CS371m - Mobile Computing

Maps





Using Google Maps

- This lecture focuses on using Google Maps inside an Android app
- Alternatives Exist:
 - -Open Street Maps
 - -<u>http://www.openstreetmap.org/</u>
- If you simply want to display a "standard Google map" from your app ...



Clicker Question

- What Android component we can use from our app to display a map of a location?
- A. Activity
- B. Broadcast Receiver
- C. Content Resolver
- D. Intent
- E. Service

Displaying Standard Google Map

- Create URI based on desired location.
- From location sample:

```
// show current location on map
public void showMap(View view) {
    if (lastKnownLocation != null) {
        // Create a Uri from an intent string.
        // // Use last known location.
        double lat = lastKnownLocation.getLatitude();
        double lng = lastKnownLocation.getLongitude();
        String locationURI = "geo:" + lat + "," + lng;
        Uri uriForMappingIntent = Uri.parse(locationURI);
```

Create and Fire Intent

// Make the Intent explicit by setting the Google Maps package. // If want to use user's preferred map app, don't do this! mapIntent.setPackage("com.google.android.apps.maps");

// Attempt to start an activity that can handle the Intent
if (mapIntent.resolveActivity(getPackageManager()) != null) {
 startActivity(mapIntent);

Adjusting Zoom Level

- Specify Zoom Level for Google Map
- Zoom Levels in Google Maps, 0 21
 - 0 is the whole earth
 - 21 individual buildings

```
// show current location on map
public void showMap(View view) {
    if (lastKnownLocation != null) {
        // Create a Uri from an intent string.
        // // Use last known location.
        double lat = lastKnownLocation.getLatitude();
        double lng = lastKnownLocation.getLongitude();
        String locationURL = "geo:" + lat + "," + lng;
        locationURI += "?z=10";
        Uri uriForMappingIntent = Uri.parse(locationURI);
    }
}
```

Zoom Comparisons

zoom = 5

zoom = 10

zoom = 15



Searching for Locations and Label

 Can create URIs that search for a location and provide a label





INCORPORATING MAPS IN YOUR APPS

Using Google Maps

- Not standard Android
- Requires an API key from Google
- <u>https://developers.google.com/maps/do</u> <u>cumentation/android/</u>
- required to use MapView class or MapFragments
- Must add a reference to the Maps API Key in each MapView (xml or code)

Using Google Maps API v2

- For Android, Google Maps API v2 part of the Google Play Services sdk
- Download via SDK Manager

🛄 🙀 Google APIS	Э	Э	🐼 instaneu
📄 🧰 Extras			
🔄 💼 Android Support Library		21.0.3	👼 Installed
🔄 💼 Android Support Repository		11	😿 Installed
🔄 🔂 Google AdMob Ads SDK		11	👼 Installed
C Services for Froyo		1.00	Mot installed
Google Play services		22	😿 Installed
E Panacitany		15	
		F	- • • • •

Include Google Play Services in Manifest

- To make use of Google Play Services add data to manifest
- Google Play Services has a host of non standard android tools
 - -"simple location API"
 - "activity recognition"

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

Obtaining an API Key

Most web APIs require a key to use

-a few do not such as the Yahoo finance API

- Same with Google Maps API
- New way of obtaining keys via Android Studio and Google Developers console is mostly painless
- Old way of obtaining the key required some knowledge regarding how apps are published and was painful

New Way to Get Maps API Key

Add No Activity

 Create a Google Maps Activity in Android Studio





New Way to Get Maps API Key

 Look at google_maps_api.xml file in new project

N:---

TODO: Before you run your application, you need a Google Maps API key.

To get one, follow this link, follow the directions and press "Create" a

https://console.developers.google.com/flows/enableapi?apiid=maps android

Go to Developer Console

Register your application for Google Maps Android API in Google Developers Console

Google Developers Console allows you to manage your application and monitor API usage.

Select a project where your application will be registered

You can use one project to manage all of your applications, or you can create a different project for each application.

Create a new project

Continue

T.

