ArcGIS Runtime

- Family of SDKs for multiple platforms
  - Consistent capabilities
- Native to the platform
  - For building great apps
- Lightweight and fast
- Powerful
- Easy
Part of the ArcGIS platform

- **ArcGIS Online / Portal**
  - Maps, services, content and Organization branding

- **ArcGIS for Server**
  - Services

- **ArcGIS for Desktop**
  - Packages
SDK Platforms

Android 10.1.1
Java/Android

JavaSE 10.1.1
Java Win/Linux

QT 10.2 BETA
C++ Win/Linux

iOS 10.1.1
Objective-C Cocoa Touch

OS X 10.2 BETA
Objective-C Cocoa

Windows Mobile 10.1.1
.NET

WPF 10.1.1
.NET/XAML

Windows Phone 10.1.1
.NET/XAML

Windows Store 10.2 BETA
.NET/XAML

Core Runtime
ArcGIS Runtime SDK for iOS

- Build native applications using Objective-C
  - iPhone 3GS, iPhone 4 & 4S, 5, iPod Touch, iPad
  - iOS 4.3 and up
Web or Native applications?

- ESRI supports both

- Advantages of native applications
  - Tighter integration with other native apps
  - Access to resources
    - Contacts, calendar events, photos
    - Marketing/Hosting/Reporting via AppStore

- Disadvantages
  - Dedicated effort to write and maintain
Before you begin…

- Intel-based Mac
  - OSX 10.7 or 10.8 (Lion & Mountain Lion)

- Xcode (IDE) from the App Store
  - Simulator
  - iOS SDK
- ArcGIS Runtime for iOS v10.1.1 Update 1

To test and deploy on actual hardware or older iOS…

- Join Apple’s iOS Developer Program
  - Standard : AppStore distribution
  - Enterprise : In-House distribution
Objective-C
Objective-C basics

- **Class = Interface + Implementation**

- **MyController.h**

```objective-c
@interface MyController: UIViewController
    // method declarations here
@end
```

- **MyController.m**

```objective-c
#import "MyController.h"

@implementation MyController
    // method implementations here
@end
```
Objective-C basics

• Contd.

• Protocol
  - Declaring a Protocol

```objective-c
@protocol UIApplicationDelegate
@required
    // method declarations here
@end

@protocol MyController: UIViewController <UIApplicationDelegate>
@end
```

- Adopting a protocol

```objective-c
@interface MyController: UIViewController <UIApplicationDelegate>
@end
```
**Objective-C basics**

- Contd.

- **Invoking methods** = passing messages to objects

<table>
<thead>
<tr>
<th>C# / Java</th>
<th>Objective C</th>
</tr>
</thead>
<tbody>
<tr>
<td>foo.alloc();</td>
<td>[foo alloc];</td>
</tr>
<tr>
<td>foo.alloc().init();</td>
<td>[[foo alloc] init];</td>
</tr>
<tr>
<td>point.setCenter(c);</td>
<td>[point setCenter:c];</td>
</tr>
<tr>
<td>point.set(x,y);</td>
<td>[point set X: x andY: y];</td>
</tr>
</tbody>
</table>
Objective-C basics

• Contd.

• Messages are read like English
  - presentViewController:
  - writeToFile:
  - layerFailedToLoad:

• Can get verbose
  mapView:failedLoadingLayerForLayerView:withError:
Memory Management…

- Garbage Collection Is For Kids

- Real developers manage their own memory

- You own an object if you
  - alloc
    
    ```
    MyClass* foo = [MyClass alloc];
    ```
  - retain
    
    ```
    [foo retain];
    ```
  - Or copy
    
    ```
    [foo copy];
    ```

- If you own an object, you’re responsible for releasing it

  ```
  [foo release];
  ```
Memory Management Part 2 (Autorelease)…

• Garbage Collection Is For Kids

• Autorelease Pools help

• Scope based buckets for catching and releasing objects.
Memory Management Part 3 (ARC)…

• Garbage Collection Is For Kids

• Since iOS 5.1…

• Let ARC do the work
  - If you have an object, you don’t need to do a thing.

