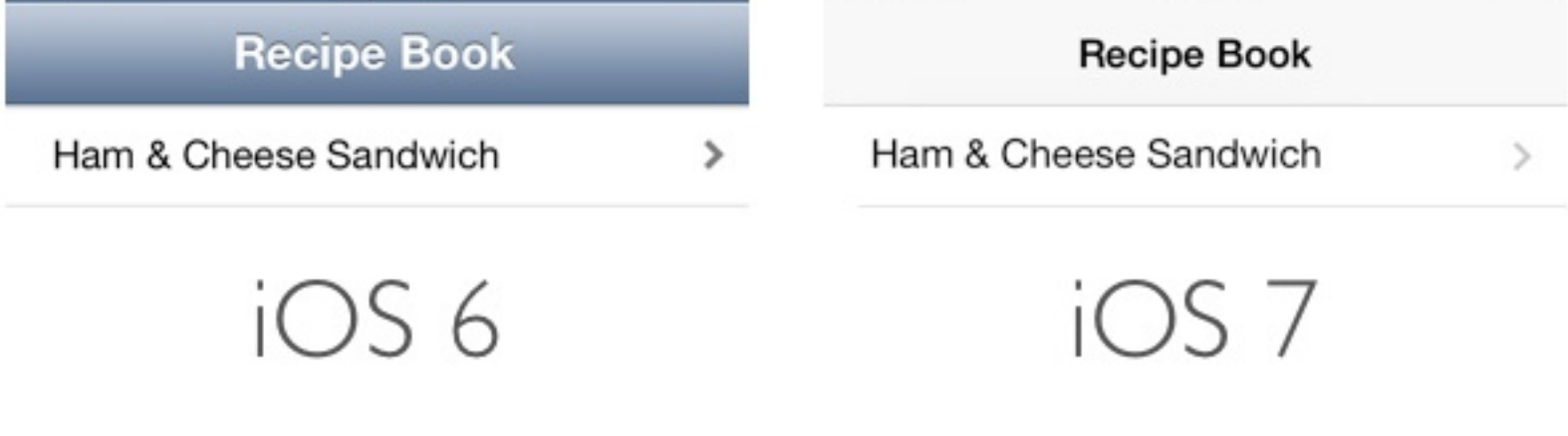


iOS 7中默认的导航栏

在开始定制之前，我们先来看看iOS 7中默认导航栏的外观。通过Xcode用Single View Controller模板创建一个工程。然后将view controller嵌入到一个navigation controller中。如果你不想从头开始，那么也可以在这里[下载到这个示例工程](#)。Xcode 5包含有iOS 6和iOS 7模拟器，我们可以在这两个不同的模拟器版本中运行示例程序，进行对比，如下图所示：



如上图所示，在iOS 7中的导航栏默认情况下跟状态栏是交织在一起的，并且它的颜色也被修改为亮灰色。

设置导航栏的背景颜色

在iOS 7中，不再使用tintColor属性来设置导航栏的颜色，而是使用barTintColor属性来修改背景色。我们可以在AppDelegate.m文件中的方法didFinishLaunchingWithOptions:里面添加如下代码来修改颜色：

```
1. [[UINavigationController appearance] setBarTintColor:[UIColor yellowColor]];
```

效果如下图所示：



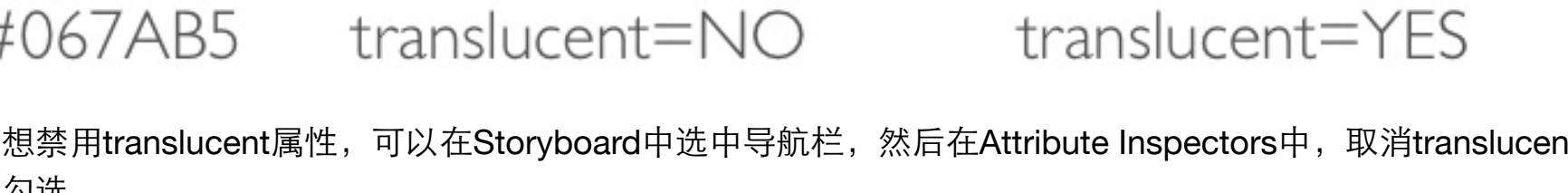
一般情况，我们都会使用自己的颜色，下面这个宏用来设置RGB颜色非常方便：

```
1. #define UIColorFromRGB(rgbValue) [UIColor colorWithRed:((float)((rgbValue & 0xFF0000) >> 16))/255.0 green:((float)((rgbValue & 0xFF00) >> 8))/255.0 blue:((float)((rgbValue & 0xFF))/255.0 alpha:1.0)]
```

