

CS 696 Mobile Phone Application Development  
Fall Semester, 2010  
Doc 18 MapKit  
Nov 2, 2010

Copyright ©, All rights reserved. 2010 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

## References

Location Awareness Programming Guide, Apple Documentation, <http://developer.apple.com/library/ios/#documentation/UserExperience/Conceptual/LocationAwarenessPG/Introduction/Introduction.html>

More iPhone 3 Development: Tacking iPhone SDK 3, Mark, LaMarche, Apress, 2009, Chapter 11

# Maps

## Classes

MKMapView

MKMapViewDelegate

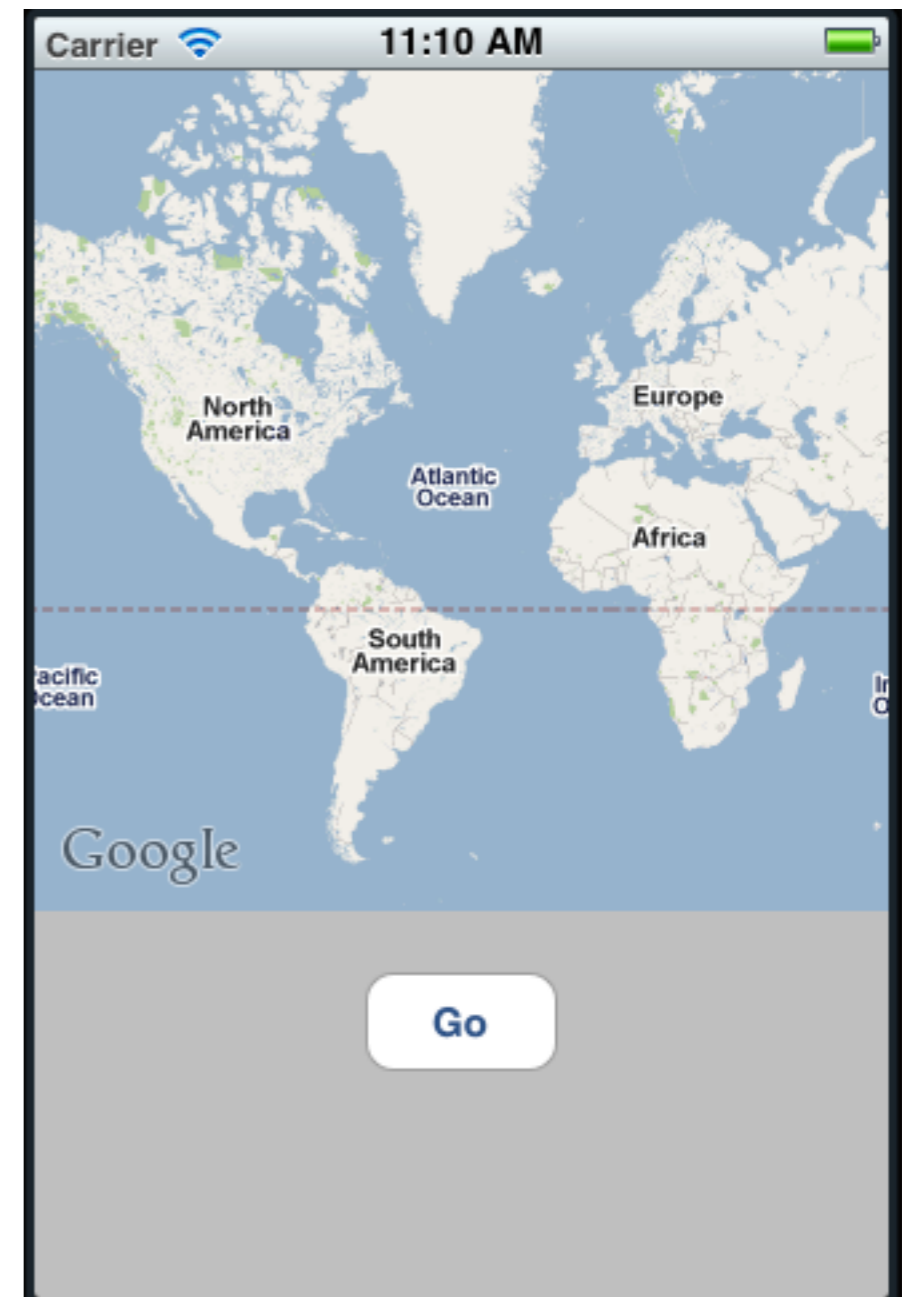
MKReverseGeocoderDelegate

CLLocationManagerDelegate

## Frameworks

CoreLocation

MapKit



# Google Maps

Mapkit uses Google maps

License Agreement

<http://code.google.com/apis/maps/iphone/terms.html>

# Location & Geocoding

Latitude & Longitude

<+32.76572793, -117.07319879>

Reverse  
Geocoding



Address

5500 Campanile Drive  
San Diego, CA 92182

Geocoding



CoreLocation only supports reverse geocoding

# CoreLocation

CLLocation  
CLLocationManager  
CLHeading

Uses

GPS

Wifi

Cell Network

More accurate the more power

kCLLocationAccuracyBest;

kCLLocationAccuracyNearestTenMeters;

kCLLocationAccuracyHundredMeters;

kCLLocationAccuracyKilometer;

kCLLocationAccuracyThreeKilometers;

location: <+32.76572793, -117.07319879> +/- 100.00m  
**CLLocation**  
(speed 1.00 mps / course -1.00)  
@ 2010-11-02 01:56:29 GMT

Point in real world plus direction and speed

@property CLLocationCoordinate2D coordinate;  
@property CLLocationDistance altitude;  
@property CLLocationAccuracy horizontalAccuracy;  
@property CLLocationAccuracy verticalAccuracy;  
@property CLLocationDirection course;  
@property CLLocationSpeed speed;  
- (NSDate \*)timeStamp;  
- (CLLocationDistance)distanceFromLocation:(CLLocation \*)location

# CLLocationManager

Starting point for Location

```
@property CLLocation *location;  
