

让不懂编程的人爱上iPhone开发(2013秋iOS7版)-第12篇

上一篇的内容比较简单，只不过是修修补补而已。而这一篇我们将会把游戏从单一界面拓展到多界面。

现在开始吧。

游戏的主体已经基本建立了，作为一个艺术家，最自豪的事情就是在作品上署上自己的大名。作为产品的开发者也不例外，虽然创作的是软件产品，但仍然可以归为创作家的系列。而且，我们还需要给玩家提供一些简单的帮助，让他们知道如何进行游戏。

所以我们将要添加一个about界面，这个新的about界面中包含了传说中的text view（文本视图），里面放了一堆说明文字当然还有一个用来关闭当前界面的按钮。我们可以通过触碰游戏主界面右下角的(i)按钮进入这个界面。

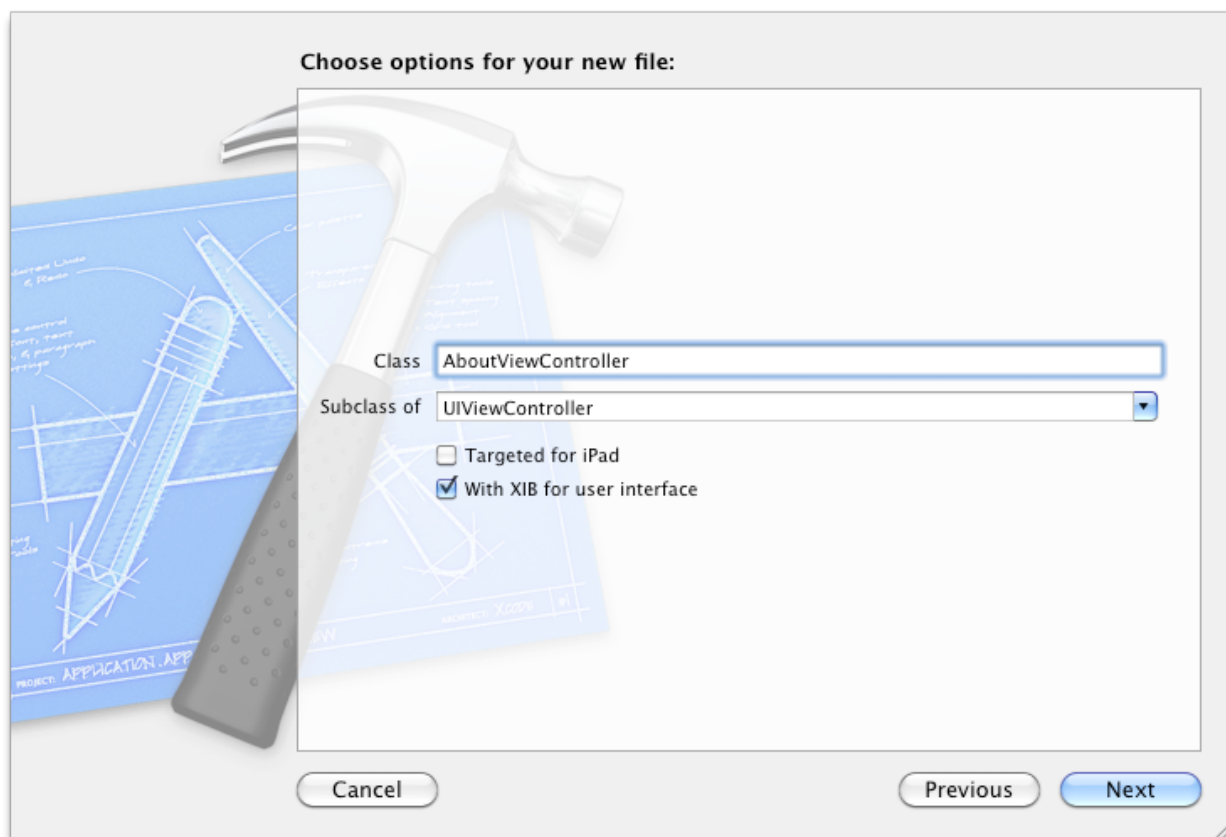
绝大多数的iOS应用或游戏中都不止有一个界面，即便是最简单的游戏也是如此，因此我们需要学习如何在应用中添加多个界面。

