MapKit 使用技巧

1、概述

插入MapView，设置Delegate（一般为Controller），Annotations记录兴趣位置点（AnnotationView用来显示兴趣位置点），annotation是可选的，选中的annotation会显示callout，用来显示信息。

2、设置地图显示类型：

mapView.mapType = MKMapTypeStandard; 标注

mapView.mapType = MKMapTypeSatellite; 卫星

mapView.mapType = MKMapTypeHybrid; 混合

3、显示用户位置

设置为可以显示用户位置：

mapView.showsUserLocation = YES;

判断用户当前位置是否可见（只读属性）：

userLocationVisible

得到用户位置坐标：当userLocationVisible为YES时

CLLocationCoordinate2D coords = mapView.userLocation.location.coordinate;

4、坐标范围

MKCoordinateRegion用来设置坐标显示范围。

包括两部分：Center（CLLocationCoordinate2D struct，包括latitude和longitude），坐标中心

和Span（MKCoordinateSpan struct，包括latitudeDelta和longitudeDelta），缩放级别

MKCoordinateRegion viewRegion = MKCoordinateRegionMakeWithDistance(center,2000, 2000);

以上代码创建一个以center为中心，上下各1000米，左右各1000米得区域，但其是一个矩形，不符合MapView的横纵比例

MKCoordinateRegion adjustedRegion = [mapView regionThatFits:viewRegion];

以上代码创建出来一个符合MapView横纵比例的区域

[mapView setRegion:adjustedRegion animated:YES];

以上代码为：最终显示该区域

5、Delegate

使用MapView须符合MKMapViewDelegate协议

5.1、地图加载Delegate

当需要从Google服务器取得新地图时

mapViewWillStartLoadingMap:

当成功地取得地图后

mapViewDidFinishLoadingMap:

当取得地图失败后（建议至少要实现此方法）

mapViewDidFailLoadingMap:withError:

5.2、范围变化Delegate

当手势开始（拖拽，放大，缩小，双击）

mapView:regionWillChangeAnimated:

当手势结束（拖拽，放大，缩小，双击）

mapView:regionDidChangeAnimated:

判断坐标是否在MapView显示范围内：

CLLocationDegrees leftDegrees = mapView.region.center.longitude –(mapView.region.span.longitudeDelta / 2.0);

CLLocationDegrees rightDegrees = mapView.region.center.longitude +(mapView.region.span.longitudeDelta / 2.0);

CLLocationDegrees bottomDegrees = mapView.region.center.latitude –(mapView.region.span.latitudeDelta / 2.0);

CLLocationDegrees topDegrees = self.region.center.latitude +(mapView.region.span.latitudeDelta / 2.0);

if (leftDegrees > rightDegrees) { // Int'l Date Line in View

leftDegrees = -180.0 - leftDegrees;

if (coords.longitude > 0) // coords to West of Date Line

coords.longitude = -180.0 - coords.longitude;

}

If (leftDegrees <= coords.longitude && coords.longitude <= rightDegrees && bottomDegrees <= coords.latitude && coords.latitude <= topDegrees) {

// 坐标在范围内

}

6、Annotation

Annotation包含两部分：Annotation Object和Annotation View

Annotation Object必须符合协议MKAnnotation，包括两个方法：title和subtitle，分别用于显示注释的标题和子标题。还有coordinate属性，返回CLLocationCoordinate2D，表示Annotation的位置

然后，需使用mapView:viewForAnnotation: 方法来返回MKAnnotationView或者MKAnnotationView的子类用来显示Annotation（注意：这里显示的不是选中Annotation后的弹出框）

你可以子类化MKAnnotationView，然后再drawRect：方法里面进行自己的绘制动作（这个方法很蠢）

你完全可以实例化一个MKAnnotationView，然后更改它的image属性，这样很简单。

7、添加移除Annotation

添加一个Annotation

[mapView addAnnotation:annotation];

添加一个Annotation数组

[mapView addAnnotations:[NSArray arrayWithObjects:annotation1, annotation2, nil]];

移除一个Annotation

removeAnnotation:

移除一个Annotation数组

removeAnnotations:

移除所有Annotation

[mapView removeAnnotations:mapView.annotations];

8、选中Annotation

一次只能有一个Annotation被选中，选中后会出现CallOut（浮动框）

简单的CallOut显示Title和SubTitle，但你也可以自定义一个UIView作为CallOut（与自定义的TableViewCell一样）

可通过代码选中Annotation：

selectAnnotation:animated:

或者取消选择：

deselectAnnotation:animated:

9、显示Annotation

通过mapView:viewForAnnotation: 方法显示Annotation，每在MapView中加入一个Annotation，就会调用此方法

示例（与tableView:cellForRowAtIndexPath: 很相似）

- (MKAnnotationView \*) mapView:(MKMapView \*)theMapView viewForAnnotation:(id <MKAnnotation>) annotation {

static NSString \*placemarkIdentifier = @"my annotation identifier";

if ([annotation isKindOfClass:[MyAnnotation class]]) {

MKAnnotationView \*annotationView = [theMapView dequeueReusableAnnotationViewWithIdentifier:placemarkIdentifier];

if (annotationView == nil) {

annotationView = [[MKAnnotationView alloc] initWithAnnotation:annotation reuseIdentifier:placemarkIdentifier];

annotationView.image = [UIImage imageNamed:@"blood\_orange.png"];

}

else

annotationView.annotation = annotation;

return annotationView;

}

return nil;

}

10、取得真实地址

示例：

初始化MKReverseGeocoder

MKReverseGeocoder \*geocoder = [[MKReverseGeocoder alloc] initWithCoordinate:coordinates];

geocoder.delegate = self;

[geocoder start];

如果无法处理坐标，则调用reverseGeocoder:didFailWithError: 方法

- (void)reverseGeocoder:(MKReverseGeocoder \*)geocoder didFailWithError:(NSError \*)error {

NSLog(@"Error resolving coordinates: %@", [error localizedDescription]);

geocoder.delegate = nil;

[geocoder autorelease];

}

如果成功，则调用reverseGeocoder:didFindPlacemark: 并把信息存储在MKPlacemark 中

didFindPlacemark:(MKPlacemark \*)placemark {

NSString \*streetAddress = placemark.thoroughfare;

NSString \*city = placemark.locality;

NSString \*state = placemark.administrativeArea;

NSString \*zip = placemark.postalCode;

// Do something with information

geocoder.delegate = nil;

[geocoder autorelease];

}

最后奉献一枚APP源码，由论坛会员qdvictory分享

对于初步接触mapview编程的朋友比较有帮助。

主要知识点：

1.mapview的简单使用。

2.google api的相关调用。

3.mapview大头钉的操作，长按插入大头钉等。

App Store：http://itunes.apple.com/cn/app/telllocation/id467293790?l=en&mt=8

git链接：https://github.com/qdvictory/TellLocation