@property id <CLLocationManagerDelegate> delegate;  
@property CLLocationDistance distanceFilter;  
@property CLLocationAccuracy verticalAccuracy;
```

- (void)startUpdatingLocation
- (void)stopUpdatingLocation
- (void)startUpdatingHeading
- (void)stopUpdatingHeading
- (void)startMonitoringForRegion:(CLRegion \*)region  
    desiredAccuracy:(CLLocationAccuracy)accuracy



# CLLocationManagerDelegate

## Callbacks for Location Change

- locationManager:didUpdateToLocation:fromLocation:
- locationManager:didFailWithError:

## Callbacks for Heading Change

- locationManager:didUpdateHeading:
- locationManagerShouldDisplayHeadingCalibration:

## Callbacks for Region Events

- locationManager:didEnterRegion:
- locationManager:didExitRegion:
- locationManager:monitoringDidFailForRegion:withError:

# MapKit

Displays Google Maps  
Reverse Geocoding  
Pins on maps (annotations)

MKMapView  
MKMapViewDelegate  
MKAnnotation  
MKPlacemark  
MKUserLocation  
MKReverseGeocoder

# MKMapView

Displays map

Map & Satellite

Panning & zooming

Annotations

User location

# MKMapViewDelegate

## Responding to Map Position Changes

- mapView:regionWillChangeAnimated:
- mapView:regionDidChangeAnimated:

## Loading the Map Data

- mapViewWillStartLoadingMap:
- mapViewDidFinishLoadingMap:
- mapViewDidFailLoadingMap:withError:

## Tracking the User Location

- mapViewWillStartLocatingUser:
- mapViewDidStopLocatingUser:
- mapView:didUpdateUserLocation: **Text**
- mapView:didFailToLocateUserWithError:

## Managing Annotation Views

- mapView:viewForAnnotation:
- mapView:didAddAnnotationViews:
- mapView:annotationView:calloutAccessoryControlTapped:

## Dragging an Annotation View

- mapView:annotationView:didChangeDragState:fromOldState:

## Selecting Annotation Views

- mapView:didSelectAnnotationView:
- mapView:didDeselectAnnotationView:

## Managing Overlay Views

- mapView:viewForOverlay:
- mapView:didAddOverlayViews:

# MKAnnotation

Add to map view to show pins

Only a protocol

Need to create class that implements protocol

```
@property CLLocationCoordinate2D coordinate;
```

```
@property NSString *title;
```

```
@property NSString *subtitle;
```

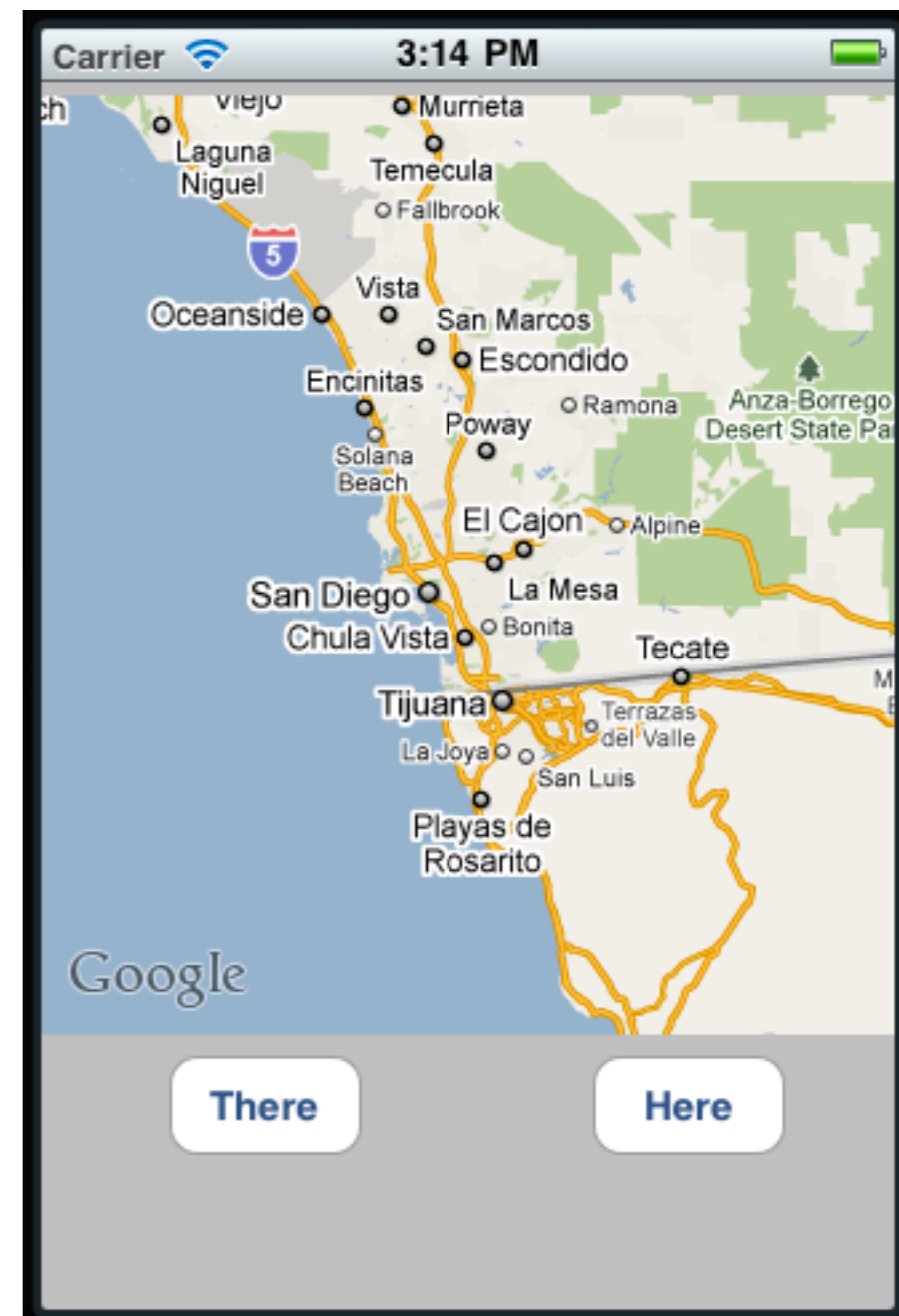
# MKReverseGeocoder

Reverse geocodes latitude/longitude to address

Needs delegate to get result

# Example

Need to add Frameworks:  
MapKit  
CoreLocation



# MapViewController

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

@interface MapViewController : UIViewController <CLLocationManagerDelegate,
MKReverseGeocoderDelegate, MKMapViewDelegate>{

