

开启定位

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.

    //初始化容器
    self.typeAnnotations = [[NSMutableArray alloc] init];
    self.locations = [[NSMutableArray alloc] init];

    self.manager = [[CLLocationManager alloc] init];
    // 向用户请求权限，CLLocationWhenInUseUsageDescription必须要添加到info.plist
    [self.manager requestWhenInUseAuthorization];
    self.manager.delegate = self;
    //设置精确度
    self.manager.desiredAccuracy = kCLLocationAccuracyBest;
    //设置更新频率，当移动距离超过15m的时候，返回新的点
    [self.manager setDistanceFilter:15];

    self.mapView.delegate = self;
}
```

自定义Annotation

```
#import <Foundation/Foundation.h>
#import <MapKit/MapKit.h>

typedef enum{
    AnnotationGo,
    AnnotationSuspend,
    AnnotationEnd,
    AnnotationNow
} AnnotationType;

@interface QYAnnotation : NSObject<MKAnnotation>

@property (nonatomic, assign)CLLocationCoordinate2D coordinate;

@property (nonatomic, assign)AnnotationType type;

@end
```

位置事件所做的处理

```
#pragma mark - location delegate

-(void)locationManager:(CLLocationManager *)manager didUpdateLocations:(NSArray *)locations{
    CLLocation *location = locations.lastObject;
    [self.locations addObject:location];

    QYAnnotation *annotation = [[QYAnnotation alloc] init];
    annotation.coordinate = location.coordinate;

    if (self.nowLocation) {
        // 将现有的点移除;
        if (self.nowAnnotation) {
            [self.mapView removeAnnotation:self.nowAnnotation];
        }
        annotation.type = AnnotationNow;
        [self.mapView addAnnotation:annotation];
        self.nowAnnotation = annotation;
    }else {
        //设置region的中心
        //设置region
        CLLocationCoordinate2D coordinate =location.coordinate;
        MKCoordinateRegion region = MKCoordinateRegionMake(coordinate, MKCoordinateSpanMake(0.05f, 0.05f));
        self.mapView.region = region;

        annotation.type = AnnotationGo;
        [self.mapView addAnnotation:annotation];
        [self.typeAnnotations addObject:annotation];
    }

    MKMapPoint *pointArray = malloc(sizeof(MKMapPoint)*self.locations.count);
    for (int i = 0; i < self.locations.count; i++) {
        CLLocation *loca = [self.locations objectAtIndex:i];
        pointArray[i] = MKMapPointForCoordinate([loca coordinate]);
    }

    MKPolyline *route = [MKPolyline polylineWithPoints:pointArray count:self.locations.count];
    free(pointArray);
}
```

```

[self.mapView addOverlay:route];
self.nowLocation = location;

}

```

返回注释视图

```

#pragma mark - map delegate

-(MKAnnotationView *)mapView:(MKMapView *)mapView viewForAnnotation:(id<MKAnnotation>)annotation{
    if ([annotation isKindOfClass:[QYAnnotation class]]) {

        QYAnnotation *annotation_ = (QYAnnotation *)annotation;
        static NSString *Identifier = @"QYAnnotation";
        MKAnnotationView *annotationView = [mapView dequeueReusableAnnotationViewWithIdentifier:Identifier];
        if (!annotationView) {
            annotationView = [[MKAnnotationView alloc] initWithAnnotation:annotation reuseIdentifier:Identifier];
        }

        switch (annotation_.type) {
            case AnnotationGo:
            {
                [annotationView setCenterOffset:CGPointMake(0, -12)];
                [annotationView setImage:[UIImage imageNamed:@"map_start_
icon"]];
                [annotationView setDraggable:YES];
            }
            break;
            case AnnotationNow:
            {
                [annotationView setImage:[UIImage imageNamed:@"currentloc
ation"]];
            }
            break;
            case AnnotationSuspend:{
                [annotationView setCenterOffset:CGPointMake(0, -12)];
                [annotationView setImage:[UIImage imageNamed:@"map_susoen
d_icon"]];
            }
        }
    }
}

```

```

        break;
    case AnnotationEnd:{
        [annotationView setCenterOffset:CGPointMake(0, -12)];
        [annotationView setImage:[UIImage imageNamed:@"map_stop_i
con"]];
    }
    break;
    default:
        break;
    }

    return annotationView;
}
return nil;
}

```

返回覆盖层视图

```

-(MKOverlayRenderer *)mapView:(MKMapView *)mapView viewForOverlay:(id<MKO
verlay>)overlay{
    if ([overlay isKindOfClass:[MKPolyline class]]) {
        MKPolyline *line = (MKPolyline *)overlay;

        MKPolylineRenderer *polylineRenderer =
            [[MKPolylineRenderer alloc] initWithPolyline:line];

        [polylineRenderer setLineWidth:3.0];
        [polylineRenderer setFillColor:[UIColor blueColor]];
        [polylineRenderer setStrokeColor:[UIColor blueColor]];
        return polylineRenderer;
    }
    return nil;
}

```

三个按钮的事件

```
#pragma mark - action
```

```
- (IBAction)run:(id)sender {  
    NSLog(@"%d",[CLLocationManager authorizationStatus]);  
    [self.manager startUpdatingLocation];  
}  
  
- (IBAction)pause:(id)sender {  
    //添加一个暂停点  
    QYAnnotation *annotation = [[QYAnnotation alloc] init];  
    annotation.type = AnnotationSuspend;  
    annotation.coordinate = self.nowLocation.coordinate;  
    [self.mapView addAnnotation:annotation];  
    [self.typeAnnotations addObject:annotation];  
    [self.manager stopUpdatingLocation];  
}  
  
- (IBAction)stop:(id)sender {  
    //添加一个结束点  
    QYAnnotation *annotation = [[QYAnnotation alloc] init];  
    annotation.type = AnnotationEnd;  
    annotation.coordinate = self.nowLocation.coordinate;  
    [self.mapView addAnnotation:annotation];  
    [self.typeAnnotations addObject:annotation];  
    [self.manager stopUpdatingLocation];  
}
```