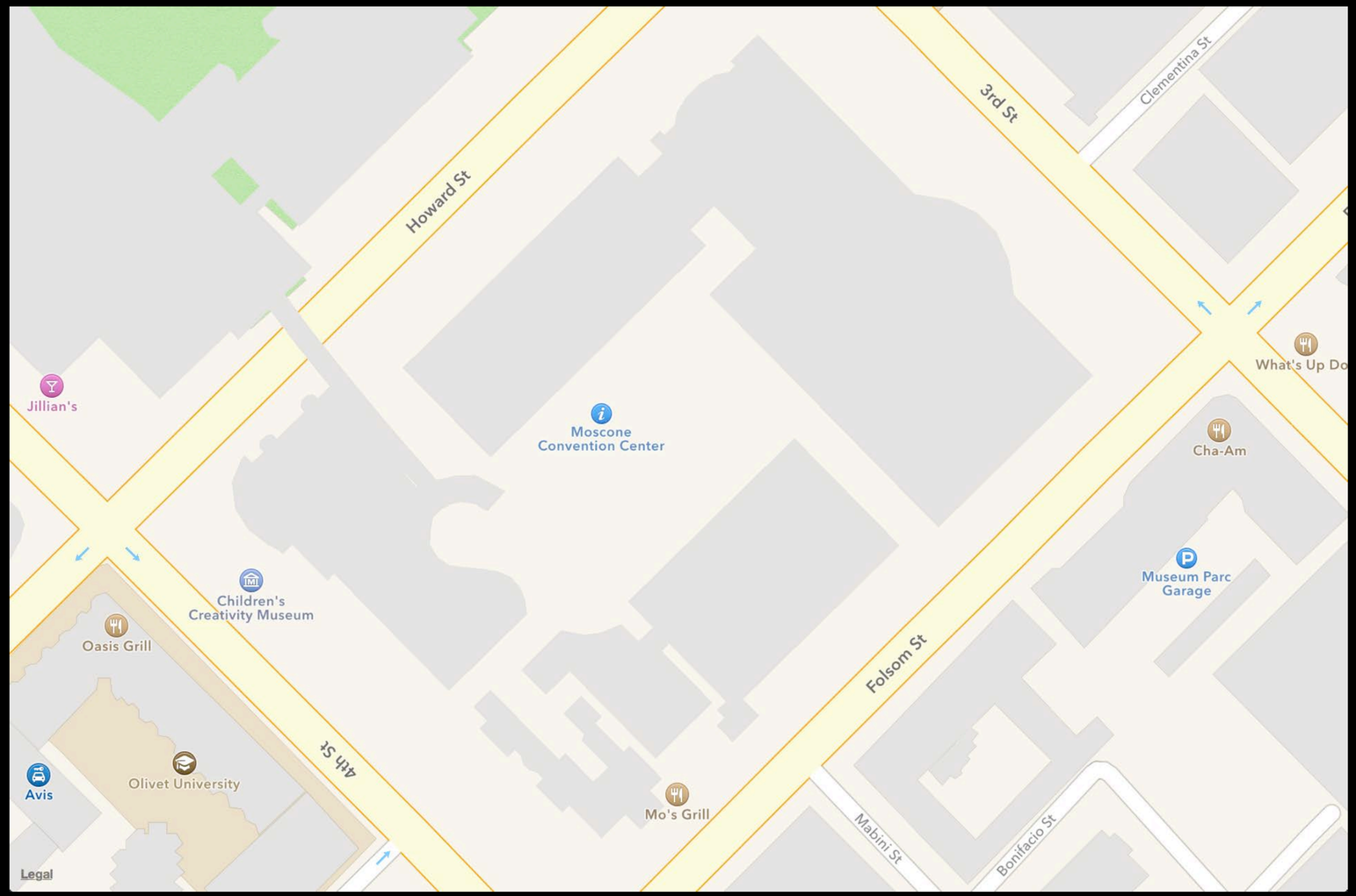
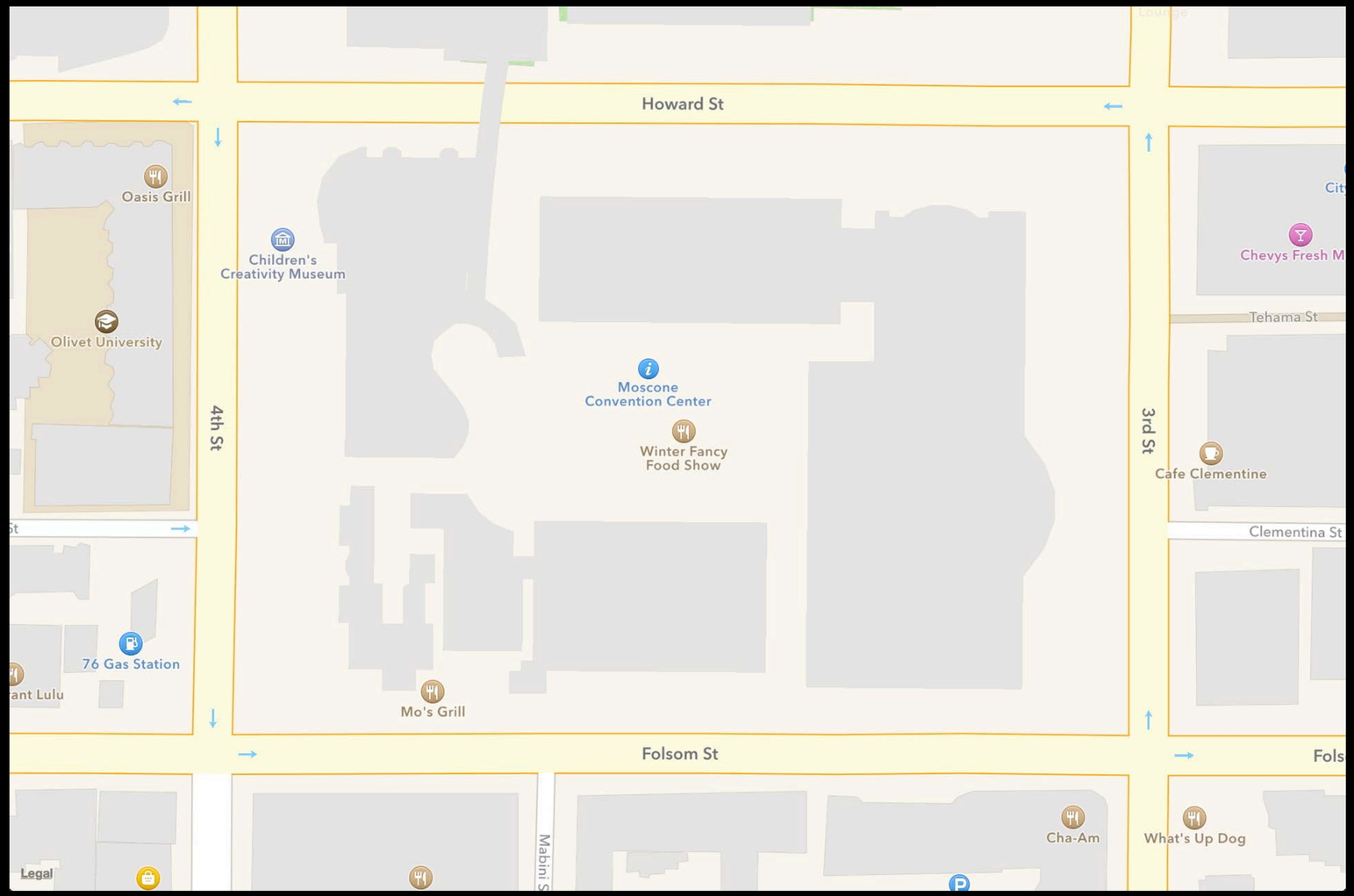
Putting Map Kit in Perspective

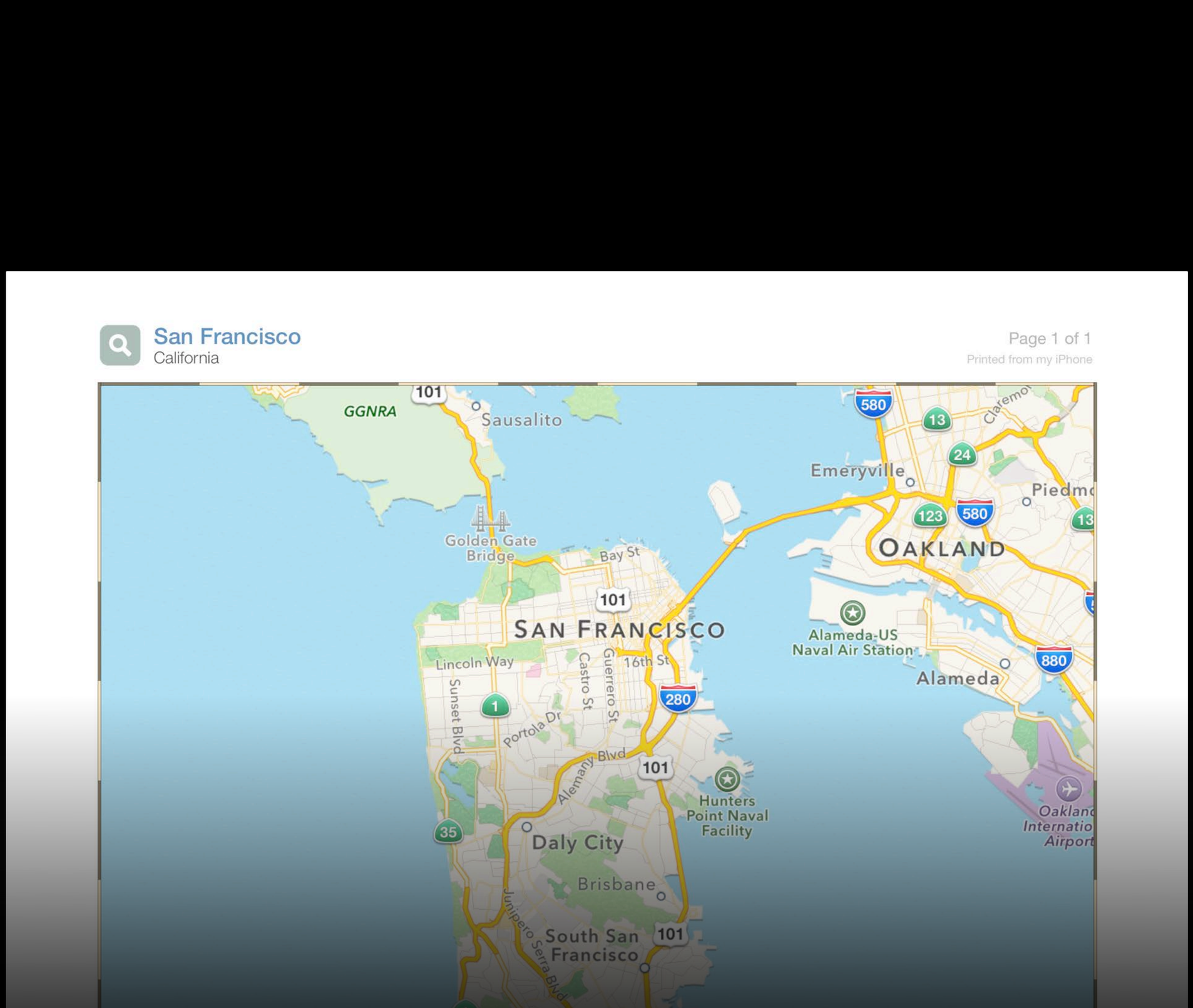
Session 309

Aroon Pahwa
iOS Software Engineer


Howard St
3rd St
Clementina St
Jillian's
Moscone
Convention Center
What's Up Do
Cha-Am
Children's
Creativity Museum
Oasis Grill
Museum Parc
Garage
Folsom St
4th St
Olivet University
Avis
Mo's Grill
Mabini St
Bonifacio St
Legal


Howard St
Oasis Grill
Children's
Creativity Museum
Olivet University
4th St
Moscone
Convention Center
Winter Fancy
Food Show
3rd St
Cafe Clementine
Chevys Fresh M
Tehama St
Clementina St
76 Gas Station
Mo's Grill
Folsom St
Cha-Am
What's Up Dog
Legal


CANADA
Vancouver
Montréal
Toronto
Chicago
UNITED STATES
New York
Washington D.C.
San Francisco
Los Angeles
North Pacific Ocean
North Atlantic Ocean
MEXICO
Mexico City
CUBA
HAITI
HONDURAS
NICARAGUA
PANAMA
Caracas
VENEZUELA
GUYANA
COLOMBIA
ECUADOR
Legal


San Francisco
California
Page 1 of 1
Printed from my iPhone
GGNRA
101
Sausalito
Golden Gate Bridge
Bay St
101
SAN FRANCISCO
Lincoln Way
Sunset Blvd
Castro St
Guerrero St
16th St
280
1
Portola Dr
Alemany Blvd
101
Hunters Point Naval Facility
35
Daly City
Brisbane
Junipero Serra Blvd
South San Francisco
101
580
13
24
Emeryville
123
580
OAKLAND
Alameda-US Naval Air Station
880
Alameda

Now in Map Kit

What Is a 3D Map?

What Is a 3D Map?

Add Perspective to Your Map

What Is a 3D Map?

Add Perspective to Your Map

Create Cinematic Map Transitions

What Is a 3D Map?

Add Perspective to Your Map

Create Cinematic Map Transitions

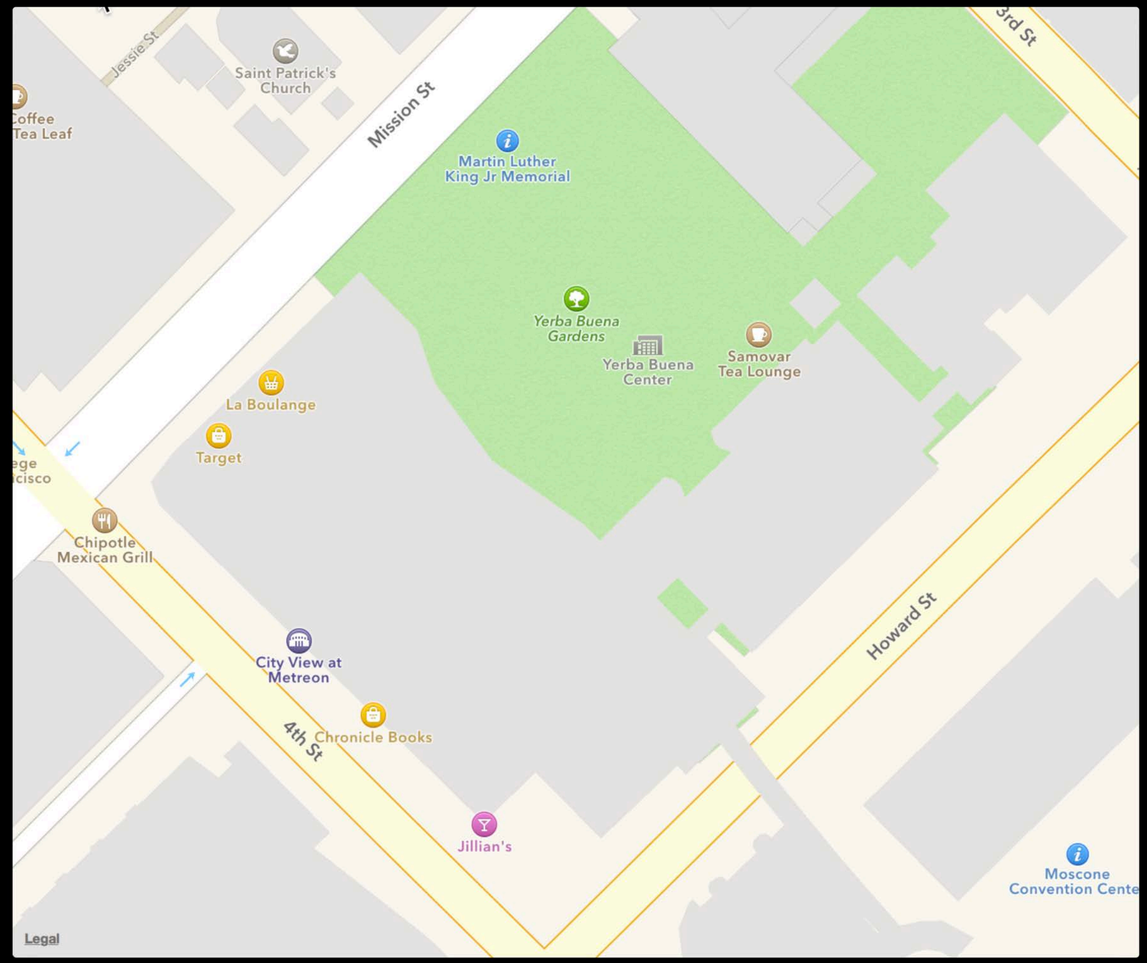
Produce Static Map Snapshots

What Is a 3D Map?

What Is a 3D Map?

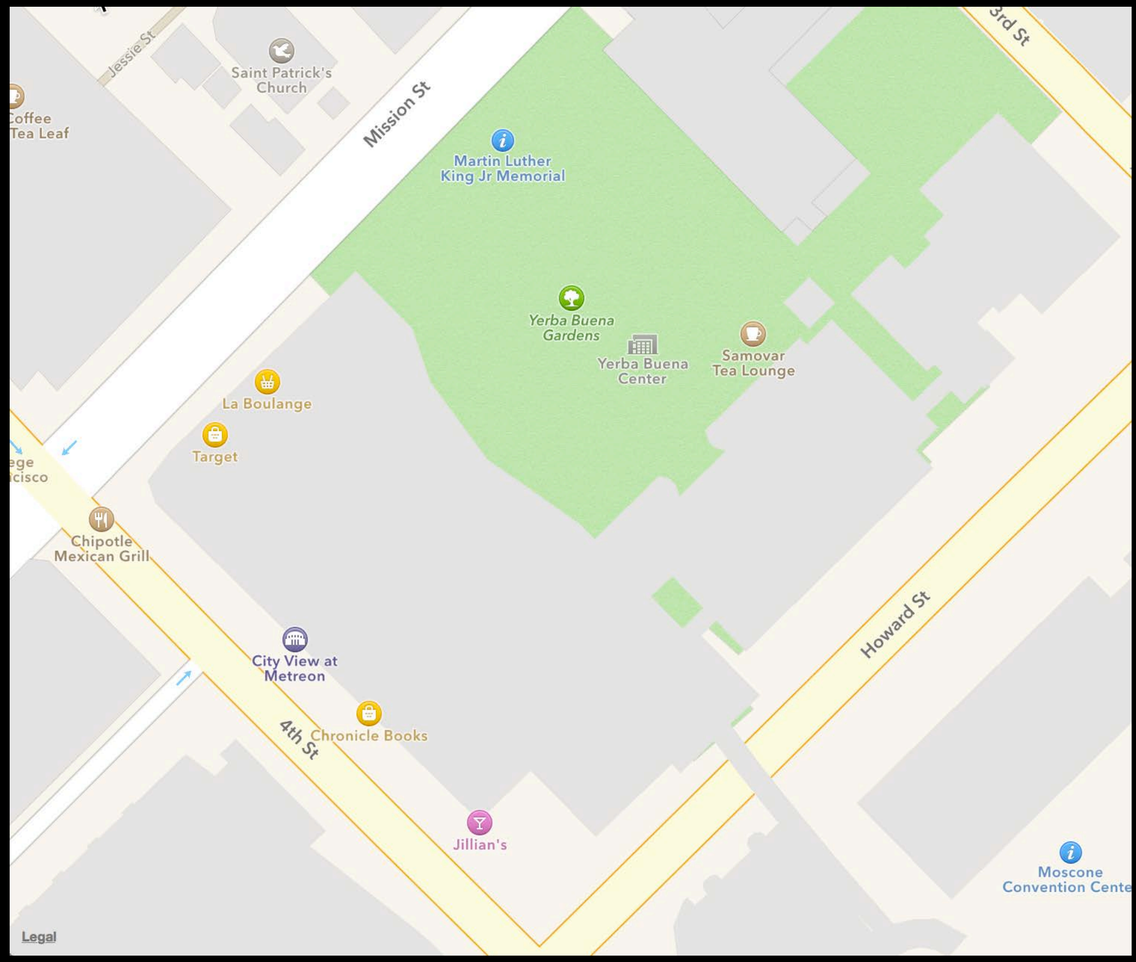
What Is a 3D Map?

