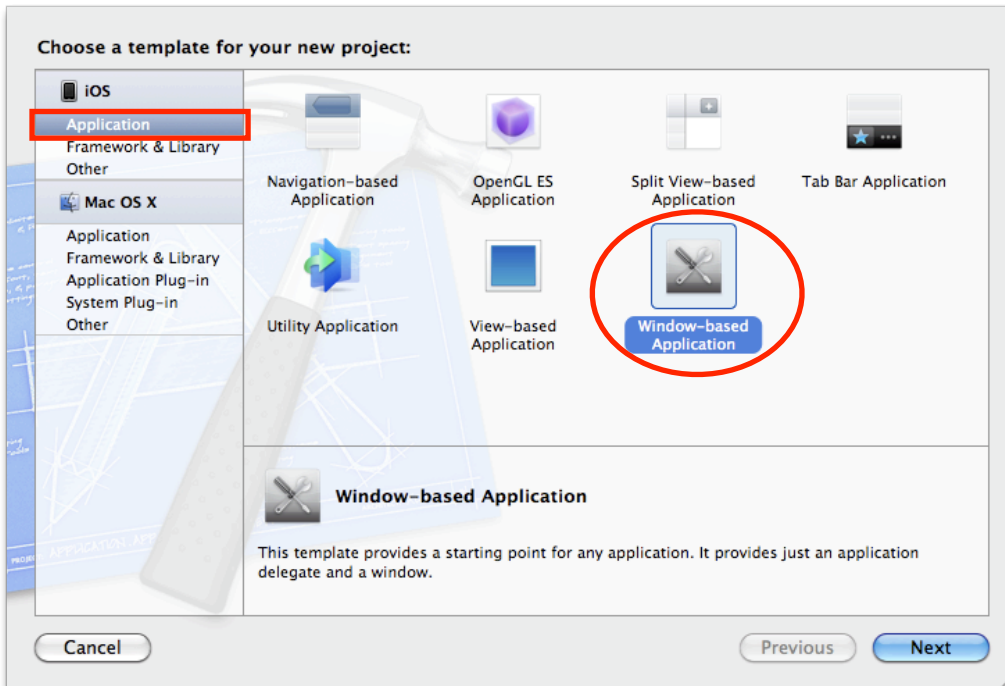


Lab CodeUIViewController

Step 1. 開啓一個新的project, 選擇window-based application

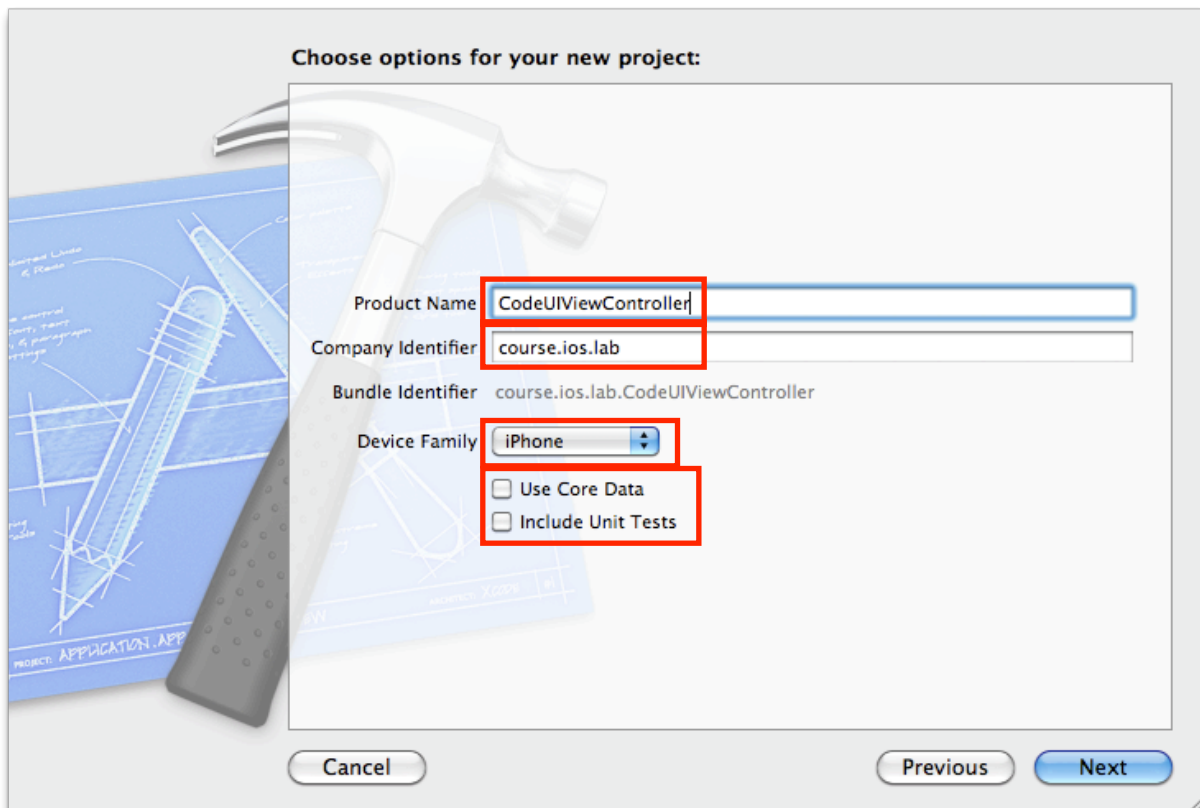


Step 2. 並將此專案命名為 **CodeUIViewController**

