Customizing Maps with Overlays

James Howard
Software Engineer Map Kit Team

Agenda

- Map Kit Review
- What's new in Map Kit
- Demos, Demos, Demos, Demos

Map Kit Review

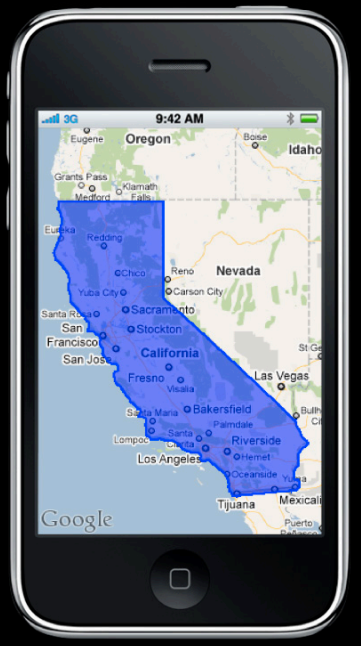
- Originally introduced in iPhone OS 3.0
- Allows you to add a map to your app
  - Uses Google for map data
  - Responds to panning and zooming gestures
  - You can annotate the map with pins or custom annotations
- Reverse geocoding

Map Kit Review

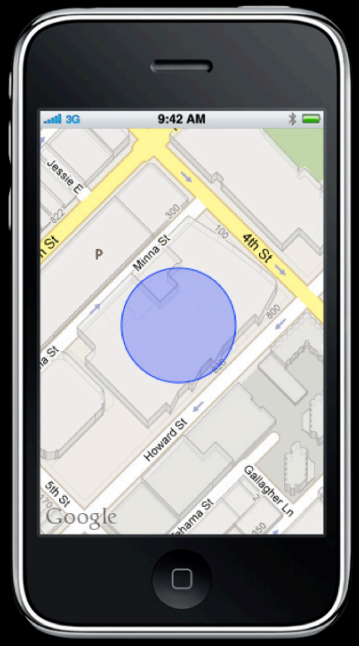
What's New in Map Kit

- Overlays
- Draggable Annotations
- New delegate messages

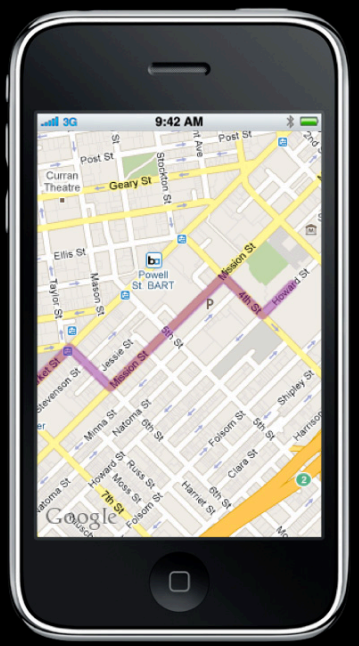
Annotations vs. Overlays



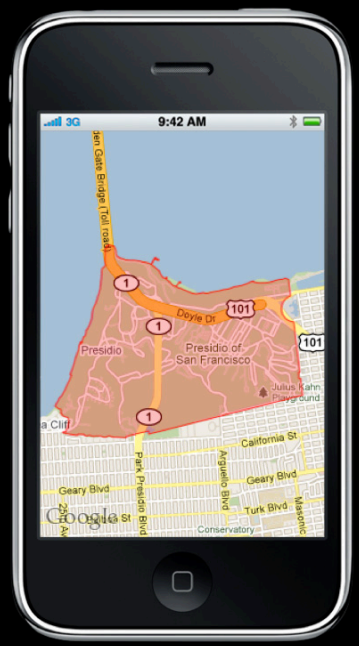
Meet the Overlays



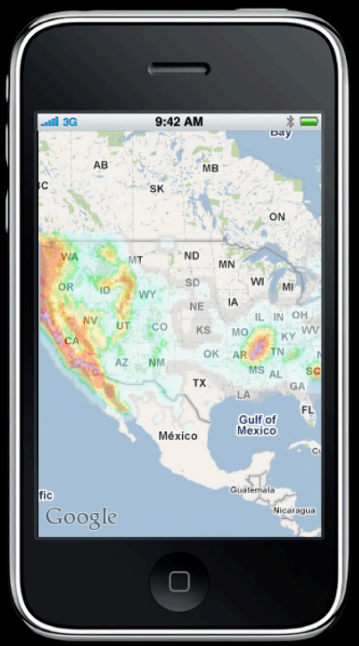
MKCircle



MKPolyline



MKPolygon



Custom Overlay

Map View Layer Ordering

2. Annotation views
1. Overlay views
0. Base map

Adding Overlays to the Map

```
- (void)viewDidLoad
{
    CLLocationCoordinate2D center = CLLocationCoordinate2DMake(37.784, -122.400);
    MKCircle *circle = [MKCircle circleWithCenterCoordinate:center radius:200];
    [map addOverlay:circle];
}
```

```
- (MKOverlayView *)mapView:(MKMapView *)map viewForOverlay:(id <MKOverlay>)overlay
{
    MKCircleView *circleView = [[MKCircleView alloc] initWithOverlay:overlay];
    circleView.strokeColor = [UIColor redColor];
    circleView.fillColor = [[UIColor redColor] colorWithAlphaComponent:0.4];
    return [circleView autorelease];
}
```

