


Class 2



Interface Control

10:48

First Second

 HPCLab
High-Performance Computing Laboratory

Name

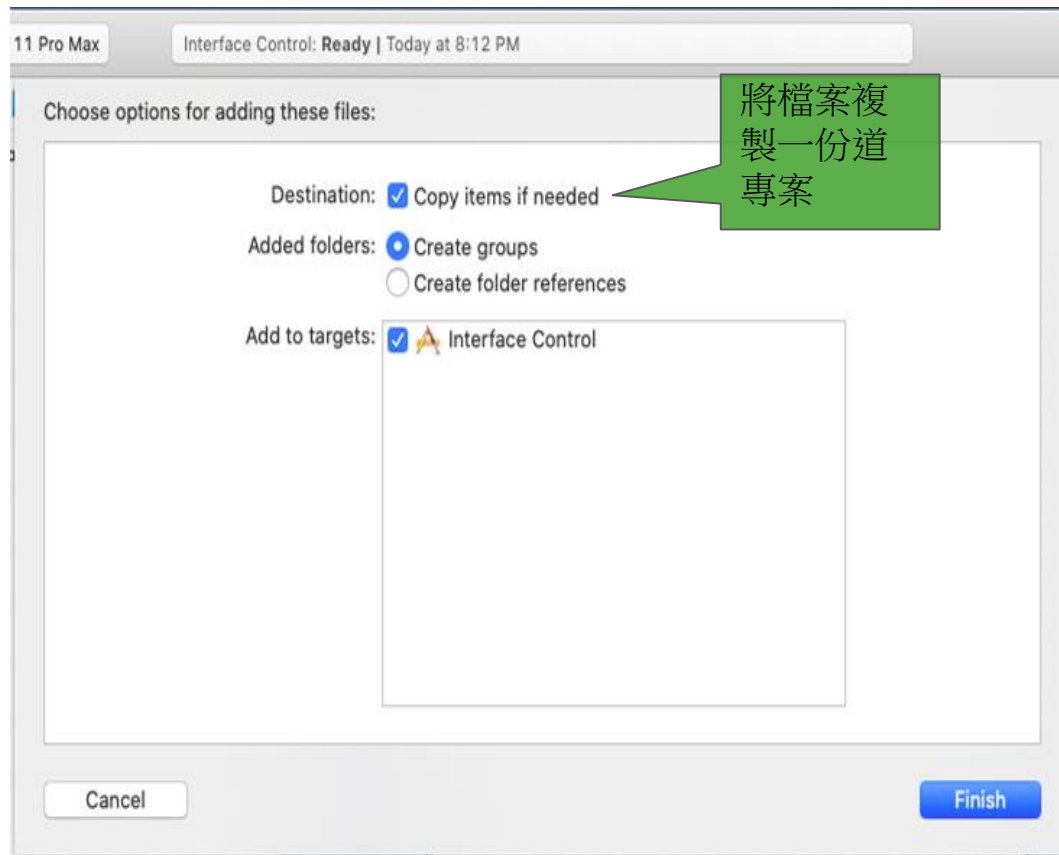
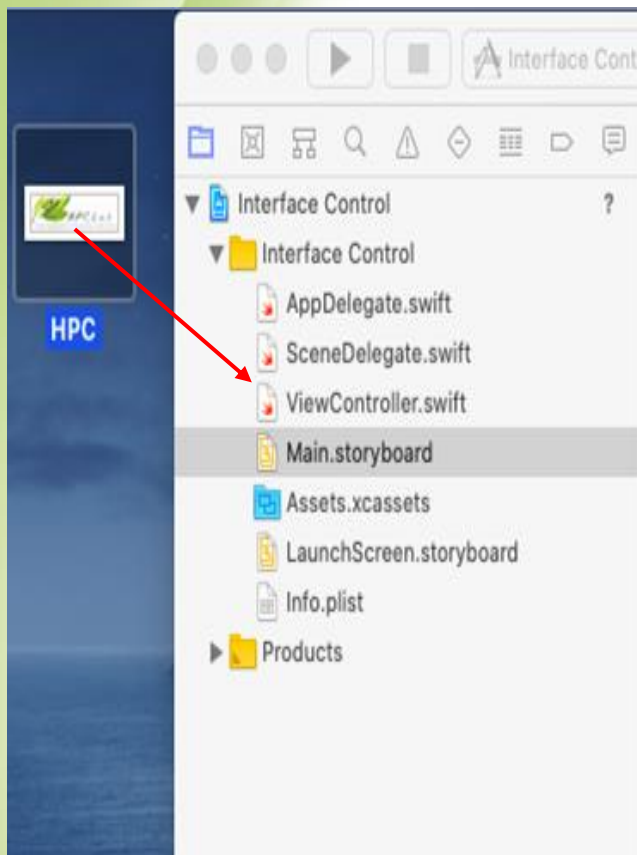
Number