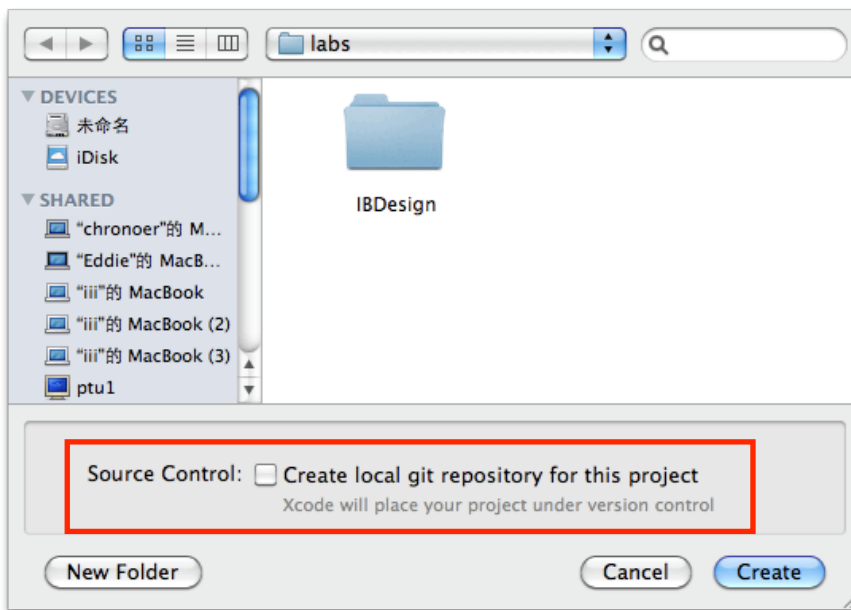
Company Identifier是填入Bundle的名稱,在此統一填入**course.ios.lab** (也可自行填入)

Device Family選擇**iPhone**

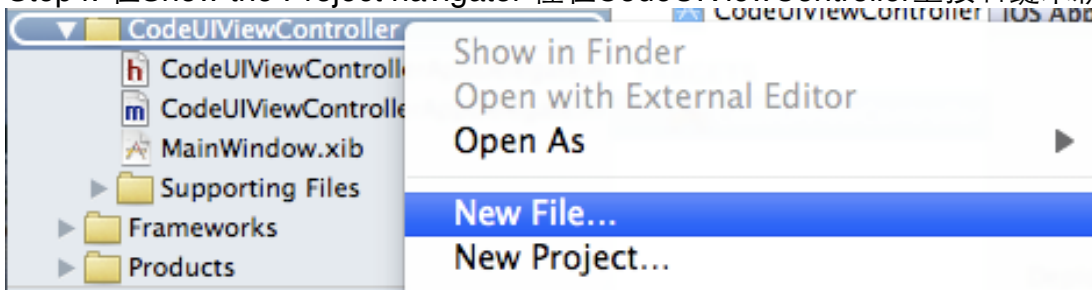
在這專案不勾選Use Core Data,而Include Unit Tests是做語意測試用,在此我們統一不勾選



