让不懂编程的人爱上iPhone开发(2013球iOS7版)-第11篇

欢迎回到我们的iPhone开发教程系列，中秋佳节倍思亲，天凉好个秋。不知道最新的土豪金iPhone 5S你是否买到了？

不满意的话可以向Cook厨子吐槽一番。或者你想要对苹果的未来发展战略指点江山挥斥方遒的话，可以联系cook本人，听说他是个好基友，相信会虚心听取你的建议的。如果你是萌妹子就算了。

不过别忘了这个季节也是学习新知识的最佳时机。让我们继续前进吧。

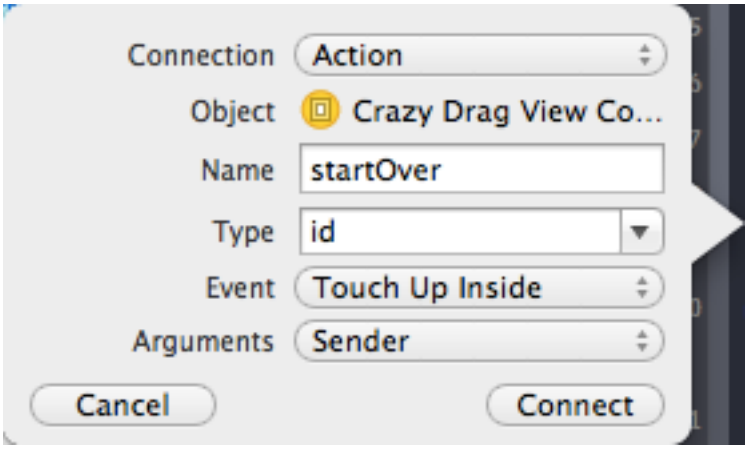
重新来过

别害怕，哥不是让你抛弃之前所有的源代码，从零开始重新构建这个项目！这里说的是游戏界面里面的“重新来过”按钮。在我们的to-do清单里面曾经提到过，这个按钮负责重置玩家的得分，同时让游戏回合数重置为首回合。如果你要和其他人一起来玩这款游戏，这个按钮就会非常有用。比如你先来玩10个回合，记录下总的得分，然后用“重新来过”按钮重置游戏得分。接着你的朋友再来玩10个回合，并记录下总的得分。最后比较下你们两的得分就知道谁更厉害。

小练习：
现在可以自己尝试着实现一下“重新来过”这个按钮的功能。在之前的学习中，我们已经知道如何让视图控制器对按钮的触碰做出响应，当然现在你肯定也直到如何来更改score和round变量的数值了。

要事第一！所以首先让我们把“重新来过”按钮和某个动作关联在一起。
还记得该怎么做吗？

首先，在Xcode中点击Main.storyboard文件，选中重新来过那个按钮，按住control键，用鼠标拖一条线到assistant editor中代码区的@interface和@end之间，最好在之前的sliderMoved方法定义代码前后。按照下图的内容，把Connection选择为Action，把Name那里填上startOver



点击Connect创建关联。

-(IBAction)startOver;

此时CrazyDragViewController.m中@interface和@end之间的代码如下：

```
@interface CrazyDragViewController (){

    int currentValue;
    int targetValue;
    int score;
    int round;

}
- (IBAction)startOver:(id)sender;
- (IBAction)sliderMoved:(UISlider*)sender;
@property (strong, nonatomic) IBOutlet UISlider *slider;

@property (strong, nonatomic) IBOutlet UILabel *targetLabel;
@property (strong, nonatomic) IBOutlet UILabel *scoreLabel;
@property (strong, nonatomic) IBOutlet UILabel *roundLabel;
```

```
- (IBAction)showAlert:(id)sender;
@end
```

最后当然是点击.m文件（或者切换到专门编辑代码的Tab），在showAlert方法的下面添加这个动作的具体实现代码：

```
- (IBAction)startOver:(id)sender {

    [self startNewGame];
    [self updateLabels];
}
```

这里的代码很简单。当玩家触碰“重新来过”的按钮后，我们会调用startNewGame方法重新启动游戏。同时，我们需要调用updateLabels方法来更新相关的score,round和target标签的数值。

你看，只要你在写代码的时候给方法或者变量的定义比较符合语言习惯，即便是一个完全不懂编码的人也大致能看懂。不过有些程序猿喜欢偷懒，假如这里是[self sNG]，你还能明白是什么意思吗？我反正是不能。但很有些人喜欢这么干，还叫嚣看不懂是你自己太嫩太菜鸟，图洋图森破，其实就是装B，显得自己逼格高。

当然，这时我们会看到红色的错误提示，因为startNewGame这个方法之前没有定义过。所以我们需要在startNewRound方法的下面添加startNewGame方法的实现代码：

```
-(void)startNewGame{

    score = 0;
    round = 0;
    [self startNewRound];
}
```

在上面的代码中，每当重新启动新的游戏时，我们会把玩家的总得分重置为0，然后把游戏总回合数重置为0。因为score和round都是实例变量，我们也不需要在这里重新定义它们的数据类型。然后就是调用startNewRound方法开始新的游戏回合。