☒

0.5

Symbol	20
Times New Roman	21
Zapfino	22
Chalkduster	23

Image View



將檔案複製一份到專案

Image View

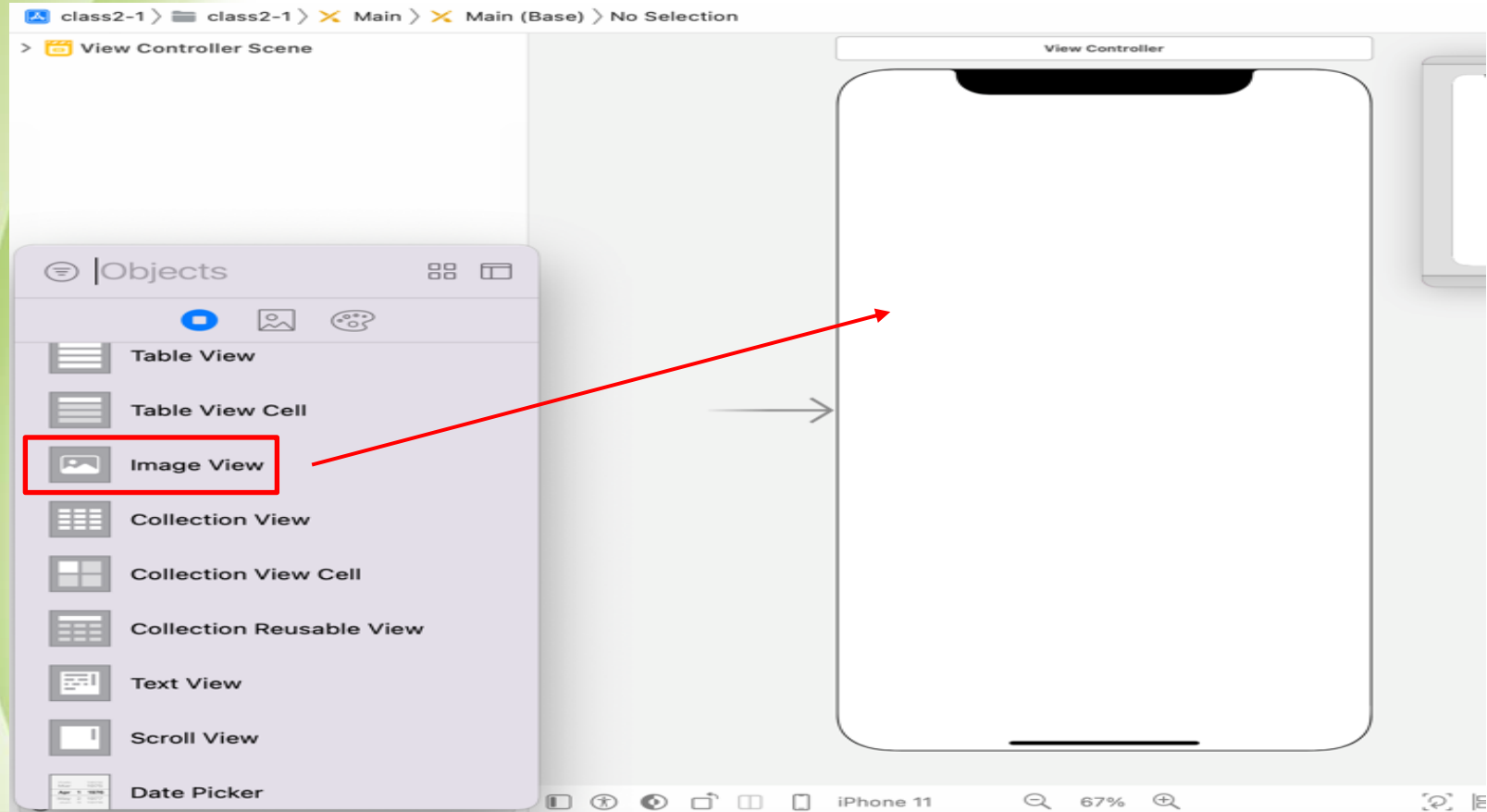


Image View

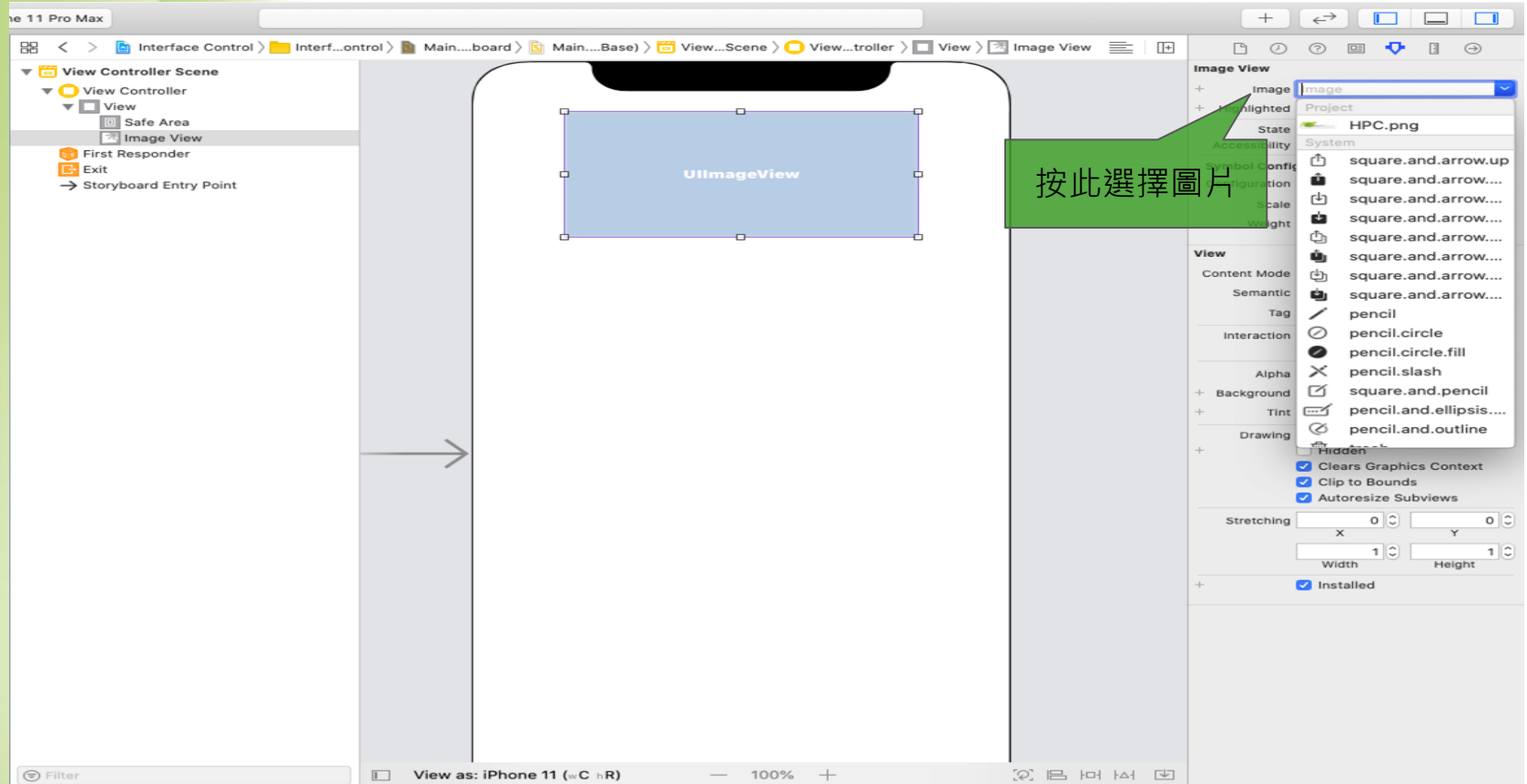


Image View

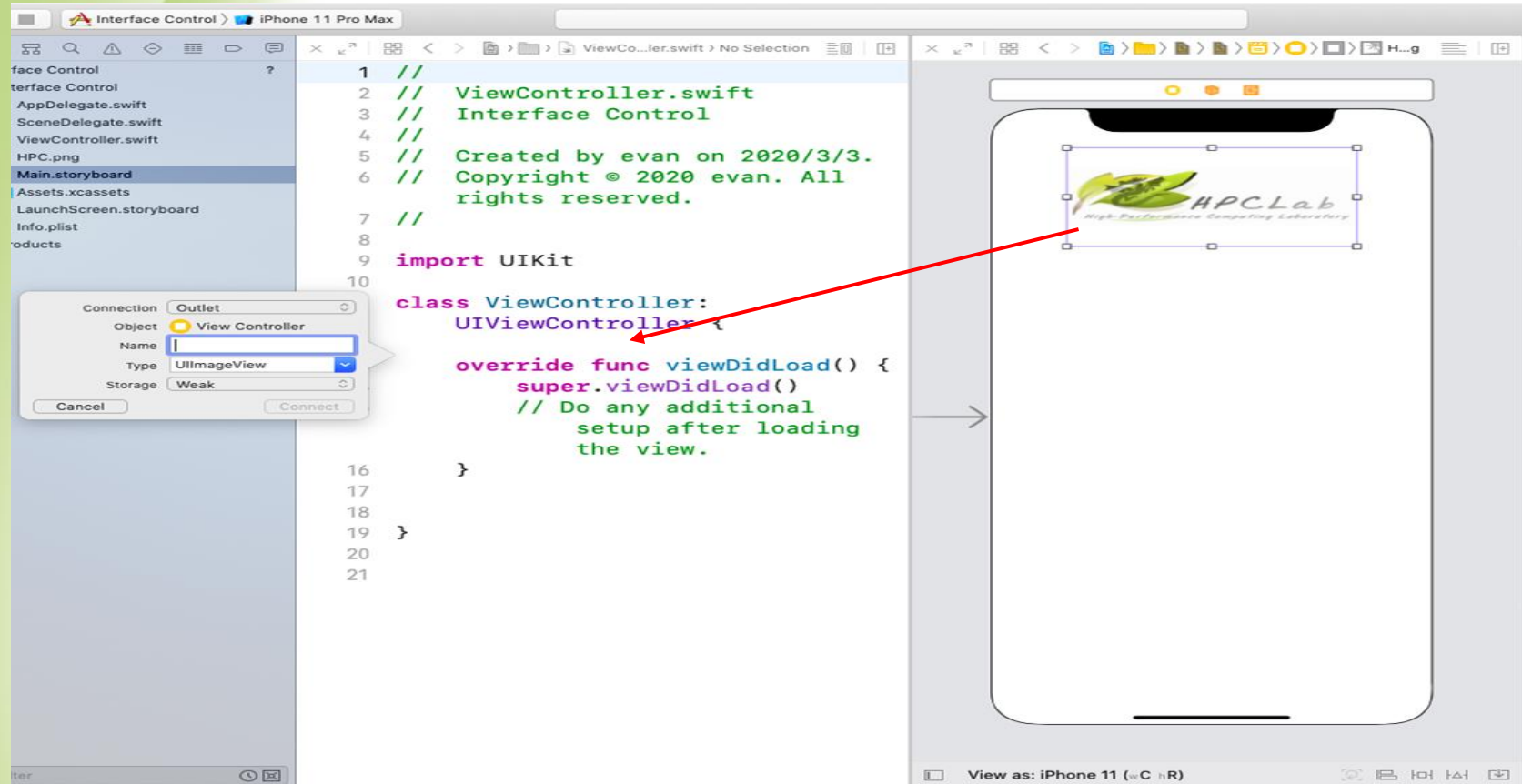


Image View

```
import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var HPC: UIImageView!

    override func viewDidLoad() {
        super.viewDidLoad()
        HPC.contentMode = .sc
```

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

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

scaleToFill
scaleAspectFit
scaleAspectFill

Scale To Fill : 放大圖片讓圖片充滿view

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 fill :
保持長寬比的原則下，放大到充滿整個 view

Image View

```
iPhone 11 Pro Max    Finished running Interface Control on iPhone 11 Pro Max    1 1 1
Interface Control > Interface Control > ViewController.swift > viewDidLoad()
1  //
2  //  ViewController.swift
3  //  class2-1
4  //
5  //  Created by nick on 2022/2/14.
6  //
7
8
9  import UIKit
10
11  class ViewController: UIViewController {
12
13      @IBOutlet weak var HPC: UIImageView
14
15      override func viewDidLoad() {
16          super.viewDidLoad()
17          // Do any additional setup after loading the view.
18          HPC.contentMode = .scaleAspectFit
19          HPC.image = UIImage(named: "HPC.png")
20      }
21
22
23  }
24
25
```

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

The image shows the Xcode Simulator interface. On the left, the code editor displays a Swift file named `ViewController.swift` with the following content:

```
1 //  
2 // ViewController.swift  
3 // class2-1  
4 //  
5 // Created by nick on 2022-02-14  
6 //  
7  
8 import UIKit  
9  
10 class ViewController: UIViewController {  
11  
12     @IBOutlet weak var HPC: UIImageView!  
13     override func viewDidLoad() {  
14         super.viewDidLoad()  
15         HPC.contentMode = .scaleAspectFit  
16         HPC.image = UIImage(named: "HPC.png")  
17     }  
18  
19 }  
20  
21  
22
```

In the center, the 'Keyboard' menu is open, showing the following options:

- Input
 - Keyboard
 - Use the Same Keyboard Language as macOS
 - Connect Hardware Keyboard
 - Toggle Software Keyboard
 - Touch Pressure
 - Audio Input
 - Audio Output
 - Increase Volume
 - Decrease Volume
 - External Displays

Two green callout boxes provide additional information:

- A box pointing to the 'Keyboard' menu item contains the text: 假如沒跳出鍵盤 可以按此出現 (If the keyboard doesn't pop up, you can click here to make it appear).
- A box pointing to the 'Toggle Software Keyboard' option contains the text: 可以將電腦鍵盤 輸入停用 (You can disable computer keyboard input).

On the right, the simulator displays a mobile app interface with the 'APCLab' logo and the text 'High-Performance Computing Laboratory'. Below the logo, there are two input fields labeled 'Name' and 'Number'.

At the bottom of the simulator, a status bar shows the date and time: 2022-02-14 12:48:49.011549+0800 class2-1[4091:135804] C keyboard iPhone-PortraitTruffle-NumberPad; using 27

KeyBoard Control

The screenshot shows the Xcode IDE with the Interface Builder window on the right and the Swift code editor on the left. The storyboard displays a mobile app interface with the HPC Lab logo and two text input fields labeled "Name" and "Number". A red arrow points from the "Name" text field to the Swift code, specifically to the line `@IBOutlet weak var nameTextField: UITextField!`. Another red arrow points from the "Did End On Exit" event in the connection menu to the same line of code. A green callout box with a pointer to the "Did End On Exit" event contains the text: 將事件選為Did End On Exit (按下return觸發的事件). The Swift code defines a `ViewController` class with three IBOutlets: `HPC` (UIImageView), `nameTextField` (UITextField), and `numTextField` (UITextField). The code is as follows:

```
// Created by nick on 2022/2/14.
//

import UIKit

class ViewController: UIViewController {

    @IBOutlet weak var HPC: UIImageView!
    @IBOutlet weak var nameTextField:
        UITextField!

    @IBOutlet weak var numTextField:
        UITextField!

}
```