Demo

Using built in overlay classes

Process for Adding Overlays

- Add an overlay model object to the map
- Provide the corresponding view from the delegate

Custom Overlays

- MKOverlay (model object)

```
@property (nonatomic, readonly) MKMapRect boundingMapRect;
```

- MKOverlayView

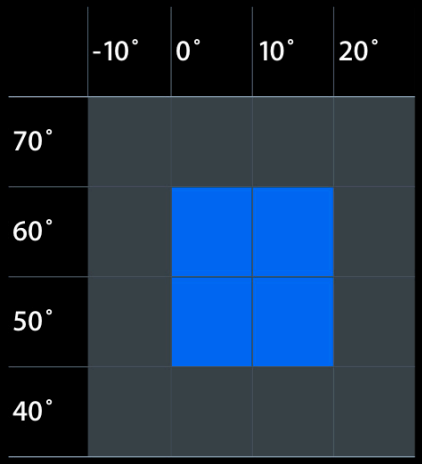
```
- (void)drawMapRect:(MKMapRect)mapRect
          zoomScale:(MKZoomScale)zoomScale
          inContext:(CGContextRef)context;
```

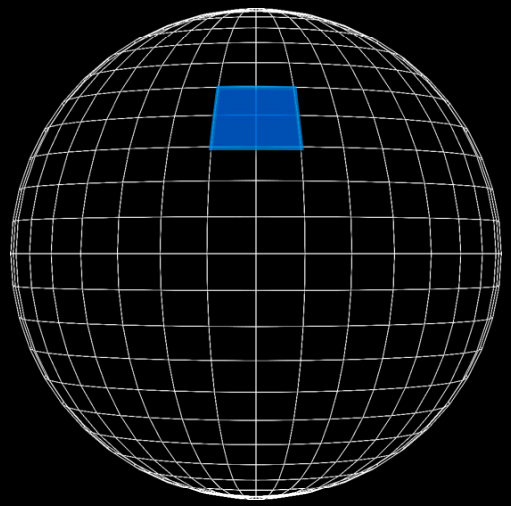
Map Projection

MKMapPoint

- Overlays must be drawn using projected coordinates
- MKMapPoint can represent any point on the map and is linearly proportional to screen points
  - Screen Point = MKMapPoint • MKZoomScale
- Use MKMapPointForCoordinate to convert from latitude/longitude to MKMapPoints

Gridded Data





Demo

Projecting data in a custom overlay

Subclassing MKOverlayView

- Drawing is asynchronous
  - Drawing code must be thread-safe
  - Model data in MKOverlay must either be protected with a lock or be immutable
- To use UIKit drawing functions, use UIGraphicsPushContext
- Drawn tiles are automatically cached
  - Use setNeedsDisplayInMapRect: sparingly to maximize use of cache

Mutable Overlays

- All of the built in Map Kit overlays are immutable
- To get mutability, build your own custom overlay
- Only update the part of the map that has changed
- Protect overlay model data with a lock

Demo

Mutable overlay

Mutable Polyline

- Use a single custom overlay rather than remove/re-add new MKPolyline
- Protect list of points with a read-write lock
- Simplify and clip list of points before creating CGPath

Raster Map Overlays

- Allows you to overlay your own map imagery on top of MKMapView
- Images must be warped to match the Mercator projection used by Map Kit
- Images should be cut into tiles at multiple levels of detail
- Tiles can be loaded either from within your application's bundle or asynchronously over the network

Tiling a Raster Image

Zoom Level 10

Image source: NOAA, http://www.nauticalcharts.noaa.gov/

Creating Raster Tiles from an Image

1. Install Geospatial Data Abstraction Library [http://www.gdal.org]
2. Get information about your image using gdalinfo
3. Create a vrt description of your image using gdal_translate
   - If your image is not already in a geographically annotated format such as GeoTIFF, you will need to assign a spatial reference system and list at least 4 control points
4. Generate image tiles using gdal2tiles.py -p mercator image.vrt

Demo

Raster Map Overlays

Lessons Learned with Tile Map

- Image must be cut into tiles with power of two zoom levels
- You can convert MKZoomScale to zoom level
- Map Kit min/max zoom level is independent from your tiles's min/max zoom level
- Map Kit tile size is independent from your tiles's size

Load Your Tiles on Demand

- -[MKOverlayView canDrawMapRect:zoomScale:] → tile needed
- Return NO from -[MKOverlayView canDrawMapRect:zoomScale:] unless you already have the tile
  - -[MKOverlay setNeedsDisplayInMapRect:zoomScale:] when you get the tile

Load Your Tiles on Demand

- Coalesce tile requests into a fixed size stack
  - Most recent request is the most important
  - Old requests may no longer be visible
  - Stack size should be roughly 2x the number of tiles needed to cover the screen
  - When a request falls off the back of the stack, call -[MKOverlay setNeedsDisplayInMapRect:zoomScale:]

More Information

Mark Malone
Integration Technologies Evangelist
mgm@apple.com

Documentation
Map Kit Framework Reference
http://developer.apple.com/iphone/library/documentation/MapKit/Reference/MapKit_Framework_Reference/

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

Labs

Map Kit Lab	Application Frameworks Lab B Thursday 2:00PM – 4:15PM

Summary

- Built in Map Kit overlays let you add lines and shapes to your map
- Custom overlays let you draw arbitrary content atop the map

Q&A


WWDC10