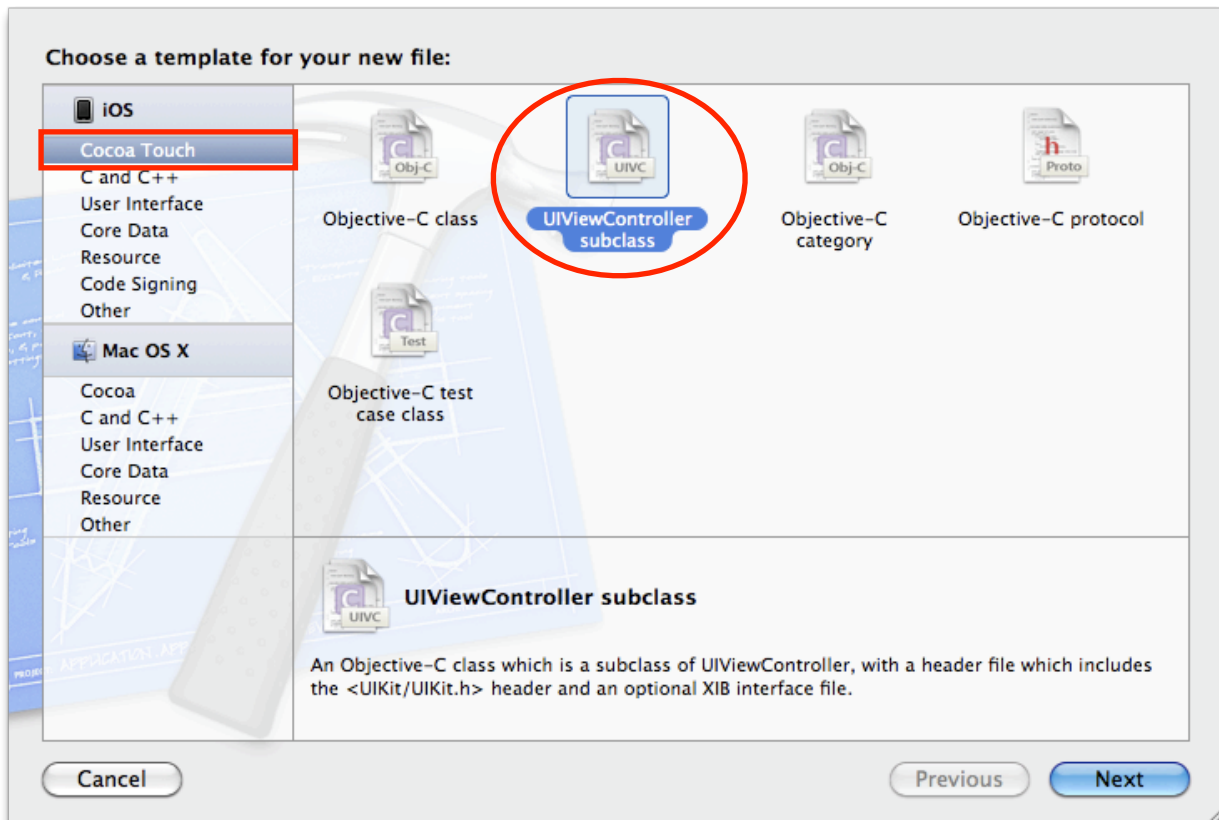
Step 3. 選擇存檔的位置, 在此我們不做version control, 統一不勾選Create local git repository for this project



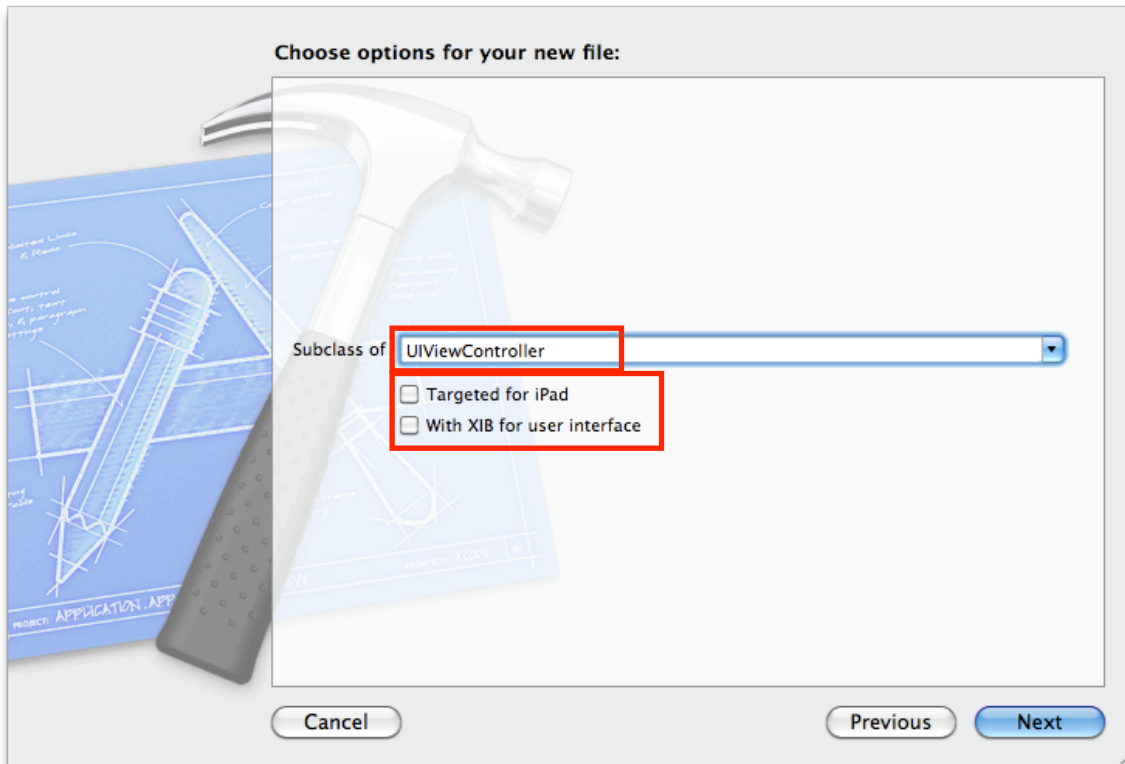
Step 4. 在Show the Project navigator 裡在CodeUIViewController上按右鍵來新增檔案



