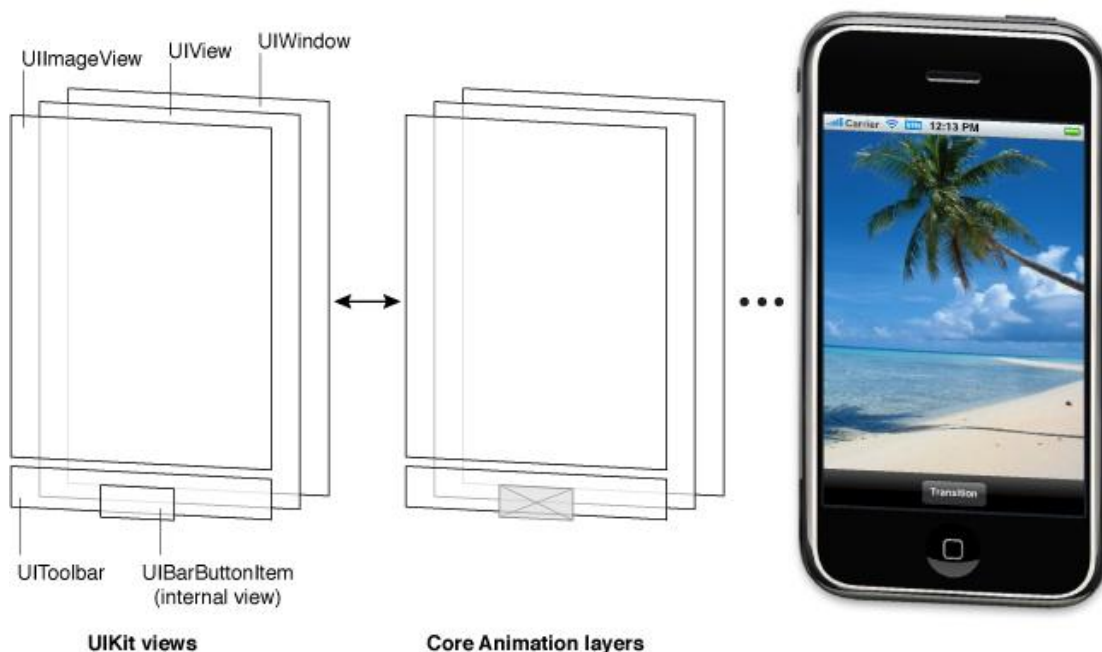


Iphone 开发（九）理解 subView，手动实现多个视图切换

在 Iphone 的视图中，其实就是一个一个 view，一层 view 上面放一层 view，一个 view 上面放一群 view，甚至 UIWindow 也是一个 view，在网上找了一张图片很能说明这个问题：



可见我们能够看到的都是一个 view 视图，而我们能对其进行操作，是因为 UINavigationController 和 UIView 都是 UIResponder 的子类。这时我们对视图进行操作时需要掌握几个比较重要的概念和几个常用的方法。一个是 superView 和 subView 的概念，一个是 UIView 和 UIControl 类对象的区别。

当我们生成一个独立的 view 视图时，往往是新建一个 UIViewController 的文件并伴随一个 xib 文件，这个 xib 文件中的 file's owner 肯定是这个 UIViewController 类了，而它的 Objects 一般就是一个 UIView 对象（一张干净的画布），我们往往可以将这个 UIView 改为 UIControl，通过修改它的 Custom Class 项来实现，这时这一大张画布就可以像它上面那些 button 之类的控件一样来响应方法了，我们在隐藏软键盘时就用到了这一点。

而 superView 和 subView 的概念更好理解，view 上可以放控件，那么这个 view 就是这些控件的 superView。而 UIWindow 上可以叠一层又一层的 view，UIWindow 就是这些 view 的 superView。先放上去的 index 为 0，然后依次累加。处理 superView 和 subView 往往用到几个方法：

`removeFromSuperview;`//调用者为 subView

`insertSubview:atIndex;`//调用者为 superView

了解了这些我们来看一个实例，很简单，实现在根视图控制器上添加两个 **view**（都是整页覆盖屏幕的），然后点击屏幕分别切换显示。

新建一个项目，然后添加两个 **UIViewController** 类并附带 **xib** 文件，分别取名为 **FirstViewController,SecondViewController**;在 **viewController** 中进行如下操作:

viewController.h:

[plain] [view_plaincopy](#)

```
1. #import <UIKit/UIKit.h>
2. #import "FirstViewController.h"
3. #import "SecondViewController.h"
4. @interface ViewController : UIViewController
5. @property (retain ,nonatomic) FirstViewController *firstViewController;
6. @property (retain ,nonatomic) SecondViewController *secondViewController;
7. @end
```