- Start with a 2D map



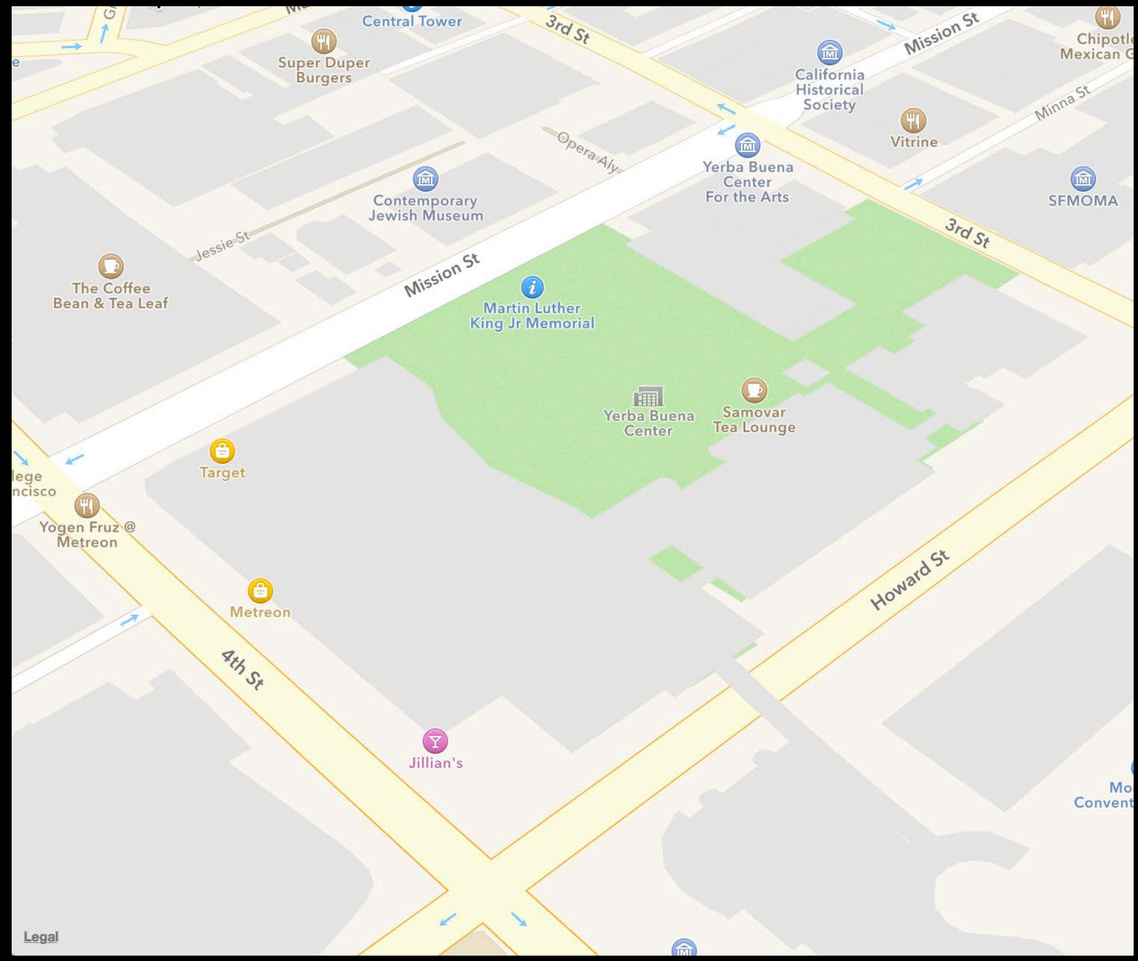
What Is a 3D Map?

- Start with a 2D map
- Add pitched views



What Is a 3D Map?

- Start with a 2D map
- Add pitched views
- Extrude buildings



What Is a 3D Map?

- Start with a 2D map
- Add pitched views
- Extrude buildings
- Sometimes called 2.5D

What Is a 3D Map?

A 2D map in a 3D environment

What Is a 3D Map?

Getting around

What Is a 3D Map?

Getting around

- Just like Maps.app on iOS
  - Two finger rotate rotates the map
  - Two finger vertical pan pitches the map

What Is a 3D Map?

Getting around

- Just like Maps.app on iOS
  - Two finger rotate rotates the map
  - Two finger vertical pan pitches the map
- Just like Maps.app on OS X
  - Trackpad gestures
  - Compass and zoom controls

What Is a 3D Map?

Getting around

- Just like Maps.app on iOS
  - Two finger rotate rotates the map
  - Two finger vertical pan pitches the map
- Just like Maps.app on OS X
  - Trackpad gestures
  - Compass and zoom controls
- Available in the iOS Simulator
  - Rotate—option + drag in a circle
  - Pitch—option + shift + drag vertically

Adding Perspective to Your Map

Adding Perspective to Your Map

What you need to do

- Recompile with iOS 7 SDK
- Adapt to changes in existing API behavior
  - Map display
  - Annotations
  - Overlays
  - Geometry conversions
- Adopt new API to take advantage of 3D
  - Camera

Adapting to Existing API Changes

Map display

- Two APIs
  - `[MKMapView setVisibleMapRect:]`
  - `[MKMapView setRegion:]`
- Produces a 2D map, no rotation
- May span the 180th Meridian

Adapting to Existing API Changes

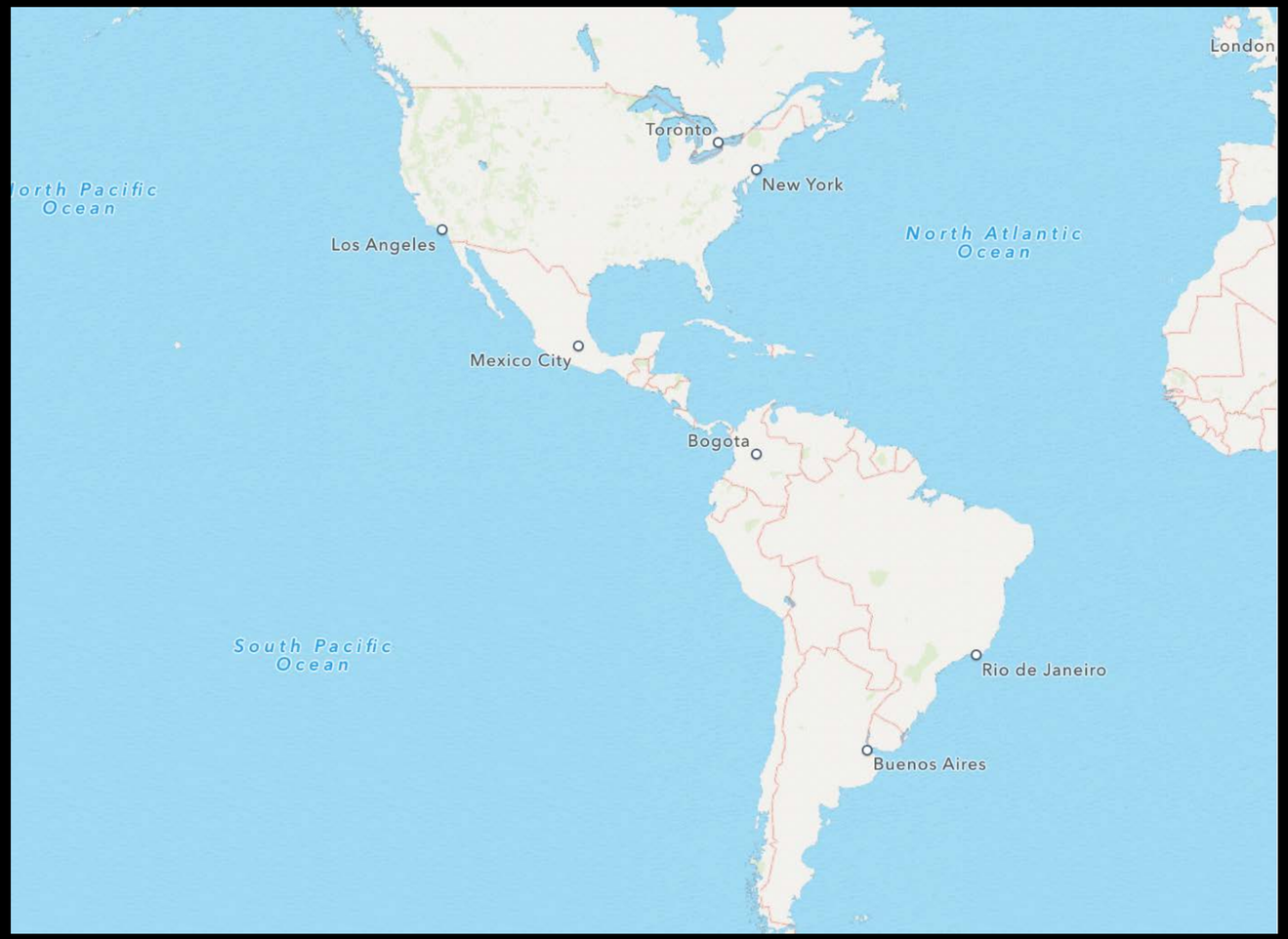
Map display

```
(MKCoordinateRegion) $0 = {
  center = {
    latitude = 0
    longitude = 180
  }
  span = {
    latitudeDelta = 115
    longitudeDelta = 115
  }
}
```

Adapting to Existing API Changes

Map display

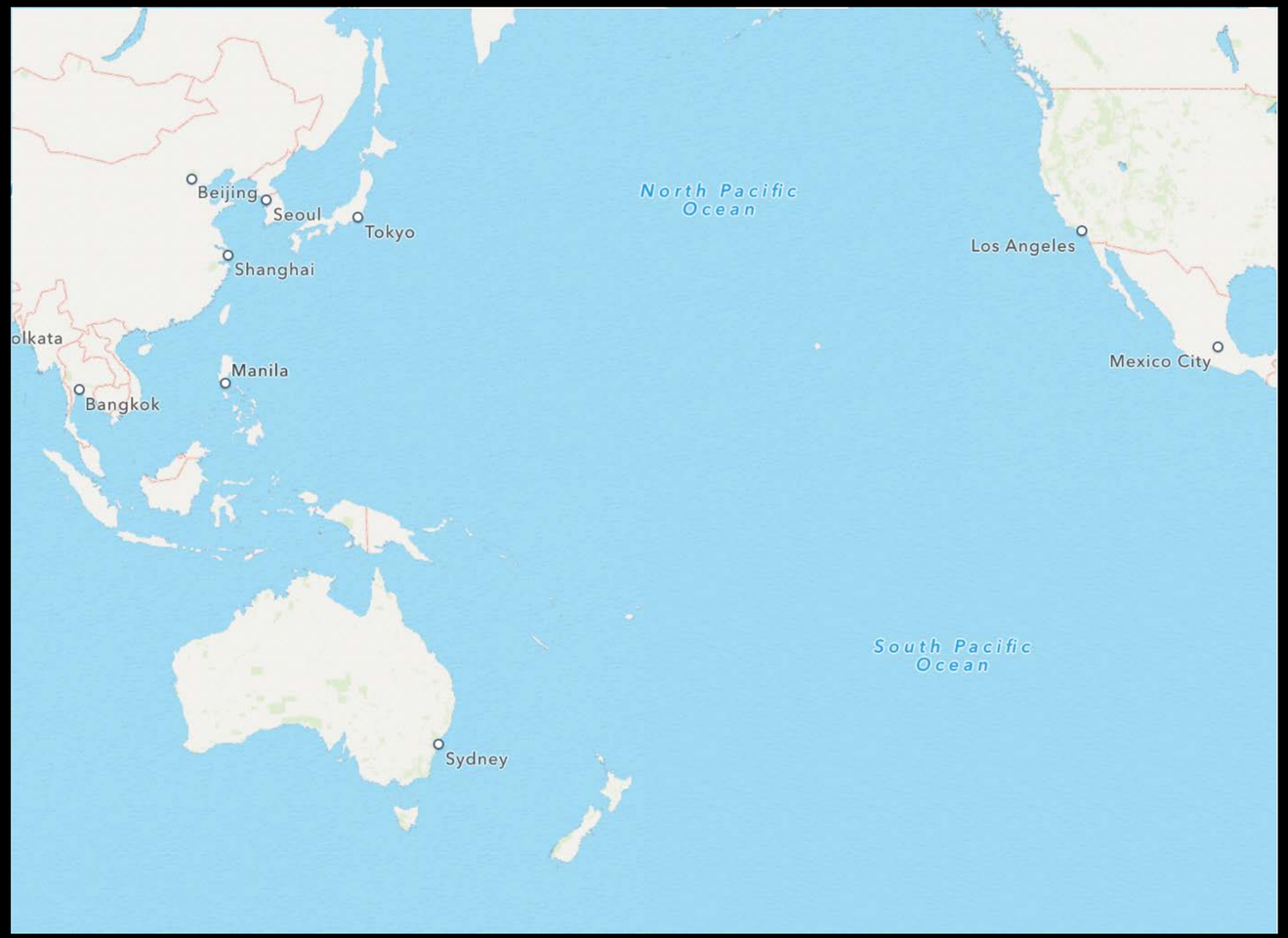
```
(MKCoordinateRegion) $0 = {
  center = {
    latitude = 0
    longitude = 180
  }
  span = {
    latitudeDelta = 115
    longitudeDelta = 115
  }
}
```



Adapting to Existing API Changes

Map display

```
(MKCoordinateRegion) $0 = {
  center = {
    latitude = 0
    longitude = 180
  }
  span = {
    latitudeDelta = 115
    longitudeDelta = 115
  }
}
```



Adapting to Existing API Changes

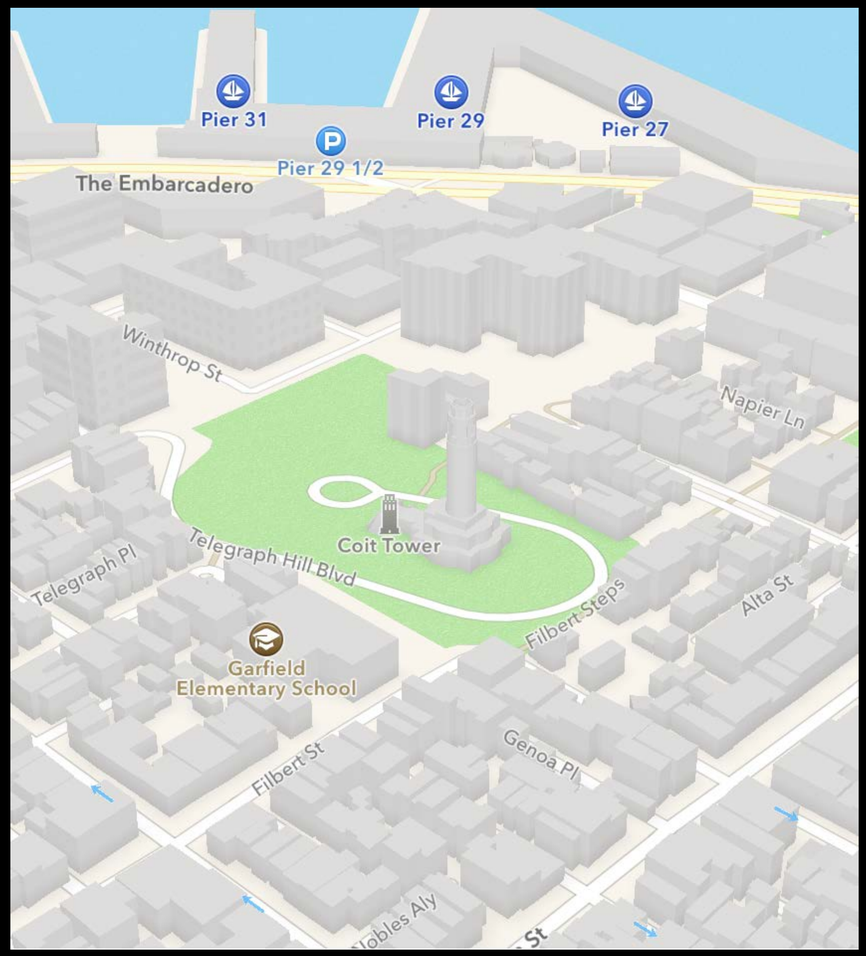
Map display

- Two APIs
  - `[MKMapView visibleMapRect]`
  - `[MKMapView region]`

Adapting to Existing API Changes

Map display

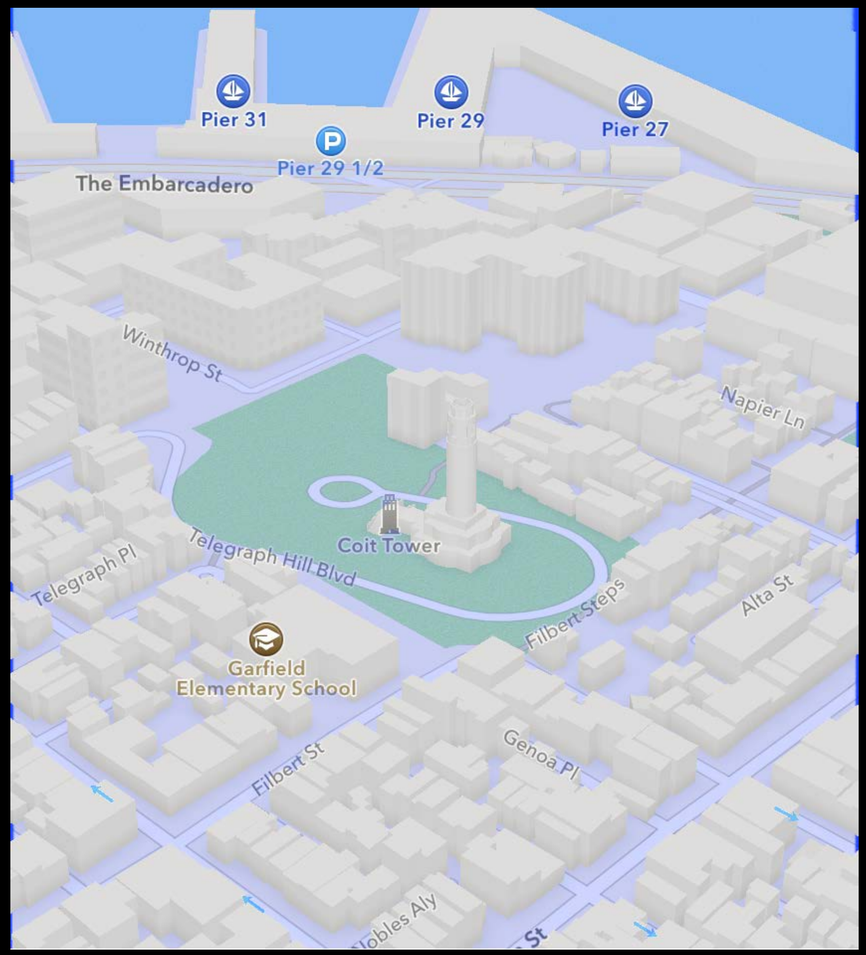
- Two APIs
  - `[MKMapView visibleMapRect]`
  - `[MKMapView region]`