还记得我之前说过的吗？iOS应用中的每个界面都对应一个独立的视图控制器。当你看到“界面”的时候，就应该想到“视图控制器”。Xcode已经为我们自动创建了CrazyDragViewController这个视图控制器，但这个关于界面的视图控制器则需要我们手动来创建。

下面的东西可是全新的，不要眨眼啊。

在Xcode中，点击顶部菜单栏中的File，然后选中New File...在弹出的窗口中选中Objective-C class。

点击Next,



照着上面输入相关信息：

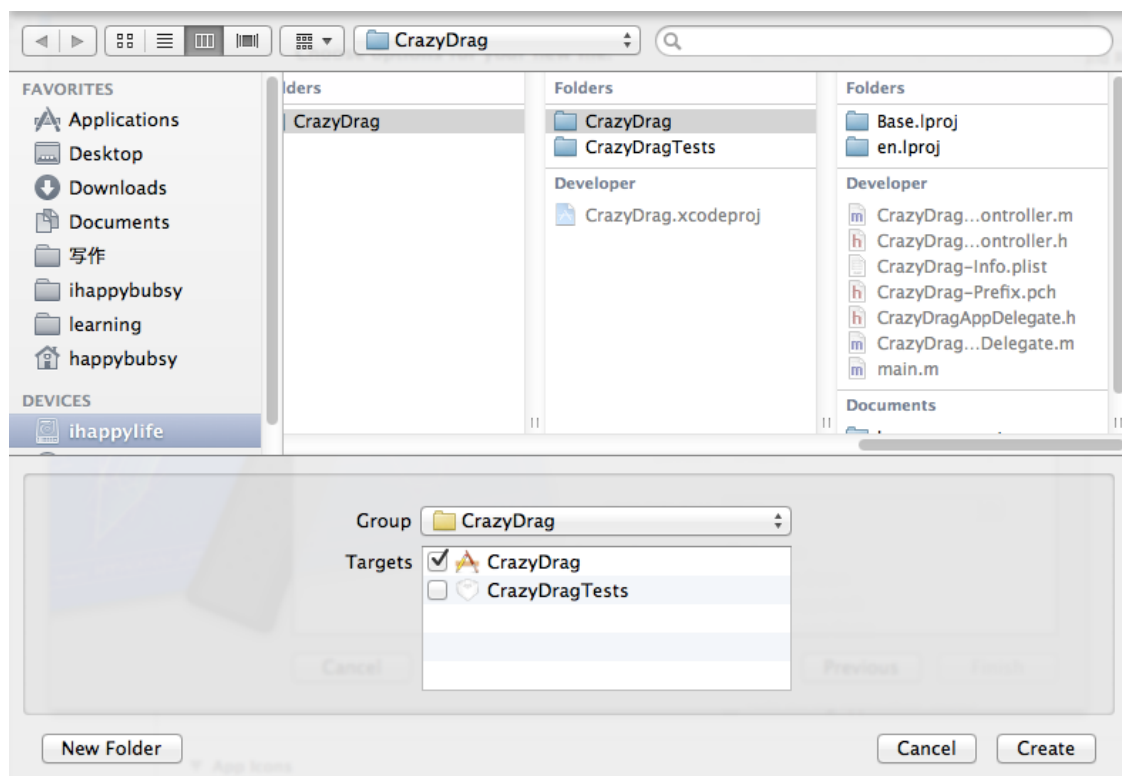
Class:AboutViewController

Subclass of: UIViewController

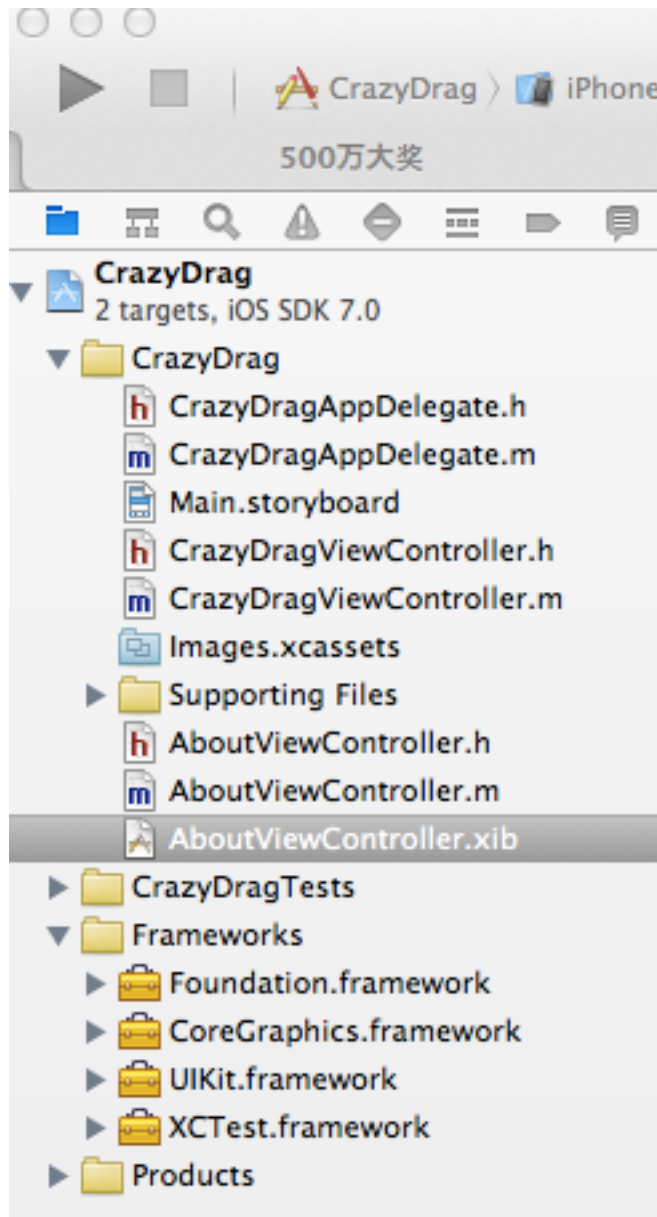
Targeted for iPad:不勾选

With XIB for user interface:这里要选中。（如果不选的话，就不会提供一个xib界面文件，也就意味着视图控制器里面的所有视觉控件都需要你手动编写。当然，如果你足够强大，或者习惯这样就随便你了。作为入门教程，特别是针对产品和设计人员看的入门教程，不选它简直是没天理啊！）

点击Next，Xcode会让你选择保存的地方，不要想太多，直接默认Create就好了。不过要记得勾选Targets后面的那个选框。



好了，Xcode帮我们创建了3个文件，也就是视图控制器的三剑客，.h,.m和.xib文件。



还记得我们之前说过的，在iOS应用中创建界面的三种方式吗？之前我们学到的是第一种，也就是使用故事板storyboard的形式，此外还有手写代码的形式，其实最常用的还是使用xib的形式。

它们的作用是神马呢？.h用来声明属性，动作方法或其它方法，.m中放置这些方法的实现代码，同时可以声明实例变量，.xib则可以用Interface Builder来进行控件的可视化摆放以及动作方法，属性和控件元素关联在一起。那么storyboard和xib的区别在哪里？

简单点说，storyboard里面通常会放置应用中多个视图控制器的界面，并使用特定的元素将不同界面关联起来，对于逻辑关系非常清晰的应用，使用storyboard可以省掉不少代码，而且看起来也一目了然。

