

# Mobila applikationer och trådlösa nät

HI1033

Lecture 8

Today's topics

- Location Based Services
- Google Maps API
- MultiMedia




# Location Based Services

- LocationManager provides access to location based services
- Location providers, providing different technologies for location finding
  - Global Positioning System, GPS
  - Cell ID
  - WiFi
- - Power use – GPS expensive
  - Speed
  - Accuracy
  - Bearing
  - Cost

Sony Ericsson



 Location consent

Allow Google's location service to collect anonymous location data. Collection will occur even when no applications are running.

Agree

Disagree

XPERIA



# Location Based Services, components

- LocationManager
  - provides access to Android system location services
- manager = (LocationManager)  
ctx.getSystemService(*Context.LOCATION\_SERVICE*);
- LocationListener
  - interface for receiving notifications from the location manager on location changed
- Location
  - representation of a geographic location, plus a time stamp

# Location Based Services, settings

- In AndroidManifest.xml  
`<uses-permission  
android:name="android.permission.ACCESS_FINE_LOCATION"/>`  
(or `ACCESS_COARSE_LOCATION`)

- Define accuracy and power requirements to decide which provider to use

```
Criteria criteria = new Criteria();  
criteria.setAccuracy(Criteria.ACCURACY_FINE);  
criteria.setPowerRequirement(Criteria.POWER_MEDIUM);  
criteria.setAltitudeRequired(false);  
criteria.setCostAllowed(false);
```

- `provider = locationManager.getBestProvider(criteria, true);`

# Listening for location updates

- Implement a LocationListener

```
class LocationListenerExample implements LocationListener {  
  
    @Override  
    public void onLocationChanged(Location location) {  
        ... // calculate, update ui  
    }  
  
    @Override  
    public void onStatusChanged(String provider, int status, Bundle  
        extras) { ... }  
  
    @Override  
    public void onProviderEnabled(String provider) { ... }  
  
    @Override  
    public void onProviderDisabled(String provider) { ... }  
}
```

# Listening for location updates

- Start, *and stop*, receiving location information -  
manager.requestLocationUpdates(...)  
- manager.removeLocationUpdates(...)
- Example (in Activity)

```
protected void onResume() {  
    super.onResume();  
    locationManager.requestLocationUpdates(  
        bestProvider, 5000, 10,  
        locationManager);  
}
```

```
protected void onPause() {  
    super.onPause();  
    locationManager.removeUpdates(locationListener);  
}
```

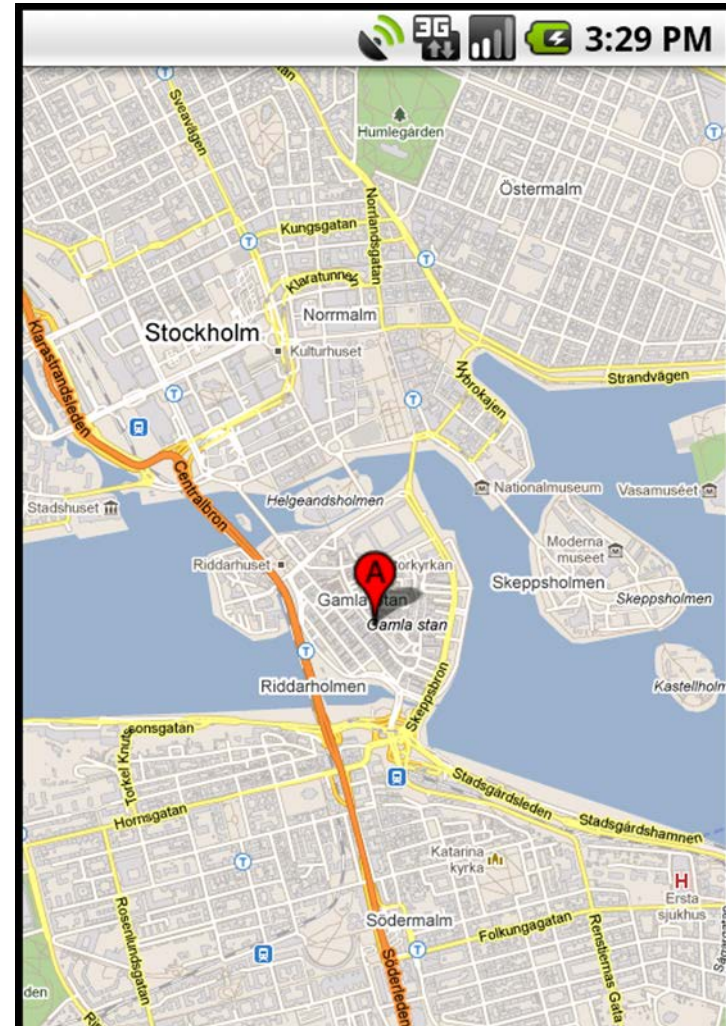
# Geo-coding

- android.location.Geocoder translates between longitude/latitude and “addresses”
- Lookups are done at a (Google-) server, hence requires android.permission.INTERNET
- *Should be performed in the background*
- Forward: address to coordinates
- Reverse: coordinates to address
- ```
addresses = geocoder.getFromLocation(  
    location.getLatitude(),  
    location.getLongitude(), 5);
```



