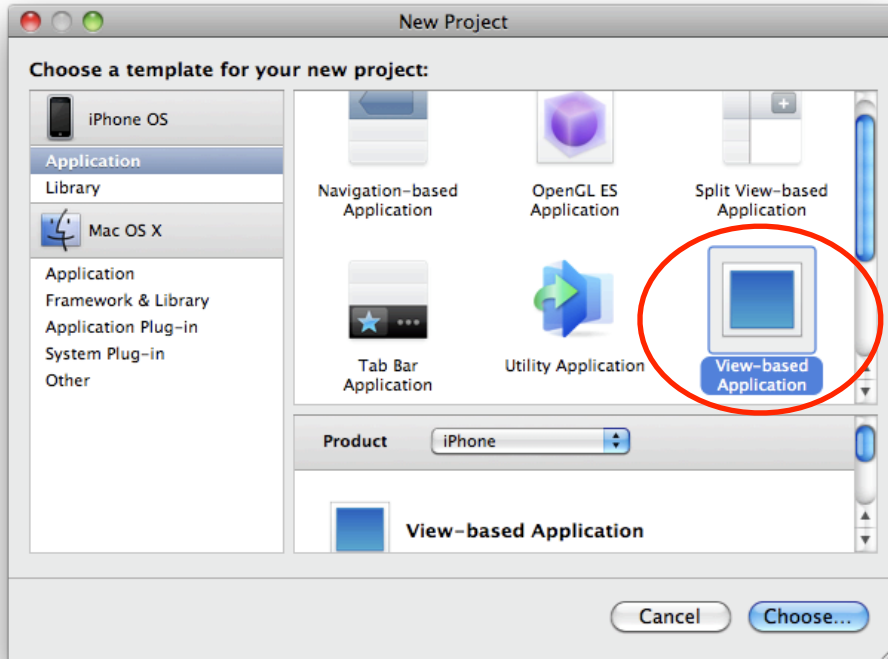


Lab WebBrowser

Step1. 在File開啓一個新的project, 選擇 view based application, 將project命名為 WebBrowser



Step2. 在WebBrowserViewController.h裡, 加入 <UIWebViewDelegate> 來遵循 UIWebView 的protocol, 並且加入View上對應的物件以及我們會使用到的變數,還有三個 Button物件對應的IBAction

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

```
@interface WebBrowserViewController : UIViewController
<UIWebViewDelegate>{
    IBOutlet UIWebView *webView;
    IBOutlet UIButton *go;
    IBOutlet UIButton *back;
    IBOutlet UIButton *forward;
    IBOutlet UITextField *address;
    NSURL *url;
    UIActivityIndicatorView *spinner;
    UILabel *loadingLabel;
}
```

```
-(IBAction) pressGo;
-(IBAction) pressForward;
-(IBAction) pressBack;
@end
```

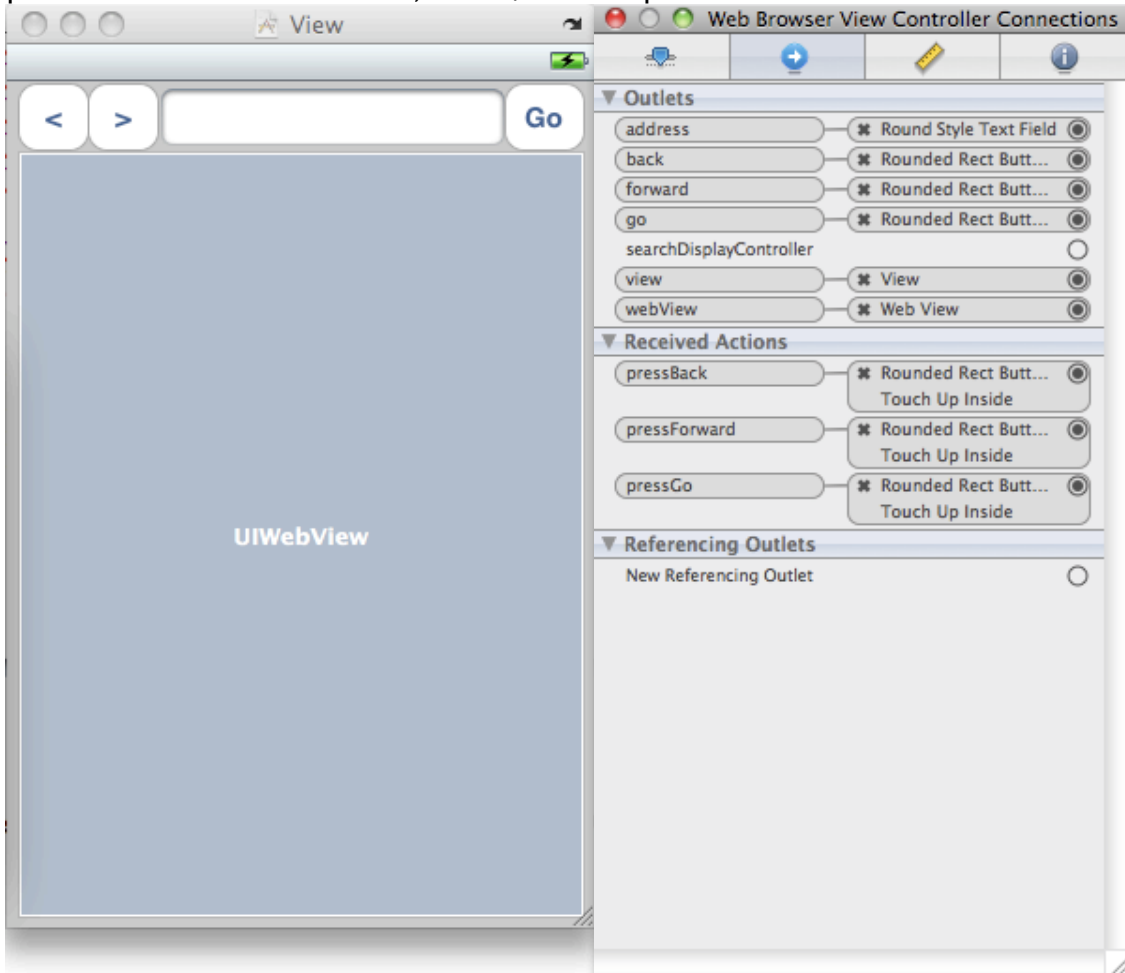
Step3. 在Resources > 開啓 WebBrowserViewController.xib, 裡,分別在左上方以及右上方加入三個Button,分別將Title命名為“<”,“>”,以及“Go”,以及在正上方加入一個UITextField,下方剩餘的空間加入一個UIWebView,並且將對應的物件還有IBAction做連結.