将上面这个宏放到AppDelegate.m文件中，然后通过这个宏来创建一个UIColor对象(根据指定的RGB)。如下示例：

```
1. [[UINavigationController appearance] setBarTintColor:UIColorFromRGB(0x067AB5)];
```

默认情况下，导航栏的translucent属性为YES。另外，系统还会对所有的导航栏做模糊处理，这样可以让iOS 7中导航栏的颜色更加饱和。如下图，是translucent值为NO和YES的对比效果：



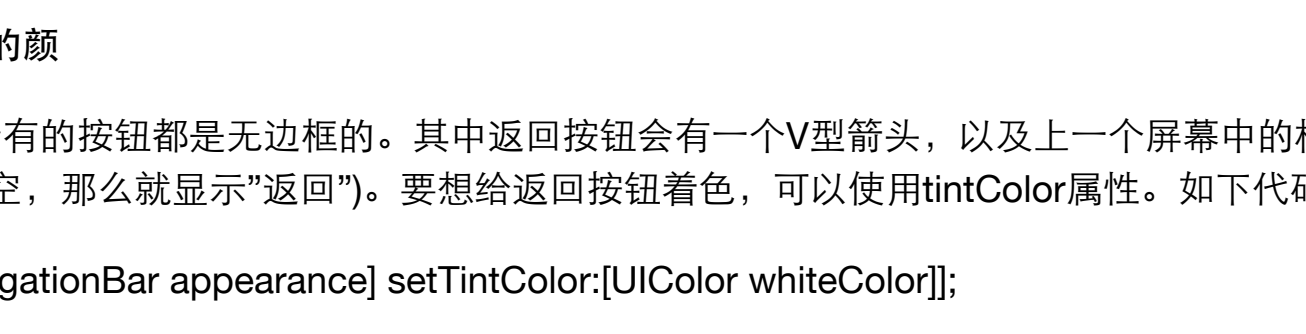
要想禁用translucent属性，可以在Storyboard中选中导航栏，然后在Attribute Inspectors中，取消translucent的勾选。

在导航栏中使用背景图片

如果希望在导航栏中使用一个图片当做背景，那么你需要提供一个稍微高一点图片(这样可以延伸到导航栏背后)。导航栏的高度从44 points(88 pixels)变为了64 points(128 pixels)。我们依然可以使用 setBackgroundImage:方法为导航栏设置自定义图片。如下代码所示：

```
1. [[UINavigationController appearance] setBackgroundImage:[UIImage imageNamed:@"nav_bg.png"] forBarMetrics:UIBarMetricsDefault];
```

示例工程中提供了两个背景图片：nav_bg.png 和 nav_bg_ios7.png。运行一下试试看吧，如下效果：

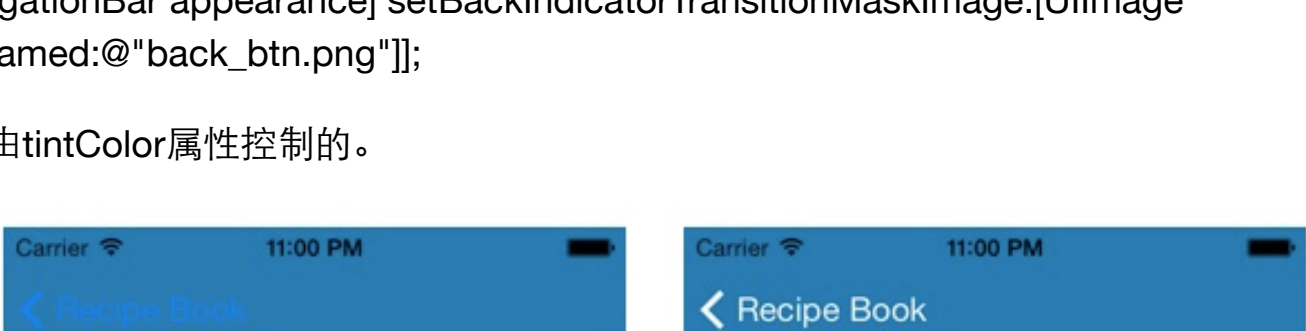


定制返回按钮的颜色

在iOS 7中，所有的按钮都是无边框的。其中返回按钮会有一个V型箭头，以及上一个屏幕中的标题(如果上一屏幕的标题是空，那么就显示“返回”)。要想给返回按钮着色，可以使用tintColor属性。如下代码所示：

```
1. [[UINavigationController appearance] setTintColor:[UIColor whiteColor]];
```