Create Key

API	API Manager	Credentials
¢	Overview	4
0+	Credentials	Create Android API key
		Name Android key 1
		Restrict usage to your Android apps (Optional) Android devices send API requests directly to Google. Google verifies that each request comes from an Android app that matches a package name and SHA-1 signing-fingerprint name that you provide. Get the package name from your AndroidManifest.xml file. Use the following command to get the fingerprint. Learn more
		kevtool -list -v -keystore mystore.keystore
	API key	
	Here is your API	key
	AIzaSyDWtuMQON	1PTWWAajQz27BaUs

0K

Last Step

 Copy API Key into google_maps_api.xml file

nttps://aevelopers.google.com/maps/aocumentation/anaroia/sta

OLD PAINFUL WAY

Signing Apps

- deploying apps on the Google Play requires signing the app with a certificate
- development and debugging uses an automatic key creation process

- invisible to us

In release mode you must create your own private key to sign apps

– use of keytool program from Java SDK

 <u>http://developer.android.com/guide/publishi</u> <u>ng/app-signing.html</u>

Signing Apps

- A Java Keystore is a file (or files) that stores security certificates
- Included in the JDK (Java Development Kit) is the keytool program
- Used to create manipulate the keystore

:) Program Files	Java	▶ jdk1.7.0 ▶ bin		
Share with 🔻	Burn	New folder		
Name	^		Date modified	Туре
💷 jstatd.exe			9/6/2011 10:38 AM	Application
ど jvisualvm.exe			9/6/2011 10:38 AM	Application
💷 keytool.exe			9/6/2011 10:38 AM	Application

Signing Apps via Android Studio

- Android Studio provides a GUI to run keytool for you
- Build ->
 Generate
 Signed APK

Key store path: C:\User	s\scottm\android_keystore.jks ····								
Password:	••••• Co <u>n</u> firm: •••••••••								
Key									
<u>A</u> lias: Hell	o Android								
Password:	••••• <u>C</u> onfirm: ••••••								
Validity (years):	.5								
Certificate									
First and Last Name:	Michael Scott								
Organizational Unit:	Department of Computer Science								
Organization:	University of Texas at Austin								
City or <u>L</u> ocality:	Austin								
State or Province:	ТХ								
Country Code (<u>X</u> X):	01								
L									