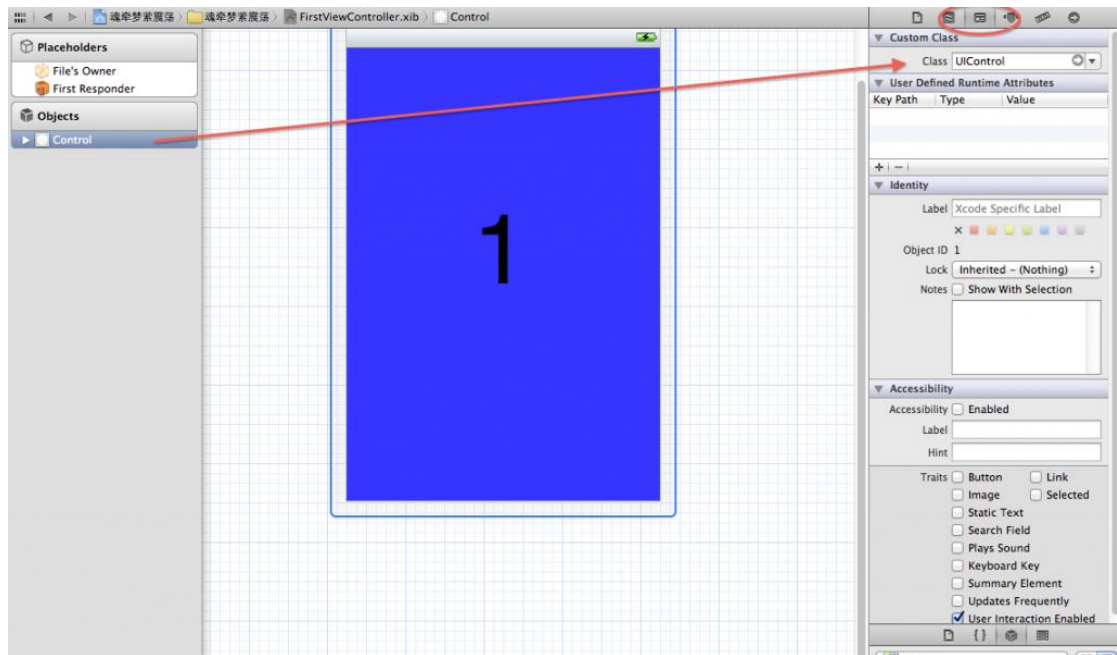
viewController.m:

[plain] [view_plaincopy](#)

```
1. #import "ViewController.h"
2.
3.
4. @implementation ViewController
5. @synthesize firstViewController;
6. @synthesize secondViewController;
7.
8.
9. - (void)viewDidLoad
10. {
```

```
11.    [super viewDidLoad];
12.    firstViewController = [[FirstViewController alloc] initWithNibName:@"FirstViewController" bundle:nil];
13.    secondViewController = [[SecondViewController alloc] initWithNibName:@"SecondViewController" bundle:nil];
14.
15.    [self.view addSubview:firstViewController.view];
16.    [self.view addSubview:secondViewController.view];
17.
18. }
19.
20. - (void)viewDidUnload
21. {
22.    [super viewDidUnload];
23.    // Release any retained subviews of the main view.
24. }
25.
26. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)interfaceOrientation
27. {
28.    return (interfaceOrientation != UIInterfaceOrientationPortraitUpsideDown);
29. }
30.
31. @end
```

以上代码实现了在 `viewDidLoad` 中声明两个对象，然后添加到主视图上，这样在打开模拟器时会看到 `secondView` 的视图（后添加的在上面，`index` 为 1）；然后在 `FirstViewController.xib` 文件中将 View 的 Custom Class 发为 `UIControl`，背景改一下，贴个 label 内容为 1；



在 FirstViewController.h 中添加一个输出口：

[plain] [view plaincopy](#)

```
1. #import <UIKit/UIKit.h>
2.
3. @interface FirstViewController : UIViewController
4. -(IBAction)clickH:(id)sender;
5. @end
```

