**iOS开发UI篇—UIWindow简单介绍**

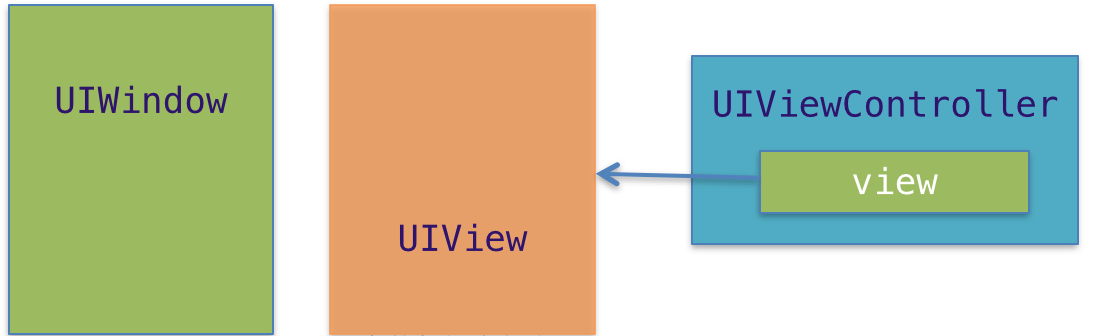
**一、简单介绍**

UIWindow是一种特殊的UIView，通常在一个app中只会有一个UIWindow

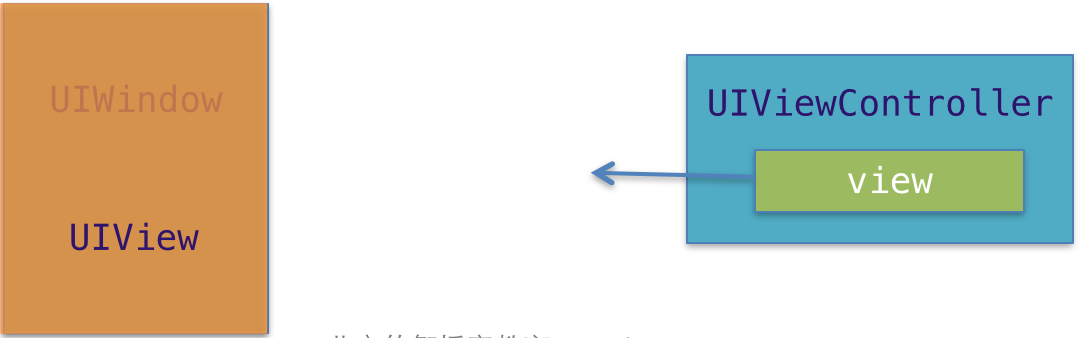
iOS程序启动完毕后，创建的第一个视图控件就是UIWindow，接着创建控制器的view，最后将控制器的view添加到UIWindow上，于是控制器的view就显示在屏幕上了