keystore file

0:	FE	ED	FE	ED	00	00	00	02	00	00	00	01	00	00	00	01	þíþí
10:	00	$0 \mathrm{D}$	68	65	6C	6C	6 F	20	61	6E	64	72	$6 \mathrm{F}$	69	64	00	hello android.
20:	00	01	4B	CC	5B	B7	$\mathbf{F0}$	00	00	05	00	30	82	04	\mathbf{FC}	30	KÌ[·ð0∎.ü0
30:	0E	06	0 A 0	2B	06	01	04	01	2A	02	11	01	01	05	00	04	+ *
40:	82	04	E8	85	46	05	$\mathbf{B0}$	CC	67	8 0	BD	Α4	F6	40	25	B6	I.èIF.°Ìg.½¤ö@%¶
50:	B3	1D	A6	\mathbf{BF}	94	03	52	7A	31	8 0	B2	68	50	5 F	34	\mathbf{FB}	³.¦¿∎.Rz1.²hP_4û
60:	9 F	72	D7	0E	1B	75	3A	7A	72	A9	\mathbf{BE}	22	D5	4 F	D5	A2	∎r×u∶zr©¾"ÕOÕ¢
70:	A6	32	BC	AΒ	44	39	00	4 F	11	43	89	55	04	38	9B	24	¦2¼«D9.0.C∎U.8∎\$
80:	λA	89	11	B7	A6	\mathbf{EF}	E3	C9	E6	48	9B	10	27	B2	Α4	04	ª∎.·¦ïãÉæH∎.'²¤.
90:	AC	97	73	3A	D0	1D	7F	76	\mathbf{FC}	22	3D	A7	λÀ	9C	49	27	∽∎s:Đ.∎vü"=§ª∎I'
A0:	CD	$\mathbf{B0}$	\mathbf{BE}	B0	CA	1F	83	5C	81	72	66	E5	A5	F9	1F	B3	Ͱ¾°Ê.∎∖rfạ¥ù.³
B0:	67	16	A5	9C	$\mathbf{F0}$	FA	43	49	27	3B	DD	E0	$\mathbf{B4}$	6A	50	3E	g.¥∎ðúCI';Ÿà́jP>
C0 :	80	91	10	\mathbf{FC}	E3	03	29	29	34	$\mathbf{B8}$	9E	A1	D9	34	41	C5	Į́,üã.))4,∎iÙ4AÅ
D0 :	CC	A9	DD	66	B7	48	63	1E	ÀΕ	66	E2	35	32	DA	21	29	Ì@Ÿf∙Hc.@fâ52Ú!)
E0:	F2	D3	A6	48	F8	B0	CB	F7	CE	F9	В9	$1\mathrm{B}$	81	5 C	A0	01	òÓ¦Hø°Ë÷Îù¹.∖ .
F0:	74	44	F7	D8	DB	58	Α4	B5	CD	96	0C	$\mathbf{B}\mathbf{A}$	F8	47	7 F	14	tD÷ØÛX¤µÍ∎,ºøG∎.
100:	86	AF	AF	90	D5	9A	49	5D	24	20	9 F	C8	88	E5	38	A8	∎O∎I]\$ ĮĖ∎å8"
110:	9D	0 D	6B	DD	\mathbf{BE}	E7	9B	E5	AD	AB	C0	AD	F0	E2	33	79	.kݾç∎å–«Å–ðâ3y
120:	61	4B	79	9D	29	31	25	28	41	ΑE	6A	51	72	0D	32	E7	aKy)1%(A®jQr.2ç
130:	8A	9C	1F	B8	5B	\mathbf{BF}	B4	66	5E	C0	$1\mathrm{B}$	\mathbf{EB}	9A	C5	3B	AE	∎∎.,[¿´f^Å.ë∎Å;®
140:	1C	DC	C4	F9	8E	B7	\mathbf{BF}	7B	C4	\mathbf{DF}	18	3D	E9	\mathbf{FE}	5F	C3	.ŪĀù∎·¿{ĀB.=éþ_Ā
150:	6C	D4	43	D8	74	F6	95	EE	31	6B	15	53	C4	DD	D0	80	lÔCØtö∎î1k.SÄŸÐ∎
160:	D3	87	5E	1A	98	B4	04	19	7C	3E	02	0E	48	56	65	32	Ó∎^.∎´ >HVe2
170:	$0 \mathrm{F}$	54	56	40	CD	C4	8E	AC	98	97	43	A2	3A	7D	DE	A7	.TV@ÍÄ∎¬∎∎C¢:}Þ§
180:	7F	F9	C5	3D	46	3C	13	39	82	D8	6E	03	15	DD	CD	CD	∎ùÅ=F<.9∎ØnÝÍÍ

Obtaining an API Key

- For Google Maps API v2
- One key tied to one signing certificate
- Same key used for all instances of app
- Normally sign apps with different certificates
- If so different API keys required if two different apps use maps and signed with different certificates

Obtaining an API Key

- To obtain a Google Maps API key we need the SHA-1 fingerprint of the signing certificate
- A short form of the certificate based on the SHA-1 hashing algorithm
- run keytool from the command line to pull out fingerprint of certificate

fingerprint via keytool

prompt>keytool -v -keystore
 <keystore_file_name> -alias
 <certificate_name> -storepass
 <keystore_password> -keypass
 <certificate_password>

C:\Users\scottm>keytool -list -v -keystore android_keystore.jks -alias s Alias name: sample_certificate Creation date: Feb 27, 2015 Entry type: PrivateKeyEntry Certificate chain length: 1 Certificate[1]: Owner: CN=Michael Scott, OU=Computer Science Department, O=University of Issuer: CN=Michael Scott, OU=Computer Science Department, O=University o Serial number: 6eee9bb2 Valid from: Fri Feb 27 12:57:48 CST 2015 until: Sat Feb 14 12:57:48 CST Certificate fingerprints: ØF:E8:90:9E:32:23:AE:1A:CC:10:F2:71:B9:38:91:69 MD5 : SHA1: E5:07:2B:D9:CA:C7:4A:4B:5A:A4:B6:AC:60:49:A4:06:4E:31:F3: SHA256: 73:63:A0:25:94:DE:2B:4A:0B:AF:B9:5C:82:60:8E:F0:CA:2A:C Signature algorithm name: SHA256withRSA **Version:** 3

debug certificate

- In development you are using a pre generated debug keystore to sign apps
- Happens behind the scenes
- Security settings on device



debug certificate

- possible to obtain API key tied to your debug keystore
- works in development
- would need to change manifest with certificate used to sign the app

Obtaining the API key

- The SHA-1 fingerprint is a 20 digit hexadecimal number
- Use <u>Google APIs console</u> to obtain key for Maps
 - requires Google account
- ... and must agree to the terms of service.

