

Switch View With Curve Animation

- Press `⌘⇧N`, select *Window-Based Application* and type in a project name of *View Switcher*.
- Click the *Classes* folder, type `⌘N`, select *UIViewController subclass*, click *Next*,
- Name it *SwitchViewController.m*, make sure "*SwitchViewController.h*" is checked
 - Repeat twice more to create *BlueViewController.m* and *YellowViewController.m*
- Click the *Resources* folder, type `⌘N`, select *User Interfaces*, select *View XIB*, save as *BlueView.xib*.
 - Repeat the steps to create a second nib file called *YellowView.xib*.

- Make the following changes to **View_SwitcherAppDelegate.h**

```
#import <UIKit/UIKit.h>
@class SwitchViewController;
@interface View_SwitcherAppDelegate : NSObject <UIApplicationDelegate> {
    IBOutlet UIWindow *window;
    IBOutlet SwitchViewController *switchViewController;
}
@property (nonatomic, retain) UIWindow *window;
@property (nonatomic, retain) SwitchViewController *switchViewController;
@end
```

The IBOutlet declaration is an outlet that will point to our root controller. Now when we're in Interface Builder and add an instance of the SwitchViewController class to *MainWindow.xib*, we'll already have an outlet to connect it to.

- Click **View_SwitcherAppDelegate.h** to add the root controller's view to app's main window

```
#import "View_SwitcherAppDelegate.h"
#import "SwitchViewController.h"
@implementation View_SwitcherAppDelegate
@synthesize window;
@synthesize switchViewController;
- (void)applicationDidFinishLaunching:(UIApplication *)application {

    // Override point for customization after app launch
    [window addSubview:switchViewController.view];
    [window makeKeyAndVisible];
}
- (void)dealloc {
    [window release];
    [switchViewController release];
    [super dealloc];
}
@end
```

We implemented the switchViewController outlet and added the root controller's view to the window.

- Add outlets or actions to the **SwitchViewController.h**

```
#import <UIKit/UIKit.h>
@class BlueViewController;
@class YellowViewController;
@interface SwitchViewController : UIViewController {
    YellowViewController *yellowViewController;
    BlueViewController *blueViewController;
}
@property (retain, nonatomic) YellowViewController *yellowViewController;
@property (retain, nonatomic) BlueViewController *blueViewController;
-(IBAction)switchViews:(id)sender;
@end
```

- Write **SwitchViewController.m** to switch between the yellow view and the blue view with flip animation.

```
#import "SwitchViewController.h"
#import "BlueViewController.h"
#import "YellowViewController.h"
@implementation SwitchViewController
@synthesize blueViewController;
@synthesize yellowViewController;
```

```

- (void)viewDidLoad
{
    BlueViewController *blueController = [[BlueViewController alloc]
initWithNibName:@"BlueView"
                                bundle:nil];

    self.blueViewController = blueController;
    [self.view addSubview:blueController.view atIndex:0];
    [blueController release];
}
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
    if (self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil]) {
        // Initialization code
    }
    return self;
}
- (IBAction)switchViews:(id)sender
{
    if (self.yellowViewController == nil)
    {
        YellowViewController *yellowController =
[[YellowViewController alloc] initWithNibName:@"YellowView"
                                bundle:nil];
        self.yellowViewController = yellowController;
        [yellowController release];
    }

    [UIView beginAnimations:@"View Flip" context:nil];
    [UIView setAnimationDuration:1.25];
    [UIView setAnimationCurve:UIViewAnimationCurveEaseInOut];

    UIViewController *coming = nil;
    UIViewController *going = nil;
    UIViewAnimationTransition transition;

    if (self.blueViewController.view.superview == nil)
    {
        coming = blueViewController;
        going = yellowViewController;
        transition = UIViewAnimationTransitionCurlUp;
    }
    else
    {
        coming = yellowViewController;
        going = blueViewController;
        transition = UIViewAnimationTransitionCurlDown;
    }

    [UIView setAnimationTransition:transition forView:self.view cache:YES];

    [coming viewWillAppear:YES];
    [going viewWillDisappear:YES];
    [going.view removeFromSuperview];
    [self.view addSubview:coming.view atIndex:0];
    [going viewDidDisappear:YES];
    [coming viewDidAppear:YES];

    [UIView commitAnimations];
}
- (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation {
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}
- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning]; // Releases the view if it doesn't have a superview
    // Release anything that's not essential, such as cached data
    if (self.blueViewController.view.superview == nil)
        self.blueViewController = nil;
    else
        self.blueViewController = nil;
}

