Dissecting the “Kit” in MapKit

Justin Miller • Development Seed / MapBox • @*incanus77*

CocoaConf PDX • October 26, 2012 • Portland, OR

Introduction

- I've been programming about 15 years
  - I've been using Cocoa about 8 years
- Work for Development Seed on MapBox
  - User & developer tools for custom maps
  - I work mostly on iOS developer tools

Overview

- I'm not here to teach you MapKit
  - Goal is to provide general overview
    - Then: replacement using Cocoa concepts
  - Come back to this later if you need to
    - I'll post slides & sample code later

MapKit

- Released with iOS 3 in 2009
- Originally partnered with Google
- Most of current functionality at debut
  - iOS 4 added shape overlays
  - iOS 5 added user tracking & rotation
  - iOS 6 updated cosmetics & data source

MapBox SDK

- Originally based on 2008 Route-Me project
- Predated original MapKit
- Refactored in Alpstein fork in 2011
- Forked into our SDK in early 2012
- I started hacking on Route-Me in 2010
- Entirely open source (BSD license)

Project Stats

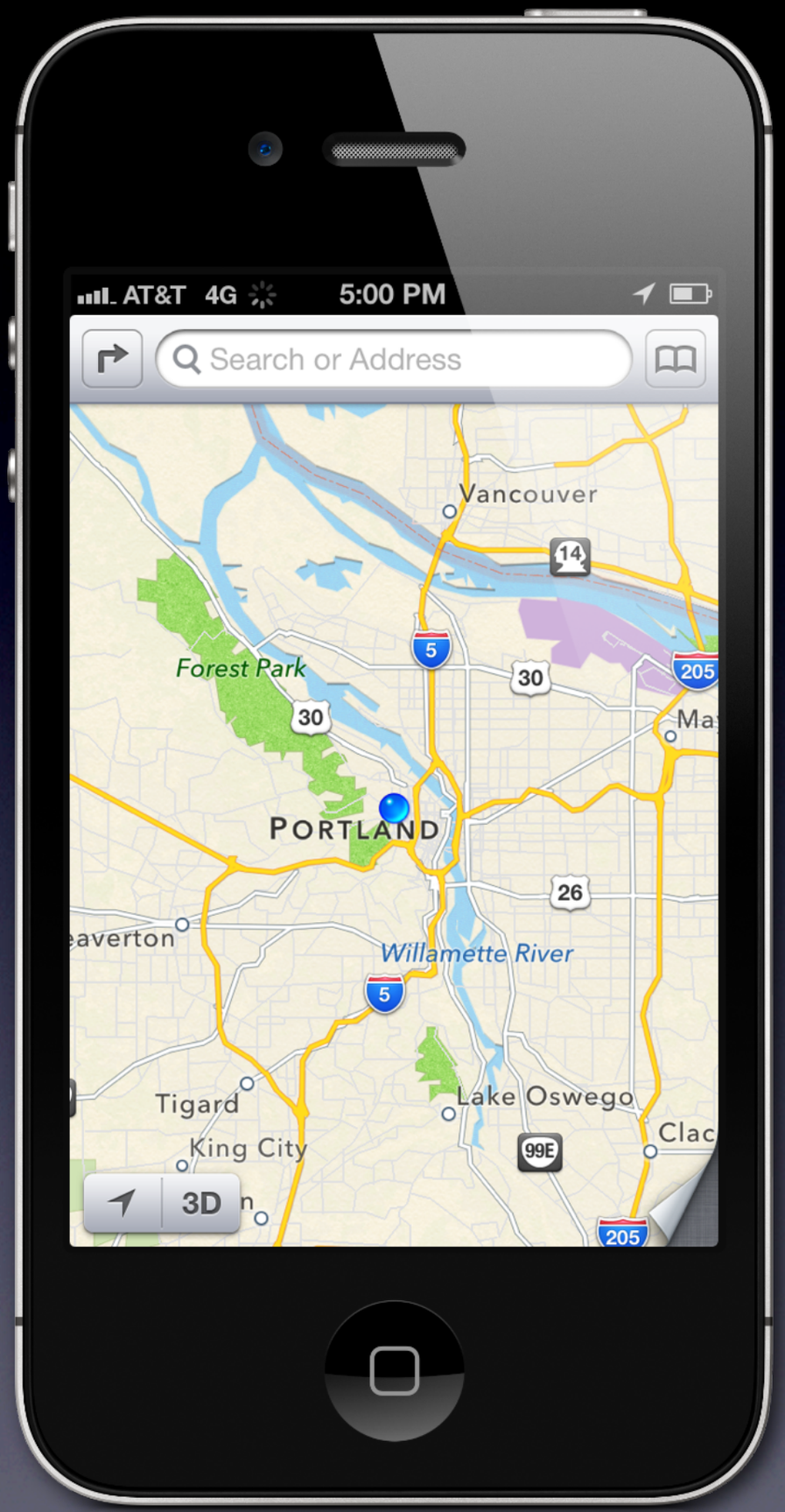
- About 15,000 lines of code
- Our fork has about 140 GitHub tickets
- About 35 forks of our project
- About 200 “stars”
- Currently at about 50 classes
  - Many of these are “private”