Using Maps API Key in App

Must add key to manifest

```
<meta-data
android:name="com.google.android.maps.v2.API_KEY"
android:value="API_KEY"/>
```

 KEY POINT: Replace API_KEY in the second line only (android:value) with the API key you obtained in previous steps

DISPLAYING A MAP INSIDE YOUR APP

Google Maps Terms of Service

5 5 5 5 5 5

Google Maps/Google Earth APIs Terms of Service

Thank you for your interest in the Google Maps/Google Earth APIs. The Google Maps/Google Earth APIs are a collection of services that allow you to include maps, geocoding, places, and other Content from Google in your web pages or applications.

Last Updated: January 23, 2017

- Some Highlights
 - may include ads in future
 - Google may limit number of transactions
 - You will not use the Service or Content for or in connection with (a) real-time navigation or route guidance; or (b) automatic or autonomous vehicle control.
 - Must include Google Play Services Attribution in your apps "legal notices"

Permissions

Recommended Permissions for manifest when using Google Maps inside your app

Display Simple Map in App

- Hello Map
- Like Hello World, but layout file becomes:

 More on Fragments later, but they are a UI component between Activities and GUI Widgets

Hello Map Activity

• Lots of new classes!!

import android.app.Activity; import android.os.Bundle; import com.google.android.gms.maps.CameraUpdate; import com.google.android.gms.maps.CameraUpdateFactory; import com.google.android.gms.maps.GoogleMap; import com.google.android.gms.maps.MapFragment; import com.google.android.gms.maps.model.BitmapDescriptorFactory; import com.google.android.gms.maps.model.BitmapDescriptorFactory; import com.google.android.gms.maps.model.LatLng; import com.google.android.gms.maps.model.Marker; import com.google.android.gms.maps.model.Marker; import com.google.android.gms.maps.model.MarkerOptions;

Specifying Locations

• Latitude and Longitude

public class SimpleMapActivity extends Activity {

static final LatLng AUSTIN = new LatLng(30.287, -97.737);
static final LatLng ARLINGTON = new LatLng(32.751, -97.083);
private GoogleMap map;

GoogleMap object Used for most interactions with map.
onCreate for Simple Map App

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity_simple_map);

MapFragment mapFrag =

((MapFragment) getFragmentManager()

.findFragmentById(R.id.map));

More on Fragments later ...

```
mapFrag.getMapAsync(new OnMapReadyCallback() {
    @Override
    public woid onMapReady(GoogleMap.go
```

```
public void onMapReady (GoogleMap googleMap) {
    map = googleMap;
```

```
setMarkers();
```

```
moveCamera();
```

adding Markers in onCreate

first Marker uses default, pin and has a title, "Austin"

 second Marker uses a different icon and adds text after the title

Center and Zoom

- Running app as is produces this:
- Centered where???
- Zoomed out
- Recall, zoom levels 0 to 21



CameraUpdate center=

CameraUpdateFactory.*newLatLng*(**AUSTIN**); CameraUpdate zoom=CameraUpdateFactory.*zoomTo*(5);

map.moveCamera(center);
map.animateCamera(zoom);





Resources for Working with Google Maps

 <u>https://developers.google.com/maps/</u> <u>documentation/android-api/intro</u>

Google Maps APIs



Stack Overflow Ask a question under the google-maps tag.



GitHub Fork our samples and try them yourself.



Blog Read the latest updates, customer stories, and tips.



Issue Tracker Something wrong? Send us a bug report!

