wayne23

管理

随笔分类

Flash(36)
FluorineFX(7)
iOS(41)
Objective-C(5)
WEB(28)
其它(19)

随笔档案

2016年4月 (2)
2015年11月 (1)
2015年9月 (1)
2015年5月 (45)
2015年3月 (1)
2015年2月 (2)
2015年1月 (2)
2014年10月 (5)
2014年8月 (4)
2014年7月 (3)
2014年6月 (4)
2014年5月 (1)
2014年4月 (3)
2014年3月 (16)
2013年11月 (2)
2013年3月 (1)
2013年1月 (1)
2012年8月 (2)
2011年9月 (1)
2010年12月 (2)
2010年11月 (1)
2010年9月 (1)
2010年7月 (35)

相册

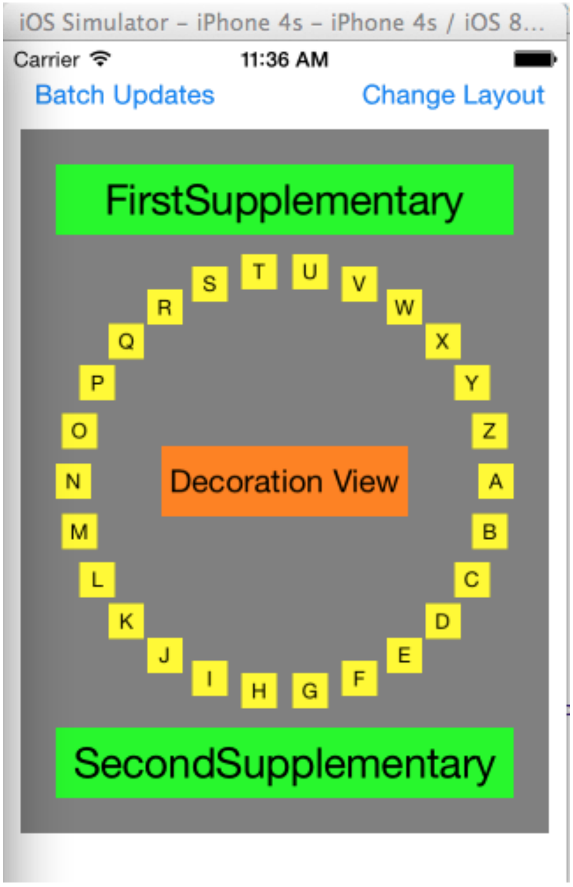
blogImg(100)

最新评论

1. Re:一个UICollectionView自定义layout的实现

O(∩_∩)O谢谢

一个UICollectionView自定义layout的实现


iOS Simulator – iPhone 4s – iPhone 4s / iOS 8...
Carrier
11:36 AM
Batch Updates
Change Layout
FirstSupplementary
S T U V W X Y Z A B C D E F G H I J K L M N O P Q R
Decoration View
SecondSupplementary

```
#import <UIKit/UIKit.h>

@interface AppDelegate : UIResponder <UIApplicationDelegate>

@property (strong, nonatomic) UIWindow *window;

@property (strong, nonatomic) NSMutableArray *letterArray;

@end
```

```
#import "AppDelegate.h"

@interface AppDelegate ()
```

--星红

2. Re:三种ViewController跳转的异同

不加注释。。。。表示有点混乱

--Mr_Deng

3. Re:一个UICollectionView自定义layout的实现

太感谢了，帮大忙了

--小雨点软件

4. Re:CocoaPods安装和使用

学习了， thx

--柴禾

5. Re:AS3 CookBook学习整理(四)

使用ascb的Pen.drawStar(x,y,points,innerRadius,outerRadius,ratation)，这句话怎么理解。看了，还是没能自己画出一个星形，求解，急需。谢谢!

