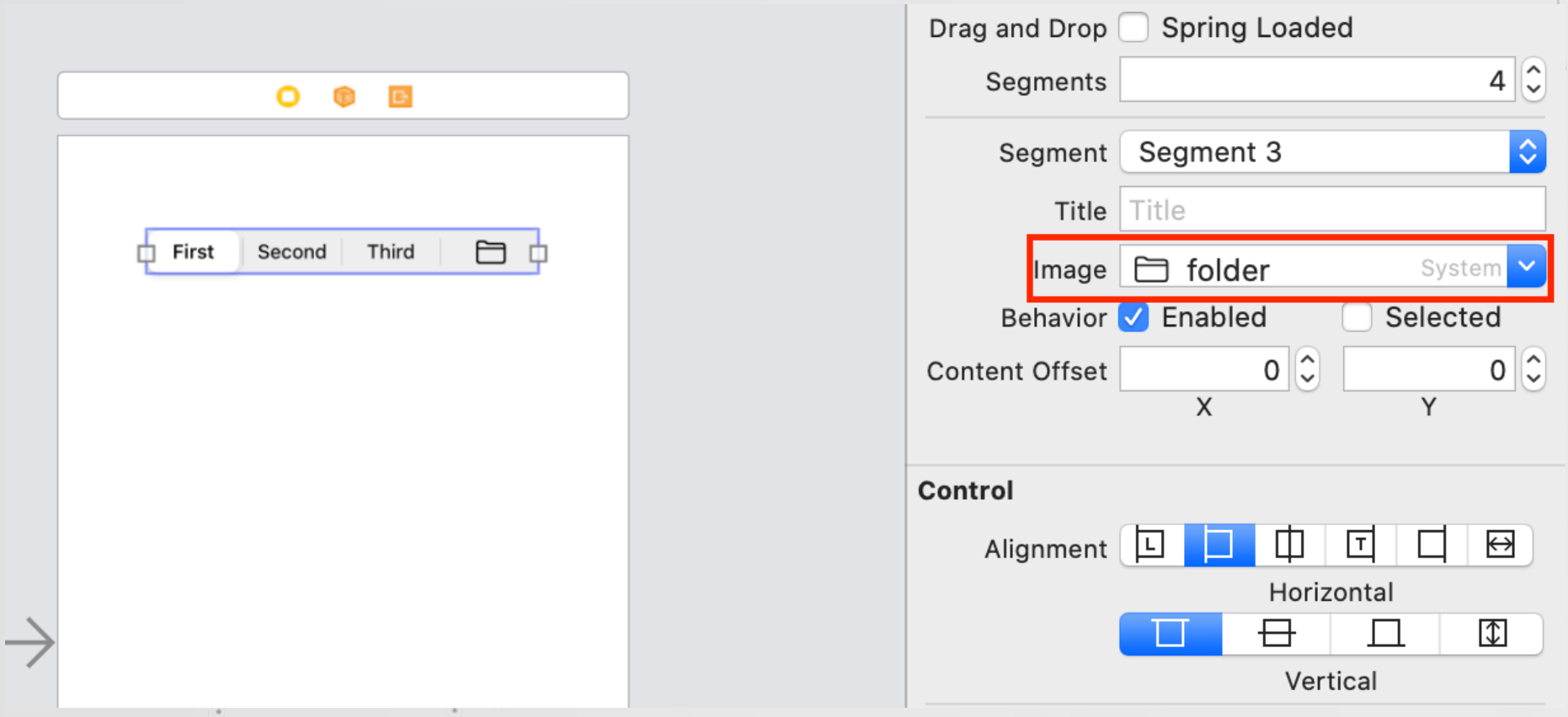
❖ A horizontal control made of multiple segments, each segment functioning as a discrete button.

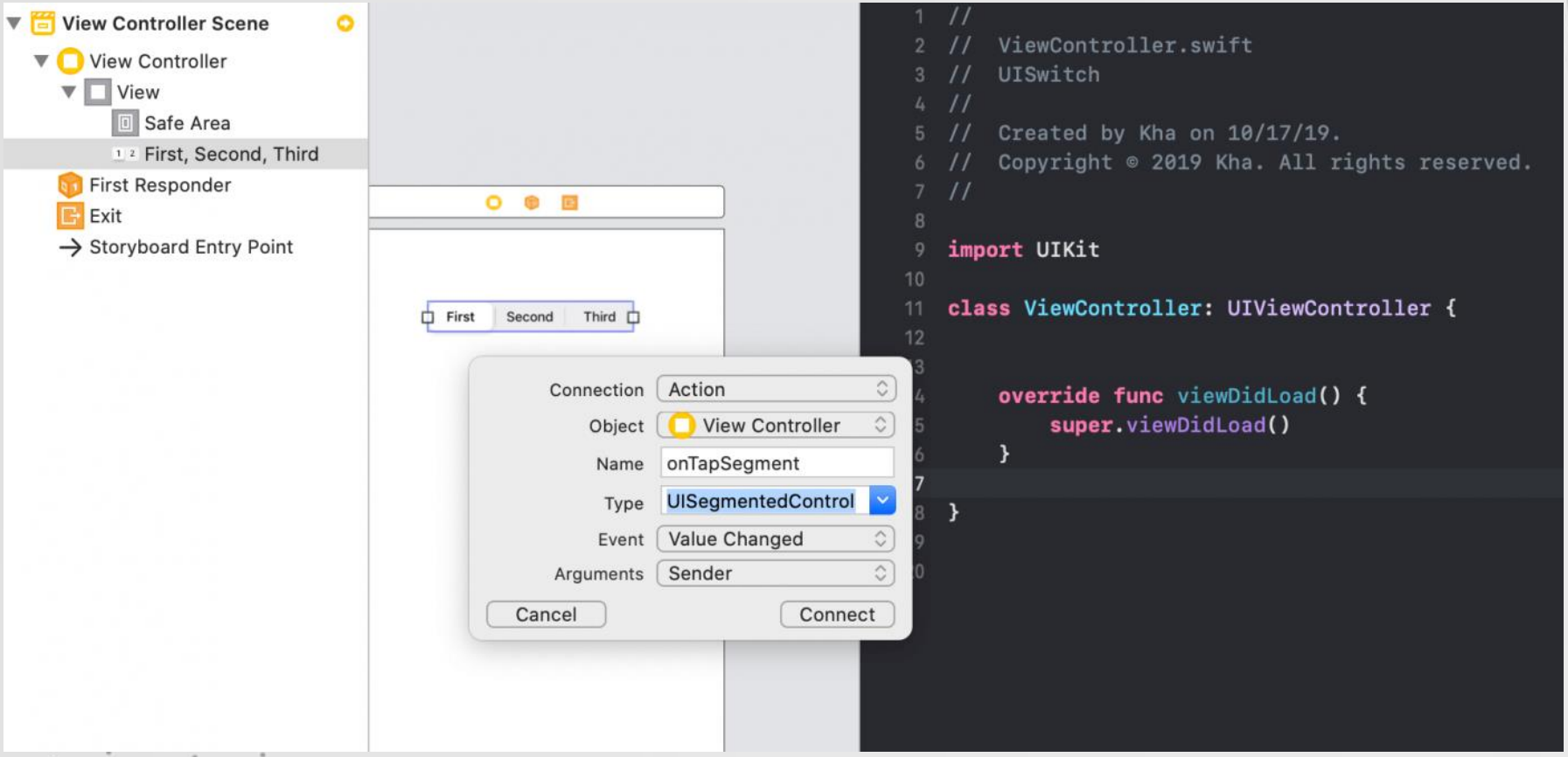
❖ A segmented control can display a title (an NSString object) or an image (UIImage object). The UISegmentedControl object automatically resizes segments to fit proportionally within their superview unless they have a specific width set.

