地理围栏

地理围栏（Geogencing）也叫区域监视（regional monitoring），指的是能够知道设备已进入或离开指定的地图区域。iOS在siri中充分利用了这项功能，使其能够完成类似于下面的任务“在我离开办公室时提醒我带上面包”；“在我回到家时提醒我将烤肉放进烤箱”。iOS还在Passbook中使用了地址围栏功能，让用户能够再主屏幕上看到相关的凭证。

在ios8中如果要使用该功能，在申请授权时，需使用`requestAlwaysAuthorization`,因为`requestWhenInUseAuthorization`授权不支持该功能。

定义边界

用GPX文件模拟运行

```
<?xml version="1.0"?>
<gpx version="1.1" creator="Xcode">
    <wpt lat="34.76705250" lon="113.76887056"></wpt>
    <wpt lat="34.77" lon="113.76887056"></wpt>
    <wpt lat="34.78" lon="113.76887056"></wpt>
    <wpt lat="34.79" lon="113.76887056"></wpt>
    <wpt lat="34.80" lon="113.76887056"></wpt>
    <wpt lat="34.81" lon="113.76887056"></wpt>
    <wpt lat="34.82" lon="113.76887056"></wpt>
    <wpt lat="34.83" lon="113.76887056"></wpt>
    <wpt lat="34.84" lon="113.76887056"></wpt>
    <wpt lat="34.85" lon="113.76887056"></wpt>
    <wpt lat="34.86" lon="113.76887056"></wpt>
    <wpt lat="34.87" lon="113.76887056"></wpt>
</gpx>
```

开启定位服务，确定当前位置；

CLRegion

添加地理围栏区域；

MKCircle

添加圆形覆盖视图，辅助显示围栏区域

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    //开启定位服务
    self.mapView.delegate = self;
    self.locationManager = [[CLLocationManager alloc] init];
    if ([CLLocationManager authorizationStatus] == kCLAuthorizationStatusNotDetermined) {
        [self.locationManager requestAlwaysAuthorization];
    }
    self.locationManager.desiredAccuracy = kCLLocationAccuracyBest;
    self.locationManager.distanceFilter = 10;
    self.locationManager.delegate = self;
    if ([CLLocationManager locationServicesEnabled]) {
        [self.locationManager startUpdatingLocation];
    }else{
        NSLog(@"无法使用定位服务");
    }

    //添加地理围栏
    CLLocationCoordinate2D coordinate = CLLocationCoordinate2DMake(34.77, 113.76887056);
    MKCoordinateRegion region = MKCoordinateRegionMake(coordinate, MKCoordinateSpanMake(0.1, 0.1));
    [self.mapView setRegion:region animated:YES];

    CLRegion *regioncl = [[CLCircularRegion alloc] initWithCenter:coordinate radius:1000 identifier:@"regioncl"];
    [self.locationManager startMonitoringForRegion:regioncl];

    添加圆形覆盖层，辅助显示围栏区域
    MKCircle *circle = [MKCircle circleWithCenterCoordinate:coordinate radius:1000];
    [self.mapView addOverlay:circle];



}
```

更新当前位置

```
-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(N
SArray *)locations{
    CLLocation *location = locations.lastObject;

    MKPointAnnotation *point = [[MKPointAnnotation alloc] init];
    point.coordinate = location.coordinate;

    [self.mapView addAnnotation:point];

    if (self.nowAnnotation) {
        [self.mapView removeAnnotation:self.nowAnnotation];
    }

    self.nowAnnotation = point;
}
```

MKCircleRenderer

显示圆形覆盖层视图

```
-(MKOverlayRenderer *)mapView:(MKMapView *)mapView rendererForOverlay:(id
<MKOverlay>)overlay{
    if ([overlay isKindOfClass:[MKCircle class]]) {
        MKCircleRenderer *renderer = [[MKCircleRenderer alloc] initWithCi
rcle:overlay];
        renderer.lineWidth = 2.f;
        renderer.strokeColor = [UIColor redColor];
        renderer.fillColor = [UIColor colorWithRed:0 green:1 blue:0 alpha
:.5];
        return renderer;
    }
    return nil;
}
```

设备进入或者离开监视区域后，位置管理器将这一点告诉其委托：调用方法:locationManager:didEnterRegion: 或 locationManager:didEnterRegion:获取监视区域的标示符。这个标识符是在让位置管理器对区域监视时指定。

```
-(void)locationManager:(CLLocationManager *)manager didEnterRegion:(CLReg
ion *)region{
    NSLog(@"====%@", region.identifier);
}

-(void)locationManager:(CLLocationManager *)manager didExitRegion:(CLRegi
on *)region{
    NSLog(@">>>>%@", region.identifier);
}
```