• Not garbage collection
• Deterministic
  - As good performance as you could code yourself.
Memory Management... Properties

- One last trick
- Properties make memory management easier
- Syntactic sugar – dot notation

```
@interface MyController: UIViewController
@property (nonatomic, strong) MyObject* foo;
@end
```

```
@implementation MyController
@synthesize foo = _foo; // NOT NEEDED with Xcode 4.5
@end
```

```
myController.foo = bar; // bar automatically retained
myController.foo = nil; // bar automatically released
```

Monitor memory footprint with Instruments
Objective-C Summary

- Class = \texttt{@interface (.h) + @implementation (.m)}
- Async response… \texttt{delegate}
- Memory Management (ARC, Autorelease Pools)
What you can do with the SDK

- Display maps
- Perform analysis
- Visualize results
- Collect data
Displaying a Map

- **UI Component**: AGSMapView
  - Responds to gestures
    - Pinch to zoom & rotate
    - Drag to pan
    - Tap & Hold to magnify
  - Displays GPS location
    - Auto pan
Adding data to your map

• Mashup layers
  - ArcGIS Server Tiled layer
  - ArcGIS Server Dynamic layer
  - ArcGIS Server Image
  - Bing
  - Open Street Map
  - Graphics
  - Sketch

• Display WebMaps
  - ArcGIS.com
  - ArcGIS Portal
Demo Summary

- AGSMapView
- Added a tiled layer from a REST service
- Zoomed to an extent
- Delegate Pattern
- UIViewController and UIView
WebMap and the Portal API

- **WebMap**
  - Saved Mashup
  - Configured symbols
  - Defined Popups
  - Saved in ArcGIS.com
Portal API

- Find and Load resources (e.g. a WebMap) from ArcGIS.com (or a private portal)

- Pattern:
  - AGSPortal
    - Init With Credential
  - AGSPortalDelegate: **Portal Loaded**
    - Search or drill down (e.g. for WebMaps)
  - AGSPortalDelegate: **Results Found**
    - If it’s a WebMap, we could open it in the AGSMapView
Demo
Using a WebMap
Demo Summary

- Loaded AGSW ebMap into AGSMapView
- No zoom needed
- No adding layers needed
Performing Analysis

• Using Tasks

  • Query, Find, Identify Task
    - Search for features in the map

  • Geoprocessing Task
    - Spatial analysis using GP tools and models

• Locator
  - Geocode and reverse geocode addresses
  - ArcGIS World Geocoder (POI, Global Find, etc.)
Performing Analysis

- Contd.

- Geometry Service
  - Perform geometry operations on the server

- Routing Task
  - Point-to-point and multipoint driving directions
  - Barriers, Time Windows, Best Sequence

- Closest Facility Task
  - Find nearest facility

- Service Area Task
  - Compute drive times and service areas
Performing Analysis (native)

- Contd.

- Geometry Engine
  - native, high performance engine for performing geometric operations on the device
Common Pattern for using Tasks

1. Adopt the Task Delegate protocol

```objective-c
@interface MyController: UIViewController <AGSLocatorDelegate>
@property (nonatomic, strong) AGSLocator *locator;
@end
```

2. Implement the protocol methods

```objective-c
- (void)locator:(AGSLocator *)locator
  operation:(NSOperation *)op
  didFind:(NSArray *)results {
    // todo
}

- (void)locator:(AGSLocator *)locator
  operation:(NSOperation *)op
  didFailToFindWithError:(NSError*)error {
    // todo
}
```
Common Pattern for using Tasks