❖ When you add and remove segments, you can request that the action be animated with sliding and fading effects.

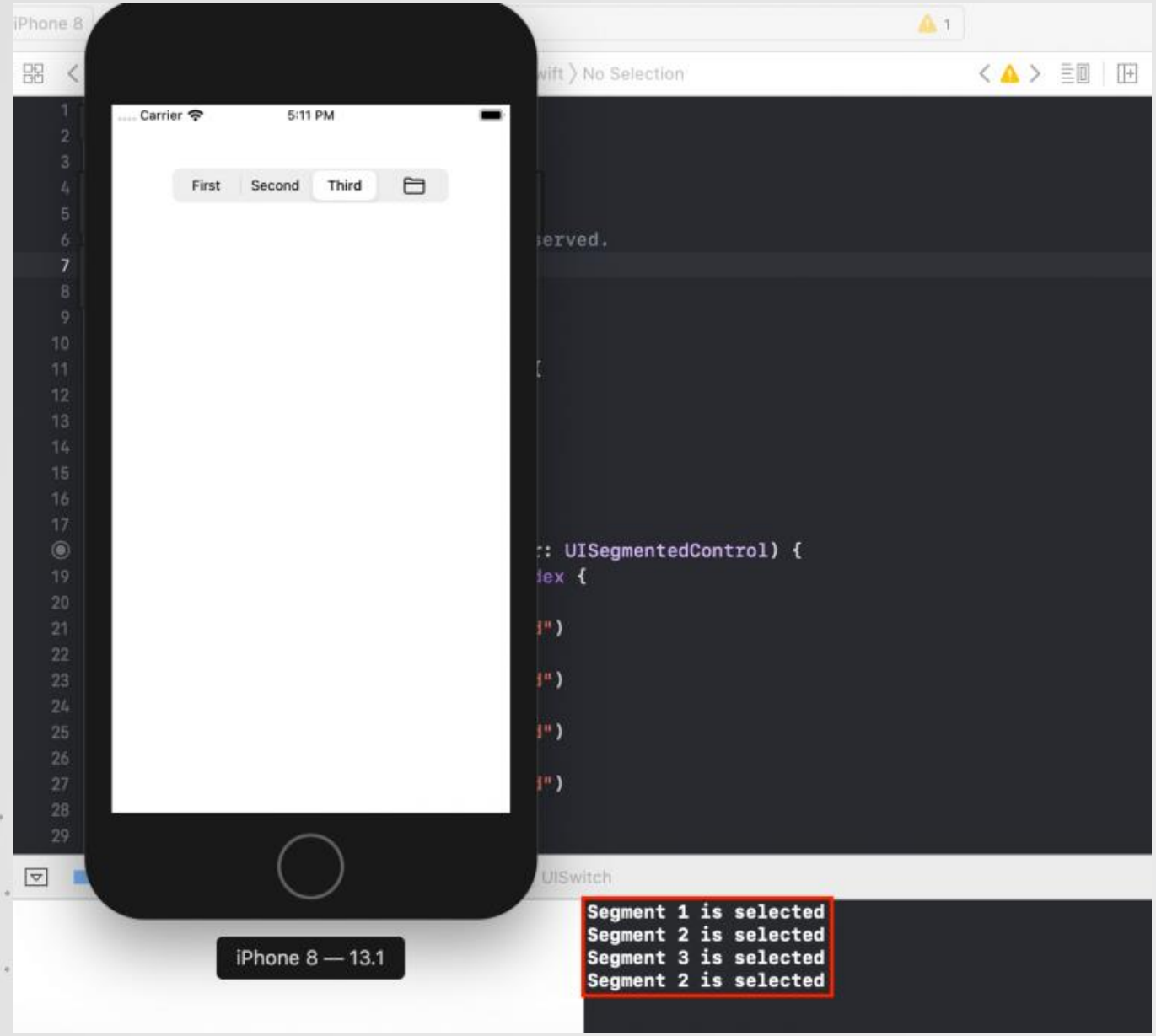
Create UISegmentedControl by drag and drop