The connection menu on the left shows the following settings:

- Connection: Action
- Object: View Controller
- Name: nameTextField
- Type: UITextField
- Event: Did End On Exit
- Arguments: Sender

KeyBoard Control

```
8
9  import UIKit
10
11 class ViewController: UIViewController {
12
13     @IBOutlet weak var HPC: UIImageView!
14     @IBOutlet weak var nameTextField: UITextField!
15     @IBOutlet weak var numTextField: UITextField!
16
17     @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

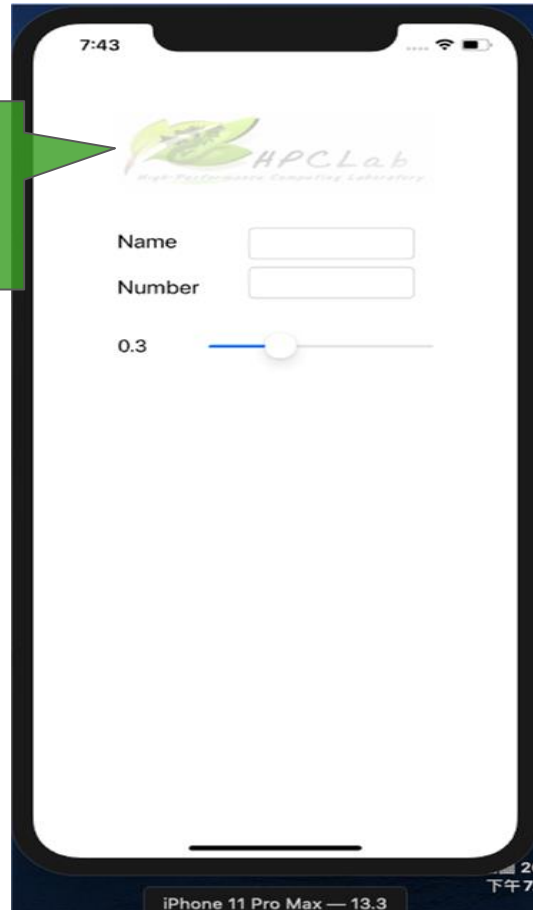
Using Slider

```
override fun 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

```
1 @IBOutlet weak var numTextField:
    UITextField!
16
17 @IBOutlet weak var alpha:
    UILabel!
18
19 @IBAction func nameTextField(_
    sender: UITextField) {
20     sender.resignFirstResponder()
21 }
22
23 @IBAction func bgTouchDown(_
    sender: UIControl) {
24
25     numTextField
26     .resignFirstResponder()
27 }
28
29 @IBAction func alphaSlider(_
    sender: UISlider) {
30
31 }
```

The UI preview on the right shows a form with the following elements:

- Name**: A text input field.
- Number**: A text input field.
- Label**: A slider control.

Red arrows indicate the connections between the code and the UI:

- An arrow points from the `nameTextField` action in the code to the `nameTextField` property in the UI preview.
- An arrow points from the `alphaSlider` action in the code to the `alphaSlider` property in the UI preview.

Using Slider

Interface Control on iPhone 11 Pro Max

Mai...ard > Mai...ase) > Vie...ene > Vie...oller > Control > Horizontal Slider < ⚠ > ☰ | +

HPCLab
High-Performance Computing Laboratory

Name

Number

Label

此處可以修改最大最小值

Slider

Value

Minimum

Maximum

+ Min Image

+ Max Image

+ Min Track

+ Max Track

+ Thumb Tint

Events ☒ Continuous Updates

Control

Alignment

Horizontal

Vertical

State ☐ Selected

☒ Enabled

☐ Highlighted

View

Content Mode

Using Slider

```
12
13 class ViewController: UIViewController {
14
15     @IBOutlet weak var HPC: UIImageView!
16     @IBOutlet weak var nameTextField: UITextField!
17     @IBOutlet weak var numTextField: UITextField!
18
19     @IBOutlet weak var alpha: UILabel!
20
21     @IBAction func nameTextField(_ sender: UITextField) {
22         sender.resignFirstResponder()
23     }
24
25     @IBAction func bgTouchDown(_ sender: UIControl) {
26         numTextField.resignFirstResponder()
27     }
28
29     @IBAction func alphaSlider(_ sender: UISlider) {
30         alpha.text = String(sender.value)
31     }
32 }
```

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

Interface Control | Build Interface Control: **Failed** | Today at 7:29 PM

face Control
terface Control
AppDelegate.swift
SceneDelegate.swift
ViewController.swift
HPC.png
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
oducts

```
11 class ViewController:  
    UIViewController {  
12  
    @IBOutlet weak var HPC:  
        UIImageView!  
    @IBOutlet weak var  
        nameTextField:  
            UITextField!  
    @IBOutlet weak var  
        numTextField: UITextField!  
16  
    @IBOutlet weak var alpha:  
        UILabel!  
22  
  
    @IBAction func  
        nameTextField(_ sender:  
            UITextField) {  
23  
        sender  
            .resignFirstResponder  
                ()  
24
```

Connection: Outlet
Object: View Controller
Name: sliderValue
Type: UISlider
Storage: Weak
Cancel Connect

