



# **BEGIN ANDROID JOURNEY IN HOURS**

**CS425 / CSE 424 / ECE 428 [Fall 2009]**

**Sept. 14, 2009  
Ying Huang**

# REFERENCE

- Online development guide
  - <http://developer.android.com/guide/index.html>
- Book resource
  - “Professional Android Application Development”, by Reto Meier, (Wrox, [amazon link](#))
  - “Android A programmers guide”, by J.F. DiMarzio, (McGraw Hill, [amazon link](#))
  - “Beginning.Android”, by Mark L. Murphy, (Apress, [amazon link](#))
  - “Pro Android”, by Sayed Y. Hashimi, Satya Komatineni, (Apress, [amazon link](#))



# MOBILE OS

- Symbian
- iPhone
- RIM's BlackBerry
- Window mobile
- Linux
- Palm webOS
- Android
- ....

symbian  
OS





# WHAT IS ANDROID?

- *Google → OHA (Open Handset Alliance)*
  - *The first truly **open** and **comprehensive** platform for mobile devices, all of the software to run a mobile phone but without the **proprietary obstacles** that have hindered mobile innovation.*
  - Linux OS kernel
  - Java programming
  - Open source libraries: SQLite, WebKit, OpenGL





# WHY ANDROID

- A simple and powerful SDK
- No licensing, distribution, or development fees
- Development over many platform
  - Linux, Mac OS, windows
- Excellent documentation
- Thriving developer community
  
- For us
  - Java-based, easy to import 3<sup>rd</sup> party Java library
  - Funding (40+ G1 phones)
  - Prize (amazon's kindle)
  - Job opportunity



# ANDROID SDK FEATURE



- GSM, EDGE, and 3G networks, WiFi, Bluetooth
  - ~~API Support for Bluetooth, WiFi Ad hoc mode~~
- Libraries
  - Media, SQLite, WebKit, SSL
- **Hardware** control: **MP3**
  - Accelerometer, compass, microphone, camera, GPS
  - touch screen, power
- Location-based service, map (Google API) **MP3**



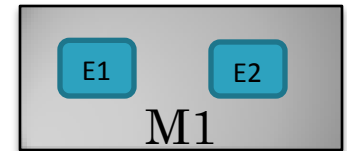
# TOOLS

- The Android Emulator
  - Implementation of the Android virtual machine
  - Test and debug your android applications.
- Dalvik Debug Monitoring Service ([DDMS](#))
  - Monitor and Control the Dalvik virtual machines
  - Logcat (see logged msgs)
- Android Debug Bridge ([ADB](#))
  - Manage the state of an emulator instance or Android-powered device
  - Copy files, install compiled application packages, and run shell commands.
- [Traceview](#)
  - Graphical analysis tool for viewing the trace logs from your Android application
  - Debug your application and profile its performance
- MkSDCard **MP2**
  - Creates an SDCard disk image



# RUN TWO NETWORKING EMULATORS IN A COMPUTER (UPDATED) - ADB

- Q: run two networking emulators in a computer A using the public IP address of A, during debugging and demo?



- A1: `telnet + redir` (MP1 doc)