First
Second
Third
Drag and Drop
Spring Loaded
Segments
4
Segment
Segment 3
Title
Title
Image
folder
System
Behavior
Enabled
Selected
Content Offset
0
0
X
Y
Control
Alignment
Horizontal
Vertical


View Controller Scene
View Controller
View
Safe Area
First, Second, Third
First Responder
Exit
Storyboard Entry Point
First
Second
Third
Connection
Action
Object
View Controller
Name
onTapSegment
Type
UISegmentedControl
Event
Value Changed
Arguments
Sender
Cancel
Connect
//
// ViewController.swift
// UISwitch
//
// Created by Kha on 10/17/19.
// Copyright © 2019 Kha. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
}
}

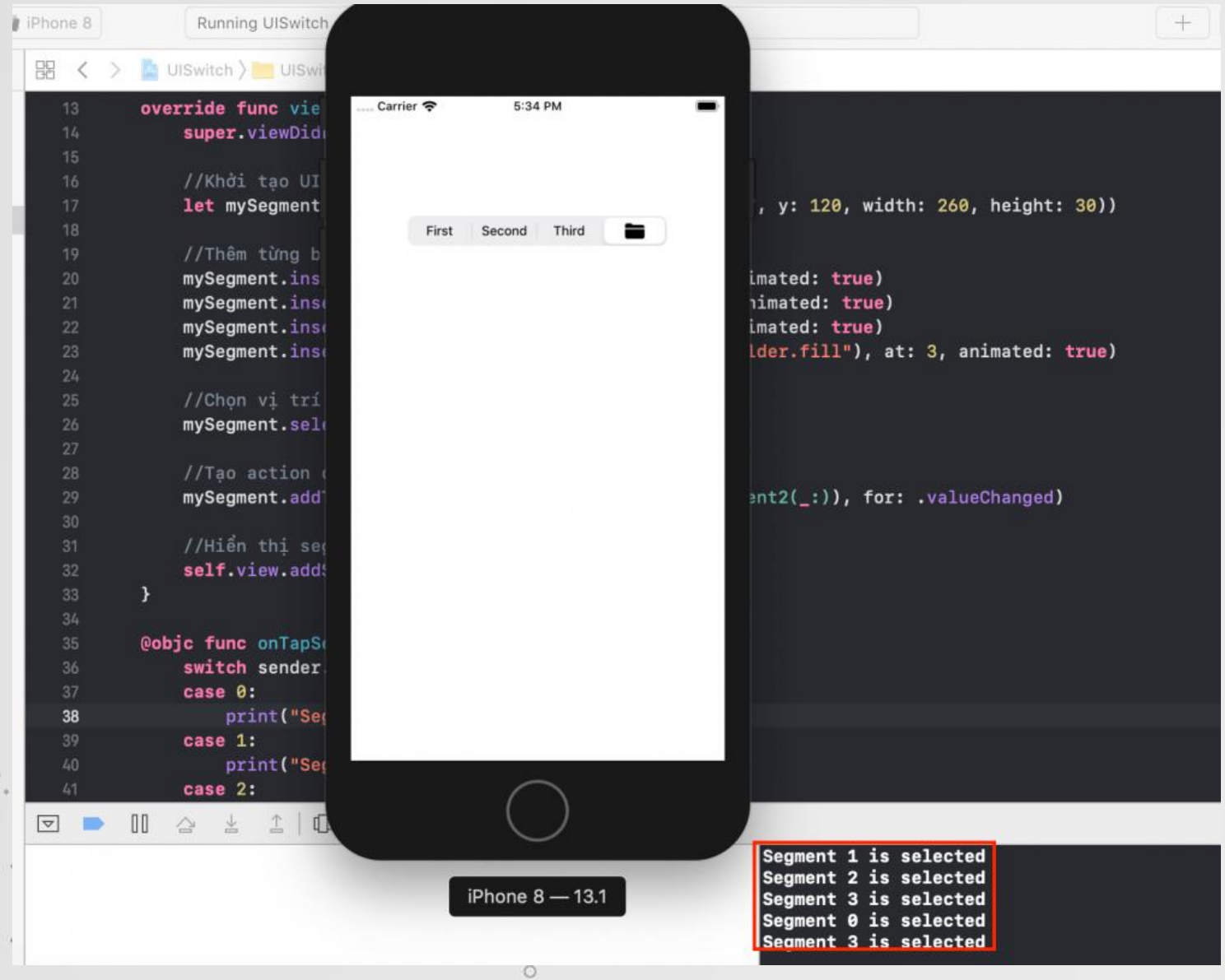
```swift
@IBAction func onTapSegment(_ sender: UISegmentedControl) {
    switch sender.selectedSegmentIndex {
    case 0:
        print("Segment 0 is selected")
    case 1:
        print("Segment 1 is selected")
    case 2:
        print("Segment 2 is selected")
    case 3:
        print("Segment 3 is selected")
    default:
        break
    }
}
```


iPhone 8
No Selection
Carrier
5:11 PM
First
Second
Third
served.
: UISegmentedControl) {
dex {
UISwitch
Segment 1 is selected
Segment 2 is selected
Segment 3 is selected
Segment 2 is selected
iPhone 8 — 13.1

Create UISegmentedControl with code

```
class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

        //Khởi tạo UISegmentedControl với frame
        let mySegment = UISegmentedControl(frame: CGRect(x: 57, y: 120, width: 260, height: 30))

        //Thêm từng button cho Segment
        mySegment.insertSegment(withTitle: "First", at: 0, animated: true)
        mySegment.insertSegment(withTitle: "Second", at: 1, animated: true)
        mySegment.insertSegment(withTitle: "Third", at: 2, animated: true)
        mySegment.insertSegment(with: UIImage(systemName: "folder.fill"), at: 3, animated: true)

        //Chọn vị trí selectedSegmentIndex mặc định là 0
        mySegment.selectedSegmentIndex = 2

        //Tạo action cho Segment
        mySegment.addTarget(self, action: #selector(onTapSegment2(_:)), for: .valueChanged)

        //Hiển thị segment trên view
        self.view.addSubview(mySegment)
    }
```

```
    @objc func onTapSegment2(_ sender: UISegmentedControl) {
        switch sender.selectedSegmentIndex {
        case 0:
            print("Segment 0 is selected")
        case 1:
            print("Segment 1 is selected")
        case 2:
            print("Segment 2 is selected")
        case 3:
            print("Segment 3 is selected")
        default:
            break
        }
    }
}
```

