Class 2


TAIWAN
TECH
National Taiwan University of
Science and Technology


HPCLab
High-Performance Computing Laboratory

Interface Control

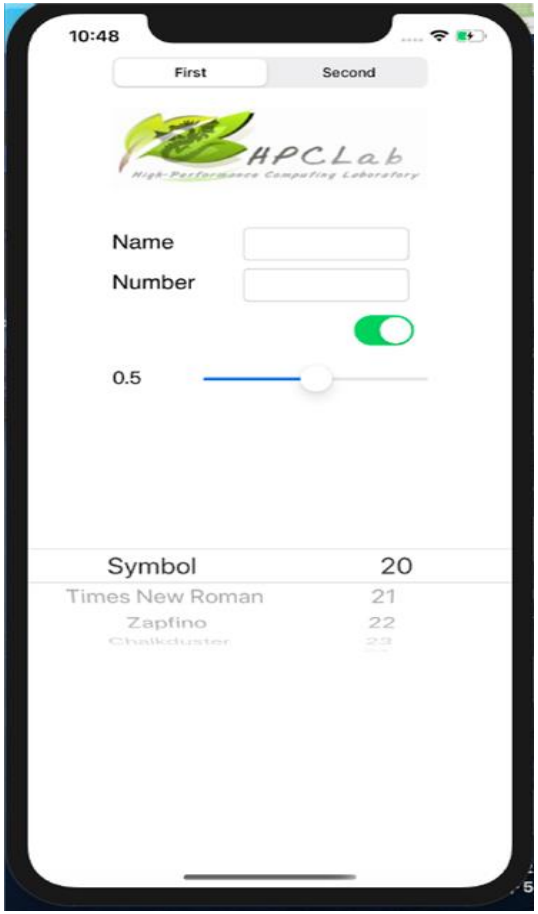


Image View

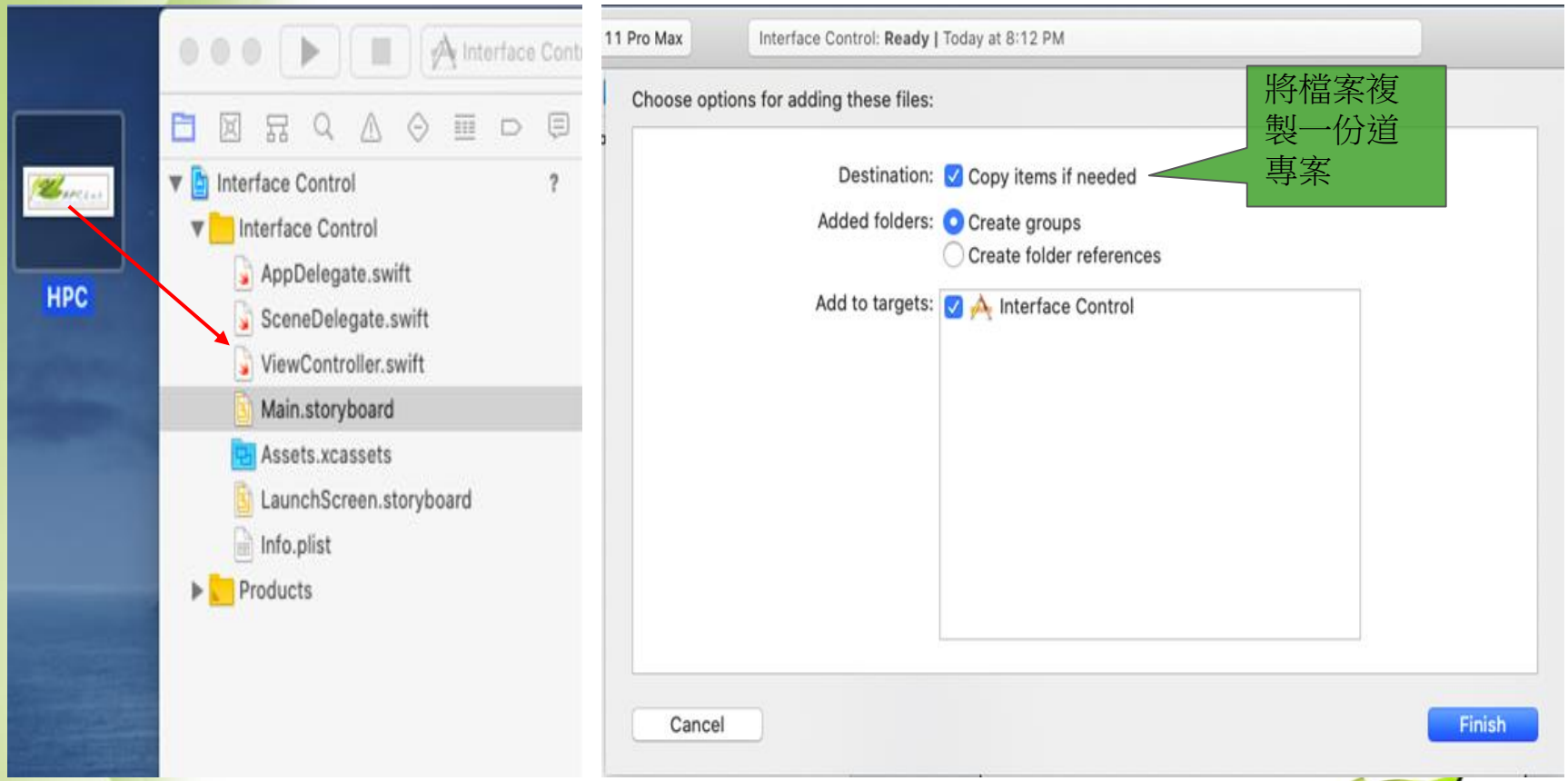

HPC
Interface Control
Interface Control
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
11 Pro Max
Interface Control: Ready | Today at 8:12 PM
Choose options for adding these files:
將檔案複製一份道專案
Destination:
Copy items if needed
Added folders:
Create groups
Create folder references
Add to targets:
Interface Control
Cancel
Finish

Image View

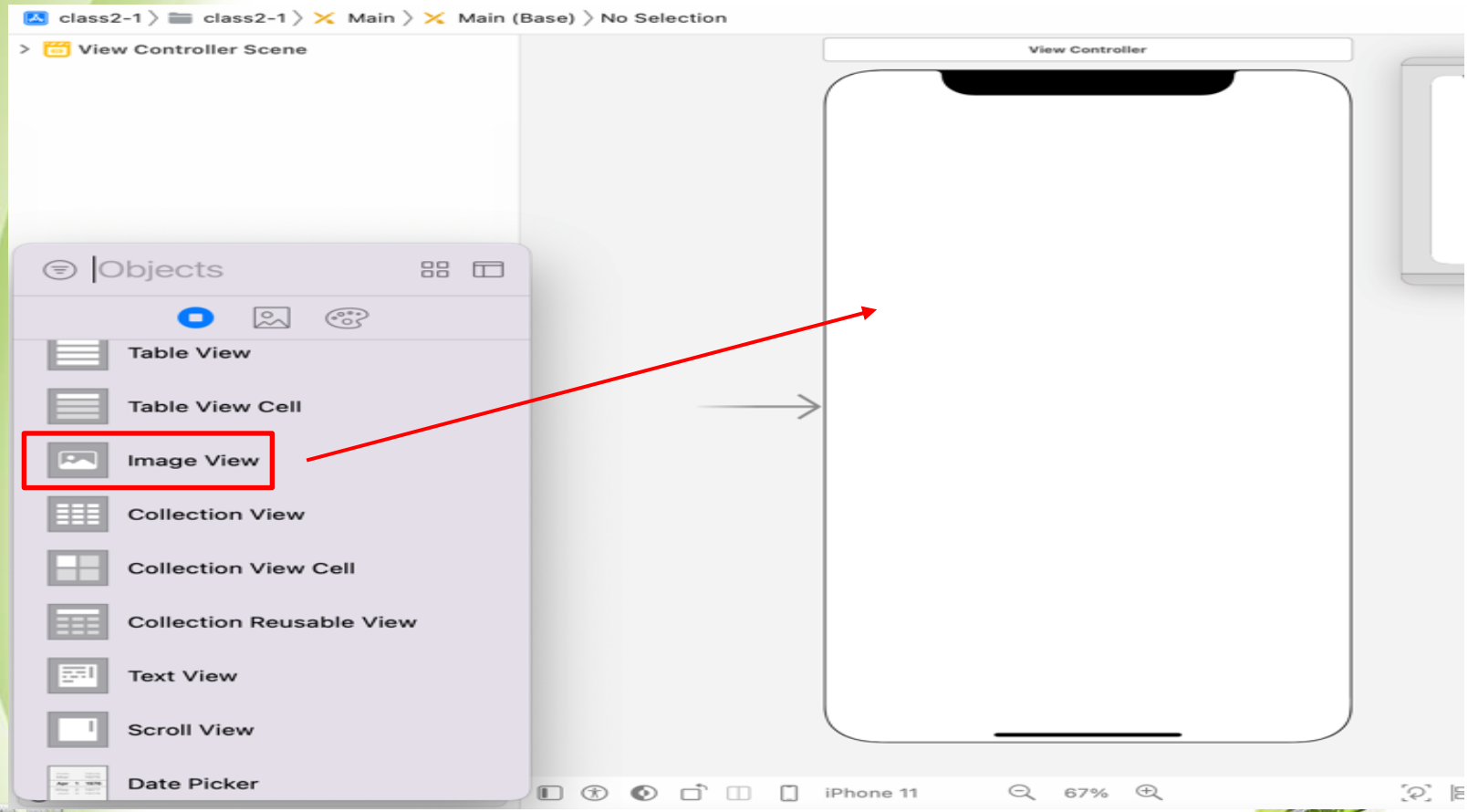

class2-1 > class2-1 > Main > Main (Base) > No Selection
View Controller Scene
View Controller
Objects
Table View
Table View Cell
Image View
Collection View
Collection View Cell
Collection Reusable View
Text View
Scroll View
Date Picker
iPhone 11
67%