3. Instantiate the task

```swift
self.locator = [AGSLocator locatorWithURL:[NSURL URLWithString:kGeoLocatorURL]];
```

4. Set Delegate

```swift
self.locator.delegate = self;
```

5. Perform operation

```swift
NSOperation* op = [self.locator findWithParameters:params];
```
Demo
Using a GeoService
Demo Summary

• Seen the locator (ArcGIS World Geocoder) in use

• Show the result on a map

• Zoom the map to the right extent
Visualizing Results

- Graphics
  - Geometry
  - Attribute
  - Symbol

- Symbols
  - Picture, Marker, Line, Fill
  - Composite
  - Text
Visualizing Results

- Contd.

```swift
//create the symbol
AGSPictureMarkerSymbol *marker =
[ AGSPictureMarkerSymbol pictureMarkerSymbolWithNameNamed: @"BluePushpin.png"];

//create the graphic
AGSGraphic *graphic = [AGSGraphic graphicWithGeometry:point
    symbol: marker
    attributes: dictionary
    infoTemplateDelegate:self.calloutTemplate];

//add the graphic to the graphics layer
[self.graphicsLayer addGraphic:graphic];
```
Visualizing Results

• Contd.

• Renderers
  - Simple
  - Unique Value
  - Class Breaks
  - Temporal
Respond to Map events through Delegates

• Map Delegates

  • Layer Delegate
    - Map/Layer loaded, failed to load
    - `<AGSMapViewLayerDelegate>`

  • Touch Delegate
    - Tap, Double Tap, Tap and Hold
    - `<AGSMapViewTouchDelegate>`

• Callout Delegate
  - Did Show Callout, Did Click Accessory Button
  - `<AGSMapViewCalloutDelegate>`
Which Delegate?  (AGSMapView)

Properties

- BOOL allowMagnifierToPanMap
- AGSCallout * callout
  - id< AGSMapViewCalloutDelegate > calloutDelegate
- AGSColor * gridLineColor
- CGFloat gridLineWidth
- CGFloat gridSize
- BOOL interacting
- BOOL lastChangeFromInteraction
  - id< AGSMapViewLayerDelegate > layerDelegate
- BOOL loaded
- AGSTimeExtent * timeExtent
  - id< AGSMapViewTouchDelegate > touchDelegate
- AGSEnvelope * visibleAreaEnvelope
Responding to Map Touch events

1. Adopt the Delegate protocol

```objective-c
@interface MyController: UIViewController <AGSMapViewTouchDelegate>
...
```

2. Implement the protocol methods

```objective-c
@implementation MyController

- (void) mapView:(AGSMapView *)mapView 
didClickAtPoint:(CGPoint)screen 
mapPoint:(AGSPoint *)mappoint 
graphics:(NSDictionary *)graphics {
    // handle touch event
}
```

3. Set Delegate

```objective-c
self.mapView.touchDelegate = self;
```
Demo
Handling Async Results
Demo Summary

- Show a pattern to track Async Results
- Show how to hitch information to an Async Request
Visualizing Results

• Contd.

• Callout
  - Displayed automatically when user taps on a graphic

• Content
  - Title
  - Detail
  - Image
  - Accessory button

  - OR: Custom UI View
Specifying Content for the Callout

1. Adopt the Delegate Protocol

```objc
@interface MyController: UIViewController
<AGSInfoTemplateDelegate>
...
@end
```

2. Implement the protocol methods

```objc
@implementation MyController

- (NSString *) titleForGraphic:(AGSGraphic *)graphic
screenPoint:(CGPoint)screen
    mapPoint:(AGSPoint *)map {
    //todo
}

- (NSString *) detailForGraphic:(AGSGraphic *)graphic
screenPoint:(CGPoint)screen
    mapPoint:(AGSPoint *)map {
    //todo
}
```

3. Set the `InfoTemplate` delegate on the graphic

```objc
AGSGraphic *graphic = ...
graphic.infoTemplateDelegate = self;
```
Responding to the Callout on Accessory button

