What's New in Map Kit

Session 304

James Howard
Software Engineer Maps Team

Overview

Overview

- Map Kit on OS X

Overview

- Map Kit on OS X
- iOS 7 UI Updates

Overview

- Map Kit on OS X
- iOS 7 UI Updates
- New Perspectives

Overview

- Map Kit on OS X
- iOS 7 UI Updates
- New Perspectives
- New Overlays Functionality

Overview

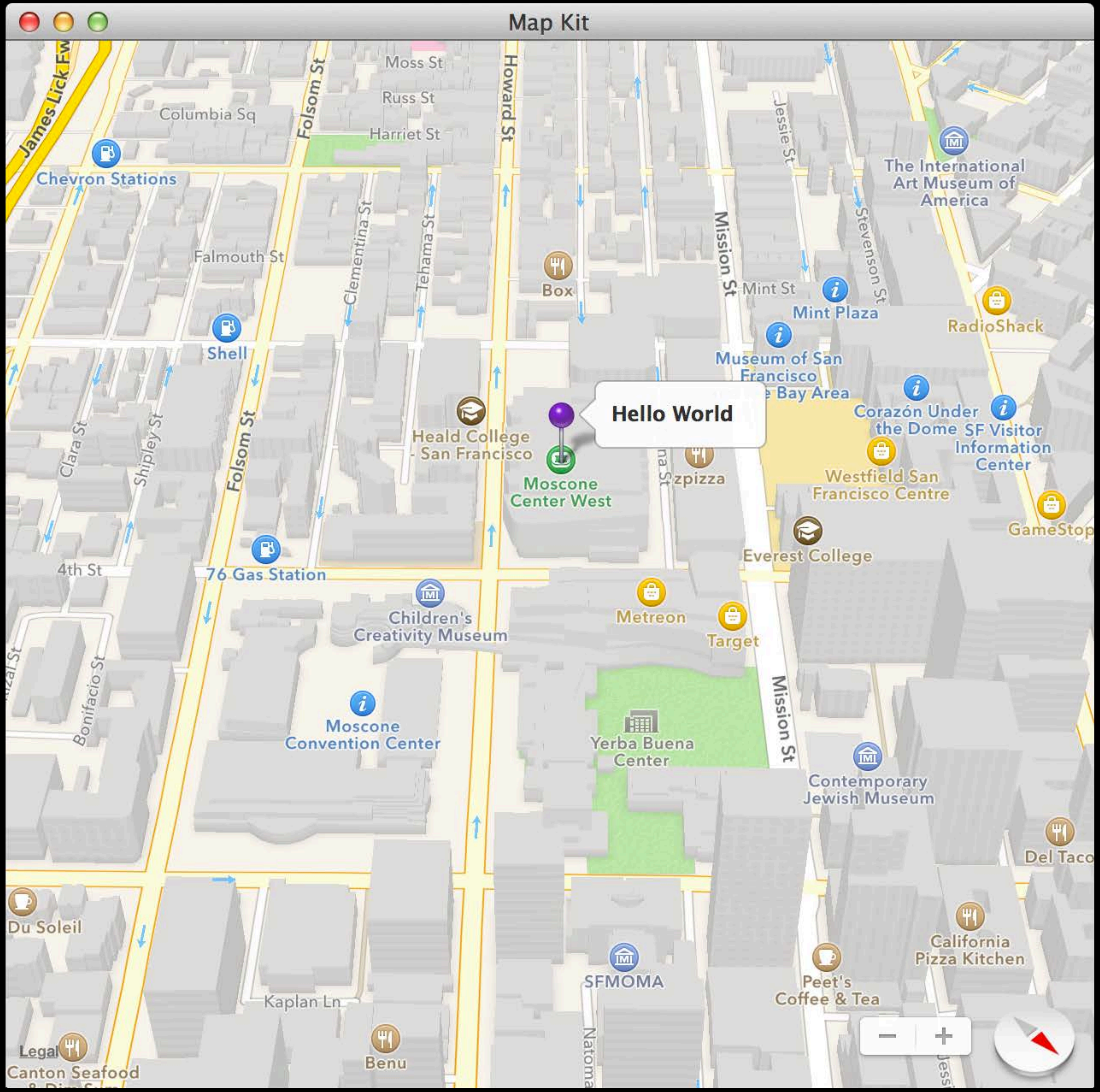
- Map Kit on OS X
- iOS 7 UI Updates
- New Perspectives
- New Overlays Functionality
- Search API

Overview

- Map Kit on OS X
- iOS 7 UI Updates
- New Perspectives
- New Overlays Functionality
- Search API
- Directions API

Map Kit for OS X

Map Kit on OS X



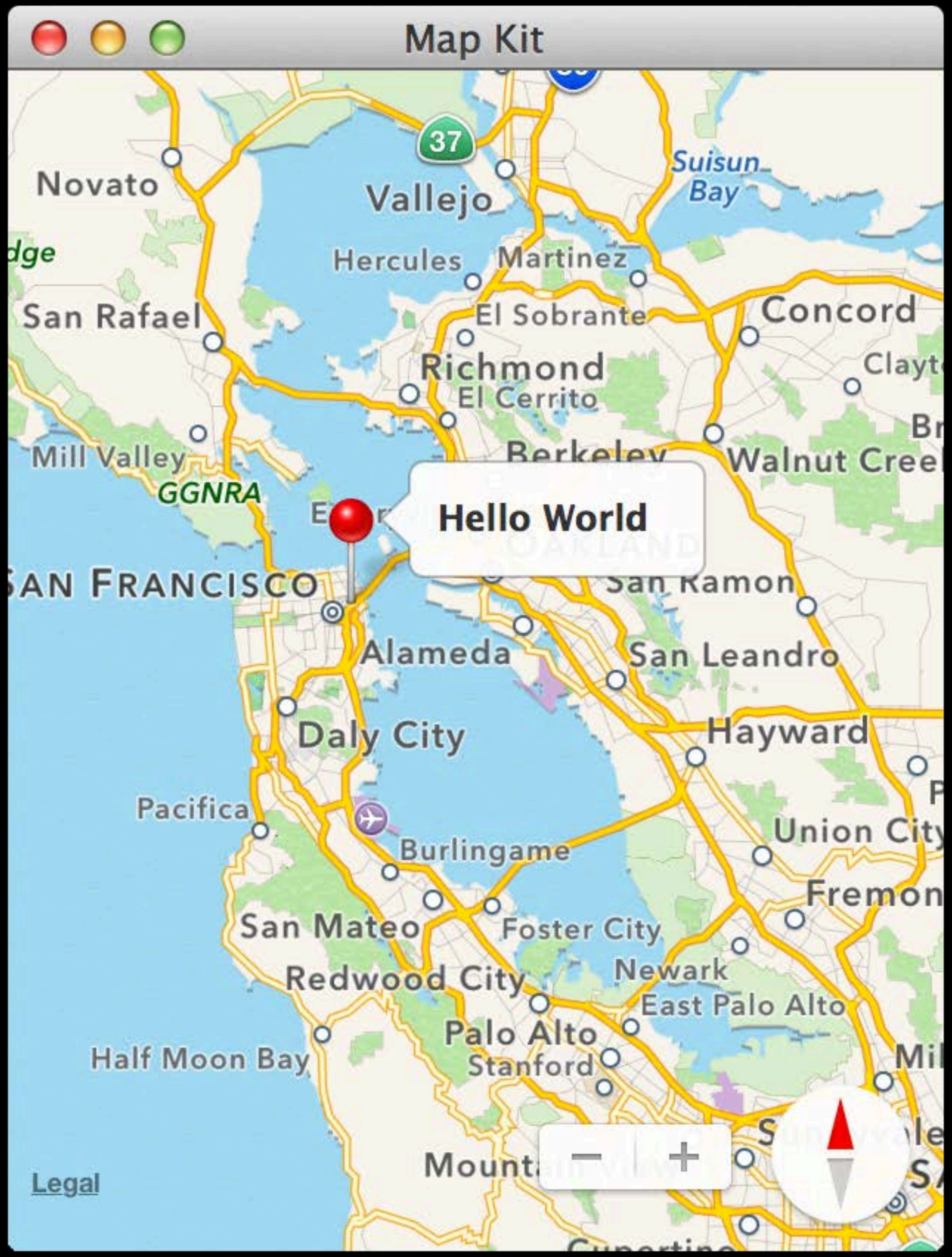
Map Kit on OS X

Just like iOS

- MKMapView
  - Nearly identical API
    - Differences are primarily AppKit types vs UIKit types
  - Annotations
  - Overlays
- Local Search
- Directions

Map Kit on OS X

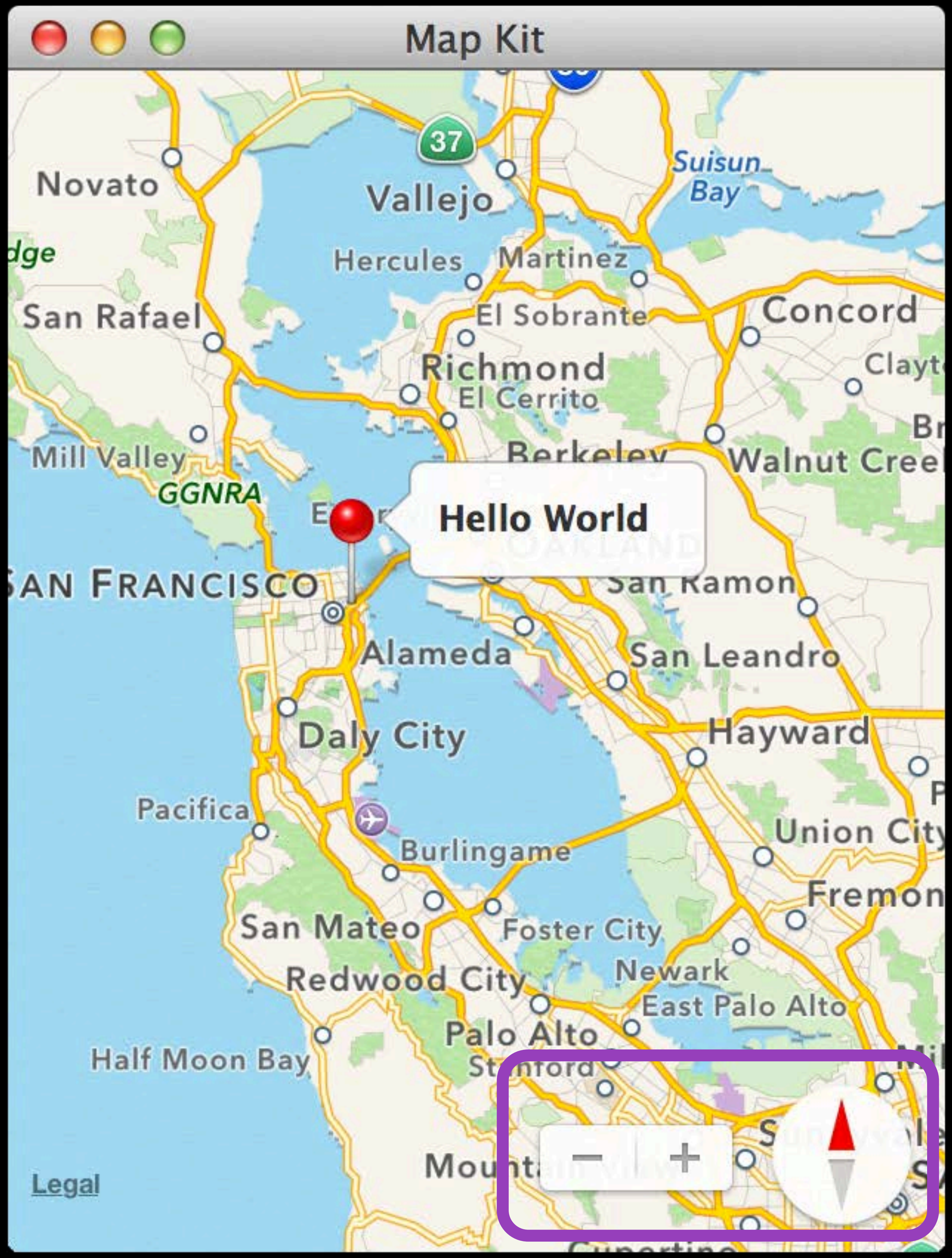
UI Differences



Map Kit on OS X

UI Differences

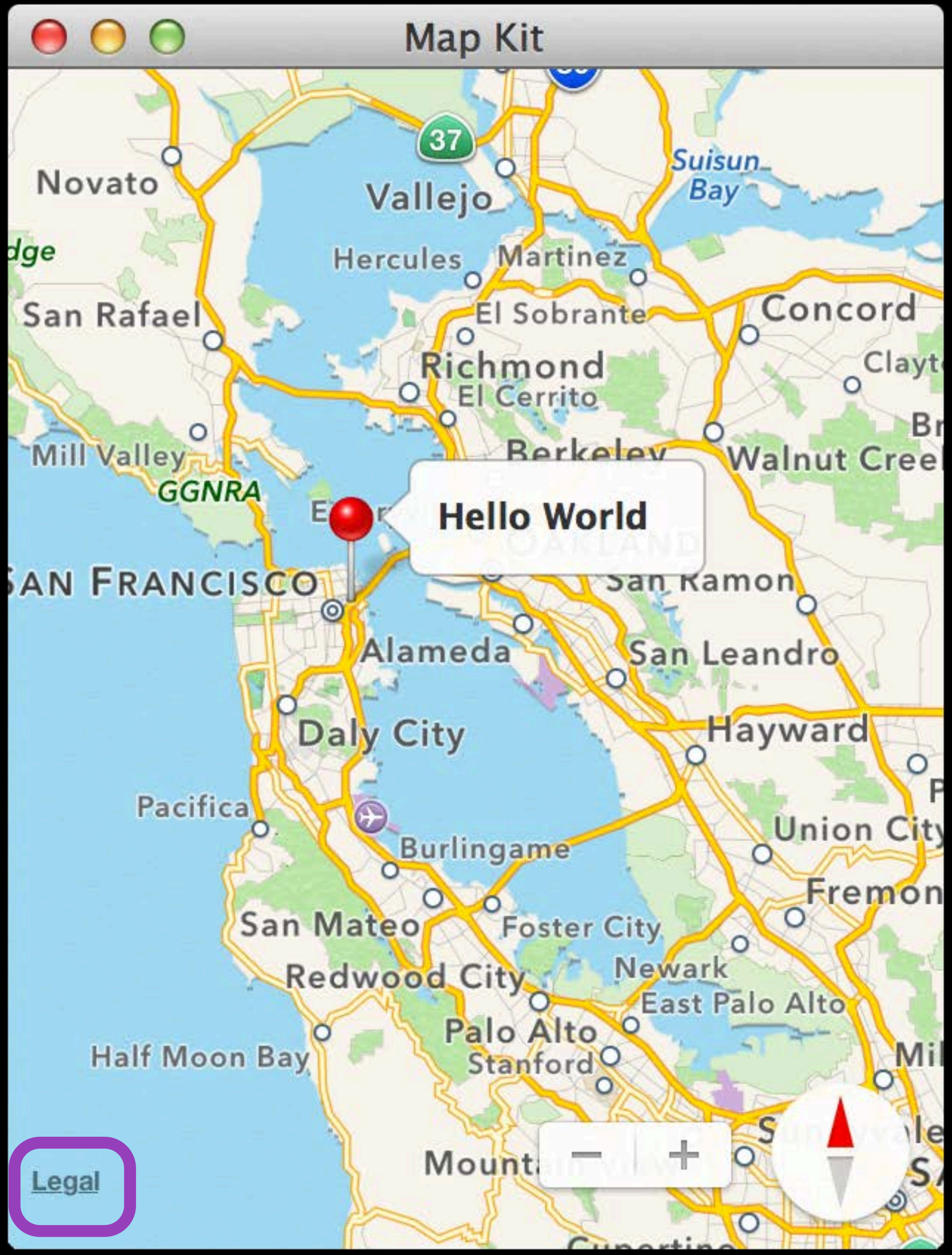
- Zoom / Compass controls
  - Can be shown or hidden via API



Map Kit on OS X

UI Differences

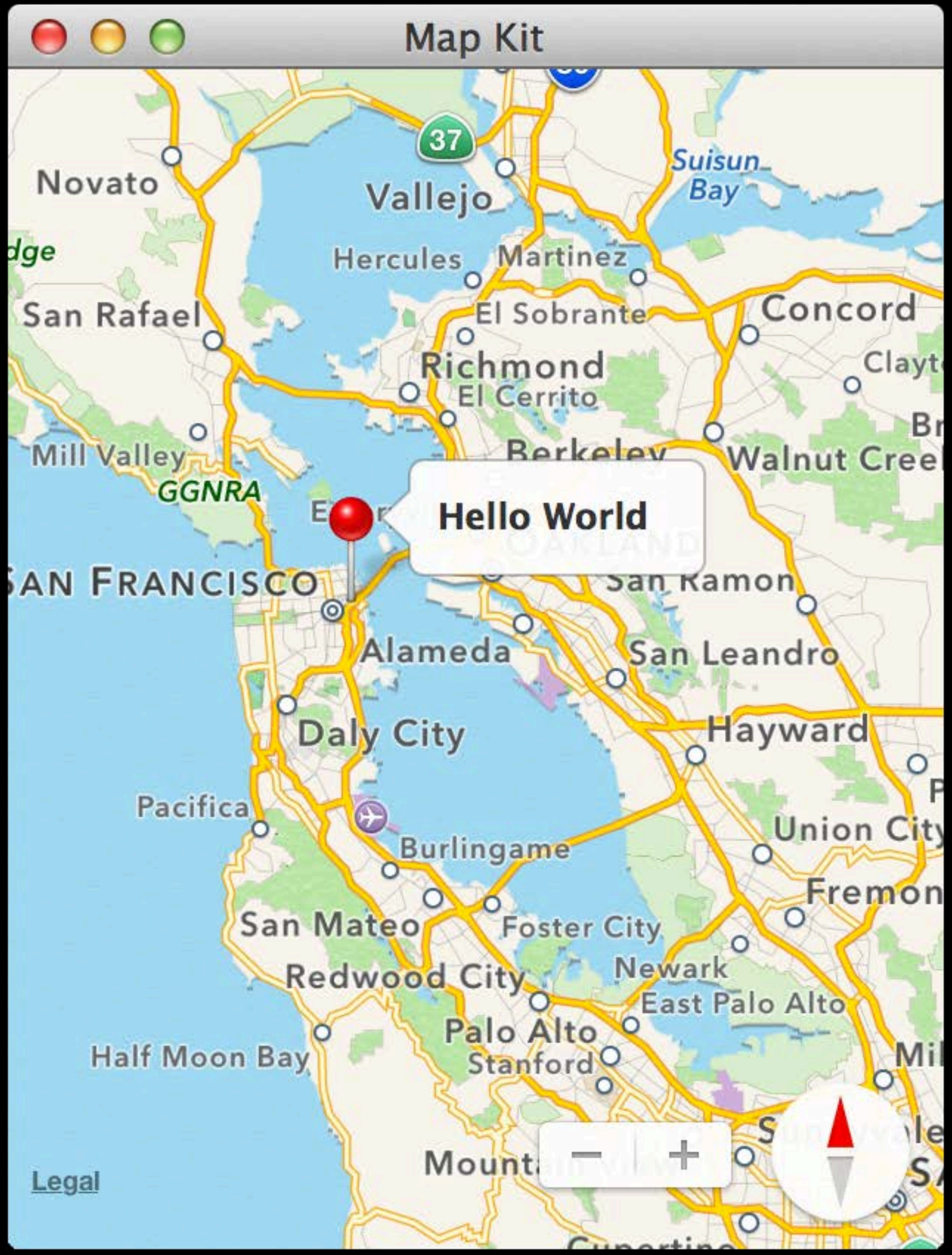
- Zoom / Compass controls
  - Can be shown or hidden via API
- Map data attribution label in lower left
  - Cannot be hidden, don't obscure it



Map Kit on OS X

UI Differences

- Zoom / Compass controls
  - Can be shown or hidden via API
- Map data attribution label in lower left
  - Cannot be hidden, don't obscure it
- Annotation view callout on side of annotation instead of top



Map Kit on OS X

Codesigning requirements

- Available for apps in the Mac App Store
- Uses `com.apple.developer.maps` entitlement
- Enabled via the Xcode 5 capabilities setting