Image View

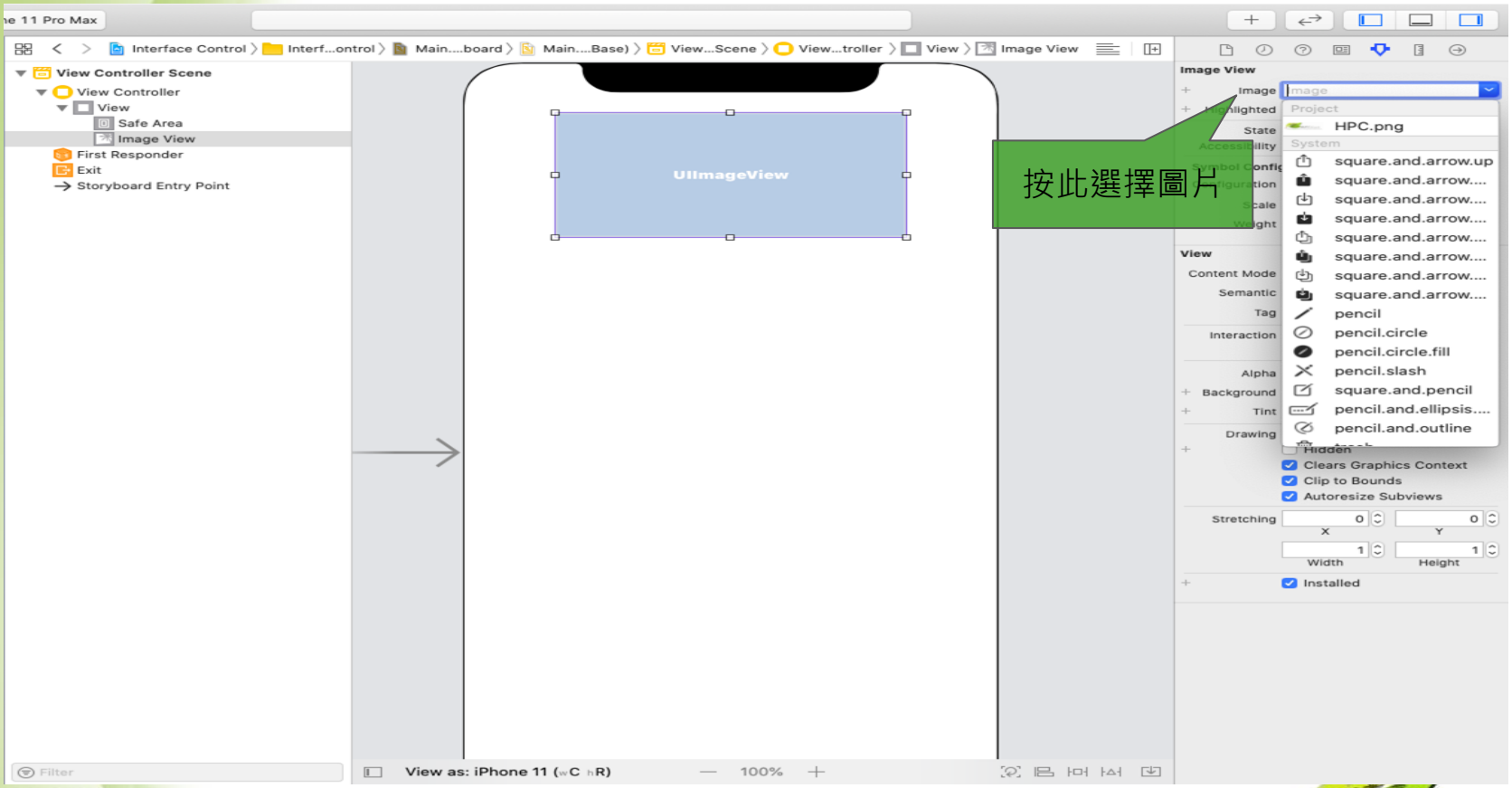

View Controller Scene
View Controller
View
Safe Area
Image View
First Responder
Exit
Storyboard Entry Point
UIImageView
按此選擇圖片
Image View
Image
Project
HPC.png
System
square.and.arrow.up
pencil
pencil.circle
pencil.circle.fill
pencil.slash
square.and.pencil
pencil.and.ellipsis....
pencil.and.outline
Clears Graphics Context
Clip to Bounds
Autoresize Subviews
Stretching
Width
Height
Installed
View as: iPhone 11
100%

Image View

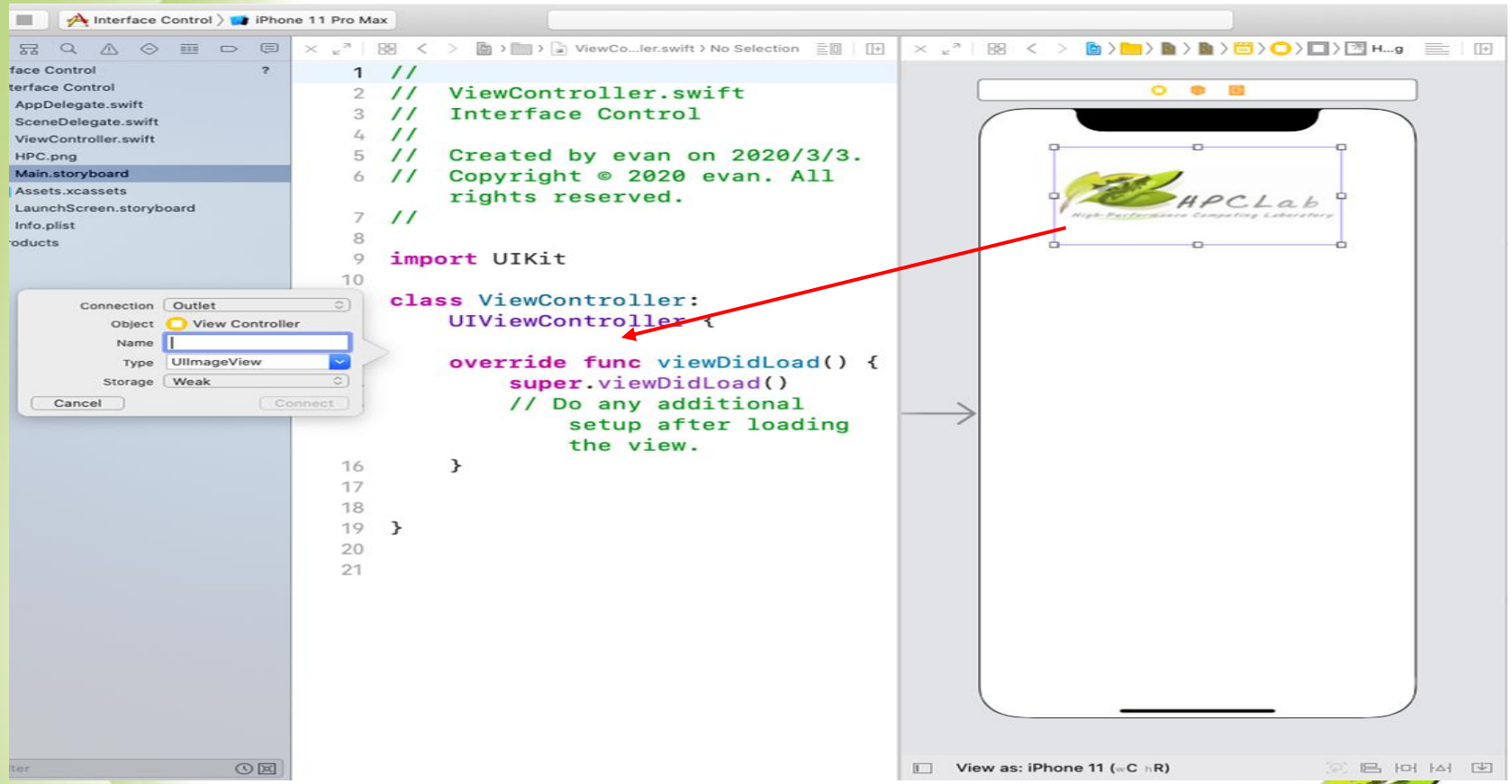

Interface Control
iPhone 11 Pro Max
face Control
terface Control
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
HPC.png
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
oducts
Connection Outlet
Object View Controller
Name
Type UIImageView
Storage Weak
Cancel
Connect
1 //
2 // ViewController.swift
3 // Interface Control
4 //
5 // Created by evan on 2020/3/3.
6 // Copyright © 2020 evan. All rights reserved.
7 //
8
9 import UIKit
10
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
16 }
17
18
19 }
20
21
HPCLab
High-Performance Computing Laboratory
View as: iPhone 11 (wC hR)

