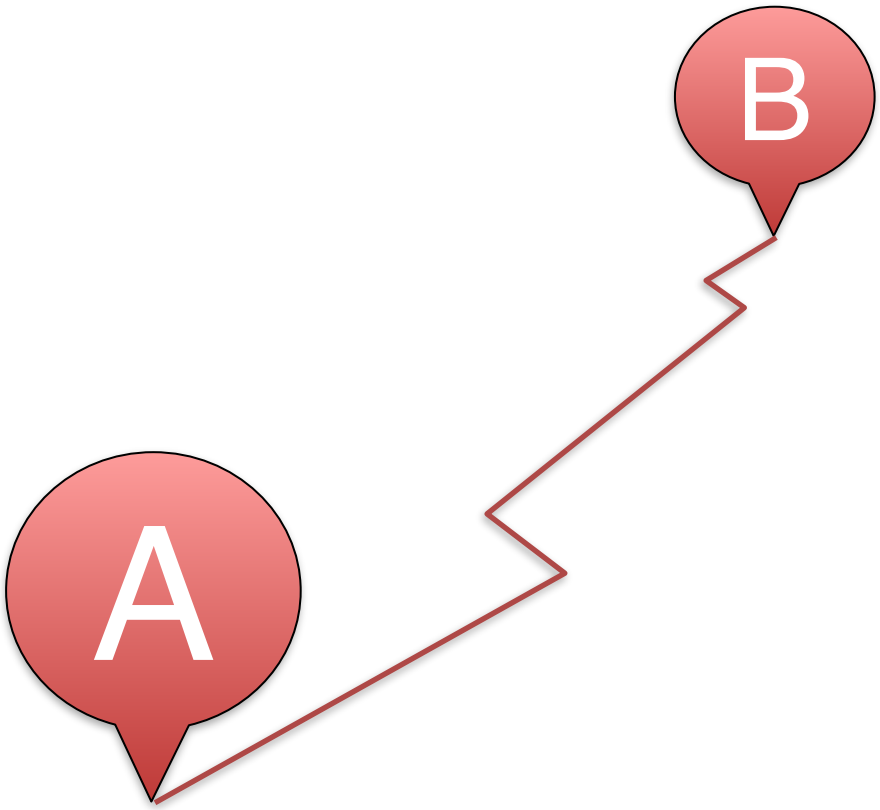
高德
AutoNavi

iOS SDK公开课

第四课 路线规划

路径规划



目录

- 选择目的地选择
  - 长按手势
  - 坐标转换
- 路线规划
  - 搜索路线
  - 绘制路线

选择目的地

利用长按手势选择目的地

1. 取得手势屏幕坐标
2. 屏幕坐标转经纬度坐标，获得目的地经纬度

坐标转换接口

MAMapView中

```
/*!
 @brief 将经纬度转换为指定view坐标系的坐标
 @param coordinate 经纬度
 @param view 指定的view
 @return 基于指定view坐标系的坐标
 */
- (CGPoint)convertCoordinate:
(CLLocationCoordinate2D)coordinate toPointToView:(UIView
*)view;

/*!
 @brief 将指定view坐标系的坐标转换为经纬度
 @param point 指定view坐标系的坐标
 @param view 指定的view
 @return 经纬度
 */
- (CLLocationCoordinate2D)convertPoint:(CGPoint)point
toCoordinateFromView:(UIView *)view;
```

选择目的地

长按手势初始化

```
@interface ViewController ()<MAMapViewDelegate, AMapSearchDelegate, UITableViewDataSource,
UITableViewDelegate, UIGestureRecognizerDelegate>
{
…
    UILongPressGestureRecognizer *_longPressGesture;
    MAPointAnnotation *_destinationPoint;
}
@end

- (void)initAttributes
{
    _annotations = [NSMutableArray array];
    _pois = nil;

    _longPressGesture = [[UILongPressGestureRecognizer alloc] initWithTarget:self
action:@selector(handleLongPress:)];
    _longPressGesture.delegate = self;
    [_mapView addGestureRecognizer:_longPressGesture];
}
```