MapKit Parts

- Show Apple's, then ours in detail
- Three main components:

1. Map view: main UI element & container
2. Annotations: point & shape overlays
3. User location services: "blue dot"

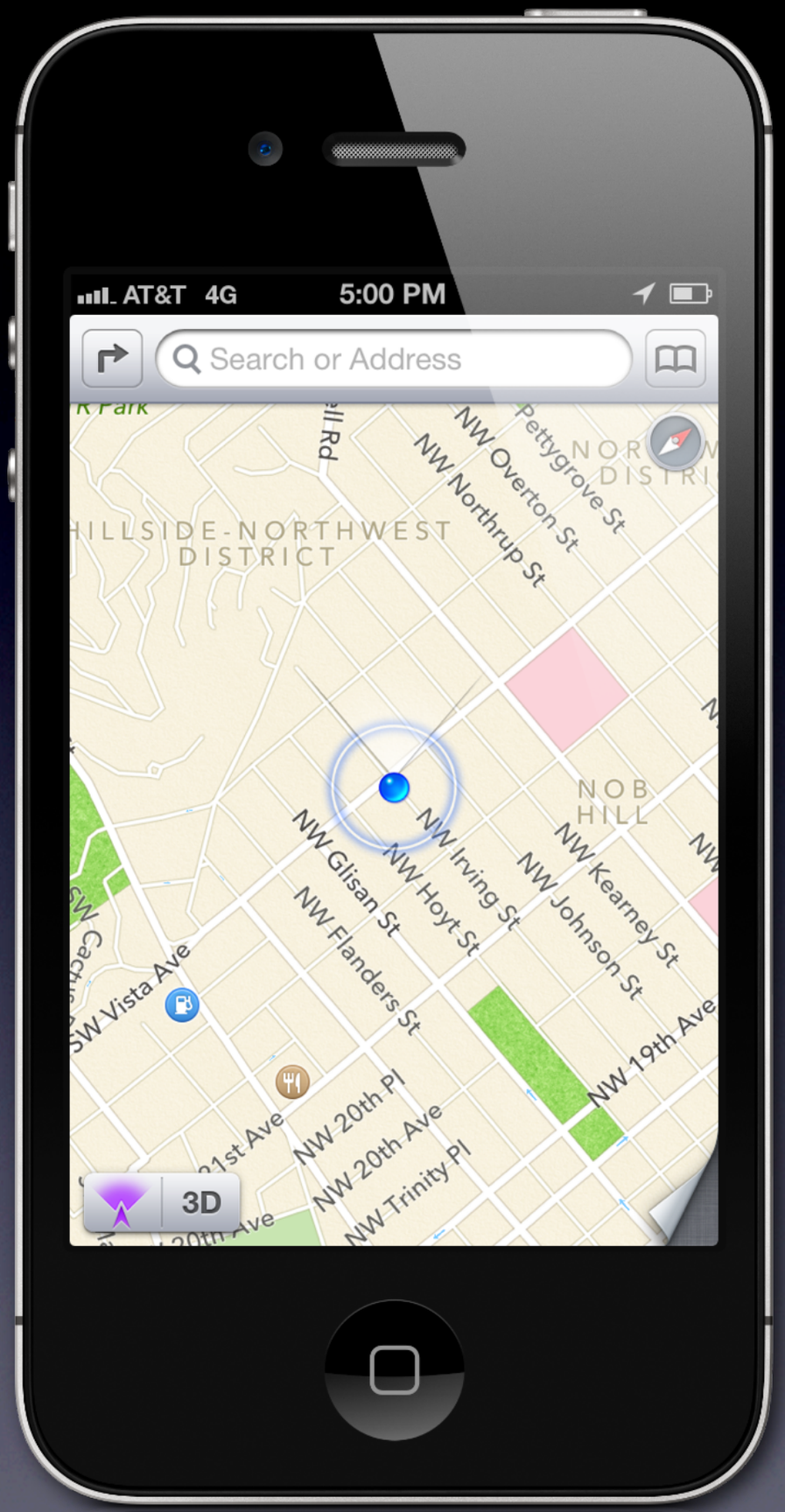
1. Map View


AT&T 4G
5:00 PM
Search or Address
Vancouver
14
5
Forest Park
30
30
205
