Android Programming
Lecture Catchup

10/12/2011
ItemizedOverlay

• To use an ItemizedOverlay, write your own class that:
  – Extends ItemizedOverlay<OverlayItem> instead of Overlay
    • An ItemizedOverlay holds multiple OverlayItems
  – Define Constructor:
    • Takes a Drawable component as a parameter to be used as the marker image for all items
    • If using Toasts, also takes a Context variable as a parameter
    • Chooses whether to have item slightly above target point or centered on target point (super(boundCenter(...)) or (super(boundCenterBottom(...)))
    • Creates and adds OverlayItems to a list of items to be shown in the layer
    • Calls populate() to process all items in the list
  – Implement public int size() so that it returns accurate number of markers to appear in the overlay
  – Implement OverlayItem createItem(int i) to return a reference to the ith OverlayItem out of the list
  – Implement boolean onTap(int i) to respond to a tap on the ith item
Drawable Resources

• Under the project res/ folder, create a new folder called drawable

• Enter the drawable/ folder and ask to Import from General, File System

• In Import dialog, navigate to folder containing images to use

• Choose images of interest

• Image types: PNG (preferred), JPG, GIF
  – Turned into bitmaps before drawn
Location Providers

• Almost every phone supports the following two providers
  – GPS:
    LocationManager.GPS_PROVIDER
  – Cell Network:
    LocationManager.NETWORK_PROVIDER

• Right now, assume we will use GPS
  – We will come back to a way to search for a provider
Choosing a Location Provider

- Can fill out a Criteria object, indicating features you need in your Location Provider
- Can also specify accuracy on bearing, altitude,…
- Would use the String returned in place of where hard-coded provider previously (see 2 slides back)

```java
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

Criteria providerCriteria = new Criteria();
providerCriteria.setAccuracy(Criteria.ACCURACY_COARSE); // or ACCURACY_FINE
providerCriteria.setPowerRequirement(Criteria.POWER_LOW); // or POWER_MEDIUM or POWER_HIGH
providerCriteria.setAltitudeRequired(false);
providerCriteria.setBearingRequired(false);
providerCriteria.setSpeedRequired(false);
providerCriteria.setCostAllowed(true);

String bestProvider = locationManager.getBestProvider(providerCriteria, true); // send in criteria, and a boolean (t/f) indicating
                        // whether only want enabled (turned-on) providers
Log.v("WakeLocator", "Using " + bestProvider + " provider!");
```

In the Emulator, under these criteria, returns “GPS” but on my Droid it returns “Network”
Disabled Providers?

• If you require a particular provider
  – Query the state of the provider
  – If disabled, prompt the user to turn on the provider

```java
locationManager = (LocationManager) getSystemService(Context.LOCATION_SERVICE);

boolean gpsAvailable = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);
if (gpsAvailable == false) {
    Toast toastToShow = Toast.makeText(this, "Please turn on your GPS\nand restart the app!", Toast.LENGTH_LONG);
    toastToShow.show();
}
```

At this point, just re-starting isn’t actually enough – they would need to force kill the app; we now need to start thinking about things such as:

- Turning off the GPS when the Activity is hidden
- Handling GPS events in more than just onCreat