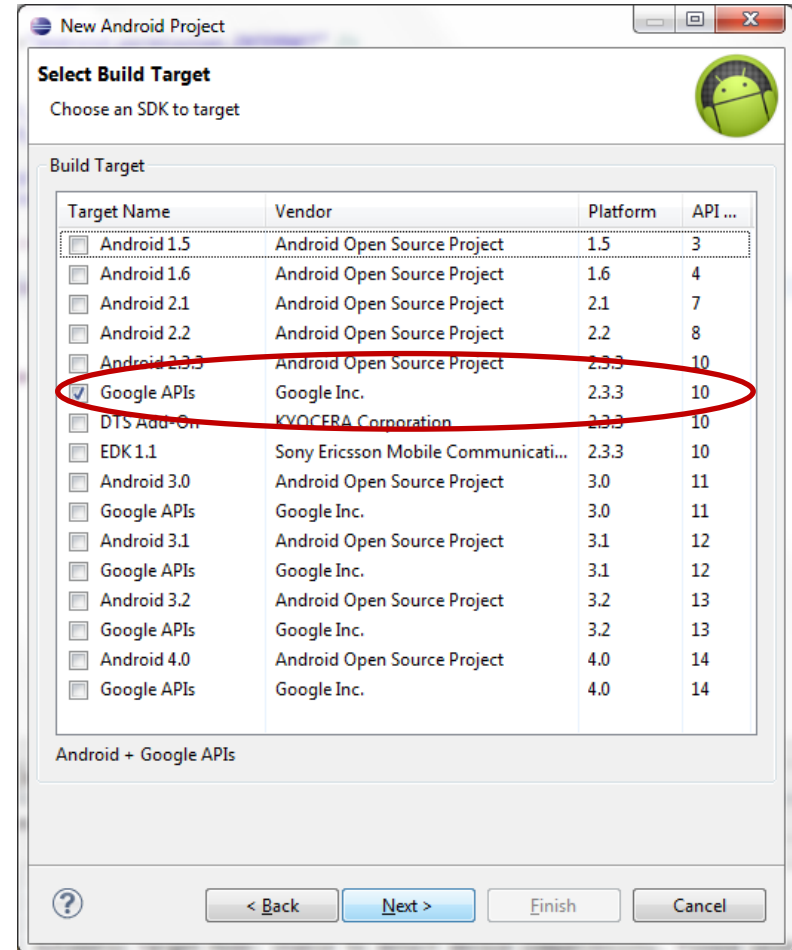
# MapActivity and MapView

- MapView – UI for presenting geographical data
- MapActivity handles application life cycle and background service required when displaying maps
- Interactive:
  - Overlay
  - MyLocationOverlay
  - MapController



# MapActivity and MapView

- Not a standard Android API – requires Google APIs
- uses-library:  
android:name="com.google.android.maps"
- uses-permission:  
android.permission.INTERNET



# MapActivity and MapView

- API key required to embed Google Maps in your application
- Instructions:  
<http://code.google.com/intl/sv-SE/android/add-ons/google-apis/mapkey.html>  
*NB: Make sure you get the MD5 fingerprint (use keytool -v -list ...)*
- Different keys for development/debugging and release
- Add the API key in the layout file  

```
<com.google.android.maps.MapView  
...  
    android:apiKey="01I..." />
```

# MapActivity and MapView

```
public class HelloGoogleMapsAct extends MapActivity {  
  
    private MapView mapView;  
    private HelloItemizedOverlay itemizedOverlay;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState) {  
        ...  
        setContentView(R.layout.main);  
  
        mapView = (MapView) findViewById(R.id.mapview);  
        mapView.setBuiltInZoomControls(true);  
    }  
}
```

# MapController

- Example, zoom in and center

```
GeoPoint point =  
    new GeoPoint(59324082, 18071136);  
MapController mapController =  
    mapView.getController();  
mapController.setZoom(15); // 1-21  
mapController.setCenter(point);
```