手势 、目的地标记

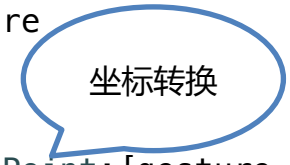
长按手势响应

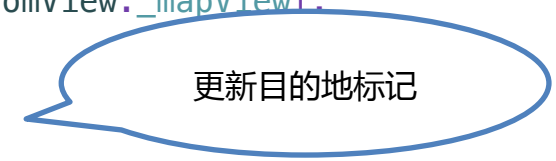
```
- (void)handleLongPress:(UILongPressGestureRecognizer *)gesture
{
    if (gesture.state == UIGestureRecognizerStateBegan)
    {
        CGPoint p = [gesture locationInView:_mapView];
        NSLog(@"press on (%f, %f)", p.x, p.y);
    }

}
```

选择目的地

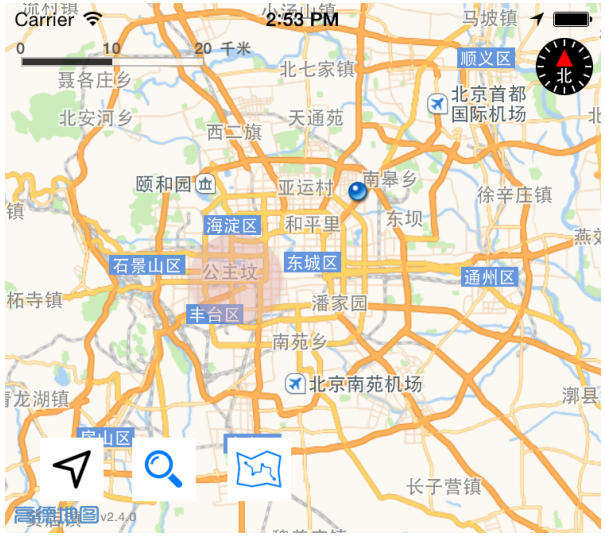
在长按手势响应中

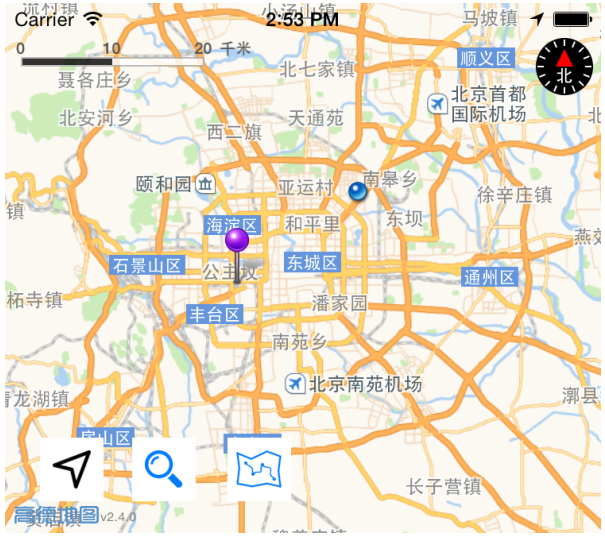
```
- (void)handleLongPress:(UILongPressGestureRecognizer *)gesture
{
    if (gesture.state == UIGestureRecognizerStateBegan)
    {
        CLLocationCoordinate2D coordinate = [_mapView convertPoint:[gesture
locationInView:_mapView] toCoordinateFromView:_mapView];

        // 添加标注
        if (_destinationPoint != nil)
        {
            // 清理
            [_mapView removeAnnotation:_destinationPoint];
            _destinationPoint = nil;
        }

        _destinationPoint = [[MAPointAnnotation alloc] init];
        _destinationPoint.coordinate = coordinate;
        _destinationPoint.title = @"Destination";

        [_mapView addAnnotation:_destinationPoint];
    }

}
```





选择目的地


Carrier
2:53 PM
0
10
20 千米
北
马坡镇
顺义区
北京首都
国际机场
聂各庄乡
北七家镇
北安河乡
西二旗
天通苑
颐和园
亚运村
南皋乡
徐辛庄镇
海淀区
和平里
东坝
石景山区
公主坟
东城区
通州区
柏寺镇
潘家园
丰台区
南苑乡
北京南苑机场
龙湖镇
长子营镇
v2.4.0


Carrier
2:53 PM
0
10
20 千米
北
马坡镇
顺义区
北京首都
国际机场
聂各庄乡
北七家镇
北安河乡
西二旗
天通苑
颐和园
亚运村
南皋乡
徐辛庄镇
海淀区
和平里
东坝
石景山区
公主坟
东城区
通州区
柏寺镇
潘家园
丰台区
南苑乡
北京南苑机场
龙湖镇
长子营镇
v2.4.0