address連到正上方的UITextField, back 連到左上方左邊的Button, forward 連到左上方右邊的Button, go 連到右上方的Button, webView連到下方的UIWebView

pressBack連到左上方左邊的Button, 並選擇Touch Up Inside

pressForward連到左上方右邊的Button, 並選擇Touch Up Inside

pressGo連到右上方的Button, 並選擇Touch Up Inside



Step4. 在WebBrowserViewController.m裡, 加入三個Button對應的IBAction pressGo{}, pressForward{}, 以及 pressBack{}

```
#import "WebBrowserViewController.h"
```

```
@implementation WebBrowserViewController
```

```
-(IBAction) pressGo{
    NSLog(@"Go");
    url = [NSURL URLWithString:address.text];
    [webView loadRequest:[NSURLRequest requestWithURL:url]];
    [address endEditing:YES];
}
```

```

-(IBAction) pressForward{
    NSLog(@"Forward");
    [webView goForward];
}

-(IBAction) pressBack{
    NSLog(@"Back");
    [webView goBack];
}

```

Step5. 同樣的在WebBrowserViewController.m裡, 將 ViewDidLoad 這個Method的Mark去掉, 加入webView的預設值和Delegation

```

// Implement viewDidLoad to do additional setup after loading the
view, typically from a nib.
- (void)viewDidLoad {
    [super viewDidLoad];
    webView.delegate = self;
    webView.scalesPageToFit = YES;
}

```

Step6. 同樣的在WebBrowserViewController.m裡,

加入 - (void)webViewDidStartLoad:(UIWebView *)webView{}

以及 - (void)webViewDidFinishLoad:(UIWebView *)webView{}

來當網頁正在Loading時,顯示Loading字樣,當網頁Loading完成時,關閉Loading字樣

```

- (void)webViewDidStartLoad:(UIWebView *)webView{
    if (!spinner) {
        spinner = [[UIActivityIndicatorView alloc]
initWithActivityIndicatorStyle:UIActivityIndicatorViewStyleGray];

        loadingLabel = [[UILabel alloc] init];
        loadingLabel.text = @"Loading...";
        loadingLabel.textColor = [UIColor grayColor];
        [loadingLabel sizeToFit];

        CGFloat totalWidth = spinner.frame.size.width +
loadingLabel.frame.size.width;

        CGRect frame1 = spinner.frame;
        frame1.origin.x = (self.view.bounds.size.width - totalWidth) / 2.0;
        frame1.origin.y = (self.view.bounds.size.height -
frame1.size.height) / 2.0;

        CGRect frame2 = loadingLabel.frame;
        frame2.origin.x = (self.view.bounds.size.width - totalWidth) / 2.0 +
frame1.size.width;
        frame2.origin.y = (self.view.bounds.size.height -
frame2.size.height) / 2.0;

```

```

        spinner.frame = frame1;
        loadingLabel.frame = frame2;

        [spinner startAnimating];
        [self.view addSubview:spinner];
        [self.view addSubview:loadingLabel];
    }
}

- (void)webViewDidFinishLoad:(UIWebView *)webView{
    if (spinner) {
        [spinner stopAnimating];
        [spinner removeFromSuperview];
        [spinner release];
        spinner = nil;

        [loadingLabel removeFromSuperview];
        [loadingLabel release];
        loadingLabel = nil;
    }
}

```

Step7. Simulation (command + enter)



