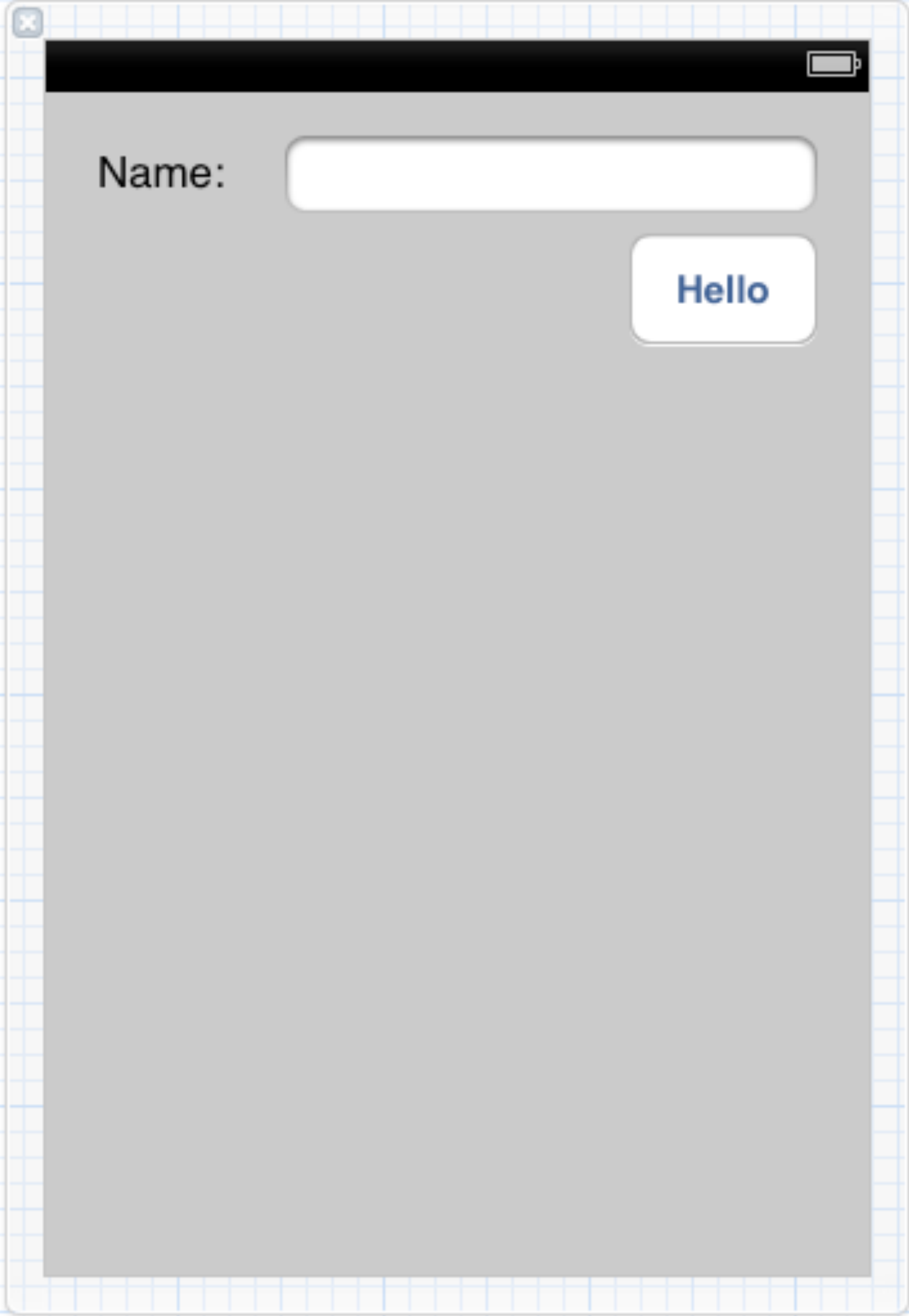
關閉虛擬鍵盤 (Deactivate Virtual Keyboard)