}
@property (nonatomic, retain) IBOutlet MKMapView *mapView;

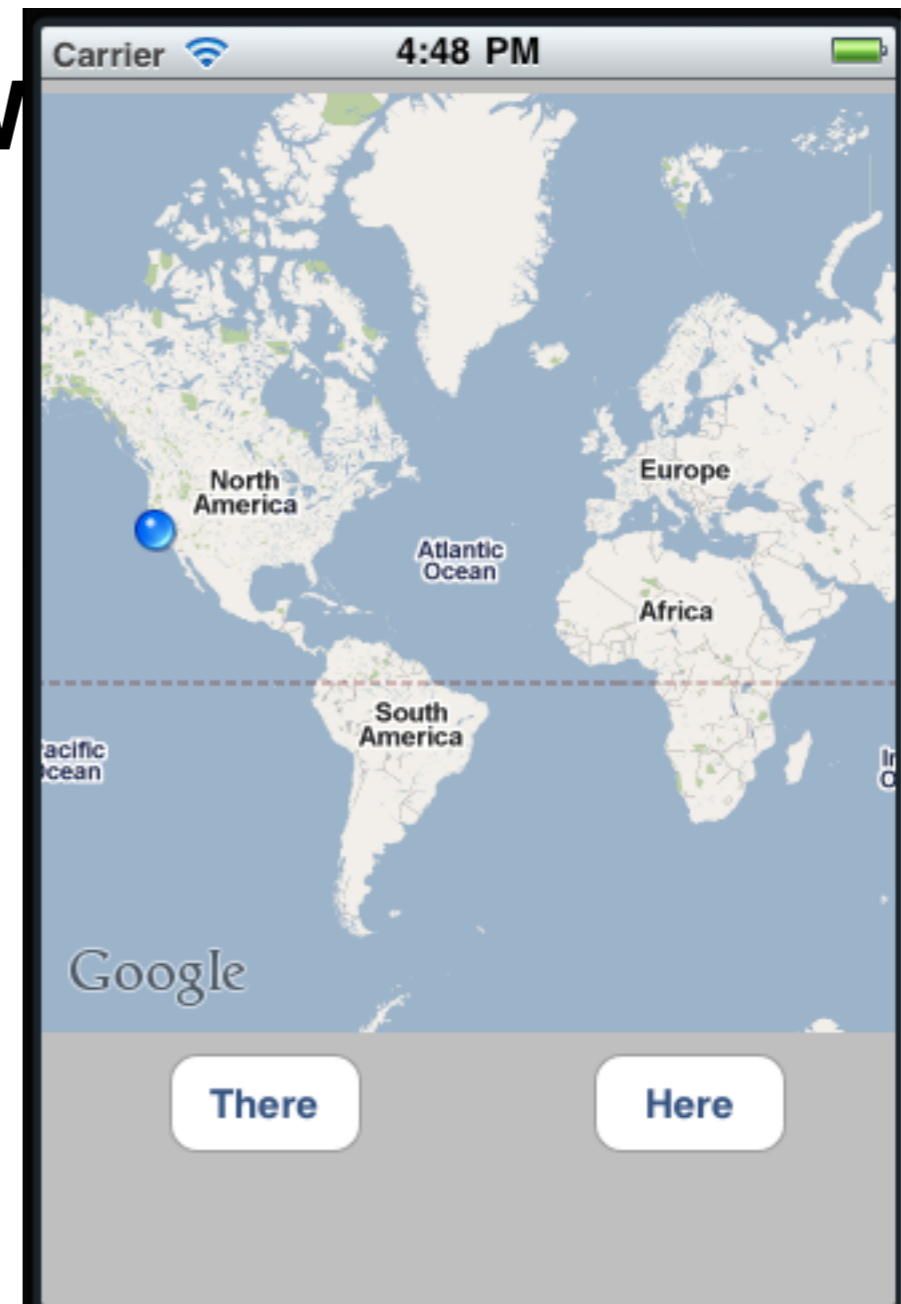
- (IBAction)findMe;
- (IBAction)showThere;