PORTLAND
26
Willamette River
5
Tigard
King City
Lake Oswego
99E
3D
205

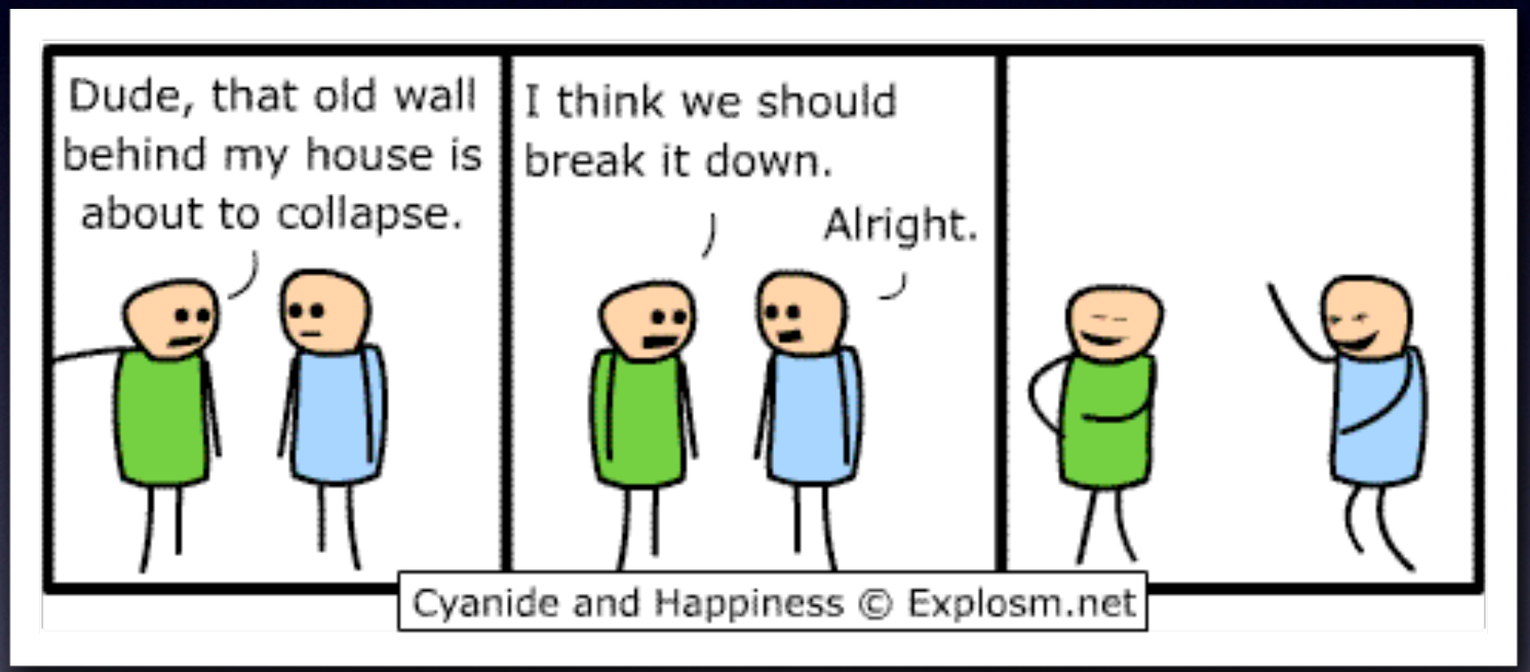
2. Annotations


AT&T
8:04 PM
Embassy suites
Lemon Island
Embassy Suites Portland...
30 reviews on Yelp
Portland International Airport
NE Airport Way
NE Cascades Pkwy
NE Alderwood Rd
213
Columbia Slough
30
30
30
3D