Adapting to Existing API Changes

Map display

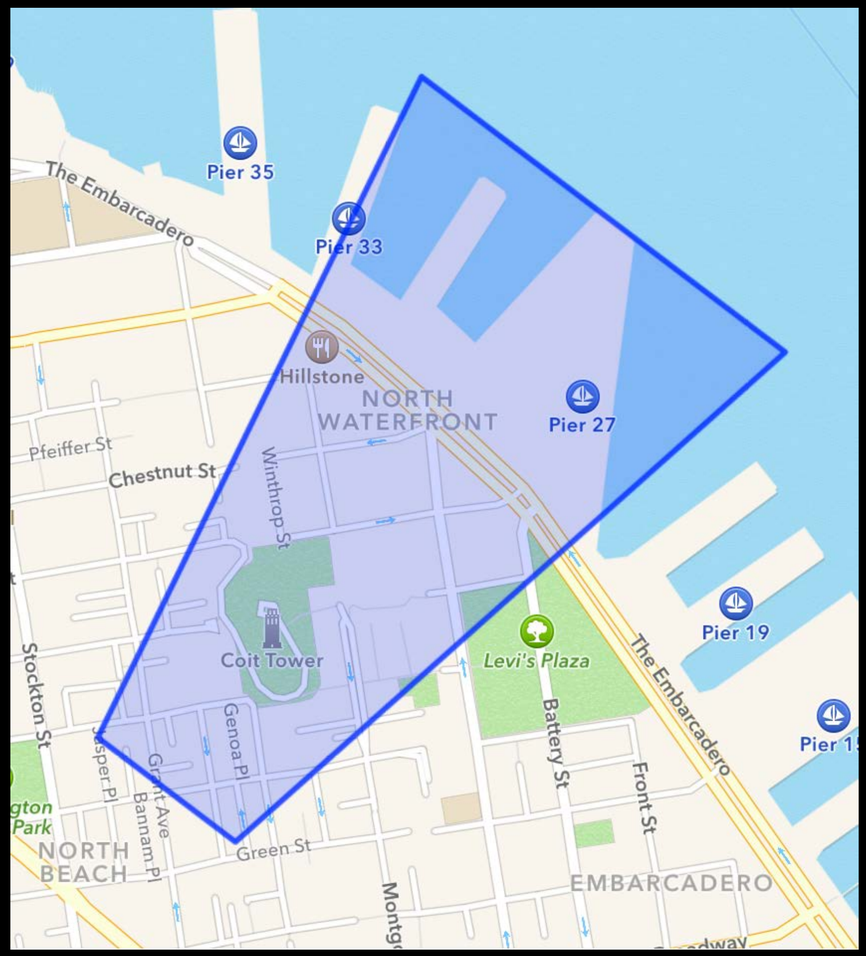
- Two APIs
  - `[MKMapView visibleMapRect]`
  - `[MKMapView region]`



Adapting to Existing API Changes

Map display

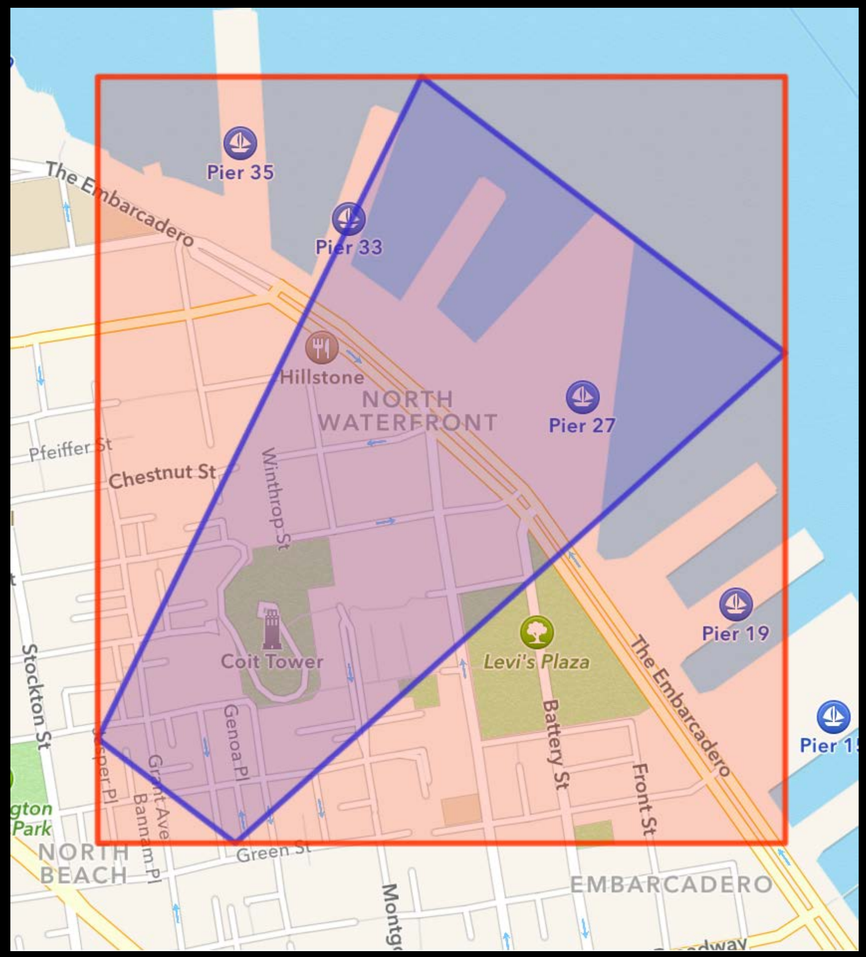
- Two APIs
  - `[MKMapView visibleMapRect]`
  - `[MKMapView region]`



Adapting to Existing API Changes

Map display

- Two APIs
  - `[MKMapView visibleMapRect]`
  - `[MKMapView region]`
- Contains the visible area
- May span the 180th Meridian
- Ideal for culling your data



Adapting to Existing API Changes

Map display

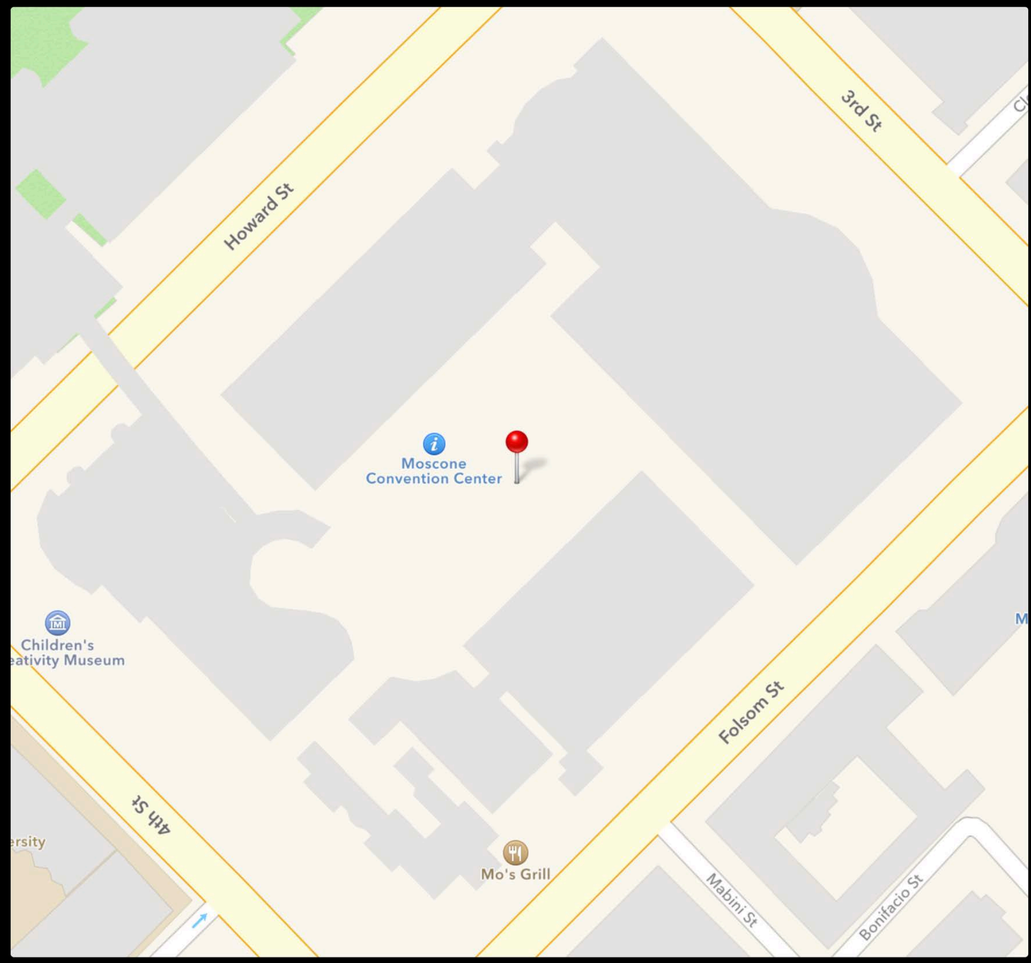
- One more API

```
@property centerCoordinate
```

- Coordinate at center of screen
- Simulates panning

Adapting to Existing API Changes

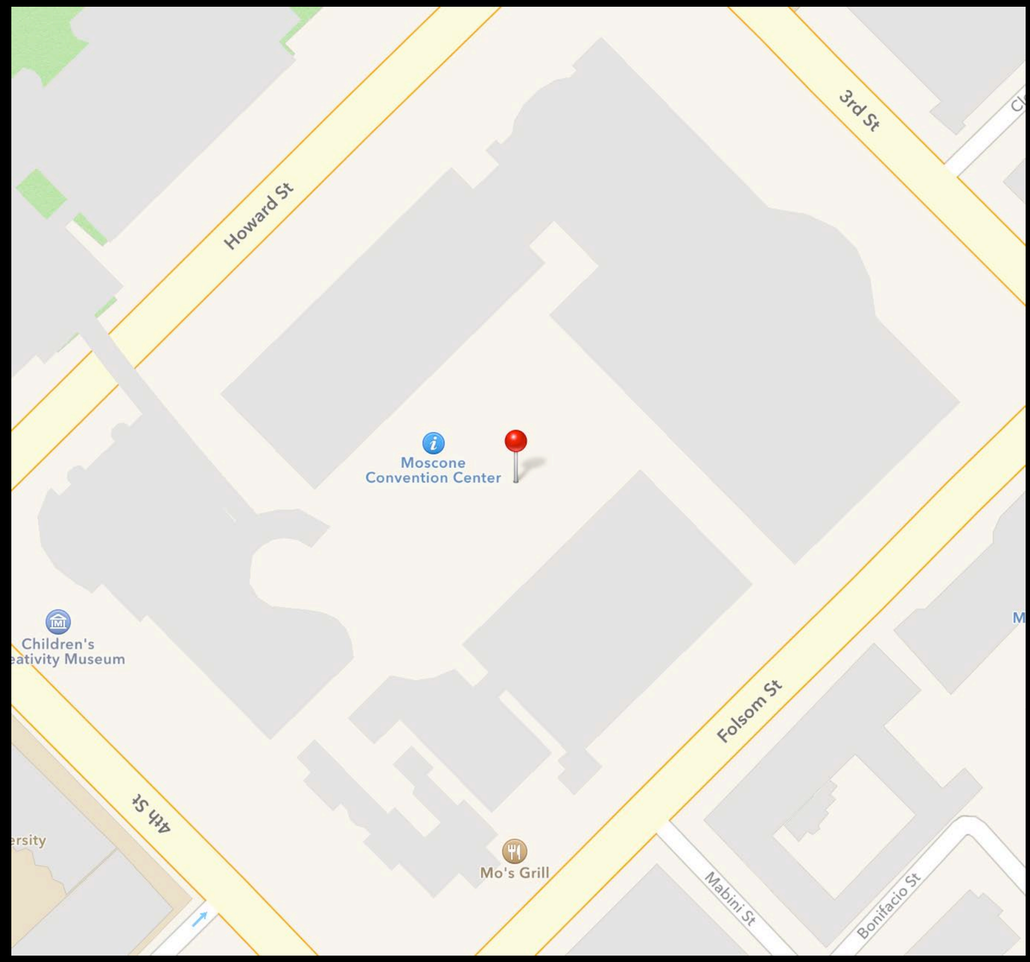
Annotations



Adapting to Existing API Changes

Annotations

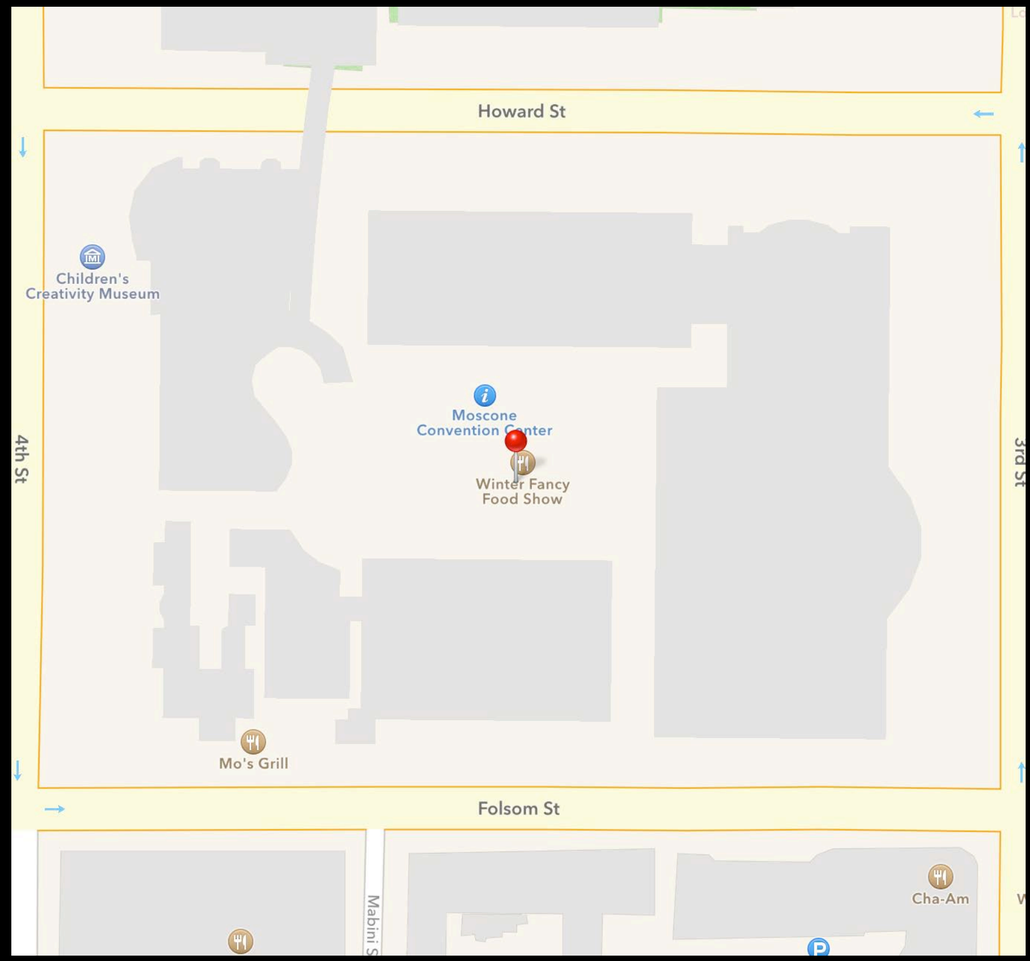
- Always upright



Adapting to Existing API Changes

Annotations

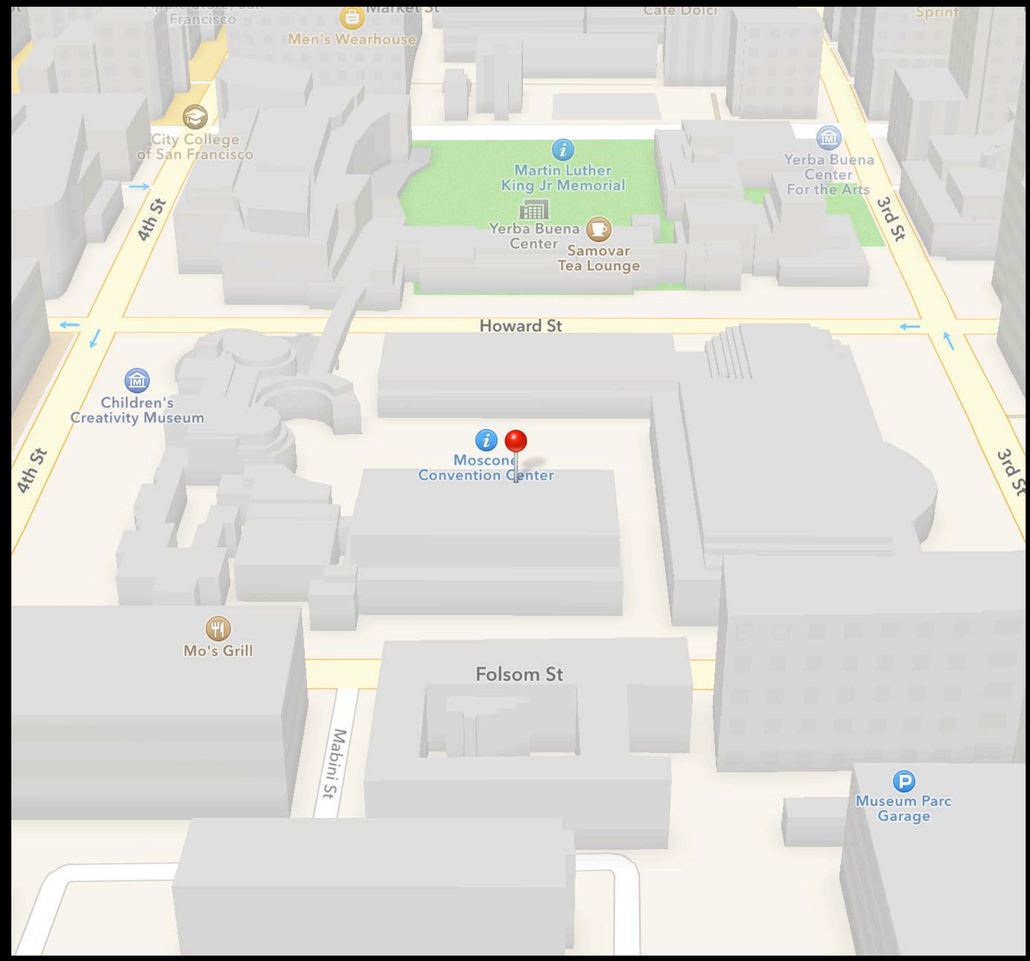
- Always upright
- Always face the screen



