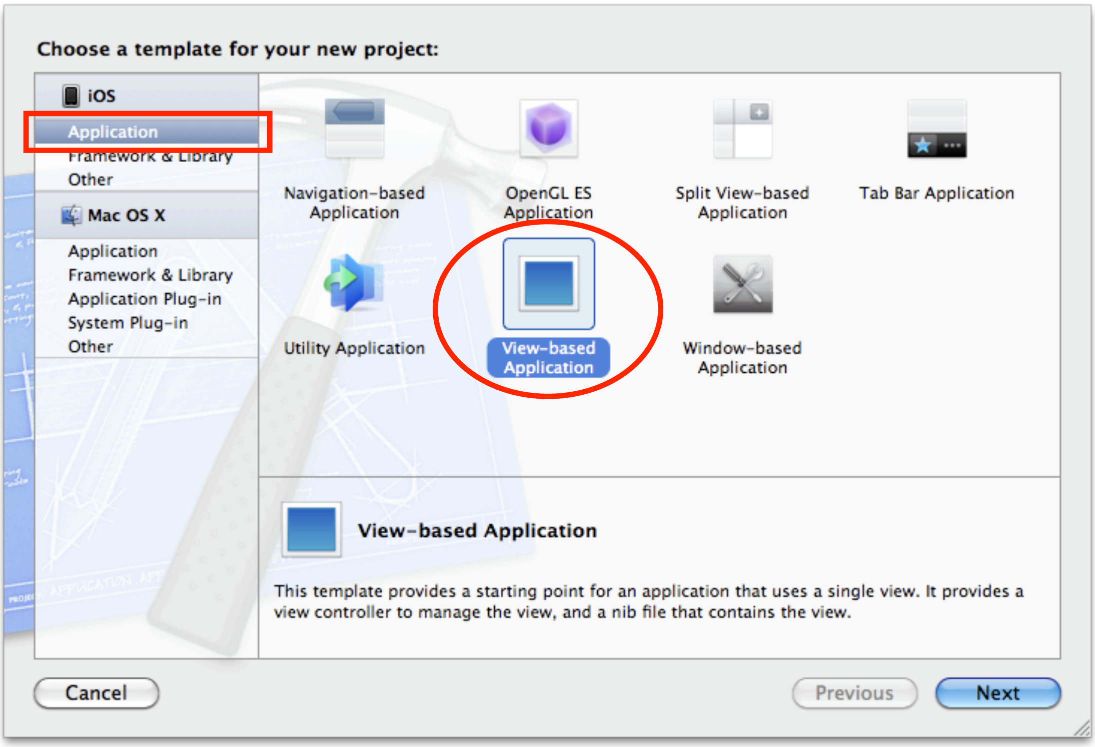
Lab SimpleGesture

Step 1. 在File>New>New Project開啓一個新的專案, 在iOS的Application目錄裡面選擇 view based application



Choose a template for your new project:

View-based Application

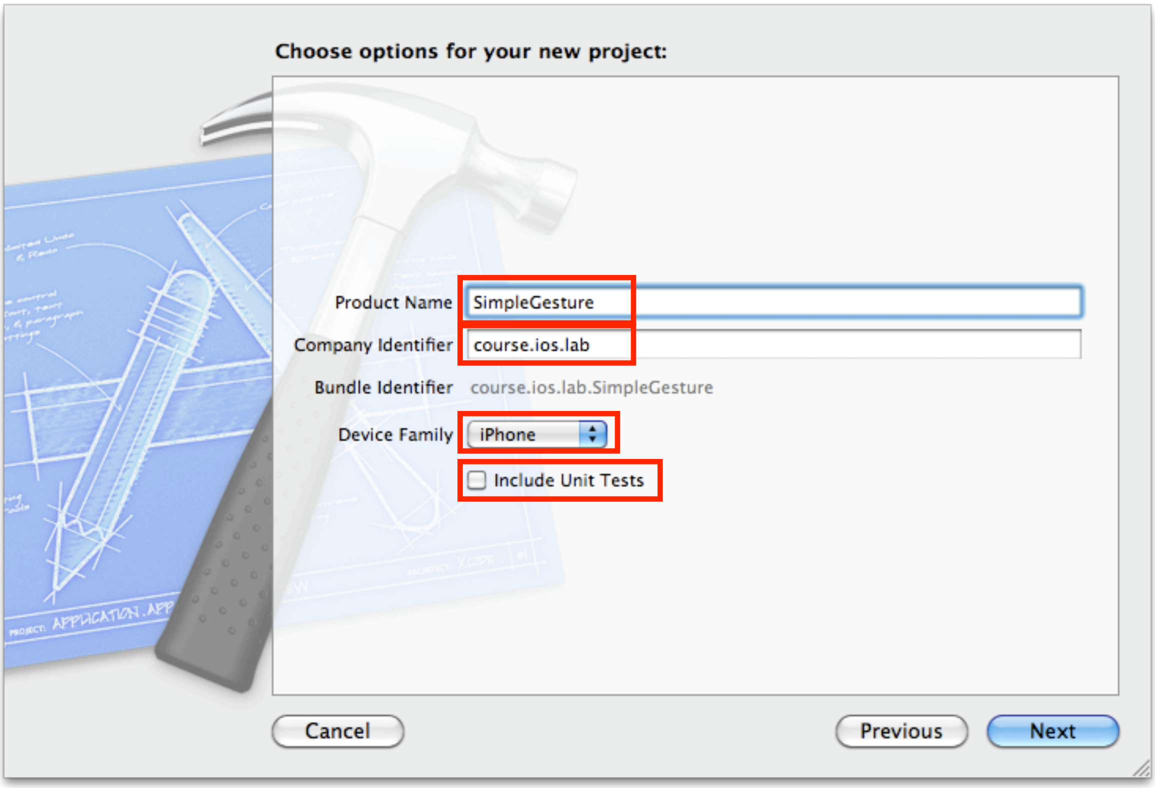
This template provides a starting point for an application that uses a single view. It provides a view controller to manage the view, and a nib file that contains the view.

Step 2. 並將此專案命名為 **SimpleGesture**

Company Identifier是填入Bundle的名稱,在此統一填入**course.ios.lab** (也可自行填入)

Device Family選擇**iPhone**

Include Unit Tests是做語意邏輯測試用,可勾選也可不勾選,在此我們統一**不勾選**, 存檔



Choose options for your new project:

Product Name: SimpleGesture

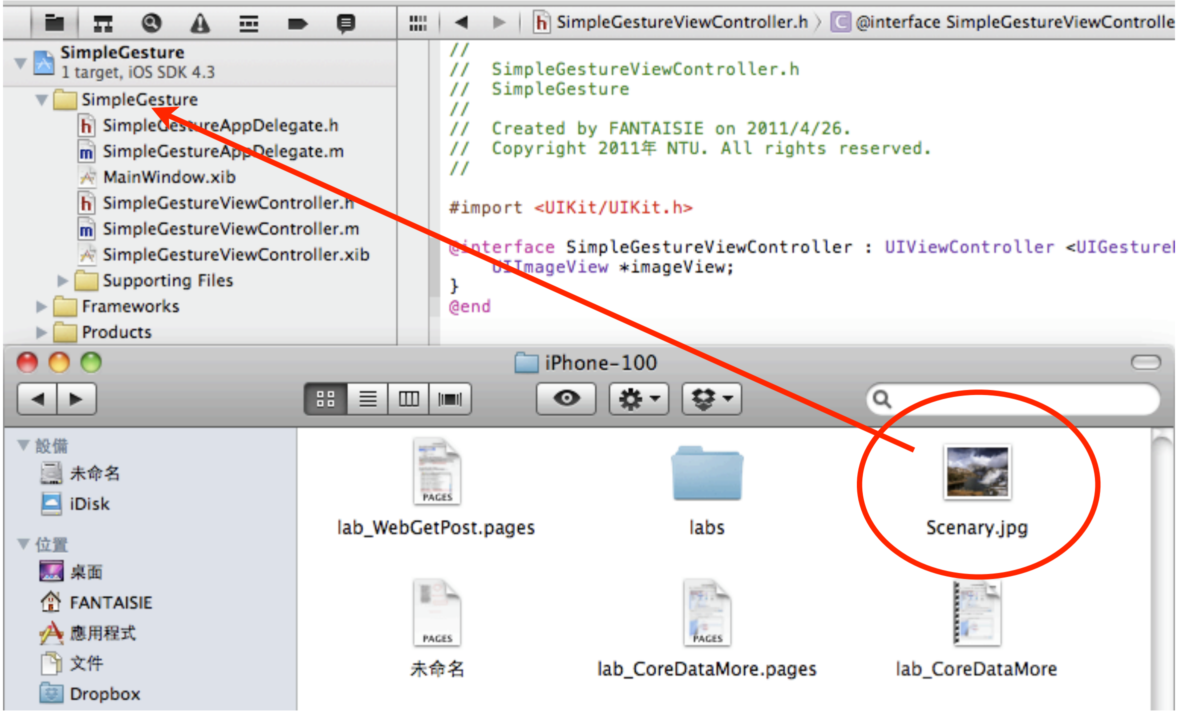
Company Identifier: course.ios.lab

Bundle Identifier: course.ios.lab.SimpleGesture

Device Family: iPhone

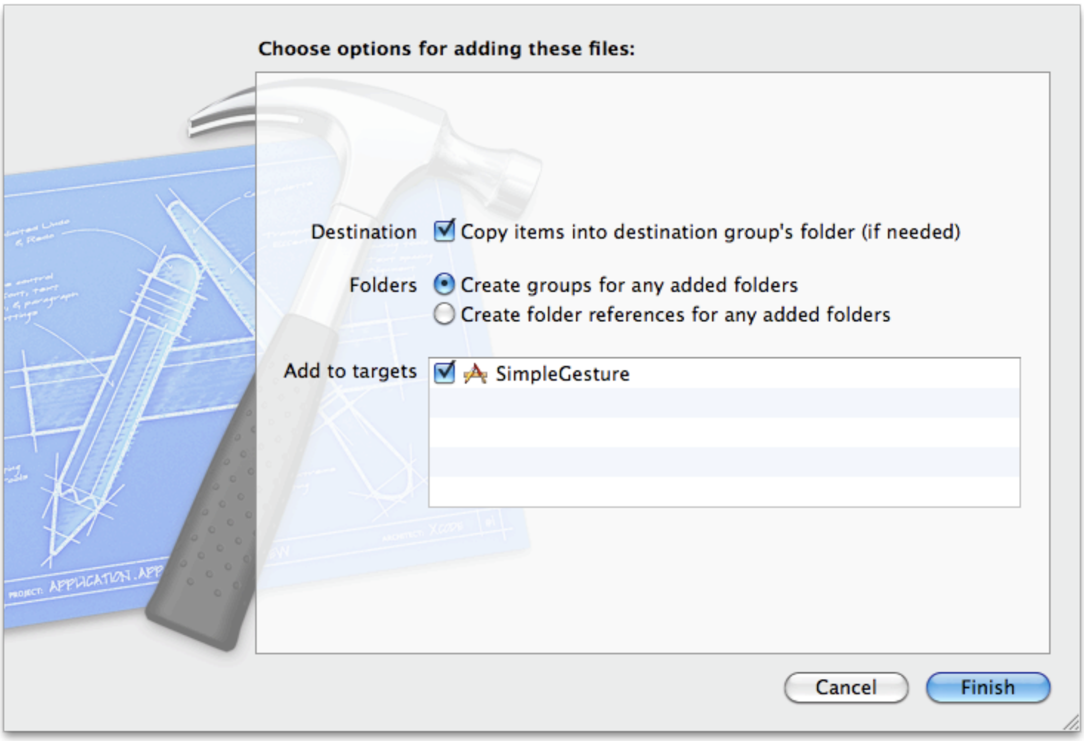
□ Include Unit Tests

Step 3. 選擇一個圖檔(解析度任意), 拖拉到Project內, 在此我們的圖檔檔名為**Scenary.jpg**



```
//
//  SimpleGestureViewController.h
//  SimpleGesture
//
//  Created by FANTAISIE on 2011/4/26.
//  Copyright 2011年 NTU. All rights reserved.
//

#import <UIKit/UIKit.h>

@interface SimpleGestureViewController : UIViewController <UIGesture
    UIImageView *imageView;
}
@end
```