Demo

Map Kit on OS X

Alexandre Carlhian
Software Engineer Maps Team

Map Kit on OS X

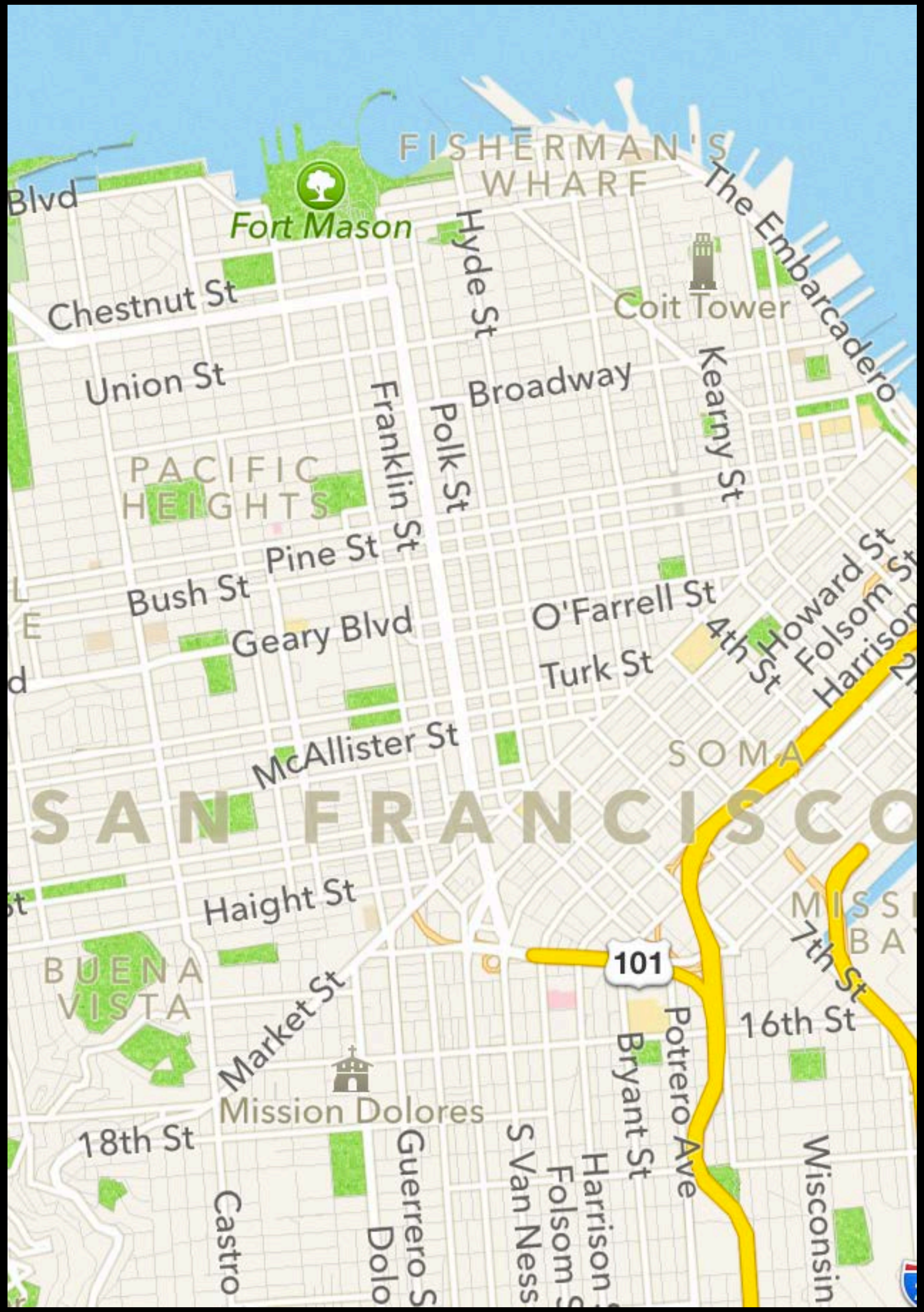
Lessons learned

- Adding Map Kit to your project
- Controlling user interaction with the map view
- Showing annotations on your map
- Creating custom annotation views
- Creating custom callout views

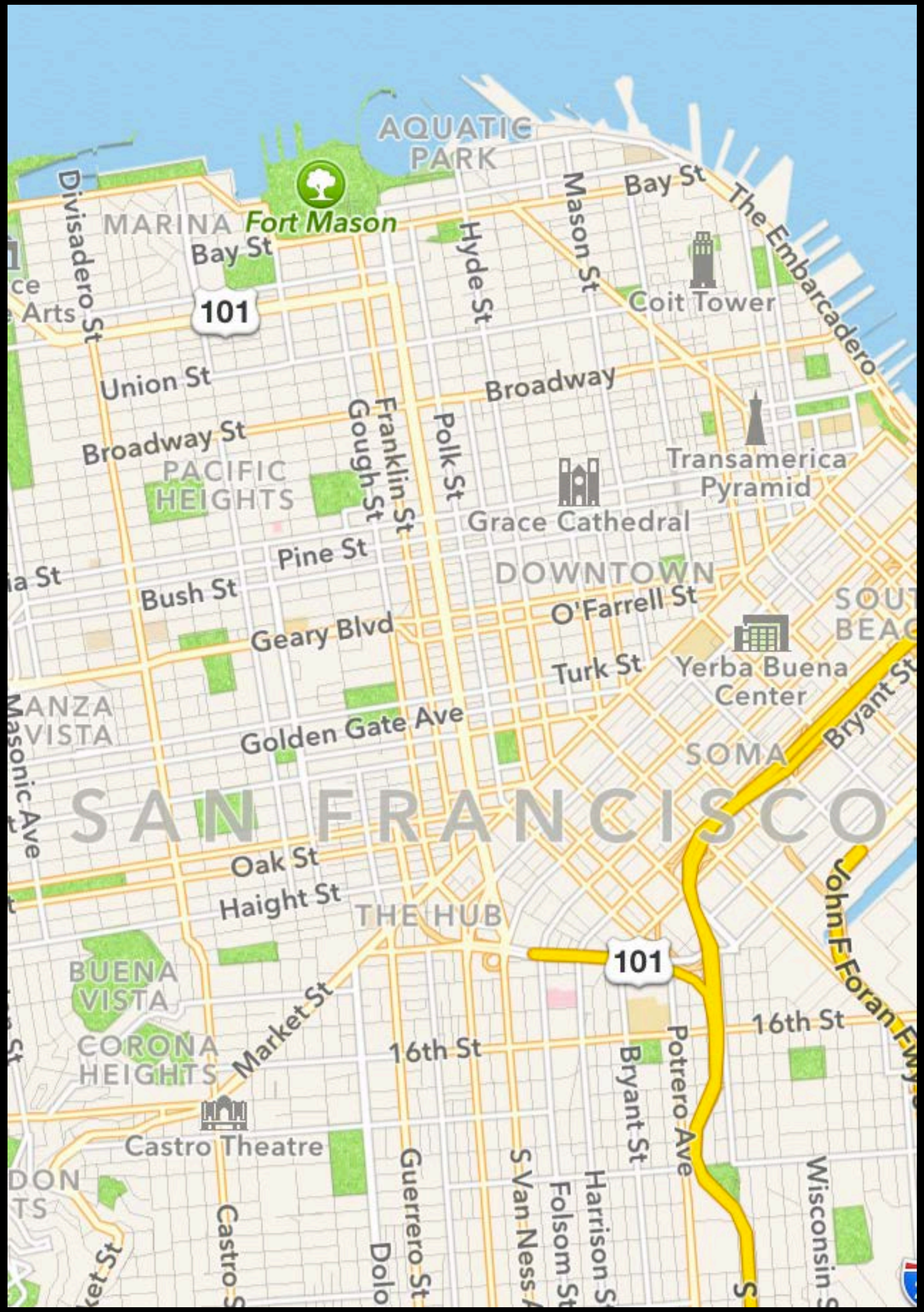
New UI in iOS 7

New UI

Updated cartography



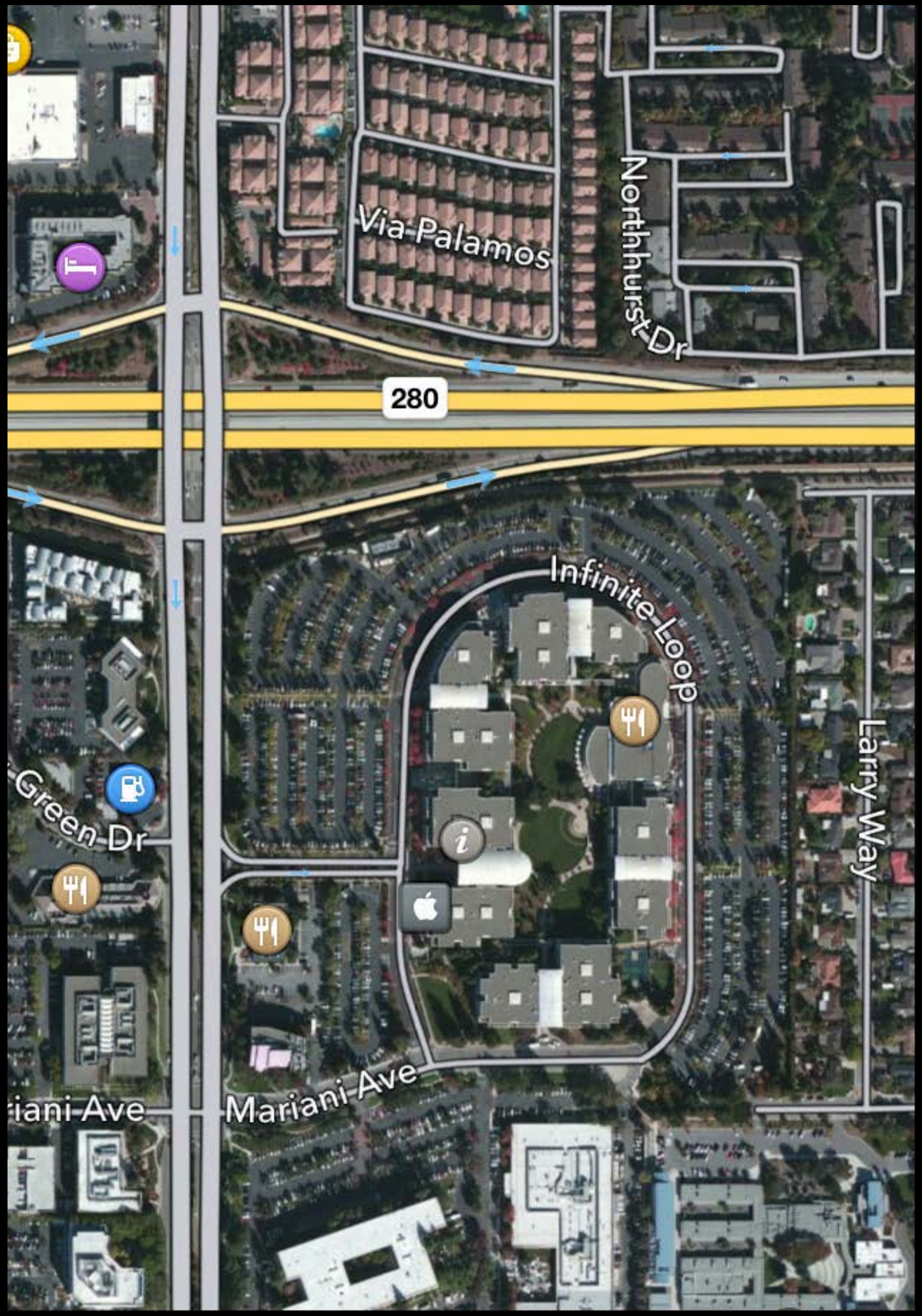
iOS 6



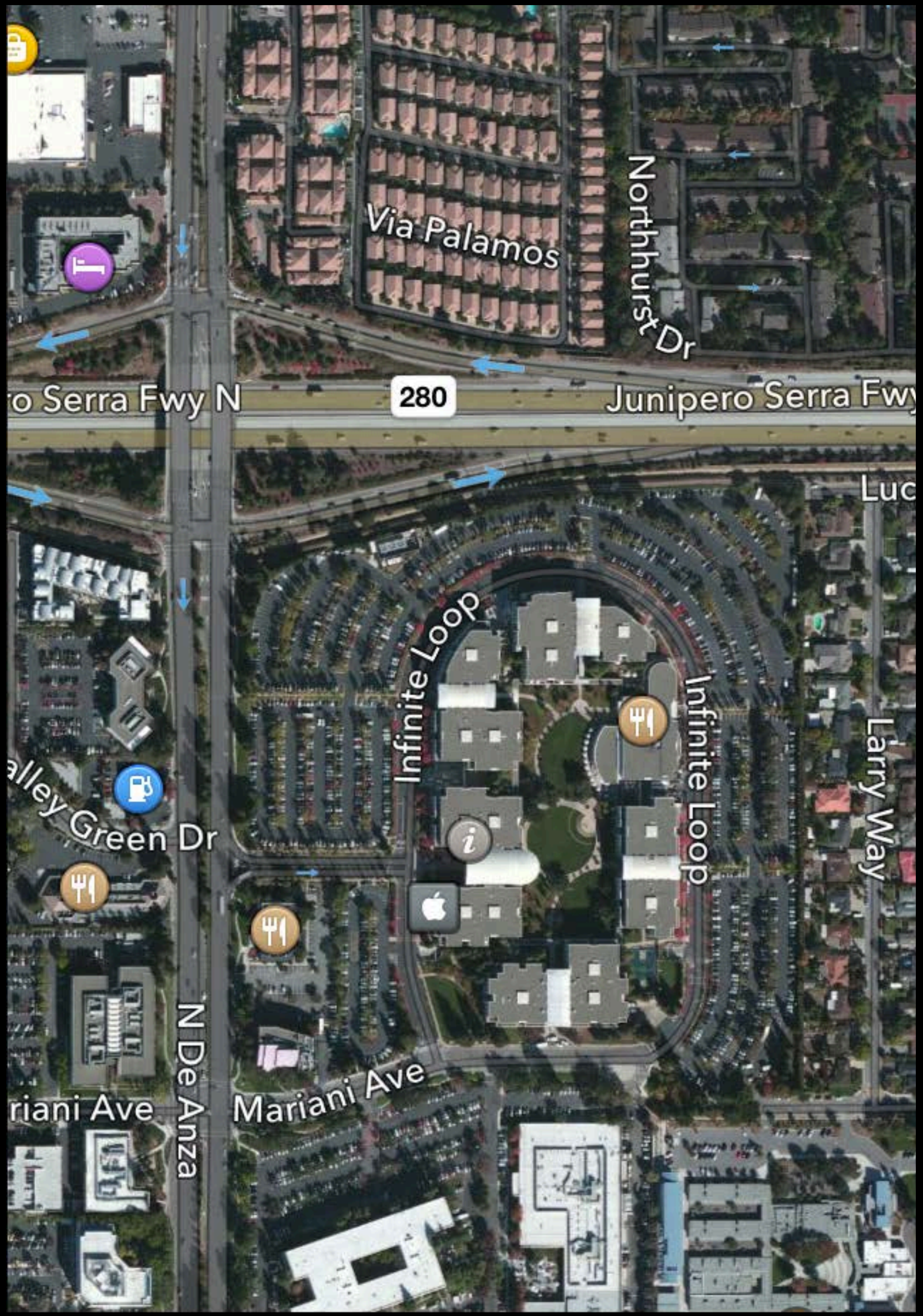
iOS 7

New UI

Updated MKMapTypeHybrid



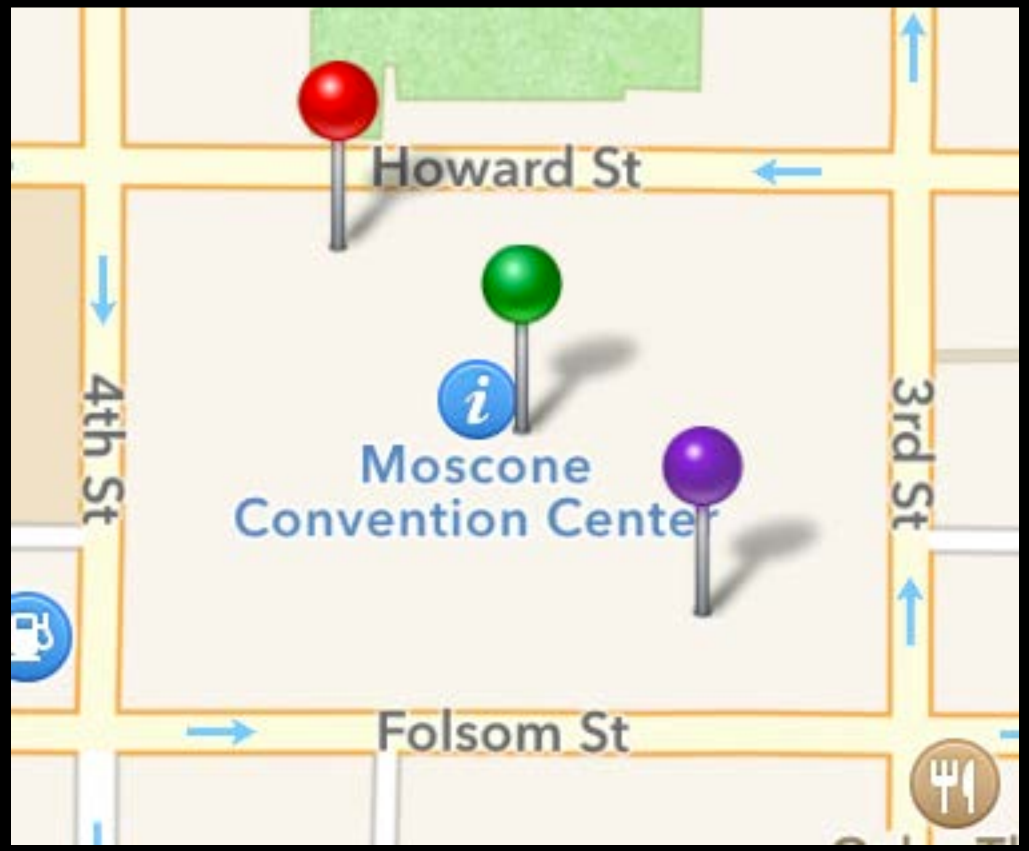
iOS 6



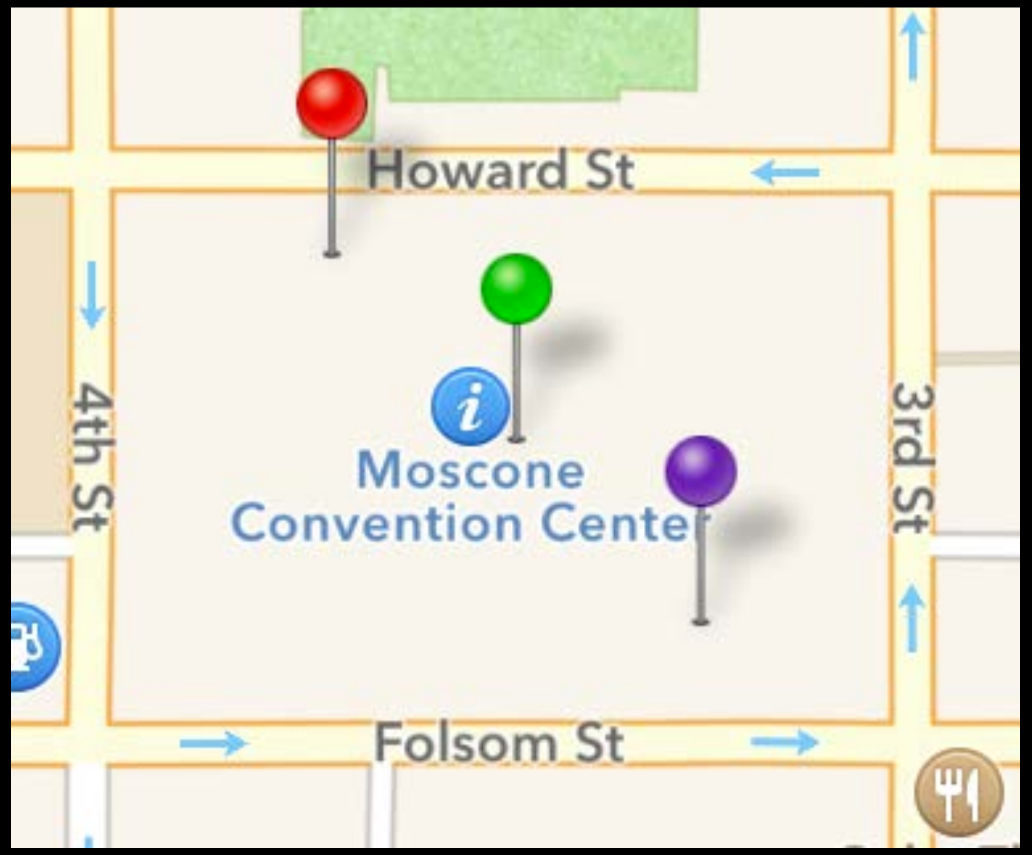
iOS 7

New UI

MKPinAnnotationView



iOS 6

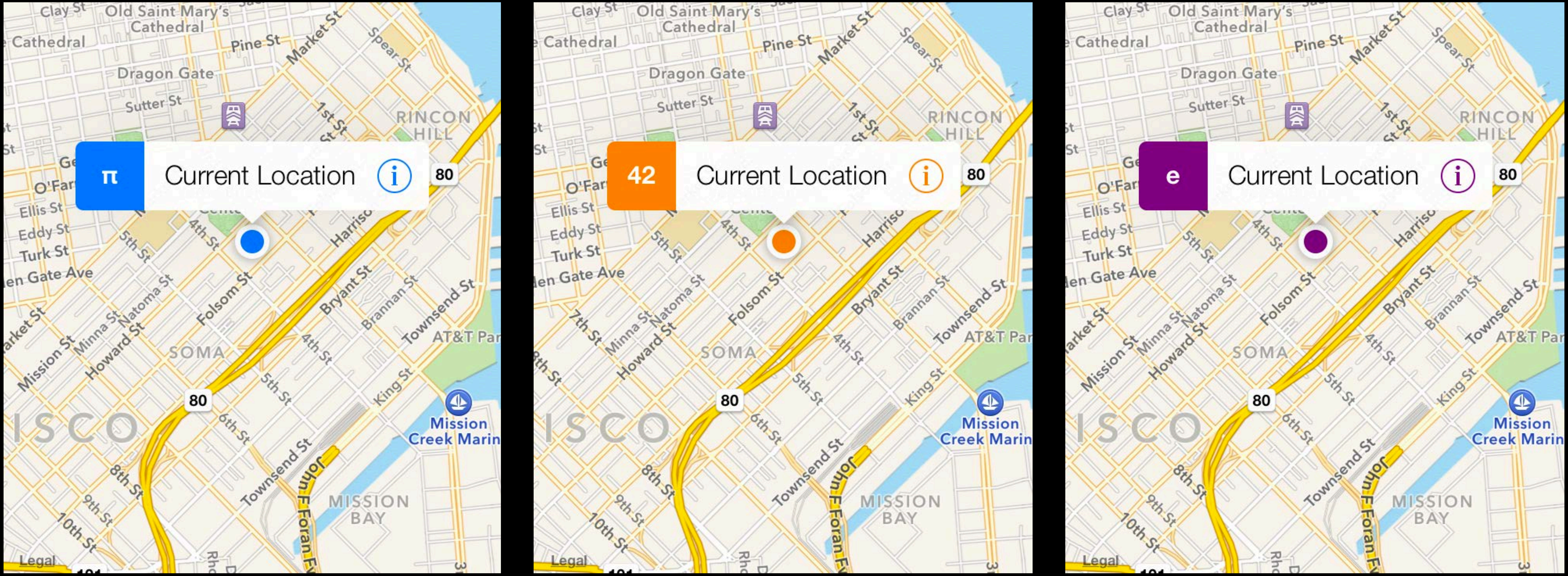


iOS 7

Same dimensions (32, 39)