Adapting to Existing API Changes

Annotations

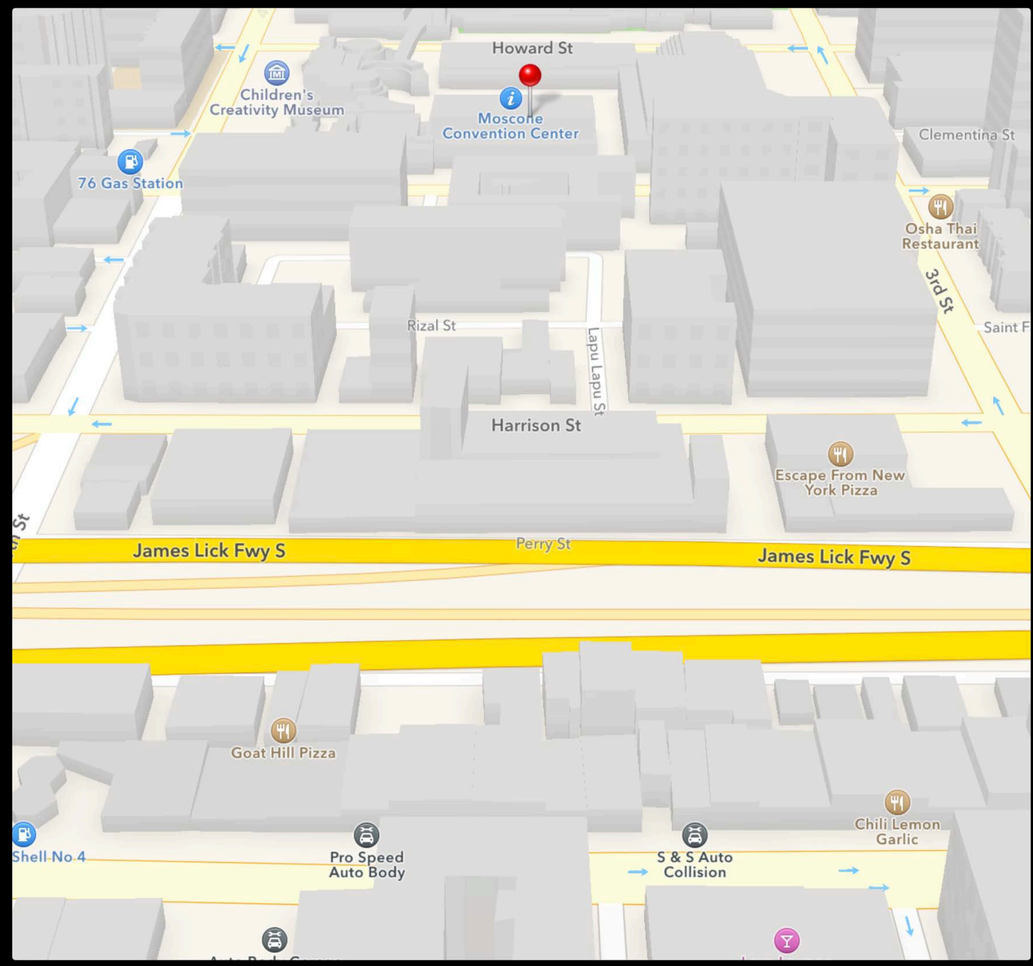
- Always upright
- Always face the screen
- Always the same size



Adapting to Existing API Changes

Annotations

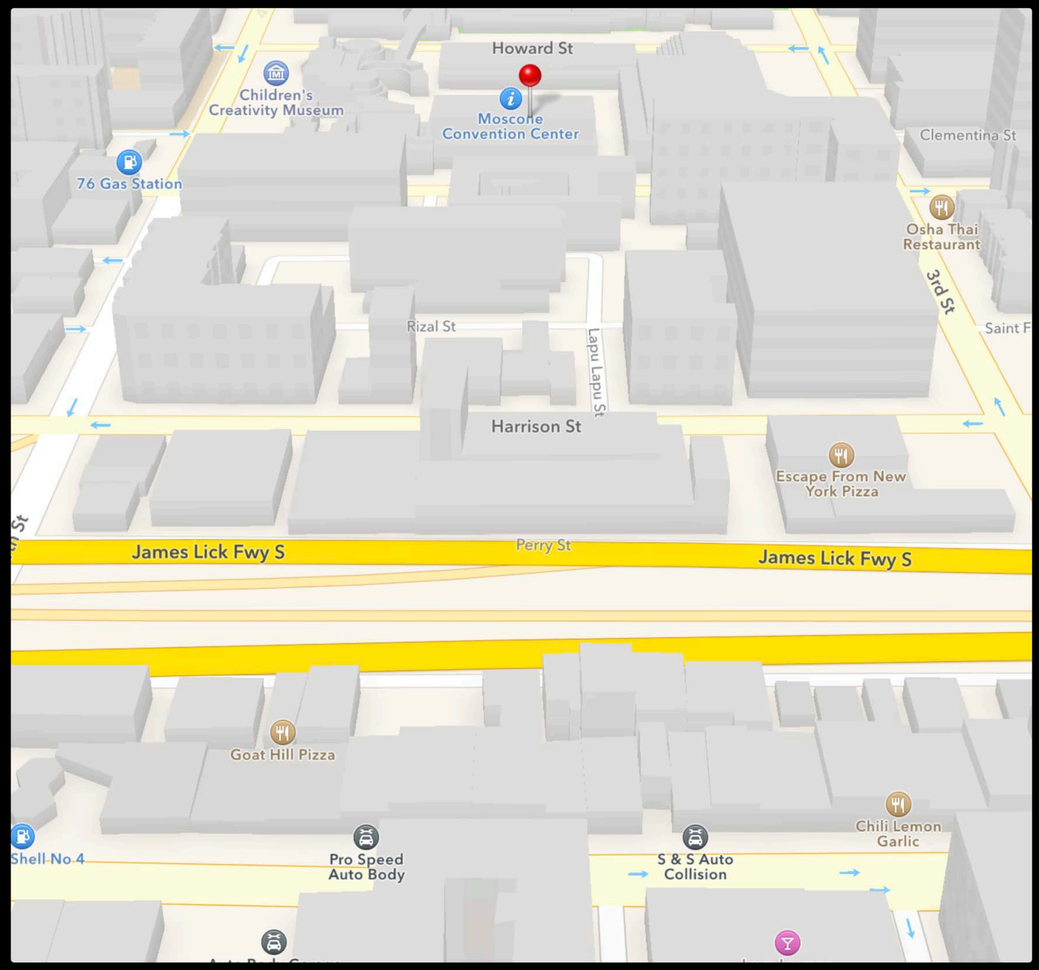
- Always upright
- Always face the screen
- Always the same size
- Always track the map



Adapting to Existing API Changes

Annotations

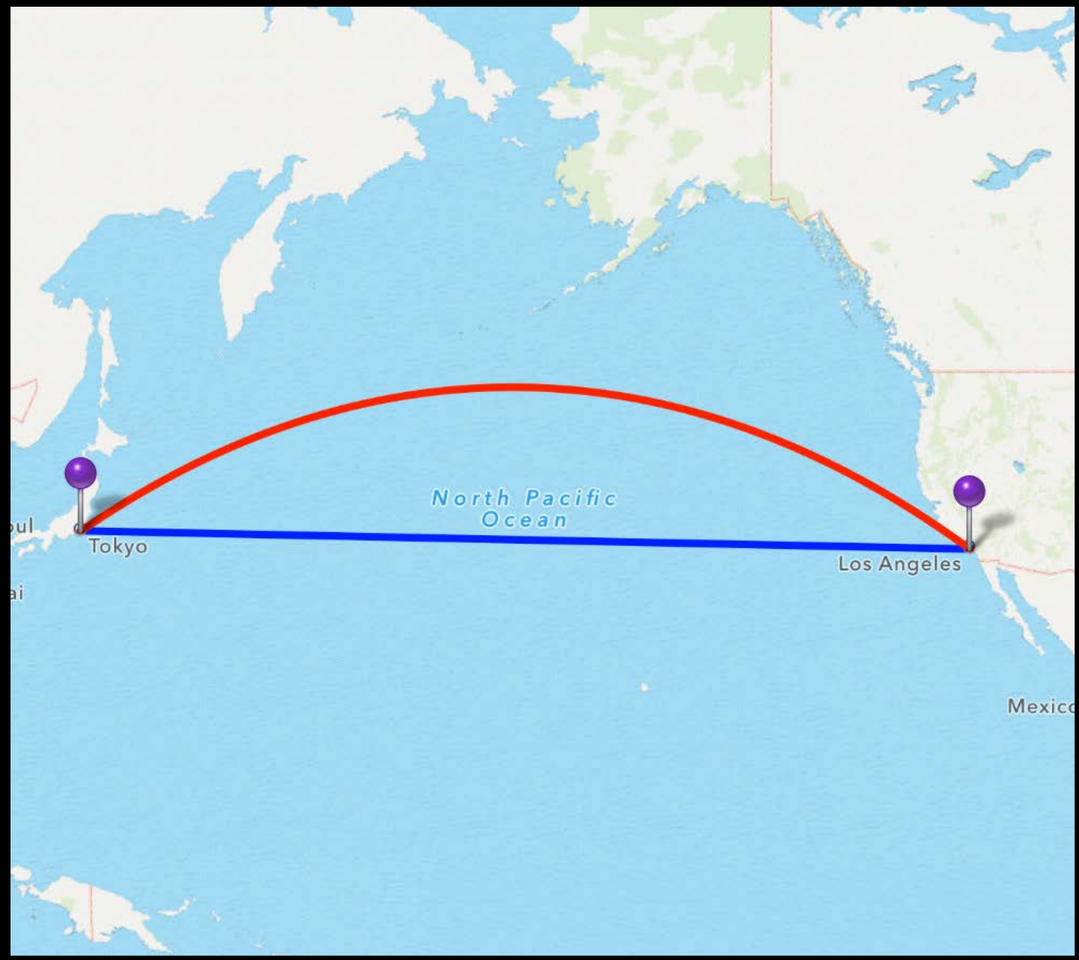
- Always upright
- Always face the screen
- Always the same size
- Always track the map
- iOS—based on UIView
- OS X—based on NSView



Adapting to Existing API Changes

Overlays

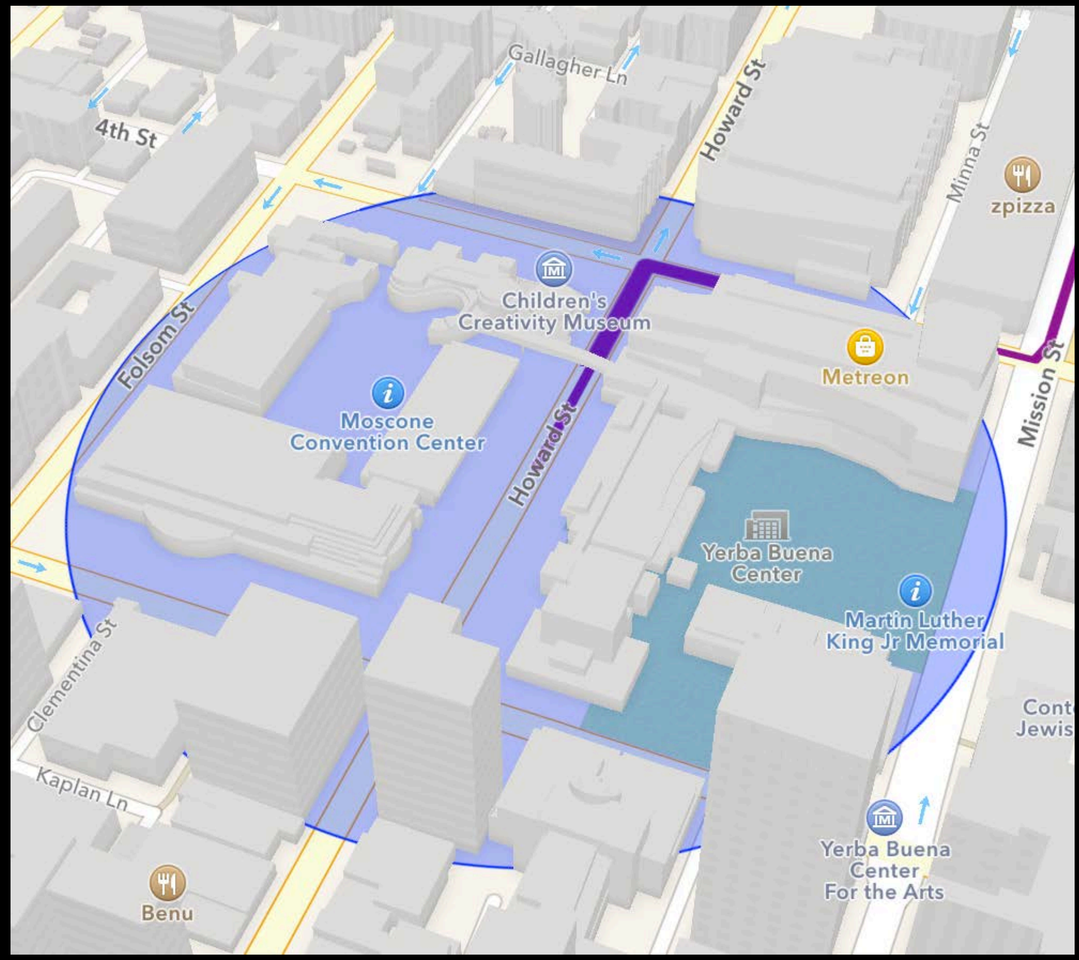
- Polylines take the shortest path across the map



Adapting to Existing API Changes

Overlays

- Polylines take the shortest path across the map
- Occluded by 3D buildings

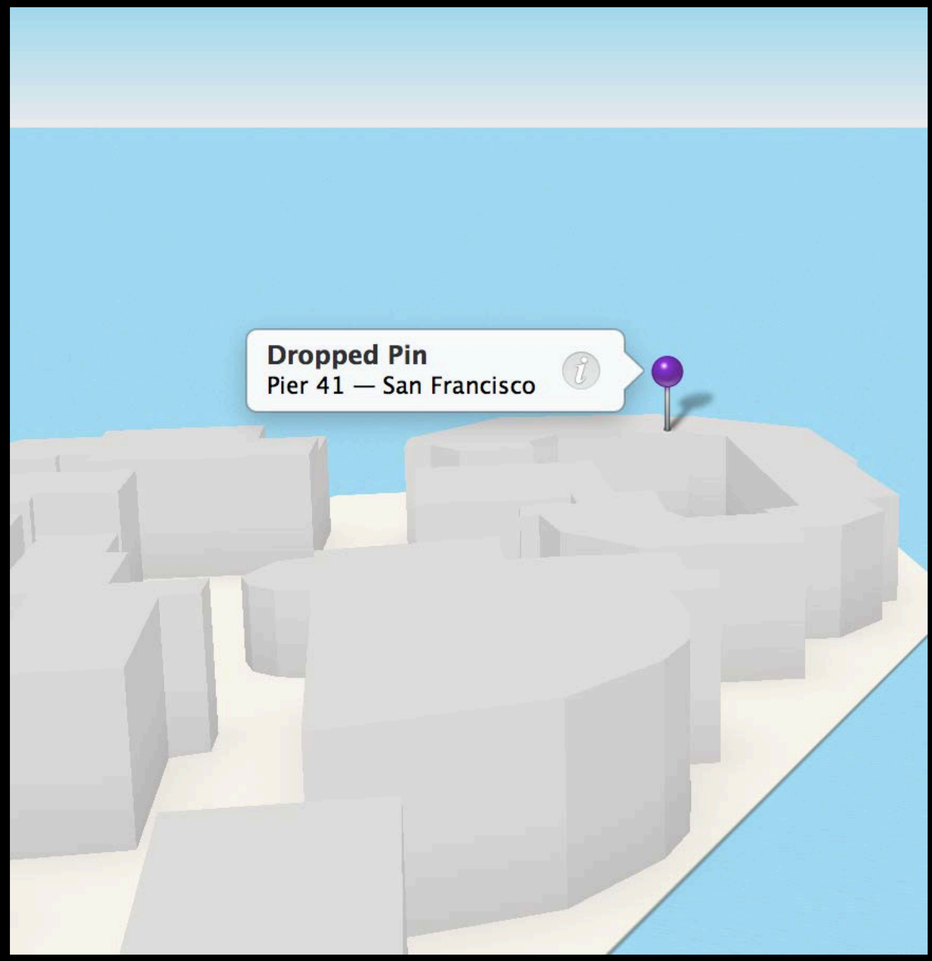


Adapting to Existing API Changes

Geometry conversions

Adapting to Existing API Changes

Geometry conversions

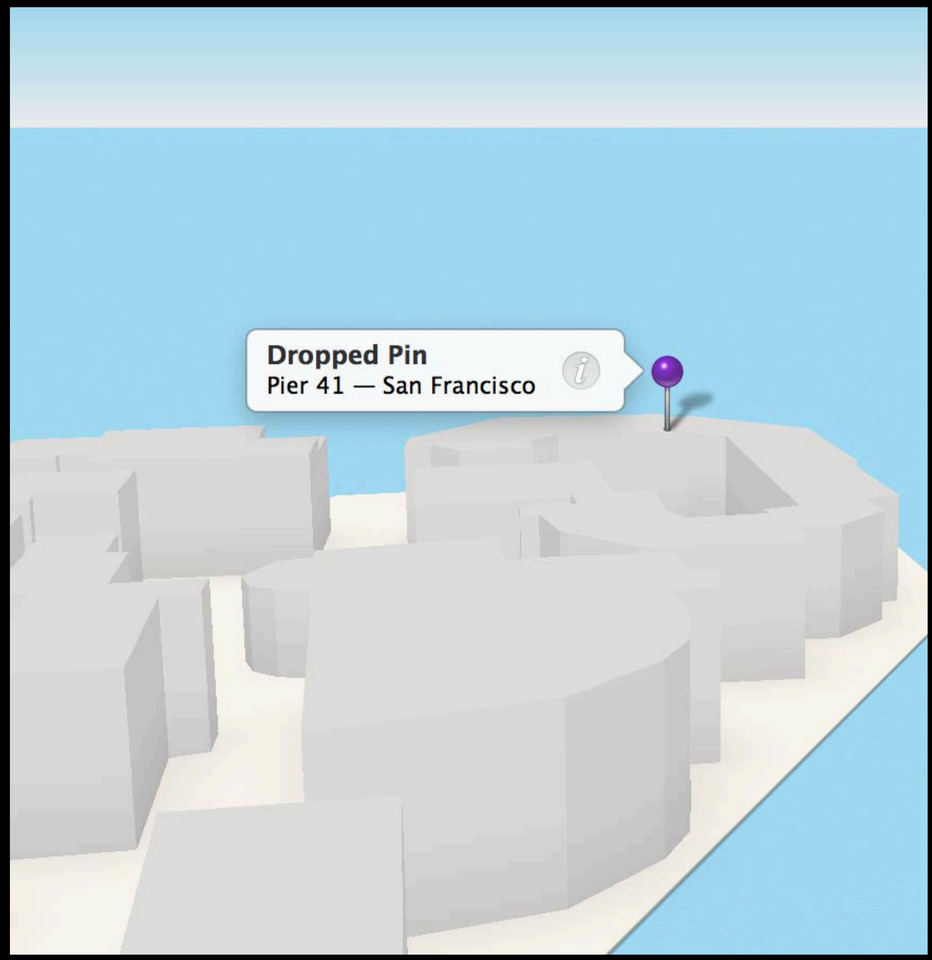


Adapting to Existing API Changes

Geometry conversions

- Four existing methods on MKMapView

```
-convertPoint:toCoordinateFromView:
-convertCoordinate:toPointToView:
-convertRect:toRegionFromView:
-convertRegion:toRectToView:
```