勾選 **Copy items into destination group's folder (if needed)**



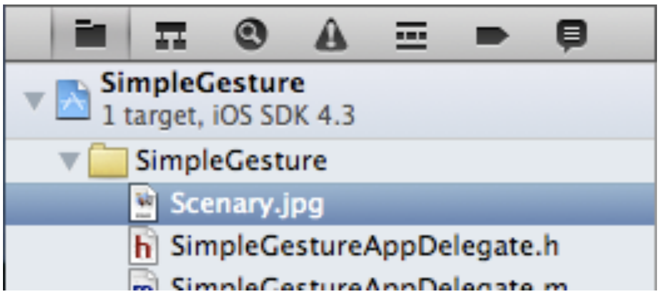
Choose options for adding these files:

Destination ☑ Copy items into destination group's folder (if needed)

Folders ● Create groups for any added folders
○ Create folder references for any added folders

Add to targets ☑ SimpleGesture

圖檔出現在我們的 Project Navigator 中



Step 4. 開啓 SimpleGestureViewController.h, 加入一個UIImageView叫做 **imageView**, 以及宣告兩個我們Rotate和Pinch會用到的method

```
#import <UIKit/UIKit.h>

@interface SimpleGestureViewController : UIViewController {
    UIImageView *imageView;
}
- (void)handleRotationFrom:(UIRotationGestureRecognizer *)recognizer;
- (void)handlePinchFrom:(UIPinchGestureRecognizer *)recognizer;
@end
```

Step 5. 開啓 SimpleGestureViewController.m , 找到 - (void)ViewDidLoad 然後把Mark去掉, 加入下面的程式去為我們ViewController的view去增加兩個UIGestureRecognizer, 去執行兩個不同的method, 然後將我們的imageView做初始化, 包括設定frame, source, 位置, 最後加到我們的view裡面成為一個subView

注意:

`imageView.image = [UIImage imageNamed:@"Scenary.jpg"];` 裡的@”Scenary.jpg 要使用你自己加入的圖檔名稱

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    UIGestureRecognizer *recognizer;

    recognizer = [[UIRotationGestureRecognizer alloc]
initWithTarget:self action:@selector(handleRotationFrom:)];
    [self.view addGestureRecognizer:recognizer];
    [recognizer release];

    recognizer = [[UIPinchGestureRecognizer alloc] initWithTarget:self
action:@selector(handlePinchFrom:)];
    [self.view addGestureRecognizer:recognizer];
    [recognizer release];