New UI

New callout, user location view, and view tint

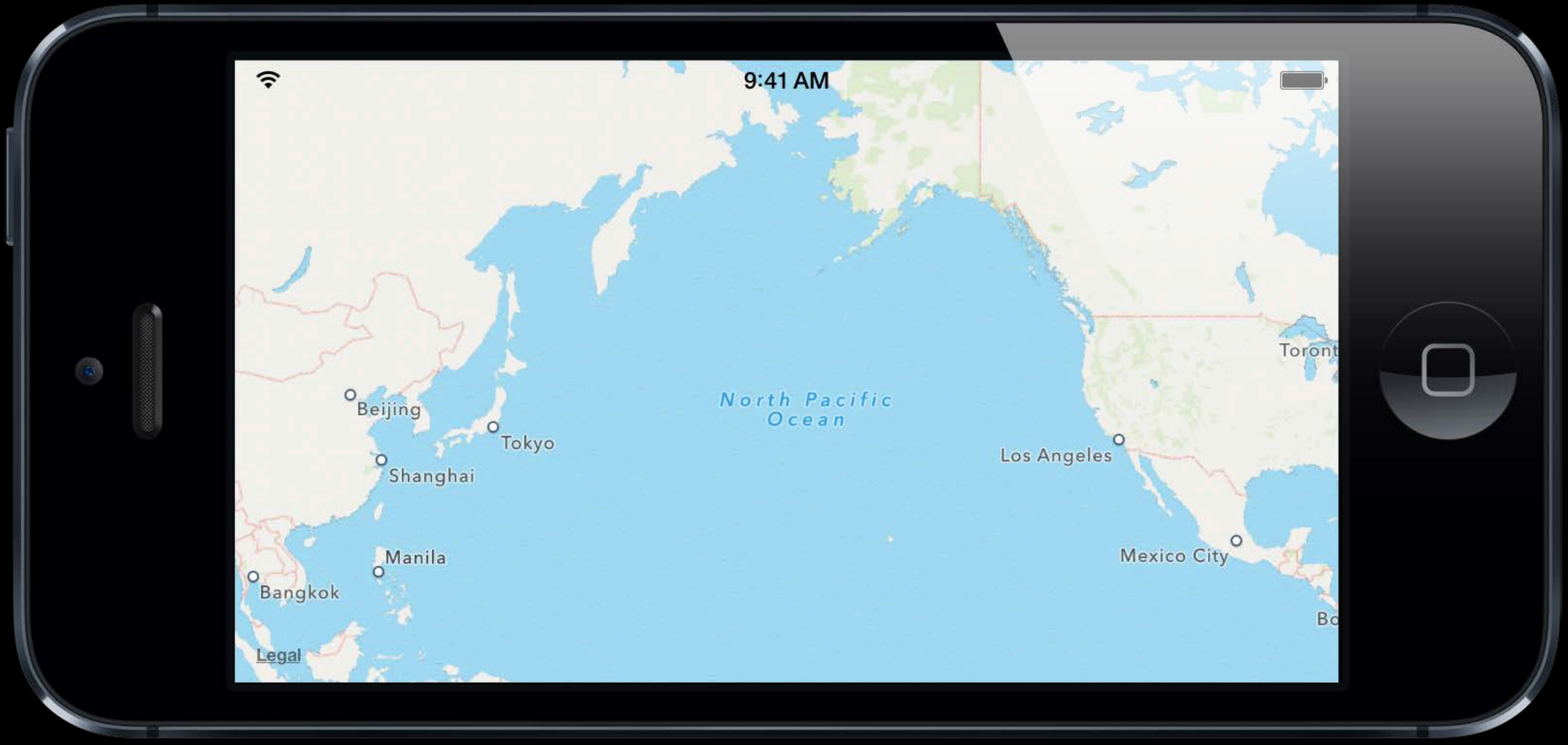


New Perspectives

New Perspectives

Spanning the 180th meridian

- Pan via touch
- Programmatically via API



New Perspectives

Positioning over 180th meridian

```
map.centerCoordinate = CLLocationCoordinate2DMake(0, 180);
```

New Perspectives

Positioning over 180th meridian

```
map.region = MKCoordinateRegionMakeWithDistance(
     CLLocationCoordinate2DMake(0, 180),
     1000.0 * 1000.0,
     1000.0 * 1000.0
);
```

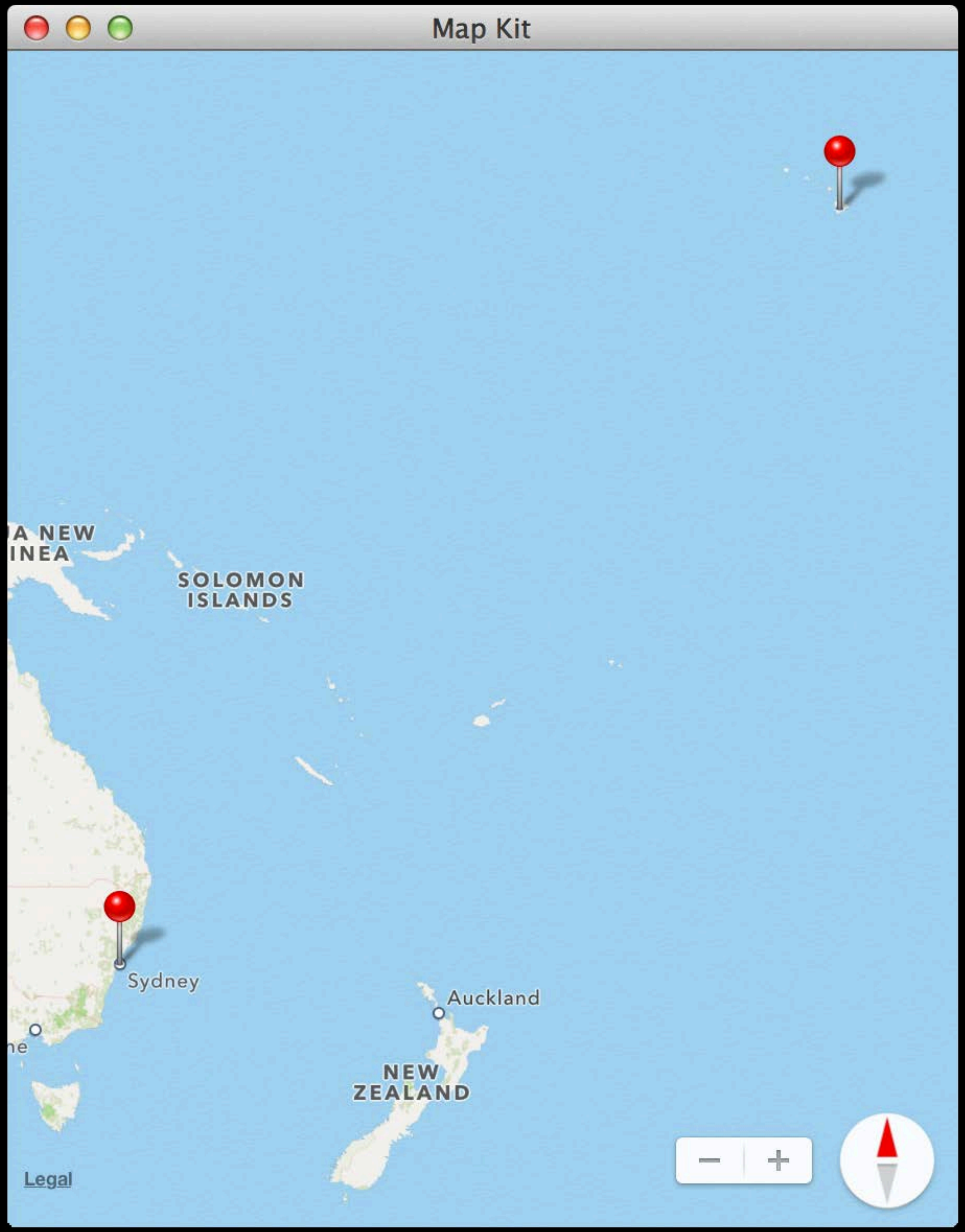
New Perspectives

Positioning over 180th meridian

```
map.visibleMapRect = MKMapRectInset(
     MKMapRectMake(
       MKMapRectGetMaxX(MKMapRectWorld),
       MKMapRectGetMidY(MKMapRectWorld),
       0, 0
     ),
     -1000.0 * 1000.0 * MKMapPointsPerMeterAtLatitude(0),
     -1000.0 * 1000.0 * MKMapPointsPerMeterAtLatitude(0)
);
```

Spanning the 180th Meridian

Enclosing annotation views on screen



Spanning the 180th Meridian

Enclosing annotation views on screen

```
MKPointAnnotation *a = [[MKPointAnnotation alloc] init];
a.title = @"Honolulu, HI";
a.coordinate = CLLocationCoordinate2DMake(19.47695,-155.566406);

MKPointAnnotation *b = [[MKPointAnnotation alloc] init];
b.title = @"Sydney, Australia";
b.coordinate = CLLocationCoordinate2DMake(-26.431228,151.347656);

[map showAnnotations:@[ a, b ] animated:NO];
```

Spanning the 180th Meridian

Enclosing annotation views on screen

```
MKPointAnnotation *a = [[MKPointAnnotation alloc] init];
a.title = @"Honolulu, HI";
a.coordinate = CLLocationCoordinate2DMake(19.47695,-155.566406);

MKPointAnnotation *b = [[MKPointAnnotation alloc] init];
b.title = @"Sydney, Australia";
b.coordinate = CLLocationCoordinate2DMake(-26.431228,151.347656);

[map showAnnotations:@[ a, b ] animated:NO];
```

Spanning the 180th Meridian

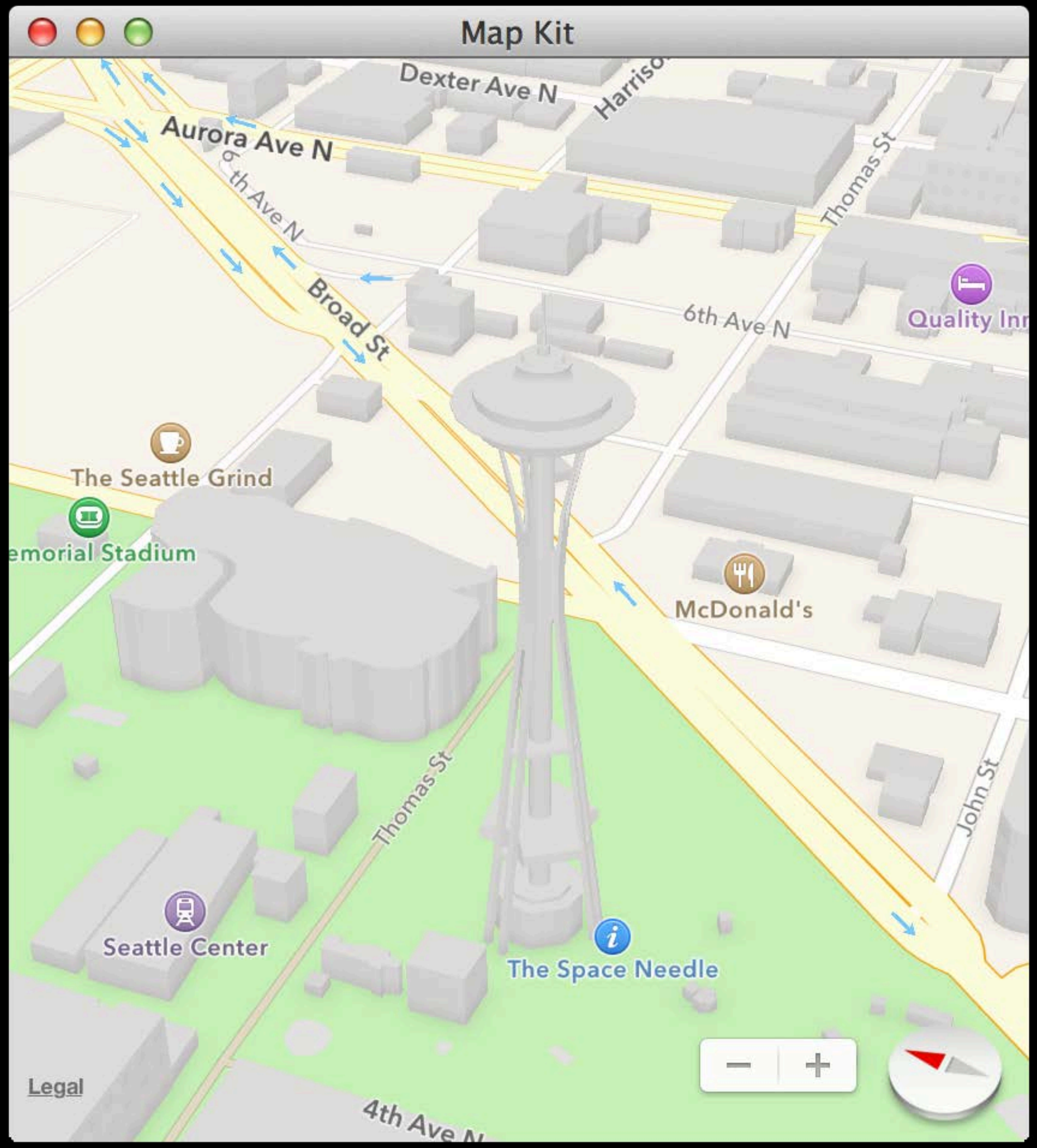
Enclosing annotation views on screen

```
MKPointAnnotation *a = [[MKPointAnnotation alloc] init];
a.title = @"Honolulu, HI";
a.coordinate = CLLocationCoordinate2DMake(19.47695,-155.566406);

MKPointAnnotation *b = [[MKPointAnnotation alloc] init];
b.title = @"Sydney, Australia";
b.coordinate = CLLocationCoordinate2DMake(-26.431228,151.347656);

[map showAnnotations:@[ a, b ] animated:NO];
```

New Perspectives

3D

- On by default on iOS 7 and OS X
- New APIs for working with 3D



Related Sessions

Putting Map Kit in Perspective	Pacific Heights Thursday 2:00 PM

Overlays

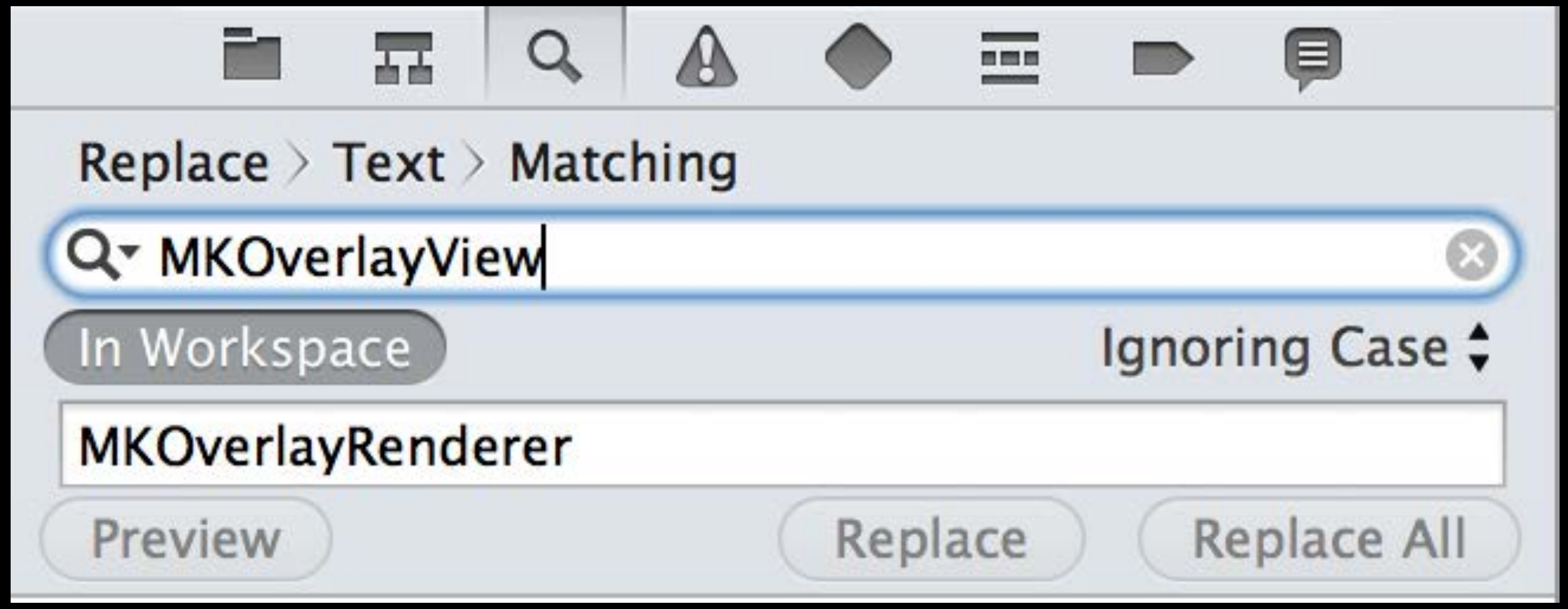
Overlays Overview

- MKOverlayRenderer
- Overlays beneath labels
- MKGeodesicPolyline
- MKTileOverlay
- Map data replacement
- Better performance

Overlays

MKOverlayRenderer

- Replace MKOverlayView with MKOverlayRenderer
- MKOverlayRenderer is lighter weight
- API is the same except no longer a view