--石准

```
@end

@implementation AppDelegate

- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions
{
    self.letterArray = [NSMutableArray array];
    for(int i=0; i<26; i++)
    {
        [self.letterArray addObject:[NSString
stringWithFormat:@"%C",(unichar)(65+i)]];
    }

    return YES;
}
```

```
#import <UIKit/UIKit.h>

@interface ViewController : UIViewController

@end
```

```
#import "ViewController.h"
#import "AppDelegate.h"
#import "CollectionViewDataSource.h"
#import "DraggableCircleLayout.h"
#import "LSCollectionViewHelper.h"

@interface ViewController ()
{
    UICollectionView *collectionView;
    CollectionViewDataSource *cvDataSource;
}
@end

@implementation ViewController

- (IBAction)ChangeLayoutClickHandler:(id)sender
{
    if([collectionView.collectionViewLayout isKindOfClass:
[CircleLayout class]])
    {
        UICollectionViewFlowLayout *layout =
[[UICollectionViewFlowLayout alloc] init];
        collectionView.collectionViewLayout = layout;
    }
    else
    {
        CircleLayout *layout = [[CircleLayout alloc] init];
        collectionView.collectionViewLayout = layout;
    }
}

- (IBAction)BatchUploadClickHandler:(id)sender
{
    //这里有个细节需要注意，最好是将删除操作放在添加操作前面，因为无论你顺序如何，始终都会先执行删除操作。
    //如果代码顺序是先添加后删除，但实际执行顺序是先删除后添加，可能会因为索引不对影响代码逻辑。
    [collectionView performBatchUpdates:^{
        NSMutableArray *letterArray = [self getLetterArray];
        //删除四个元素
        NSIndexPath *path1 = [NSIndexPath indexPathForItem:0 inSection:0];
        NSIndexPath *path2 = [NSIndexPath indexPathForItem:1 inSection:0];
        NSIndexPath *path3 = [NSIndexPath indexPathForItem:2 inSection:0];
        NSIndexPath *path4 = [NSIndexPath indexPathForItem:3 inSection:0];

        NSIndexSet *indexSet = [NSIndexSet indexSetWithIndexesInRange:NSMakeRange(0,4)];

        [indexSet enumerateIndexesUsingBlock:^(NSUInteger idx, BOOL *stop){
            NSLog(@"%lu", (unsigned long)idx);
        }];

        [letterArray removeObjectsAtIndexes:indexSet];

        NSArray *array = [NSArray arrayWithObjects:path1, path2, path3, path4, nil];
        [collectionView deleteItemsAtIndexPaths:array];

        //添加一个元素
        [letterArray addObject:@"1"];

        [collectionView insertItemsAtIndexPaths:[NSArray arrayWithObject:[NSIndexPath indexPathForItem:letterArray.count-1 inSection:0]]];
    } completion:nil];
}

- (void)viewDidLoad
{
    [super viewDidLoad];

//    UICollectionViewFlowLayout *layout =
[[UICollectionViewFlowLayout alloc] init];
//    CircleLayout *layout = [[CircleLayout alloc] init];
    DraggableCircleLayout *layout = [[DraggableCircleLayout
alloc] init];

    collectionView = [[UICollectionView alloc]
initWithFrame:CGRectMake(10, 50, 300, 400)
collectionViewLayout:layout];

    collectionView.backgroundColor = [UIColor grayColor];
    collectionView.draggable = YES;

    [collectionView registerClass:[UICollectionViewCell
class] forCellWithReuseIdentifier:@"LetterCell"];
    [collectionView registerClass:[UICollectionReusableView
class] forSupplementaryViewOfKind:@"FirstSupplementary"
withReuseIdentifier:@"ReuseID"];
    [collectionView registerClass:[UICollectionReusableView
class] forSupplementaryViewOfKind:@"SecondSupplementary"
withReuseIdentifier:@"ReuseID"];

    cvDataSource = [CollectionViewDataSource alloc];
    collectionView.dataSource = cvDataSource;
    collectionView.delegate = cvDataSource;

    [self.view addSubview:collectionView];

    UITapGestureRecognizer *tapRecognizer =
[[UITapGestureRecognizer alloc] initWithTarget:self
action:@selector(tapGestureHandler:)];
    [collectionView addGestureRecognizer:tapRecognizer];
}

- (void)tapGestureHandler:(UITapGestureRecognizer *)sender
{
    CGPoint point = [sender locationInView:collectionView];
    NSIndexPath *tappedCellPath = [collectionView
indexPathForItemAtPoint:point];

    NSMutableArray *letterArray = [self getLetterArray];