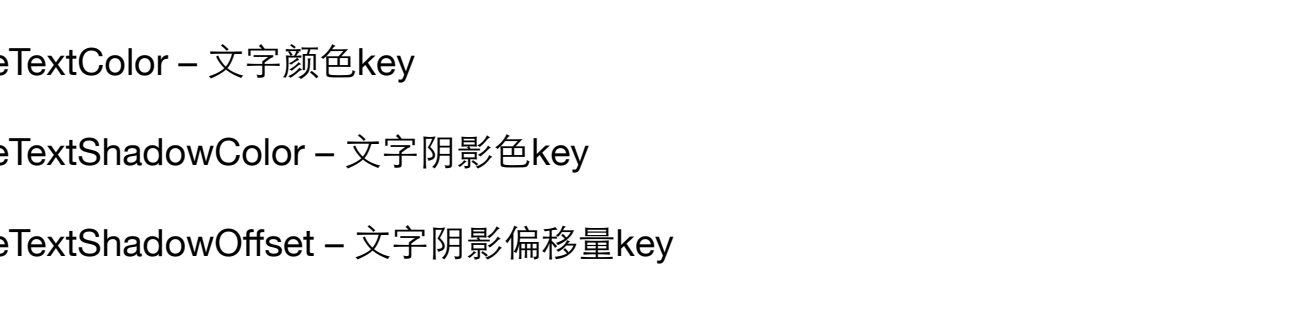
除了返回按钮，tintColor属性会影响到所有按钮标题和图片。



如果想要用自己的图片替换V型，可以设置图片的backIndicatorImage和backIndicatorTransitionMaskImage。如下代码所示：

```
1. [[UINavigationController appearance] setBackIndicatorImage:[UIImage imageNamed:@"back_btn.png"]];
2. [[UINavigationController appearance] setBackIndicatorTransitionMaskImage:[UIImage imageNamed:@"back_btn.png"]];
```

图片的颜色是由tintColor属性控制的。



修改导航栏标题的字体

跟iOS 6一样，我们可以使用导航栏的titleTextAttributes属性来定制导航栏的文字风格。在text attributes字典中使用如下一些key，可以指定字体、文字颜色、文字阴影色以及文字阴影偏移量：

UIFontAttributeFont – 字体key

UIColorAttributeTextColor – 文字颜色key

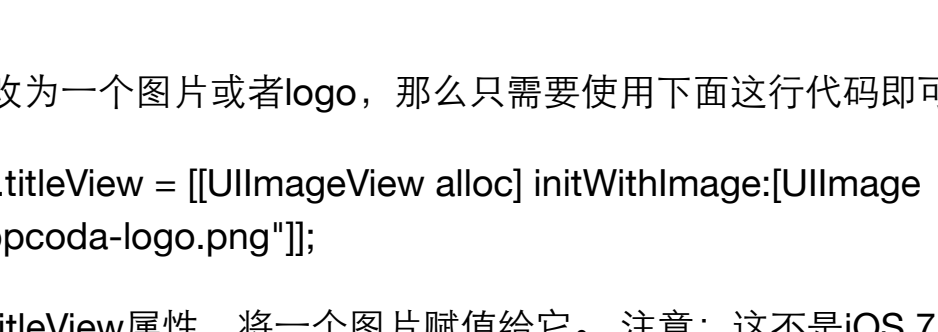
UIColorAttributeTextShadowColor – 文字阴影色key

UIColorAttributeTextShadowOffset – 文字阴影偏移量key

如下代码所示，对导航栏的标题风格做了修改：

```
1. NSShadow *shadow = [[NSShadow alloc] init];
2. shadow.shadowColor = [UIColor colorWithRed:0.0 green:0.0 blue:0.0 alpha:0.8];
3. shadow.shadowOffset = CGSizeMake(0, 1);
4. [[UINavigationController appearance] setTitleTextAttributes: [NSDictionary dictionaryWithObjectsAndKeys:
5. [UIColor colorWithRed:245.0/255.0 green:245.0/255.0 blue:245.0/255.0 alpha:1.0], NSForegroundColorAttributeName,
6. shadow, NSShadowAttributeName,
7. [UIFont fontWithName:@"HelveticaNeue-CondensedBlack" size:21.0], NSFontAttributeName, nil]];
```

运行效果如下图所示：

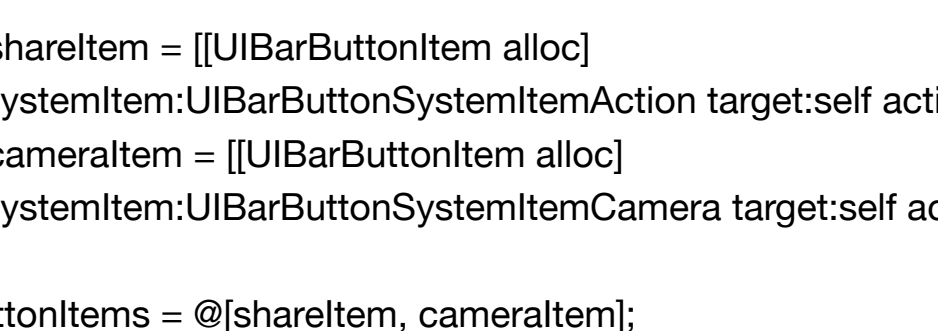


修改导航栏标题为图片

如果要想将导航栏标题修改为一个图片或者logo，那么只需要使用下面这行代码即可：

```
1. self.navigationItem.titleView = [[UIImageView alloc] initWithImage:[UIImage imageNamed:@"appcoda-logo.png"]];
```

