Building Mobile Applications with the ArcGIS API for Flex

Lloyd Heberlie, Andy Gup
Lloyd Heberlie,
Product Engineer
Flex API Team
lheberlie@esri.com
@lheberlie

Andy Gup,
Tech Lead for Web APIs, Android
Esri Developer Network
agup@esri.com
@agup

Yann Cabon
Flex Software Developer
Flex API Team
@yanncabon
Demo – Hello World
Why mobile?
Many different screen sizes
Many different pixel densities
Different usage patterns
Hmmm... application size + data?

VS.

Less

More
How come my pages load so slooow?

VS.

Mostly connected

Always connected
How long will my battery last...?
Requirements

- Flash Builder 4.7
- AIR 3.6 SDK
- Adobe Flex 4.6 SDK
- ArcGIS API for Flex v3.2
- Security certificates
- An actual device!
Mobile-ready Flex

- Two finger tap to zoom out
- Double tap to zoom in
- Pinch close > out
- Pinch open > in
- Map.multiTouchEnabled (true)
- Map.multiTouchRotationEnabled (true)
Mobile application life-cycle events

Initialize – initializing internal components

CreationComplete – component has finished validating

ViewActivate – view is visible

Map.load – map object has finished loading

LayerEvent.load – layer has finished loading
Reacting to Device State

Event.DEACTIVATE
Event.ACTIVATE

NativeApplications.nativeApplication.addEventListener(
  Event.DEACTIVATE, handleDeactivate, false, 0, true);

public function handleDeactivate(event:Event) {
  geoLocation.removeEventListener(GeolocationEvent.UPDATE,
  geolocationUpdateHandler);
}


Maintain map state

View transitions, application open/close

Persistent
- SharedObject
- MySQL
- SD Card

Single session
- navigator.pushView();
- Singleton
Maintain map state

SettingsView

data.centerLat = 37.003;
data.centerLon = 104.012;
data.zoomLevel = 12;

navigator.pushView(views.MainView, data);

MapView

<s:VGroup>
  <s:Label text="{data.zoomLevel}"/>
  <s:Label text="{data.centerLat}"/>
  <s:Label text="{data.centerLon}"/>
</s:VGroup>
Demo – Application Life Cycle
Detecting Map Touch Events

Use cases

• Touch map to draw (point, line, polygon)
• Touch map to open graphic popup
• Touch map to detect closest graphic or feature
• Pinch zoom map
• Drag map
Demo – Detecting Touch Events
Working with GPS

- Use `flash.sensors.Geolocation.Geolocation()`
- Huge battery hog – use it sparingly
- GPS accuracy is affected by WiFi on/off
- Map layer must be loaded before drawing GPS graphic
- There may be delays in getting initial location
- Shut down GPS when app is minimized
Working with GPS

GeolocationEvent

lat
lon
heading
altitude
speed
timestamp
horizontalAccuracy
verticalAccuracy
Demo – GPS
Routing on Mobile

- Need starting and ending location
- Start location can be user entry or GPS
- Drawing the static route + textual directions
- Real-time routing
Demo – Real-time routing
Responsive design

- CSS/Media Queries
- Looks very similar responsive design in JavaScript
- Device detection and determining pixel density

```css
@media (application-dpi: 240) and (os-platform: "Android") {
  s|Button {
    color: 13;
  }
}
```
Demo – Responsive Design
Performance Considerations

- Be aware of sluggish UX
- Managing multiple layers on mobile
- Managing the size of map data accessed on the device
- Mobile skins vs non-mobile skins
- GPS can kill battery life
- Using Vector<> for feature manipulation vs Array
Futures

Test drive the new API components
- Directions
- Geocoder

Identify areas of improvements for future API releases

Provide a boilerplate application
- Map centric
- Nice looking
- Develop a set of skins optimized for mobile
- Smartphone and tablet
- Implement some functionalities
Tablet view
Smartphone
Responsive layout
Responsive layout
Demo – Futures
Lloyd Heberlie,
Produce Engineer
Flex API Team
lheberlie@esri.com
@lheberlie

Andy Gup,
Tech Lead for Web APIs, Android
Esri Developer Network
agup@esri.com
@agup

Yann Cabon
Developer
Flex API Team
@yanncabon
Understanding our world.