但遗憾的是，真实世界中的应用往往非常复杂，不同界面间的相互跳转有时候复杂到令你恼火的地步。此时就要考虑使用xib或者手写代码。

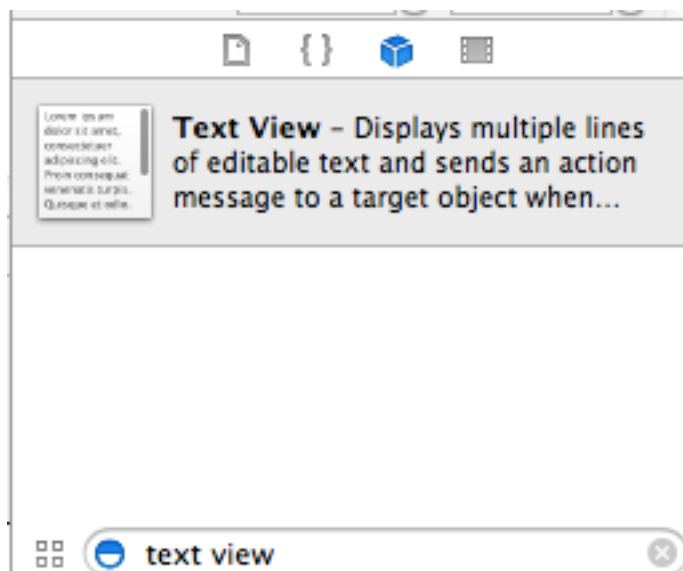
xib和.h,m文件共同构成一个视图控制器，一个xib文件通常只对应一个界面。使用xib虽然不如storyboard那么一目了然，可以一下子知道所有界面间的逻辑关系，但好处在于灵活度非常高。在Xcode5中对xib的支持更是少了一个新的台阶，WWDC 2013的相关视频中对此有介绍。

最后当然是纯手写代码了，所有的界面元素（UIKit）都可以用纯代码的形式来实现。使用这种方式的代码灵活性，可移植性以及协作性最强，当然也是代码最多且对生手挑战最大的方式。

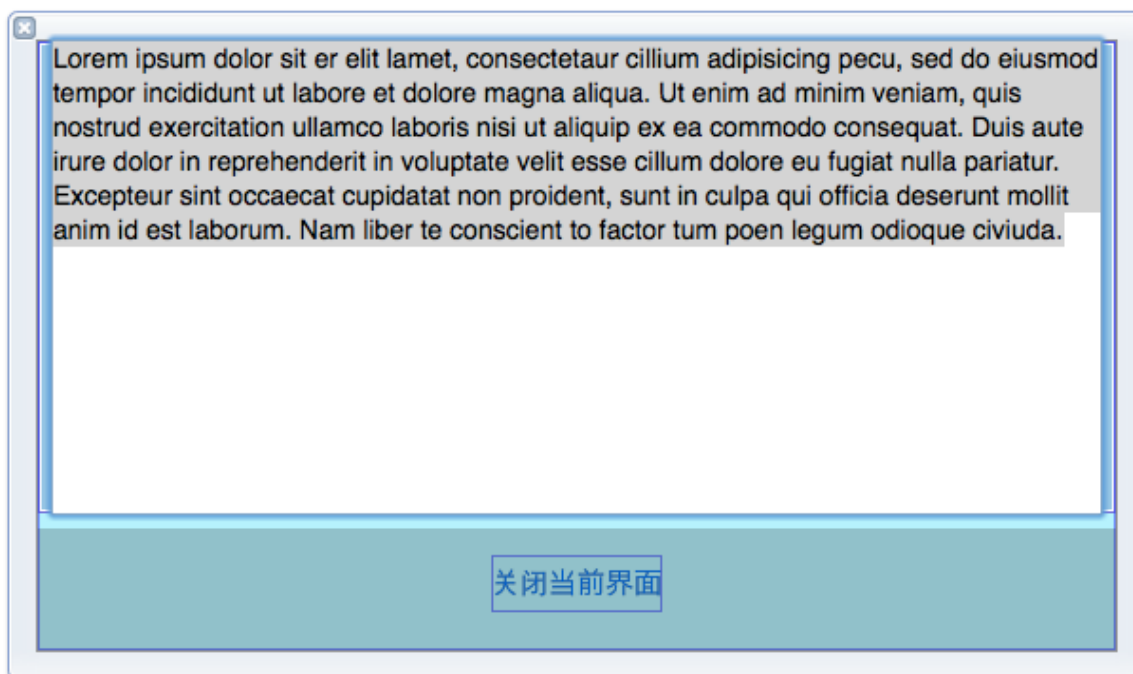
在本系列教程中，我们会优先采用storyboard和xib来编辑界面，而在Sprite Kit的教程中，则基本使用手写代码的形式。

在Xcode中点击AboutViewController.xib，然后点击白色的视图选中它，在Xcode右侧的面板中切换到Attributes Inspector。在Simulated Metrics下面将Orientation更改为Landscape。顺便把背景色改的和之前的游戏界面一致。