需要得到拉霸的起始值
因此要將拉霸作為
Outlet連結過去

Name
Number
Label

Using Slider

將Label顯示的數字設為小數後一位

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

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

在viewDidLoad中
設定Label的初值

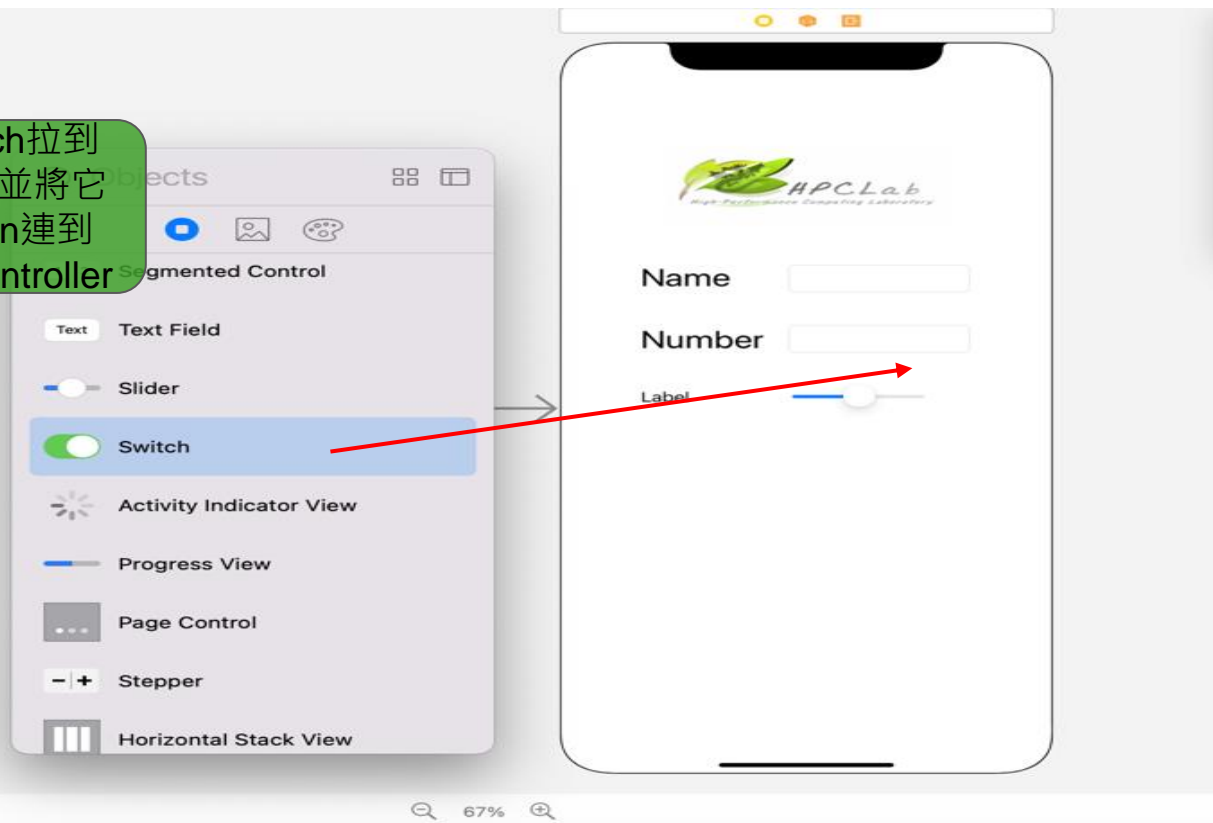
```
35 override func viewDidLoad() {  
36     super.viewDidLoad()  
37     // Do any additional setup after loading the view  
38     HPC.contentMode = .scaleAspectFit  
39     HPC.image = UIImage(named: "HPC.png")  
40     alpha.text = String(format: "%.1f", sliderValue.value)|  
41 }  
42 }
```

Switch Button



Switch Button

將Switch拉到
View後並將它
以Action連到
viewController



Switch Button

er > View > Switch

Switch

Title

Style Automatic

State On

+ On Tint Default

+ Thumb Tint Default

Control

Alignment Horizontal

Menu Shows as Primary Action

State Selected

Enabled

Highlighted

Tooltip Tooltip

View

Content Mode Scale To Fill

Semantic Unspecified

Tag 0

Interaction User Interaction Enabled

Multiple Touch

Alpha 1

+ Background Default

+ Tint Default

Drawing Opaque

Hidden

Clears Graphics Context

Clips to Bounds

Autoresize Subviews

Stretching 0 0 1

將初始狀態設為關閉

TAIWAN TECH National Taiwan University Science and Technology

HPCLab High-Performance Computing Laboratory

Switch Button

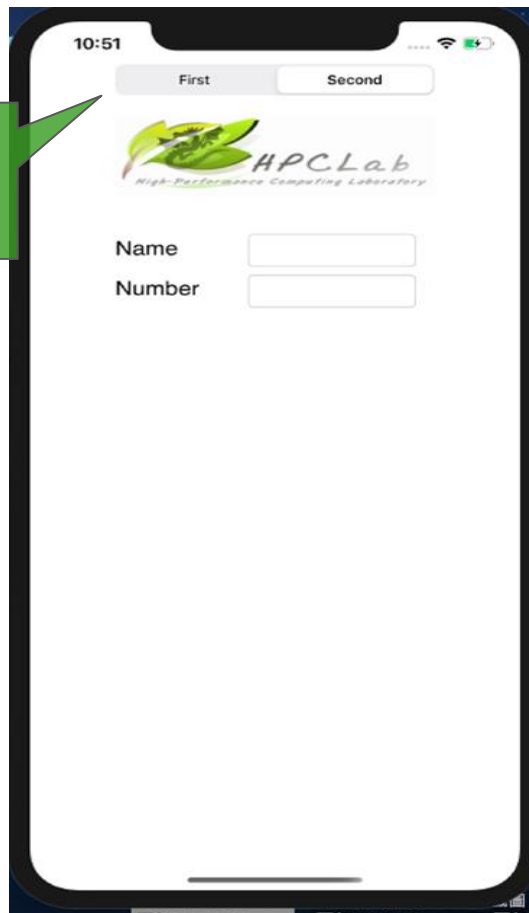
```
◎ @IBAction func secureText(_ sender: UISwitch) {  
36     if sender.isOn{  
37         numTextField.isSecureTextEntry = true  
38     } else {  
39         numTextField.isSecureTextEntry = false  
40     }  
41 }  
42  
43
```

