

## 使用web view替代 text view

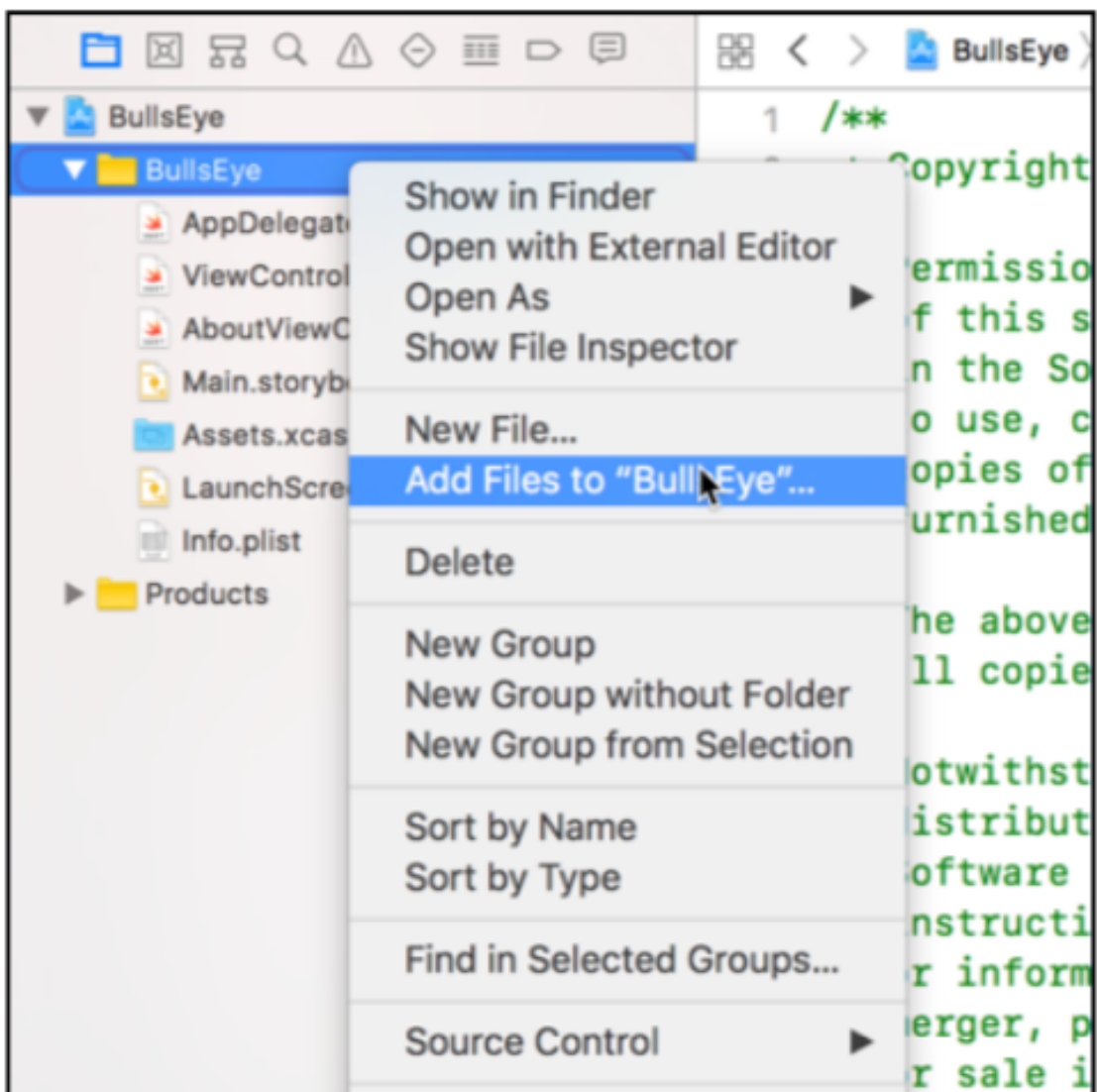
关于界面的内容还可以继续优化。

现在我们可以尝试用web view（网页视图）来替代text view（文本视图）。

在Xcode中切换到Main.storyboard，然后选中关于界面中的文本视图，用delete键删掉它。然后拖一个Web view到刚才的位置，并让它占据刚才Text view所填充的空间。