    imageView = [[UIImageView alloc] initWithFrame:CGRectMake(0.0, 0.0,
150.0, 150.0)];
    imageView.image = [UIImage imageNamed:@"Scenary.jpg"];
    imageView.center = self.view.center;
    [self.view addSubview:imageView];
    [imageView release];
}
```

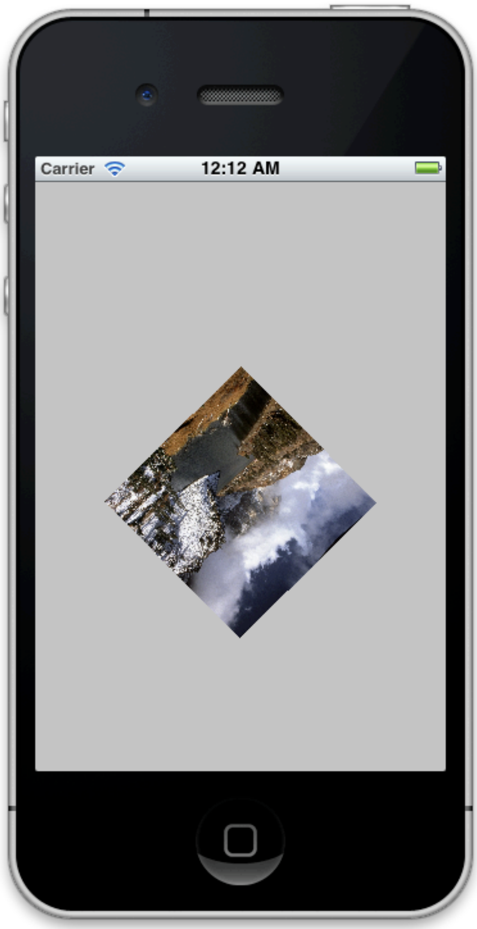
Step 6. 找到 - (void) dealloc 然後在這裡在dealloc時將我們的imageView做release

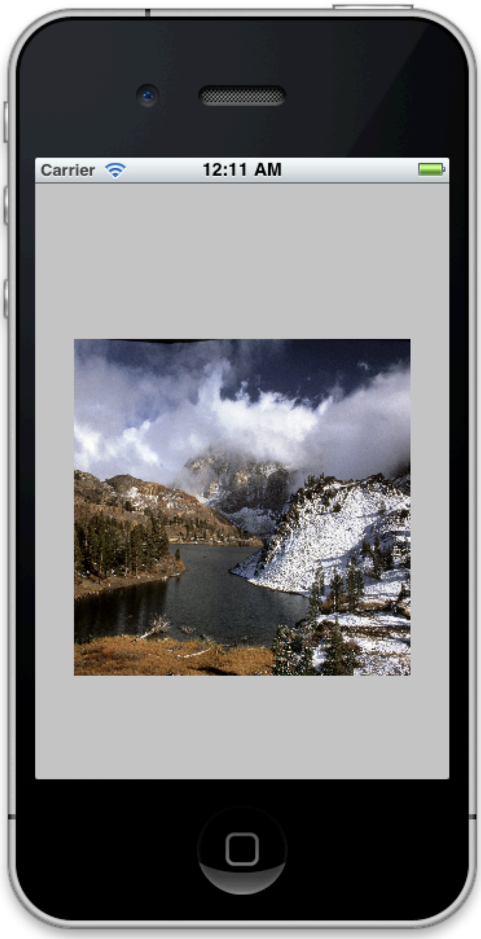
```
- (void)dealloc
{
    [imageView release];
    [super dealloc];
}
```

Step 7. 實作Rotate和Pinch時會執行的兩個method, 主要是使用 CGAffineTransform 去依照recognizer所認到的行為或參數值去改變imageView的屬性和顯示,後面beginAnimations到commitAnmations是在手指(鼠標)放開後在一秒內回復到原來imageView的呈現方式, 如果將整段Mark掉的話imageView就會保留在因為Rotation或是Pinch而更改後的狀態

```
- (void)handleRotationFrom:(UIRotationGestureRecognizer *)recognizer {

    imageView.transform = CGAffineTransformMakeRotation([recognizer 
rotation]);
     [UIView beginAnimations:nil context:NULL];
     [UIView setAnimationDuration:1];
    imageView.transform = CGAffineTransformIdentity;
     [UIView commitAnimations];
}

- (void)handlePinchFrom:(UIPinchGestureRecognizer *)recognizer{

    imageView.transform = CGAffineTransformMakeScale(recognizer.scale, 
recognizer.scale);

     [UIView beginAnimations:nil context:NULL];
     [UIView setAnimationDuration:1];
    imageView.transform = CGAffineTransformIdentity;
     [UIView commitAnimations];
}
```

Step 12. Run (⌘+R)
使用option(alt)加滑鼠(觸控版)就可模擬兩指Rotate(不改變兩點距離)和Pinch(縮放)的功能


Carrier
12:12 AM


Carrier
12:11 AM