    if(tappedCellPath)
    {
        //删除点击的cell
        [letterArray
removeObjectAtIndex:tappedCellPath.item];
        [collectionView deleteItemsAtIndexPaths:[NSArray
arrayWithObject:tappedCellPath]];
    }
    else
    {
        //如果点击空白处，在末尾添加一个随机小写字母
        unichar asciiX = (unichar)[self getRandomNumber:97
to:97+26];
        [letterArray addObject:[NSString
stringWithFormat:@"%C",asciiX]];
```

```
        NSIndexPath *path = [NSIndexPath
indexPathForItem:letterArray.count-1 inSection:0];
        [collectionView insertItemsAtIndexPaths:[NSArray
arrayWithObject:path]];
    }
}

- (NSMutableArray *)getLetterArray
{
    AppDelegate *appDelegate = (AppDelegate
*)[[UIApplication sharedApplication] delegate];

    return appDelegate.letterArray;
}

- (int)getRandomNumber:(int)from to:(int)to
{
    return (int)(from + (arc4random() % (to-from)));
}

- (void)didReceiveMemoryWarning {
    [super didReceiveMemoryWarning];
    // Dispose of any resources that can be recreated.
}

@end
```

```
#import <UIKit/UIKit.h>
#import "UICollectionView+Draggable.h"

@interface CollectionViewDataSource :
NSObject<UICollectionViewDataSource_Draggable,
UICollectionViewDelegate>

@end
```

```
#import <Foundation/Foundation.h>
#import "AppDelegate.h"
#import "CollectionViewDataSource.h"

@implementation CollectionViewDataSource


- (NSInteger)numberOfSectionsInCollectionView:
```

```
(UICollectionView *)collectionView
{
    return 1;
}

- (NSInteger)collectionView:(UICollectionView
*)collectionView numberOfItemsInSection:(NSInteger)section
{
    return [self getLetterArray].count;
}

- (UICollectionViewCell *)collectionView:(UICollectionView
*)collectionView cellForItemAtIndexPath:(NSIndexPath
*)indexPath
{
    UICollectionViewCell *cell = [collectionView
dequeueReusableCellWithReuseIdentifier:@"LetterCell"
forIndexPath:indexPath];

    //先移除可重用cell里面的子元素(否则会出现新旧交叠)
    [cell.subviews
makeObjectsPerformSelector:@selector(removeFromSuperview)];

    cell.backgroundColor = [UIColor yellowColor];

    UILabel *label = [[UILabel alloc] init];
    label.text = [[self getLetterArray]
objectAtIndex:indexPath.row];
    label.font = [UIFont systemFontOfSize:12];
    [label sizeToFit];
    label.center = CGPointMake(cell.bounds.size.width/2,
cell.bounds.size.height/2);
    [cell addSubview:label];

    return cell;
}

- (UICollectionReusableView *)collectionView:
(UICollectionView *)collectionView
viewForSupplementaryElementOfKind:(NSString *)kind
atIndexPath:(NSIndexPath *)indexPath
{
    UICollectionReusableView *view = [collectionView
dequeueReusableSupplementaryViewOfKind:kind
withReuseIdentifier:@"ReuseID" forIndexPath:indexPath];

    view.backgroundColor = [UIColor greenColor];

    UILabel *label = [[UILabel alloc] init];
    label.text = kind;
    label.font = [UIFont systemFontOfSize:24];
    [label sizeToFit];
    label.center = CGPointMake(view.bounds.size.width/2,
view.bounds.size.height/2);
```

```
    [view addSubview:label];

    return view;
}


- (void)collectionView:(UICollectionView *)collectionView
didSelectItemAtIndexPath:(NSIndexPath *)indexPath
{
    NSLog(@"你选择了");

//    [self.myArray removeObjectAtIndex:indexPath.row];
//
//    [collectionView deleteItemsAtIndexPaths:[NSArray
arrayWithObject:indexPath]];
}

- (BOOL)collectionView:(LSCollectionViewHelper
*)collectionView canMoveItemAtIndexPath:(NSIndexPath
*)indexPath
{
    NSLog(@"canMoveItemAtIndexPath");
    return YES;
}

- (void)collectionView:(LSCollectionViewHelper
*)collectionView moveItemAtIndexPath:(NSIndexPath
*)fromIndexPath toIndexPath:(NSIndexPath *)toIndexPath
{
    NSLog(@"moveItemAtIndexPath");

    NSMutableArray *data = [self getLetterArray];

    NSNumber *index = [data
objectAtIndex:fromIndexPath.item];
    [data removeObjectAtIndex:fromIndexPath.item];
    [data insertObject:index atIndex:toIndexPath.item];