1. Adopt the delegate protocol

```objective-c
@interface MyController: UIViewController <AGSMapViewCalloutDelegate>
...
@end
```

2. Implement the protocol methods

```objective-c
@implementation MyController

- (void) mapView:(AGSMapView*) mapView
didClickCalloutAccessoryButtonForGraphic:(AGSGraphic*) graphic {
    //todo
}
```

3. Set the delegate

```objective-c
self.mapView.calloutDelegate = self;
```
Popups

AGSGraphic + AGSPopupInfo → AGSPopup → AGSPopupContainerViewController

WebMap
FeatureLayer
Graphic
Manually
Collecting Data

• Using Feature layers & Popups
  • Feature Layers edit data through Feature Services

• Popups provide UI to
  - Display and edit attributes
  - Manage attachments
  - View charts, media

• Popups configured through WebMaps
  - Attributes to display & edit
  - User friendly aliases and hints
  - Formatting for numbers, dates
Collecting Data

Using Popups

- Edit feature
  - Attributes
  - Geometry
  - Attachments
Editing Attributes

Using Popups

- Input based on field data type
- Support for
  - Subtypes
  - Domains
- Validation
  - Length
  - Numeric range
Managing Attachments

Using Popups

- View & Download
- Add
- Delete
Editing Geometry

- Use GPS location

- Use Sketch Layer
  - Interactively create & reshape geometries
  - Point, line, polygon
  - Undo, redo changes
Application Based on Device Type

- iPhone / iPod Touch
- iPad
- Universal app
Testing

- **ALWAYS** test using a physical device!
  - Performance
    - App
    - Network
    - 3G
  - Memory
  - User experience

- **Instruments**
  - Leaks
  - Zombies
More Resources

- **iOS Resource Center** ([resources.arcgis.com](resources.arcgis.com), [developers.arcgis.com](developers.arcgis.com))
  - Conceptual help, API Reference
  - Blog, Forums
  - Download API v10.1.1 Update 1

- **Samples on ArcGIS.com**
  - ArcGIS for iOS Developer Samples group

- **Web Course : Getting Started with the ArcGIS API for iOS**
  - [training.esri.com](training.esri.com)
Whats coming for Runtime…

• 10.2
  - Offline map use
    - Being productive offline
    - Network/geocode/search
  - Performance (static/dynamic mode)
  - Security (OAuth, SAML)
  - Simplification of apis
  - New developer site (with better doc system)

• Beyond
  - Offline analysis
  - Local data support
    - Raster and Vector
  - 3D
## ArcGIS Runtime Licensing

<table>
<thead>
<tr>
<th>SDK</th>
<th>Development</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>WPF</td>
<td>EDN</td>
<td>Deployment Packs Basic and Standard</td>
</tr>
<tr>
<td>Java</td>
<td>EDN</td>
<td>Deployment Packs Basic and Standard</td>
</tr>
<tr>
<td>Qt</td>
<td>Beta</td>
<td>Deployment Packs Basic and Standard</td>
</tr>
<tr>
<td>iOS</td>
<td>Developers.ArcGIS.com Download</td>
<td>Organization Subscriptions or Portal \ Server</td>
</tr>
<tr>
<td>Android</td>
<td>Developers.ArcGIS.com Download</td>
<td>Organization Subscriptions or Portal \ Server</td>
</tr>
<tr>
<td>Win Phone</td>
<td>Developers.ArcGIS.com Download</td>
<td>Organization Subscriptions or Portal \ Server</td>
</tr>
<tr>
<td>Windows Store</td>
<td>Beta</td>
<td>Organization Subscriptions or Portal \ Server</td>
</tr>
<tr>
<td>OS X</td>
<td>Beta</td>
<td>Organization Subscriptions Portal \ Server</td>
</tr>
</tbody>
</table>
Questions?
Feedback

- Session ID #175

http://flowchainsensei.files.wordpress.com/2012/09/megaphone.jpg
Understanding our world.