Map Options

- Adding UI controls
- Handling user gestures
- Handling user events
- move and zoom camera
- Draw on the map
 - markers, information windows, shapes, overlays
- Heatmaps, Marker Clusters

Heatmaps



Marker Clusters

- Too many markers can clutter the display
- Especially on smaller displays
- Marker clusters used to group together markers until user zooms in



The Map Object

OLD SLIDES PRE GOOGLE MAPS API 2

Debug Key

• Portion of debug.keystore

/ debug.keystore - Hex

0: FE ED	FE ED (00 00 00	02 00	00 0	D 01	00	00	00	01	þíþí
10: 00 OF	61 6E 6	64 72 6F	69 64	64 6	5 62	75	67	6B	65	androiddebugke
20: 79 00	00 01 3	32 40 69	7D D2	00 0	0 02	BC	30	82	02	y2@i}Ò¼O∎.
30: B8 30	OE 06 (DA 2B 06	01 04	01 2.	A 02	11	01	01	05	,0+ *
40: 00 04	82 02 4	A4 72 C5	15 FB	DB 8	3 59	66	FD	EC	56	∎.¤r <u>A</u> .ûU∎YfýìV
50: 9B 6B	2A 92 9	92 12 AF	76 64	EF 6	4 D8	10	C2	FE	7C	[k*´´. vdïdØ.Aþ
60: CF 1D	1C 8D H	FF E7 98	9B F5	48 4	5 9F	E5	14	E5	9F	I.yçioHLia.a
70: 11 DE	96 9D 5	5D 93 A8	9F 37	EC 4	5 71	D3	A6	7A -	DD	.P[] [/iLqO¦zY
80: 00 90	DD E9 E	BF DF 57	28 CE	B5 2	J 30	86 1 D	UD ED	38	48	.YechW(lµ Ull.;H
90: 77 95	6/ UA 4	AB DU 32 AD 10 35	L8 4E	99 8 20 E	9 Z3	AD 24	FB CO	/8	00	WIG. (BZENII#-ux.
AU: DB CC	A/ 1/ 4	4D IC AF SF SD SF	AU 4B	20 E	9 F9 - 40	21	5U ЭБ	57 410	05	UIS.M. Keu! Y.
DU: 00 7A	0D D4 3	36 30 86 No to ta	LD 21	50 D'	- 4F / DE	20	2E 00	4D 70		$4 + - 1 \text{or} \hat{T} \cdot T 0 \cdot 1 \text{om}$
CU: 34 /4 DO: 96 11		υο Γ7 34 λ1 ΓΓ λί	02 CE DE 70	F/ 3 D0 7	4 DF N EC	00 00	ጋጋ	20	4D 55	4001.01.1÷15,∎un ∎ C∵i≜li≂1≂⊆∎úz^
FO: 90 7F		RO 11 R2	31 93	37 B	5 JJ 7 1 A	56	96	65	28	ειλ *λ21 72 ι=/
FO: 31 A5	3D F4 3	27 FF 33	49 29	0,10	2 ፲፫ 7 ፑፚ	73	99	BΔ	17	1¥=37b3T@/∩ú⊊∎Չ
100 69 EB	EC 70 T	D2 59 CE	74 59	EC 0	Å 06	4E	83	$\frac{2R}{2R}$	71	iëìnÒVÎzVì N∎+a
110: E2 5E	OE FO O	DE 03 24	35 ČB	98 Å	3 51	52	25	50	ĒĈ	â^.ð * 5Ё∎€OR%Pì
120: 4C A2	98 DO I	D5 A3 DA	00 58	25 C	Ă D9	FĈ	F1	ĀĒ	3F	Lo ĐÔ£Ú X%ÊÙuñ ?
130: 13 94	6B 59 1	14 42 5A	E9 82	89 7	3 B3	49	C8	63	8D	. kY.BZé s'IÈc
140: 6E 25	23 3E 0	C9 A8 CC	36 E3	C6 3	5 9E	4F	07	4F	EΒ	n%#>ɨÌ6ãÆ5∎0.0ë
150: AO D8	0E 1D 1	1A EC 88	0A D4	69 2	1 2D	B6	87	67	E9	ØìĮ.Ôi!—¶Įgé
160: 5A 36	43 02 4	4E 57 AA	2E 49	9A 3	2 F4	5D	75	58	4 A 👘	Z6C.NWª.I∎2ô]uXJ
170: A2 9F	9D 5B 5	56 70 A5	47 8D	55 3	174	49	DA	F6	6B	¢∎[Vp¥GU1tIÚök
180: DC 3E	E6 B1 (00 FO 7F	E3 43	30 E	2 8E	72	4 A	DA	36	Ü>æ±.ð∎ãCOå∎rJÚ6
190: 3B 6A	BD 8C (DC F5 4C	50 1E	OC 5	2 36	CB	54	31	B7	;j½∎.õLP.,R6ET1•
1AO: 27 F9	10 90 I	D8 3A 9B	FE BC	70 C	C 51	1B	2A	38	A3	'ù.Ø:∎þ¼pIQ.*8£
1B0: 5C CE	FC 72 H	F3 88 32	4D B5	48 C	3 38	BB	64	FD	6E	∖Iuró∎2MµHA8≫dýn
1CU: UE 73	<u>93 70 E</u>	<u>EB 9D 7B</u>	35 5B	B2 1	<u>р В1</u>	<u>C8</u>	ΕĒ	E4	ΒB	.slpe{5[².±Epa»