接下来在Xcode右下角的面板中切换到Object Library,找到 Button,把它拖动到视图上，把它改名为“关闭”，然后放在视图底部的中央位置。
然后找到Text View，用鼠标拖动到白色的视图上，然后让它覆盖按钮之上的大部分空间。



注意这里还有一个Text Field，不过它是用来输入单行文本的（比如用户名和密码）。Text View（文本视图）里面则可以放置多行文本。
最后我们得到的是类似下面的东西：



默认情况下，文本视图里面包含了一大堆拉丁语占位文字（也就是传说中的“Lorem Ipsum”，如果你恰好懂得拉丁语，或许会感到很幸福，无人爱苦，亦无人寻之欲之，乃因其苦... 参考<http://baike.baidu.com/view/2339924.htm>）。

双击文本视图，然后修改其中的文字，随便你用怎么样的文字都可以。

如果你想偷懒，可以直接用下面的这些文字。

*** 土豪真金 ***

欢迎加入土豪俱乐部！修炼土豪真金的方式很简单，你只需要拖动界面中的滑动条就好了。

你的目标是让滑动条的结点尽可能接近预设的分数，越接近表示你的土豪成分越高。欢迎获取真金！

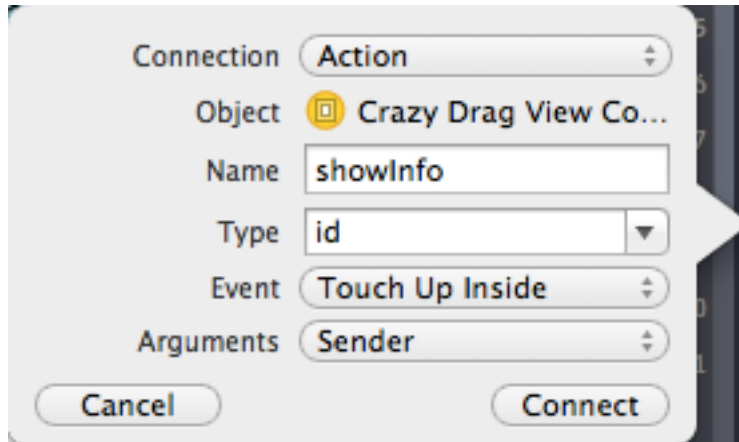
别忘了在右侧面板的Attributes Inspector选项中取消选中Editable，否则玩家就可以自己输入文本了。在我们这款游戏里面，暂时不想搞这些东西。



好了，关于界面的xib界面文件就是这样了。

接下来，我们需要在主界面（CrazyDragViewController)里面添加一些代码，从而当玩家触碰(i)按钮的时候打开关于界面。

在Xcode中切换到Main.storyboard，然后选中(i)这个按钮，创建一个关系到assistant editor的编辑器，把connection那里改为action，把Name那里填上showInfo，点击connect创建关联，如下图：



接下来肿么办？切换到CrazyDragViewController.m,在startOver方法的下面添加下面的方法：

```
- (IBAction)showInfo:(id)sender {  
  
    AboutViewController *controller = [[AboutViewController  
alloc]initWithNibName:@"AboutViewController" bundle:nil];  
    controller.modalTransitionStyle = UIModalTransitionStyleFlipHorizontal;  
    [self presentViewController:controller animated:YES completion:nil];  
  
}
```

很可怕，是吗？这么一大堆代码，貌似没有一句能看懂，肿么办？

还是<苦逼程序猿的生存指南>中最经典的那句话，Don't Panic!不要恐慌，不管看到多么高深晦涩的东西，都要相信它们的本质上是简单的，毕竟是人写出来的东西，而且这货也不是爱因斯坦图灵这种超高智商的牛人。别害怕，慢慢来，仔细看。

首先看第一行代码，AboutViewController是我们刚刚手动创建的一个视图控制器的名称，这个应该没问题吧。在这一行代码中，我们创建并初始化了一个AboutViewController对象，然后把它保存在一个本地变量controller里面：

```
AboutViewController *controller = [[AboutViewController alloc]  
initWithNibName:@"AboutViewController" bundle:nil];
```