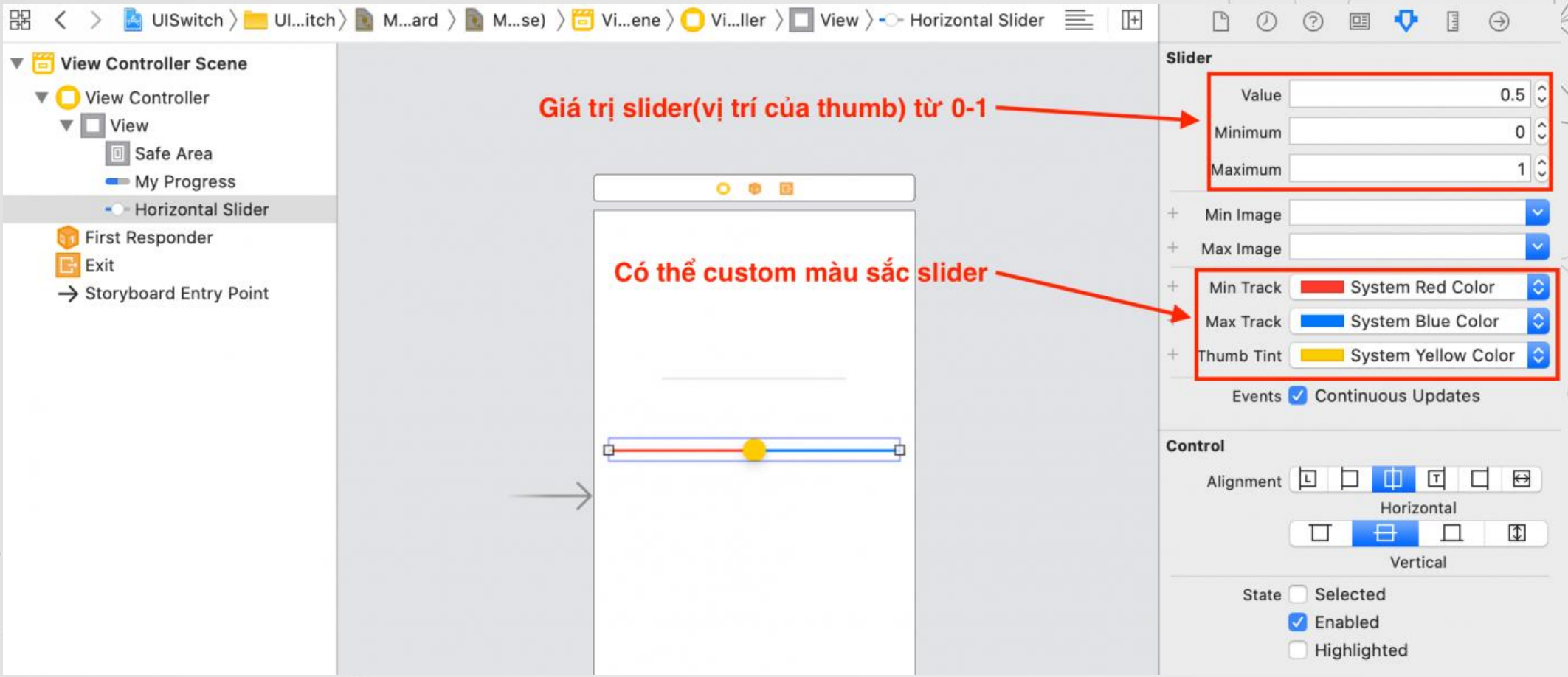

iPhone 8
Running UISwitch
UISwitch
13 override func vie
14 super.viewDid
15
16 //Khởi tạo UI
17 let mySegment
18
19 //Thêm từng b
20 mySegment.ins
21 mySegment.ins
22 mySegment.ins
23 mySegment.ins
24
25 //Chọn vị trí
26 mySegment.sel
27
28 //Tạo action
29 mySegment.add
30
31 //Hiển thị se
32 self.view.add
33 }
34
35 @objc func onTapS
36 switch sender.
37 case 0:
38 print("Se
39 case 1:
40 print("Se
41 case 2:
, y: 120, width: 260, height: 30))
imated: true)
imated: true)
imated: true)
lder.fill"), at: 3, animated: true)
ent2(_:)), for: .valueChanged)
Carrier
5:34 PM
First
Second
Third
iPhone 8 — 13.1
Segment 1 is selected
Segment 2 is selected
Segment 3 is selected
Segment 0 is selected
Segment 3 is selected

UISlider

❖ A control used to select a single value from a continuous range of values.

❖ As you move the thumb of a slider, it passes its updated value to any actions attached to it. The appearance of sliders is configurable; you can tint the track and the thumb, and provide images to appear at the ends of the slider.