}

- (NSMutableArray *)getLetterArray
{
    AppDelegate *appDelegate = (AppDelegate
*)[[UIApplication sharedApplication] delegate];

    return appDelegate.letterArray;
}

@end
```

```
#import <UIKit/UIKit.h>
```

```
@interface MyCollectionReusableView :
UICollectionReusableView

@end
```

```
#import "MyCollectionReusableView.h"

@implementation MyCollectionReusableView

- (instancetype)initWithFrame:(CGRect)frame
{
    self = [super initWithFrame:frame];

    if (self)
    {
        self.backgroundColor = [UIColor orangeColor];

        UILabel *label = [[UILabel alloc] init];
        label.text = @"Decoration View";
        label.font = [UIFont systemFontOfSize:18];
        [label sizeToFit];
        label.center = CGPointMake(frame.size.width/2,
frame.size.height/2);
        [self addSubview:label];
    }

    return self;
}

@end
```

```
#import <UIKit/UIKit.h>

@interface CircleLayout : UICollectionViewLayout

@end
```

```
#import "AppDelegate.h"
#import "CircleLayout.h"
#import "CollectionViewDataSource.h"
#import "MyCollectionReusableView.h"

@interface CircleLayout()
{
    CGSize cvSize;
    CGPoint cvCenter;
    CGFloat radius;
    NSInteger cellCount;
}

@property (strong, nonatomic) NSMutableArray *indexPathsToAnimate;

@end

@implementation CircleLayout

- (void)prepareLayout
{
    [super prepareLayout];

    [self registerClass:[MyCollectionReusableView class] forDecorationViewOfKind:@"MyDecoration"];

    cvSize = self.collectionView.frame.size;
    cellCount = [self.collectionView numberOfItemsInSection:0];
    cvCenter = CGPointMake(cvSize.width / 2.0, cvSize.height / 2.0);
    radius = MIN(cvSize.width, cvSize.height) / 2.5;
}

- (CGSize)collectionViewContentSize
{
    return self.collectionView.bounds.size;
}

- (NSArray *)layoutAttributesForElementsInRect:(CGRect)rect
{
    NSMutableArray *array = [NSMutableArray array];

    //add cells
    for (int i=0; i<cellCount; i++)
    {
        NSIndexPath *indexPath = [NSIndexPath indexPathForItem:i inSection:0];

        UICollectionViewLayoutAttributes *attributes = [self layoutAttributesForItemAtIndexPath:indexPath];

        [array addObject:attributes];
    }

    //add first supplementaryView
    NSIndexPath *indexPath = [NSIndexPath indexPathForItem:0 inSection:0];
    UICollectionViewLayoutAttributes *attributes = [self layoutAttributesForSupplementaryViewOfKind:@"FirstSupplementary" atIndexPath:indexPath];
    [array addObject:attributes];
    //add second supplementaryView
    attributes = [self
layoutAttributesForSupplementaryViewOfKind:@"SecondSupplement
ary" atIndexPath:indexPath];
    [array addObject:attributes];

    //add decorationView
    attributes = [self
layoutAttributesForDecorationViewOfKind:@"MyDecoration"
atIndexPath:indexPath];
    [array addObject:attributes];

    return array;
}

- (UICollectionViewLayoutAttributes
*)layoutAttributesForItemAtIndexPath:(NSIndexPath *)indexPath
{
    UICollectionViewLayoutAttributes *attributes =
[UICollectionViewLayoutAttributes
layoutAttributesForCellWithIndexPath:indexPath];

    attributes.size = CGSizeMake(20, 20);
    attributes.center = CGPointMake(cvCenter.x + radius *
cosf(2 * indexPath.item * M_PI / cellCount),
                                    cvCenter.y + radius *
sinf(2 * indexPath.item * M_PI / cellCount));

    return attributes;
}

- (UICollectionViewLayoutAttributes
*)layoutAttributesForSupplementaryViewOfKind:(NSString
*)elementKind atIndexPath:(NSIndexPath *)indexPath
{
    UICollectionViewLayoutAttributes *attributes =
[UICollectionViewLayoutAttributes
layoutAttributesForSupplementaryViewOfKind:elementKind
withIndexPath:indexPath];

    attributes.size = CGSizeMake(260, 40);
    if([elementKind isEqual:@"FirstSupplementary"])
    {
        attributes.center = CGPointMake(cvSize.width/2, 40);
    }
    else
    {