目的
針對UITextField和UITextView把虛擬鍵盤收起來的方法的整理。

環境
- iOS 6.1
- Xcode 4.6.2

UITextField
寫一個single view application來做實驗，xib的版面如下圖



接著在ViewController.h加入一個IBOutlet和一個IBAction分別和UITextField和UIButton設定link

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

```
@interface ViewController : UIViewController {
    IBOutlet UITextField* nameField;
}
-(IBAction) sayHello;

@end
```

- 按button的事件時將虛擬鍵盤關閉

```
// 方法一
-(void)sayHello {
    NSLog( @"Hello, %@", [nameField text] );
    [nameField resignFirstResponder];
}
```

- 點app背景時關閉

```
// 方法二: view上的event
-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {
    [nameField resignFirstResponder];
}
```

- 實作UITextFieldDelegate並註冊到view上的UITextField

ViewController.h

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

@interface ViewController : UIViewController <UITextFieldDelegate> {
    IBOutlet UITextField* nameField;
}
-(IBAction) sayHello;

@end
```

Partial of ViewController.m

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a nib.
    [self initView];
}

// 方法三: 實作UITextFieldDelegate，將虛擬鍵盤右下角改為「完成」，實作delegate method，註冊處理的delegate
-(void)initView {
    // 把虛擬鍵盤右下角改為完成
    // nameField.returnKeyType = UIReturnKeyDone;
    [nameField setReturnKeyType:UIReturnKeyDone];

    // 註冊nameField的event delegate
    // nameField.delegate = srlf;
    [nameField setDelegate:self];
}

-(BOOL)textFieldShouldReturn:(UITextField *)textField {
    NSLog( @"textFieldShouldReturn called" );
    [nameField resignFirstResponder];
    return YES;
}
```

UITextView

因為這個control是多行輸入，因此虛擬鍵盤上returnKeyType不會是完成。所以能繼續用的是按到view那招，或者另一招是加上UINavigationControler。

- 點到View Controller容器

```
-(void)touchesEnded:(NSSet *)touches withEvent:(UIEvent *)event {
    [memoTextView resignFirstResponder];
}
```

- 使用UINavigationController

這邊比較複雜，UINavigationController必須在UIWindow初始化app第一個view controller的時候就加進去。也就是說按照Single View Application來說，雖然我們在ViewController裡面可以看到一個navigationControl的property，但是這個property是readonly的，在UIWindow初始化view controller之後就無法設定了。就是我們要對Single View Application的AppDelegate下列這段程式動手腳。

AppDelegate.m修改前

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]] autorelease];
    // Override point for customization after application launch.
    self.viewController = [[[ViewController alloc]
initWithNibName:@"ViewController" bundle:nil] autorelease];
    self.window.rootViewController = self.viewController;
    [self.window makeKeyAndVisible];
    return YES;
}
```

解釋之前先看一下Supporting Files這個group裡的main.m這裡面有所有程式entry point的main程式。

main.m

```
int main(int argc, char *argv[])
{
    @autoreleasepool {
        return UIApplicationMain(argc, argv, nil,
NSStringFromClass([AppDelegate class]));
    }
}
```

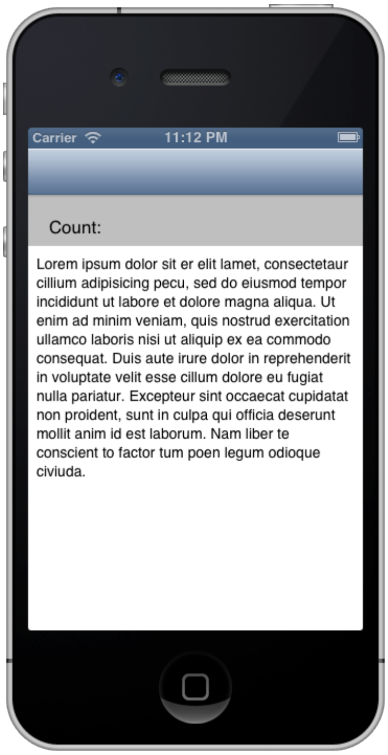
在iOS應用程式就是使用UIApplicationMain載入AppDelegate。回到AppDelegate class看到application載入第一個事件這裡會初始化UIWindow並設定UIWindow物件的rootViewController。修改的策略**就是把rootViewController換成UINavigationController，然後由UINavigationController帶起來我們app第一頁的ViewController。**

AppDelegate.m修改後