Image View

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var HPC: UIImageView!
    override func viewDidLoad() {
        super.viewDidLoad()
        HPC.contentMode = .sc
    }
```

Type 'UIView.ContentMode' has no member...

- scaleToFill
- scaleAspectFit
- scaleAspectFill

scaleToFill: UIView.ContentMode

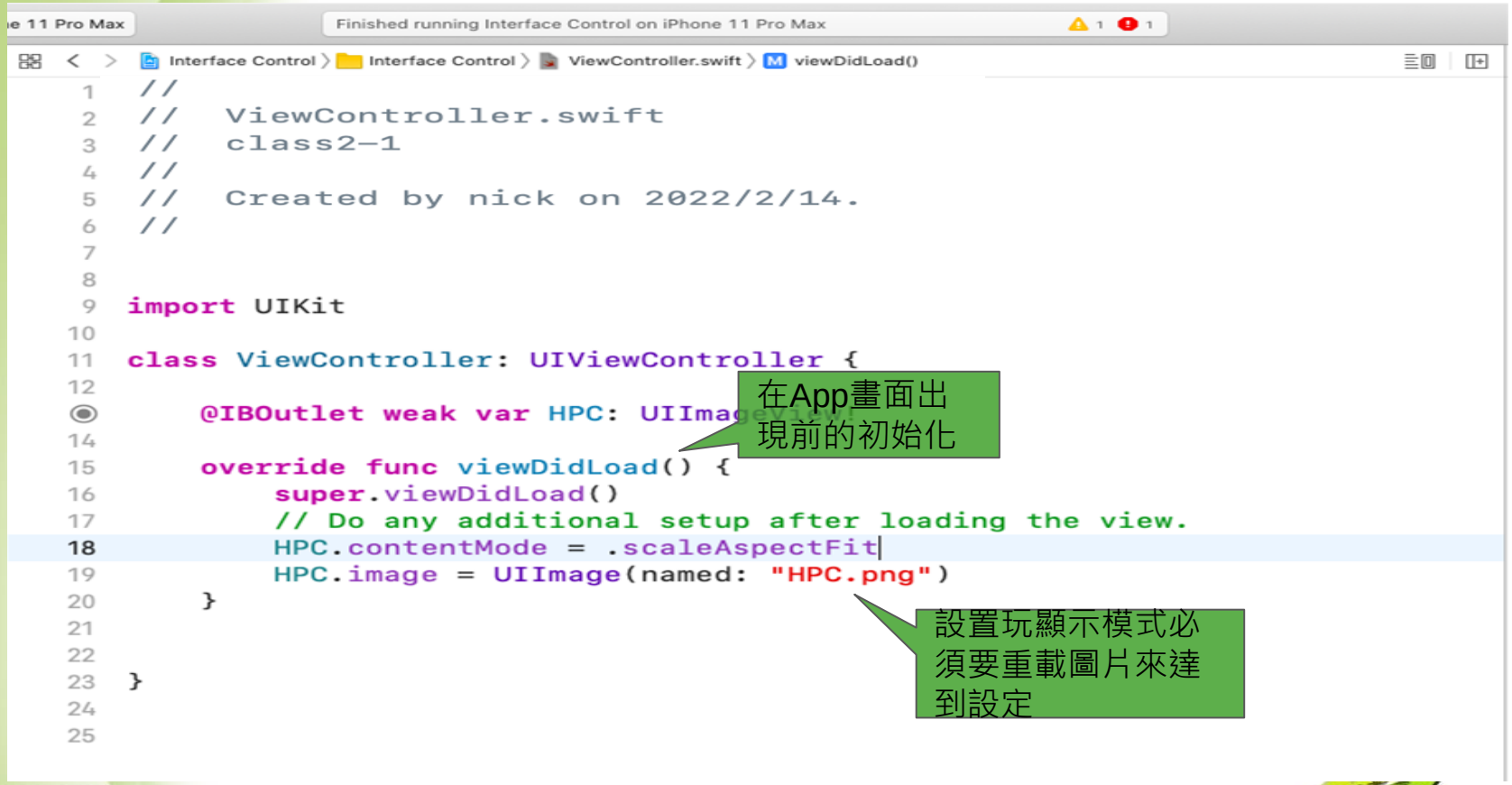
The option to scale the content to fit the size of itself by changing the aspect ratio of the content if necessary.

scale aspect fit：保持長寬比的原則下，放大到其中一邊達到image view的邊緣為止

Scale To Fill：放大圖片讓圖片充滿view

scale aspect fill：保持長寬比的原則下，放大到充滿整個 view

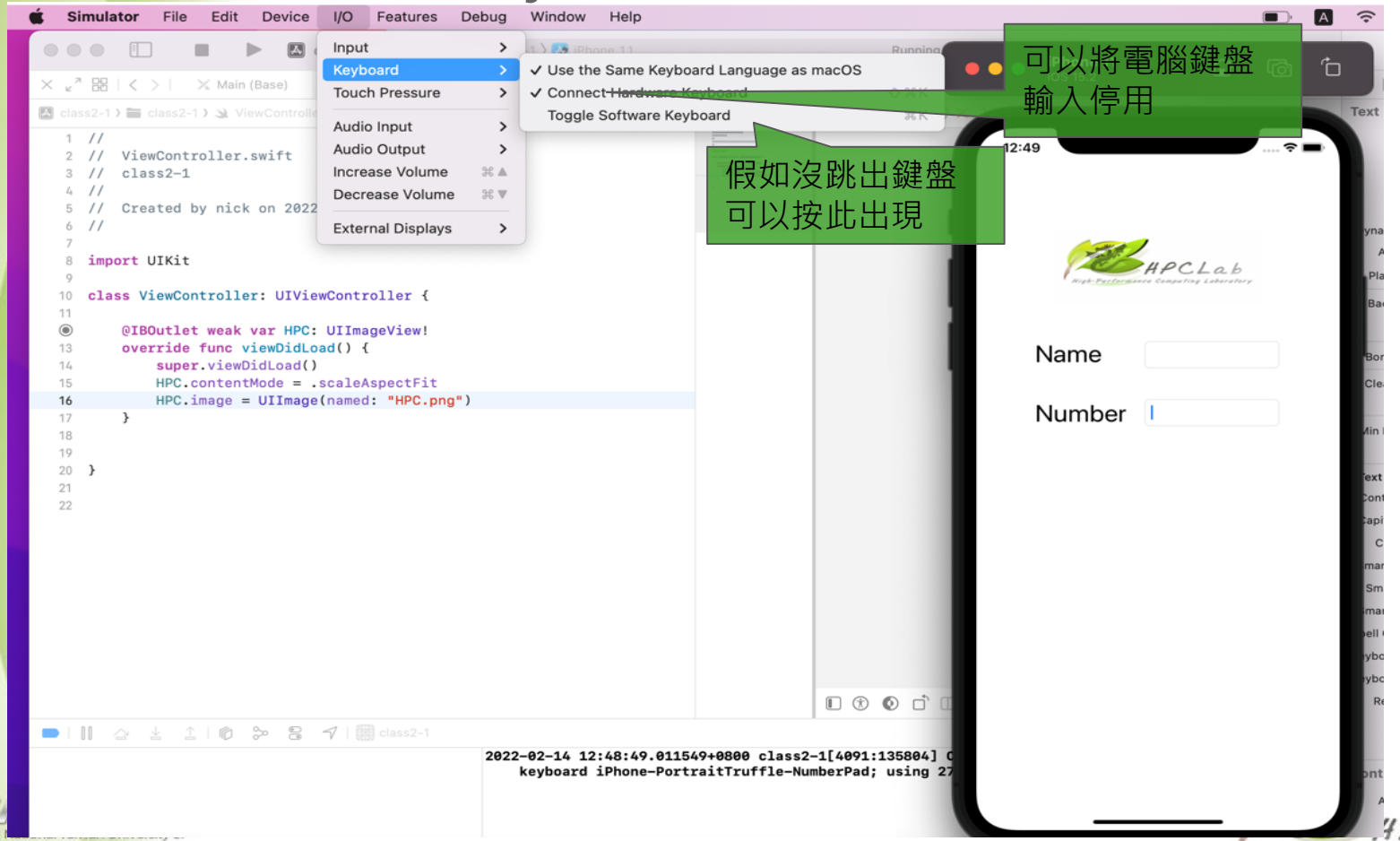
Image View


e 11 Pro Max
Finished running Interface Control on iPhone 11 Pro Max
Interface Control
Interface Control
ViewController.swift
viewDidLoad()
//
// ViewController.swift
// class2-1
//
// Created by nick on 2022/2/14.
//
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var HPC: UIImageView!
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view.
HPC.contentMode = .scaleAspectFit
HPC.image = UIImage(named: "HPC.png")
}
}
在App畫面出現前的初始化
設置玩顯示模式必須要重載圖片來達到設定

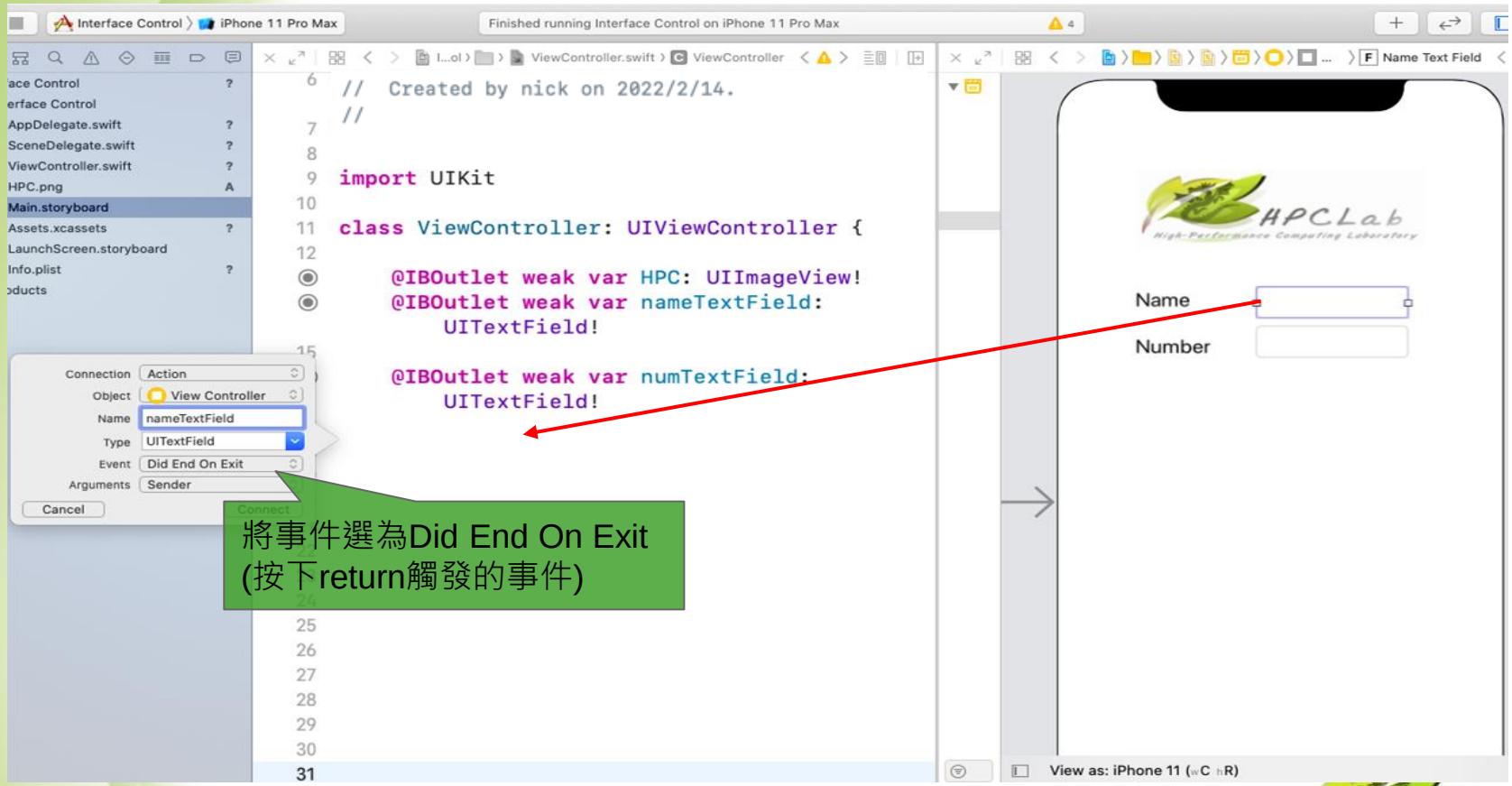
KeyBoard Control

KeyBoard Control

KeyBoard Control



KeyBoard Control



KeyBoard Control

```
8
9  import UIKit
10
11 class ViewController: UIViewController {
12
       @IBOutlet weak var HPC: UIImageView!
       @IBOutlet weak var nameTextField: UITextField!
       @IBOutlet weak var numTextField: UITextField!
16
       @IBAction func nameTextField(_ sender: UITextField) {
18         sender.resignFirstResponder()
19     }
20
21
```

當我們按下nameTextField時他變為FirstResponder因此開啟鍵盤

將TextField取消FirstResponder因此鍵盤會消失

但是數字鍵盤沒有return因此要使用另一種方法將它縮小

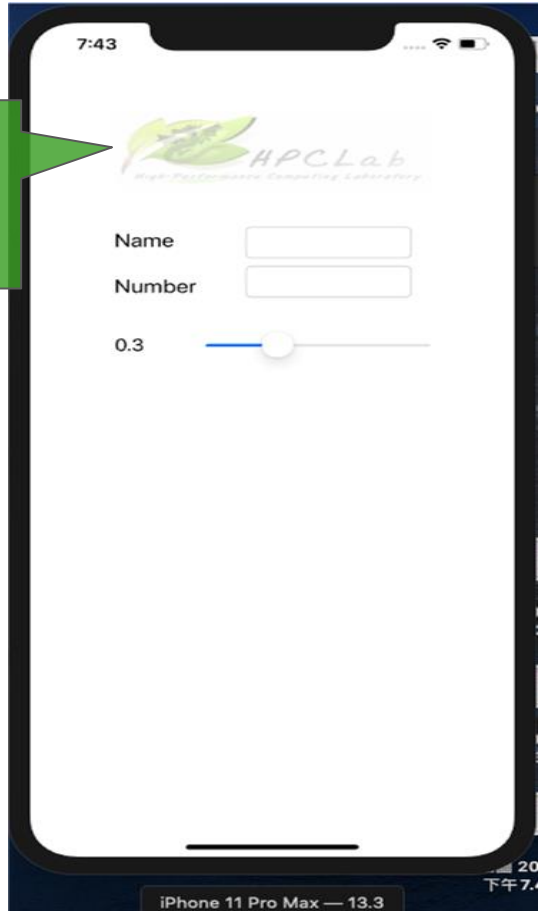
Using Slider