@end
```

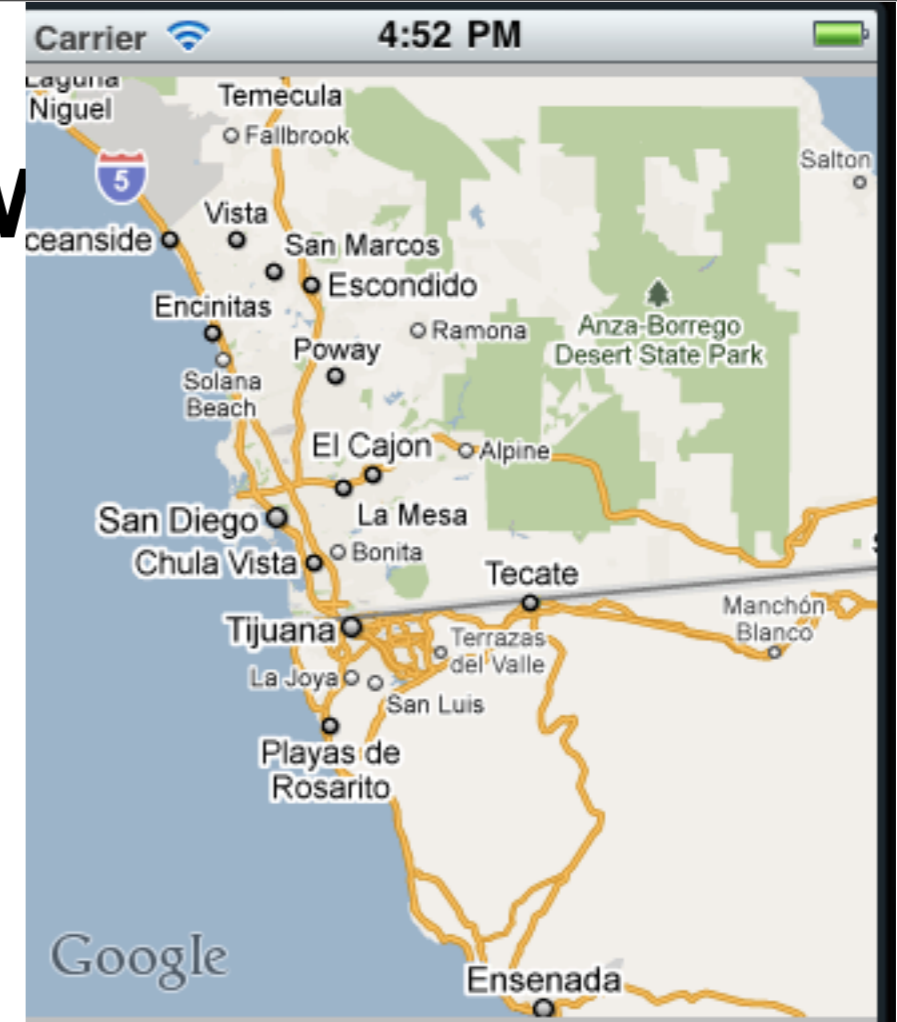


# Showing the World

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
  
    mapView.mapType = MKMapTypeStandard;  
    mapView.showsUserLocation = YES;  
}
```



# Restricting the Map



```
- (void)viewDidLoad {  
    [super viewDidLoad];
```

```
    CLLocationCoordinate2D sanDiego = {latitude: 32.76572795, longitude: -117.07319880};  
    MKCoordinateSpan span = {latitudeDelta: 0.9, longitudeDelta: 0.5};  
    MKCoordinateRegion county = {sanDiego, span};  
    mapView.region = county;  
    mapView.mapType = MKMapTypeStandard;  
    mapView.showsUserLocation = YES;  
}
```

# Adding an annotation

```
- (IBAction)showThere {  
    AnnotatedLocation *annotation = [[AnnotatedLocation alloc] init];  
    CLLocationCoordinate2D there = {latitude: 32.955, longitude: -117.2459};  
    annotation.coordinate = there;  
    [annotation setTitle: @"There"];  
    annotation.subtitle = @"You are not here";  
    [mapView addAnnotation:annotation];  
    [annotation release];  
}
```