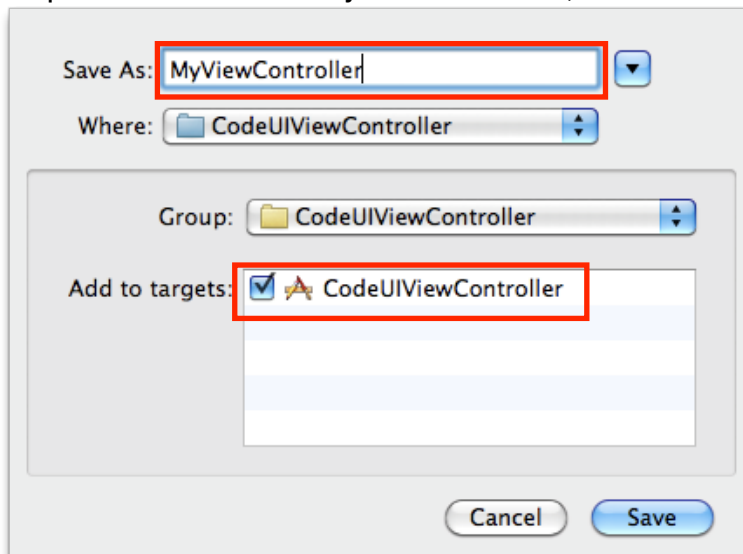
Step 5. 在iOS裡的Cocoa Touch裡選擇UIViewController subclass



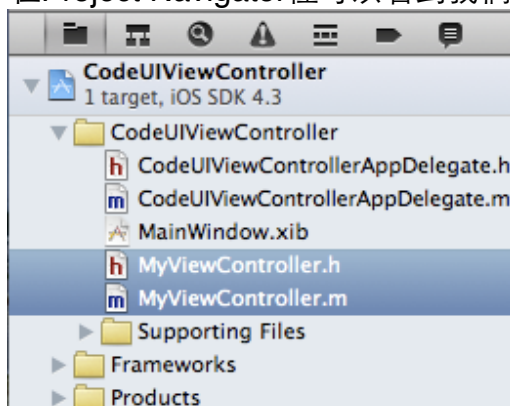
Step 6. 選擇Subclass of **UIViewController**,在此不勾選Targeted for iPad,也不選擇With XIB for user interface



Step 7. 將Class命名為MyViewController, 並Add to targets勾選CodeUIViewController



在Project Navigator裡可以看到我們新增的MyViewController.h 和 MyViewController.m



Step 8. 在Project Navigator裡選擇 MyViewController.h , 在裡面我們定義一個label 一個 button, 以及一個給 button用的 action, myAction

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

@interface MyViewController : UIViewController {
    UIButton *myButton;
    UILabel *myLabel;
}

-(void) myAction;

@end
```

Step 9. 在Project Navigator裡選擇 MyViewController.m, 實作 **constructor**, 我們新增的 method **myAction**, 以及 **destructor**, destructor是在系統產生的 -(void)dealloc{} 加入程式碼.