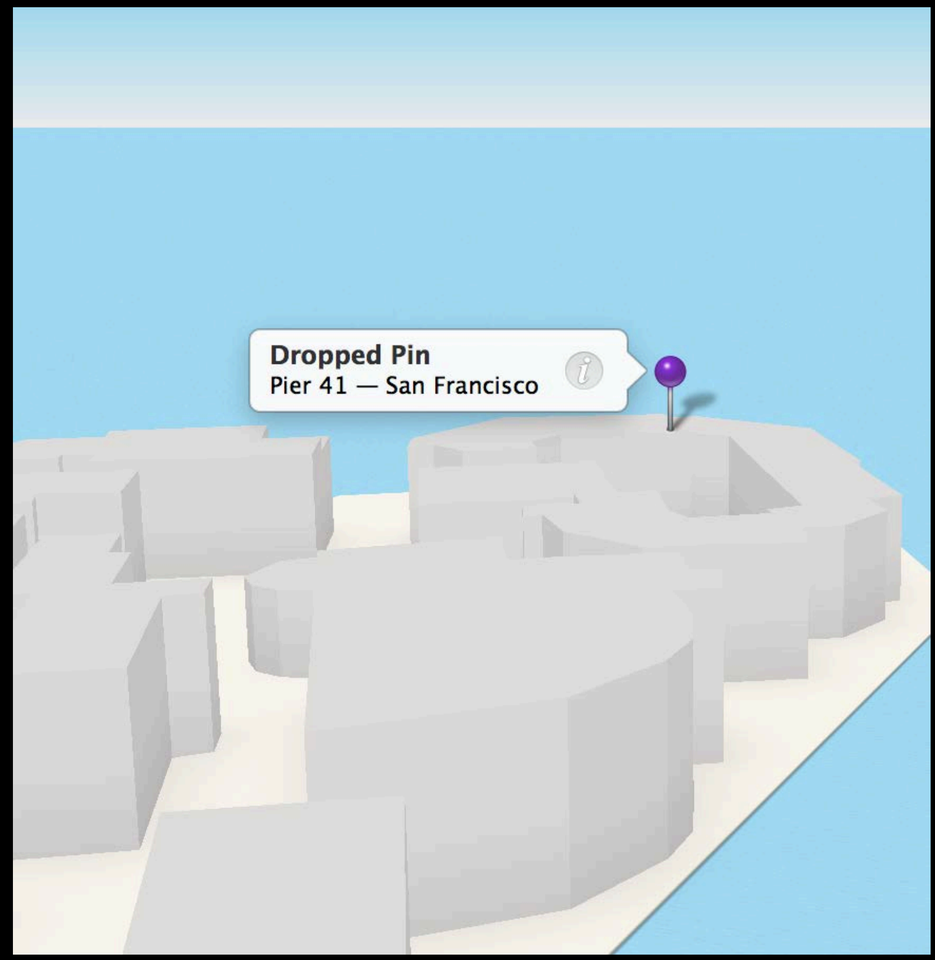
Adapting to Existing API Changes

Geometry conversions

- Four existing methods on MKMapView

```
-convertPoint:toCoordinateFromView:
-convertCoordinate:toPointToView:
-convertRect:toRegionFromView:
-convertRegion:toRectToView:
```

- Can return invalid values



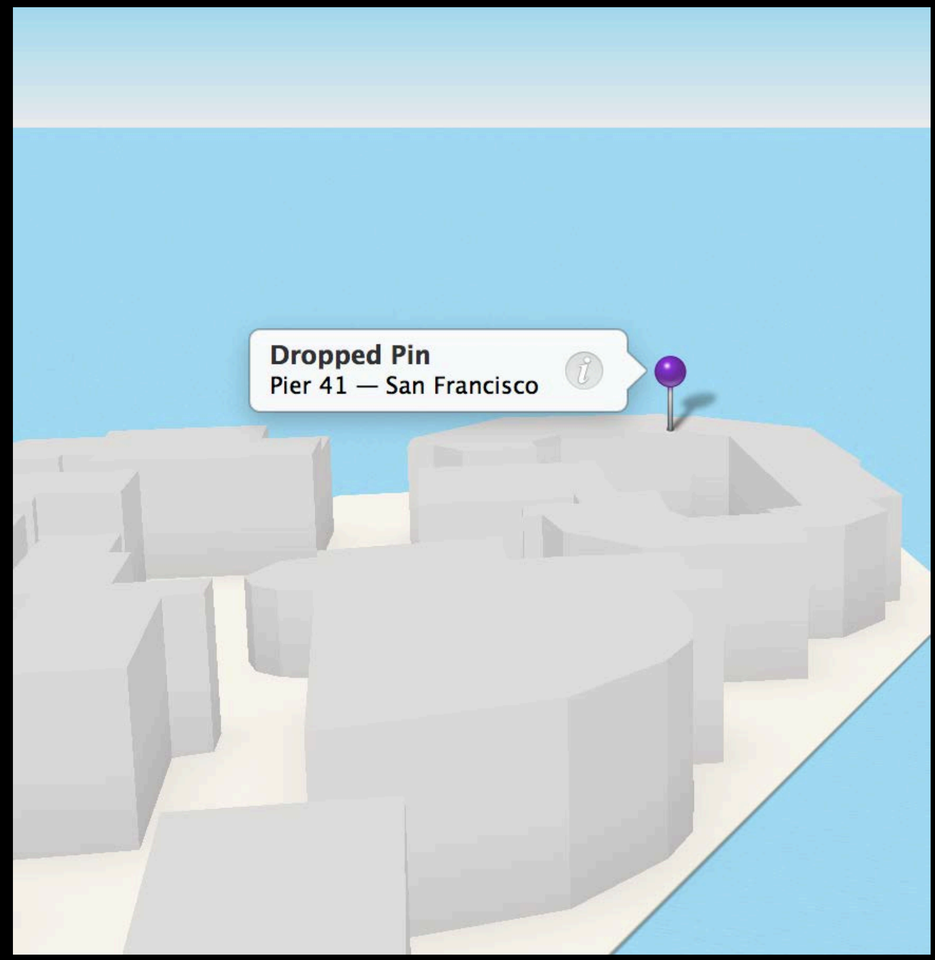
Adapting to Existing API Changes

Geometry conversions

- Four existing methods on MKMapView

```
-convertPoint:toCoordinateFromView:
-convertCoordinate:toPointToView:
-convertRect:toRegionFromView:
-convertRegion:toRectToView:
```

- Can return invalid values
- Test for `kCLLocationCoordinate2DInvalid`



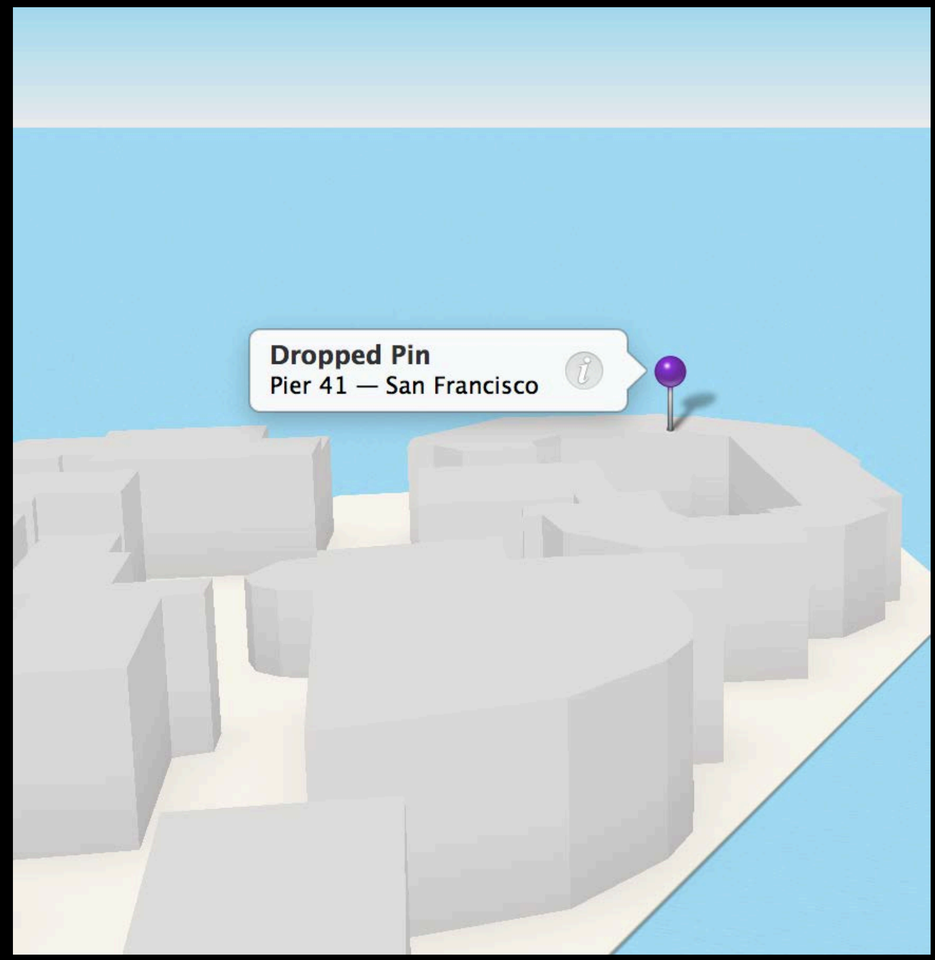
Adapting to Existing API Changes

Geometry conversions

- Four existing methods on MKMapView

```
-convertPoint:toCoordinateFromView:
-convertCoordinate:toPointToView:
-convertRect:toRegionFromView:
-convertRegion:toRectToView:
```

- Can return invalid values
- Test for `kCLLocationCoordinate2DInvalid`
- Test for `CGRectNull`



Adapting to Existing API Changes

Recap

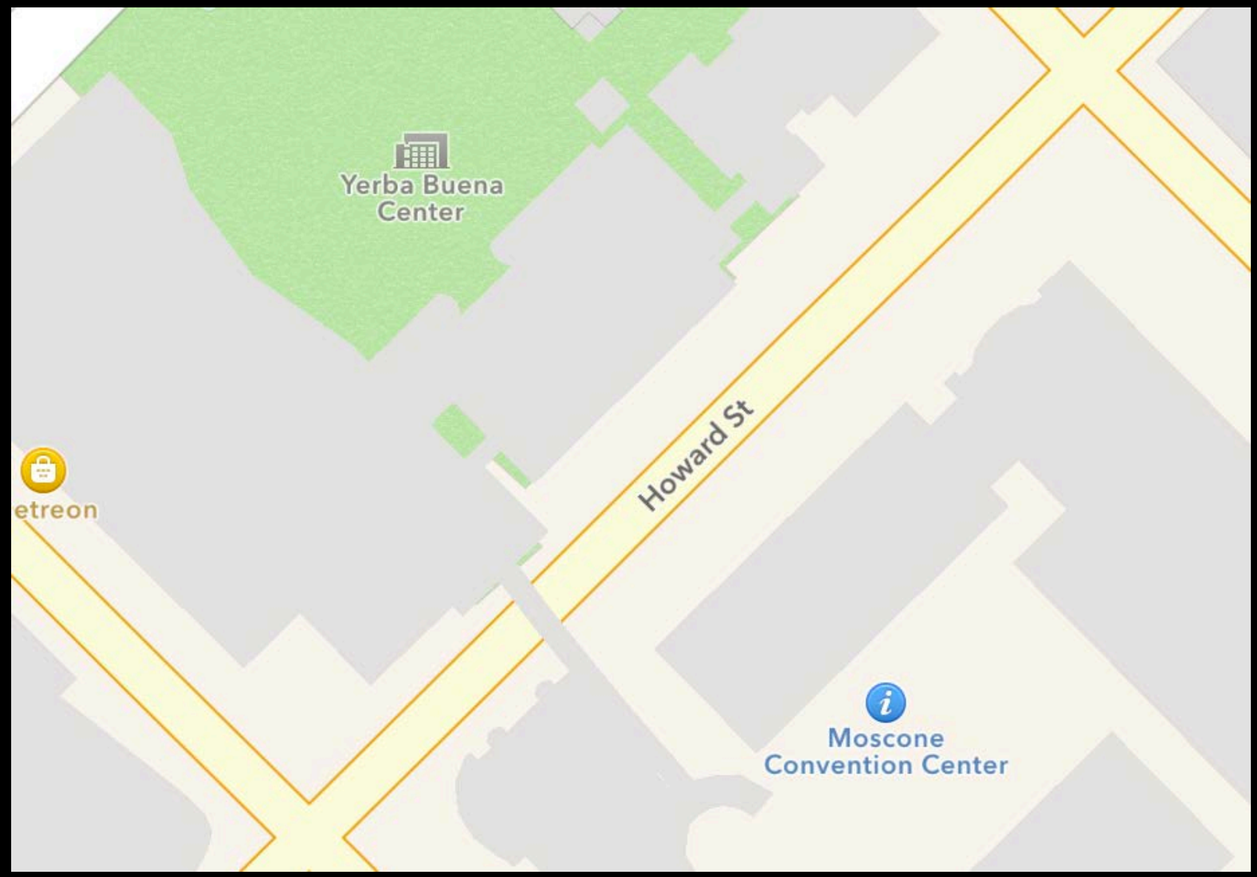
- Mostly compatible
- Annotations track the map
- Overlays cross the 180th Meridian
- Check for invalid geometry