```
override func touchesEnded(_ touches: Set<UITouch>, with event: UIEvent?) {
    numTextField.resignFirstResponder()
}
```

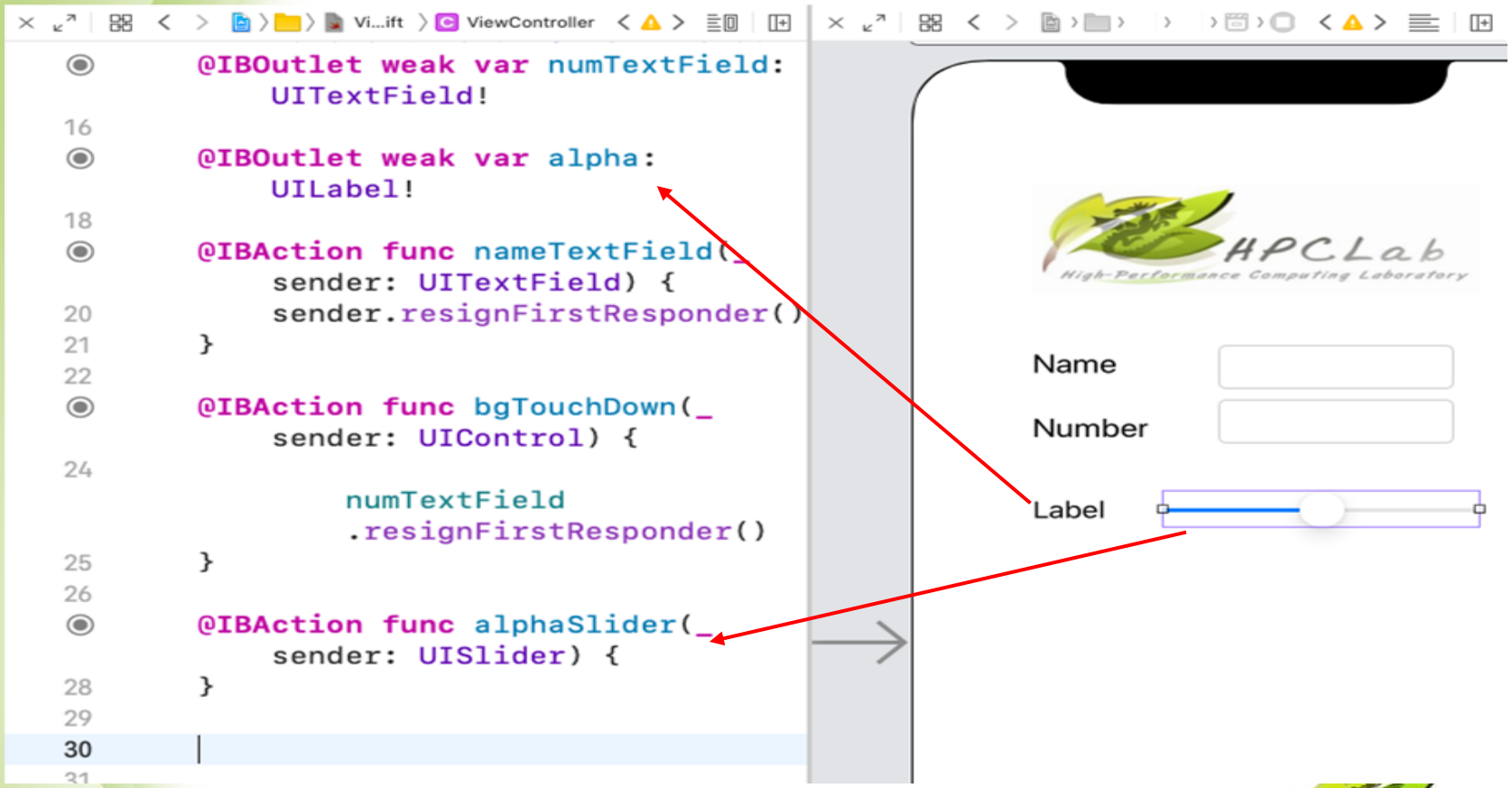
當點擊事件結束時將numTextField取消FirstResponder

Using Slider

調整Slider即可調整圖片透明度



Using Slider



Using Slider

Using Slider

```
12
    @IBOutlet weak var HPC: UIImageView!
    @IBOutlet weak var nameTextField: UITextField!
    @IBOutlet weak var numTextField: UITextField!
16
    @IBOutlet weak var alpha: UILabel!
18
    @IBAction func nameTextField(_ sender: UITextField) {
20      sender.resignFirstResponder()
21  }
22
    @IBAction func bgTouchDown(_ sender: UIControl) {
24      numTextField.resignFirstResponder()
25  }
26
    @IBAction func alphaSlider(_ sender: UISlider) {
28      alpha.text = String(sender.value)
29  }
30
```

當滑動Slider時將Label的值改變

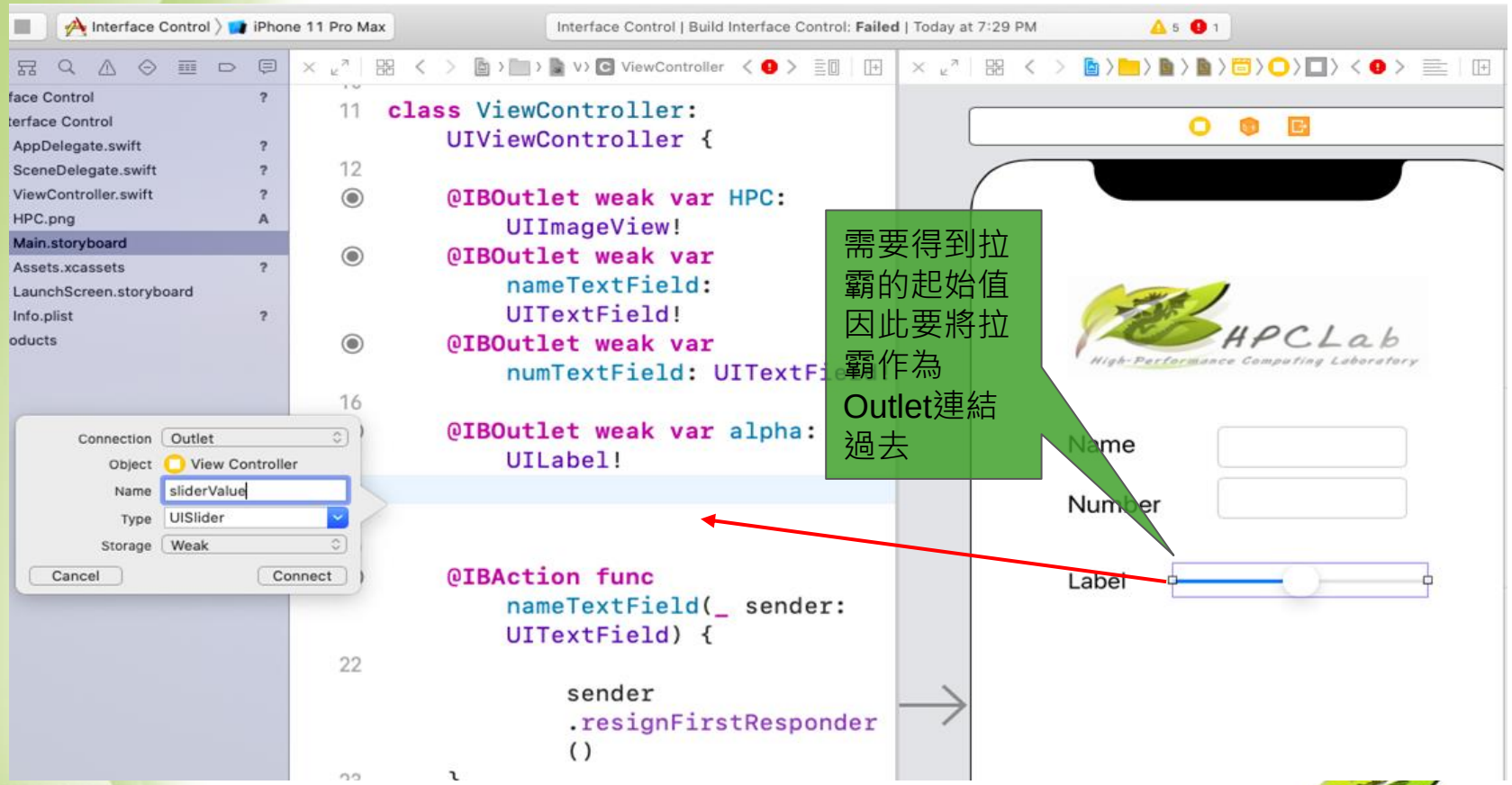
Using Slider

Label初值並未設定因此並不會準確顯示拉霸一開始的值

當調整拉霸時數字會改變

接著要將顯示數字改為顯示小數第一位而已

Using Slider



Using Slider

```
@IBAction func alphaSlider(_ sender: UISlider) {
    alpha.text = String(format: "%.1f",sender.value)
    HPC.alpha = CGFloat(sender.value)
    HPC.image = UIImage(named: "HPC.png")
}
```

將Label顯示的數字設為小數後一位

圖片的alpha值為CGFloat型態因此要將型態轉為CGFloat

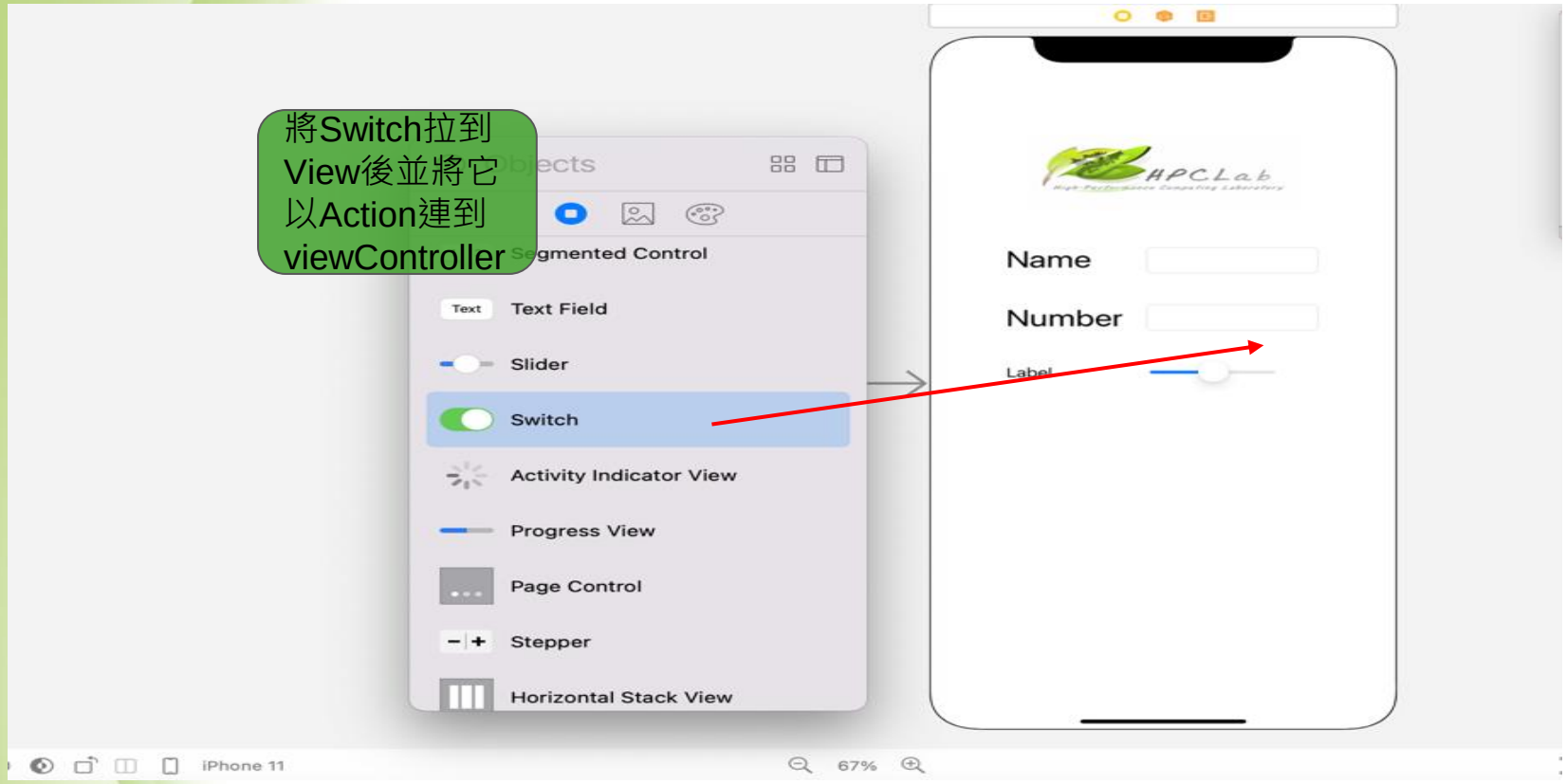
設定alpha值後要再重設圖片來刷新

