

Multimedia

CS 2046

Mobile Application Development

Fall 2010



Announcements

- Assignment 3 is on CMS.
 - Due Friday, 11/19.
- Last lecture: Friday, 11/12.
 - How to release your apps on Android Market
 - Compatibility with older devices
 - Free vs. ad-supported business models
 - Integrating advertising
- Office Hours next week:
 - Jeff: MF 11:15 - 12:15
 - Jae: W 12 - 1



Intro of the Day - Maps API

- External library bundled with Android
 - As with Geocoder class, need to set the build target and emulator to “Google APIs”, add `<uses-library>`.
 - Need Internet permission for downloading map data.
 - In addition, if using MapView, need an API key from Google.
 - <http://code.google.com/android/add-ons/google-apis/mapkey.html>
- Main class is MapView
 - Displays Map, handles pan, zoom
 - Can control programmatically, draw overlays.



MapView Example

- Once you get an API key, can specify MapView with:

```
<com.google.android.maps.MapView  
    android:id="@+id/map"  
    android:clickable="true"  
    android:layout_width="fill_parent"  
    android:layout_height="fill_parent"  
    android:apiKey="..." />
```



ItemizedOverlay

- Extend ItemizedOverlay to lay items on top of map.

```
private class ExampleItemizedOverlay
    extends ItemizedOverlay<OverlayItem> {
    private ArrayList<OverlayItem> mOverlays =
        new ArrayList<OverlayItem>();
    public ExampleItemizedOverlay(Drawable defaultMarker) {
        super(boundCenter(defaultMarker));
    }
    public void addOverlay(OverlayItem overlay) {
        mOverlays.add(overlay);
        populate();
    }
    protected OverlayItem createItem(int i) {
        return mOverlays.get(i);
    }
    public int size() {
        return mOverlays.size();
    }
}
```



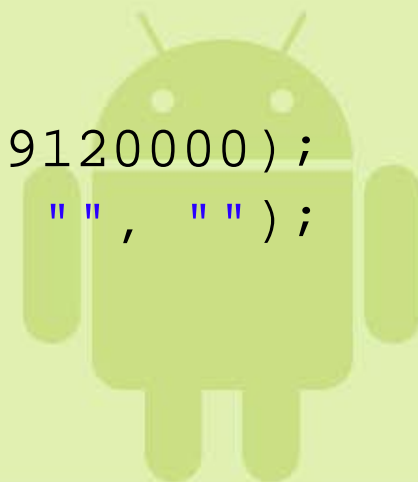
MapActivity

- Your Activity extends MapActivity, not Activity.
 - Must override isRouteDisplayed()
 - Whether you are displaying a route (informational)

```
MapView map = (MapView) findViewById(R.id.map);  
map.setBuiltInZoomControls(true);
```

```
ExampleItemizedOverlay overlay = new  
    ExampleItemizedOverlay(  
        getResources().getDrawable(R.drawable.icon));
```

```
GeoPoint point = new GeoPoint(19420000, -99120000);  
OverlayItem item = new OverlayItem(point, "", "");  
overlay.addOverlay(item);  
map.getOverlays().add(overlay);
```



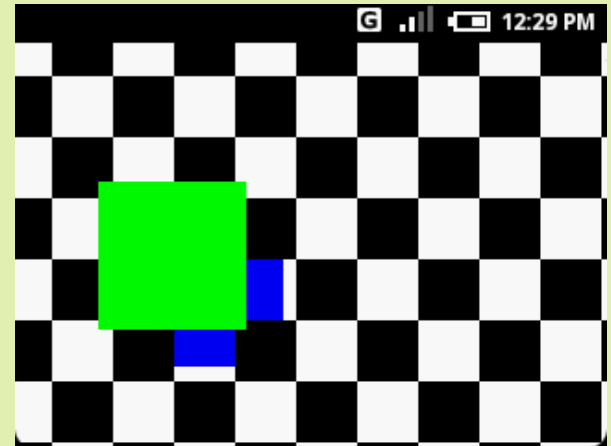
MapView Summary

- Plenty more can be done with MapViews
 - More types of Overlays
 - Clickable for more information
 - Controlling the location that appears on screen
- Tutorial:
<http://developer.android.com/guide/tutorials/views/hello-mapview.html>
- Documentation:
<http://code.google.com/android/add-ons/google-apis/reference/com/google/android/maps/MapView.html>
 - Not part of Open Source Android



Camera

- `android.hardware.Camera`
 - Permission: `android.permission.CAMERA`
- Emulator: essentially, non-functioning.
 - Preview - moving block
 - Taking a picture always results in the same (non)picture.



- Untested – abstraction interface:
<http://www.tomgibara.com/android/camera-source>
Uses webcam on emulator.



