**iOS开发UI篇—UIScrollView控件实现图片轮播**

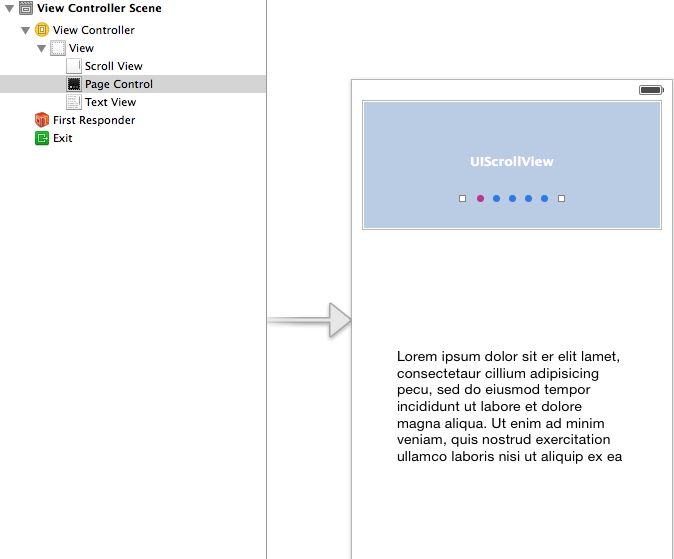
一、实现效果

实现图片的自动轮播

二、实现代码

storyboard中布局



代码：

[](javascript:void(0);)

1 #import "YYViewController.h"

2

3 @interface YYViewController () <UIScrollViewDelegate>

4 @property (weak, nonatomic) IBOutlet UIScrollView \*scrollview;

5 /\*\*

6 \* 页码

7 \*/

8 @property (weak, nonatomic) IBOutlet UIPageControl \*pageControl;

9

10 @property (nonatomic, strong) NSTimer \*timer;

11 @end

12

13 @implementation YYViewController

14

15 - (void)viewDidLoad

16 {

17 [super viewDidLoad];

18

19 // 图片的宽

20 CGFloat imageW = self.scrollview.frame.size.width;

21 // CGFloat imageW = 300;

22 // 图片高

23 CGFloat imageH = self.scrollview.frame.size.height;

24 // 图片的Y

25 CGFloat imageY = 0;

26 // 图片中数

27 NSInteger totalCount = 5;

28 // 1.添加5张图片

29 for (int i = 0; i < totalCount; i++) {

30 UIImageView \*imageView = [[UIImageView alloc] init];

31 // 图片X

32 CGFloat imageX = i \* imageW;

33 // 设置frame

34 imageView.frame = CGRectMake(imageX, imageY, imageW, imageH);

35 // 设置图片

36 NSString \*name = [NSString stringWithFormat:@"img\_0%d", i + 1];

37 imageView.image = [UIImage imageNamed:name];

38 // 隐藏指示条

39 self.scrollview.showsHorizontalScrollIndicator = NO;

40

41 [self.scrollview addSubview:imageView];

42 }

43

44 // 2.设置scrollview的滚动范围

45 CGFloat contentW = totalCount \*imageW;

46 //不允许在垂直方向上进行滚动

47 self.scrollview.contentSize = CGSizeMake(contentW, 0);

48

49 // 3.设置分页

50 self.scrollview.pagingEnabled = YES;

51

52 // 4.监听scrollview的滚动

53 self.scrollview.delegate = self;

54

55 [self addTimer];

56 }

57

58 - (void)nextImage

59 {

60 int page = (int)self.pageControl.currentPage;

61 if (page == 4) {

62 page = 0;

63 }else

64 {

65 page++;

66 }

67

68 // 滚动scrollview

69 CGFloat x = page \* self.scrollview.frame.size.width;

70 self.scrollview.contentOffset = CGPointMake(x, 0);

71 }

72

73 // scrollview滚动的时候调用

74 - (void)scrollViewDidScroll:(UIScrollView \*)scrollView

75 {

76 NSLog(@"滚动中");

77 // 计算页码

78 // 页码 = (contentoffset.x + scrollView一半宽度)/scrollView宽度

79 CGFloat scrollviewW = scrollView.frame.size.width;

80 CGFloat x = scrollView.contentOffset.x;

81 int page = (x + scrollviewW / 2) / scrollviewW;

82 self.pageControl.currentPage = page;

83 }

84

85 // 开始拖拽的时候调用

86 - (void)scrollViewWillBeginDragging:(UIScrollView \*)scrollView

87 {

88 // 关闭定时器(注意点; 定时器一旦被关闭,无法再开启)

89 // [self.timer invalidate];

90 [self removeTimer];

91 }

92

93 - (void)scrollViewDidEndDragging:(UIScrollView \*)scrollView willDecelerate:(BOOL)decelerate

94 {

95 // 开启定时器

96 [self addTimer];

97 }

98

99 /\*\*

100 \* 开启定时器

101 \*/

102 - (void)addTimer{

103

104 self.timer = [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(nextImage) userInfo:nil repeats:YES];

105 106 }

107 /\*\*

108 \* 关闭定时器

109 \*/

110 - (void)removeTimer

111 {

112 [self.timer invalidate];

113 }

114 @end

[](javascript:void(0);)

提示：以下两个属性已经和storyboard中的控件进行了连线。

@property (weak, nonatomic) IBOutletUIScrollView \*scrollview;

@property (weak, nonatomic) IBOutletUIPageControl \*pageControl;

补充：定时器NSTimer

   定时器 适合用来隔一段时间做一些间隔比较长的操作

 NSTimeInterval:多长多件操作一次

 target :操作谁

 selector : 要操作的方法

 userInfo: 传递参数

 repeats: 是否重复

  self.timer = [NSTimer scheduledTimerWithTimeInterval:1 target:self selector:@selector(nextImage) userInfo:nil repeats:YES];