當按鈕設定
為開時將數
字隱藏

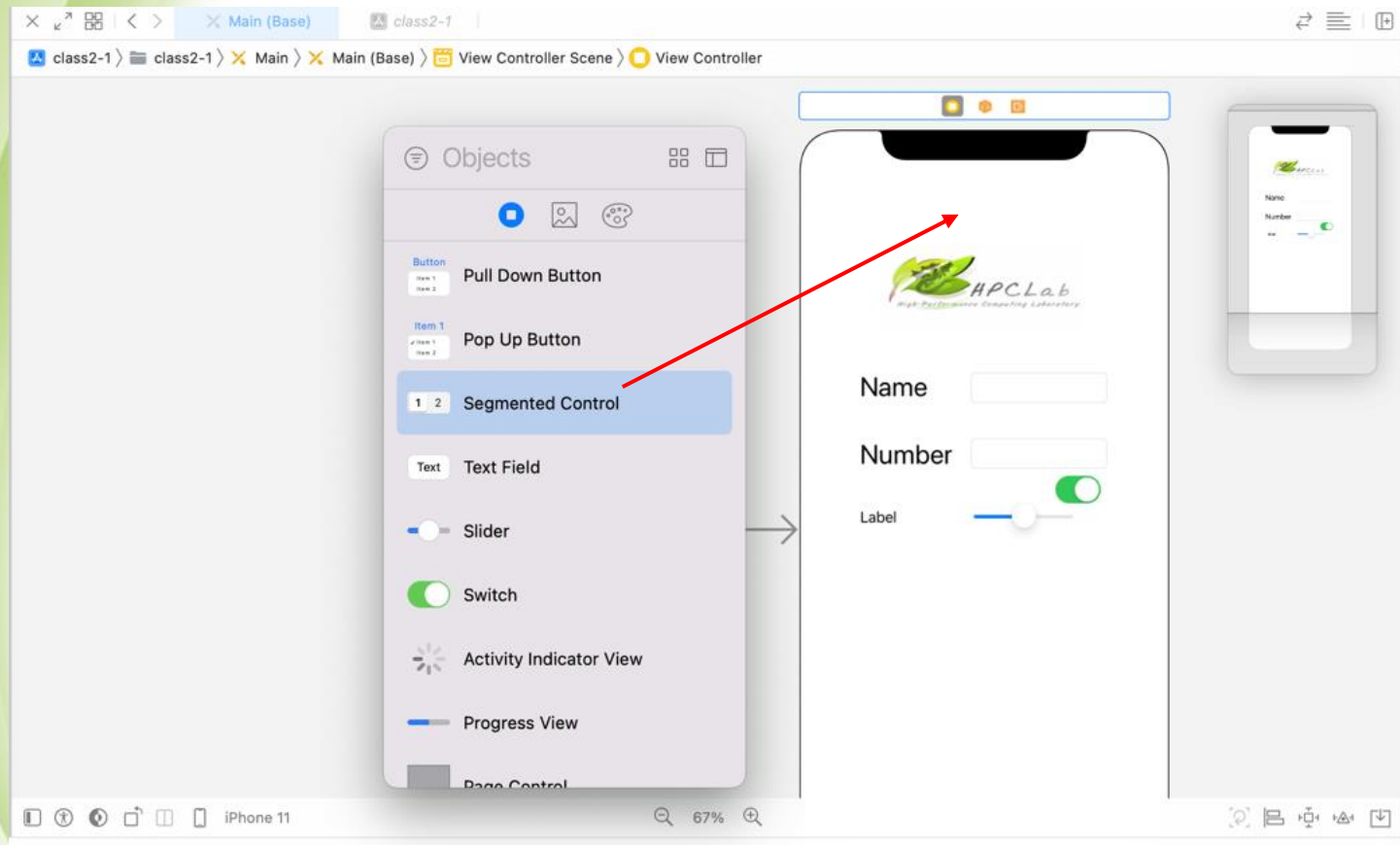
當按鈕設定
為關時顯示
數字

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


```
Interface Control > Interface Control > ViewController.swift > ViewController
37     }
38
39     @IBAction func secureText(_ sender: UISwitch) {
40         if sender.isOn{
41             numTextField.isSecureTextEntry = true
42         } else {
43             numTextField.isSecureTextEntry = false
44         }
45     }
46
47     @IBAction func segmentCtrl(_ sender: UISegmentedControl) {
48         if sender.selectedSegmentIndex == 0{
49             slider.isHidden = false
50             `switch`.isHidden = false
51             alpha.isHidden = false
52         } else {
53             slider.isHidden = true
54             `switch`.isHidden = true
55             alpha.isHidden = true
56         }
57     }
58
```

