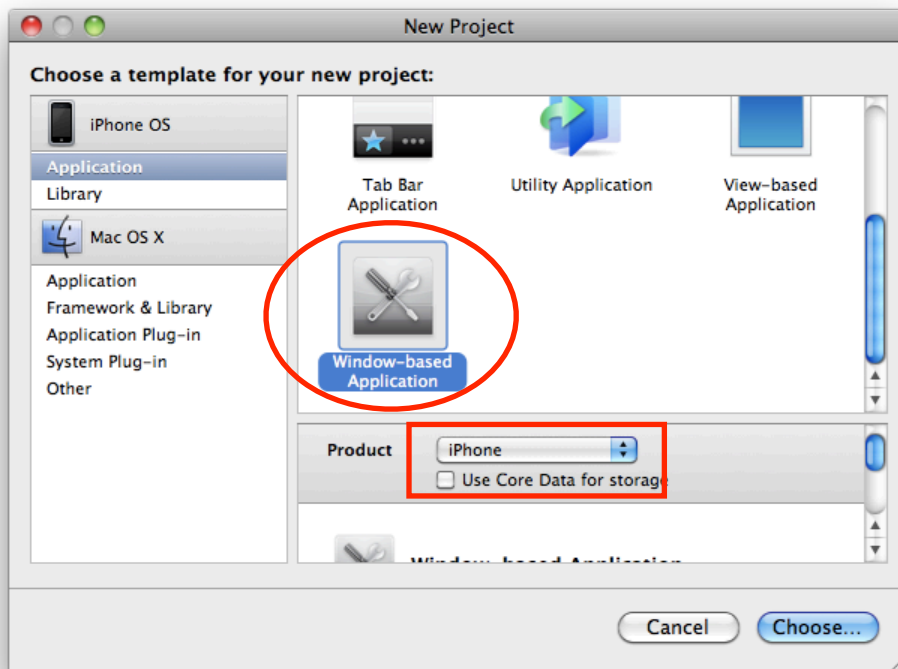
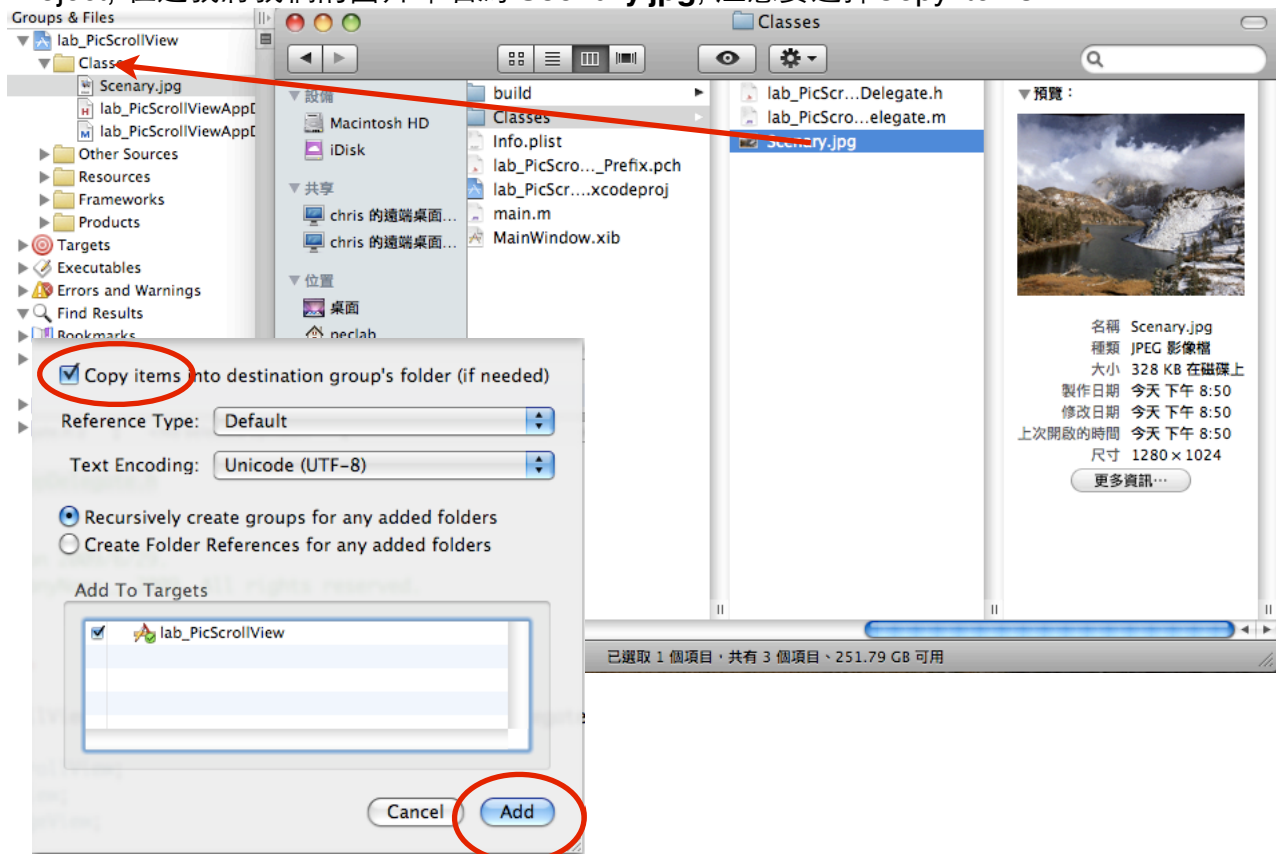