好吧，这么多方括号一定让你感到很恐惧了。先暂且不管具体的细节。不过假如你有浓厚的兴趣，可以告诉你一个小技巧。当你看到这样的完全陌生的代码时该怎么办。比如initWithNibName是神马东东？这个时候你可以按住键盘上的option键，把鼠标放到initWithNibName的上面会出现一个问号，然后点击initWithNibName，就会弹出官方的

帮助文档，它会告诉你这个方法的作用是使用一个nib文件（或者xib文件，一回事）初始化一个视图控制器。如果你还要了解更多细节还可以在弹出的帮助文档里面点击链接查看更多内容。

可能你对创建并初始化视图控制器的代码还是有点头大，不过别害怕，再复杂的东西只要你天天接触就会习惯的。还记得卖油翁里面的那句话吗，“唯手熟尔”。很多高手之所以成为高手并非过人的天赋，而是因为在某个领域专注的去学习，研究和实践，才能成为高手和专家。

在创建并初始化了这个AboutViewController类型的视图控制器之后，接下来的这行代码设置了新视图控制器的transition style（视图转换模式）。所谓的视图转换模式是指从当前界面切换到新的界面时会采用的动画形式。这里我们想使用类似翻页的效果，所以十九选择了横向翻页的动画切换效果。如果你想了解更多的相关信息，可以按住option键，然后点击modalTransitionStyle，会看到相关的英文解释。点击Presentation Transition Styles链接，就可以在Organizer的文件中看到所有相关的视图切换模式，共有四种，分别是：

```
typedef enum {  
    UIModalTransitionStyleCoverVertical = 0,  
    UIModalTransitionStyleFlipHorizontal,  
    UIModalTransitionStyleCrossDissolve,  
    UIModalTransitionStylePartialCurl,  
} UIModalTransitionStyle;
```

好吧，树藤摸瓜我们又看到了一个新东西，枚举变量。这里不详细解释，只需要知道枚举变量的作用是列出几个特定类型的常量数值。其中 UIModalTransitionStyleCoverVertical是指从下往上切换界面， UIModalTransitionStyleFlipHorizontal是指横向翻页效果， UIModalTransitionStyleCrossDissolve是指类似交叉溶解的切换效果， UIModalTransitionStylePartialCurl是指部分卷曲的切换效果（还记得ibook应用里面那令人经验的视觉效果吗？~）文字说明不如实际的视觉展示，你可以自行换用其中的一种来查看具体的视觉效果。

还记得something.somethingElse的点表示法吗？其实这就意味着modalTransitionStyle是AboutViewController的一个属性，所有的视图控制器都有这个属性。

好了，现在我们已经创建了所需的视图控制器对象，同时也设置了它的属性（至少是一个~），接下来就可以在界面上显示它了：

```
[self presentViewController:controller animated:YES completion:nil];
```

在上面的代码中，presentViewController方法的作用是在当前视图控制器之上打开一个新的界面。它告诉CrazyDragViewController这个当前界面所使用的视图控制器以动画的形式

打开关于界面。如果把animated:设置为No,那么就不会有翻页效果,新的界面很突兀的出现。为了提升用户体验,我们最好还是使用某种看起来很精巧的动画来实现界面切换。

同样的,虽然哥在这里解释了presentViewController方法的作用,你可能还是想了解更多相关信息。按住option键,点击presentViewController,在弹出的文本中会看到更多信息。它会告诉我们presentViewController方法的具体作用,包括为指定的视图控制器设置presentedViewController属性,重新调整视图控制器的视图大小,然后把视图添加到视图体系中,等等。如果你还想了解更多信息,可以继续查看其中的链接。

授人以鱼不如授人以渔,如果我只是简单告诉你每一行代码的作用,你可能当时知道了。但后面遇到新的完全陌生的代码又该如何呢?特别是开源项目或者是借鉴其它人的代码时,经常会有看不懂的地方。虽然部分可以借助注释说明或者良好的编码习惯和命名习惯来看懂,但也有很多地方是需要通过查询方法或函数的定义及相关信息来了解的。无论你使用哪种编程语言,都会大量用到第三方类库,包括系统类库,框架和函数库。所以,要擅于通过帮助文档来迅速学习和掌握新的东西。下次碰到这种看起来陌生而又可怕的代码时,还是那句话,Don't Panic~!不要恐慌!