Preparing to Take Pictures

- Obtain instance of Camera with `Camera.open()`.
 - Locks camera, prevents other applications from using.
 - May take long time to complete – use worker thread.
- Set the preview display with `setPreviewDisplay`
 - Requires a `SurfaceHolder` to render previews onto.
 - Start with API Demos,
`com.example.android.apis.graphics.CameraPreview`
- Call `startPreview()` to begin preview
 - Preview must be running to take photos



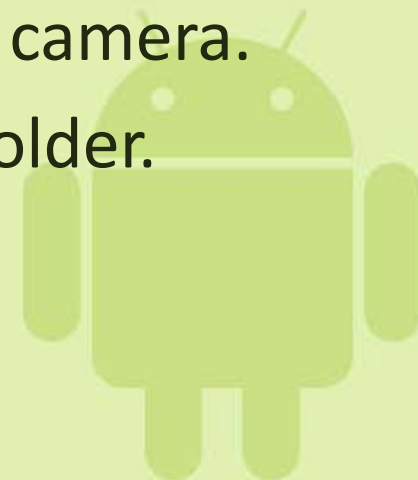
Taking Pictures

- Call `takePicture()` to capture a photo.
 - Capture `KeyEvent.KEYCODE_CAMERA`
 - Common version takes three arguments:
 - `Camera.ShutterCallback`
 - Triggered when image has been captured
 - Typical use – play a shutter sound
 - `Camera.PictureCallback`
 - Occurs when RAW image data is available
 - `Camera.PictureCallback`
 - Occurs when JPEG compressed image is available



After Taking Pictures

- Preview display will stop
 - If you want to take more pictures, call `startPreview()`
 - Call `stopPreview()` when exiting.
- Call `release()` to release camera.
 - Should (also) be done in `onPause()`
 - (Also) call `open()` in `onResume()` to reopen camera.
 - API Demo handles this as part of `SurfaceHolder`.



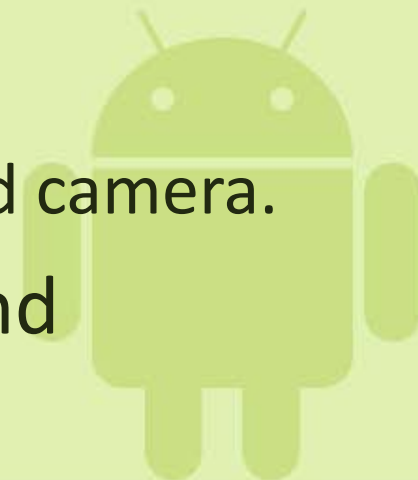
Configuring Camera

- Get default settings with `getParameters`
 - Returns `Camera.Parameters` object
 - Can modify and set with `setParameters`
 - Flash, Exposure, JPEG quality, Zoom
 - All dependent on hardware
 - Call `getSupported*()` functions before relying on them.
- Set the orientation with `setDisplayOrientation`
 - For portrait-mode applications



Recording Video

- Uses MediaRecorder class
 - General recording of audio and/or video
 - Does not work on emulator.
- As before, obtain Camera instance, start preview.
- Call unlock() to allow recorder to access Camera.
- Use MediaRecorder to take video
 - Call setCamera to point it to your initialized camera.
- Call reconnect() to reacquire camera, and stopPreview()/release() when finished.



MediaPlayer

- Play audio and video from:
 - Res/raw folder
 - Files in internal or external storage
 - Streams (over internet connections)
- Audio plays over standard output device
 - Speaker or headset

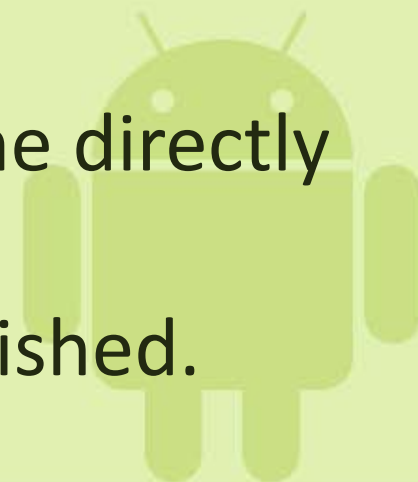


Playing Audio

- Raw resource:

```
MediaPlayer mp = MediaPlayer.create(context, R.raw.sound);  
mp.start();  
...  
mp.stop();
```

- If stopped, must call `mp.reset()` and `mp.prepare()` before calling `mp.start()` again.
- Can pause with `mp.pause()`, then resume directly with `mp.start()`.
- Make sure to call `mp.release()` when finished.



Playing File or Stream

- Can pass a Uri object pointing to a local file or internet stream.
 - Works for HTTP/RTSP streams
 - Works for certain file formats – search for “progressive download” to see how.