在 FirstViewController.m 中进行如下操作：

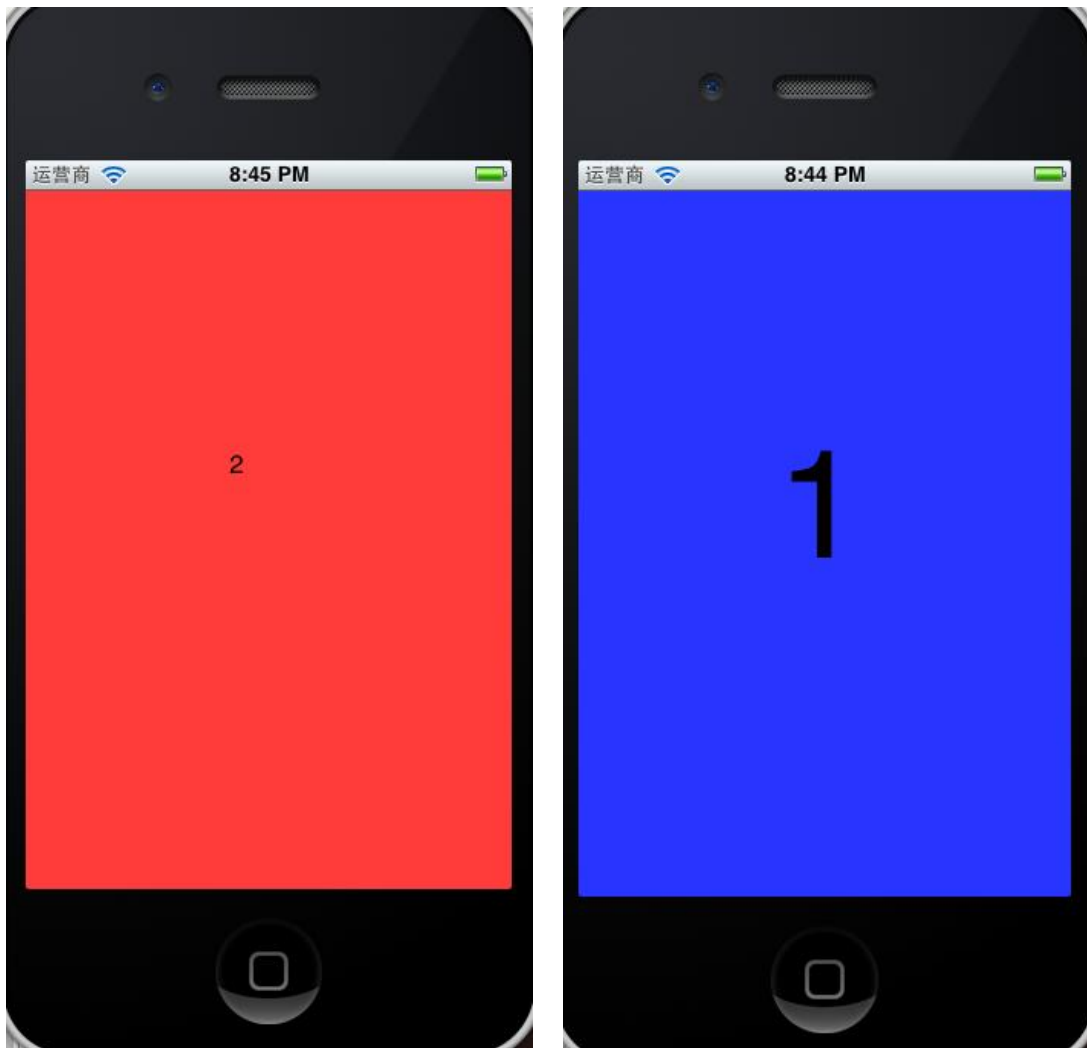
[plain] [view plaincopy](#)

```
1. #import "FirstViewController.h"
2.
3. @interface FirstViewController ()
4.
5. @end
6.
7. @implementation FirstViewController
8. -(IBAction)clickH:(id)sender
```

```
9. {
10.
11.     [self.view.superview insertSubview:self.view atIndex:0];
12.     //将当前的 view 放到最底部。
13. }
14. - (id)initWithNibName:(NSString *)nibNameOrNil bundle:(NSBundle *)nibBundleOrNil
    rNil
15. {
16.     self = [super initWithNibName:nibNameOrNil bundle:nibBundleOrNil];
17.     if (self) {
18.         // Custom initialization
19.     }
20.     return self;
21. }
22.
23. - (void)viewDidLoad
24. {
25.     [super viewDidLoad];
26.     // Do any additional setup after loading the view from its nib.
27. }
28.
29. - (void)viewDidUnload
30. {
31.     [super viewDidUnload];
32.     // Release any retained subviews of the main view.
33.     // e.g. self.myOutlet = nil;
34. }
35.
36. - (BOOL)shouldAutorotateToInterfaceOrientation:(UIInterfaceOrientation)inter
    faceOrientation
```

```
37. {  
38.     return (interfaceOrientation == UIInterfaceOrientationPortrait);  
39. }  
40.  
41. @end
```

这样就利用 **insert** 方法在点击该视图时将其放入底部，另一个视图就显示出来了，在 **secondViewController** 中进行同样的操作，就会实现初始化后点击屏幕来回切换的效果。当然，这是在两个视图的情况下，只是两个不停的换位置的效果，**demo** 用于理解方法，如果是多视图大应用的话常常要进行 **remove** 操作，这样会提高效率。



关键字: IOS , Iphone 开发 , 基础 , 入门 ,
superView ,subView ,insertSubview , removeFromSuperview , 手动多视图
切换