```
35      override func viewDidLoad() {
36          super.viewDidLoad()
37          // Do any additional setup after loading th
38          HPC.contentMode = .scaleAspectFit
39          HPC.image = UIImage(named: "HPC.png")
40          alpha.text = String(format: "%.1f",sliderValue.value)
41      }
42  }
```

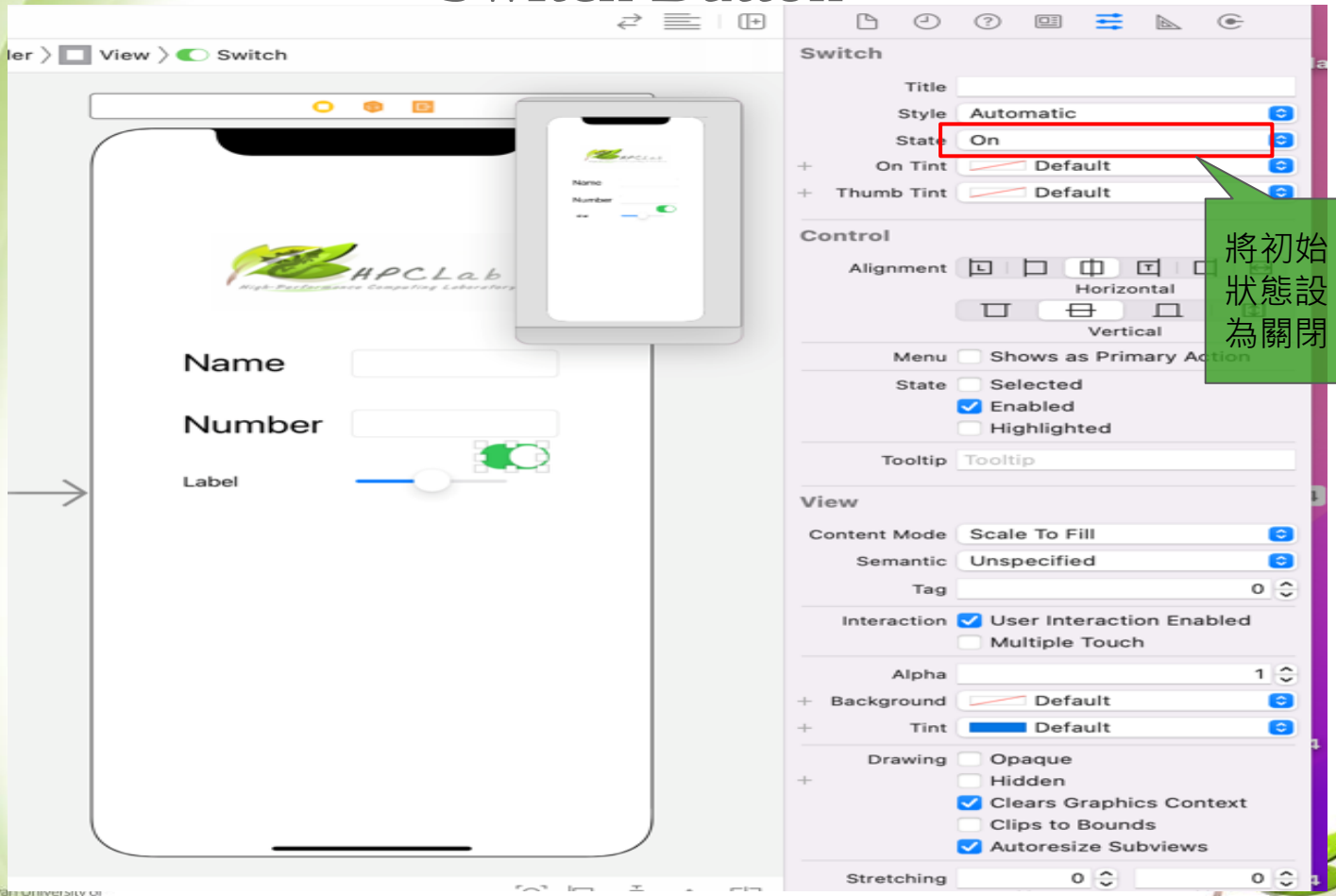
在viewDidLoad中設定Label的初值

Switch Button

Switch Button



Switch Button



Switch Button

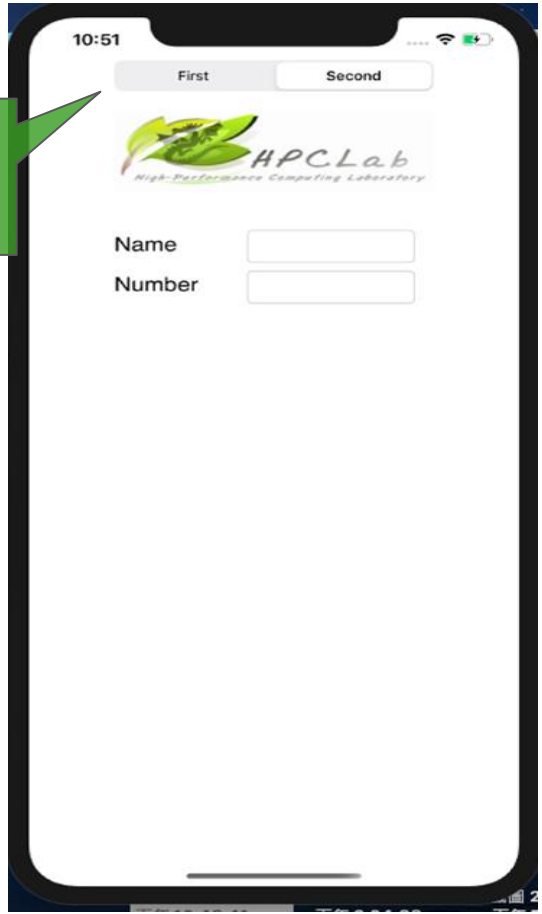
```
@IBAction func secureText(_ sender: UISwitch) {
    if sender.isOn{
        numTextField.isSecureTextEntry = true
    } else {
        numTextField.isSecureTextEntry = false
    }
}
```

當按鈕設定為開時將數字隱藏

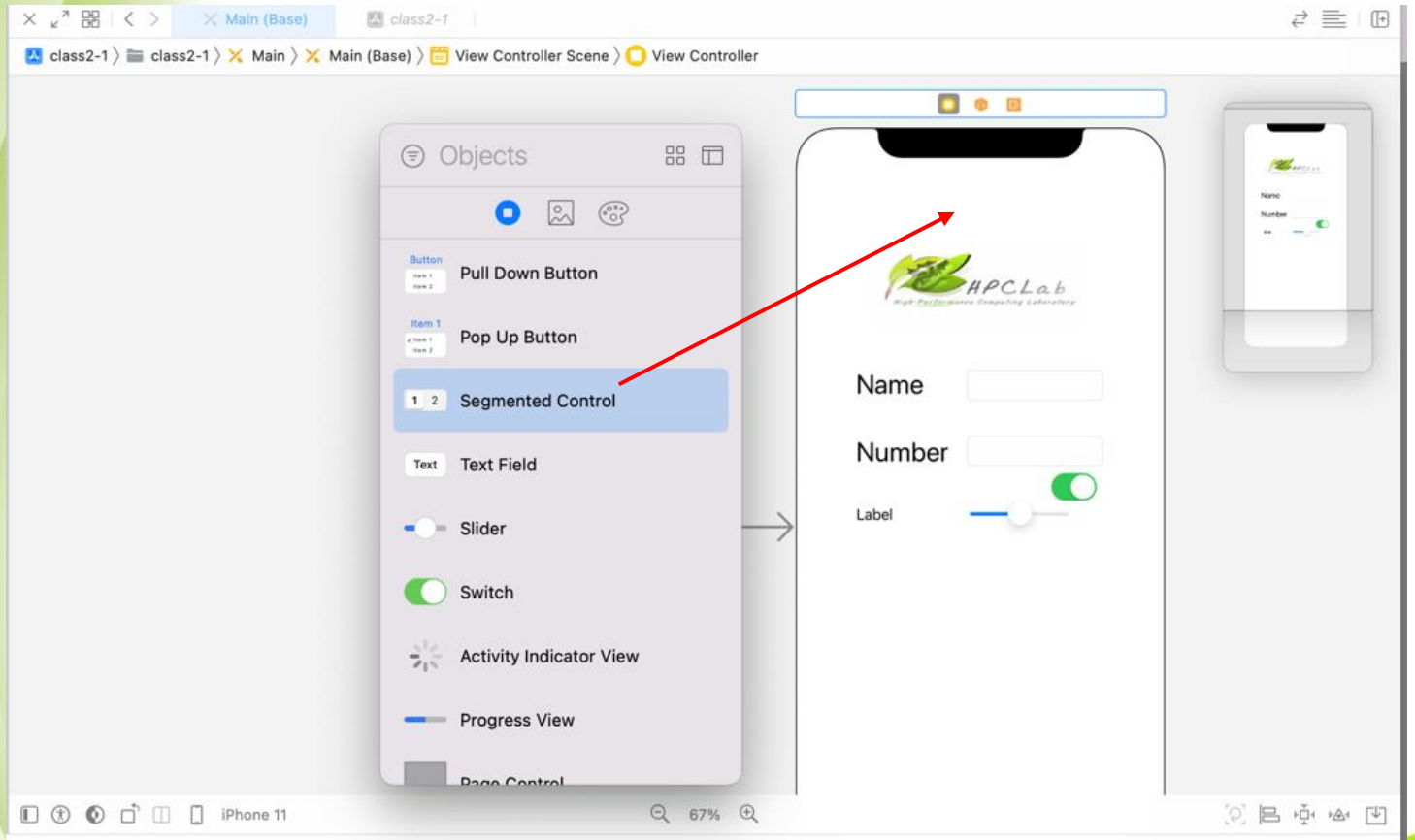
當按鈕設定為關時顯示數字

Segmented Control

切換Segmented來達到顯示或隱藏其他元件



Segmented Control

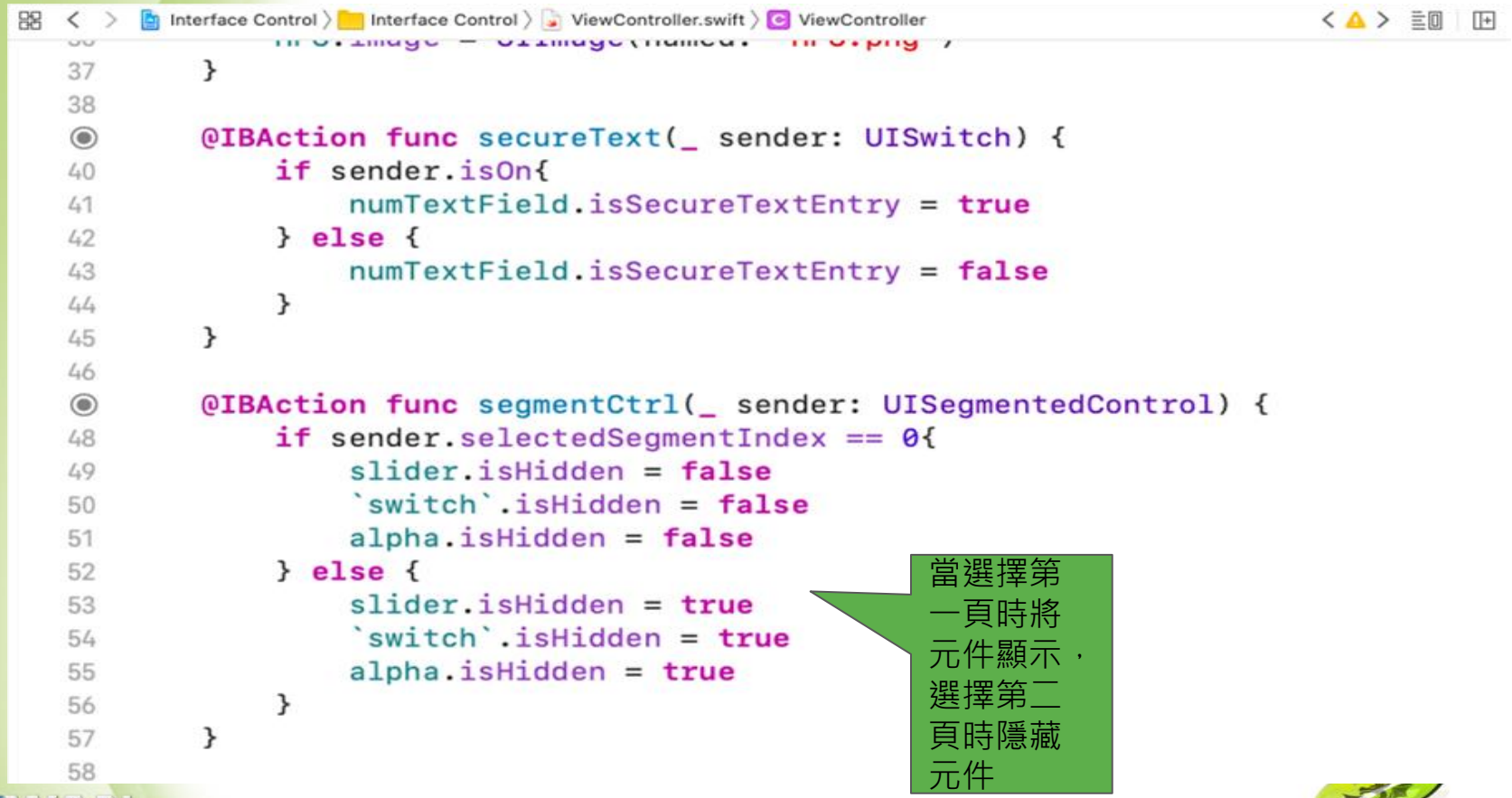


Segmented Control

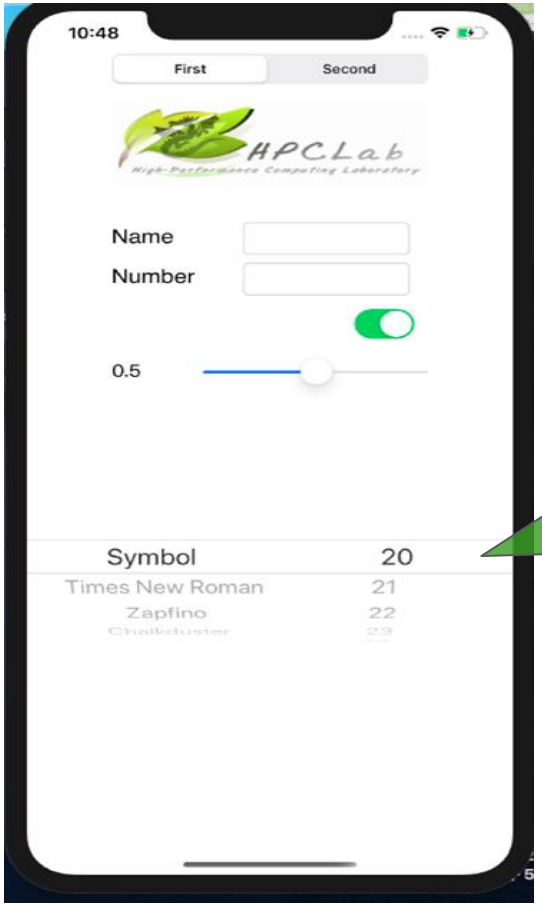
```
@IBOutlet weak var HPC: UIImageView!
@IBOutlet weak var numberTextField: UITextField!
@IBOutlet weak var nameTextField: UITextField!
@IBOutlet weak var alpha: UILabel!
@IBOutlet weak var sliderValue: UISlider!
@IBOutlet weak var `switch`: UISwitch!
@IBOutlet weak var slider: UISlider!
```

宣告的
Outlet元件

Segmented Control

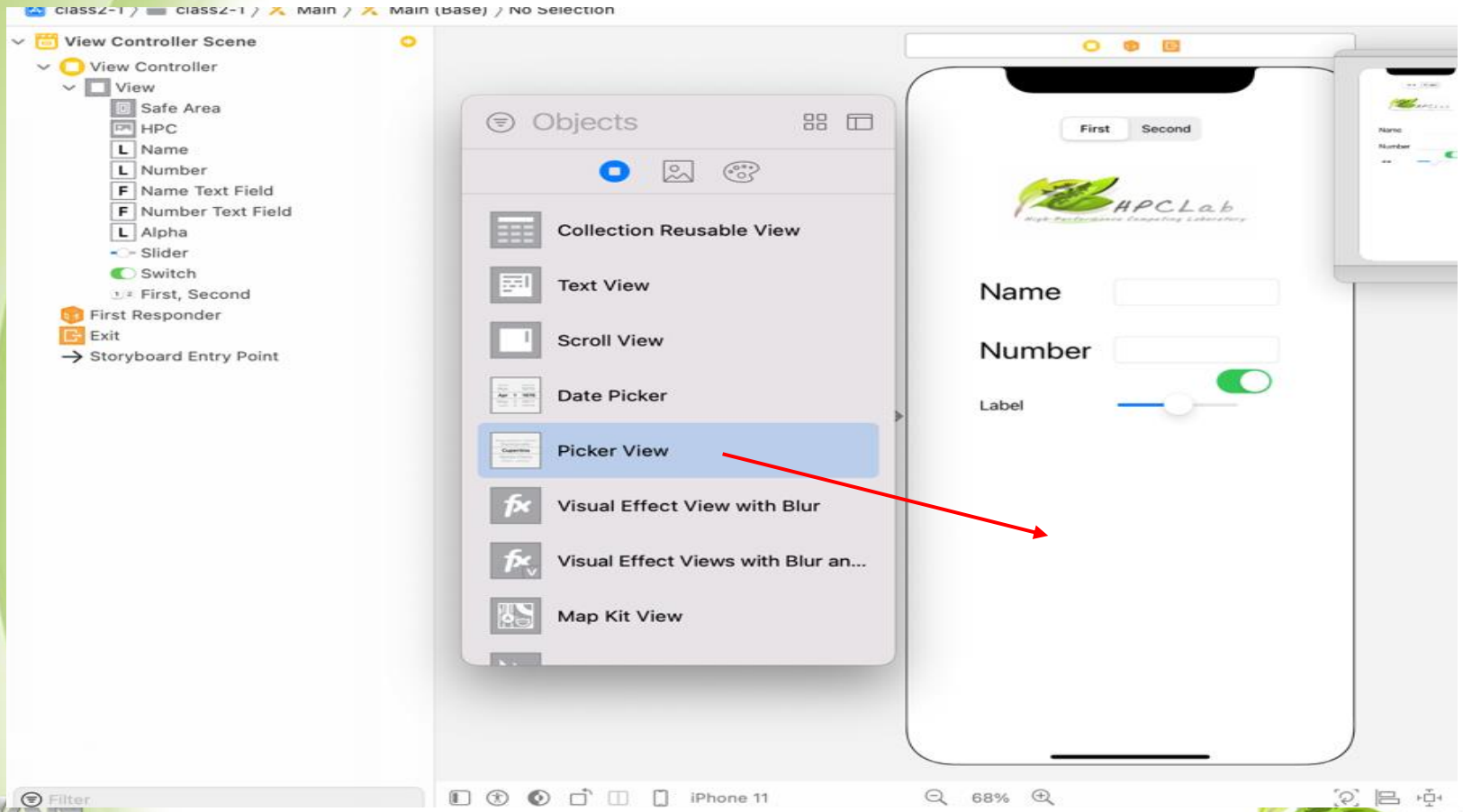


Picker View

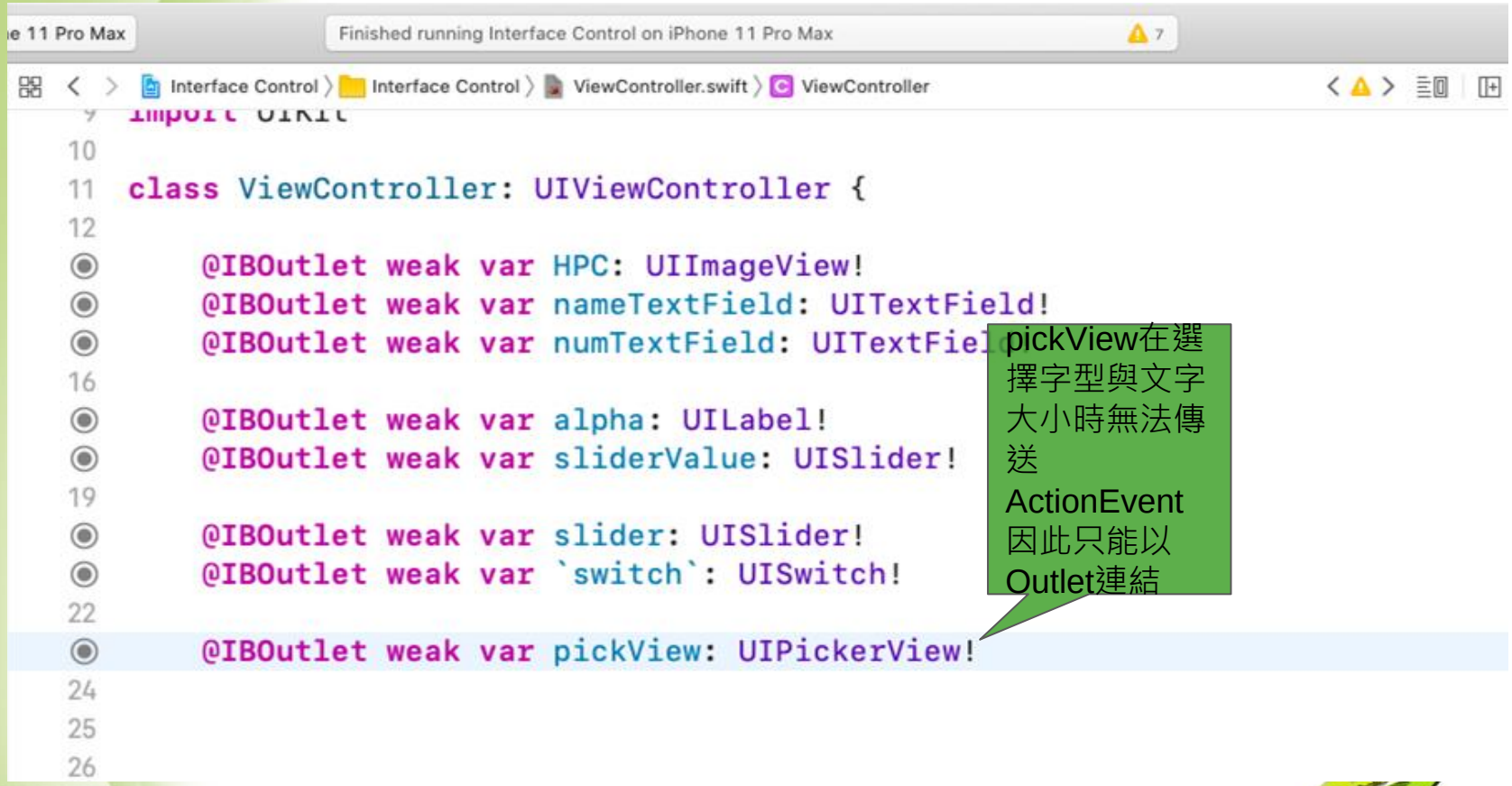


左邊選擇字型右邊選擇文字大小

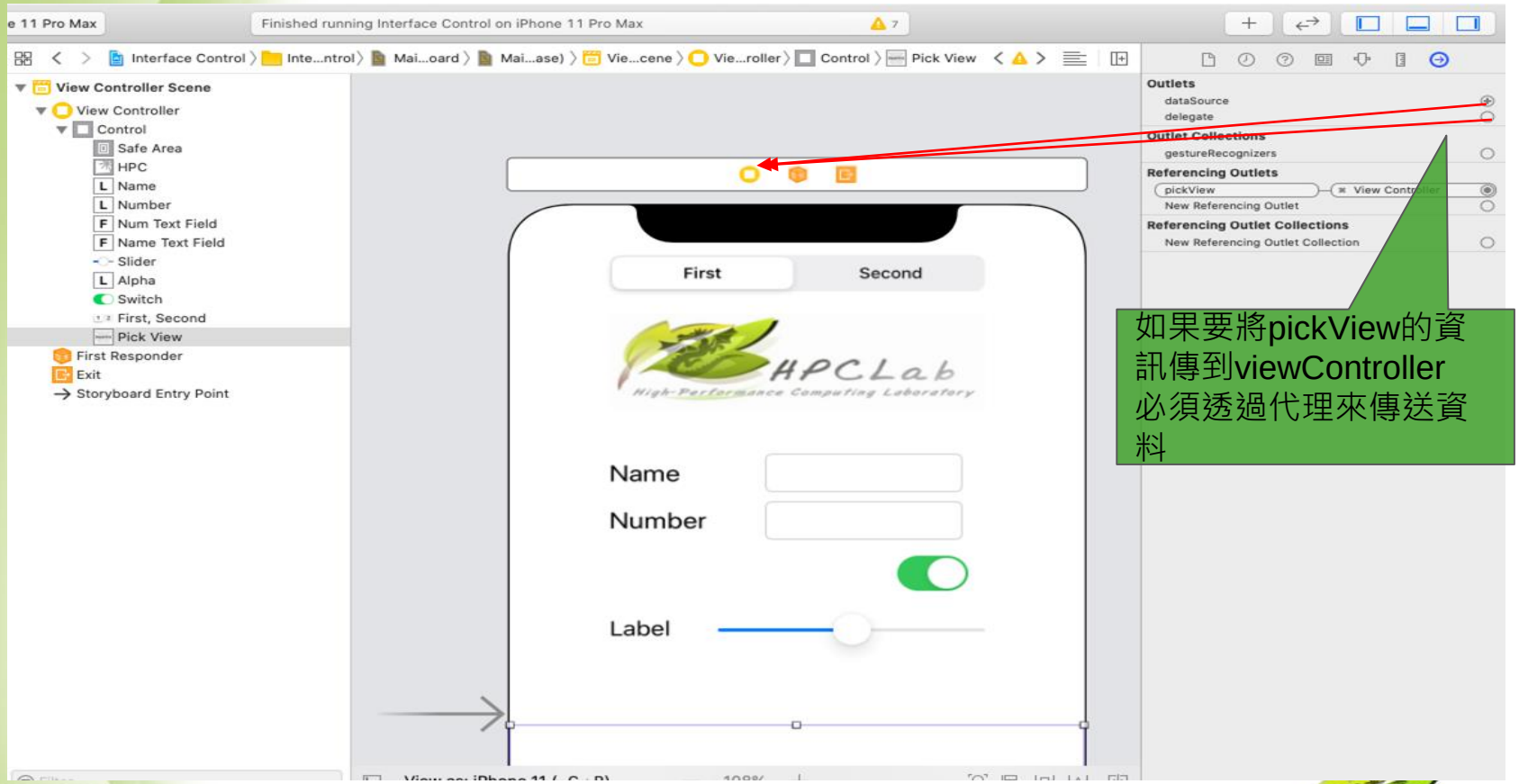
Picker View


View Controller Scene
View Controller
View
Safe Area
HPC
Name
Number
Name Text Field
Number Text Field
Alpha
Slider
Switch
First, Second
First Responder
Exit
Storyboard Entry Point
Objects
Collection Reusable View
Text View
Scroll View
Date Picker
Picker View
Visual Effect View with Blur
Visual Effect Views with Blur an...