上面的代码简单的修改了titleView属性，将一个图片赋值给它。注意：这不是iOS 7中的新功能，之前的iOS版本就可以已经有了。具体效果如下图所示：



添加多个按钮

同样，这个技巧也不是iOS 7的，开发者经常会在导航栏中添加多个按钮，所以我决定在这里进行介绍。我们可以在导航栏左边或者右边添加多个按钮。例如，我们希望在导航栏右边添加一个照相机和分享按钮，那只需要使用下面的代码即可：

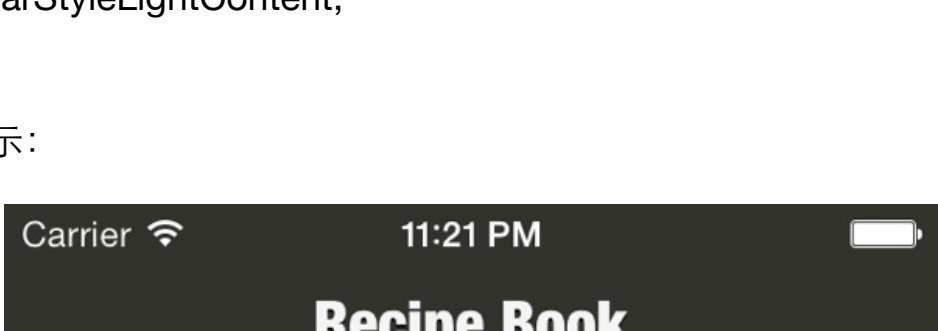
```
1. UIBarButtonItem *shareItem = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemAction target:self action:nil];
2. UIBarButtonItem *cameraItem = [[UIBarButtonItem alloc] initWithBarButtonSystemItem:UIBarButtonSystemItemCamera target:self action:nil];
3.
4. NSArray *actionButtonItems = @[shareItem, cameraItem];
5. self.navigationItem.rightBarButtonItemItems = actionButtonItems;
```

如下效果：



修改状态栏的风格

在老版本的iOS中，状态栏永远都是白色风格。而在iOS 7中，我们可以修改每个view controller中状态栏的外观。通过UIStatusBarStyle常量可以指定状态栏的内容是暗色或亮色。默认情况下，状态栏的显示是暗色。也就是说，状态栏上的时间、电池指示器和Wi-Fi信号显示为暗色。如果导航栏中使用暗色为背景，那么看起来的效果如下图所示：



如上图这种情况下，我们可能希望将导航栏的风格修改为亮色。这里有两个方法可以实现。在iOS 7中，我们可以在每个view controller中overridingpreferredStatusBarStyle:方法，如下所示：

```
1. -(UIStatusBarStyle)preferredStatusBarStyle
2. {
3.     return UIStatusBarStyleLightContent;
4. }
```

上面代码的效果如下图所示：



在iOS 7中，通过上面的方法来修改状态栏风格非常的棒。另外，我们也可以使用UIApplication的 statusBarStyle方法来设置状态栏，不过，首先需要停止使用View controller-based status bar appearance。在project target的Info tab中，插入一个新的key，名字为View controller-based status bar appearance，并将其值设置为NO。

Key	Type	Value
Bundle versions string, short	String	1.0
Bundle identifier	String	com.appcoda.\${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Main storyboard file base name	String	Main
Bundle version	String	1.0
Bundle name	String	\${PRODUCT_NAME}
Executable file	String	\$(EXECUTABLE_NAME)
Application requires iPhone environment	Boolean	YES
Supported interface orientations	Array	(3 items)
Bundle display name	String	\${PRODUCT_NAME}
Bundle creator OS Type code	String	????
Bundle OS Type code	String	APPL
View controller-based status bar appearance	Boolean	NO
Localization native development region	String	en
Required device capabilities	Array	(1 item)

然后就可以使用下面的代码来设置状态栏风格了：

```
1. [[UIApplication sharedApplication] setStatusBarStyle:UIStatusBarStyleLightContent];
```

隐藏状态栏

有时候我们需要隐藏状态栏，那么此时我们在view controller中override方法prefersStatusBarHidden:即可，如下代码所示：

```
1. -(BOOL)prefersStatusBarHidden
2. {
3.     return YES;
4. }
```