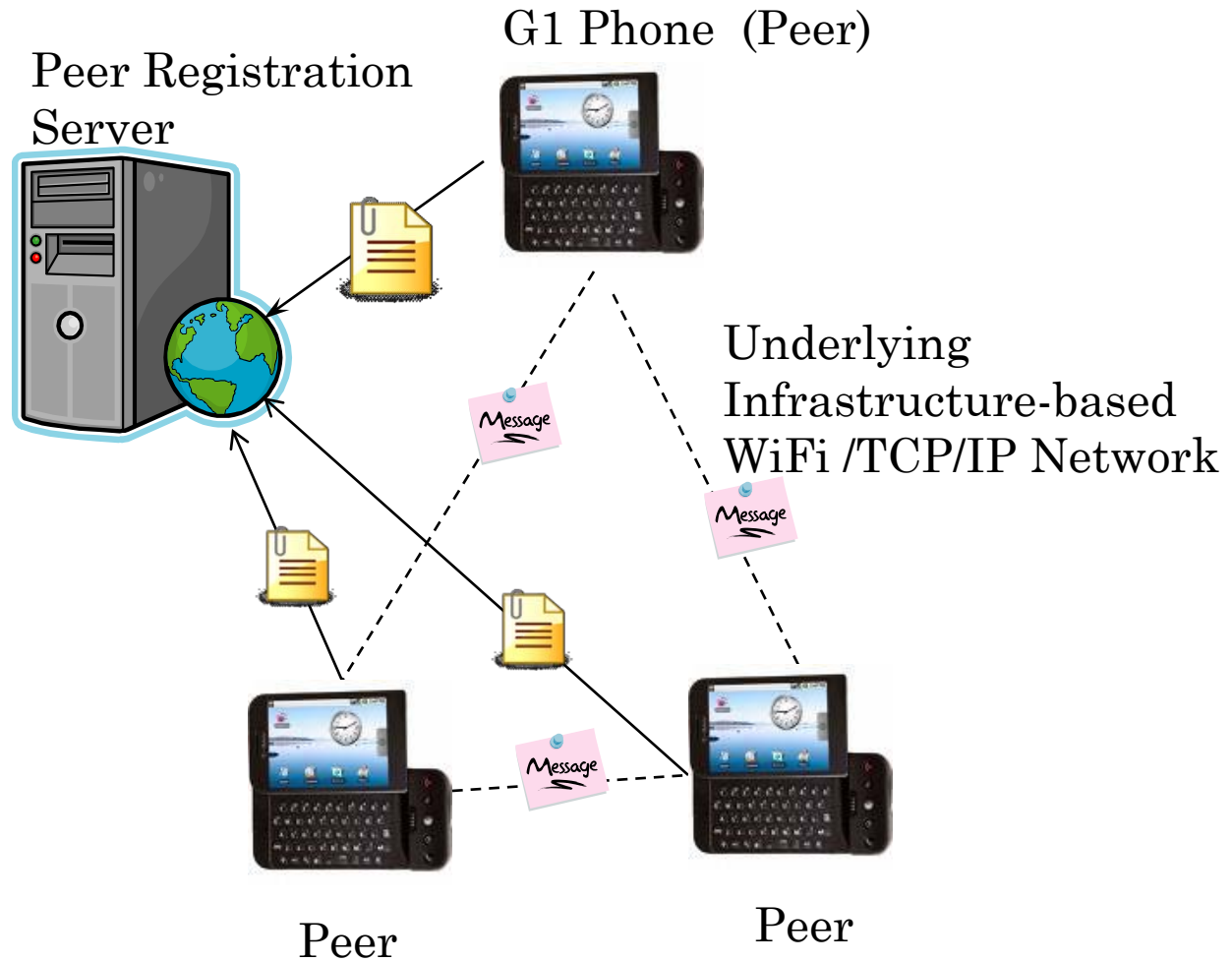
- A2: `adb forward`

- 1) Port forward to connect Android from localhost
  - `abd -s emulator-5554 forward tcp:15216 tcp:15216`
- 2) Use a proxy server which can listen on `my_public_ip:15216` and forward the data to `localhost:15216`
  - `stcppipe localhost 15216 15216`