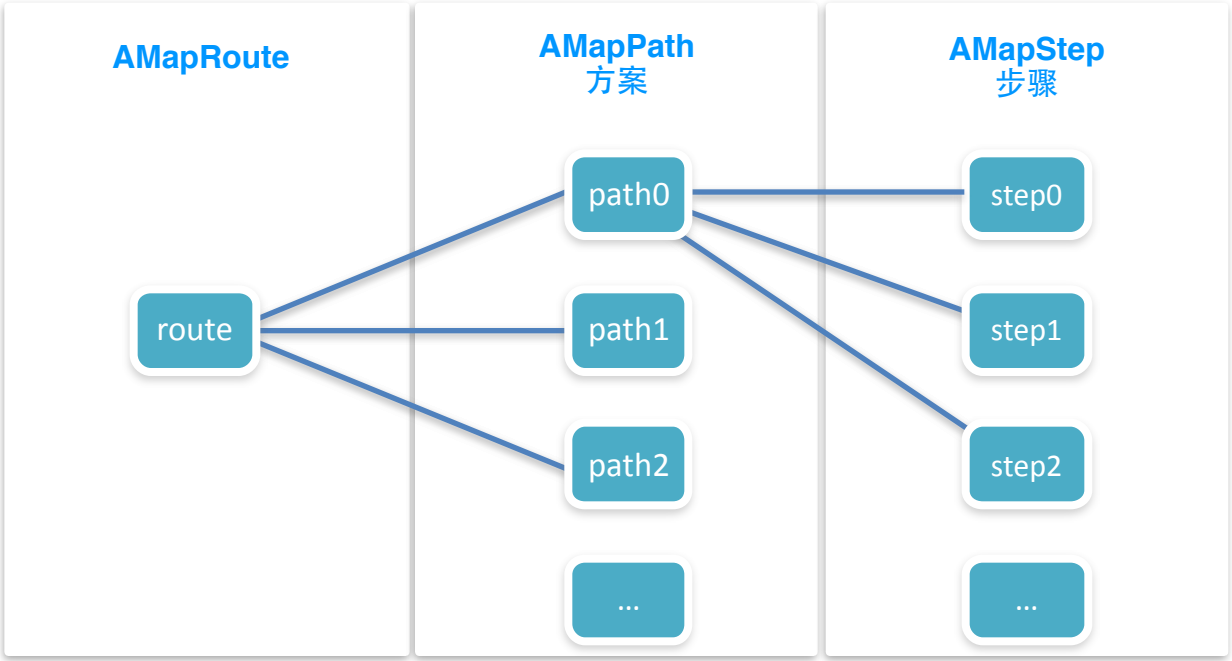
搜索路线

路线搜索类型

AMapSearchType_NaviDrive	驾车
AMapSearchType_NaviWalking	步行
AMapSearchType_NaviBus	公交

返回结果类型AMapRoute

以步行结果举例



搜索路线

添加搜索按钮

```
- (void)initControls
{
…
    UIButton *pathButton = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    pathButton.frame = CGRectMake(140, CGRectGetHeight(_mapView.bounds) - 60, 40, 40);
    pathButton.autoresizingMask = UIViewAutoresizingFlexibleRightMargin |
UIViewAutoresizingFlexibleTopMargin;
    pathButton.backgroundColor = [UIColor whiteColor];
    [pathButton setImage:[UIImage imageNamed:@"path"] forState:UIControlStateNormal];

    [pathButton addTarget:self action:@selector(pathAction)
forControlEvents:UIControlEventTouchUpInside];

    [_mapView addSubview:pathButton];

}
```

搜索按钮响应方法

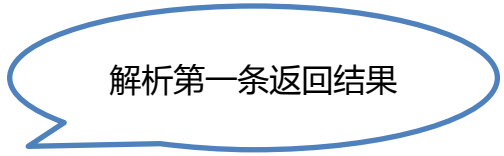
```
- (void)pathAction
{
    if (_destinationPoint == nil || _currentLocation == nil || _search == nil)
    {
        NSLog(@"path search failed");
        return;
    }

    AMapNavigationSearchRequest *request = [[AMapNavigationSearchRequest alloc] init];

    // 设置为步行路径规划
    request.searchType = AMapSearchType_NaviWalking;

    request.origin = [AMapGeoPoint locationWithLatitude:_currentLocation.coordinate.latitude
longitude:_currentLocation.coordinate.longitude];
    request.destination = [AMapGeoPoint locationWithLatitude:_destinationPoint.coordinate.latitude
longitude:_destinationPoint.coordinate.longitude];

    [_search AMapNavigationSearch:request];
}
```