Adopting MKMapCamera

Made for 3D

Adopting MKMapCamera

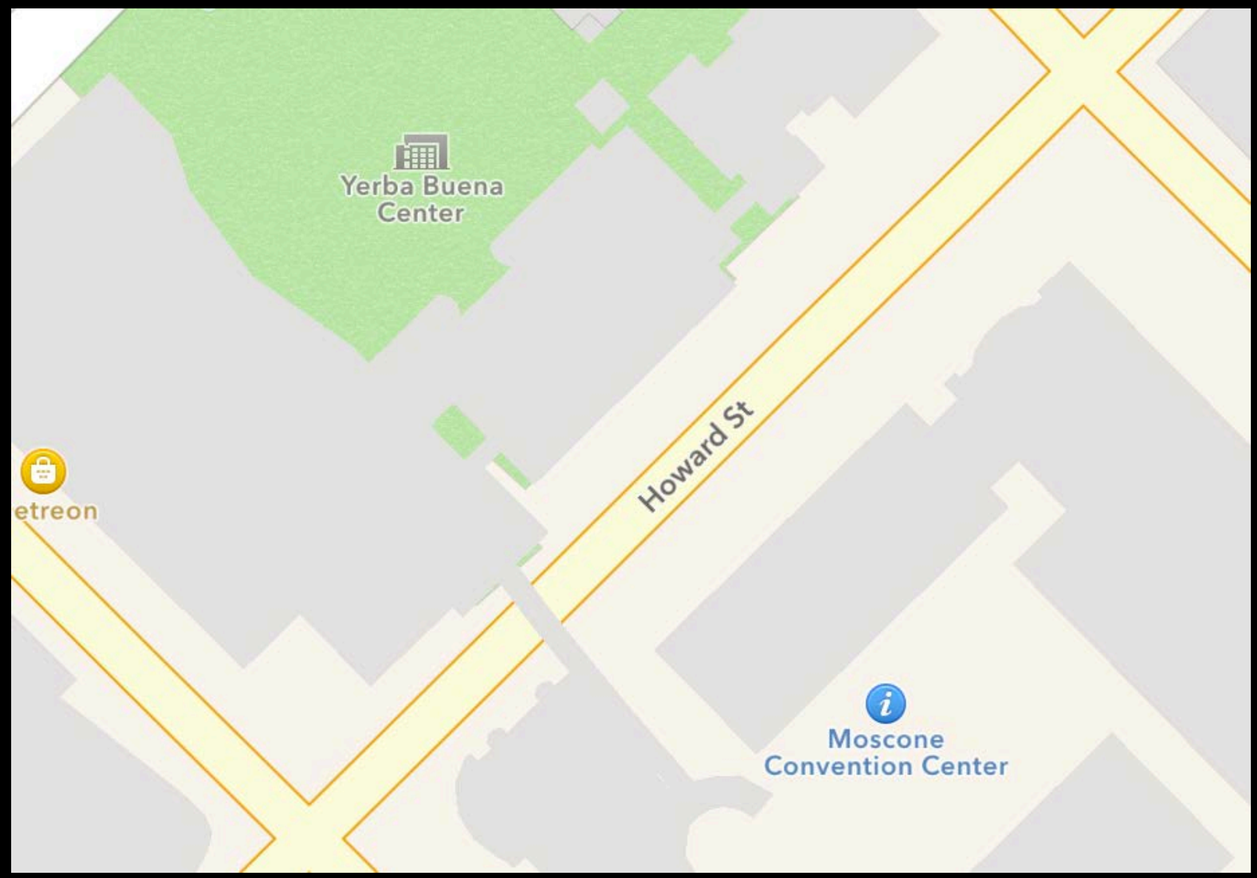
What is MKMapCamera



Adopting MKMapCamera

What is MKMapCamera

North

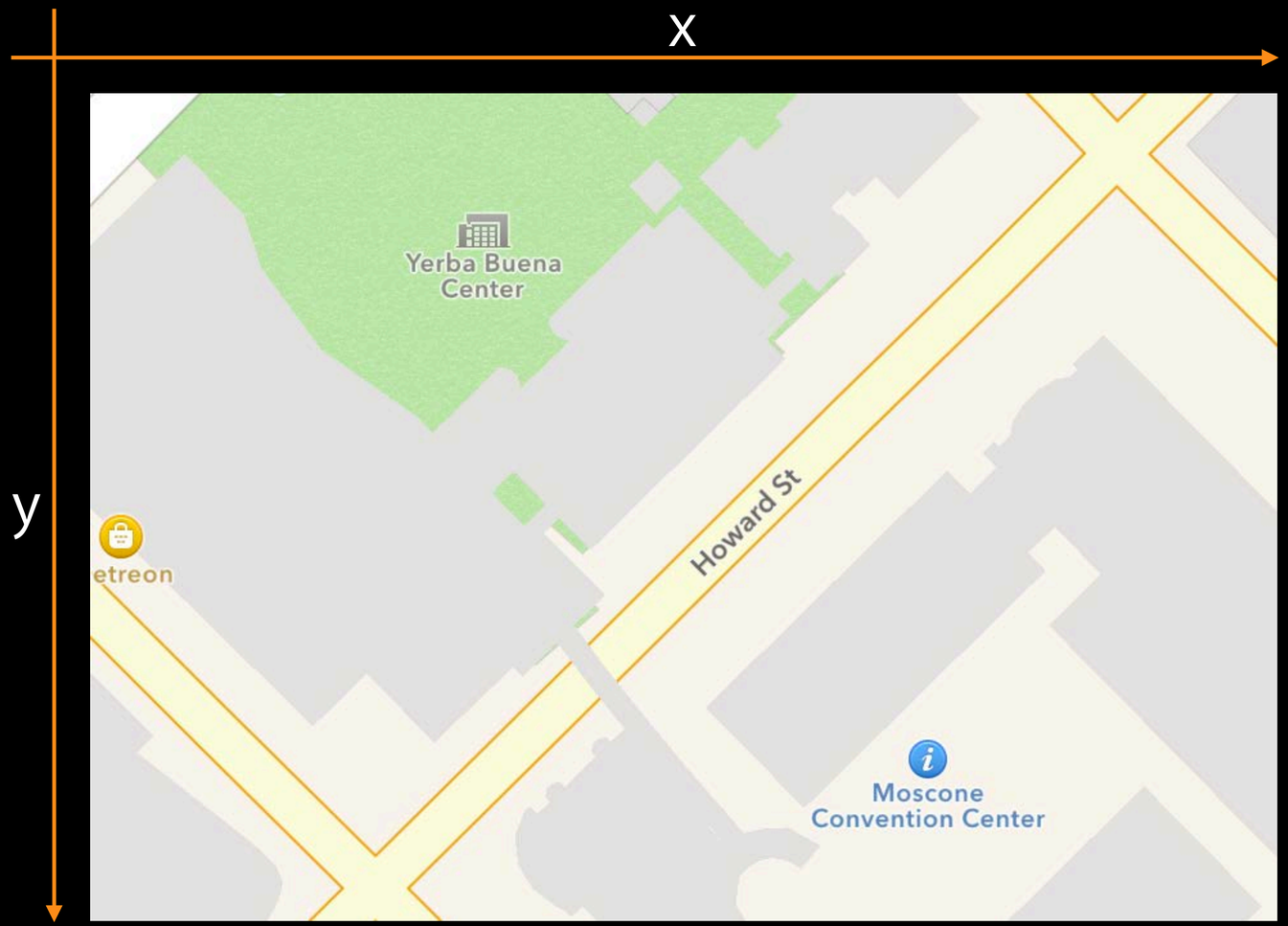


West East

South

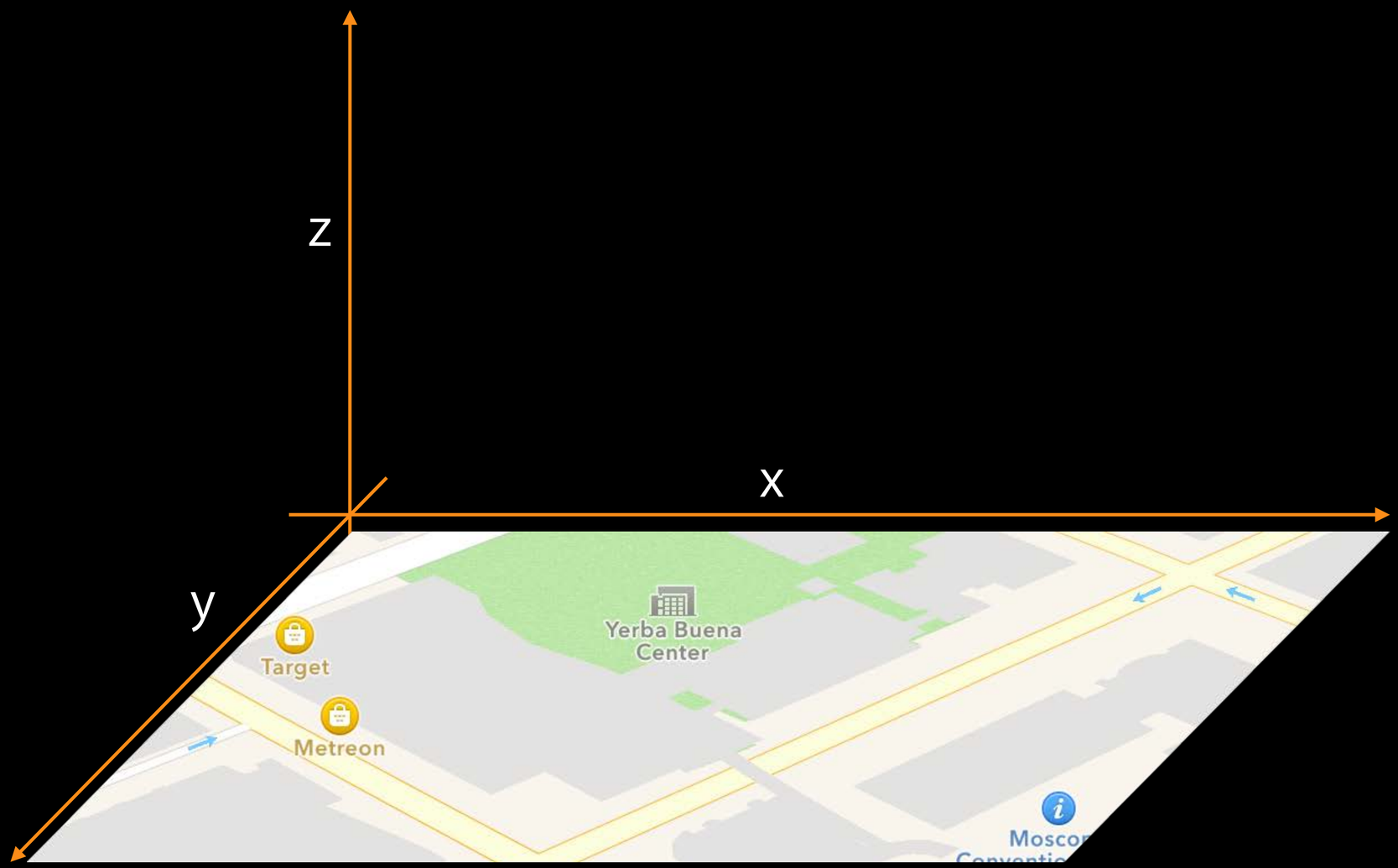
Adopting MKMapCamera

What is MKMapCamera



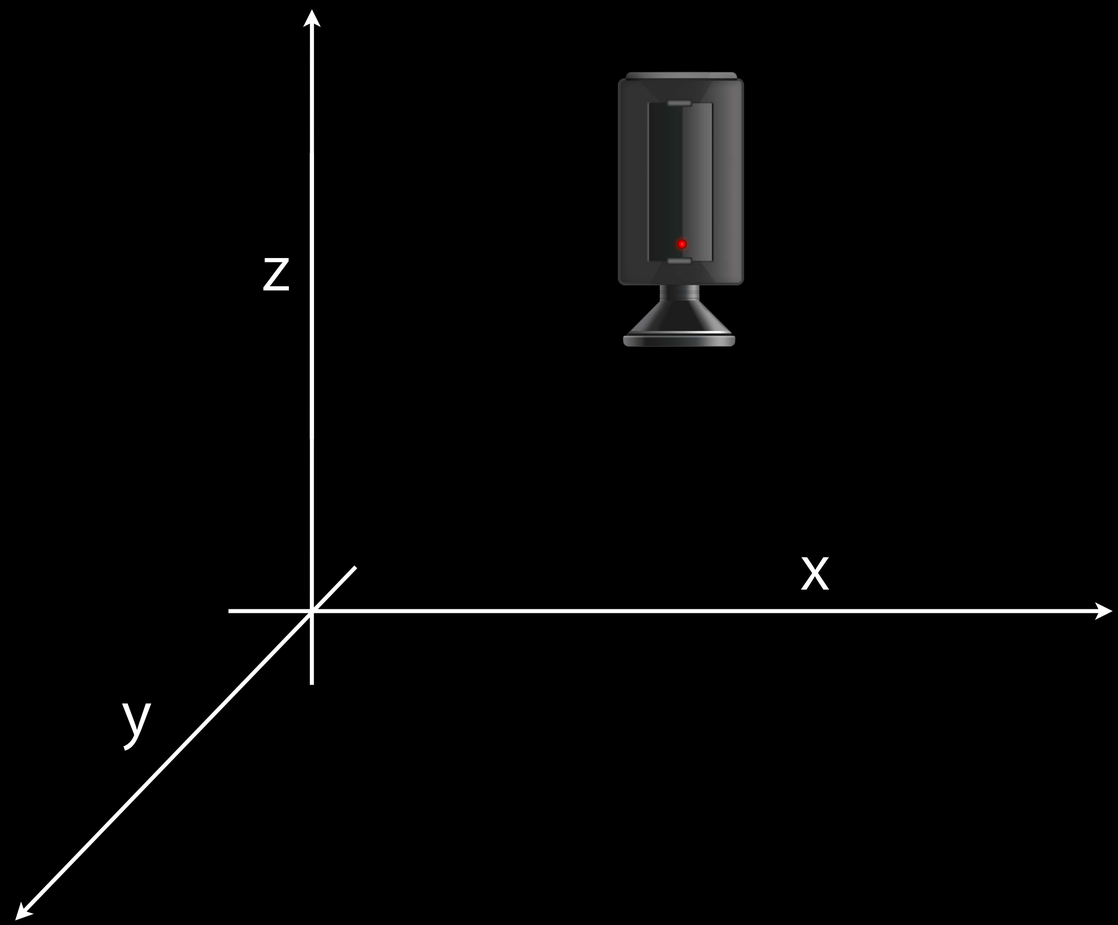
Adopting MKMapCamera

What is MKMapCamera



Adopting MKMapCamera

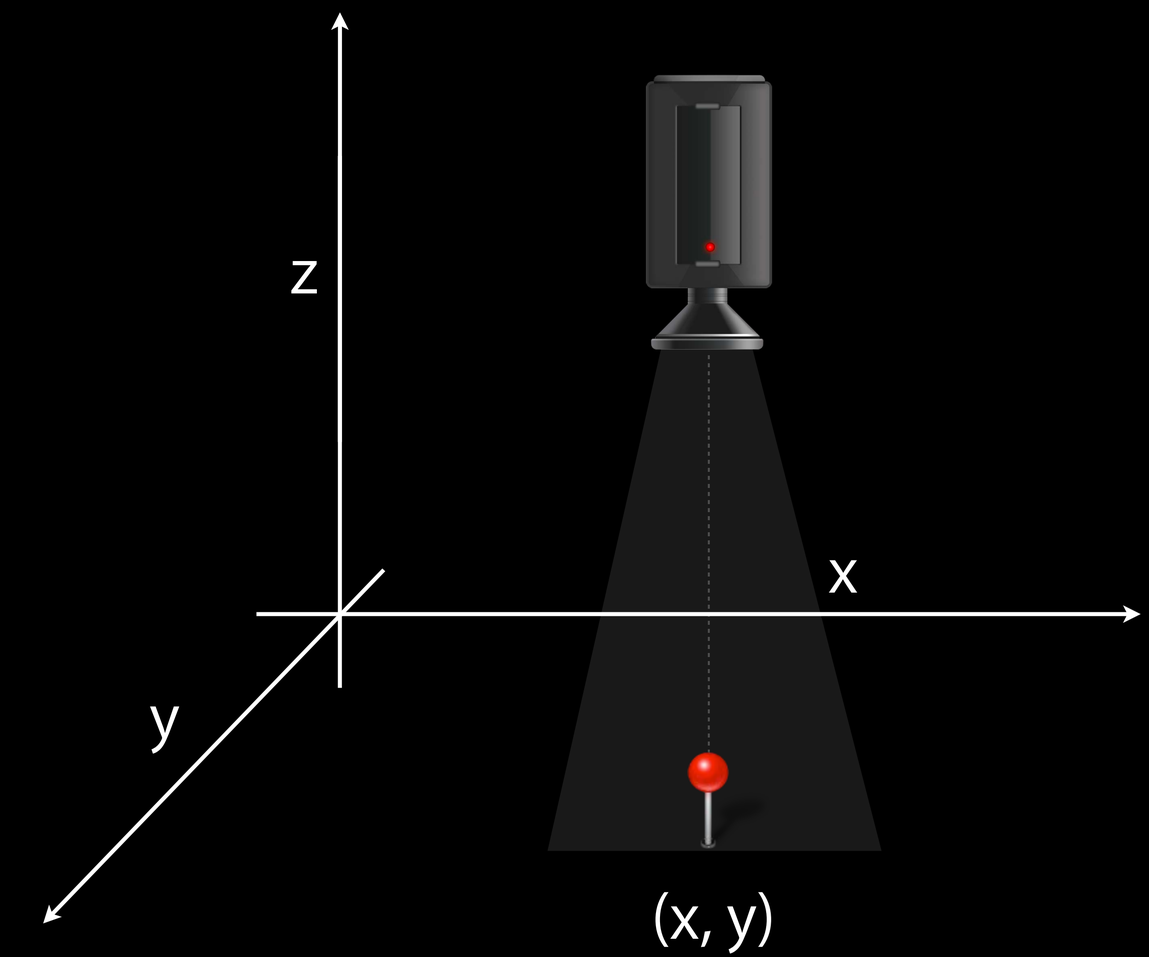
Four basic properties



Adopting MKMapCamera

Four basic properties

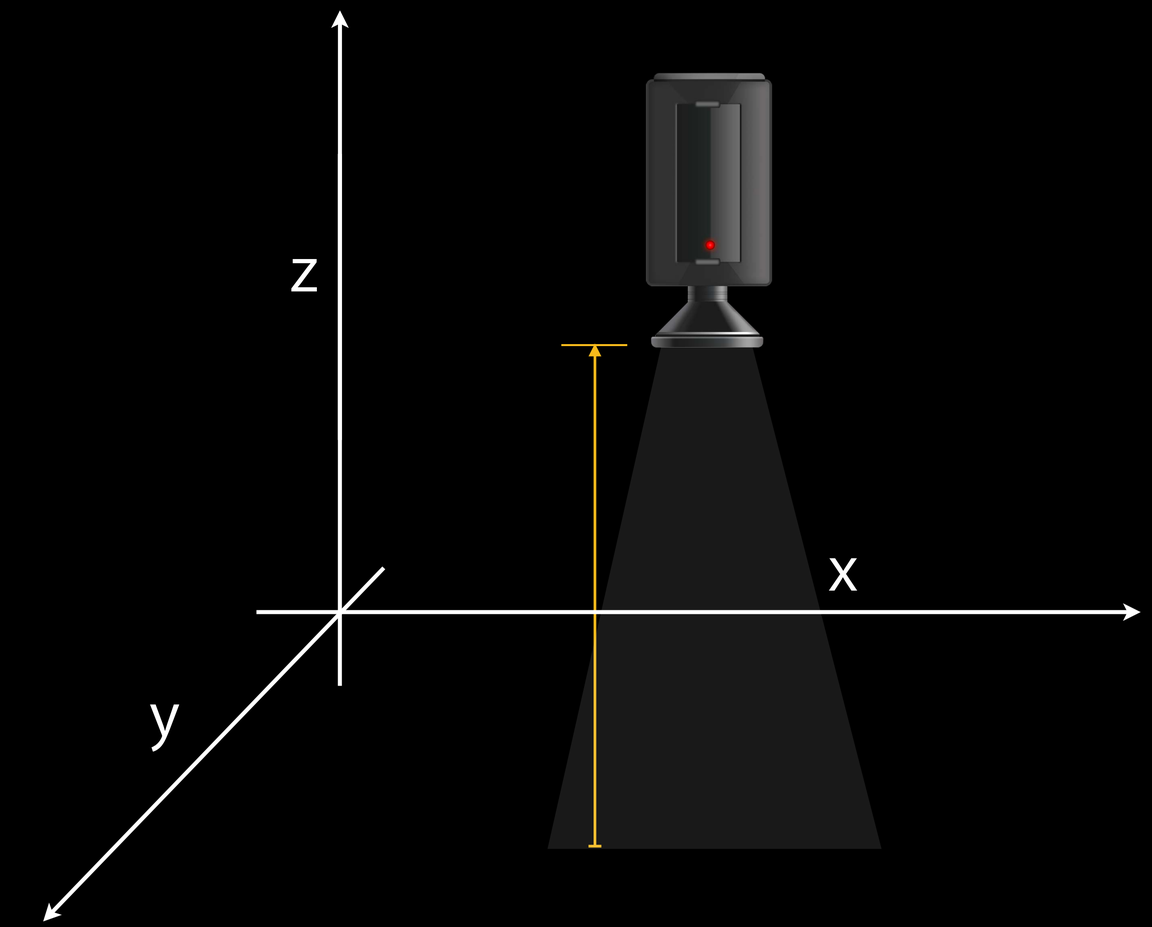
- Center coordinate
  - Point on the ground
  - Appears at screen center