Lab PicScrollView

Step1. 在File開啓一個新的project, 選擇 Window-Based Application, 將project取名為 lab_PicScrollView



Step2. 隨意找尋一張size大於iPhone
[UIScreen mainScreen].applicationFrame.size = 320x480 的圖片並加入我們的 Project, 在這我將我們的圖片命名為 **Scenary.jpg**, 注意要選擇Copy items



Step3. 在 lab_PicScrollViewAppDelegate.h 裡加入 <UIScrollViewDelegate> 來遵循 UIScrollViewDelegate 這個 protocol，並宣告一些我們會使用到的物件

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

@interface lab_PicScrollViewAppDelegate : NSObject
<UIApplicationDelegate, UIScrollViewDelegate> {
    UIWindow *window;
    UIScrollView *myScrollView;
    UIView *myContentView;
    UIImageView *myImageView;
    UIImage *myImage;
}

@property (nonatomic, retain) IBOutlet UIWindow *window;

@end
```

Step4. 在 lab_PicScrollViewAppDelegate.m 裡的

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {}
-
```

實作我們的myScrollView，判斷Scale的 myContentView，以及作為圖像顯示的 myImageView

注意：程式碼裡面myImage = [UIImage imageNamed:@"Scenary.jpg"];裡的 Scenary.jpg 要寫入你自己剛加入專案的圖片檔名稱

```
- (BOOL)application:(UIApplication *)application didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    // Override point for customization after application launch.
    myScrollView = [[UIScrollView alloc] initWithFrame:[UIScreen mainScreen].applicationFrame];
    myScrollView.backgroundColor = [UIColor blackColor];
    myScrollView.delegate = self;
    myScrollView.bouncesZoom = YES;

    myImage = [UIImage imageNamed:@"Scenary.jpg"];
    CGRect myFrame = CGRectMake(0, 0, myImage.size.width, myImage.size.height);
    myImageView = [[UIImageView alloc] initWithFrame:myFrame];
    myImageView.image = myImage;

    myContentView = [[UIView alloc] initWithFrame:
        CGRectMake(0, 0, myImage.size.width, myImage.size.height)];
    myScrollView.contentSize = myContentView.frame.size;
    myScrollView.minimumZoomScale = myScrollView.frame.size.width / myImage.size.width;
    myScrollView.maximumZoomScale = 1.0;

    [myContentView addSubview:myImageView];
    [myScrollView addSubview:myContentView];
    [window addSubview:myScrollView];
    [myImageView release];
    [myContentView release];
    [myScrollView release];

    [window makeKeyAndVisible];

    return YES;
}
```

```
myScrollView = [[UIScrollView alloc] initWithFrame:[UIScreen
 mainScreen].applicationFrame];
myScrollView.backgroundColor = [UIColor blackColor];
myScrollView.delegate = self;
myScrollView.bouncesZoom = YES;
```

我們開啓與mainScreen Frame同大小的ScrollView，將背景設為黑色，並委任<UIScrollViewDelegate> 這個protocol, 把Zooming時bounce設為YES

```
myImage = [UIImage imageNamed:@"Scenary.jpg"];
CGRect myFrame = CGRectMake(0, 0, myImage.size.width,
myImage.size.height);
myImageView = [[UIImageView alloc] initWithFrame:myFrame];
myImageView.image = myImage;
這邊是加入我們的圖檔成為一個ImageView
```

```
myContentView = [[UIView alloc] initWithFrame:
CGRectMake(0, 0, myImage.size.width, myImage.size.height)];
這邊是把每次Zooming後更新的Scale的View作初始化
```

```
myScrollView.contentSize = myContentView.frame.size;
myScrollView.minimumZoomScale = myScrollView.frame.size.width /
myImage.size.width;
myScrollView.maximumZoomScale = 1.0;
這邊是把更新的ContentView給ScrollView，並設定Zooming最小Fit ScrollView
的大小，最大顯示原圖的大小
```

```
[myContentView addSubview:myImageView];
[myScrollView addSubview:myContentView];
[window addSubview:myScrollView];
[myImageView release];
[myContentView release];
[myScrollView release];
最後作addSubview和release的動作,避免Memory Leak
```

Step5. 在 lab_PicScrollViewAppDelegate.m 裡加入viewForZoomingInScrollView
去知道View的Scaling

```
-(UIView *)viewForZoomingInScrollView:(UIScrollView *)myScrollView
{
    return myContentView;
}
```

Step6. Simulation (command + enter)

原來預設是顯示原圖大小



使用單鍵(相當於單點觸碰)可以拉到整張圖片的各個位置



使用 option(alt) + 滑鼠左鍵 可相當於兩點觸碰



Fit the main screen