        attributes.center = CGPointMake(cvSize.width/2,
cvSize.height-40);
    }

    return attributes;
}
```

```
- (UICollectionViewLayoutAttributes
*)layoutAttributesForDecorationViewOfKind:(NSString
*)elementKind atIndexPath:(NSIndexPath *)indexPath
{
    UICollectionViewLayoutAttributes *attributes =
[UICollectionViewLayoutAttributes
layoutAttributesForDecorationViewOfKind:elementKind
withIndexPath:indexPath];

    attributes.size = CGSizeMake(140, 40);
    attributes.center = CGPointMake(cvSize.width/2,
cvSize.height/2);

    return attributes;
}

//当边界更改时是否更新布局
- (BOOL)shouldInvalidateLayoutForBoundsChange:
(CGRect)newBounds
{
    CGRect oldBounds = self.collectionView.bounds;

    if (CGRectGetWidth(newBounds) !=
CGRectGetWidth(oldBounds))
    {
        return YES;
    }

    return NO;
}

//通知布局，collection view里有元素即将改变，这里可以收集改变的元素
indexPath和action类型。
-(void)prepareForCollectionViewUpdates:(NSArray *)updateItems
{
    [super prepareForCollectionViewUpdates:updateItems];

    NSMutableArray *indexPaths = [NSMutableArray array];

    for(UICollectionViewUpdateItem *updateItem in
updateItems)
    {
        //UICollectionUpdateActionInsert,
        //UICollectionUpdateActionDelete,
        //UICollectionUpdateActionReload,
        //UICollectionUpdateActionMove,
        //UICollectionUpdateActionNone

        NSLog(@"before index:%d,after index:%d,action:%d",
updateItem.indexPathBeforeUpdate.row,updateItem.indexPathAfte
rUpdate.row,updateItem.updateAction);

        switch (updateItem.updateAction) {
            case UICollectionUpdateActionInsert:
                [indexPaths
addObject:updateItem.indexPathAfterUpdate];
```

```
                break;
            case UICollectionUpdateActionDelete:
                [indexPaths
addObject:updateItem.indexPathBeforeUpdate];
                break;
            case UICollectionUpdateActionMove:
                [indexPaths
addObject:updateItem.indexPathBeforeUpdate];
                [indexPaths
addObject:updateItem.indexPathAfterUpdate];
                break;
            default:
                NSLog(@"unhandled case: %@", updateItem);
                break;
        }
    }

    self.indexPathsToAnimate = indexPaths;
}

//当一个元素被插入collection view时，返回它的初始布局，这里可以加入一
些动画效果。
- (UICollectionViewLayoutAttributes
*)initialLayoutAttributesForAppearingItemAtIndexPath:
(NSIndexPath *)itemIndexPath
{
    UICollectionViewLayoutAttributes *attr = [self
layoutAttributesForItemAtIndexPath:itemIndexPath];

    if([self.indexPathsToAnimate
containsObject:itemIndexPath])
    {
        attr.transform =
CGAffineTransformRotate(CGAffineTransformMakeScale(10,10),M_P
I);
        attr.center =
CGPointMake(CGRectGetMidX(self.collectionView.bounds),
CGRectGetMidY(self.collectionView.bounds));
        [self.indexPathsToAnimate
removeObject:itemIndexPath];
    }

    return attr;
}



- (NSArray *)getLetterArray
{
    AppDelegate *appDelegate = (AppDelegate
*)[[UIApplication sharedApplication] delegate];

    return appDelegate.letterArray;
}

@end
```

```
#import "CircleLayout.h"
#import "UICollectionViewLayout_Warpable.h"

@interface DraggableCircleLayout : CircleLayout
<UICollectionViewLayout_Warpable>

@property (readonly, nonatomic) LSCollectionViewLayoutHelper
*layoutHelper;

@end
```

```
#import "DraggableCircleLayout.h"
#import "LSCollectionViewLayoutHelper.h"

@interface DraggableCircleLayout()
{
    LSCollectionViewLayoutHelper *_layoutHelper;
}
@end

@implementation DraggableCircleLayout

- (LSCollectionViewLayoutHelper *)layoutHelper
{
    if(_layoutHelper == nil) {
        _layoutHelper = [[LSCollectionViewLayoutHelper
alloc] initWithCollectionViewLayout:self];
    }
    return _layoutHelper;
}

- (NSArray *)layoutAttributesForElementsInRect:(CGRect)rect
{
    return [self.layoutHelper
modifiedLayoutAttributesForElements:[super
layoutAttributesForElementsInRect:rect]];
}

@end
```