當選擇第一頁時將元件顯示，選擇第二頁時隱藏元件

Picker View

10:48

First Second

 HPCLab
High-Performance Computing Laboratory

Name

Number

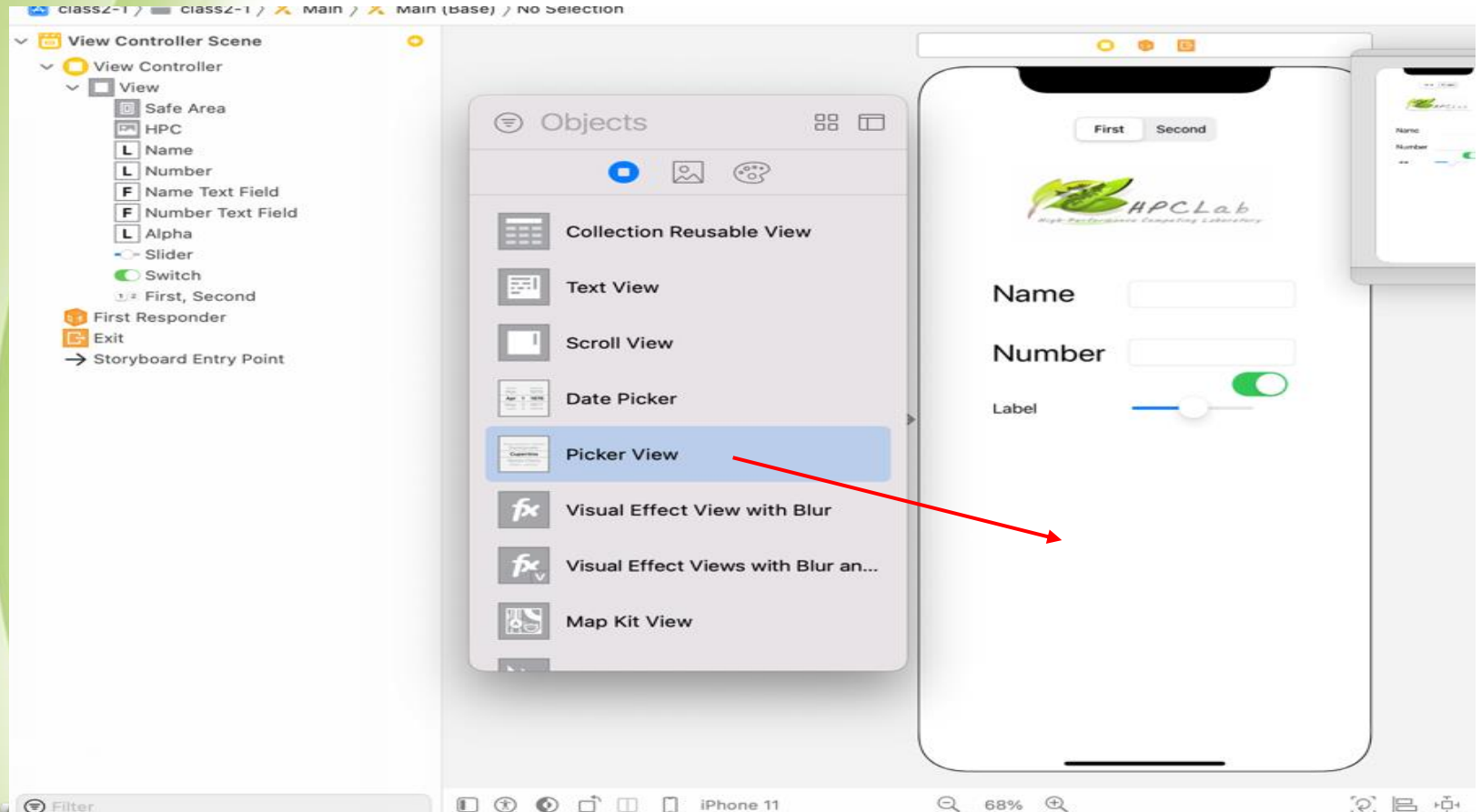
☒

0.5

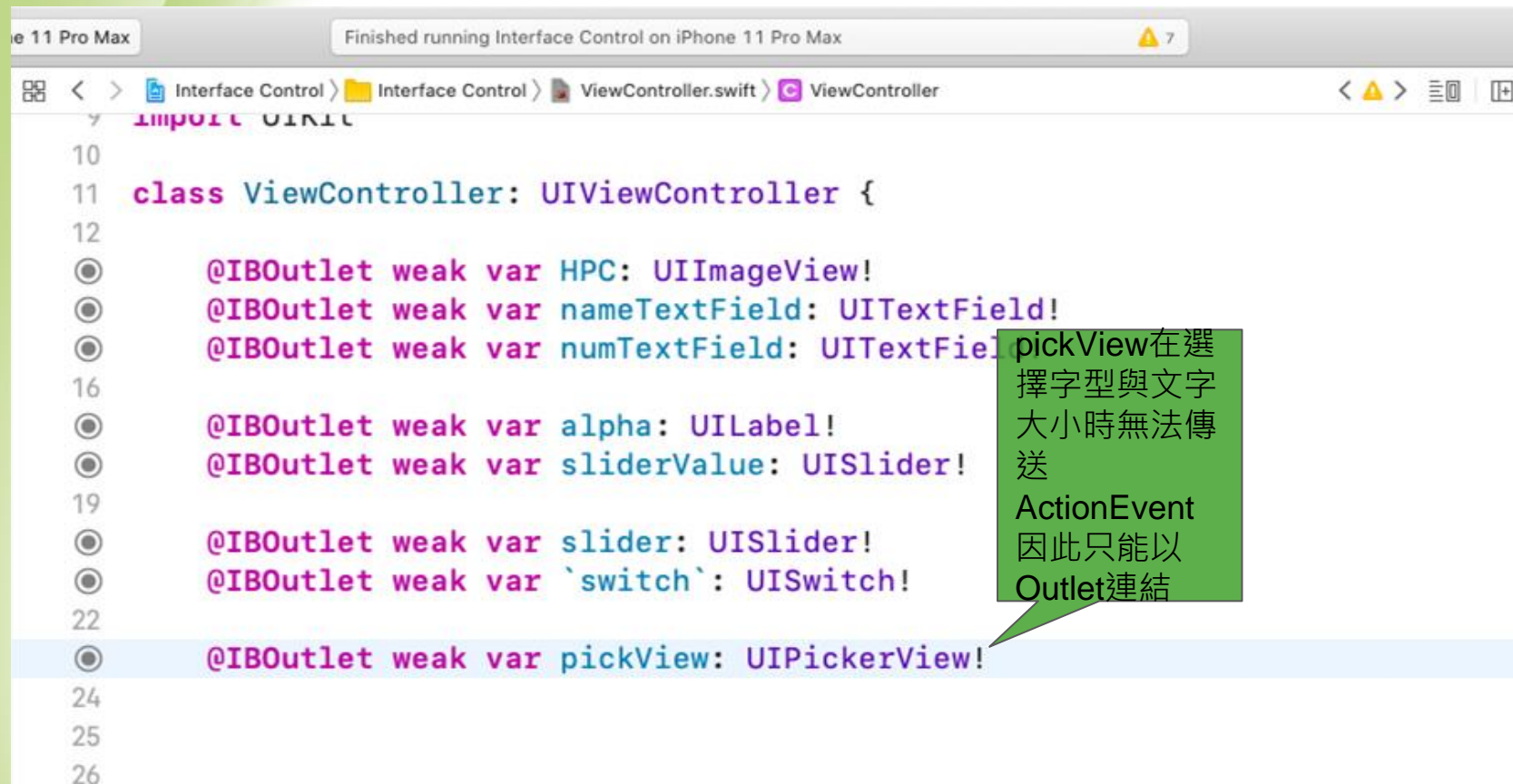
Symbol	20
Times New Roman	21
Zapfino	22
Chalkduster	23