# Overlays

```
public class HelloGoogleMapsActivity extends MapActivity {  
    ...  
    private HelloItemizedOverlay itemizedOverlay;  
  
    private void addOverlayItem() {  
        GeoPoint point = new GeoPoint(59324082, 18071136);  
        OverlayItem item = new OverlayItem(  
            point, "Hello, World!", "Welcome to Stockholm!");  
        itemizedOverlay.addOverlayItem(item);  
    }  
    ...  
}
```

- Requires:

```
public class HelloItemizedOverlay extends  
    ItemizedOverlay<OverlayItem> { ...
```

# Finding the location/maps

- LocationBasedServices.zip
- HelloGoogleMaps.zip
- Location Based Services in Android:  
<http://developer.android.com/guide/topics/location/obtaining-user-location.html>
- A good start with maps:  
<http://developer.android.com/resources/tutorials/views/hello-mapview.html>



Devices

| Name               | State  |
|--------------------|--------|
| emulator-5554      | Online |
| system_process     | 64     |
| com.android.phone  | 119    |
| android.process.ac | 143    |
| com.android.alarm  | 170    |

Emulator Control

Location Controls

Manual GPX KML

Decimal  
 Sexagesimal

Longitude:

Latitude:

Threads Heap Allocation Tracker File Explorer

| Name   | Size | Date       | Time  | Permissions | Info |
|--------|------|------------|-------|-------------|------|
| data   |      | 2011-02-28 | 09:39 | drwxrwx--x  |      |
| sdcard |      | 2011-02-28 | 09:39 | d---rwxr-x  |      |
| system |      | 2009-12-19 | 02:17 | drwxr-xr-x  |      |

LogCat Console

| Time           | pid   | tag      | Message                                              |
|----------------|-------|----------|------------------------------------------------------|
| 02-28 10:10... | D 422 | jdwp     | JDWP shutting down net...                            |
| 02-28 10:10... | I 422 | dalvikvm | Debugger has detached; object registry had 1 entries |

Filter:



# Simulating location in the emulator

1. In Eclipse, open the Dalvik Debug Monitor Service, DDMS, perspective
2. In the Emulator Control pane, scroll to Location Control
3. Enter *longitude and latitude* and send
  - ...or use a predifened file (GPX or KML)
  - ...or, use the console:  
telnet localhost 5554  
> geo fix <*longitude*> <*latitiude*> [<altitude>]