需要注意的是，这里的游戏回合数round重置为0,为什么不是1呢？这是因为startNewRound方法中会让round自动加1，如果这里的round重置为1，那么新游戏的第1回合数就是2了。可能我这里有点啰嗦了，实际上你可以自己修改下代码就知道了。（这就是为什么我们不用日常生活的语言来写代码，虽然符合人类的表达习惯，但不如编程语言简洁，逻辑性也不够严密）。

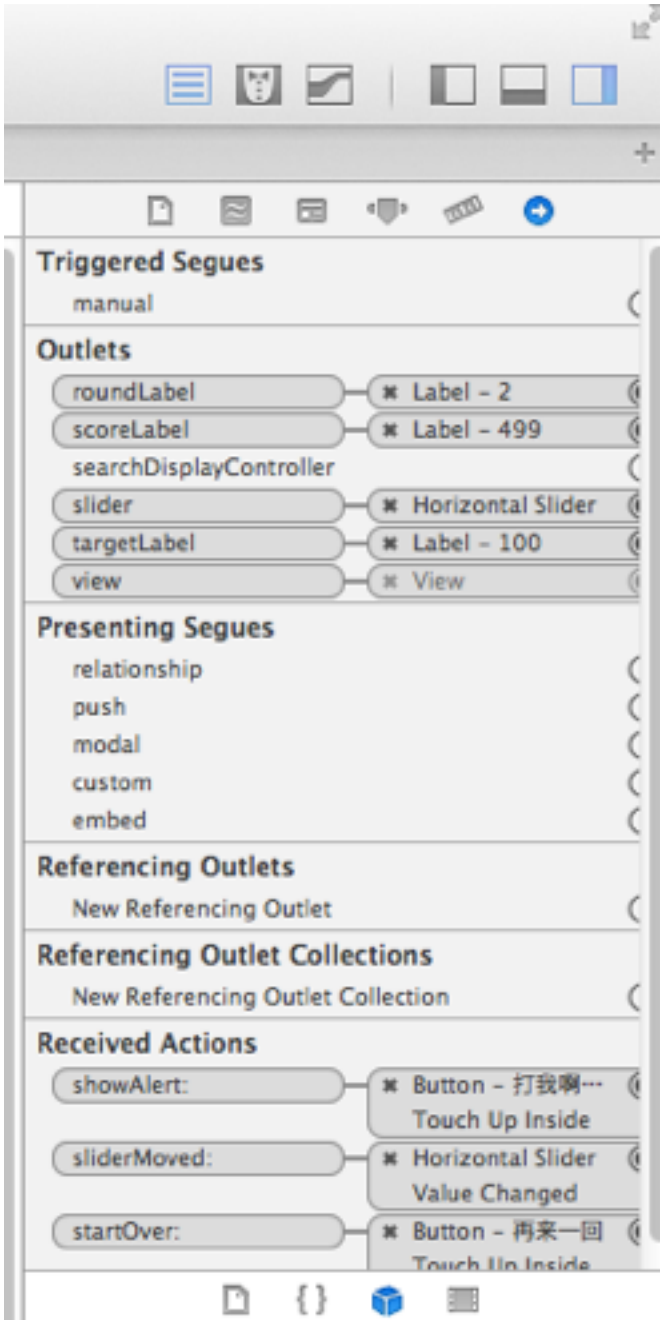
最后别忘了，我们需要在viewDidLoad方法用将原来的startNewRound方法用startNewGame方法来替代。虽然程序的实际运行效果不会发生变化，但是这样会让源代码的逻辑更加清晰，也便于今后的扩展和修改。

好了，现在可以点击Run运行下游戏看看效果。
当你触碰“重新来过”按钮时，游戏就会重新开始。

科普：方法的实现代码究竟该放在哪里？

在学习iOS开发的过程中，你可能早就有了这样的问题，某个方法的实现代码究竟应该放在文件的哪个位置呢？在Xcode4.3之前，方法的放置位置非常重要。如果你调用了某个方法，但却没有在该方法的前面定义它，编译器就会给你错误提示。和人类的习惯相似，编译器通常也是自顶向下来查看代码的。
不过在目前的Xcode版本(Xcode4.3及以后的版本）中，某个方法实现代码的具体位置并没有那么严格的规定，但为了照顾人类的阅读习惯，基本是在@implementation之后，@end之前。如果有属性合成代码，则通常放在@synthesize相关代码的后面。不过按照惯例，如果某个方法需要调用另外一个方法，则该方法的代码通常放在所调用方法的下面。比如这里的startNewGame方法的位置最好放在startNewRound方法的下面，而startOver方法的位置则最好放在startNewGame方法的下面。

提示：
如果你忘了storyboard界面文件中的控件具体和哪个动作方法关联在一起，可以选中Crazy Drag View Controller，然后在Xcode右侧面板的Connections Inspector中查看。其中会显示所有和File’s Owner（也就是CrazyDragViewController这个视图控制器）相关的关联。



好了，今天的学习暂时到此结束，下一篇我们将学习另一个非常重要的概念。同时将现在的单一界面应用扩展为多界面应用。让我们共同期待吧。

最后送福利一张。


萌宠团