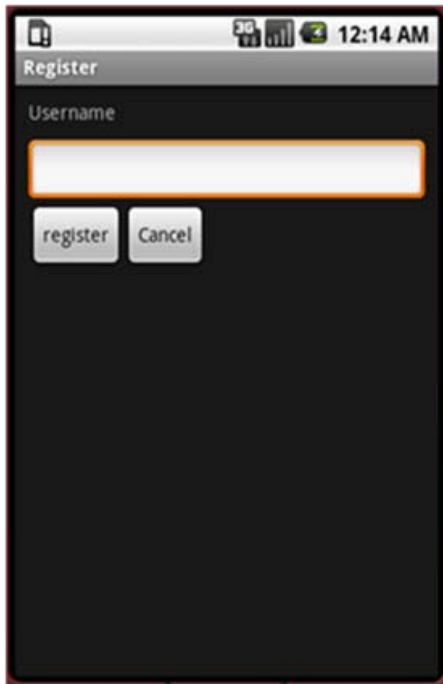




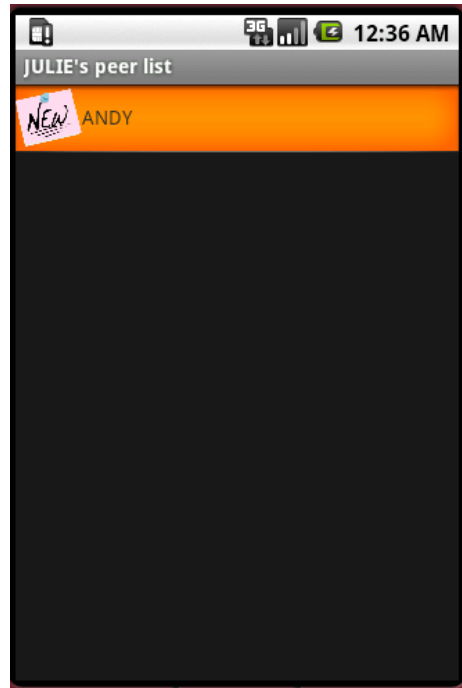
# MP1



# MP1 PROJECT STRUCTURE



Registration



PeerList

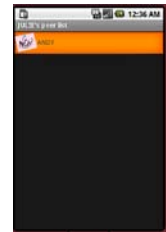


Messaging



# ANDROID APPLICATION ARCHITECTURE

- Views:
  - Building block for user interface components.
- Activities
  - A single, focused thing that the user can do.
  - Interaction with users: creating a window to place **UI**
  - full-screen windows, floating windows, embedded inside of another activity
  - **Ex: Registration, Peerlist, Messaging GUI**



# ANDROID APPLICATION ARCHITECTURE

- Services (Background)

- Ex: Network Operation

- Intent

- Inter-communication among activities or services

- Resource

- Externalization of strings and graphics

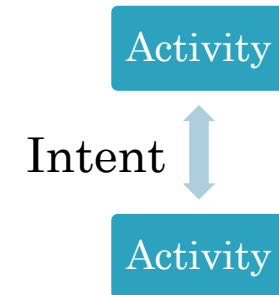
- Notification

- signaling users: Light, sound, icon, dialog, notification

- Ex: new message arrives

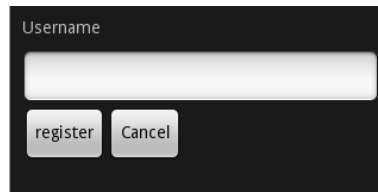
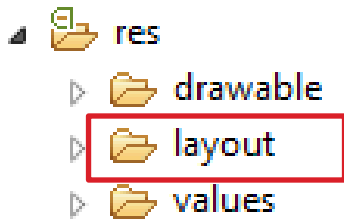
- Content Providers

- share data between applications



# VIEW

## Layout of visual interface



## Java Code

- Initialize

@Override

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.screen);
}
```

- Access

```
TextView myTextView =
    (TextView)findViewById(R.id.myTextView);
```

screen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com
/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
<TextView
    android:id="@+id/myTextView"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Hello World, HelloWorld"
/>
</LinearLayout>
```



# VIEW COMPONENT

## Widget Toolbox

- TextView, EditText, Button, Form, TimePicker...

- ListView (PeerList)

- Update list by arrays

- ArrayAdapter

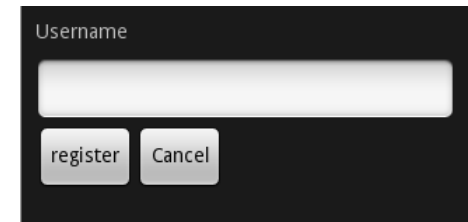
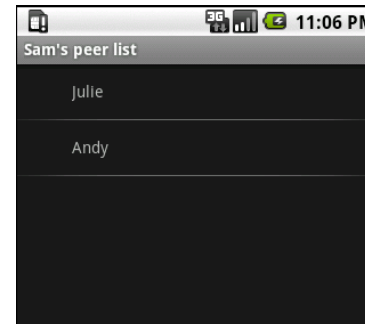
- `myListView.setAdapter`

