**iOS开发UI篇—Button基础**

一、简单说明

一般情况下，点击某个控件后，会做出相应反应的都是按钮

按钮的功能比较多，既能显示文字，又能显示图片，还能随时调整内部图片和文字的位置

二、按钮的三种状态

normal（普通状态）

默认情况（Default）

对应的枚举常量：UIControlStateNormal

highlighted（高亮状态）

按钮被按下去的时候（手指还未松开）

对应的枚举常量：UIControlStateHighlighted

disabled（失效状态，不可用状态）

如果enabled属性为NO，就是处于disable状态，代表按钮不可以被点击

对应的枚举常量：UIControlStateDisabled

三、注意点

（1）从Xcode5开始，图片资源都放到Images.xcassets中进行管理，可以使用拖拽的方式添加项目中用到的图片到Images.xcassets中

（2）若干多个控件共用一段代码，通常使用tag。

四、代码示例

（1）

[](javascript:void(0);)

1 #import "LFViewController.h"

2

3 @interface LFViewController ()

4

5 @property (weak, nonatomic) IBOutlet UIButton \*headImageView;

6

7 @end

8

9 @implementation LFViewController

10

11 // 在OC中，绝大多数的控件的监听方法的第一个参数就是控件本身

12 //- (IBAction)left:(UIButton \*)button {

13 //

14 // NSLog(@"----");

15 //}

16 - (IBAction)move

17 {

18 // 通过frame修改head的位置

19 // 在OC中，不允许直接修改“对象”的“结构体属性”的“成员”

20 // 允许修改“对象”的“结构体属性”

21 // 1. 取出结构体属性

22 CGRect rect = self.headImageView.frame;

23 // 2. 修改结构体成员

24 rect.origin.y -= 20;

25 // 3. 设置对象的结构体属性

26 self.headImageView.frame = rect;

27 }

[](javascript:void(0);)

（2）

[](javascript:void(0);)

1 #import "LFViewController.h"

2

3 /\*\*

4 使用git

5

6 1. 创建项目时，勾选git

7 2. 开发告一段落后，选择"Source Control""Commit",并编写注释

8 \*/

9

10

11 // 枚举类型实质上就是一个整数，作用就是用来替代魔法数字

12 // 枚举类型中，指定了第一个整数之后，后面的数字会递增

13 typedef enum

14 {

15 kMovingDirTop = 10,

16 kMovingDirBottom,

17 kMovingDirLeft,

18 kMovingDirRight,

19 } kMovingDir;

20

21 #define kMovingDelta 50

22

23 @interface LFViewController ()

24

25 @property (weak, nonatomic) IBOutlet UIButton \*headImageView;

26

27 @end

28

29 @implementation LFViewController

30

31 - (IBAction)move:(UIButton \*)button

32 {

33 // CGRect rect = self.headImageView.frame;

34 CGPoint p = self.headImageView.center;

35

36 // magic number魔法数字，其他程序员看到代码的时候，不知道是什么意思

37 switch (button.tag) {

38 case kMovingDirTop:

39 p.y -= kMovingDelta;

40 break;

41 case kMovingDirBottom:

42 p.y += kMovingDelta;

43 break;

44 case kMovingDirLeft:

45 p.x -= kMovingDelta;

46 break;

47 case kMovingDirRight:

48 p.x += kMovingDelta;

49 break;

50 }

51

52 [UIView beginAnimations:nil context:nil];

53 [UIView setAnimationDuration:1.0];

54

55 self.headImageView.center = p;

56

57 [UIView commitAnimations];

58 }

59

60 - (IBAction)zoom:(UIButton \*)button

61 {

62 CGRect rect = self.headImageView.bounds;

63

64 // 在C语言中，关于bool的判断：非零即真

65 if (button.tag) {

66 rect.size.width += 50;

67 rect.size.height += 50;

68 } else {

69 rect.size.width -= 50;

70 rect.size.height -= 50;

71 }

72

73 // 首尾动画

74 // beginAnimations表示此后的代码要“参与到”动画中

75 [UIView beginAnimations:nil context:nil];

76 [UIView setAnimationDuration:2.0];

77

78 self.headImageView.bounds = rect;

79 // self.headImageView.alpha = 0;

80

81 // commitAnimations,将beginAnimation之后的所有动画提交并生成动画

82 [UIView commitAnimations];

83 }

84

85 @end

[](javascript:void(0);)

五、补充笔记

1. IBAction的参数

- (IBAction)left:(UIButton \*)button

（1） 在OC中，绝大多数的控件监听方法的第一个参数就是控件本身

（2） 默认连线时的参数类型是id

（3） 如果要在监听方法中，方便控件的使用，可以在连线时或者连线后，修改监听方法的参数类型

2. 修改对象的结构体成员

在OC中，不允许直接修改“对象”的“结构体属性”的“成员”，但是允许修改“对象”的“结构体属性”

修改结构体属性的成员方法如下：

（1）使用临时变量记录对象的结构体属性

（2） 修改临时变量的属性

（3）将临时变量重新设置给对象的结构体属性

3. 在程序开发中需要避免出现魔法数字(Magic Number)

使用枚举类型，可以避免在程序中出现魔法数字

（1）枚举类型实质上就是一个整数，其作用就是用来替代魔法数字

（2）枚举类型中，指定了第一个整数之后，后面的数字会递增

4. frame & bounds & center

1> frame可以修改对象的位置和尺寸

2> bounds可以修改对象的尺寸

3> center可以修改对象的位置

5. 首尾式动画

// beginAnimations表示此后的代码要“参与到”动画中

[UIView beginAnimations:nil context:nil];

// setAnimationDuration用来指定动画持续时间

[UIView setAnimationDuration:2.0];

self.headImageView.bounds = rect;

......

// commitAnimations，将beginAnimation之后的所有动画提交并生成动画

[UIView commitAnimations];