```

```

- (void)dealloc {
    [yellowViewController release];
    [blueViewController release];
    [super dealloc];
}
@end

```

- Open **MainWindow.xib** drag a *View Controller* from the library to your nib's main window
 - This allows us to add an instance of *SwitchViewController* to *MainWindow.xib*.
 - We now have five icons, and a new window containing a dashed, grey *View*.
 - We added an instance of *UIViewController*, but we need an instance of *SwitchViewController*,
- Change the *Class* to *SwitchViewController*.
 - Now the *switchViews:* action method should appear in the section labeled *Class Actions*.
 - Note the nib's main window, icon has switched from *View Controller* to *Switch View Controller*.

We now need to build our root controller's view (SwitchViewController's) to switch between the blue view and the yellow view.

- Drag a *View* from the library onto the window with a grey background that says *View*.
 - The grey background is replaced by this new view.
- Drag a toolbar onto your view, and place it at the bottom
- Double-click the button, and change its title to *Switch Views*.

Connect the instance of Switch Views button to the Switch ViewController.

- Control-drag from the *Switch Views* button to the *Switch ViewController* icon, & select *switchViews:* action.
 - If the *switchViews:* action doesn't pop up and instead you see an outlet called *delegate*, you've most likely control-dragged from the toolbar rather than the button. To fix it, just make sure you've got the button and not the toolbar selected and redo your control-drag.

Connect the instance of SwitchViewController in our nib to that outlet.

- Control-drag: *View_Switcher App Delegate* icon to *Switch View Controller* icon, select *switchViewController* outlet.

Implement the Content Views

- Open **BlueViewController.h** add the following declaration

```

#import <UIKit/UIKit.h>
@interface BlueViewController : UIViewController {
}
-(IBAction)blueButtonPressed:(id)sender;
@end

```

- Open **YellowViewController.h** add the following declaration

```

#import <UIKit/UIKit.h>
@interface BlueViewController : UIViewController {
}
-(IBAction)yellowButtonPressed:(id)sender;
@end

```

- Open **BlueView.xib** Change *File's Owner* icon to *BlueViewController*
- change the background color of this view to a blue.
- Drag a *Round RectButton* from the library over to the window.
- Double-click the button and change its title to *Press Me*. You can place the button anywhere that looks good to you.
- Switch to the connections inspector (by pressing **⌘2**),
- drag from the *Touch Up Inside* event to the *File's Owner* icon, and connect to the *blueButtonPressed:* action method.
- Control-drag from the *File's Owner* icon to the *View* icon, and select the view outlet to do that. Save the nib, Open **YellowView.xib** make almost the same exact changes to this nib file.
 - change the file's owner from *NSObject* to *YellowViewController* using the identity inspector.
 - change the view's height to 416 pixels using the size inspector
 - change the view's background to a nice yellow color using the attributes inspector.
 - add a round rectangular button to this view
 - Give it a label of *Press Me, Too*,
 - connect that button's *Touch Up Inside* event to the *yellowButtonPressed:* action method in *File's Owner*.

- control-drag from the File's Owner icon to the View icon, and connect to the view outlet.
- save the nib, and go back to Xcode.

- Add the following code to **BlueViewController.m**

```

@implementation BlueViewController
- (IBAction)blueButtonPressed:(id)sender
{
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Blue View Button
Pressed" message:@"You pressed the button on the blue view" delegate:nil
cancelButtonTitle:@"Yep, I did." otherButtonTitles:nil];
    [alert show];
    [alert release];
}
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
...

```

- Add the following code to **YellowViewController.m**

```

#import "YellowViewController.h"
@implementation YellowViewController
- (IBAction)yellowButtonPressed:(id)sender
{
    UIAlertView *alert = [[UIAlertView alloc] initWithTitle:@"Yellow View Button
Pressed" message:@"You pressed the button on the yellow view" delegate:nil
cancelButtonTitle:@"Yep, I did." otherButtonTitles:nil];
    [alert show];
    [alert release];
}
- (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil {
...

```