关于视图,窗口,视图控制器,根视图控制器,storyboard,在后面的教程中我们会进行一个详细的对比说明。

就目前这个阶段来说,了解到这一步已经足够深入了。再深入恐怕你就看不下去了:)

回到Xcode中的CrazyDragViewController.m,你会看到红色的错误提示,Use of undeclared identifier 'AboutViewController'。好吧,其实这是很多刚入门程序猿容易犯的错误。当你看到这种错误提示时,其实它是在告诉你,编译器不明白你所使用的对象究竟是什么。在代码世界中,每个对象,类都是一个独立的生命体,就如同你我一样。如果我不告诉你我姓甚名谁,干嘛的,住哪儿,你对我一无所知。比如我和苍老师素昧平生,彼此没有发生过任何关联。但我在日记中写到,今天我让苍老师帮我在淘宝上买了个ipad mini寄过来,这简直就是扯蛋YY。如果我想让苍老师帮我在淘宝上买个ipad mini,最起码我得让苍老师知道有我这样一个人。同样的,你在CrazyDragController.m里面直接说我要创建一个AboutViewController视图控制器,但是当前的视图控制器根本不知道它是个神马东西,剩下的就是扯蛋了。

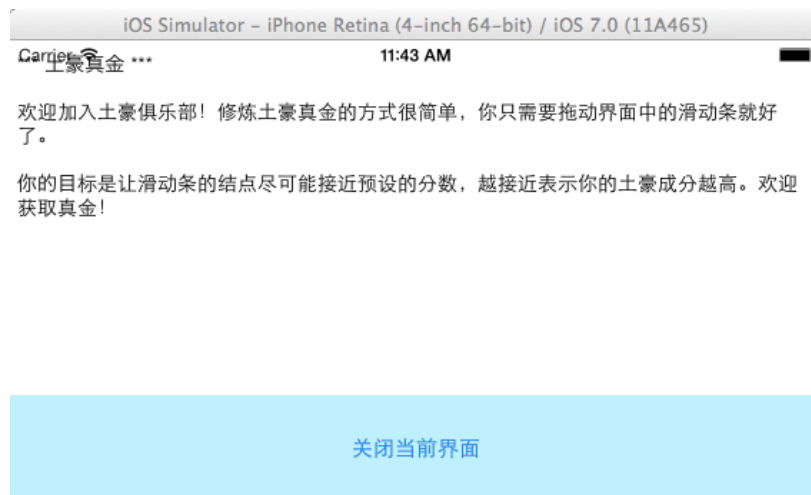
好在这一类型的错误基本不可能通过编译,也不会给产品造成隐患。

在CrazyDragViewController.m的顶部添加一行代码:

```
#import "AboutViewController.h"
```

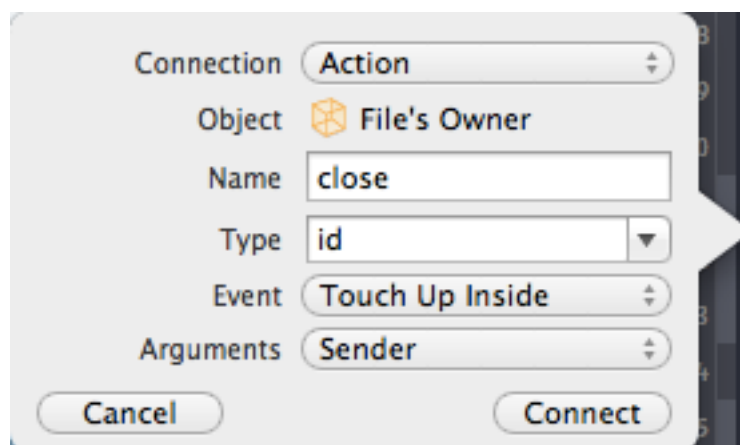
通过#import这行代码,其实我们就把AboutViewController.h的相关代码导入到了CrazyDragViewController的源文件中,这样它就知道AboutViewController这个视图控制器究竟是谁了。