Playing Video

- Works essentially the same as audio
 - Exception – need to copy resource to file first.
- Addition – pass a SurfaceHolder on which player can render the video.
 - Just like Camera's preview window
 - SurfaceView's getHolder() will work
- More complete sample in [API Demos](#)
- Aside – copying files to emulator SD card
 - <http://deltafalcon.com/2010/04/mounting-an-android-emulator-sd-card-image-in-windows/>



Media Events

- For synchronous failures, MediaPlayer can throw an exception.
 - Example – format not supported.
- For asynchronous failures or updates, MediaPlayer supports various listeners for interfacing with the player.
 - OnErrorListener
 - OnBufferingUpdateListener



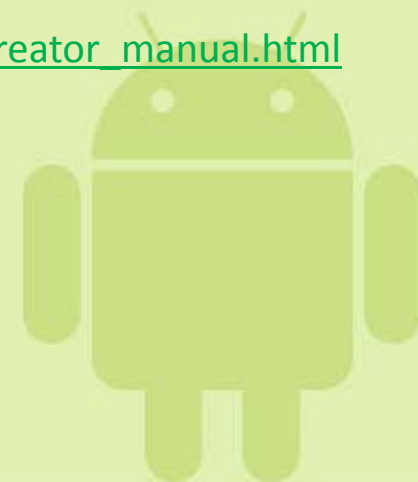
Supported Formats

- Core Formats supported on all devices
 - Device can choose to support additional formats, but should not rely on this.
 - Some may not have been added until Android 2.2 – be sure to test on older devices/emulators.
- Formats are complicated
 - In general:
 - .3gp , .mp4.m4a
 - Just containers – actual format may differ
 - .mp3, .ogg, .wav audio
 - jpg, gif, png, bmp images
- Can decode all of these, but only a few can be encoded (created):
 - .3gp audio, JPEG images, H.263 video



JET

- Create interactive soundtracks for games.
 - Designed for mobile platforms
 - MIDI format
 - Example: Super Mario World
- Uses free JET Creator
 - Manual: http://developer.android.com/guide/topics/media/jet/jetcreator_manual.html
- Play using JetPlayer class
 - Example: JetBoy in <sdk-dir>/samples



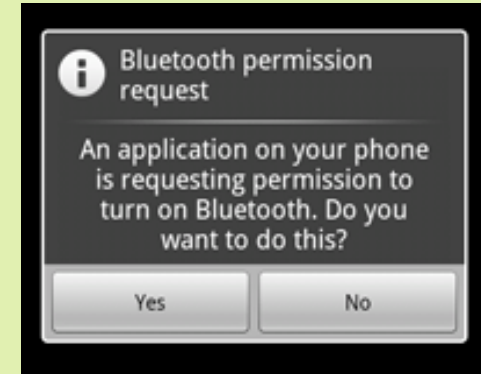
Bluetooth

- Wirelessly exchange data with other Bluetooth devices.
- Typical example – local communication between phones.
 - From first lecture: Bump (<http://bu.mp/>)
 - Simpler example – Bluetooth Chat
 - <http://developer.android.com/resources/samples/BluetoothChat/index.html>
- Uses `android.permission.BLUETOOTH`
 - Need `BLUETOOTH_ADMIN` for device discovery
- Unsupported on emulator.



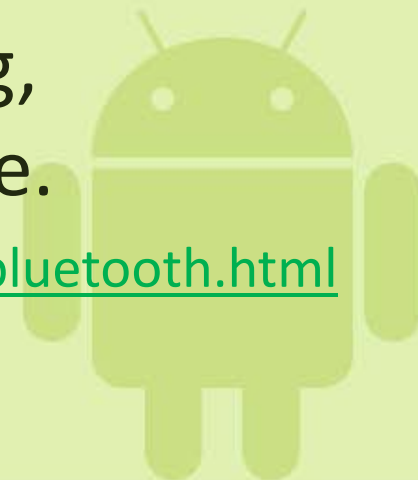
Using Bluetooth (High-level)

- Call `BluetoothAdapter.getDefaultAdapter()`
 - If null – device does not support default
 - Check returned `BluetoothAdapter.isEnabled()`
 - If false, can request user enable with `subIntent`
- Call `getBondedDevices()` to obtain any paired devices, or `startDiscovery()` to discover unpaired devices.
 - `startDiscovery()` will fire asynchronous `ACTION_FOUND` broadcasts.



Using Bluetooth (High-level)

- With `BluetoothDevice`, acquire `BluetoothSocket` for communication.
- With connection, can open `InputStream` and `OutputStream` for reading and writing data.
- Many more details to Bluetooth (pairing, connections) than can be explained here.
 - <http://developer.android.com/guide/topics/wireless/bluetooth.html>



OpenGL

- Android supports OpenGL ES API
 - ES = Embedded Systems
 - Version 1.0, corresponds to OpenGL 1.3
 - Some support for OpenGL ES 2.0 since Android 2.0
- Writing OpenGL code is out of the scope of this class.
- At a high level:
 - Create a GLSurfaceView on which to render
 - Call setRenderer on GLSurfaceView
 - Renderer has onDrawFrame method
- For overview:
<http://developer.android.com/guide/topics/graphics/opengl.html>