47

Getting MD5 Fingerprint

- use keytool program
- keytool part of Java SDK
- keytool -list -alias androiddebugkey -keystore

<path_to_debug_keystore>.keystore
-storepass android -keypass android

- gives MD5 fingerprint of the debug certificate
- keytool of Java 1.7 gives SHA1 by default — use -v after keytool, before -list

Debug API Key

Thank you for signing up for an Android Maps API key!

Your key is:

OceOzsn6afbLp8R8ZR1_9rizJTjrJIoa4w_VkCQ

This key is good for all apps signed with your certificate whose fingerprint is:

DF:6E:BD

0C:98:B3

Here is an example xml layout to get you started on your way to mapping glory:

<com.google.android.maps.MapView android:layout_width="fill_parent" android:layout_height="fill_parent" android:apiKey="OceOzsn6afbLp8R8ZR1_9rizJTjrJIoa4w_VkCQ" />

Hello MapView

• Build Target - Google, not Android

Android 2.3.3	Android Open Source Project	2.3.3	10
📝 Google APIs	Google Inc.	2.3.3	10
FDK 1 1	Sony Friesson Mobile Communicati	222	10

MapView not a standard Android class

 part of Google Maps Library
 add to manifest

<uses-library android:name="com.google.android.maps" />

 must also include INTERNET permission and LOCATION permission

Aside - Permissions

<u>http://developer.android.com/reference/</u> <u>android/Manifest.permission.html</u>

ACCESS_CHECKIN_PROPERTIES	Allows read/write access to the "properties" table in the checkin databa
ACCESS_COARSE_LOCATION	Allows an application to access coarse (e.g., Cell-ID, WiFi) location
ACCESS_FINE_LOCATION	Allows an application to access fine (e.g., GPS) location
ACCESS_LOCATION_EXTRA_COMMANDS	Allows an application to access extra location provider commands
ACCESS_MOCK_LOCATION	Allows an application to create mock location providers for testing
ACCESS_NETWORK_STATE	Allows applications to access information about networks
ACCESS_SURFACE_FLINGER	Allows an application to use SurfaceFlinger's low level features
ACCESS_WIFI_STATE	Allows applications to access information about Wi-Fi networks
ACCOUNT_MANAGER	Allows applications to call into AccountAuthenticators.
ADD_VOICEMAIL	Allows an application to add voicemails into the system.
AUTHENTICATE_ACCOUNTS	Allows an application to act as an AccountAuthenticator for the Account

MapView

A type of view for layout file

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/mainlayout"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent" >
<com.google.android.maps.MapView
```

```
android:id="@+id/mapview"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:clickable="true"
android:apiKey="Your Maps API Key"
/>
```

</RelativeLayout>

MapActivity

- Create class that extends MapActivity instead of Activity
- import com.google.android.maps.MapActivity;
- must implement isRouteDisplayed method

```
@Override
protected boolean isRouteDisplayed() {
    return false;
}
```

 must return true if any kind of route (to be followed) is displayed, per terms of use

Instance Vars and onCreate

 Add instance variables and initialize in onCreate method

private LinearLayout linearLayout;
private MapView mapView;

/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);

setContentView(R.layout.main);
mapView = (MapView) findViewById(R.id.mapview);
mapView.setBuiltInZoomControls(true);

HelloMapView

- Run app
- Displays map and allows panning and zooming



Customizing Map

- Easy to display map and allow interaction
- Customize with markers and overlays
- Overlays
 - -used to display information on top of map
 - -simple choice: ItemizedOverlay class

ItemizedOverlay

public class HelloItemizedOverlay extends ItemizedOverlay {