3. User Location Services


AT&T 4G
5:00 PM
Search or Address
HILLSIDE-NORTHWEST DISTRICT
NOB HILL
NW Overton St
NW Northrup St
Pettygrove St
NW Irving St
NW Hoyt St
NW Glisan St
NW Flanders St
NW Johnson St
NW Kearney St
NW 19th Ave
NW 20th Pl
NW 20th Ave
NW Trinity Pl
SW Vista Ave
3D

Break It Down



1. Map View

The Map View

- What is it?
  - Panning, zooming, tiled view
- How does it work?
  - `UIScrollView` as main component
  - `CATiledLayer`-backed content view(s)
  - Geographic controls affecting the content
  - Some extra gestures

Map Zoom Levels

- Zoom 0
  - One tile covers the whole world
  - Content view is 256px on a side
- Zoom Z
  - 4^Z tiles cover the whole world
  - Content view is $2^Z * 256$px on a side

CATiledLayer Use

- Every `UIView` has a `CALayer`
- Override `+[UIView layerClass]`
  - Return `[CATiledLayer class]`
- `-drawLayer:inContext:` method
  - Delegate is the owning view

CATiledLayer Drawing

- Context is your "drawing scratch space"
- Query the context for offset & bounds
- Combine this with the zoom scale
- Fetch & draw a tile to the context

Tiling Demo

Coordinates & Offsets

- Latitude/longitude e.g. (0, 0)
  - Center of square map
  - Fractionally the same at each zoom level
    - e.g. 50% over, 50% down
  - Convert between lat/long & these fractions

Offsets Demo

Map Gestures

- Pan & zoom handled by `UIScrollView`
- Extra gestures handled individually
  - Single- and double-tap
  - Two-finger single-tap
  - Additional pan for marker dragging
    - With hit testing before failing

2. Annotations

Annotations

- Simple array of annotation data objects
  - Think table cells (drawn on demand)
- Assign layer upfront or query map delegate
- Add layer to screen when visible
- Remove layer & `nil` out when off screen

Visibility & Tracking

- We use Key Value Observing (KVO)
- `-[UIScrollView contentOffset]`
- More accurate results than delegate
  - Not before or after, but *during*

Annotation Layers

- Placed on an overlay view above tiled view
- Simple points made with **`RMMarker`**
  - Image via **`-[CALayer contents]`**
- Vector shapes made with **`RMShape`**
  - Each has a **`CAShapeLayer`**

Annotation Panning

- Overlay layer gets moved with the map
- Again, via KVO on the content offset
- Extra pan gesture for marker dragging

Annotation Zooming

- Overlay layer "stretches" with the map
- Layers still "stick" to the right points
  - Positions moved during KVO updates
- Markers don't change size
- Shapes scale themselves during zooms
  - Affine transform, then path animation

Annotation Demo

3. User Location Services

User Location Services

- Tracking “blue dot”
  - Three layers: dot, accuracy circle, halo
    - Halo gets layer animation for “pulse”
    - Placement: Core Location coordinate
- Map orientation & rotation
  - Angle: Core Location heading

BTW:

github.com/0xced/UIKit-Artwork-Extractor

Heading Tracking

- Use heading angle to rotate map view
- Apply opposite rotation to markers
  - Allows them to stay upright
- All changes are animated smoothly

Rotation Demo

Tracking Mode Button

- UIBarButtonItem subclass
  - `MKUserTrackingBarButtonItem`
  - `RMUserTrackingBarButtonItem`
- Associated with a map view
  - KVO for two-way communication
- Animates all state changes

Tracking Demo

3D Mode

- You've seen this in iOS 6 MapKit
- Possible at basic level via simple transforms
- Not going to get into buildings, etc.

Rotation

- `UIRotationGestureRecognizer`
  - Similar effect to compass heading rotation
  - Apply rotation about Z axis

Tilting

- `UIPanGestureRecognizer`
  - Two finger touches
  - Roughly parallel movement
- Apply rotation about X axis

Scroll View Transforms

- Transform around Z from rotation
- Transform around X from tilt
- Combine transforms together
  - `CATransform3DConcat()`

3D Demo

Other Tidbits

- Callouts: *github.com/nfarina/calloutview*
- Compositing: multiple layered tiles
- Tile cache: memory & disk caching
- Documentation: *appledoc*
- Distribution: *CocoaPods*
- Region interactivity: *UTFGrid*

Thank You!

- Twitter/ADN: *@incanus77*
- Email: *justin@mapbox.com*
- Coding: *github.com/incanus*
- SDK: *mapbox.com/mobile*