Replace > Text > Matching

MKOverlayView

In Workspace

Ignoring Case

MKOverlayRenderer

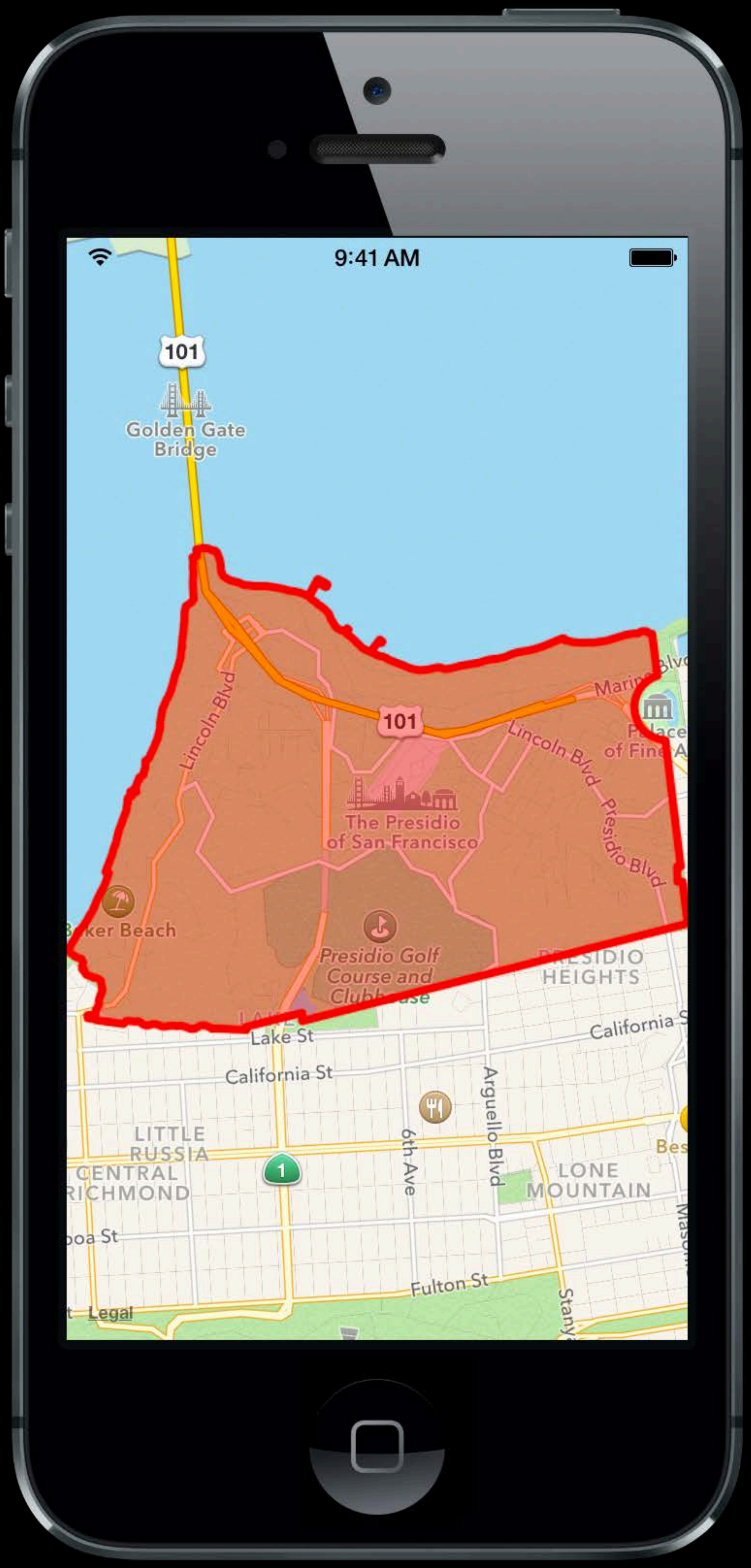
Overlays

Setting the level

- Overlays can be added to one of two levels:
    - MKOverlayLevelAboveRoads
        - Overlay rendered directly above roads, but beneath everything else (labels, points of interest, annotation views)
    - MKOverlayLevelAboveLabels
        - Overlay covers all map content, including labels, but still beneath annotation views

Overlays

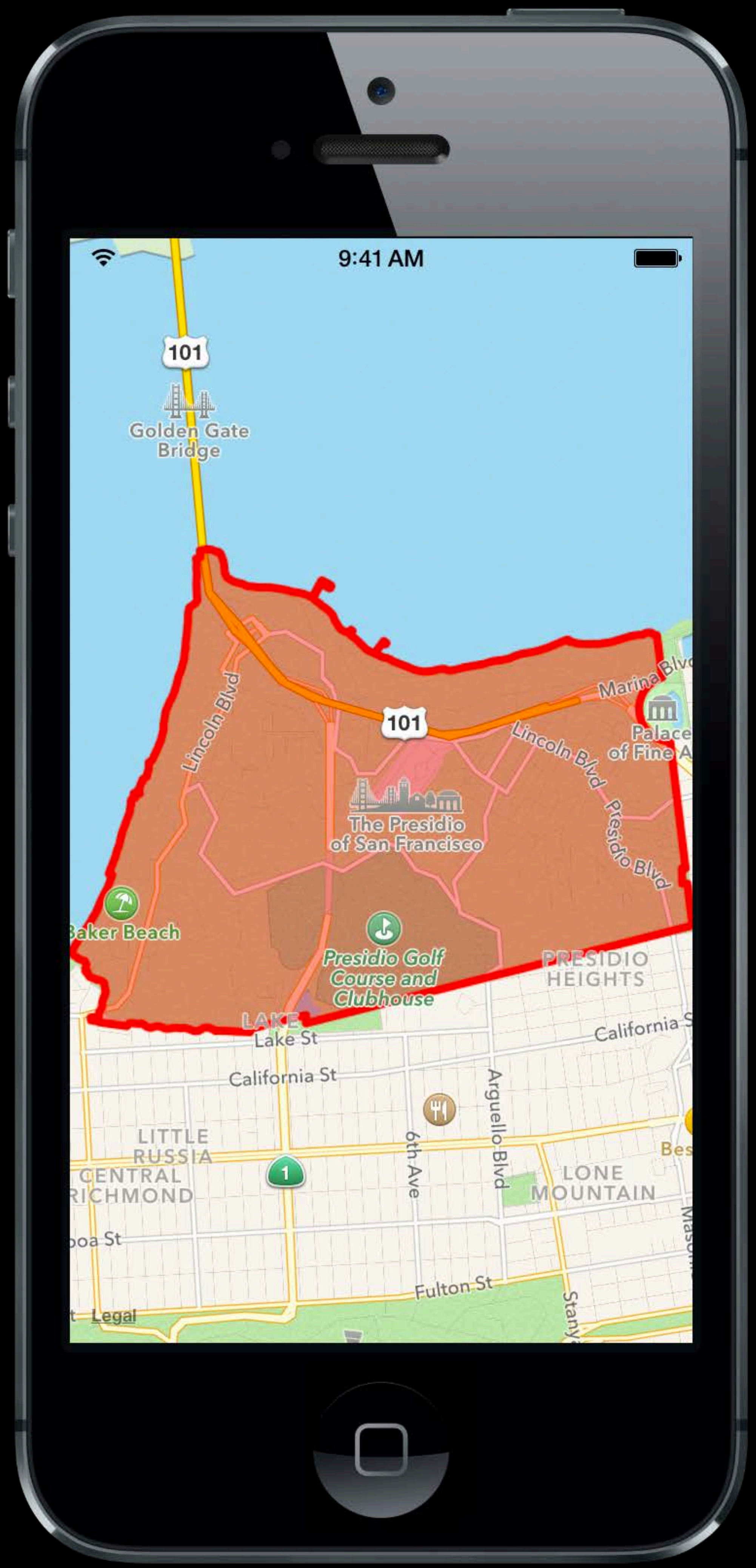
Setting the level



`MKOverlayLevelAboveLabels`

Overlays

Setting the level



`MKOverlayLevelAboveRoads`

Overlays

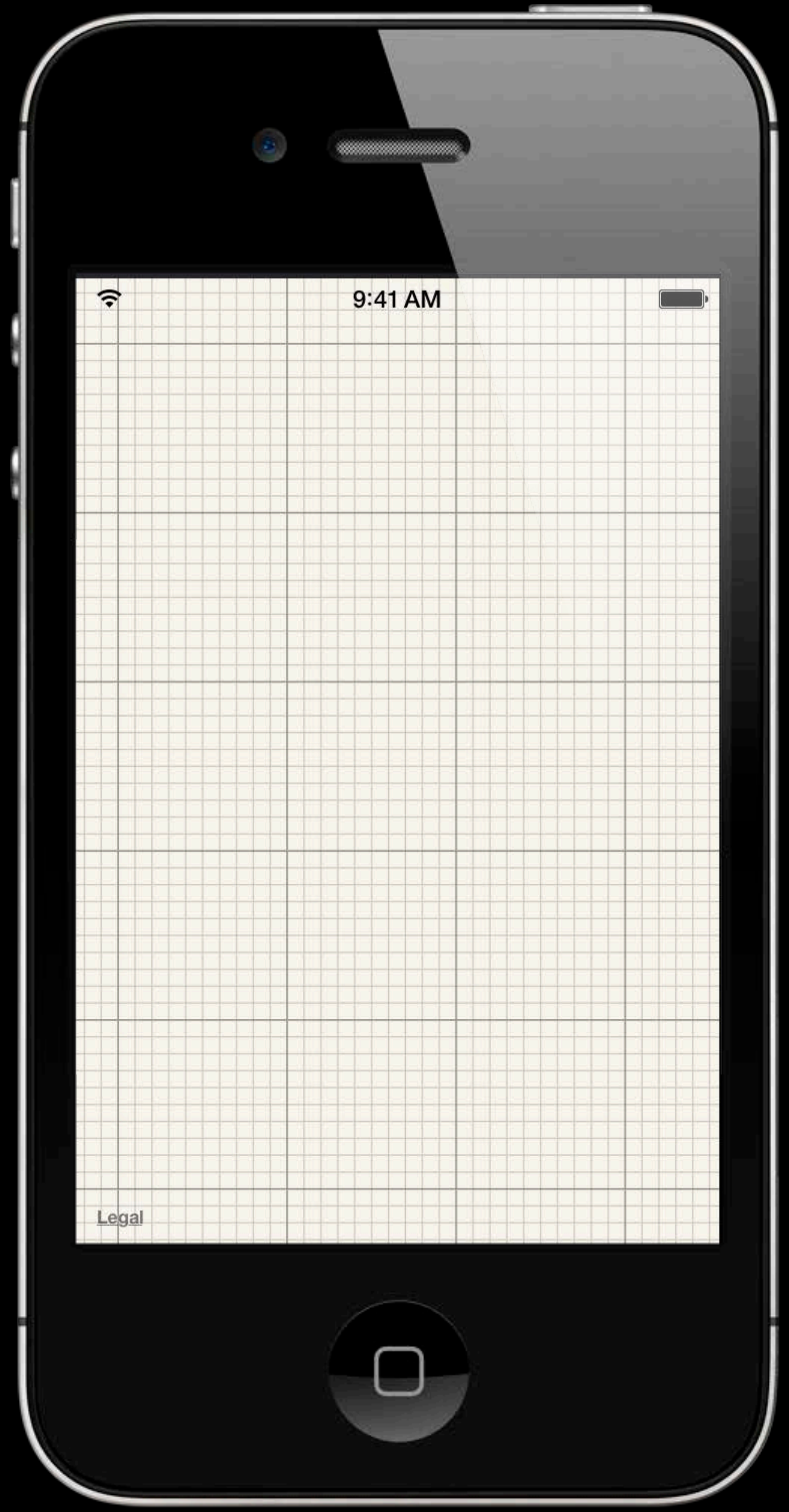
Map layers visualized

6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid

Overlays

Map layers visualized

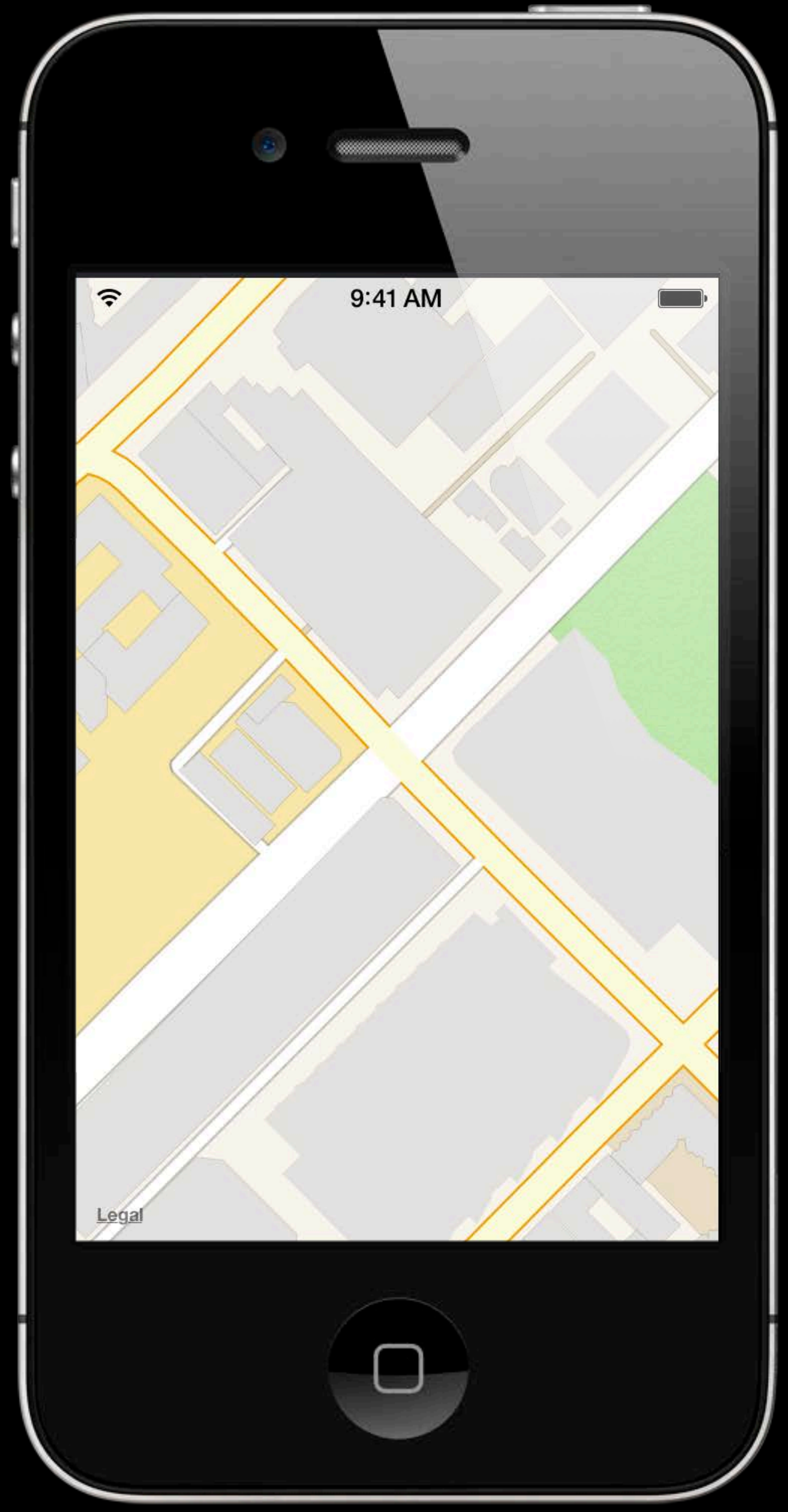
6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid



Overlays

Map layers visualized

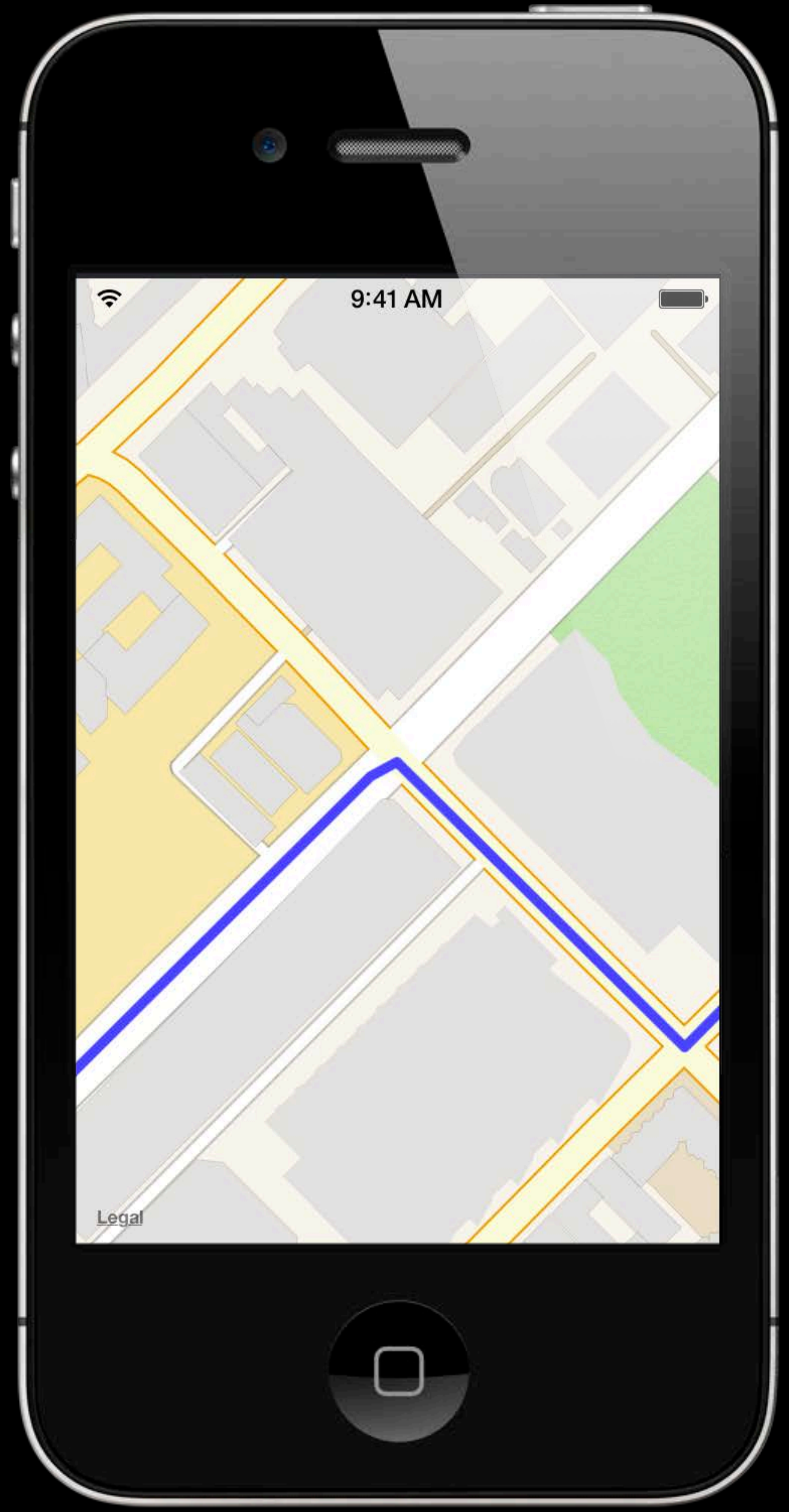
6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid



Overlays

Map layers visualized

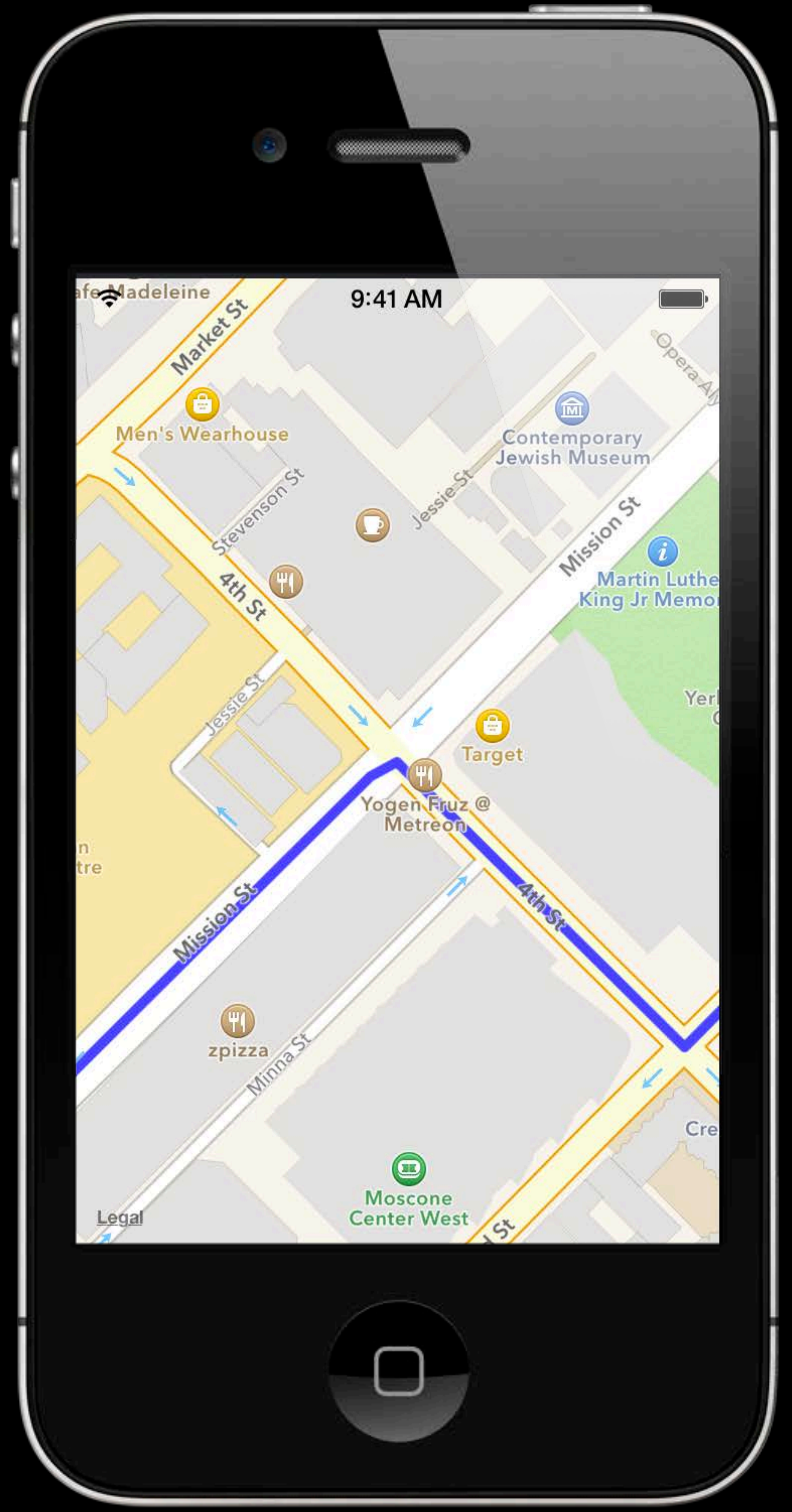
6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid



Overlays

Map layers visualized

6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid



Overlays

Map layers visualized

6. Annotation Views

5. 3D Buildings

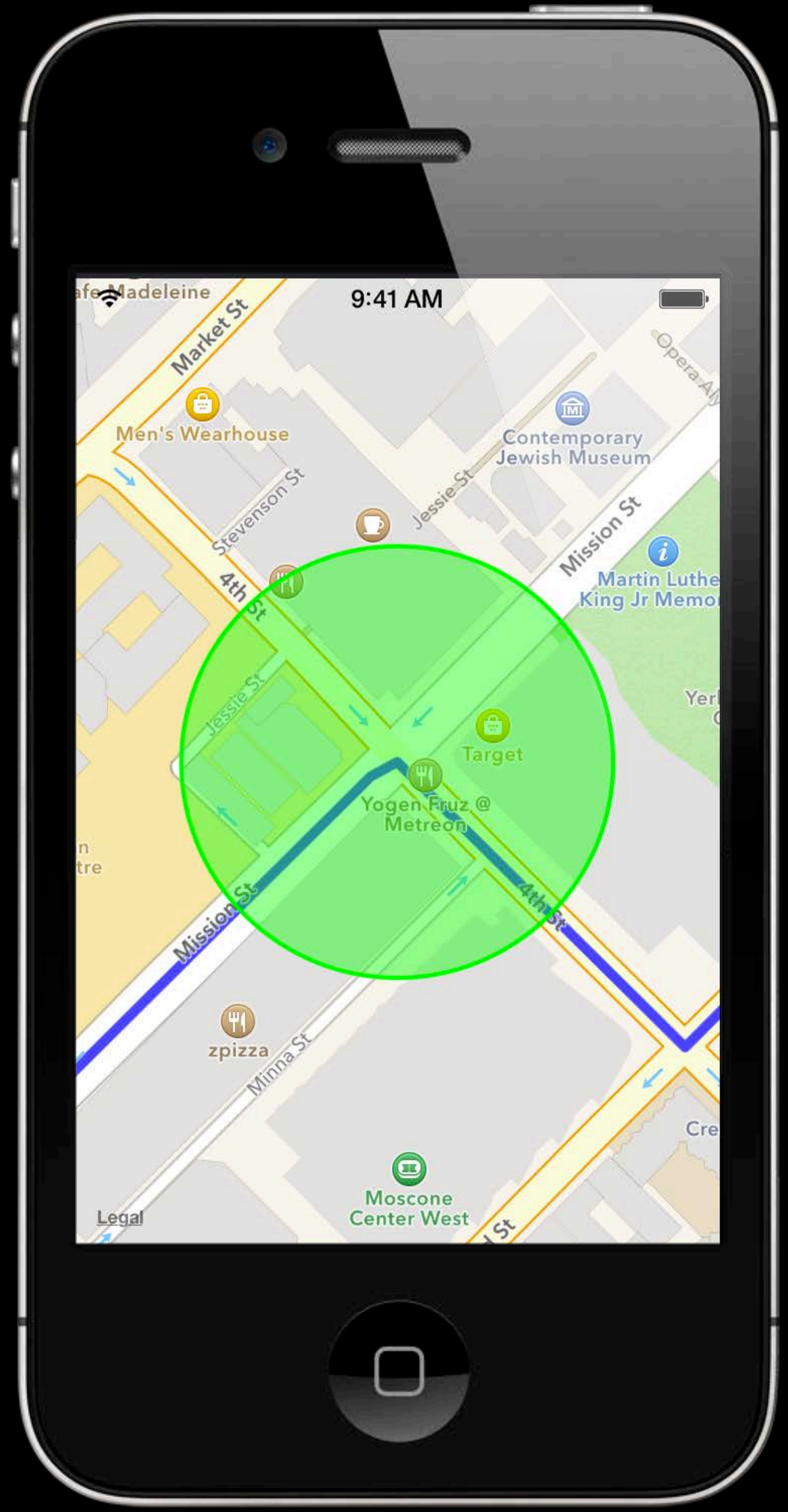
4. MKOverlayLevelAboveLabels

3. Labels

2. MKOverlayLevelAboveRoads

1. Base Map

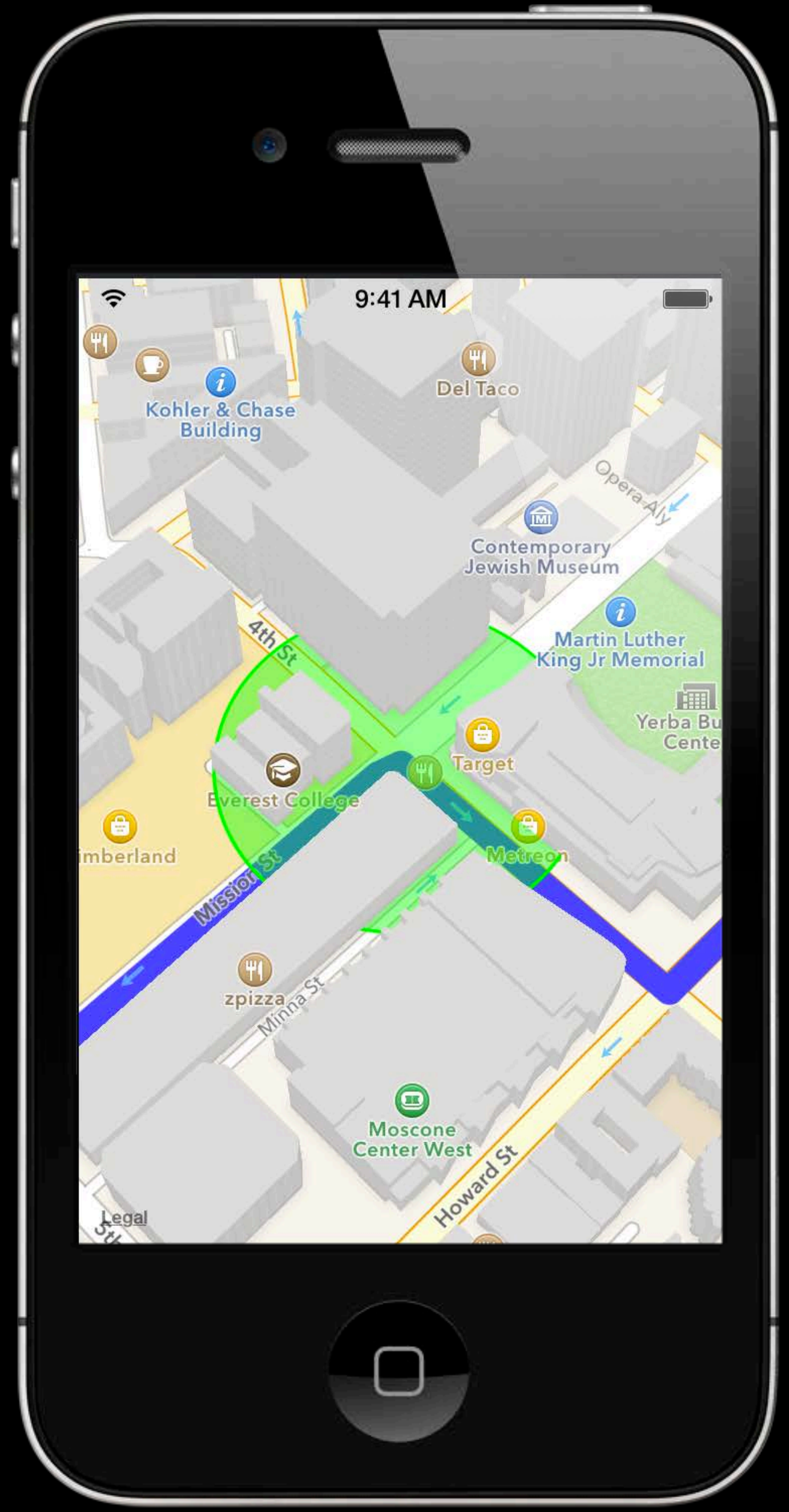
0. Grid



Overlays

Map layers visualized

6. Annotation Views
5. 3D Buildings
4. MKOverlayLevelAboveLabels
3. Labels
2. MKOverlayLevelAboveRoads
1. Base Map
0. Grid



Overlays

Map layers visualized

6. Annotation Views

5. 3D Buildings

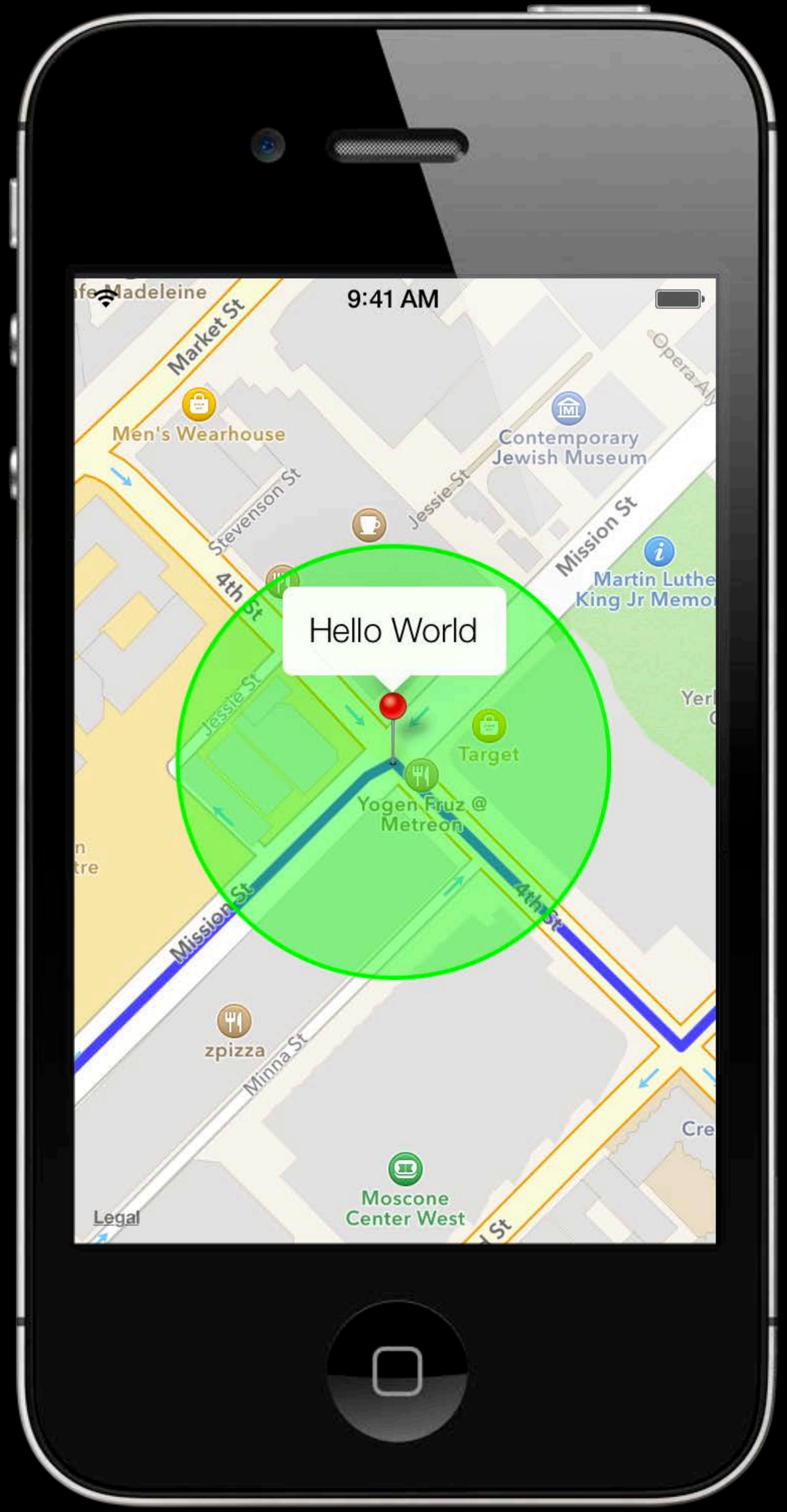
4. MKOverlayLevelAboveLabels

3. Labels

2. MKOverlayLevelAboveRoads

1. Base Map

0. Grid



Overlays

Setting the level

```
MKPolygon *presidioPolygon =
     [MKPolygon polygonWithCoordinates:presidio
                                 count:presidioSize];
[map addOverlay:presidioPolygon level:MKOverlayLevelAboveRoads];
```

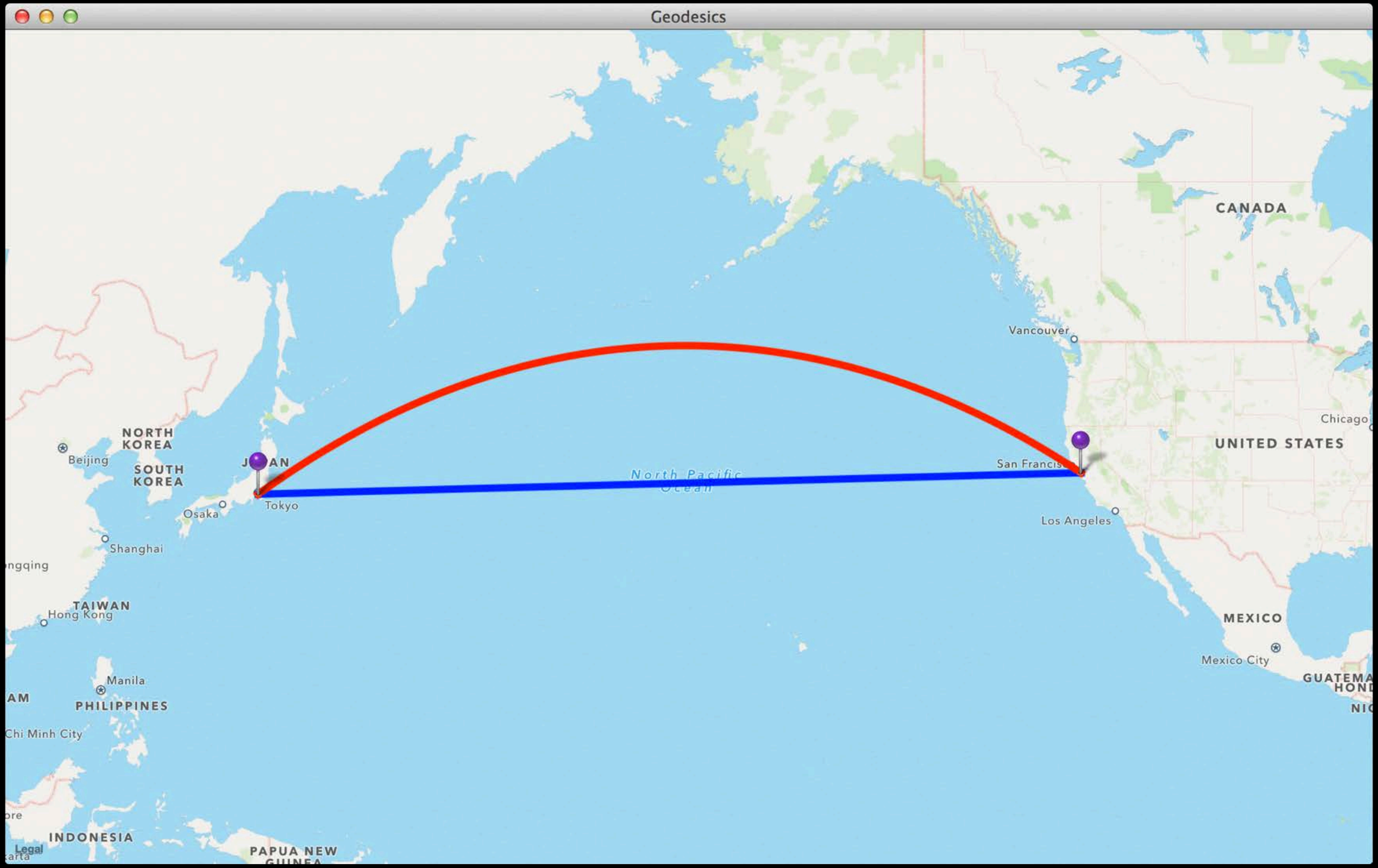
Overlays

Setting the level

```
MKPolygon *presidioPolygon =
     [MKPolygon polygonWithCoordinates:presidio
                                 count:presidioSize];
[map addOverlay:presidioPolygon level:MKOverlayLevelAboveRoads];
```

Overlays

Geodesic polyline



Overlays

Geodesic polyline

```
- (void)updateOverlays
{
    CLLocationCoordinate2D points[] =
      { start.coordinate, end.coordinate };
    geodesic = [MKGeodesicPolyline polylineWithCoordinates:points count:2];
    [map addOverlay:geodesic];
}

- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
           rendererForOverlay:(id<MKOverlay>)overlay
{
    MKPolylineRenderer *renderer =
      [[MKPolylineRenderer alloc] initWithOverlay:overlay];
    renderer.lineWidth = 5.0;
    renderer.strokeColor = [UIColor purpleColor];
    return renderer;
}
```

Overlays

Geodesic polyline

```
- (void)updateOverlays
{
    CLLocationCoordinate2D points[] =
      { start.coordinate, end.coordinate };
    geodesic = [MKGeodesicPolyline polylineWithCoordinates:points count:2];
    [map addOverlay:geodesic];
}

- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
           rendererForOverlay:(id<MKOverlay>)overlay
{
    MKPolylineRenderer *renderer =
      [[MKPolylineRenderer alloc] initWithOverlay:overlay];
    renderer.lineWidth = 5.0;
    renderer.strokeColor = [UIColor purpleColor];
    return renderer;
}
```

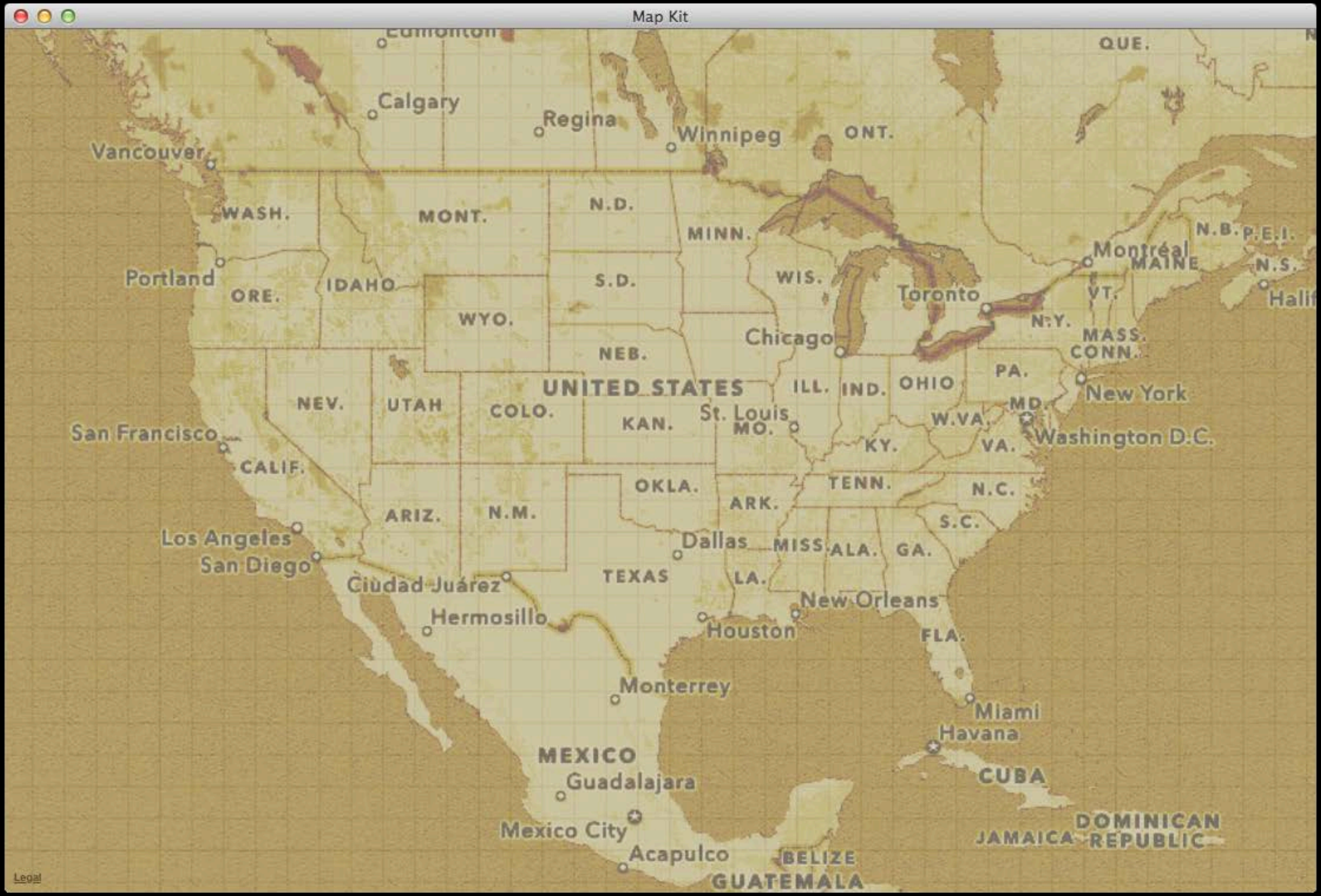
Overlays

Geodesic polyline

```
- (void)updateOverlays
{
     CLLocationCoordinate2D points[] =
       { start.coordinate, end.coordinate };
     geodesic = [MKGeodesicPolyline polylineWithCoordinates:points count:2];
     [map addOverlay:geodesic];
}

- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
            rendererForOverlay:(id<MKOverlay>)overlay
{
     MKPolylineRenderer *renderer =
       [[MKPolylineRenderer alloc] initWithOverlay:overlay];
     renderer.lineWidth = 5.0;
     renderer.strokeColor = [UIColor purpleColor];
     return renderer;
}
```