```
#import "MyViewController.h"

@implementation MyViewController

-(id) init
{
    if((self = [super init])){
    }
    return self;
}

-(void) myAction
{
    [myLabel setText:@"Hello MyViewController"];
}

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

Step 10. 同樣在 MyViewController.m 找到 loadView. 將原來的Mark去掉我們在裡面實作 UIView, UILabel 以及 UIButton 物件

```
// Implement loadView to create a view hierarchy programmatically,
// without using a nib.
- (void)loadView
{
    [super loadView];
    UIView *myView = [[UIView alloc] initWithFrame:[UIScreen
mainScreen].applicationFrame];
    myButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    myButton.frame = CGRectMake(110, 150, 100, 50);
    myLabel = [[UILabel alloc] initWithFrame:CGRectMake(100, 100, 300,
50)];
    [myButton setTitle:@"push" forState:UIControlStateNormal];
    [myLabel setText:@"Label"];
    [myButton addTarget:self action:@selector(myAction)
forControlEvents:UIControlEventTouchUpInside];
    [myView addSubview:myLabel];
    [myView addSubview:myButton];
    self.view = myView;
    [myView release];
}
```

在這段程式中

UIView* myView=[[UIView alloc] initWithFrame:[UIScreen mainScreen].applicationFrame];
這一段是開一個 UIView 形態的 view, 當作最底層的 view, 我們後面會將這個view傳給
UIController 管理.

myButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];

將 myButton 初始化, 並且將它的 type 設成前面實驗都用到的 UIButtonTypeRoundedRect.
更多的形態,請查 Help > Documentation, 找UIButtonType

myButton.frame = CGRectMake(110, 150, 100, 50);

這一段是將myButton在myView上的位置和大小定義出來

myLabel = [[UILabel alloc] initWithFrame:CGRectMake(100, 100, 300, 50)];

直接將 myLabel初始化, 而且直接用 initWithFrame 將它的位置大小定義出來

[myButton setTitle:@"push" forState:UIControlStateNormal];

[myLabel setText:@"Label"];

設定myButton 以及 myLabel上的文字

[myButton addTarget:self action:@selector(myAction)

forControlEvents:UIControlEventTouchUpInside];

我們將myAction與myButton的 touch down動作連結起來.

[myView addSubview:myLabel];

[myView addSubview:myButton];

接著我們把myLabel以及myButton加到myView裡

self.view = myView;

[myView release];

我們將 myView 傳給這個 UIViewController管理,然後就可以把 myView release掉了.

Step 11. 在Project Navigator裡選擇 codeViewControllerAppDelegate.m 在裡面先import我們的 MyViewController.h

在

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
這個程式進入點將我們的MyViewController初始化並將其View加入window成為一個subview

```
#import "CodeUIViewControllerAppDelegate.h"
```

```
#import "MyViewController.h"
```

```
@implementation CodeUIViewControllerAppDelegate
```

```
@synthesize window=_window;
```

```
- (BOOL)application:(UIApplication *)application  
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions  
{  
    // Override point for customization after application launch.
```

```
    MyViewController *myController = [[MyViewController alloc] init];  
    [self.window addSubview:myController.view];
```

```
    [self.window makeKeyAndVisible];  
    return YES;
```

```
}
```

在這段程式中

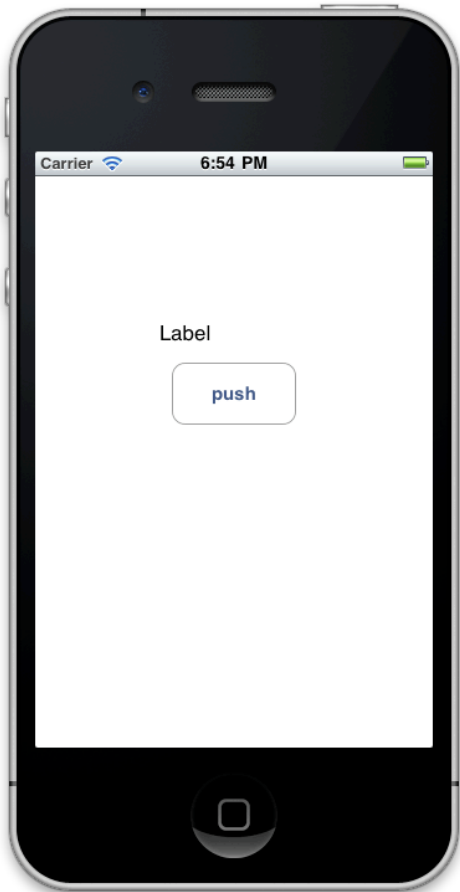
```
MyViewController* myViewController = [[MyViewController alloc] init];
```

我們首先 initialize 一個 MyViewController形態的 myViewController

```
[self.window addSubview:myViewController.view];
```

接著我們將 myViewController 的 view 交給 變成self的window的subView.

Step 12. 主畫面左上方執行Run (或是⌘+R)



當我們 按下 push的時候, label就被設定成 Hello MyViewController