Adopting MKMapCamera

Four basic properties

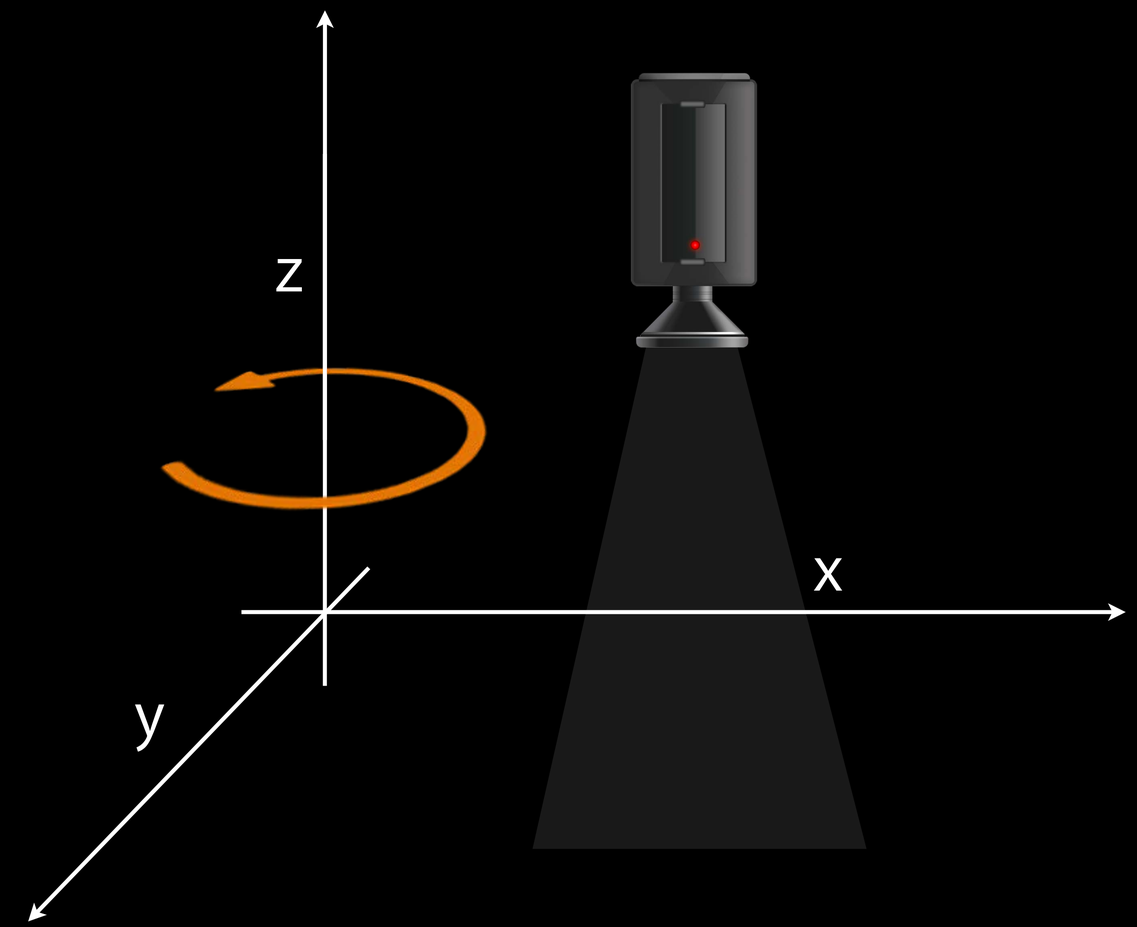
- Center coordinate
  - Point on the ground
  - Appears at screen center
- Altitude
  - Height above map



Adopting MKMapCamera

Four basic properties

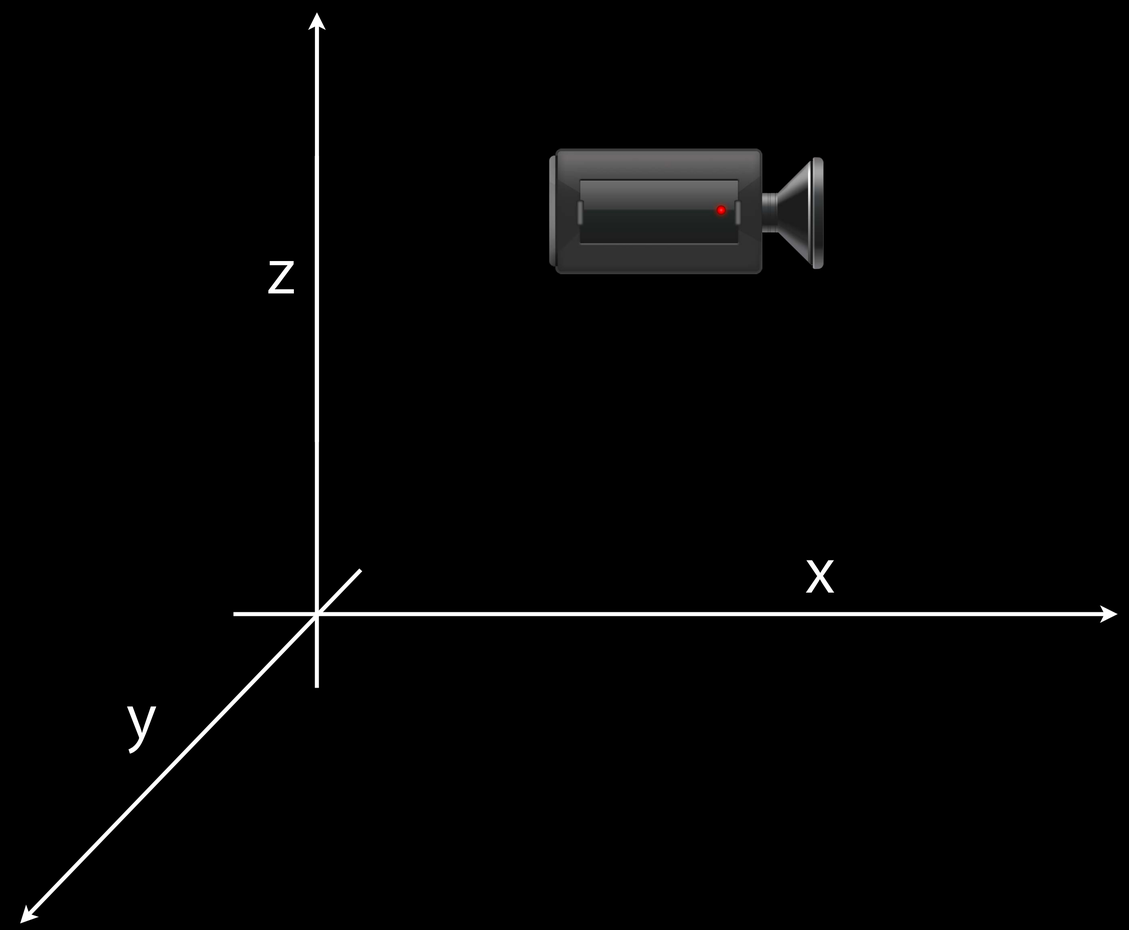
- Center coordinate
  - Point on the ground
  - Appears at screen center
- Altitude
  - Height above map
- Heading
  - Direction camera faces



Adopting MKMapCamera

Four basic properties

- Center coordinate
  - Point on the ground
  - Appears at screen center
- Altitude
  - Height above map
- Heading
  - Direction camera faces
- Pitch
  - Angle camera tilts



Adopting MKMapCamera

Four Basic Properties

```
@interface MKMapCamera

@property CLLocationCoordinate2D centerCoordinate;
@property CLLocationDistance altitude;
@property CLLocationDirection heading;
@property CGFloat pitch;

+ (id)camera;

...

@end
```

Adopting MKMapCamera

Four Basic Properties

```
@interface MKMapCamera

@property CLLocationCoordinate2D centerCoordinate;
@property CLLocationDistance altitude;
@property CLLocationDirection heading;
@property CGFloat pitch;

+ (id)camera;

...

@end
```

Adopting MKMapCamera

Four Basic Properties

```
@interface MKMapCamera

@property CLLocationCoordinate2D centerCoordinate;
@property CLLocationDistance altitude;
@property CLLocationDirection heading;
@property CGFloat pitch;

+ (id)camera;

...

@end
```

Adopting MKMapCamera

What is MKMapCamera

Adopting MKMapCamera

What is MKMapCamera

- Look at coordinate

Adopting MKMapCamera

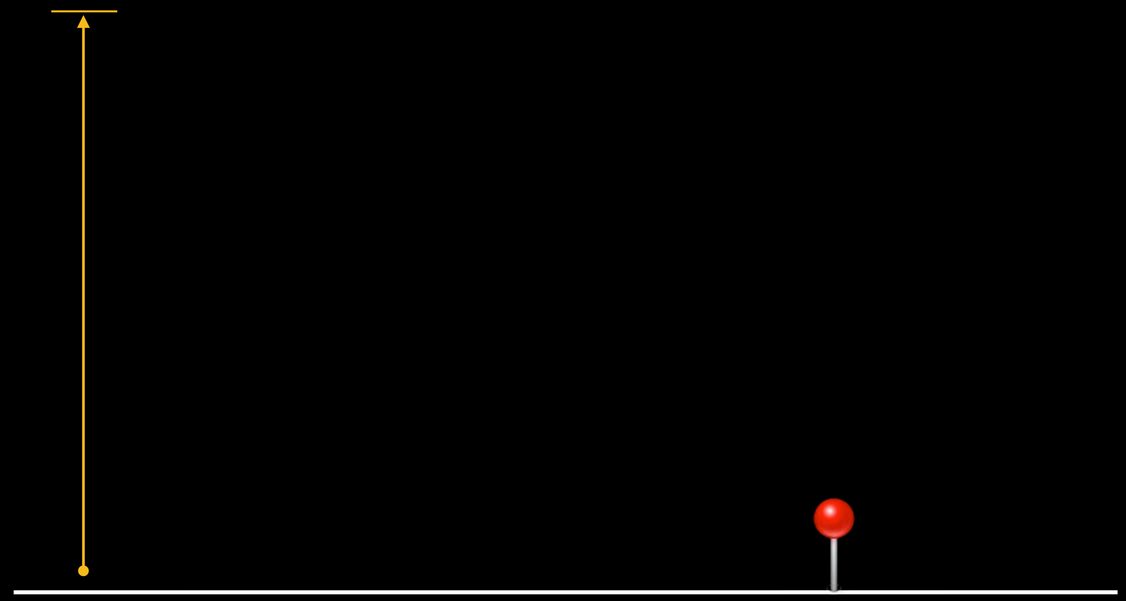
What is MKMapCamera

- Look at coordinate
- From

Adopting MKMapCamera

What is MKMapCamera

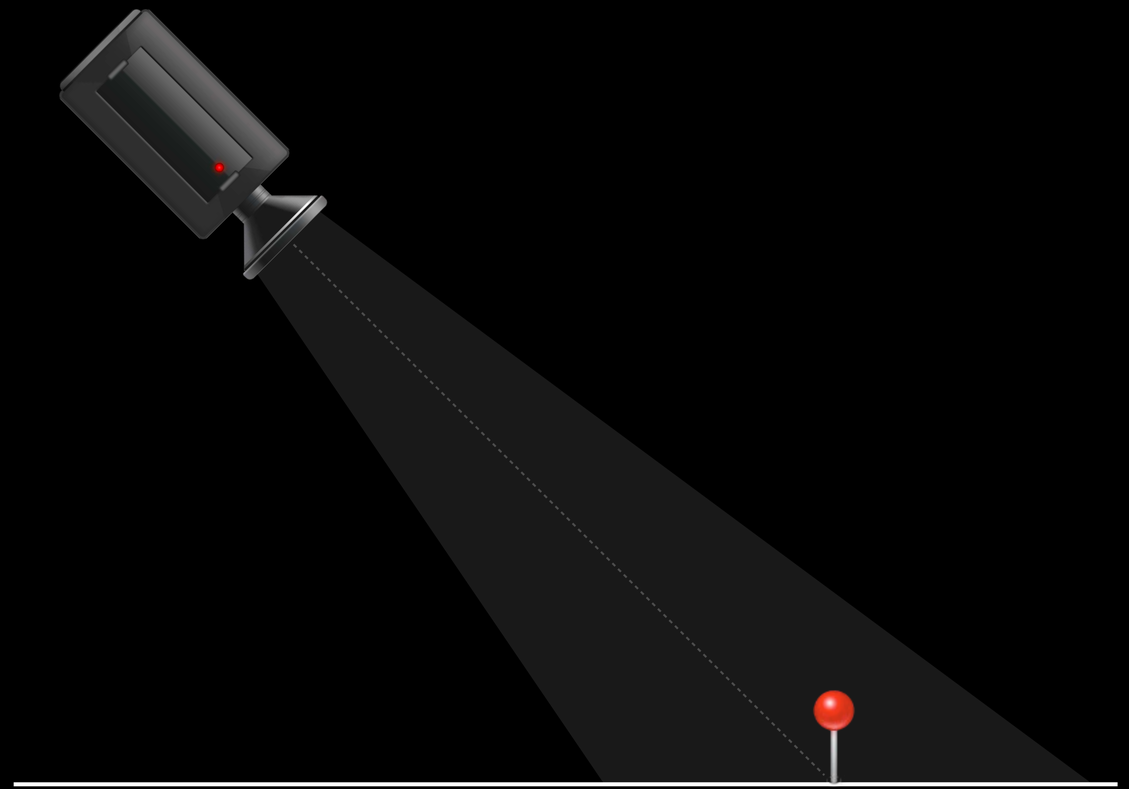
- Look at coordinate
- From
- Altitude



Adopting MKMapCamera

What is MKMapCamera

- Look at coordinate
- From
- Altitude



Adopting MKMapCamera

What is MKMapCamera


```
CLLocationCoordinate2D ground = CLLocationCoordinate2DMake(...);
CLLocationCoordinate2D eye = CLLocationCoordinate2DMake(...);
MKMapCamera *myCamera = [MKMapCamera cameraLookingAtCenterCoordinate:ground
                                                        fromEyeCoordinate:eye
                                                              eyeAltitude:100];

mapView.camera = myCamera;
```

Adopting MKMapCamera

What is MKMapCamera


```
CLLocationCoordinate2D ground = CLLocationCoordinate2DMake(...);
CLLocationCoordinate2D eye = CLLocationCoordinate2DMake(...);
MKMapCamera *myCamera = [MKMapCamera cameraLookingAtCenterCoordinate:ground
                                                        fromEyeCoordinate:eye
                                                              eyeAltitude:100];

mapView.camera = myCamera;
```

Adopting MKMapCamera

What is MKMapCamera


```
CLLocationCoordinate2D ground = CLLocationCoordinate2DMake(...);
CLLocationCoordinate2D eye = CLLocationCoordinate2DMake(...);
MKMapCamera *myCamera = [MKMapCamera cameraLookingAtCenterCoordinate:ground
                                                   fromEyeCoordinate:eye
                                                         eyeAltitude:100];

mapView.camera = myCamera;
```

Adopting MKMapCamera

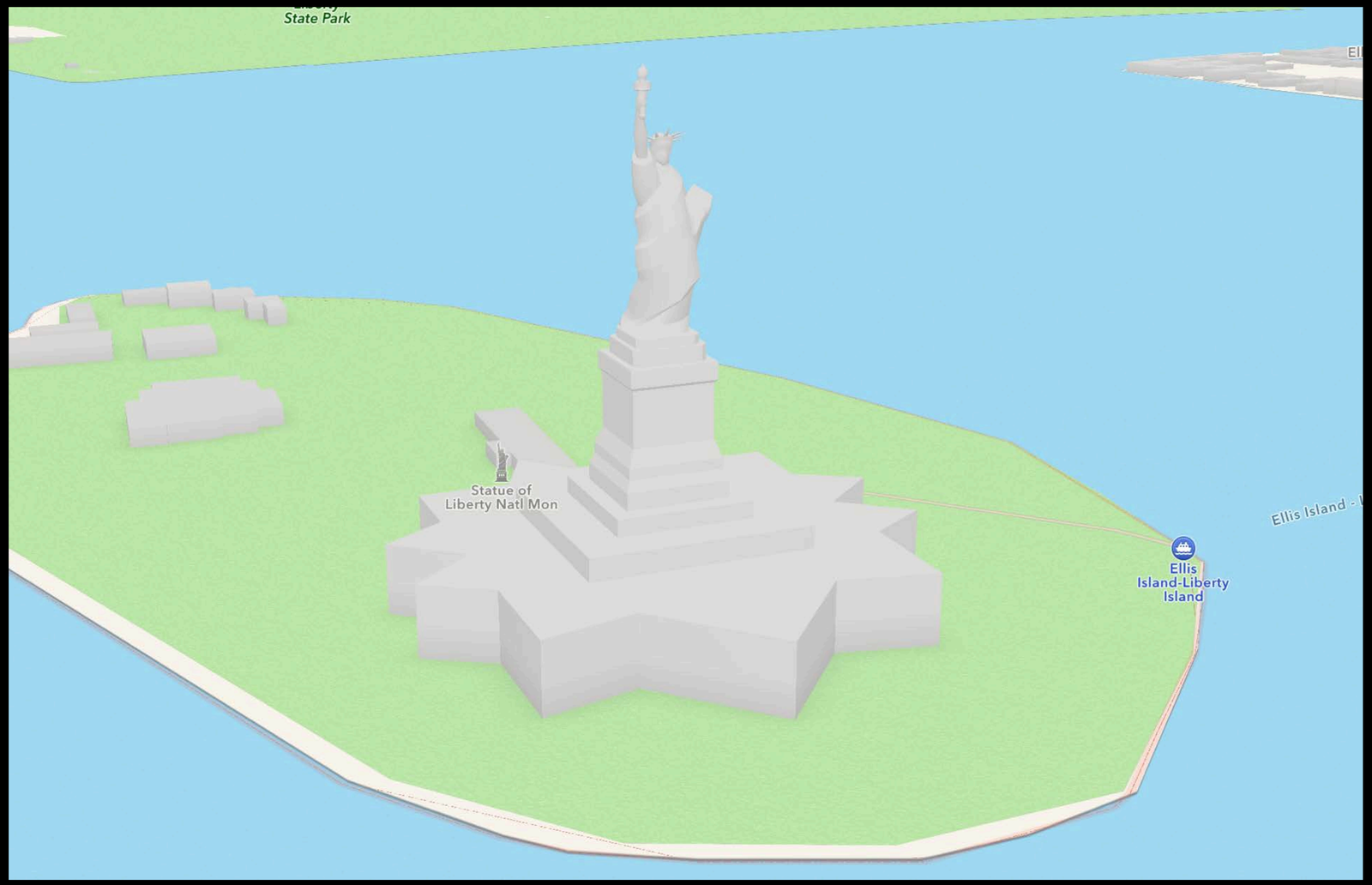
What is MKMapCamera


```
CLLocationCoordinate2D ground = CLLocationCoordinate2DMake(...);
CLLocationCoordinate2D eye = CLLocationCoordinate2DMake(...);
MKMapCamera *myCamera = [MKMapCamera cameraLookingAtCenterCoordinate:ground
                                                        fromEyeCoordinate:eye
                                                              eyeAltitude:100];

mapView.camera = myCamera;
```

Adopting MKMapCamera

What is MKMapCamera



Adopting MKMapCamera

Saving and restoring state

Adopting MKMapCamera

Saving and restoring state

- MKMapCamera implements <NSSecureCoding>

Adopting MKMapCamera

Saving and restoring state

- MKMapCamera implements <NSSecureCoding>

```
MKMapCamera *camera = [map camera];
[NSKeyedArchiver archiveRootObject:camera toFile:stateFile];
```

```
MKMapCamera *camera =
    [NSKeyedUnarchiver unarchiveObjectWithFile:stateFile];
[map setCamera:camera];
```

Adopting MKMapCamera

Saving and restoring state

- MKMapCamera implements <NSSecureCoding>

```
MKMapCamera *camera = [map camera];
[NSKeyedArchiver archiveRootObject:camera toFile:stateFile];
```

```
MKMapCamera *camera =
    [NSKeyedUnarchiver unarchiveObjectWithFile:stateFile];
[map setCamera:camera];
```

Adopting MKMapCamera

Saving and restoring state

- MKMapCamera implements <NSSecureCoding>

```
MKMapCamera *camera = [map camera];
[NSKeyedArchiver archiveRootObject:camera toFile:stateFile];
```

```
MKMapCamera *camera =
    [NSKeyedUnarchiver unarchiveObjectWithFile:stateFile];
[map setCamera:camera];
```