搜索路线

获取搜索结果

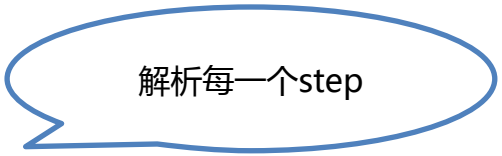
从搜索代理响应方法中获取

```
- (void)onNavigationSearchDone:(AMapNavigationSearchRequest *)request response:(AMapNavigationSearchResponse *)response
{
    if (response.count > 0)
    {
        [_mapView removeOverlays:_pathPolylines];
        _pathPolylines = nil;

        // 只显示第一条
        _pathPolylines = [self polylinesForPath:response.route.paths[0]];
        [_mapView addOverlays:_pathPolylines];

        [_mapView showAnnotations:@[_destinationPoint, _mapView.userLocation] animated:YES];
    }
}
```



路线解析方法

```
- (NSArray *)polylinesForPath:(AMapPath *)path
{
    if (path == nil || path.steps.count == 0)
    {
        return nil;
    }

    NSMutableArray *polylines = [NSMutableArray array];

    [path.steps enumerateObjectsUsingBlock:^(AMapStep *step, NSUInteger idx, BOOL *stop) {

        NSUInteger count = 0;
        CLLocationCoordinate2D *coordinates = [self coordinatesForString:step.polyline
                                                         coordinateCount:&count
                                                              parseToken:@";"];

        MAPolyline *polyline = [MAPolyline polylineWithCoordinates:coordinates count:count];
        [polylines addObject:polyline];

        free(coordinates), coordinates = NULL;
    }];

    return polylines;
}
```



搜索路线

解析经纬度串

```
- (CLLocationCoordinate2D *)coordinatesForString:(NSString *)string
                                 coordinateCount:(NSUInteger *)coordinateCount
                                      parseToken:(NSString *)token
{
    if (string == nil)
    {
        return NULL;
    }

    if (token == nil)
    {
        token = @",";
    }

    NSString *str = @"";
    if (![token isEqualToString:@","])
    {
        str = [string stringByReplacingOccurrencesOfString:token withString:@","];
    }

    else
    {
        str = [NSString stringWithString:string];
    }

    NSArray *components = [str componentsSeparatedByString:@","];
    NSUInteger count = [components count] / 2;
    if (coordinateCount != NULL)
    {
        *coordinateCount = count;
    }
    CLLocationCoordinate2D *coordinates = (CLLocationCoordinate2D*)malloc(count * sizeof(CLLocationCoordinate2D));

    for (int i = 0; i < count; i++)
    {
        coordinates[i].longitude = [[components objectAtIndex:2 * i]     doubleValue];
        coordinates[i].latitude  = [[components objectAtIndex:2 * i + 1] doubleValue];
    }

    return coordinates;
}
```

搜索路线

其中我们用到了_pathPolylines记录路线数据

_pathPolylines声明

```
@interface ViewController ()<MAMapViewDelegate, AMapSearchDelegate, UITableViewDataSource, UITableViewDelegate,
UIGestureRecognizerDelegate>
{
…
    NSArray *_pathPolylines;
}
```

_pathPolylines清空

```
- (void)handleLongPress:(UILongPressGestureRecognizer *)gesture
{
…
        // 添加标注
        if (_destinationPoint != nil)
        {
            // 清理
            [_mapView removeAnnotation:_destinationPoint];
            _destinationPoint = nil;

            [_mapView removeOverlays:_pathPolylines];
            _pathPolylines = nil;
        }
…
}
```