Overlays

MKTileOverlay



Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];

- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
            rendererForOverlay:(id<MKOverlay>)overlay
{
     return [[MKTileOverlayRenderer alloc] initWithOverlay:overlay];
}
```

Overlays

MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

```
- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
            rendererForOverlay:(id<MKOverlay>)overlay
{
    return [[MKTileOverlayRenderer alloc] initWithOverlay:overlay];
}
```

Overlays

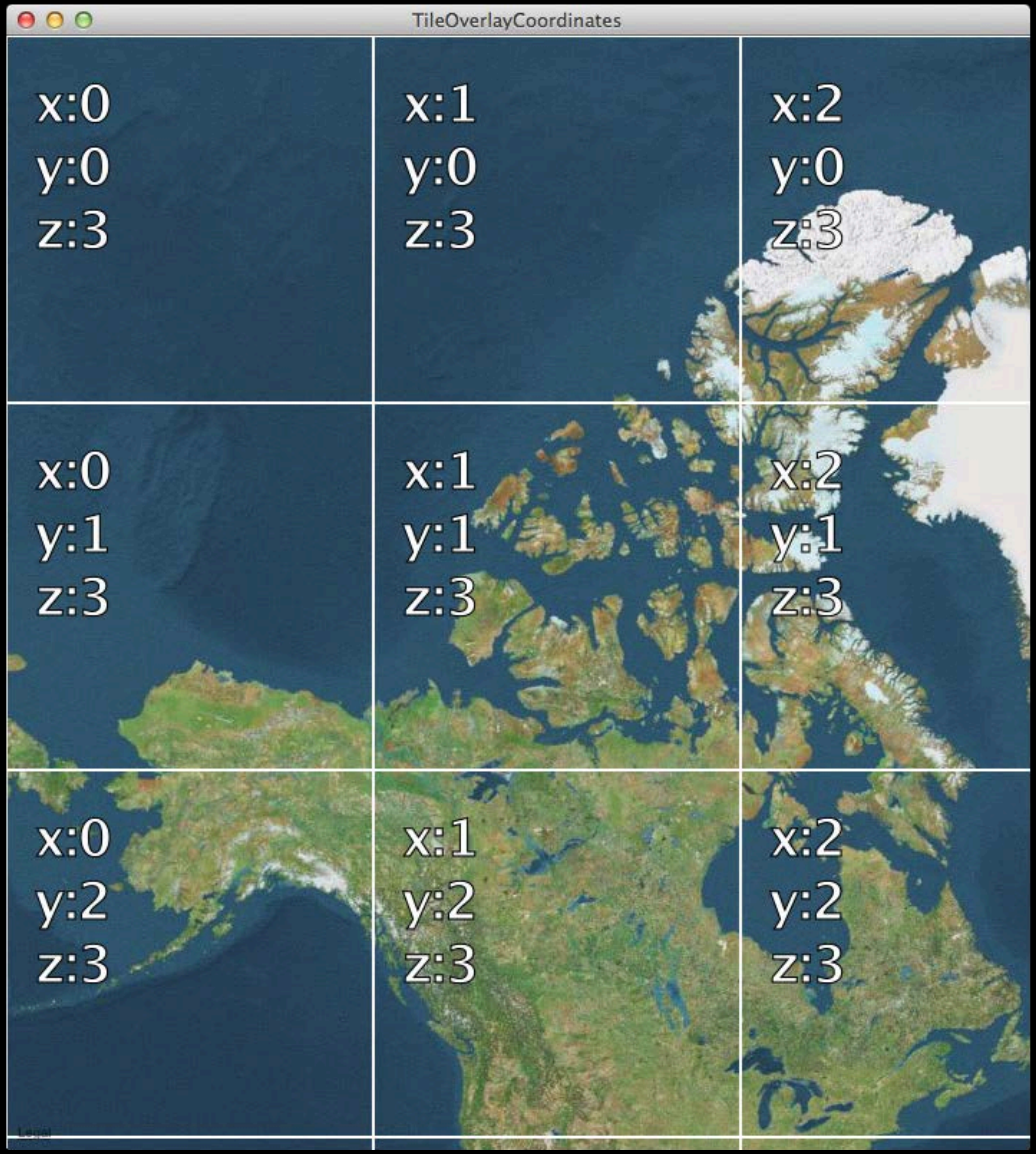
MKTileOverlay

```
NSString *template = @"http://.../tile?z={z}&x={x}&y={y}";
overlay = [[MKTileOverlay alloc] initWithURLTemplate:template];
overlay.canReplaceMapContent = YES;
[map addOverlay:overlay level:MKOverlayLevelAboveLabels];
```

```
- (MKOverlayRenderer *)mapView:(MKMapView *)mapView
            rendererForOverlay:(id<MKOverlay>)overlay
{
    return [[MKTileOverlayRenderer alloc] initWithOverlay:overlay];
}
```

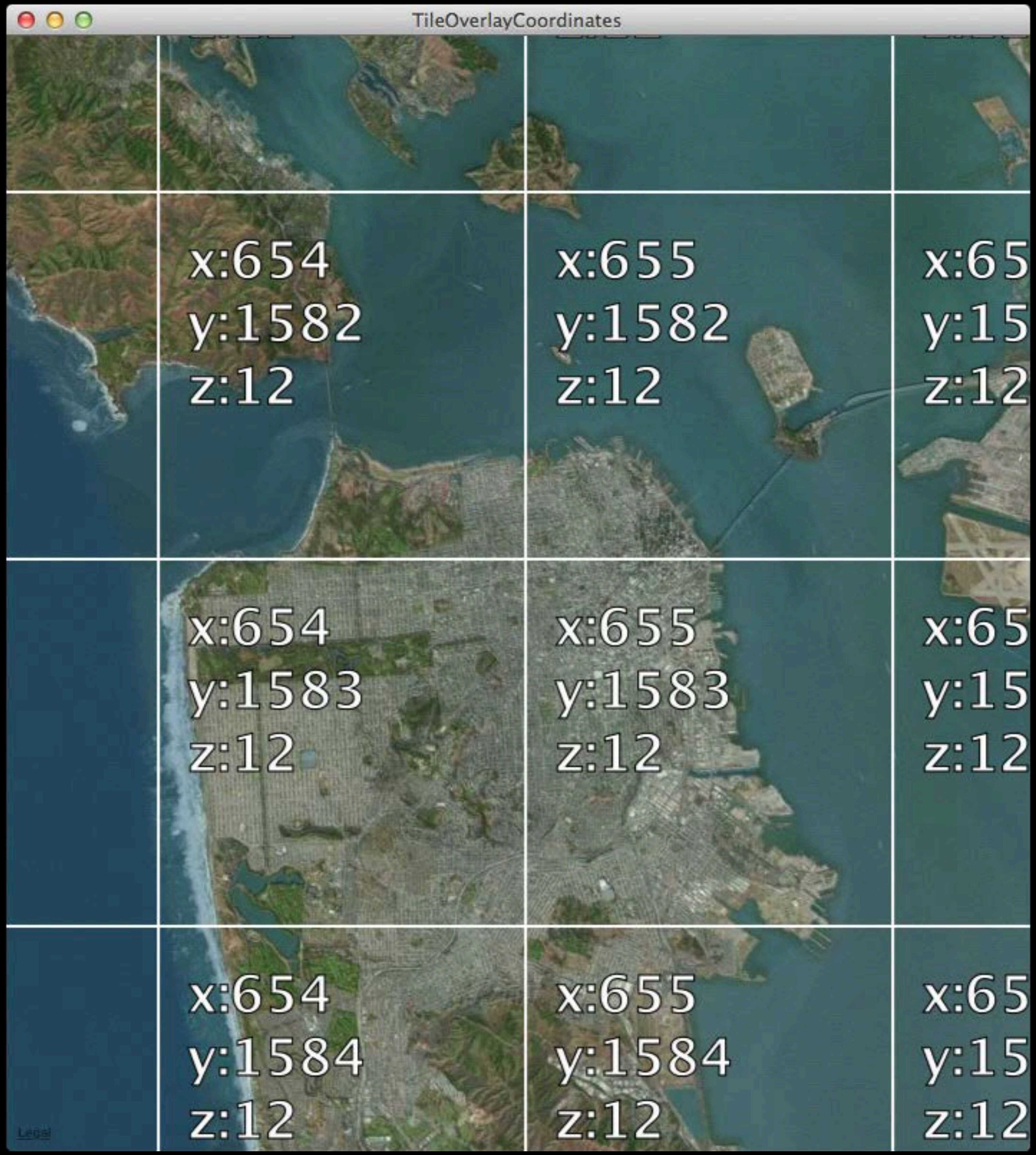
Overlays

`MKTileOverlay` coordinate system



Overlays

`MKTileOverlay` coordinate system



Overlays

MKTileOverlay coordinate system

Overlays

`MKTileOverlay` coordinate system

- Spherical mercator projection (EPSG:3857)

Overlays

`MKTileOverlay` coordinate system

- Spherical mercator projection (EPSG:3857)
- Various tools can produce tile images for use with MKTileOverlay
  - GDAL is a good one (http://www.gdal.org)

Overlays

`MKTileOverlay` coordinate system

- Spherical mercator projection (EPSG:3857)
- Various tools can produce tile images for use with MKTileOverlay
    - GDAL is a good one (http://www.gdal.org)
- Same system as popular web mapping frameworks

Overlays

Replacing Apple data with your data

Overlays

Replacing Apple data with your data

- Optional method on MKOverlay protocol

```
– (BOOL)canReplaceMapContent
```

Overlays

Replacing Apple data with your data

- Optional method on MKOverlay protocol

```
– (BOOL)canReplaceMapContent
```

- Settable property on MKTileOverlay

```
@property (assign) BOOL canReplaceMapContent
```

Overlays

Replacing Apple data with your data

- Optional method on MKOverlay protocol

```
– (BOOL)canReplaceMapContent
```

- Settable property on MKTileOverlay

```
@property (assign) BOOL canReplaceMapContent
```

- Apple map data is neither loaded nor drawn

Overlays

Replacing Apple data with your data

- Optional method on MKOverlay protocol

```
- (BOOL)canReplaceMapContent
```

- Settable property on MKTileOverlay

```
@property (assign) BOOL canReplaceMapContent
```

- Apple map data is neither loaded nor drawn
- Typically use with MKOverlayLevelAboveLabels

Overlays

Replacing Apple data with your data

- Optional method on MKOverlay protocol