Create UISlider by drag and drop

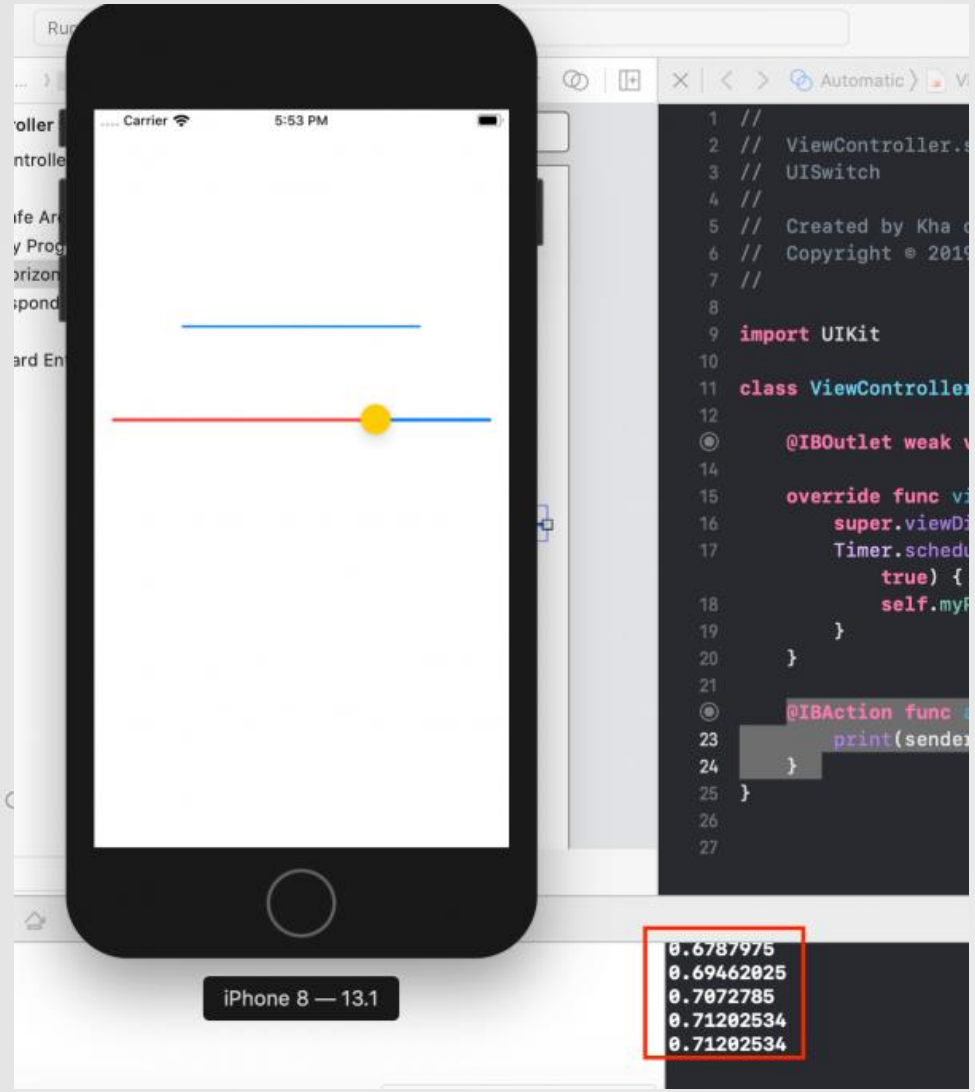


View Controller Scene
- View Controller
  - View
    - Safe Area
    - My Progress
    - Horizontal Slider
  - First Responder
  - Exit
  - Storyboard Entry Point

Connection: Action
Object: View Controller
Name: actionSlider
Type: UISlider
Event: Value Changed
Arguments: Sender
Cancel | Connect

```swift
//
//  ViewController.swift
//  UISwitch
//
//  Created by Kha on 10/17/
//  Copyright © 2019 Kha. Al
//

import UIKit

class ViewController: UIView

    @IBOutlet weak var myPro

    override func viewDidLoa
        super.viewDidLoad()
        Timer.scheduledTimer
            true) { (_) in
            self.myProgress.
        }
    }


}
```

```swift
@IBAction func actionSlider(_ sender: UISlider) {
    print(sender.value)
}
```



Create UISlider with code

```
import UIKit
import PlaygroundSupport

class MyViewController : UIViewController {

    let step:Float = 10 // If you want UISlider to snap to steps by 10

    override func viewDidLoad() {
        super.viewDidLoad()

        let view = UIView()
        view.backgroundColor = .white

        let mySlider = UISlider(frame:CGRect(x: 0, y: 0, width: 300, height:
            20))
        mySlider.center = self.view.center
        mySlider.minimumValue = 0
        mySlider.maximumValue = 100
        mySlider.isContinuous = true
        mySlider.tintColor = UIColor.green
        mySlider.addTarget(self, action:
            #selector(self.sliderValueDidChange(_:)), for: .valueChanged)

        view.addSubview(mySlider)

        UIView.animate(withDuration: 0.8) {
            mySlider.setValue(80.0, animated: true)
        }
        self.view = view
    }

    @objc func sliderValueDidChange(_ sender:UISlider!)
    {
        let roundedStepValue = round(sender.value / step) * step
        sender.value = roundedStepValue
    }
}
// Present the view controller in the Live View window
PlaygroundPage.current.liveView = MyViewController()
```

