GPS Drawing 2:
Android Location
Mobile and Ubiquitous Games
ICS 163
Donald J. Patterson
• How to get Location from Android in KitKat v4.4.2
• Walkthrough ——>
• To simplify your life, turn off rotation as a user of the phone
• Start your project as a new Android Application. This creates the template project

- Turn off rotation
- Template
- Fixed U/I
- Linked in "Play"
- Fix Up Manifest
  • Permissions
  • Plan Versioning
- Create Location Manager
- Callback Interface
- Connection Fail
- Google Play Issue
- What is my location?
- Continuos Callbacks
- Fix your U/I so that you have the fields and buttons that you care about

- Buttons
  - Static
  - Dynamic
    - On Create

- TURN OFF ROTATION
- TEMPLATE
- FIXED U/I
- LINKED IN "PLAY"
- FIX UP MANIFEST
  - Permissions
  - Play Versioning
- CREATE LOCATION MANAGER
- CALLBACK INTERFACE
- CONNECTION
  - FAIL
  - GOOGLE PLAY ISSUE
- WHAT IS MY LOCATION?
- CONTINUOUS CALLBACKS
• Remember the problems with updating the U/I on an arbitrary thread
• Remember the problems with updating the U/I on an arbitrary thread

```java
/* Queue a U/I update */
private void updateUI() {
    runOnUiThread(new Runnable(){
        @Override
        public void run() {
            if((PlaceholderFragment.locationView != null) && (lastLocation != null)){
                PlaceholderFragment.locationView.setText(lastLocation);
            }
        }
    });
}
```
GPS Drawing

- Notes on GPS Drawing assignment architecture
GPS Drawing

• Fix your U/I so that you have the fields and buttons that you care about
• Link in the Google Play Library as an Android Library
• Fix your manifest so that you have permissions for
  • Internet
  • Fine, Coarse Location
• Location Extras
GPS Drawing

- Fix your manifest so that
- You have linked the version of the google play library

```xml
<meta-data
    android:name="com.google.android.gms.version"
    android:value="@integer/google_play_services_version" />
```
GPS Drawing

• Create a Location Manager

```java
mLocationClient = new LocationClient(this, this, this);
```

• Implement the interfaces required for the above call to compile

```java
public class DevonIsAwesome extends Activity implements
GooglePlayServicesClient.ConnectionCallbacks,
GooglePlayServicesClient.OnConnectionFailedListener {
```
If the connection fails, then ask Android to fix it if it can.
If the connection fails, then ask Android to fix it if it can.

```java
@Override
public void onConnectionFailed(ConnectionResult result) {
    /*
     * Google Play services can resolve some errors it detects.
     * If the error has a resolution, try sending an Intent to
     * start a Google Play services activity that can resolve
     * error.
     */
    if (result.hasResolution()) {
        try {
            // Start an Activity that tries to resolve the error
            result.startResolutionForResult(this, CONNECTION_FAILURE_RESOLUTION_REQUEST);
            /*
             * Thrown if Google Play services canceled the original
             * PendingIntent
             */
        } catch (IntentSender.SendIntentException e) {
            // Log the error
            e.printStackTrace();
        }
    } else {
        /*
         * If no resolution is available, display a dialog to the
         * user with the error.
         */
        Dialog errorDialog = GooglePlayServicesUtil
            .getErrorDialog(result.getErrorCode(), this,
            CONNECTION_FAILURE_RESOLUTION_REQUEST);
        // If Google Play services can provide an error dialog
        if (errorDialog != null) {
            // Create a new DialogFragment for the error dialog
            ErrorDialogFragment errorFragment = new ErrorDialogFragment();
            // Set the dialog in the DialogFragment
            errorFragment.setDialog(errorDialog);
            // Show the error dialog in the DialogFragment
            errorFragment.show(getFragmentManager(), "Location Updates");
        }
    }
}
```
• If the connection works, then do what you need to

```java
/**
 * Called by Location Services when the request to connect the
 * client finishes successfully. At this point, you can
 * request the current location or start periodic updates
 */

@Override
public void isConnected(Bundle dataBundle) {
    // Display the connection status
    Toast.makeText(this, "Connected", Toast.LENGTH_SHORT).show();

    Location mCurrentLocation = mLocationClient.getLastLocation();
    lastLocation = "("+mCurrentLocation.getLatitude()+","+mCurrentLocation.getLongitude()+")";
    updateUI();
}
```
If the connection works, then do what you need to
• Other notes:

• Getting a hook to a U/I element for later updating

```java
/**
 * A placeholder fragment containing a simple view.
 */
public static class PlaceholderFragment extends Fragment {

    public static TextView locationView;

    public PlaceholderFragment() {
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                           Bundle savedInstanceState) {
        View rootView = inflater.inflate(
            R.layout.fragment_devon_is_awesome, container, false);

        locationView = (TextView) rootView.findViewById(R.id.editText1);

        return rootView;
    }
}
```