绘制路线

使用MAPolyline记录并绘制路线

MAPolyline

在地图上绘制线的数据模型，是地图覆盖物（MAOverlay）的一种

MAPolylineView

在地图上绘制线的样式，是MAOverlayView的一种

使用步骤

1 添加polyline

```
/*!
 @brief 向地图窗口添加Overlay，需要实现MAMapViewDelegate的-mapView:viewForOverlay:函数来生成标注对应的View
 @param overlay 要添加的overlay
 */
- (void)addOverlay:(id <MAOverlay>)overlay;

/*!
 @brief 向地图窗口添加一组Overlay，需要实现BMKMapViewDelegate的-mapView:viewForOverlay:函数来生成标注对应的View
 @param overlays 要添加的overlay数组
 */
- (void)addOverlays:(NSArray *)overlays;
```

2 实现MAMapView回调

```
- (MAOverlayView *)mapView:(MAMapView *)mapView viewForOverlay:(id<MAOverlay>)overlay
```

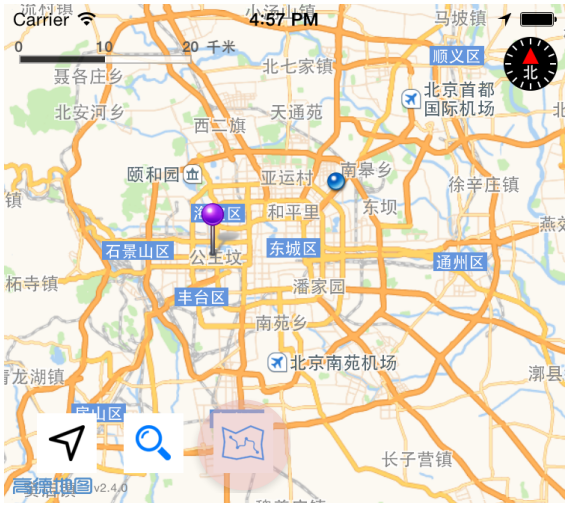
目前已经在获取并解析搜索结果后，添加polyline到地图
接下来实现MAMapView的回调方法

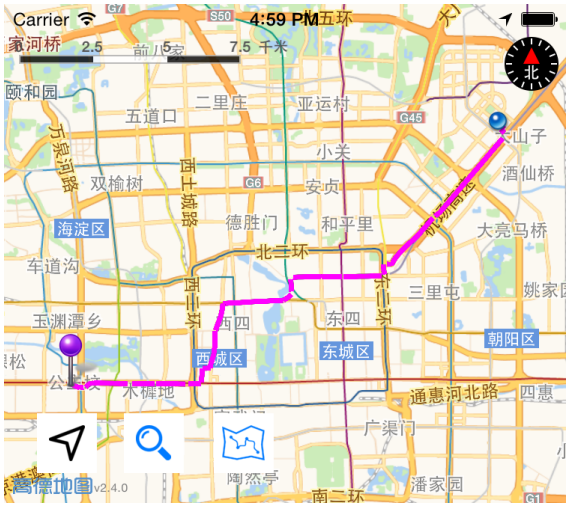
绘制路线

实现MAMapView回调方法

```
- (MAOverlayView *)mapView:(MAMapView *)mapView viewForOverlay:(id<MAOverlay>)overlay
{
    if ([overlay isKindOfClass:[MAPolyline class]])
    {
        MAPolylineView *polylineView = [[MAPolylineView alloc]
initWithPolyline:overlay];

        polylineView.lineWidth   = 4;
        polylineView.strokeColor = [UIColor magentaColor];

        return polylineView;
    }

    return nil;
}
```

搜索和绘制路线


Carrier
4:57 PM
0
10
20 千米
北
顺义区
北京首都
国际机场
北七家镇
聂各庄乡
北安河乡
天通苑
西二旗
颐和园
亚运村
南皋乡
徐辛庄镇
和平里
东坝
海淀区
东城区
石景山区
公主坟
通州区
丰台区
潘家园
南苑乡
北京南苑机场
长子营镇
高德地图
v2.4.0


Carrier
4:59 PM
2.5
5
7.5 千米
北
颐和园
五道口
二里庄
亚运村
小关
双榆树
安贞
海淀区
德胜门
和平里
北三环
车道沟
三里屯
玉渊潭乡
东四
朝阳区
西城区
东城区
公主坟
木樨地
通惠河北路
广渠门
潘家园
南二环
高德地图
v2.4.0

作业

添加功能

测量polyline的总长度（单位：米）

more...

请访问 http://lbs.amap.com/

Thanks!

高德LBS开发者论坛 http://lbsbbs.amap.com/

本课程源码下载 https://github.com/hadesh/HelloAmap/tree/lesson_04