Adopting MKMapCamera

Saving and restoring state

- MKMapCamera implements <NSSecureCoding>

```
MKMapCamera *camera = [map camera];
[NSKeyedArchiver archiveRootObject:camera toFile:stateFile];
```

```
MKMapCamera *camera =
    [NSKeyedUnarchiver unarchiveObjectWithFile:stateFile];
[map setCamera:camera];
```

Adopting MKMapCamera

MKMapCamera

Adopting MKMapCamera

MKMapCamera

Same API on OS X

Demo

Cinematic camera motion

Cinematic Camera Motion

Lessons learned

Cinematic Camera Motion

Lessons learned

- `-[MKMapView setCamera:]` is animatable

Cinematic Camera Motion

Lessons learned

- `-[MKMapView setCamera:]` is animatable
- Use `mapView:regionDidChangeAnimated:`

Cinematic Camera Motion

Lessons learned

- `-[MKMapView setCamera:]` is animatable
- Use `mapView:regionDidChangeAnimated:`
- Transition style should vary based on distance

Adopting MKMapCamera

Recap

- One stop shop for 3D
- Save and restore state
- Add a special touch to your app

Customizing User Interaction

Customizing User Interaction

- Similar to existing MKMapView APIs

```
@property zoomEnabled
@property scrollEnabled
```

Customizing User Interaction

- Similar to existing MKMapView APIs

```
@property zoomEnabled
@property scrollEnabled
```

- New MKMapView APIs

```
@property rotateEnabled
@property pitchEnabled
```

Customizing User Interaction

- Similar to existing MKMapView APIs

```
@property zoomEnabled
@property scrollEnabled
```

- New MKMapView APIs

```
@property rotateEnabled
@property pitchEnabled
```

- All on by default in iOS 7

Customizing User Interaction

- Similar to existing MKMapView APIs

```
@property zoomEnabled
@property scrollEnabled
```

- New MKMapView APIs

```
@property rotateEnabled
@property pitchEnabled
```

- All on by default in iOS 7
- Some devices do not support pitching

Customizing User Interaction

- Similar to existing MKMapView APIs

```
@property zoomEnabled
@property scrollEnabled
```

- New MKMapView APIs

```
@property rotateEnabled
@property pitchEnabled
```

- All on by default in iOS 7
- Some devices do not support pitching
- In these cases `pitchEnabled` will always return `NO`

Static Map Snapshots

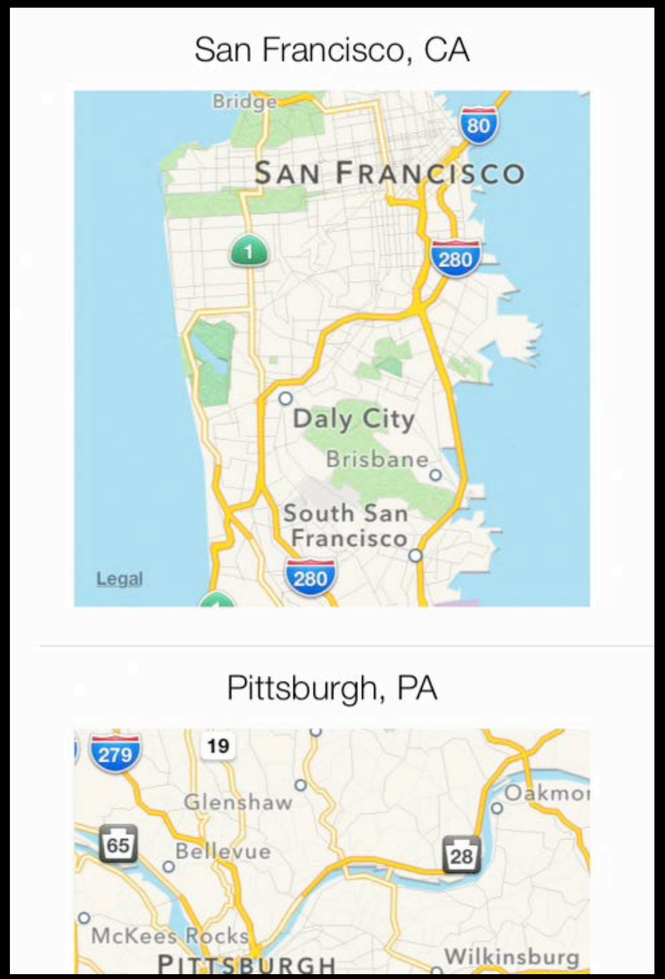
Producing beautiful maps without disrupting interaction

Static Map Snapshots

Why use a snapshot

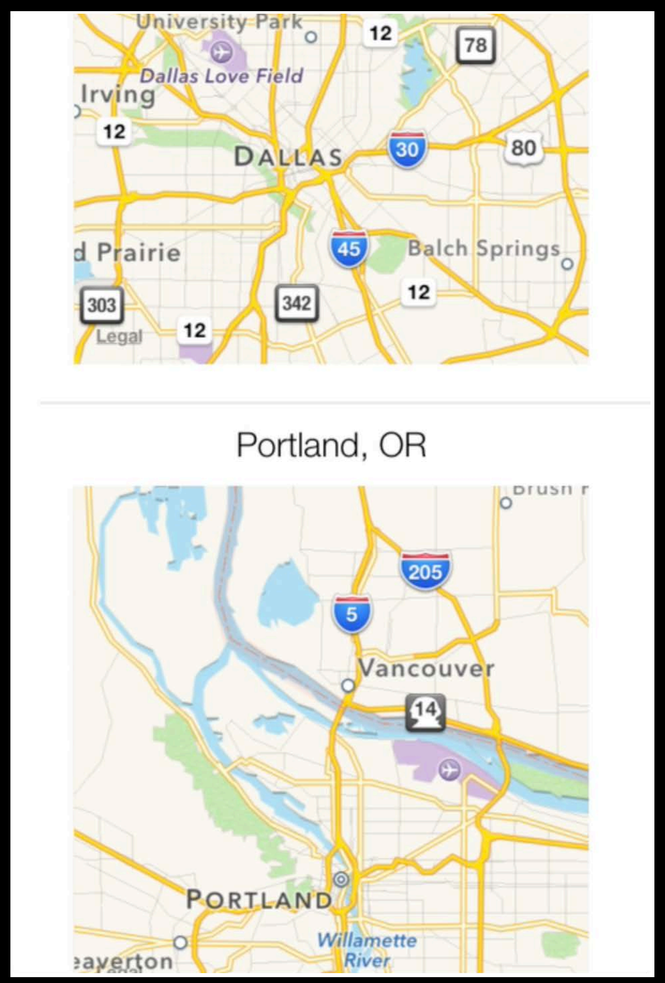
Static Map Snapshots

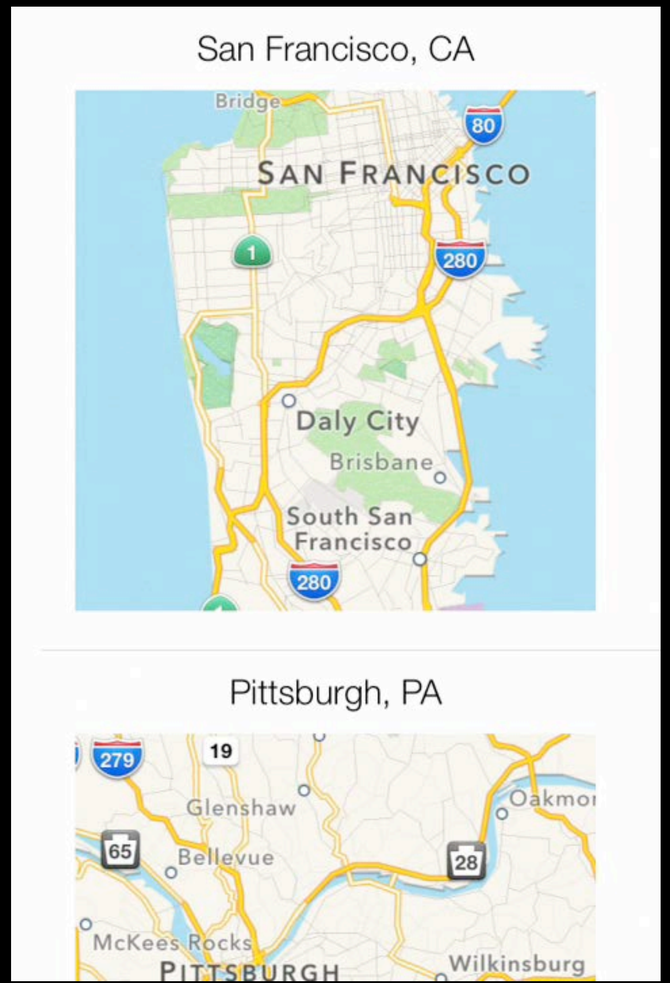
Why use a snapshot



Static Map Snapshots

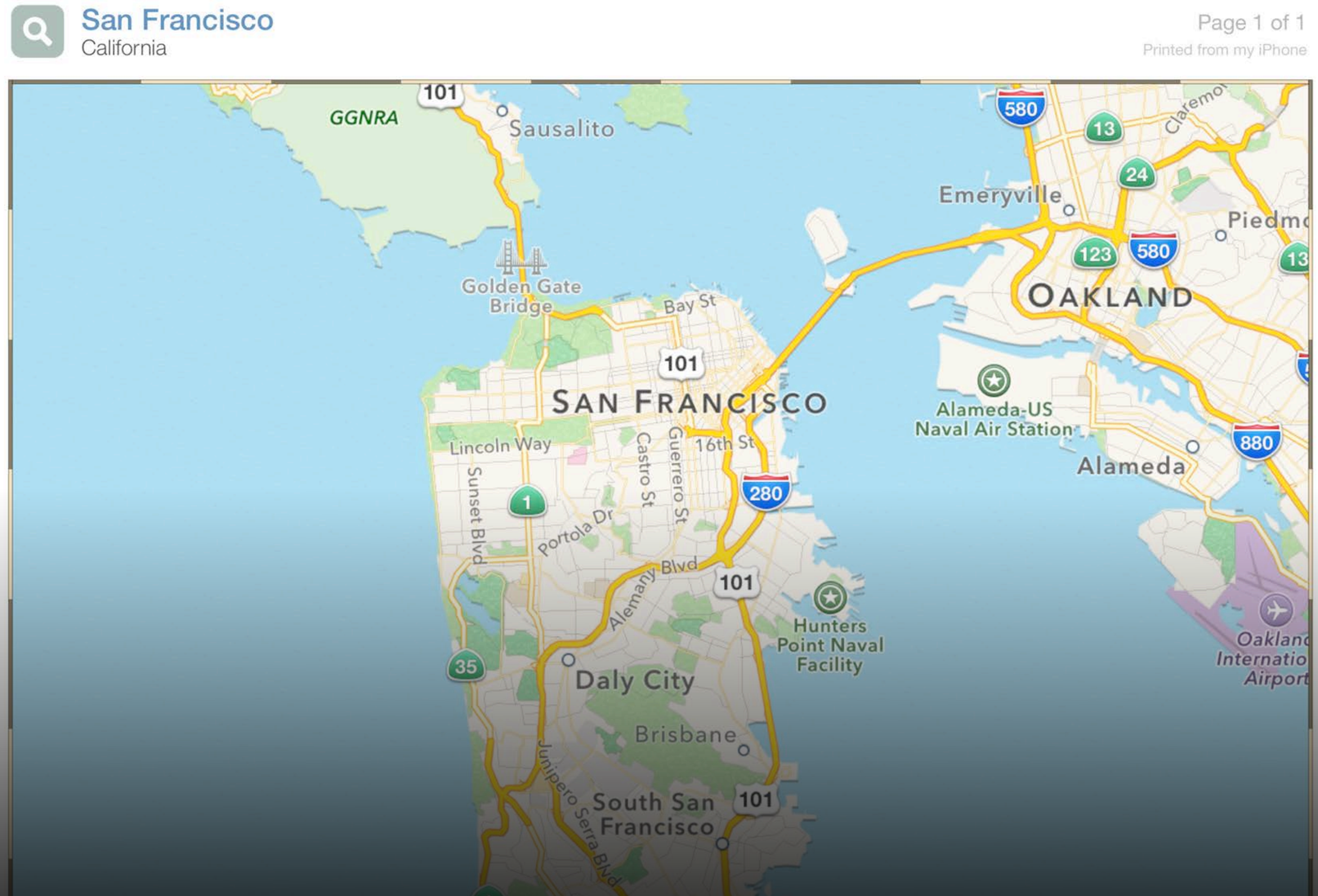
Why use a snapshot





Static Map Snapshots

Why use a snapshot



Static Map Snapshots

Creating a snapshot—three steps

Static Map Snapshots

Creating a snapshot—three steps

- Configure options

Static Map Snapshots

Creating a snapshot—three steps

- Configure options
- Create snapshotter

Static Map Snapshots

Creating a snapshot—three steps

- Configure options
- Create snapshotter
- Start asynchronous task

Static Map Snapshots

Configuring MKMapSnapshotOptions

- Image size
- Map region or camera
- Map type
- Scale (iOS only)

Static Map Snapshots

Creating a snapshot

```
MKMapSnapshotOptions *options = [[MKMapSnapshotOptions alloc] init];
options.size = CGSizeMake(512, 512);
options.scale = [[UIScreen mainScreen] scale]; // iOS only
options.camera = myCamera;
options.mapType = MKMapTypeStandard;
MKMapSnapshotter *snapshotter =
    [[MKMapSnapshotter alloc] initWithOptions:options];
[snapshotter startWithCompletionHandler:^(MKMapSnapshot *snapshot, NSError *e)
{
    if (e) ...;                                 // Handle errors
    UIImage *image = snapshot.image;            // Done!
}];
```

Static Map Snapshots

Creating a snapshot

```
MKMapSnapshotOptions *options = [[MKMapSnapshotOptions alloc] init];
options.size = CGSizeMake(512, 512);
options.scale = [[UIScreen mainScreen] scale]; // iOS only
options.camera = myCamera;
options.mapType = MKMapTypeStandard;
MKMapSnapshotter *snapshotter =
    [[MKMapSnapshotter alloc] initWithOptions:options];
[snapshotter startWithCompletionHandler:^(MKMapSnapshot *snapshot, NSError *e)
{
    if (e) ...;                                 // Handle errors
    UIImage *image = snapshot.image;            // Done!
}];
```

Static Map Snapshots

Creating a snapshot

```
MKMapSnapshotOptions *options = [[MKMapSnapshotOptions alloc] init];
options.size = CGSizeMake(512, 512);
options.scale = [[UIScreen mainScreen] scale]; // iOS only
options.camera = myCamera;
options.mapType = MKMapTypeStandard;
MKMapSnapshotter *snapshotter =
    [[MKMapSnapshotter alloc] initWithOptions:options];
[snapshotter startWithCompletionHandler:^(MKMapSnapshot *snapshot, NSError *e)
{
    if (e) ...;                                 // Handle errors
    UIImage *image = snapshot.image;            // Done!
}];
```

Static Map Snapshots

Creating a snapshot

```
MKMapSnapshotOptions *options = [[MKMapSnapshotOptions alloc] init];
options.size = CGSizeMake(512, 512);
options.scale = [[UIScreen mainScreen] scale]; // iOS only
options.camera = myCamera;
options.mapType = MKMapTypeStandard;
MKMapSnapshotter *snapshotter =
    [[MKMapSnapshotter alloc] initWithOptions:options];
[snapshotter startWithCompletionHandler:^(MKMapSnapshot *snapshot, NSError *e)
{
    if (e) ...;                                 // Handle errors
    UIImage *image = snapshot.image;            // Done!
}];
```

Static Map Snapshots

Creating a snapshot (for printing)

```
-(void)drawContentForPageAtIndex:(NSInteger)index
                          inRect:(CGRect)contentRect
{

}
```

Static Map Snapshots

Creating a snapshot (for printing)

```
-(void)drawContentForPageAtIndex:(NSInteger)index
                        inRect:(CGRect)contentRect
{
  MKMapSnapshotOptions *options = ...
  MKMapSnapshotter *snapshotter = ...

}
```

Static Map Snapshots

Creating a snapshot (for printing)

```
-(void)drawContentForPageAtIndex:(NSInteger)index
                        inRect:(CGRect)contentRect
{
  MKMapSnapshotOptions *options = ...
  MKMapSnapshotter *snapshotter = ...
  [snapshotter startWithCompletionHandler:^{
        // Asynchronous callback
    }];
}
```

Static Map Snapshots

Creating a snapshot (for printing)

```
-(void)drawContentForPageAtIndex:(NSInteger)index
                        inRect:(CGRect)contentRect
{
  MKMapSnapshotOptions *options = ...
  MKMapSnapshotter *snapshotter = ...
  [snapshotter startWithCompletionHandler:^{
        // Asynchronous callback
    }];
}
```

Static Map Snapshots

Creating a snapshot (for printing)

- Configure options
- Create a snapshotter
- Create a semaphore
- Choose a dispatch queue
- Create result variables
- Start the snapshotter
- Wait for snapshotter to complete

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Static Map Snapshots

Creating a snapshot (for printing)

```
options.scale = 2;  // iOS only
dispatch_semaphore_t snapshotSem = dispatch_semaphore_create(0);
dispatch_queue_t queue = dispatch_get_global_queue(...);
__block MKMapSnapshot *mapSnapshot = nil;
__block NSError *error = nil;
[snapshotter startWithQueue:queue
          completionHandler:^(MKMapSnapshot *snapshot, NSError *e) {
    mapSnapshot = snapshot;
    error = e;
    dispatch_semaphore_signal(snapshotSem);
}];
dispatch_semaphore_wait(snapshotSem, DISPATCH_TIME_FOREVER);
if (error) ...;
UIImage *image = mapSnapshot.image;
```

Drawing on a Snapshot

Drawing on a Snapshot

Demo

Generating static map snapshots

Generating Map Snapshots

Lessons learned

Generating Map Snapshots

Lessons learned

- iOS and OS X share similar APIs

Generating Map Snapshots

Lessons learned

- iOS and OS X share similar APIs
- Reuse annotation view classes

Static Map Snapshots

Recap

- Performance
- Printing
- Replace use of `-renderInContext:`
- Anytime you want an image

Putting Map Kit in Perspective

Summary

Putting Map Kit in Perspective

Summary

- Map Kit on OS X

Putting Map Kit in Perspective

Summary

- Map Kit on OS X
- Recompile

Putting Map Kit in Perspective

Summary

- Map Kit on OS X
- Recompile
- Adapt

Putting Map Kit in Perspective

Summary

- Map Kit on OS X
- Recompile
- Adapt
- Adopt MKMapCamera

Putting Map Kit in Perspective

Summary

- Map Kit on OS X
- Recompile
- Adapt
- Adopt MKMapCamera
- Use MKMapSnapshotter

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

What's New in Map Kit	Presidio Thursday 9:00AM	

Labs

Map Kit Lab	Service Lab B Thursday 10:15AM
Map Kit Lab	Service Lab A Thursday 3:15PM

WWDC2013