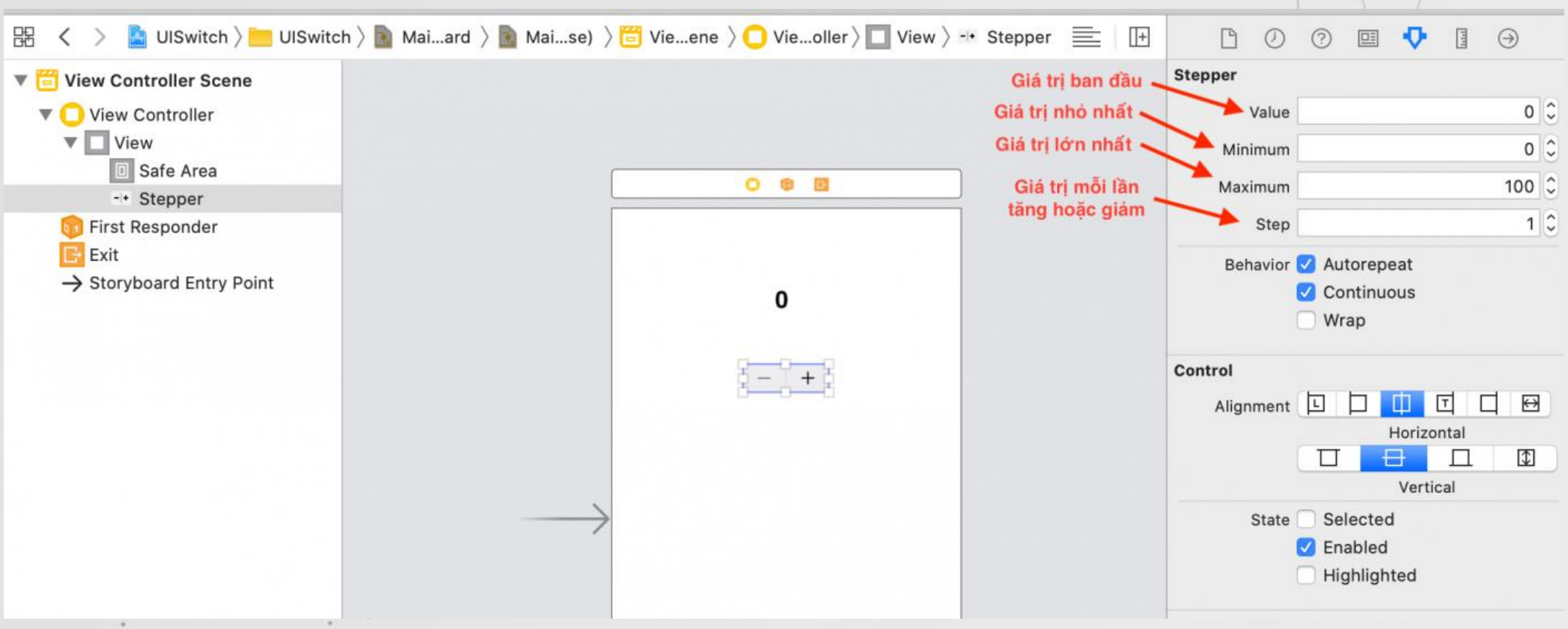
UIStepper

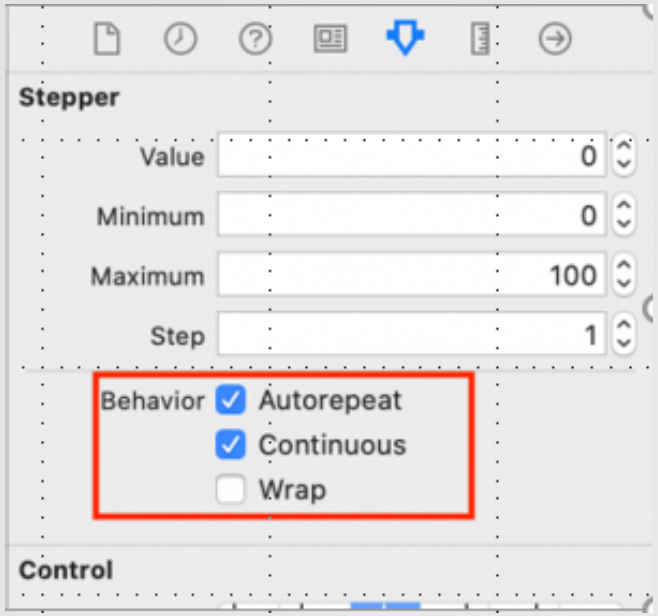
❖ A control used to increment or decrement a value.

❖ If you set stepper behavior to “**autorepeat**” (which is the default), pressing and holding one of its buttons increments or decrements the stepper’s value repeatedly. The rate of change depends on how long the user continues pressing the control.

❖ The maximum value must be greater than or equal to the minimum value. If you set a maximum or minimum value that would break this invariant, both values are set to the new value.

Create UIStepper by drag and drop





UIStepper also has 3 Behavior :

- **Autorepeat**: the value will automatically increase or decrease when the user presses the corresponding Stepper button.
- **Continuous**: the value of Stepper will update continuously, if unchecked, when the user has finished working with Stepper, the new value will be updated.
- **Wrap**: automatically rotates when Stepper value reaches maximum or minimum.