# AnnotatedLocation

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

@interface AnnotatedLocation : NSObject <MKAnnotation>{
    NSString * title;
}
@property (nonatomic, readwrite) CLLocationCoordinate2D coordinate;
@property (nonatomic, readwrite, retain) NSString *subtitle;

- (void)setTitle: (NSString *) aTitle;

@property (nonatomic, copy) NSString *streetAddress;
@property (nonatomic, copy) NSString *city;
@property (nonatomic, copy) NSString *state;
@property (nonatomic, copy) NSString *zip;

@end
```

# AnnotatedLocation

```
@implementation AnnotatedLocation
```

```
@synthesize coordinate;
```

```
@synthesize subtitle;
```

```
@synthesize streetAddress;
```

```
@synthesize city;
```

```
@synthesize state;
```

```
@synthesize zip;
```

```
- (NSString *)title { return title; }
```

```
- (void)setTitle: (NSString *) aTitle {  
    title = [aTitle copy];  
}
```

```
@end
```

# Using Location Manager

```
- (IBAction)findMe {  
    NSLog(@"FindMe");  
    CLLocationManager *finder = [[CLLocationManager alloc] init];  
    finder.delegate = self;  
    finder.desiredAccuracy = kCLLocationAccuracyBest;  
    [finder startUpdatingLocation];  
}
```

# Location Manager Delegate Methods

```
- (void)locationManager:(CLLocationManager *)manager
    didFailWithError:(NSError *)error {

    NSString *errorType = (error.code == kCLErrorDenied) ? @"Access Denied" :
        @"Unknown Error";

    UIAlertView *alert = [[UIAlertView alloc]
        initWithTitle:@"Error getting Location"
        message:errorType
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"Okay", @"Okay")
        otherButtonTitles:nil];

    [alert show];
    [alert release];
    [manager release];
}
```

# Location Manager Delegate Methods

```
- (void)locationManager:(CLLocationManager *)manager
    didUpdateToLocation:(CLLocation *)newLocation
    fromLocation:(CLLocation *)oldLocation {

    if ([newLocation.timestamp TimeIntervalSince1970] < [NSDate TimeIntervalSinceReferenceDate] - 60)
        return;

    MKCoordinateRegion viewRegion = MKCoordinateRegionMakeWithDistance(newLocation.coordinate,
                                                                    2000, 2000);
    MKCoordinateRegion adjustedRegion = [mapView regionThatFits:viewRegion];
    [mapView setRegion:adjustedRegion animated:YES];

    manager.delegate = nil;
    [manager stopUpdatingLocation];
    [manager autorelease];

    MKReverseGeocoder *geocoder = [[MKReverseGeocoder alloc]
                                    initWithCoordinate:newLocation.coordinate];

    geocoder.delegate = self;
    [geocoder start];
}
```



# Geocoder

```
- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder
    didFailWithError:(NSError *)error {
    UIAlertView *alert = [[UIAlertView alloc]
        initWithTitle:@"Error translating coordinates into location"
        message: @"Geocoder did not recognize coordinates"
        delegate:self
        cancelButtonTitle:NSLocalizedString(@"Okay", @"Okay")
        otherButtonTitles:nil];

    [alert show];
    [alert release];

    geocoder.delegate = nil;
    [geocoder autorelease];
}
```

# Geocoder

```
- (void)reverseGeocoder:(MKReverseGeocoder *)geocoder  
    didFindPlacemark:(MKPlacemark *)placemark {
```

```
    AnnotatedLocation *annotation = [[AnnotatedLocation alloc] init];
```

```
    annotation.streetAddress = placemark.thoroughfare;
```

```
    annotation.city = placemark.locality;
```

```
    annotation.state = placemark.administrativeArea;
```

```
    annotation.zip = placemark.postalCode;
```

```
    annotation.coordinate = geocoder.coordinate;
```

```
    [annotation setTitle: @"Here"];
```

```
    annotation.subtitle = @"You are here";
```

```
[mapView addAnnotation:annotation];
```

```
    [annotation release];
```

```
}
```