Map Kit View
First
Second
Name
Number
Label
iPhone 11
68%

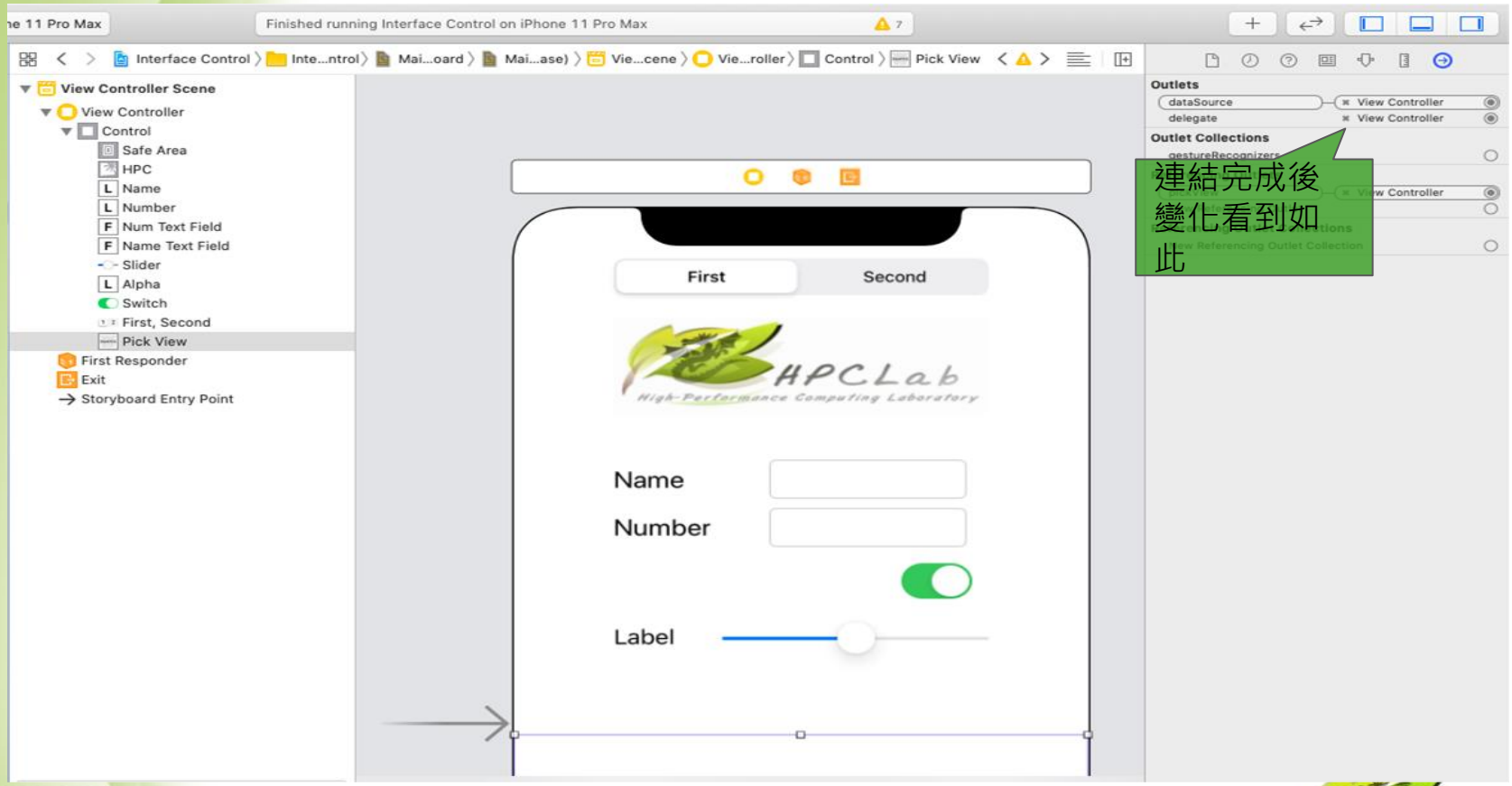
Picker View


e 11 Pro Max
Finished running Interface Control on iPhone 11 Pro Max
Interface Control
Interface Control
ViewController.swift
ViewController
import UIKit
class ViewController: UIViewController {
@IBOutlet weak var HPC: UIImageView!
@IBOutlet weak var nameTextField: UITextField!
@IBOutlet weak var numTextFiel
@IBOutlet weak var alpha: UILabel!
@IBOutlet weak var sliderValue: UISlider!
@IBOutlet weak var slider: UISlider!
@IBOutlet weak var `switch`: UISwitch!
@IBOutlet weak var pickView: UIPickerView!
pickView在選擇字型與文字大小時無法傳送ActionEvent因此只能以Outlet連結

Picker View


Finished running Interface Control on iPhone 11 Pro Max
Interface Control
Pick View
View Controller Scene
View Controller
Control
Safe Area
HPC
Name
Number
Num Text Field
Name Text Field
Slider
Alpha
Switch
First, Second
Pick View
First Responder
Exit
Storyboard Entry Point
First
Second
HPCLab
High-Performance Computing Laboratory
Name
Number
Label
Outlets
dataSource
delegate
Outlet Collections
gestureRecognizers
Referencing Outlets
pickView
New Referencing Outlet
Referencing Outlet Collections
New Referencing Outlet Collection
如果要將pickView的資訊傳到viewController必須透過代理來傳送資料

Picker View


Finished running Interface Control on iPhone 11 Pro Max
View Controller Scene
View Controller
Control
Safe Area
HPC
Name
Number
Num Text Field
Name Text Field
Slider
Alpha
Switch
First, Second
Pick View
First Responder
Exit
Storyboard Entry Point
First
Second
HPCLab
High-Performance Computing Laboratory
Name
Number
Label
Outlets
dataSource
delegate
View Controller
Outlet Collections
連結完成後變化看到如此

Picker View

class2-1 〉 class2-1 〉 ViewController 〉 ViewController