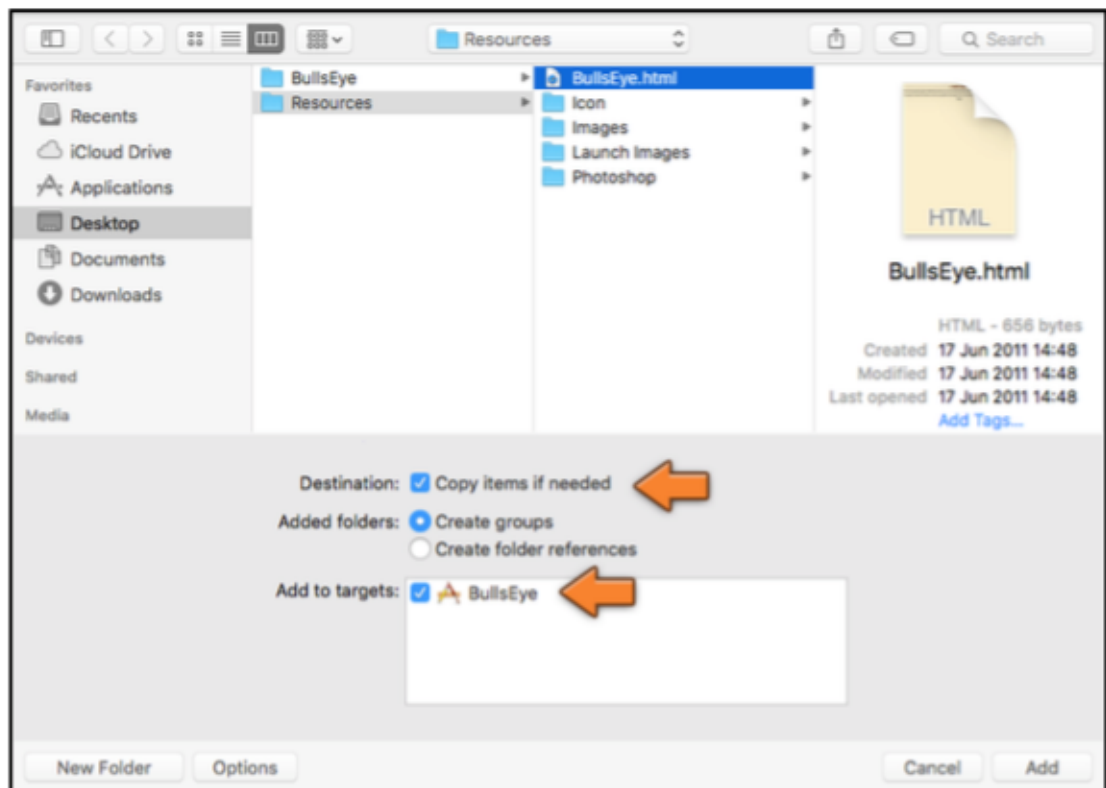
web view（网页视图）顾名思义，可以用来显示网页。我们要做的就是给它一个到某个具体网站的URL网址。当然，这里我们会先让它来显示一个准备好的HTML页面。这样就无需联机下载才能看到内容。

在Xcode左侧的Project Navigator（项目导航）点击项目名称，然后选中Add Files，把我们准备好的BullsEye.html文件添加到项目中。这是一个HTML5文档，当然你也完全可以创建自己的HTML5文档添加进去。



从文件选择器中选择BullsEye.html，这是一个包含了游戏说明的HTML5文件。

记得一定要选中Copy items if needed，以及Add to targets那边要勾选BullsEye。



点击Add就把它添加进去了。

最后在Xcode中切换到AboutViewController.swift，先添加一个outlet变量。

```
import UIKit
```

```
class AboutViewController: UIViewController {
```

```
    @IBOutlet weak var webView: UIWebView!
```

```
    override func viewDidLoad() {
```

```

        super.viewDidLoad()

        // Do any additional setup after loading the view.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }
    @IBAction func close(){
        dismiss(animated: true,completion: nil)
    }

}

```

在storyboard文件中将UIWebView视觉元素和这个新的outlet变量之间创建关联。最简单的方式当然是按住Ctrl键，从About View Controller拖一条线到Web View。

如果你反过来拖线的话会很失望的，不信试试看~

接下来回到AboutViewController.swift，添加viewDidLoad()方法：

```

    override func viewDidLoad() {
        super.viewDidLoad()

        // Do any additional setup after loading the view.

        if let url = Bundle.main.url(forResource: "BullsEye",
                                     withExtension: "html"){
            if let htmlData = try? Data(contentsOf: url){
                let baseURL = URL(fileURLWithPath: Bundle.main.bundlePath)
                webView.load(htmlData, mimeType: "text/html",
                             textEncodingName: "UTF-8",
                             baseURL: baseURL)
            }
        }
    }
}

```

这一堆代码看起来又很恐怖，不过还是那句话，别恐慌。Don't panic!  
我们先大概解释下它的作用。它的主要作用是把本地的HTML文件加载到网页视图控件中。首先我们在应用束中找到BullsEye.html文件，然后我们把它加载到一个Data对象中，最后我们让网页视图显示Data对象中的具体内容。

点击运行游戏，然后触碰(i)按钮，就会看到类似下面的界面。



在以上代码中，网页视图通过调用load:Data:MIMEType:textEncodingName:baseURL:这个方法加载具体的网页内容。其中htmlData也即刚才生成的数据对象，MIMEType就是MIME类型，是指设定某种扩展名的文件使用何种浏览器插件打开，这里不多说textEncodingName是指字符的编码方式，比如UTF-8是指Unicode编码，可以显示中文，日文，英语，以及多种语言。与之对应的还有ASCII编码，它是专门针对英语设计的。

看到这里，我们对刚才的这段代码应该有了更深的理解。

首先，我们创建了一个字符串对象，让它保存BullsEye.html在文件系统中的路径。紧接着，我们创建了一个数据对象，让它保存BullsEye.html中的具体内容。

接着，我们创建了一个URL网址对象，让它保存系统的主路径。  
最后，我们让网页视图以特定的形式来加载具体的网页内容。

好了，假如我们这里不想用固定的网页，而是想直接显示自己的个人网站内容怎么办？很简单，用下面的这段代码代替就好：

```
override func viewDidLoad() {
    super.viewDidLoad()

    // Do any additional setup after loading the view.

    // if let url = Bundle.main.url(forResource: "BullsEye",
    //                               withExtension: "html"){
    //     if let htmlData = try? Data(contentsOf: url){
    //         let baseURL = URL(fileURLWithPath: Bundle.main.bundlePath)
    //         webView.load(htmlData, mimeType: "text/html",
    //                       textEncodingName: "UTF-8",
    //                       baseURL: baseURL)
    //     }
    // }
```

```
let myURL = URL.init(string: "http://www.apple.com")
let request: URLRequest = URLRequest(url: myURL!)
webView.loadRequest(request)
}
```

我们注释了之前使用本地文件加载网页视图的代码，而使用真实的网址来替代。你可以试着使用Option键来查看帮助文档，来仔细研究下这段代码的具体含义。如果看不懂也没关系。

点击Run看看效果：



好了，该休息一下了~