左邊選擇
字型右邊
選擇文字
大小

Picker View

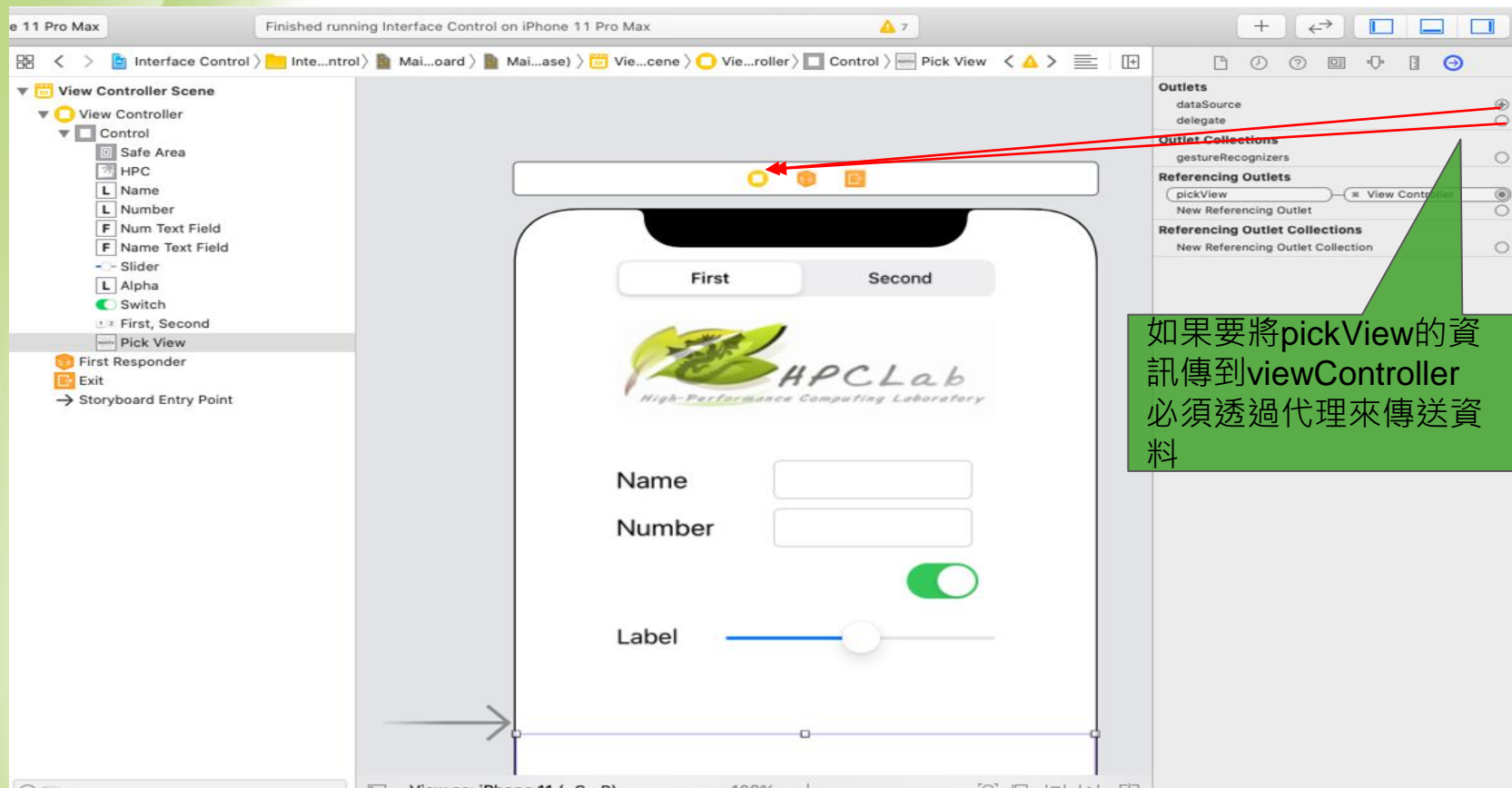


Picker View

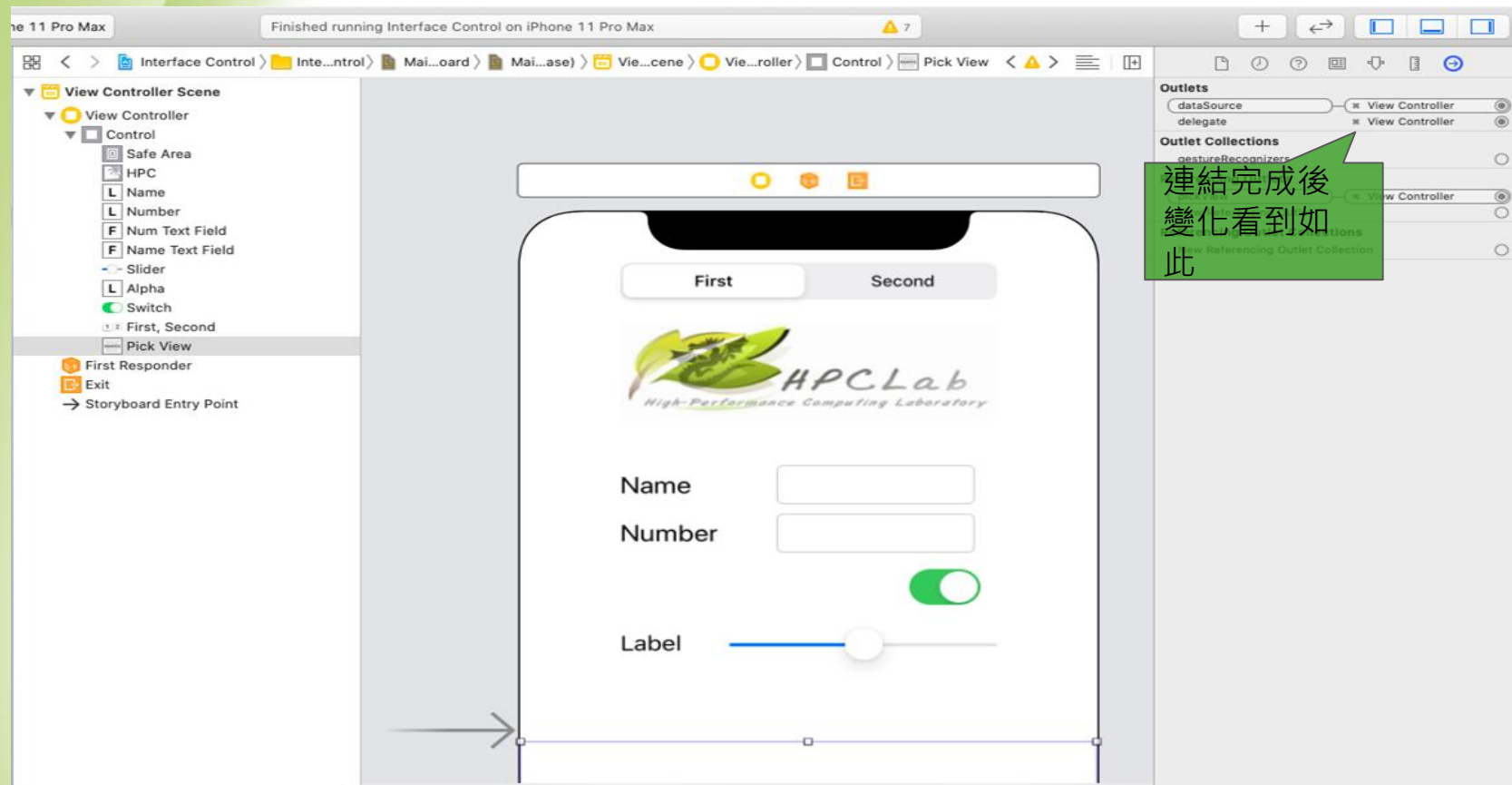


pickView在選擇字型與文字大小時無法傳送 ActionEvent 因此只能以 Outlet連結

Picker View



Picker View



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
12     @IBOutlet weak var HPC: UIImageView!
13     @IBOutlet weak var numberTextField: UITextField!
14     @IBOutlet weak var nameTextField: UITextField!
15     @IBOutlet weak var alpha: UILabel!
16     @IBOutlet weak var sliderValue: UISlider!
17     @IBOutlet weak var `switch`: UISwitch!
18     @IBOutlet weak var slider: UISlider!
19
20     @IBOutlet weak var pickview: UIPickerView!
21
22     @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.
```

上面的函
式代表有
多少欄位

```
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
```

下面的函式
是決定每一
欄有幾個選
項

```
20 @IBOutlet weak var imageView: UIImageView!
21 @IBOutlet weak var numberTextField: UITextField!
22 @IBOutlet weak var nameTextField: UITextField!
23 @IBOutlet weak var phone: UILabel!
24 @IBOutlet weak var sliderValue: UISlider!
25 @IBOutlet weak var `switch`: UISwitch!
26 @IBOutlet weak var slider: UISlider!
27
28 @IBOutlet weak var pickerView: UIPickerView!
29
30 @IBAction func nameTextField(_ sender: UITextField) {
31     sender.resignFirstResponder()
32 }
```

Picker View

```
0
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 pickerView: UIPickerView!

46
@IBOutlet weak var name: UILabel!
@IBOutlet weak var number: UILabel!
```

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]  
}
```

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

第一列回字型的陣列，row是目前選到選項的index

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

這個函式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