一个iOS程序之所以能显示到屏幕上，完全是因为它有UIWindow。也就说，没有UIWindow，就看不见任何UI界面

补充：UIWindow是创建的第一个视图控件（创建的第一个对象是UIapplication）如下图：

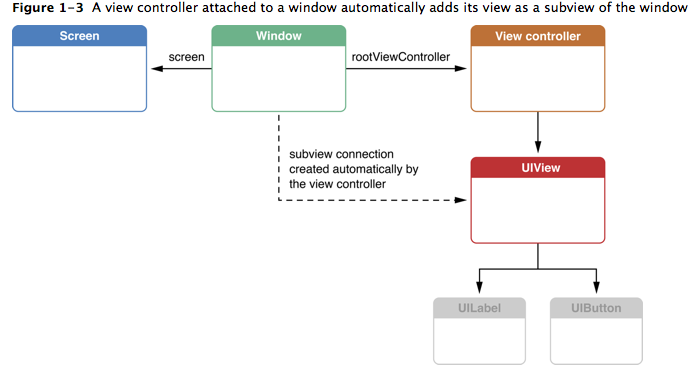


添加



先创建UIwindow,再创建控制器，创建控制器的view，然后将控制器的view添加到UIWindow上。

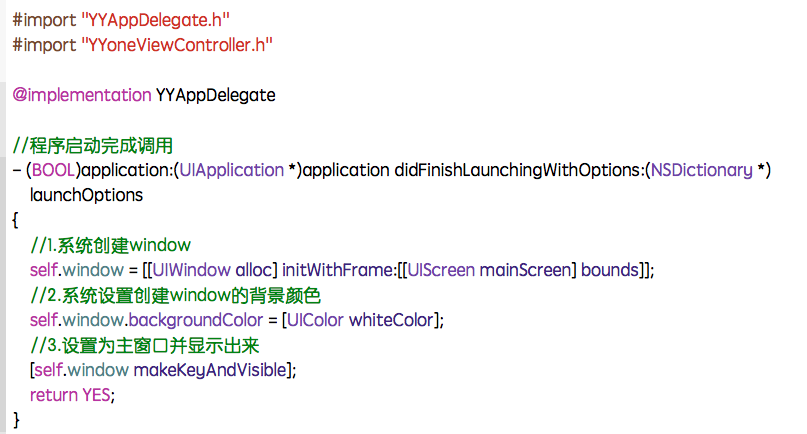
文档中关于该部分的解释：



**二、**UIWindow的创建过程

1.简单说明

创建一个空的项目，就可以看到UIWindow是怎么出来的了。在程序启动完毕之后就会调用一次，创建过程如下：



提示：应用程序启动之后，先创建Application，再创建它的代理，之后创建UIwindow。UIWindow继承自UIview。

2.把view添加到uiwindow

创建一个控制器，把view添加到uiwindow上面（有两种方式）

（1）直接将控制器的view添加到UIWindow中，并不理会它对应的控制器

[self.window  addsubview:vc.view];

（2）设置uiwindow的根控制器，自动将rootviewcontroller的view添加到window中，负责管理rootviewcontroller的生命周期

[self.window.rootviewcontroller=vc];

两个方法的区别：

以后的开发中，建议使用（2）.因为方法（1）存在一些问题，比如说控制器上面可能由按钮，需要监听按钮的点击事件，如果是1，那么按钮的事件应该由控制器来进行管理。但控制器是一个局部变量，控制器此时已经不存在了，但是控制器的view还在，此时有可能会报错。注意：方法执行完，这个控制器就已经不存在了。

问题描述1：当view发生一些事件的时候，通知控制器，但是控制器已经销毁了，所以可能出现未知的错误。

问题描述2：添加一个开关按钮，让屏幕360度旋转（两者的效果不一样）。当发生屏幕旋转事件的时候，UIapplication对象会将旋转事件传递给uiwindow,uiwindow又会将旋转事件传递给它的根控制器，由根控制器决定是否需要旋转

UIapplication->uiwindow->根控制器（第一种方式没有根控制器，所以不能跟着旋转）。

提示：不通过控制器的view也可以做开发，但是在实际开发中，不要这么做，不要直接把view添加到UIWindow上面去。因为，难以管理。

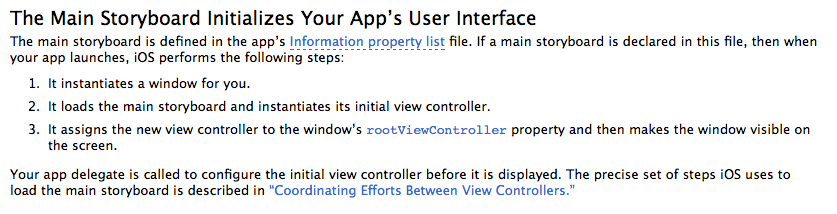
3.在有storyboard的项目中，UIWindow是如何创建的？

为什么创建一个storyboard，没有看到创建uiwindow的过程？

它其实是把创建UIWindow的过程给屏蔽起来了。可以把代理的UIWindow的属性的值打印出来NSLog(@“window=%p”,self.window);打印出来确实是有值的，说明确实创建了UIWindow.不仅创建了UIWindow，默认还创建了UIWindow对应的控制器，也可以打印进行查看。NSLog(@“%@“,self.window.rootviewcontroller);

有storyboard的项目中的创建过程：

当用户点击应用程序图标的时候，先执行Main函数，执行UIApplicationMain（）,根据其第三个和第四个参数创建Application，创建代理，并且把代理设置给application（看项目配置文件info.plist里面的storyboard的name，根据这个name找到对应的storyboard），开启一个事件循环，当程序加载完毕，他会调用代理的didFinishLaunchingWithOptions:方法。在调用didFinishLaunchingWithOptions:方法之前，会加载storyboard，在加载的时候创建一个window，接下来会创建箭头所指向的控制器，把该控制器设置为UIWindow的根控制器，接下来再将window显示出来，即看到了运行后显示的界面。（提示：关于这部分可以查看story的初始化的文档）



三、如何获取window?

1.主窗口和次窗口

【self.window makekeyandvisible】让窗口成为主窗口，并且显示出来。有这个方法，才能把信息显示到屏幕上。

   因为Window有makekeyandvisible这个方法，可以让这个Window凭空的显示出来，而其他的view没有这个方法，所以它只能依赖于Window，Window显示出来后，view才依附在Window上显示出来。

【self.window make keywindow】//让uiwindow成为主窗口，但不显示。

2.获取UIwindow

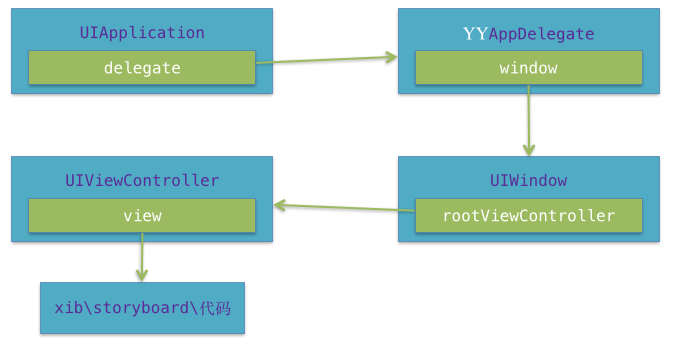
（1）[UIApplication sharedApplication].windows  在本应用中打开的UIWindow列表，这样就可以接触应用中的任何一个UIView对象(平时输入文字弹出的键盘，就处在一个新的UIWindow中)