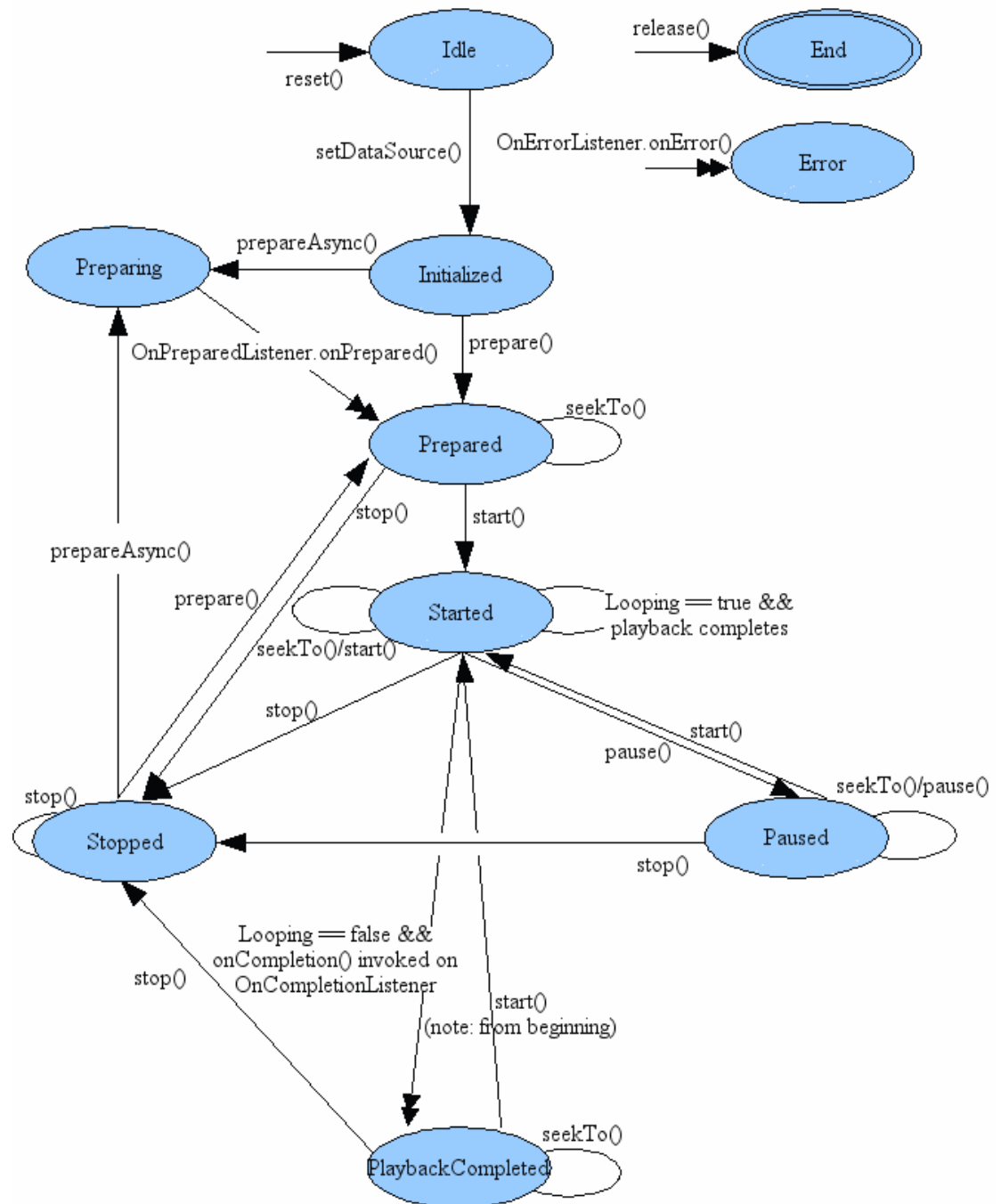
# Playing Audio and Video

- Supported audio formats:  
MP3, MIDI, Ogg Vorbis, PCM/WAVE, ...
- Supported video formats:  
MPEG-4, H.263, H.264 AVC
- Playback - MediaPlayer class  
Recording – MediaRecording class
- Play from
  - application resources (res/raw)
  - local files
  - Content Providers, such as the media store
  - network URL

# Playing Audio and Video

Playback control of audio/video files and streams is managed as a state machine

[single arrow head represent synchronous method calls, double arrow head represent asynchronous method calls]



# Playing Audio

- From resources:
- ```
Context appctx = getApplicationContext();  
player = MP.create(appCtx, R.raw.my_audio);
```
- Player is already prepared
  
- From a URI, e.g. via a ContentProvider:
- ```
Uri myUri = .....; // initialize Uri here  
MediaPlayer mediaPlayer = new MediaPlayer();  
mediaPlayer.setAudioStreamType(  
    AudioManager.STREAM_MUSIC);  
mediaPlayer.setDataSource(appctx, myUri);  
mediaPlayer.prepare();  
mediaPlayer.start();
```

# Playing Audio

- From a URL:
- ```
String url = "http://....."; // your URL here
MediaPlayer mediaPlayer = new MediaPlayer();
mediaPlayer.setAudioStreamType(
    AudioManager.STREAM_MUSIC);
mediaPlayer.setDataSource(url);
mediaPlayer.prepare(); // might take long
mediaPlayer.start();
```
- When no longer needed:  

```
player.release();
player = null;
```

# Playing video using VideoView

- Encapsulates a MediaPlayer

- ```
VideoView videoView = (VideoView)
    findViewById(R.id.video_view);
```

```
videoView.setKeepScreenOn(true);
videoView.setVideoPath("/sdcard/test2.3gp");
```

```
if (videoView.canSeekForward()) {
    videoView.seekTo(videoView.getDuration()/2);
}
videoView.start();
// [ ... ]
videoView.stopPlayback();
```

# Other media topics

- MediaPlayer –  
<http://developer.android.com/guide/topics/media/mediaplayer.html>
- Recording audio and video -MediaRecorder
- Camera, taking pictures, preview, ...
- Adding media to the media store
- Raw audio manipulation
- Speech recognition

# HTML5

- Google's CEO Eric Schmidt on HTML5:
- "It's taken 20 years because the underlying standards had to evolve from the proprietary Mac and Windows platforms. But they have both adopted HTML5 as a future direction. So that means in future a lot of apps will be running in HTML5 in both mobile and non-mobile form."





# HTML5

- Currently under development
- New syntactical features. These include the <video>, <audio>, and <canvas> elements
- Integration of Scalable Vector Graphics, SVG, content
- Drag-and-drop
- Full-scale application development for the browser possible
- Apps might be platform agnostic?
- Android comes with a HTML5 Webkit-based browser