```
– (BOOL)canReplaceMapContent
```

- Settable property on MKTileOverlay

```
@property (assign) BOOL canReplaceMapContent
```

- Apple map data is neither loaded nor drawn
- Typically use with MKOverlayLevelAboveLabels
- But does work with MKOverlayLevelAboveRoads
    - Only Apple data necessary to draw labels is loaded

Demo

MKTileOverlay

MKTileOverlay

Lessons learned

- Tiles can be bundled with your app, or downloaded from your server
- MKTileOverlay can replace Apple Map Data, or show atop it

Local Search

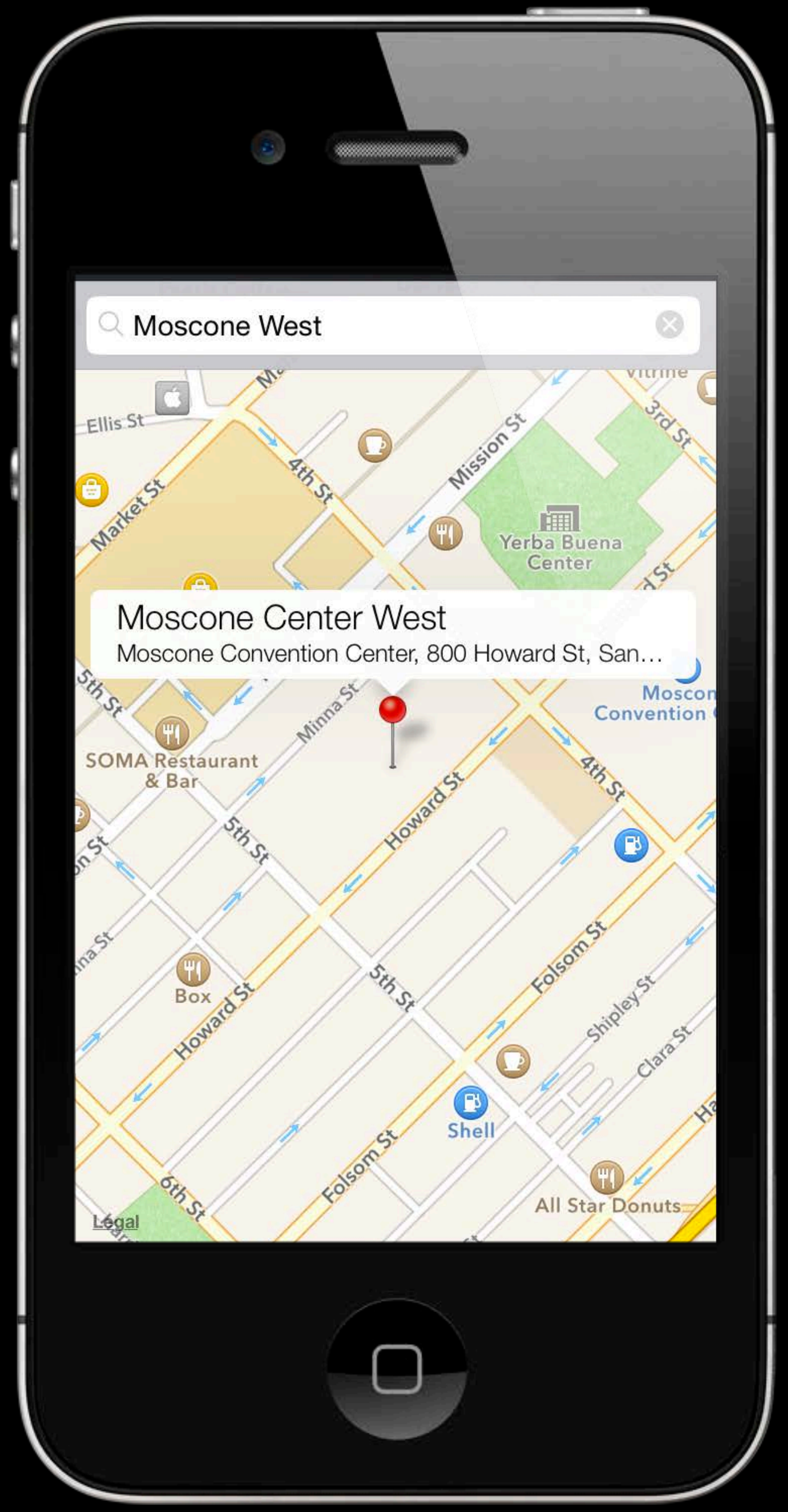
Local Search

Local Search

- Allows you to find ...

Local Search

- Allows you to find ...
    - A specific place by name



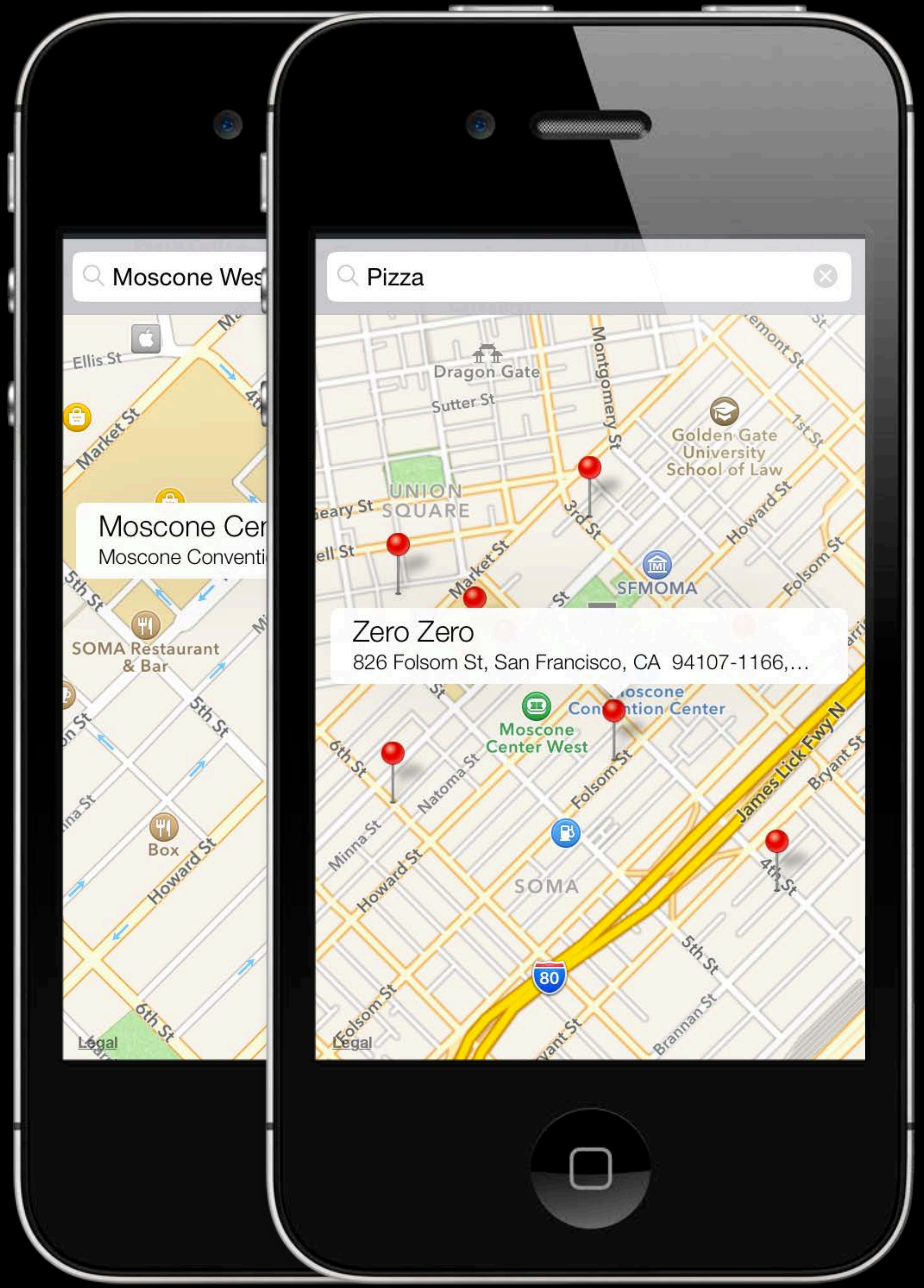
Moscone West

Moscone Center West

Moscone Convention Center, 800 Howard St, San…

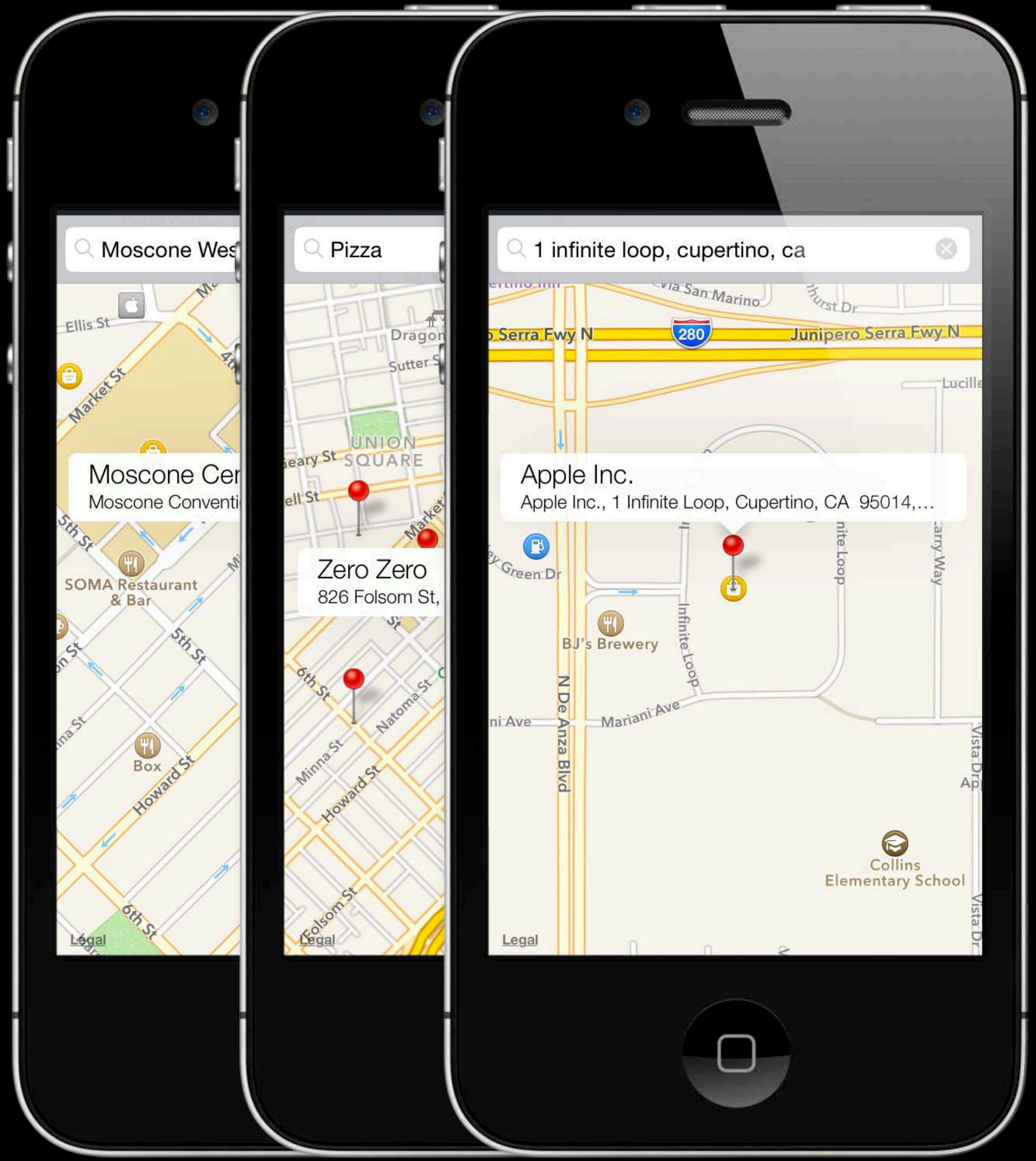
Local Search

- Allows you to find ...
  - A specific place by name
  - A list of places by type, e.g. “Pizza”



Local Search

- Allows you to find ...
  - A specific place by name
  - A list of places by type, e.g. "Pizza"
  - Addresses



Moscone Wes

Moscone Cen
Moscone Conventi

Pizza

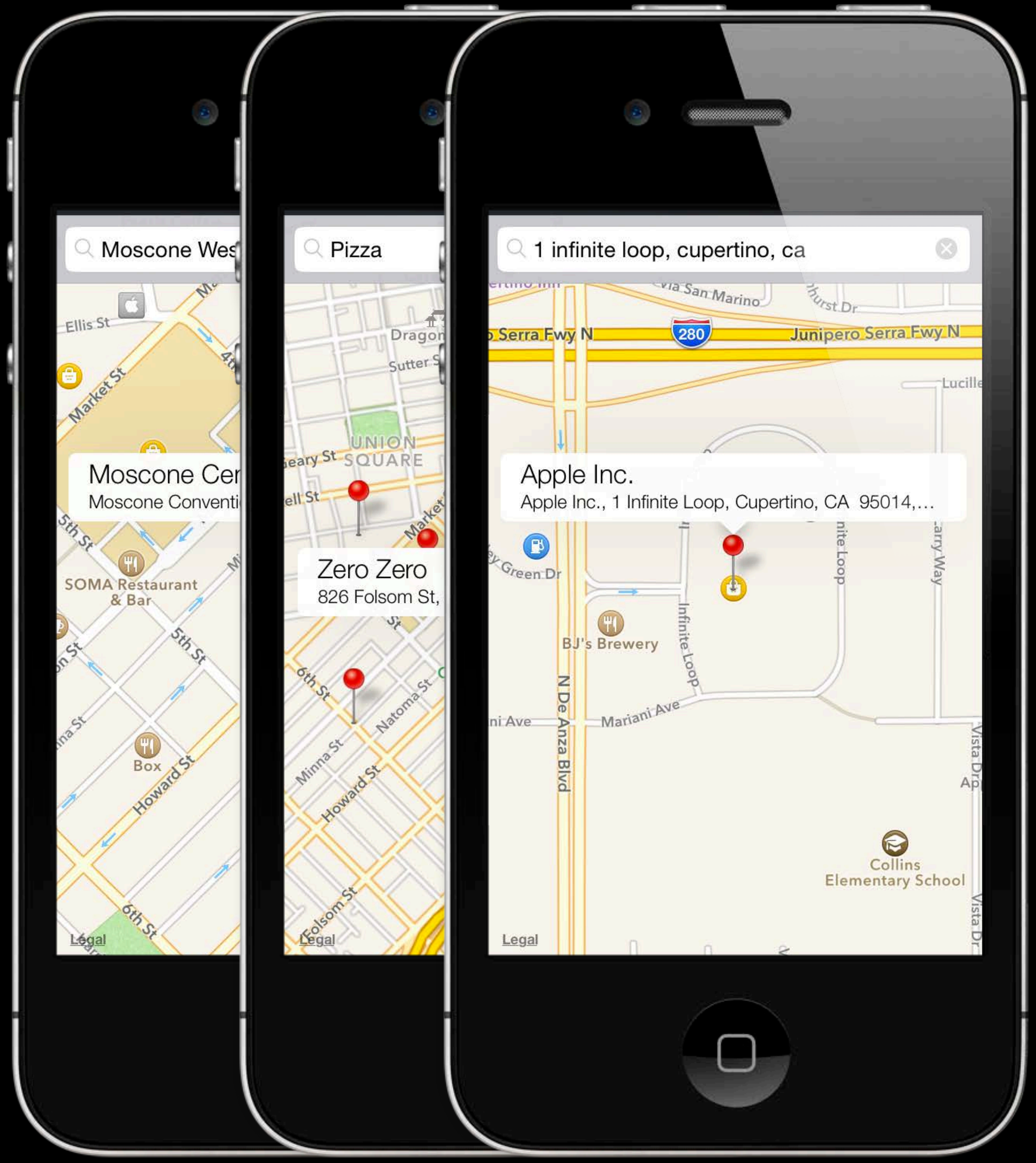
Zero Zero
826 Folsom St,

1 infinite loop, cupertino, ca

Apple Inc.
Apple Inc., 1 Infinite Loop, Cupertino, CA 95014,...

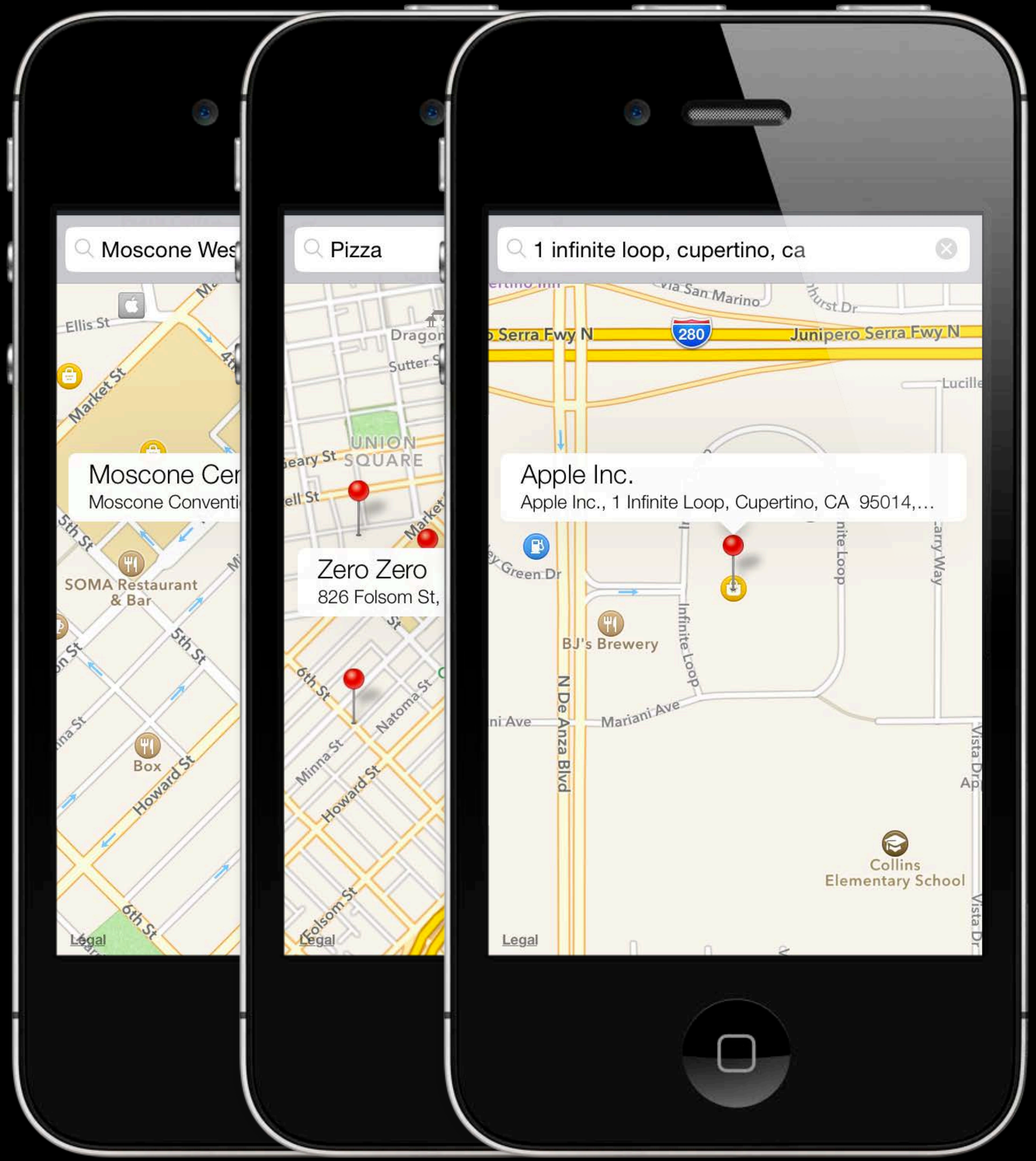
Local Search

- Allows you to find ...
    - A specific place by name
    - A list of places by type, e.g. "Pizza"
    - Addresses
- Introduced in iOS 6.1


Moscone Wes
Moscone Cer
Moscone Conventi
Pizza
Zero Zero
826 Folsom St,
1 infinite loop, cupertino, ca
Apple Inc.
Apple Inc., 1 Infinite Loop, Cupertino, CA 95014,...
Legal

Local Search

- Allows you to find ...
    - A specific place by name
    - A list of places by type, e.g. "Pizza"
    - Addresses
- Introduced in iOS 6.1
- Available on OS X 10.9



Local Search

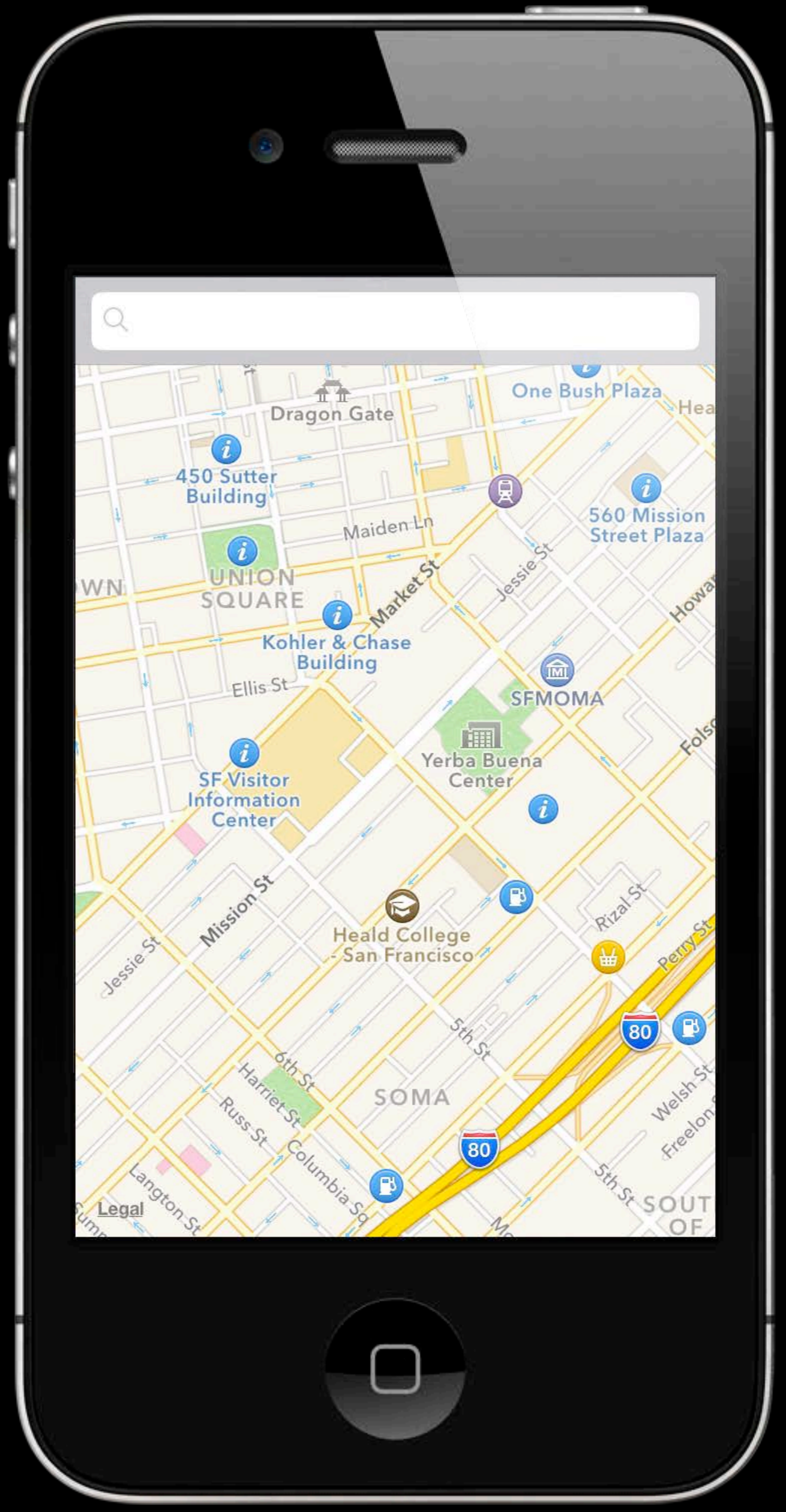
VS Geocoding

- Reverse Geocoding
    - Coordinate → Address

Local Search

VS Geocoding

- Reverse Geocoding
  - Coordinate → Address



Local Search

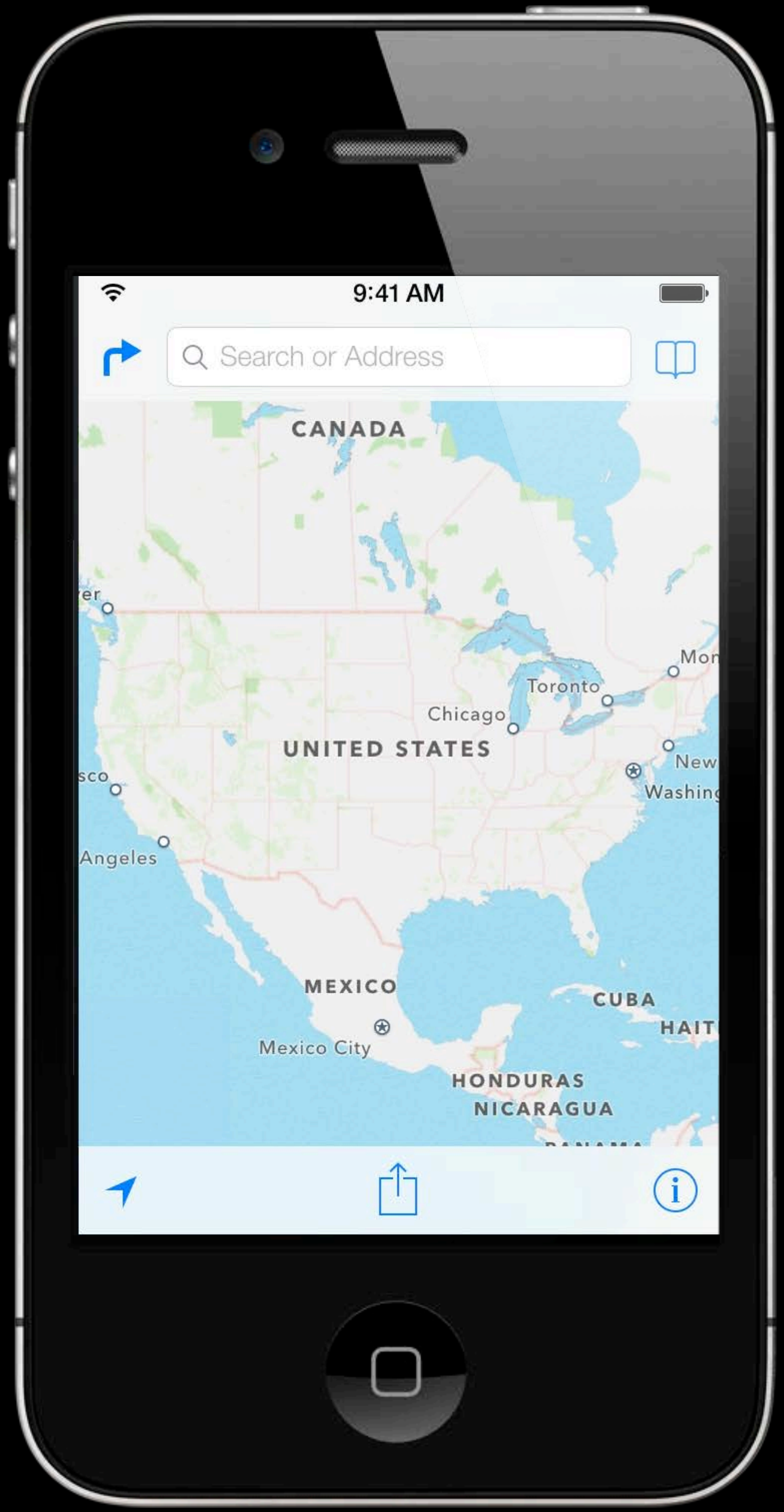
VS Geocoding

- Reverse Geocoding
    - Coordinate → Address
- Forward Geocoding
    - Address → Coordinate

Local Search

VS Geocoding

- Reverse Geocoding
  - Coordinate → Address
- Forward Geocoding
  - Address → Coordinate



Local Search

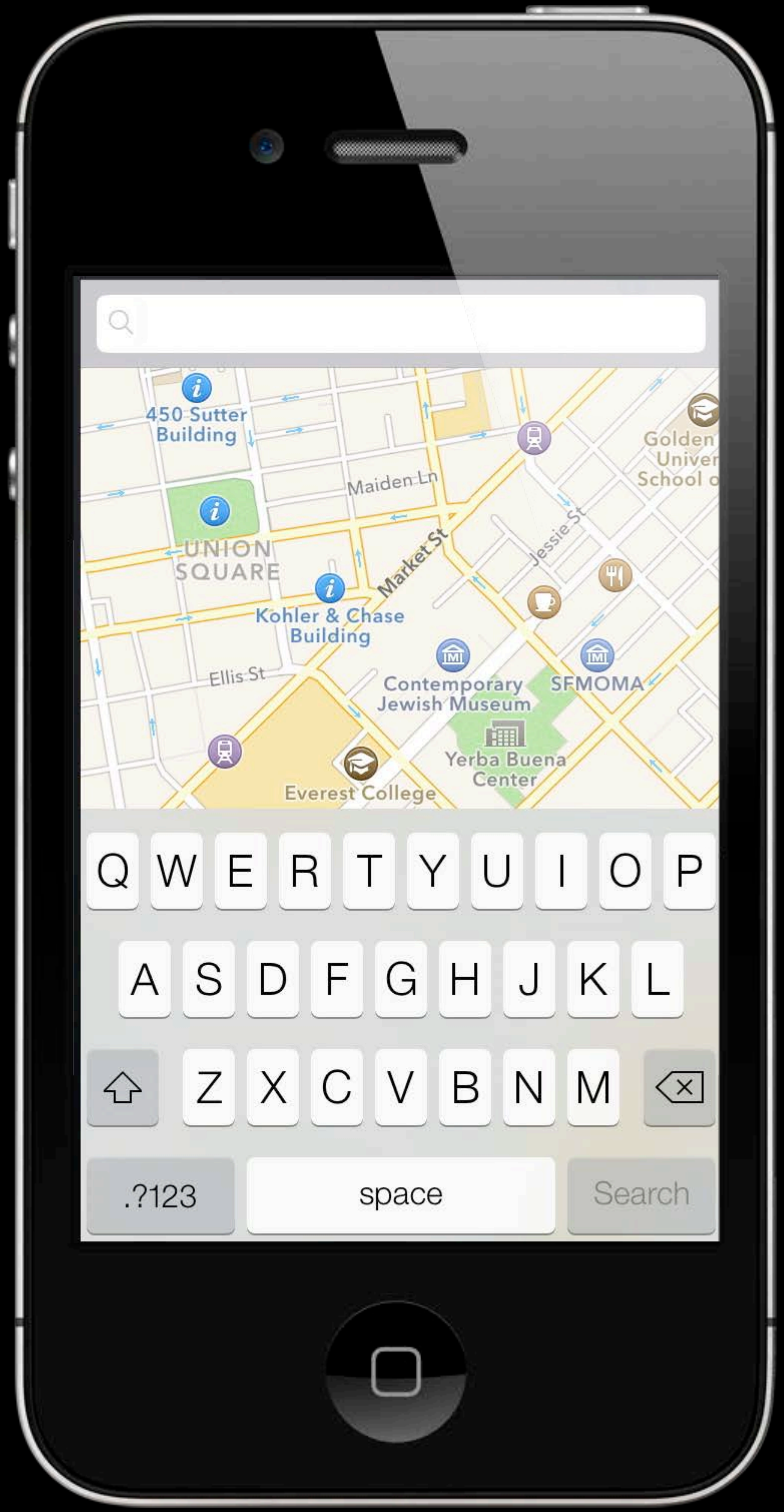
VS Geocoding

- Reverse Geocoding
  - Coordinate → Address
- Forward Geocoding
  - Address → Coordinate
- Local Search
  - Freeform query string → List of Search Results

Local Search

VS Geocoding

- Reverse Geocoding
  - Coordinate → Address
- Forward Geocoding
  - Address → Coordinate
- Local Search
  - Freeform query string → List of Search Results



Local Search Results

- Represented by MKMapItem
    - Coordinate
    - Structured Address
    - Name
    - Phone #
    - URL

Local Search

Performing search requests