好吧，看起来差不多了。让我们点击Run运行游戏，然后触碰(i)按钮来切换到新的界面。



不过似乎还有点小问题，如果我们触碰关于界面中的关闭按钮，什么也不会发生。一旦玩家进入关于界面，他就被锁定在这个空间了。想想卡布里卡里面的虚拟人，被长期关闭在一个单独的空间里面暗无天日，是多么的残忍。为了让虚拟世界和现实世界中的人都能开心起来，我们必须让关闭按钮和某个动作方法关联在一起。

好吧，和之前类似的方式。在Xcode中点击AboutViewController.xib，打开Assistant Editor，选中关闭当前界面这个按钮，按住control键，用鼠标拖一条线到代码区的@interface 和@end之间，在Connection处选择Action,在Name处填上close，如下图：



点击Connect创建关联，Ok了。

最后切换到AboutViewController.m，添加动作方法的实现代码：

```
- (IBAction)close:(id)sender {  
  
    [self.presentingViewController dismissViewControllerAnimated:YES completion:nil];  
}
```

这里的代码是开启和关闭新界面的通常模式。首先我们在CrazyDragController.m里面创建了新的AboutViewController视图控制器对象的实例，然后通过presentViewController把它添加到视图列表中，让它出现在CrazyDragController这个视图控制器的上面。当玩家对这个新界面厌倦了的时候，就可以通过发送dismissViewControllerAnimated消息给呈现它的视图控制器，也就是CrazyDragController，通过它关闭AboutViewController这个视图控制器。

我想你有点糊涂了。这么说吧，你可以把CrazyDragViewController和AboutViewController这两个视图控制器看做两张纸，最开始我们把CrazyDragViewController这张纸放在面上，后来我们把AboutViewController这张纸放在CrazyDragViewController这张纸的上面。当我们看烦了以后，就把AboutViewController这张纸拿走扔掉，继续显示CrazyDragViewController这张纸上的画面。不过和纸张的关系稍有不同的是，我们在AppDelegate中把CrazyDragViewController设置为根视图控制器，而AboutViewController只不过是通過CrazyDragViewController创建的视图控制器。你应该认出来了，self.presentingViewController意味着presentingViewController是当前视图控制器(AboutViewController)的属性，它指的是CrazyDragViewController。我们无需手动设置这个属性，UIKit会自动帮我们处理这些东西。事实上，在iOS开发的过程中，UIKit会进行很多暗箱操作，我们唯一需要了解的只是如何使用。

看不懂的东西怎么办？按住option键，点击presentingViewController，会看到详细的帮助信息。

这里需要注意的是，因为Xcode有自动提示功能，你很可能一不小心就选择了presentedViewController，它同样有这个方法：

```
[self.presentedViewController dismissViewControllerAnimated:YES completion:nil];
```

区别是什么？我们可以说CrazyDragViewController是AboutViewController的presentingViewController（CrazyDragViewController是呈现AboutViewController的视图控制器），也可以说AboutViewController是CrazyDragViewController的presentedViewController(AboutViewController是CrazyDragViewController所呈现的视图控制器)。但如果反过来就完全不成立了！

这一点作为非英语母语国家的天朝程序猿一样要当心，presenting和presented的区别，类

似的东西在iOS开发中还有很多。比如didxxx和shouldxxx的区别（一个是过去时，一个是将来时）我第一次碰到此类代码的时候，也在这里吃过亏，就是英语没学好的原因啊！

再次点击Run运行游戏，我们终于可以回到主界面了。

好吧，我们的游戏似乎已经完工了。所有的功能都具备了，而且也没有什么bug。现在的问题是-界面太难看！如果你现在把这款应用提交给App Store，百分百会被拒掉！

苹果官方提供了一个iOS 产品设计的人机交互指南，无论是程序猿，产品经理，还是设计人员，都应该反复仔细阅读。

https://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/MobileHIG/Introduction/Introduction.html#//apple_ref/doc/uid/TP40006556

即便你以后不打算做iOS开发，想做android,网页或者桌面开发，了解iOS的人机交互指南也是非常重要的。毕竟，苹果可是设计领域当之无愧的神级公司！

好了，今天的学习先到此结束吧，这一篇的内容可能对新手有点吃力，不过我们也学到了更多有用的东西。

老习惯，福利一张，for宅男，for mm。不过美女看多了也没意思，我们换点新东西吧。