```
-(void)initApp {
    UINavigationController* nav = [[UINavigationController
alloc]initWithRootViewController:self.viewController];

    self.window.rootViewController = nav;

    // 記得要release我們自己用alloc/initxxx建立的物件
    [nav release];
}
```

```
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.window = [[[UIWindow alloc] initWithFrame:[[UIScreen
mainScreen] bounds]] autorelease];
    // Override point for customization after application launch.
    self.viewController = [[[ViewController alloc]
initWithNibName:@"ViewController" bundle:nil] autorelease];

    // 如果要用navigation control 一開始rootViewController不能用這個
viewController
    //self.window.rootViewController = self.viewController;
    // for navigation control
    [self initApp];

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

然後就可以在模擬器執行看看application，可以看到原本的元件上方多出一塊Navigation Controller，這樣就設定成功。接下來就可以開始使用navigation controller



Lorem ipsum dolor sit er elit lamet, consectetaur cillium adipisicing pecu, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum. Nam liber te conscient to factor tum poen legum odioque civiuda.

接下來就可以處理Navigation Controller，我們希望使用者剛看到第一頁的時候不出現Navigation Controller，而在點選UITextView的時候出現Navigation Controller，而右邊出現「完成」的按鈕，使用者按下去的時候就把虛擬鍵盤關掉。

因為要處理UITextView的事件，所以就讓ViewController實作UITextViewDelegate protocol。

ViewController.h

```
@interface ViewController : UIViewController <UITextViewDelegate> {
    IBOutlet UILabel* countLabel;
    IBOutlet UITextView* memoTextView;
}

@end
```

ViewController.m 程式碼片段

```
-(void)textViewDidBeginEditing:(UITextView *)textView {
    // 顯示navigation bar
    //self.navigationController.navigationBarHidden = NO;
    //設定是否動畫處理
    [self.navigationController setNavigationBarHidden:NO animated:YES];
    // self.navigationControl is readonly 要比view control早建立 => 在
AppDelegate那邊就要設定
    NSLog( @"%@", self.navigationController ); // 沒有先initial的話 這個會
是 <--- null
    // 在Navigation control加上UIBarButtonItem <- 「完成」
    UIBarButtonItem *doneButton = [[UIBarButtonItem alloc]
initWithBarButtonSystemItem:UIBarButtonSystemItemDone target:self
action:@selector(doneHandler)];

    self.navigationItem.rightBarButtonItem = doneButton;

    [doneButton release];
}

-(void)doneHandler {
    NSLog( @"done..." );
    // 關閉鍵盤
    [memoTextView resignFirstResponder];
    // 關閉navigation bar button，有做沒做都可以，因為下一行程式就把navigation
control隱藏起來了
    self.navigationItem.rightBarButtonItem = nil;
    //self.navigationController.navigationBarHidden = YES;
    [self.navigationController setNavigationBarHidden:YES animated:YES];
}

-(void)initView {
    //隱藏navigation bar
    // 這樣的效果是直接隱藏
    //self.navigationController.navigationBarHidden = YES;
    // 這樣的效果是隱藏時有動態效果
    [self.navigationController setNavigationBarHidden:YES animated:NO];

    //[memoTextView setDelegate:self];
    memoTextView.delegate = self;
}

- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view, typically from a
nib.
    [self initView];
}
```