（2）[UIApplication sharedApplication].keyWindow（获取应用程序的主窗口）用来接收键盘以及非触摸类的消息事件的UIWindow，而且程序中每个时刻只能有一个UIWindow是keyWindow。

提示：如果某个UIWindow内部的文本框不能输入文字，可能是因为这个UIWindow不是keyWindow

（3）view.window获得某个UIView所在的UIWindow

四、四大对象的关系图



五、主窗口和次窗口说明

代码：

[](javascript:void(0);)

// 程序启动完毕之后就会调用一次

- (BOOL)application:(UIApplication \*)application didFinishLaunchingWithOptions:(NSDictionary \*)launchOptions

{

// 1.创建UIWindow

self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];

// 设置UIWindow的背景颜色

self.window.backgroundColor = [UIColor redColor];

// 让UIWindow显示出来(让窗口成为主窗口 并且显示出来)

// 一个应用程序只能有一个主窗口

[self.window makeKeyAndVisible];

// 让UIWindow成为主窗口

// [self.window makeKeyWindow];

// 2. 再创建一个窗口

UIWindow \*w2 = [[UIWindow alloc] initWithFrame:CGRectMake(100, 100, 200, 200)];

w2.backgroundColor = [UIColor yellowColor];

[w2 makeKeyAndVisible];

self.w2 = w2;

// 3.创建两个文本输入框

// 3.1将文本输入框添加到window中

UITextField \*tx1 = [[UITextField alloc] initWithFrame:CGRectMake(10, 10, 200, 40)];

tx1.borderStyle = UITextBorderStyleRoundedRect;

[self.window addSubview:tx1];

// 3.2将文本输入框添加到w2中

UITextField \*tx2 = [[UITextField alloc] initWithFrame:CGRectMake(10, 10, 100, 40)];

tx2.borderStyle = UITextBorderStyleRoundedRect;

[self.w2 addSubview:tx2];

// 获取应用程序的主窗口

NSLog(@"%@", [UIApplication sharedApplication].keyWindow);

return YES;

}

[](javascript:void(0);)

代码说明：

再创建一个窗口（主窗口和次窗口的区别）

局部变量，需要定义一个Window属性来保存变量。

window的属性定义为strong，就是为了让其不销毁。

一个应用程序只能有一个主窗口，程序中创建了两个Window，那么谁是主窗口？后面的窗口能覆盖前面的窗口。

提示：如果UItextfield不显示，可以考虑设置它的样式，因为其创建默认是虚线的，没有边框。

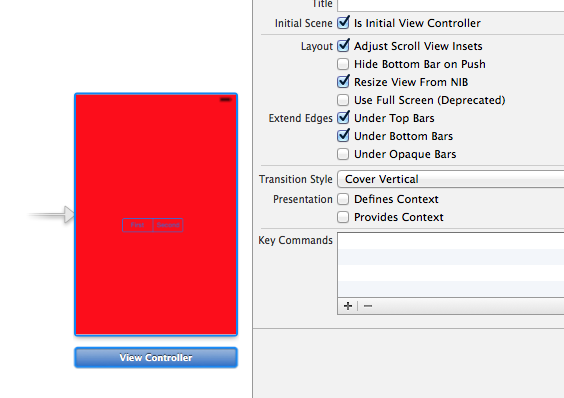
在ios7里边，主窗口和次窗口是没有区别的

在ios7以前中有区别：哪个是主窗口，后面设置为主窗口会把之前设置的覆盖掉。（只有主窗口才能响应键盘的输入事件，如果不能输入内容，可以查看是否是显示在主窗口上，不在主窗口上的不能响应。）

 六、补充说明

**在有storyboard中的创建过程：**

先执行Main函数，执行UIApplicationMain（）,根据其第三个和第四个参数创建Application，创建代理，并且把代理设置给application，根据项目配置文件info.plist里面的storyboard的name，找到对应的storyboard，接下来创建一个window，之后创建它的初始化控制器（就是箭头所指向的控制器），自动把该控制器设置为UIWindow的根控制器，接下来再将window显示出来，即看到了运行后显示的界面。



注意这个控制器属性面板上的“初始化控制器属性”。

**在没有storyboard中的创建过程：**

先执行Main函数，执行UIApplicationMain（）,根据其第三个和第四个参数创建Application，创建代理，并且把代理设置给application，开启一个事件循环，当程序加载完毕，他会调用代理的didFinishLaunchingWithOptions:方法。在该方法中，会创建一个Window,然后创建一个控制器，并把该控制器设置为UIWindow的根控制器，接下来再将window显示出来，即看到了运行后显示的界面。