```
1  //
2  //  ViewController.swift
3  //  class2-1
4  //
5  //  Created by nick on 2022/2/14.
6  //
7
8  import UIKit
9
10 class ViewController: UIViewController,UIPickerViewDataSource,UIPickerViewDelegate {
11
       @IBOutlet weak var HPC: UIImageView!
       @IBOutlet weak var numberTextField: UITextFiel
       @IBOutlet weak var nameTextField: UITextField!
       @IBOutlet weak var alpha: UILabel!
       @IBOutlet weak var sliderValue: UISlider!
       @IBOutlet weak var `switch`: UISwitch!
       @IBOutlet weak var slider: UISlider!
19
       @IBOutlet weak var pickview: UIPickerView!
21
       @IBAction func nameTextField(_ sender: UITextField) {
23         sender.resignFirstResponder()
24     }
```

viewController 必須要繼承 pickerView的協定才能與 pickerView連結

Type 'ViewController' does not conform to protocol 'UIPickerViewDataSource'

Do you want to add protocol stubs? Fix

當我們繼承完時會要求必須要實作協定的某些函式，我們可以按下Fix來看到要實作哪些函式

Picker View

class2-1 〉 class2-1 〉 ViewController 〉 ViewController

```
1  //
2  //  ViewController.swift
3  //  class2-1
4  //
5  //  Created by nick on 2022/2/14.
6  //
7
8  import UIKit
9
10 class ViewController: UIViewController,UIPickerViewDataSource,UIPickerViewDelegate {
11     func numberOfComponents(in pickerView: UIPickerView) -> Int {
12         code
13     }
14
15     func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
16         code
17     }
18
19
20     @IBOutlet weak var imageVC: UIImageView!
21     @IBOutlet weak var numberTextField: UITextField!
22     @IBOutlet weak var nameTextField: UITextField!
23     @IBOutlet weak var alpha: UILabel!
24     @IBOutlet weak var sliderValue: UISlider!
25     @IBOutlet weak var `switch`: UISwitch!
26     @IBOutlet weak var slider: UISlider!
27
28     @IBOutlet weak var pickview: UIPickerView!
29
30     @IBAction func nameTextField(_ sender: UITextField) {
31         sender.resignFirstResponder()
32     }
```

上面的函式代表有多少欄位

下面的函式是決定每一欄有幾個選項

Picker View