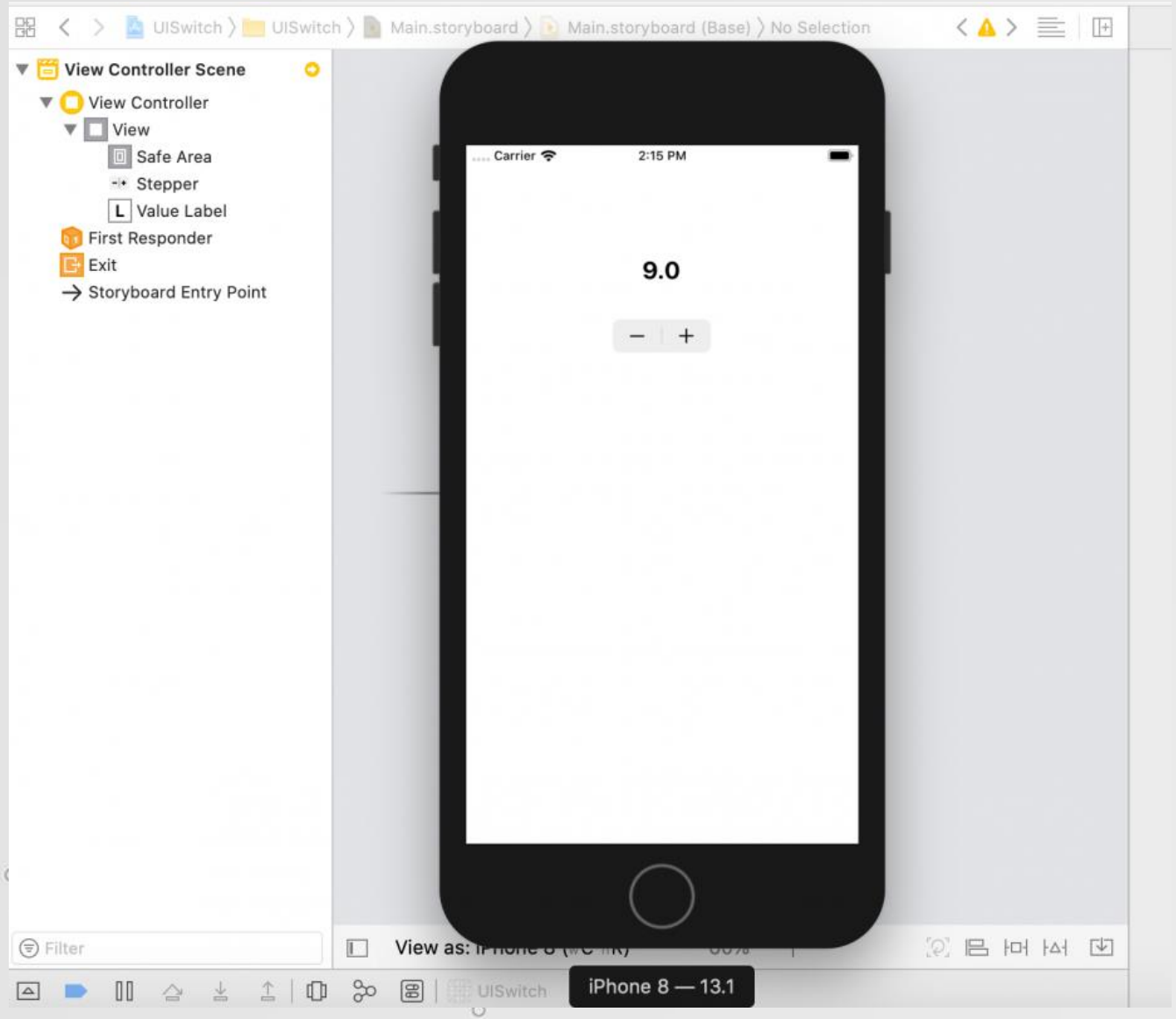
View Controller Scene
- View Controller
  - View
    - Safe Area
    - Stepper
    - 0
- First Responder
- Exit
- Storyboard Entry Point

0

Connection	Action
Object	View Controller
Name	onTapStepper
Type	UIStepper
Event	Value Changed
Arguments	Sender

Cancel Connect

```swift
//
//  ViewController.swift
//  UISwitch
//
//  Created by Kha on 10/17/19.
//  Copyright © 2019 Kha. All rights reserve
//

import UIKit

class ViewController: UIViewController {

    override func viewDidLoad() {
        super.viewDidLoad()

    }
}
```


UISwitch
UISwitch
Main.storyboard
Main.storyboard (Base)
No Selection
View Controller Scene
View Controller
View
Safe Area
Stepper
Value Label
First Responder
Exit
Storyboard Entry Point
Carrier
2:15 PM
9.0
Filter
View as:
UISwitch
iPhone 8 — 13.1

Create UIStepper with code

```
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad()
        {
            super.viewDidLoad()
            // Do any additional setup after loading the view, typically from a nib.
            let myUIStepper = UIStepper (frame:CGRect(x: 10, y: 150, width: 0,
                height: 0))
            // Resume UIStepper value from the beginning
            myUIStepper.wraps = true
            // Position UIStepper in the center of the view
            myUIStepper.center = self.view.center
            // If tap and hold the button, UIStepper value will continuously increment
            myUIStepper.autorepeat = true
            // Set UIStepper max value to 10
            myUIStepper.maximumValue = 10
            // Add a function handler to be called when UIStepper value changes
            myUIStepper.addTarget(self, action:
                #selector(self.stepperValueChanged(_:)), for: .valueChanged)
            self.view.addSubview(myUIStepper)
        }

        @objc func stepperValueChanged(_ sender:UIStepper!)
        {
            print("UIStepper is now \(Int(sender.value))")
        }

}
```

Stepper 〉 iPhone 11 | Running Stepper on iPhone 11

Stepper
- Stepper
  - AppDelegate.swift
  - calendar.png
  - ViewController.swift
  - Main.storyboard
  - Assets.xcassets
  - LaunchScreen.storyboard
  - Info.plist
- Products

ViewController.swift | Main.storyboard

Stepper 〉 Stepper 〉 ViewController.swift 〉 ViewController

```swift
import UIKit

class ViewController: UIViewController {

    override func viewDidLoad()
        {
            super.viewDidLoad()
            // Do any additional setup after loading the view, typically fro
            let myUIStepper = UIStepper (frame:CGRect(x: 10, y: 150, width:
                height: 0))
            // Resume UIStepper value from the beginning
            myUIStepper.wraps = true
            // Position UIStepper in the center of the view
            myUIStepper.center = self.view.center
            // If tap and hold the button, UIStepper value will continuously
```

Stepper

```
UIStepper is now 1
UIStepper is now 2
UIStepper is now 3
UIStepper is now 4
UIStepper is now 5
UIStepper is now 6
UIStepper is now 7
UIStepper is now 8
UIStepper is now 9
UIStepper is now 10
UIStepper is now 0
UIStepper is now 1
UIStepper is now 2
UIStepper is now 3
UIStepper is now 4
UIStepper is now 5
UIStepper is now 6
UIStepper is now 7
UIStepper is now 8
```