分类: iOS

好文要顶 关注我 收藏该文

wayne23
关注 - 0
粉丝 - 84

+加关注

0 0

(请您对文章做出评价)

« 上一篇: Block的基本用法
» 下一篇: 数组的排序及筛选

posted @ 2014-10-30 11:38 wayne23 阅读(15560) 评论(3) 编辑 收藏

评论列表

#1楼 2015-10-25 11:26 沫粢泪虞 回复 引用

有demo吗

支持(0) 反对(0)

#2楼 2015-12-28 18:57 小雨点软件 回复 引用

太感谢了，帮大忙了

支持(0) 反对(0)

#3楼 2016-04-19 17:31 星红 回复 引用

O(∩_∩)O谢谢

支持(0) 反对(0)

刷新评论 刷新页面 返回顶部

发表评论

昵称: Aldridge1

评论内容:

提交评论 退出登录 订阅评论

[Ctrl+Enter快捷键提交]

最新IT新闻:
· 如何重新发明Web?
· 电影立功 《魔兽世界》重回第一网游王座
· 中国最大直径火箭垂直总装厂房首次公开：大门81米高！
· Apple Pay网页版即将上线 以后"剁手"更方便
· Uber真的上天了，在巴西推出飞的服务
» 更多新闻...

最新知识库文章:
· 让我们来谈谈分工
· 一个32岁入门的70后程序员给我的启示
· 技术发展瓶颈的突破
· 高效编程之道：好好休息
· 快速学习者的高效学习策略
» 更多知识库文章...