```
private ArrayList<OverlayItem> mOverlays;
```

```
public HelloItemizedOverlay(Drawable defaultMarker) {
    super(boundCenterBottom(defaultMarker));
    mOverlays = new ArrayList<OverlayItem>();
}
```

```
public void addOverlay(OverlayItem overlay) {
    mOverlays.add(overlay);
    // inherited method to prepare overlays to be drawn
    populate();
}
```

ItemizedOverlay

- populate method will call createItem
- define createltem and return value from the ArrayList instance var
- define size method that returns number of overlay items

```
@Override
protected OverlayItem createItem(int i) {
    return mOverlays.get(i);
}
@Override
public int size() {
    return mOverlays.size();
}
```

Adding Overlays

- In MapActivity create OverlayItem
- add to HelloItemizedOverlay
- add to MapView
- Need a drawable for the marker
 - -res/drawable
 - issues display gif
 format images
 on some devices



Changes to HelloMapView

```
private LinearLayout linearLayout;
private MapView mapView;
private List<Overlay> mapOverlays;
private Drawable drawable;
private HelloItemizedOverlay itemizedOverlay;
```

```
/** Called when the activity is first created. */
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
```

```
setContentView(R.layout.main);
mapView = (MapView) findViewById(R.id.mapview);
mapView.setBuiltInZoomControls(true);
```

```
mapOverlays = mapView.getOverlays();
drawable = this.getResources().getDrawable(R.drawable.Longhorn);
itemizedOverlay = new HelloItemizedOverlay(drawable);
```

Add Overlay Items

- Create GeoPoint and use these to create OverlayItems
- GeoPoint based on microdegrees
 lat and long times 1,000,000
- Build OverlayItems out of GeoPoints and include strings for title and snippet text to display when drawable clicked

addOverlays method in HelloMapView

```
private void addOverlays() {
    int[] parlin = {(int) (30.284882 * 1e6), (int) (-97.740127 *1e6)};
    int[] mcdonald = {(int) (30.671581 * 1e6), (int) (-104.022431 *1e6)};
    int[] marine = {(int) (27.835926 * 1e6), (int) (-97.050372 *1e6)};
    int[][] points = {parlin, mcdonald, marine};
    String[][] titlesAndSnippets = {{"UT",
        "Parlin Hall at The University of Texas at Austin"},
        {"McDonald",
            "McDonald Observatory - University of Texas - West Texas"},
        {"Marine Biology",
                "University of Texas Marine Sciennce Institure - Port Aransas"}
    for(int i = 0; i < points.length; i++) {</pre>
        GeoPoint g = new GeoPoint(points[i][0], points[i][1]);
        OverlayItem oi = new OverlayItem(g,
                titlesAndSnippets[i][0], titlesAndSnippets[i][1]);
        itemizedOverlay.addOverlay(oi);
    }
    mapOverlays.add(itemizedOverlay);
}
```

Result

- one overlay with multiple items
- based on locations we added



Display Information

 To display information (title and snippet) of overlay override the onTap method in the ItemizedOverlay class

```
@Override
protected boolean onTap(int index) {
    OverlayItem item = mOverlays.get(index);
    AlertDialog.Builder dialog = new AlertDialog.Builder(mContext);
    dialog.setTitle(item.getTitle());
    dialog.setMessage(item.getSnippet());
    dialog.show();
    return true;
```

Results of Clicking Longhorn



Reverse Geocoding

- Find addresses from longitude/latitude
- Geocoder uses a backend that is NOT included in the core android framework
- use isPresent method to check for service

```
location = locationManager.getLastKnownLocation(
                               LocationManager.GPS_PROVIDER);
double lat = location.getLatitude();
double lng = location.getLongitude();
```

```
Geocoder gc = new Geocoder(this, Locale.getDefault());
List<Address> addresses = null;
```

try {

```
addresses = gc.getFromLocation(lat, lng, 5); // maxResults
} catch (IOException e) {}
```

Forward Geocoding

 Find longitude/latitude (and more) from address or airport code