iPhone 11 – 14.3

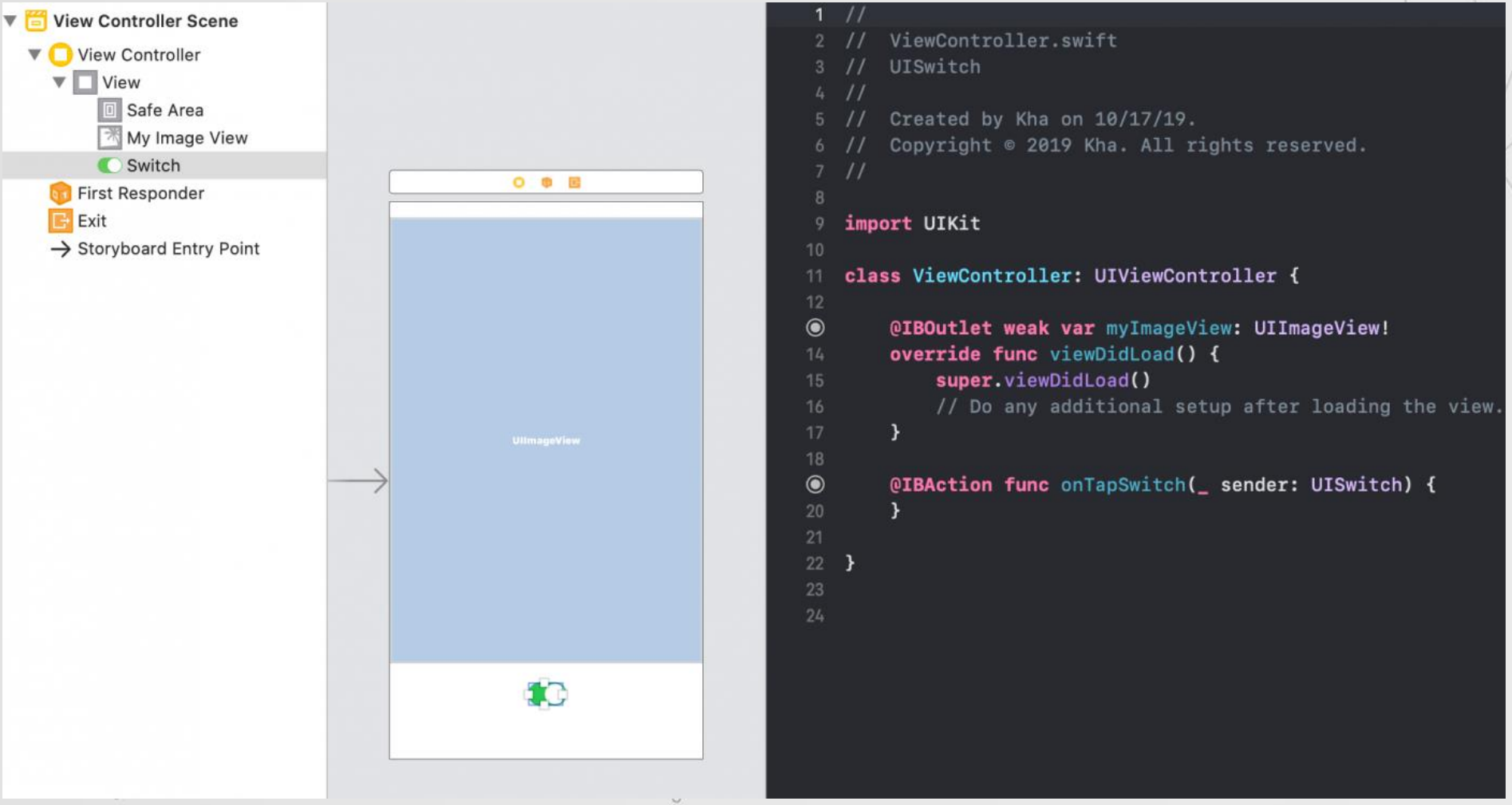
8:08
◂ DatePicker

− +

UISwitch

- A control that offers a binary choice, such as On/Off.
- The UISwitch class declares a property and a method to control its on/off state. As with UISlider, when the user manipulates the switch control (“flips” it) a valueChanged event is generated, which results in the control (if properly configured) sending an action message.
- You can customize the appearance of the switch by changing the color used to tint the switch when it is on or off.

Create UISwitch by drag and drop



```swift
@IBAction func onTapSwitch(_ sender: UISwitch) {
    if sender.isOn {
        myImageView.image = UIImage(named: "1")
    } else {
        myImageView.image = UIImage(named: "2")
    }
}
```

Create UISwitch with code

```
import UIKit
class ViewController: UIViewController {

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

        let switchDemo=UISwitch(frame:CGRect(x: 150, y: 150, width: 0, height: 0))
        switchDemo.addTarget(self, action: #selector(self.switchStateDidChange(_:)),
            for: .valueChanged)
        switchDemo.setOn(true, animated: false)
        self.view.addSubview(switchDemo)

    }

    @objc func switchStateDidChange(_ sender:UISwitch!)
    {
        if (sender.isOn == true){
            print("UISwitch state is now ON")
        }
        else{
            print("UISwitch state is now Off")
        }
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

}
```

UISwitch 〉 iPhone 11 | Running UISwitch on iPhone 11

Assets.xcassets | Main.storyboard | ViewController.swift

UISwitch 〉 UISwitch 〉 ViewController.swift 〉 No Selection

- UISwitch
  - UISwitch
    - 1.jpg
    - 2.jpeg
    - AppDelegate.swift
    - SceneDelegate.swift
    - ViewController.swift
    - Main.storyboard
    - Assets.xcassets
    - LaunchScreen.storyboard
    - Info.plist
  - Products

```swift
import UIKit
class ViewController: UIViewController {

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

        let switchDemo=UISwitch(frame:CGRect(x: 150, y: 150, width: 0, height: 0)
        switchDemo.addTarget(self, action: #selector(self.switchStateDidChange(_:
            for: .valueChanged)
        switchDemo.setOn(true, animated: false)
        self.view.addSubview(switchDemo)
```

UISwitch

```
UISwitch state is now ON
UISwitch state is now Off
UISwitch state is now ON
```

Auto | Filter | All Output | Filter

iPhone 11 – 14.3

8:22

Thanks for Watching