```
9  import UIKit
10
11 class ViewController: UIViewController,UIPickerViewDataSource,UIPickerViewDelegate {
12
13     let fontName = ["Symbol","Times New Roman","Zapfino","Chalkduster"]
14     let fontSize = ["20","21","22","23","24","25","26"]
15     var currentSize:CGFloat = 20.0
16
17     func numberOfComponents(in pickerView: UIPickerView) -> Int {
18         return 2
19     }
20
21     func pickerView(_ pickerView: UIPickerView, numberOfRowsInComponent component: Int) -> Int {
22         if(component == 0){
23             return fontName.count
24         }
25         return fontSize.count
26     }
27
28
```

先將每欄的選項以字串陣列宣告出來

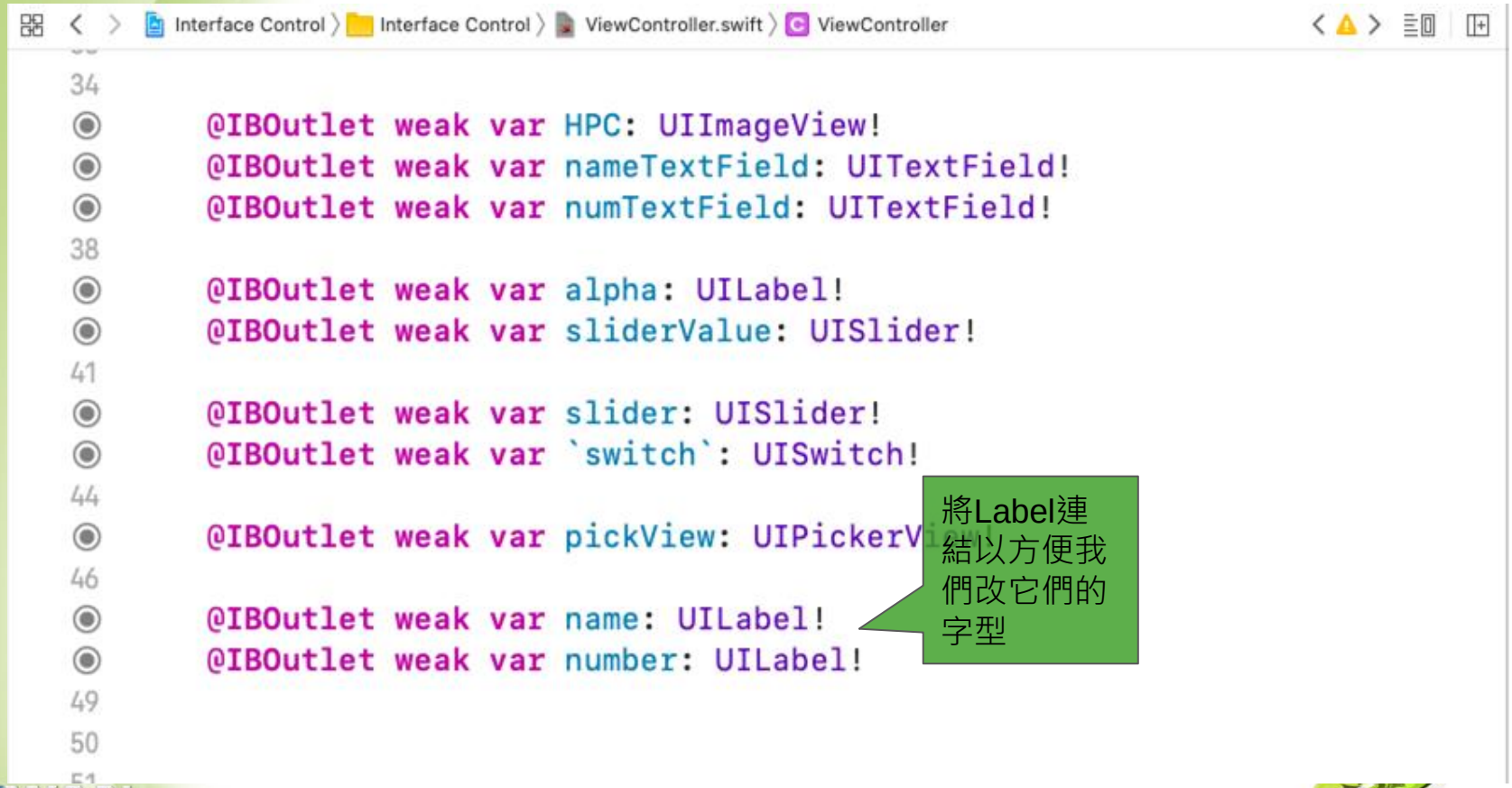
回傳2代表有兩欄的選項

component用來決定是哪一欄要回傳多少選項

如果是第一欄便回傳字型陣列有幾個元素

count來統計陣列有多少元素

Picker View


Interface Control
Interface Control
ViewController.swift
ViewController
34
@IBOutlet weak var HPC: UIImageView!
@IBOutlet weak var nameTextField: UITextField!
@IBOutlet weak var numTextField: UITextField!
38
@IBOutlet weak var alpha: UILabel!
@IBOutlet weak var sliderValue: UISlider!
41
@IBOutlet weak var slider: UISlider!
@IBOutlet weak var `switch`: UISwitch!
44
@IBOutlet weak var pickView: UIPickerView!
46
@IBOutlet weak var name: UILabel!
@IBOutlet weak var number: UILabel!
49
50
將Label連結以方便我們改它們的字型

Picker View

```
func pickerView(_ pickerView: UIPickerView, titleForRow row: Int, forComponent component:
    Int) -> String? {
    if(component == 0){
        return fontName[row]
    }
    return fontSize[row]
}
```

第一列回字型的陣列，row是目前選到選項的index

接著我們使用剛剛協定中的其他函式，這個函式用來顯示每個欄位的選項

Picker View

```
func pickerView(_ pickerView: UIPickerView, didSelectRow row: Int, inComponent component:
    Int) {
    if component == 0 {
        name.font = UIFont(name:fontName[row],size:currentSize)
        number.font = UIFont(name:fontName[row],size:currentSize)
    } else {
        currentSize = CGFloat(Double(fontSize[row])!)
        name.font = name.font.withSize(CGFloat(Double(fontSize[row])!))
        number.font = number.font.withSize(CGFloat(Double(fontSize[row])!))
    }
}
```

請自行在viewController宣告currentSize此變數

當更改第一欄的選項時將字型修改，因為不能只設定字型因此用一個變數來記錄文字大小

這個函示用來決定當我們選到該選項要做的對應事情

更改第二欄選項時將字型大小修改並記錄修改完的大小

字串要轉CGFloat要先轉為Double