```
Geocoder gc = new Geocoder(this, Locale.US);
 List<Address> addresses = null;
 try {
  addresses = gc.getFromLocationName(
              "713 N. Duchese, St., Missouri", 5);
 } catch (IOException e) {}
 double lat = addresses.get(0).getLatitude();
 double lng = addresses.get(0).getLongitude ();
 String zip = addresses.get(0).getPostalCode();
               Geocode is present: true
HelloMapView
                lat: 38.7991079, long: -90.494416, zip: 63301
HelloMapView
```

Recent Changes

- Google Maps API version 2
 - somewhat new, Released December 2012
 as part of Google Play Services SDK
- features:
 - -indoor maps
 - -simplified location services

Maps Example

- Route Tracker using Locations, MapActivity, MapView, and Google Maps
 - -from Deitel AFP-AADA
- Similar to Map My Ride
 - popular app among
 cyclists and runners



RouteTracker App





RouteTracker App

- using FrameLayout to stack components with the most recently added component on top
- ToggleButton at bottom to start and stop route tracking
- MapView added to FrameLayout
- route is an overlay to map with points and lines connecting points



RouteOverlay

Overlay with location points (every 10th) and lines connecting. Converts locations to GeoPoints. **Overloads draw**
Criteria Class

Set criteria for selecting a LocationProvider

```
Criteria criteria = new Criteria();
criteria.setAccuracy(Criteria.ACCURACY_FINE);
criteria.setBearingRequired(true);
criteria.setCostAllowed(true);
criteria.setPowerRequirement(Criteria.POWER_LOW);
criteria.setAltitudeRequired(false);
```

```
locationManager =
  (LocationManager) getSystemService(LOCATION_SERVICE);
```

```
// get the best provider based on our Criteria
String provider = locationManager.getBestProvider(criteria, true);
```

GpsStatus.Listener

- Responds to changes in GPS status
- Are we receiving GPS fixes?
- App does not track unless this is true

```
// determine whether we have GPS fix
GpsStatus.Listener gpsStatusListener = new GpsStatus.Listener() {
    public void onGpsStatusChanged(int event) {
        if (event == GpsStatus.GPS_EVENT_FIRST_FIX) {
            gpsFix = true;
            Toast results = Toast.makeText(RouteTracker.this,
               getResources().getString(R.string.toast_signal_acquired),
                Toast.LENGTH_SHORT);
        results.setGravity(Gravity.CENTER,
                results.getXOffset() / 2, results.getYOffset() / 2);
        results.show();
        }
    };
```

Simulating GPS Data

 to simulate changes in location in emulator

- GPS data in a file
 - -GPS Exchange Format (GPX)

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<gpx xmlns="http://www.topografix.com/GPX/1/1" xmlns:gpxtrx="http://</pre>
    \langle trk \rangle
        <name>Position 1 to Position 517
        <trkseg>
            <trkpt lat="42.37123092799624" lon="-71.43667976096056">
                <ele>1.0</ele>
                <time>2011-07-02T20:01:38.000Z</time>
                <name>Position 1</name>
                <sat>7</sat>
            </trkpt>
            <trkpt lat="42.37123092799624" lon="-71.43667976096056">
                <ele>1.0</ele>
                <time>2011-07-02T20:01:39.000Z</time>
                <name>Position 2</name>
                <sat>7</sat>
                <extensions>
                    <nmea:speed>0.0</nmea:speed>
                </extensions>
            </trkpt>
            <trkpt lat="42.37123092799624" lon="-71.43667976096056">
                <ele>1.0</ele>
                <time>2011-07-02T20:01:40.000Z</time>
                <name>Position 3</name>
                <sat>7</sat>
                <extensions>
                    <nmea:speed>0.0</nmea:speed>
                </extensions>
            </trkpt>
```

Creating GPX Files

- Many apps and programs
- One option for Android devices
- GPSLogger
- gpsbabel to convert
 between



various GPS formats

-gpx has different versions

Running GPX files in App

- DDMS
- Emulator Control Tab
- GPX Tab
- Load

Location Contro Manual GPX Load GPX	ls KML			
Name	Longitude	Latitude	Elevat	Description

Running GPX

Location Cont	rols				
Manual GP	X KML				
Load GPX]				
Name Longitu		le Latitude		Elevat	De
Name		Poi First Point		Time	La
Position 517		517	Tue Aug 0	Tue Aug 02 15:01:3	
	Sne	ed: 1X	2		



Geocoding

- Finding addresses from lat / long and vice versa
- Reverse geocoding: find address from lat and long
- Forward geocoding: find lat and long from address