```
// Initialize request object
MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
request.naturalLanguageQuery = query;
request.region = self.map.region

// Initialize search object
MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request]

// Run search and display results
[search startWithCompletionHandler:
 ^(MKLocalSearchResponse *response, NSError *error)
{

}];
```

Local Search

Performing search requests

```
// Initialize request object
MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
request.naturalLanguageQuery = query;
request.region = self.map.region

// Initialize search object
MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request]

// Run search and display results
[search startWithCompletionHandler:
 ^(MKLocalSearchResponse *response, NSError *error)
{

}];
```

Local Search

Performing search requests

```
// Initialize request object
MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
request.naturalLanguageQuery = query;
request.region = self.map.region

// Initialize search object
MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request]

// Run search and display results
[search startWithCompletionHandler:
 ^(MKLocalSearchResponse *response, NSError *error)
{

}];
```

Local Search

Performing search requests

```
// Initialize request object
MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
request.naturalLanguageQuery = query;
request.region = self.map.region

// Initialize search object
MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request]

// Run search and display results
[search startWithCompletionHandler:
 ^(MKLocalSearchResponse *response, NSError *error)
{

}];
```

Local Search

Performing search requests

```
// Initialize request object
MKLocalSearchRequest *request = [[MKLocalSearchRequest alloc] init];
request.naturalLanguageQuery = query;
request.region = self.map.region

// Initialize search object
MKLocalSearch *search = [[MKLocalSearch alloc] initWithRequest:request]

// Run search and display results
[search startWithCompletionHandler:
 ^(MKLocalSearchResponse *response, NSError *error)
{
  NSMutableArray *placemarks = [NSMutableArray array];
  for (MKMapItem *item in response.mapItems) {
    [placemarks addObject:item.placemark];
  }
  [self.map removeAnnotations:[self.map annotations]];
  [self.map showAnnotations:placemarks animated:NO];
}];
```

Directions

Directions

- New API in iOS 7 and OS X 10.9
- General purpose route finding API
- Driving and walking directions
- Alternate routes
- Time estimates
    - By default based on current traffic
    - Set departure date to future date to access historical traffic

Directions

Requesting directions

```
- (void)findDirectionsFrom:(MKMapItem *)source
                        to:(MKMapItem *)destination
{
 MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
 request.source = source;
 request.destination = destination;
 request.requestsAlternateRoutes = YES;

 MKDirections *directions = [[MKDirections alloc] initWithRequest:request]
 [directions calculateDirectionsWithCompletionHandler:
 ^(MKDirectionsResponse *response, NSError *error) {
   if (error) {
     [self handleError:error];
   } else {
     [self showDirections:response];
   }
 }];
}
```

Directions

Requesting directions

```
- (void)findDirectionsFrom:(MKMapItem *)source
                        to:(MKMapItem *)destination
{
 MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
 request.source = source;
 request.destination = destination;
 request.requestsAlternateRoutes = YES;

 MKDirections *directions = [[MKDirections alloc] initWithRequest:request]
 [directions calculateDirectionsWithCompletionHandler:
 ^(MKDirectionsResponse *response, NSError *error) {
   if (error) {
     [self handleError:error];
   } else {
     [self showDirections:response];
   }
 }];
}
```

Directions

Requesting directions

```
- (void)findDirectionsFrom:(MKMapItem *)source
                        to:(MKMapItem *)destination
{
 MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
 request.source = source;
 request.destination = destination;
 request.requestsAlternateRoutes = YES;

 MKDirections *directions = [[MKDirections alloc] initWithRequest:request]
 [directions calculateDirectionsWithCompletionHandler:
 ^(MKDirectionsResponse *response, NSError *error) {
   if (error) {
     [self handleError:error];
   } else {
     [self showDirections:response];
   }
 }];
}
```

Directions

Requesting directions

```
- (void)findDirectionsFrom:(MKMapItem *)source
                        to:(MKMapItem *)destination
{
 MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
 request.source = source;
 request.destination = destination;
 request.requestsAlternateRoutes = YES;

 MKDirections *directions = [[MKDirections alloc] initWithRequest:request]
 [directions calculateDirectionsWithCompletionHandler:
 ^(MKDirectionsResponse *response, NSError *error) {
   if (error) {
     [self handleError:error];
   } else {
     [self showDirections:response];
   }
 }];
}
```

Directions

Requesting directions

```
- (void)findDirectionsFrom:(MKMapItem *)source
                        to:(MKMapItem *)destination
{
 MKDirectionsRequest *request = [[MKDirectionsRequest alloc] init];
 request.source = source;
 request.destination = destination;
 request.requestsAlternateRoutes = YES;

 MKDirections *directions = [[MKDirections alloc] initWithRequest:request]
 [directions calculateDirectionsWithCompletionHandler:
 ^(MKDirectionsResponse *response, NSError *error) {
   if (error) {
     [self handleError:error];
   } else {
     [self showDirections:response];
   }
 }];
}
```

Directions

What’s in a response

- Source and destination
- Route(s)
    - Localized Name
        - “US 101-S”
    - Distance
    - Localized advisory notices
        - “This route requires tolls”
    - Expected travel time
    - Geometry as a polyline
    - Steps …

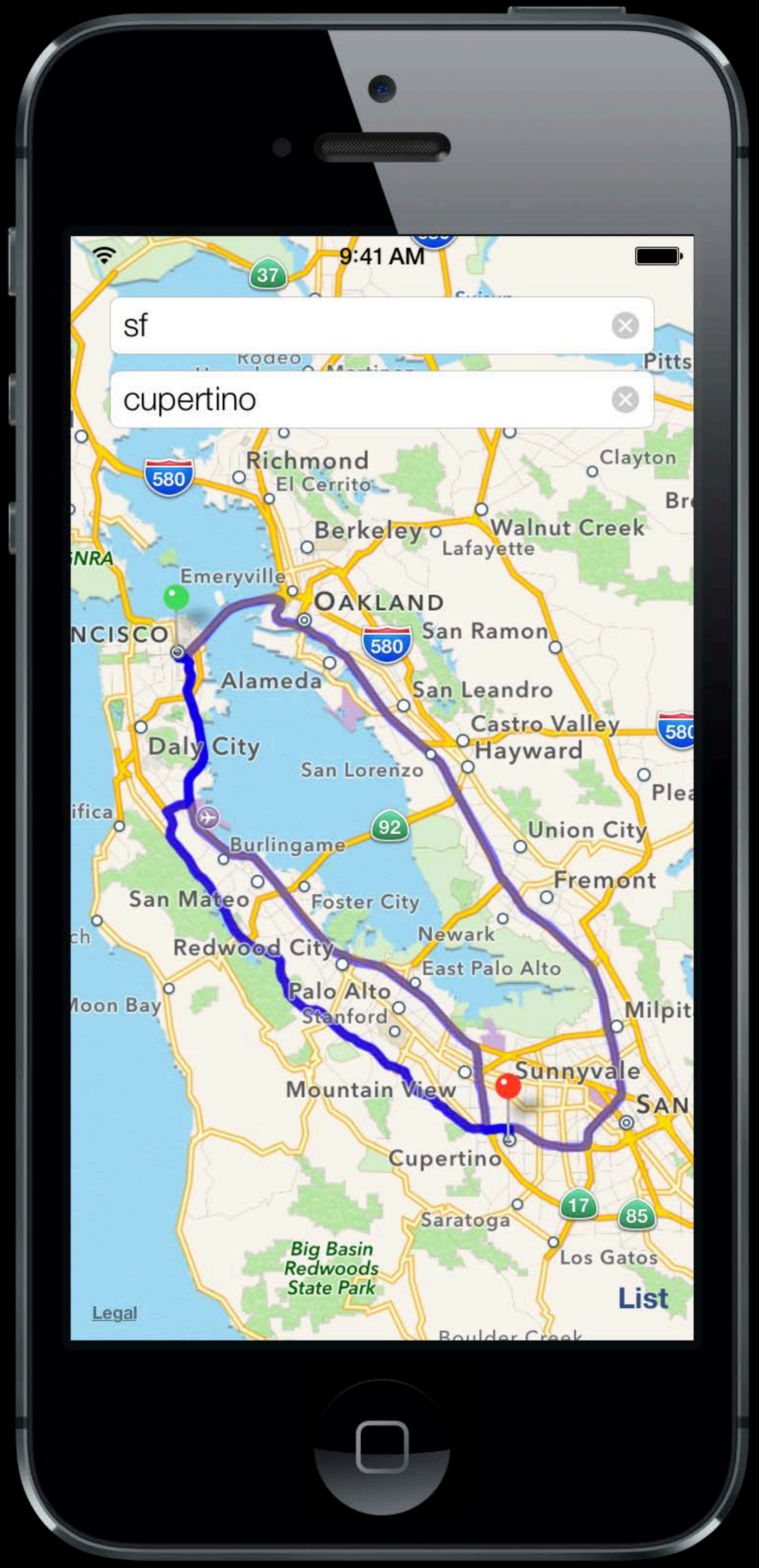
Directions

What’s in a step

- Localized instructions
    - “Take exit 11 onto De Anza Boulevard”
- Localized notice
    - Only present where applicable
    - “Do not cross tracks when lights flash”
- Distance
- Transport type (walking or driving)
    - May differ from overall route transport type
- Step geometry as polyline

Directions

Displaying routes



Directions

Displaying routes

```
- (void)showDirections:(MKDirectionsResponse *)response
{
     self.response = response;

     for (MKRoute *route in self.response.routes) {
       [self.map addOverlay:route.polyline level:MKOverlayLevelAboveRoads];
     }
}
```

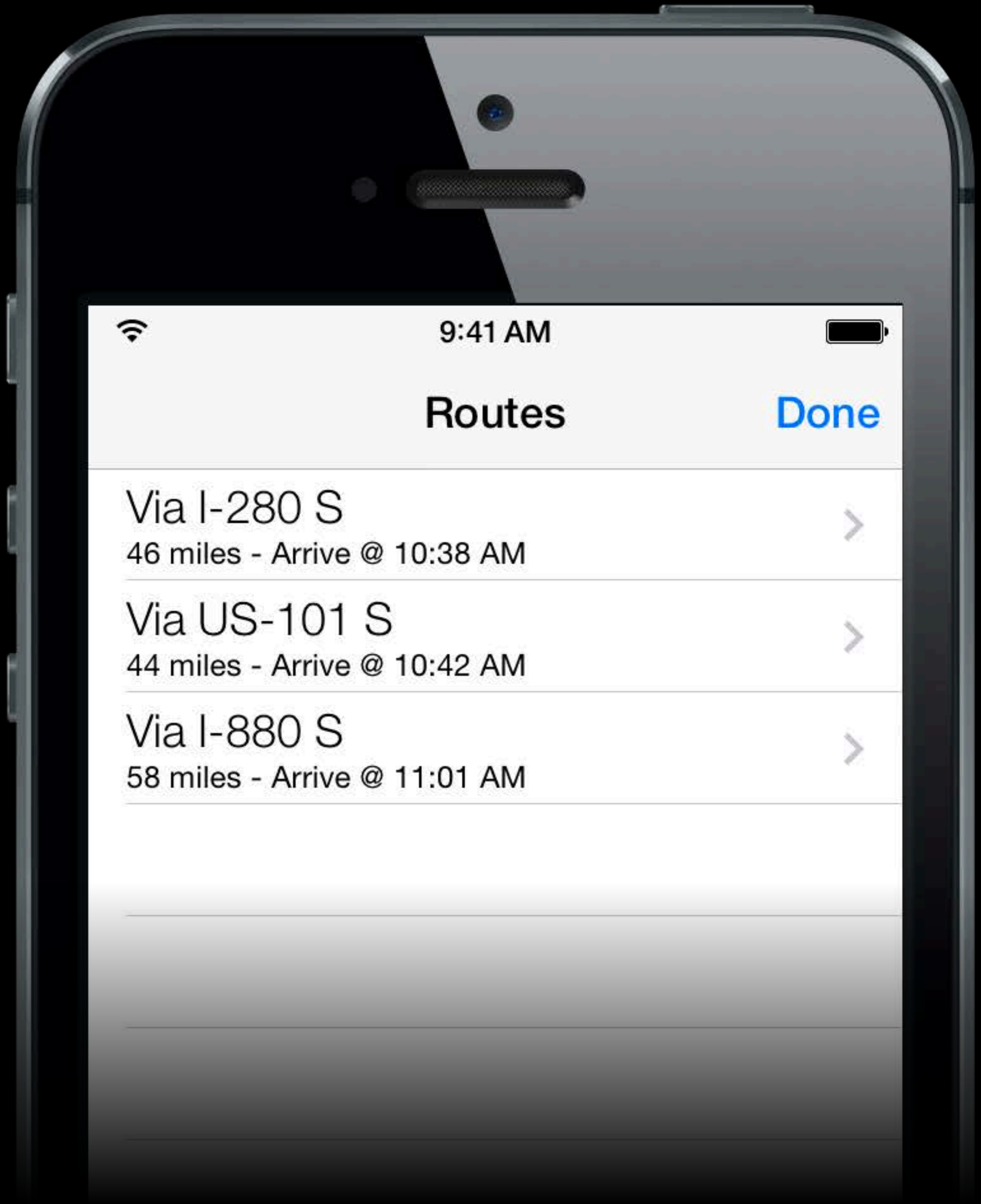
Directions

Displaying routes

```
- (void)showDirections:(MKDirectionsResponse *)response
{
    self.response = response;

    for (MKRoute *route in self.response.routes) {
      [self.map addOverlay:route.polyline level:MKOverlayLevelAboveRoads];
    }
}
```

Directions

Disambiguating routes



9:41 AM

Routes

Done

Via I-280 S
46 miles - Arrive @ 10:38 AM

Via US-101 S
44 miles - Arrive @ 10:42 AM

Via I-880 S
58 miles - Arrive @ 11:01 AM

Directions

Disambiguating routes

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // ... dequeue or create cell

  // Set main cell text label to route name
  MKRoute *route = self.response.routes[indexPath.row];
  cell.textLabel.text = [NSString stringWithFormat:@"Via %@", route.name];

  // Create formatter for distance
  MKDistanceFormatter *distanceFormat = [[MKDistanceFormatter alloc] init];

  // Set detail cell text label to distance and arrival time
  NSString *distance = [distanceFormat stringFromDistance:route.distance];
  NSString *time = formatTime(route.expectedTravelTime);
  cell.detailTextLabel.text =
    [NSString stringWithFormat:@"%@ - Arrive @ %@", distance, time];

  return cell;
}
```

Directions

Disambiguating routes

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // ... dequeue or create cell

  // Set main cell text label to route name
  MKRoute *route = self.response.routes[indexPath.row];
  cell.textLabel.text = [NSString stringWithFormat:@"Via %@", route.name];

  // Create formatter for distance
  MKDistanceFormatter *distanceFormat = [[MKDistanceFormatter alloc] init];

  // Set detail cell text label to distance and arrival time
  NSString *distance = [distanceFormat stringFromDistance:route.distance];
  NSString *time = formatTime(route.expectedTravelTime);
  cell.detailTextLabel.text =
    [NSString stringWithFormat:@"%@ - Arrive @ %@", distance, time];

  return cell;
}
```

Directions

Disambiguating routes

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // ... dequeue or create cell

  // Set main cell text label to route name
  MKRoute *route = self.response.routes[indexPath.row];
  cell.textLabel.text = [NSString stringWithFormat:@"Via %@", route.name];

  // Create formatter for distance
  MKDistanceFormatter *distanceFormat = [[MKDistanceFormatter alloc] init];

  // Set detail cell text label to distance and arrival time
  NSString *distance = [distanceFormat stringFromDistance:route.distance];
  NSString *time = formatTime(route.expectedTravelTime);
  cell.detailTextLabel.text =
    [NSString stringWithFormat:@"%@ - Arrive @ %@", distance, time];

  return cell;
}
```

Directions

Disambiguating routes

```
- (UITableViewCell *)tableView:(UITableView *)tableView
         cellForRowAtIndexPath:(NSIndexPath *)indexPath
{
  // ... dequeue or create cell

  // Set main cell text label to route name
  MKRoute *route = self.response.routes[indexPath.row];
  cell.textLabel.text = [NSString stringWithFormat:@"Via %@", route.name];

  // Create formatter for distance
  MKDistanceFormatter *distanceFormat = [[MKDistanceFormatter alloc] init];

  // Set detail cell text label to distance and arrival time
  NSString *distance = [distanceFormat stringFromDistance:route.distance];
  NSString *time = formatTime(route.expectedTravelTime);
  cell.detailTextLabel.text =
    [NSString stringWithFormat:@"%@ - Arrive @ %@", distance, time];

  return cell;
}
```

Demo

Adding walking directions

Adding Walking Directions

Lessons learned

- Directions API can be used to augment your routing app’s data
- MKRoute can be shown on MKMapView

Directions and Search Are Server Based

- Require a network connection to operate
- Don’t cache results for too long
  - More up to date results may be available

Directions and Search Usage Limits

Directions and Search Usage Limits

- No application or developer identifier wide usage limits
  - Popular apps will not be throttled for being popular

Directions and Search Usage Limits

- No application or developer identifier wide usage limits
    - Popular apps will not be throttled for being popular
- Some per device request throttling
    - Very high limits
    - Intended only to restrain buggy apps
    - Should not affect normal operation

Conclusion

Conclusion

Conclusion

- Many New Features

Conclusion

- Many New Features
  - Map Kit on OS X

Conclusion

- Many New Features
  - Map Kit on OS X
  - iOS 7 UI Updates

Conclusion

- Many New Features
    - Map Kit on OS X
    - iOS 7 UI Updates
    - 3D Maps and Spanning the 180th Meridian

Conclusion

- Many New Features
  - Map Kit on OS X
  - iOS 7 UI Updates
  - 3D Maps and Spanning the 180th Meridian
  - New Overlays Functionality

Conclusion

- Many New Features
    - Map Kit on OS X
    - iOS 7 UI Updates
    - 3D Maps and Spanning the 180th Meridian
    - New Overlays Functionality
    - Search API

Conclusion

- Many New Features
  - Map Kit on OS X
  - iOS 7 UI Updates
  - 3D Maps and Spanning the 180th Meridian
  - New Overlays Functionality
  - Search API
  - Directions API

Conclusion

- Many New Features
  - Map Kit on OS X
  - iOS 7 UI Updates
  - 3D Maps and Spanning the 180th Meridian
  - New Overlays Functionality
  - Search API
  - Directions API
- Try Them Out!

More Information

Paul Marcos
Application Services Evangelist
pmarcos@apple.com

Documentation

MKMapView
http://developer.apple.com/library/ios/#documentation/MapKit/Reference/MKMapView_Class/

Location Awareness Programming Guide
http://developer.apple.com/library/ios/#DOCUMENTATION/UserExperience/Conceptual/LocationAwarenessPG/

Apple Developer Forums
http://devforums.apple.com

Related Sessions

Putting Map Kit in Perspective	Pacific Heights Thursday 2:00 PM

Labs

Map Kit Lab	Services Lab A Thursday 10:15AM	🟩
Map Kit Lab	Services Lab A Thursday 3:15PM	🟩

WWDC2013