- Layout

- Positions of controls

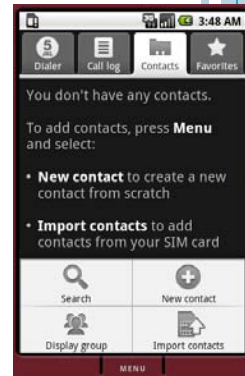
- LinearLayout, RelativeLayout

- <http://developer.android.com/guide/tutorials/views/index.html>



## Menu

- Exit app



# ACTIVITY

- **Foreground Activity:** suspended when invisible

- Visual, interactive
- Ex: Game, Map

Registration

Peer List

Messaging

- **Background Service:** Little interaction

- Ex: Hardware, power management

Network  
Operation  
Management

- **Intermittent Activity**

- Notification, service
- Expects some interactivity but does most of its work in the background.



# USER INTERACTION EVENT

- onKeyDown. onKeyUp
- onTrackBallEvent
- onTouchEvent

```
registerButton.setOnClickListener(new OnClickListener() {  
    public void onClick(View arg0) {...}}
```

```
myEditText.setKeyListener(new KeyListener() {  
    public boolean onKey(View v, int keyCode, KeyEvent event) {  
        if (event.getAction() == KeyEvent.ACTION_DOWN)  
            if (keyCode == KeyEvent.KEYCODE_DPAD_CENTER)  
            {  
                ...  
                return true;  
            }  
        return false;  
    });}
```





# APPLICATION AND COMPONENT GLUES

- An intent is an abstract description of an operation to be performed.

- Launch an activity

- Explicit

```
Ex: Intent intent = new  
Intent(MyActivity.this, MyOtherActivity.class);
```

```
Im: Intent intent = new  
Intent(Intent.ACTION_DIAL,  
Uri.parse("tel:555-2368"));
```

- Implicit: Android selects the best

- `startActivity()`;

- Subactivity: feedback

- Child: use intent as feedback, `setResult`

- Parent: `onActivityResult`

- `startActivityForResult`

- Action, data, extra parameter

- `intent.putExtra(name, property)`;



# INTENT (CNTD.)

## ○ Broadcast

- announce application events system-wide
- sendBroadcast
- MyBroadcastReceiver extends BroadcastReceiver
- registerReceiver (in java / in xml)

## ○ Intent Filter

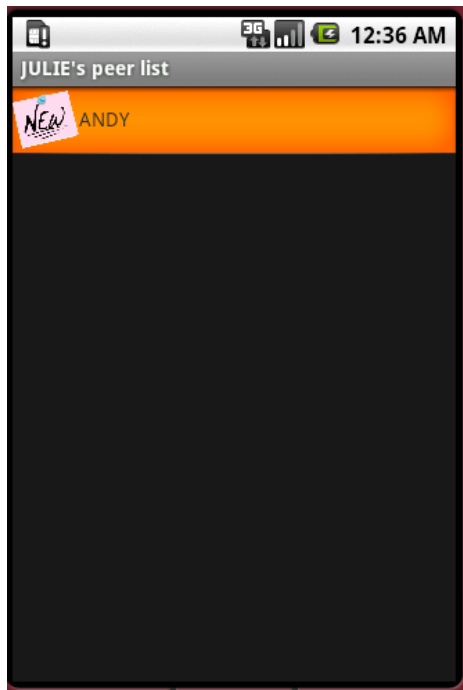
- Register Activities, Services, and Broadcast Receivers as being capable of performing an action on a particular kind of data.

## Manifest.xml

```
<activity ...>
  <intent-filter>
    <action
      android:name="com.paad.earthquake.inten
t.action.SHOW_DAMAGE">
    </action>
    <category
      android:name="android.intent.category.DE
FAULT"/>
    <category
      android:name="android.intent.category.AL
TERNATIVE_SELECTED"/>
    <data
      android:mimeType="vnd.earthquake.cursor
.item/*"/>
    </intent-filter>
  </activity>
```



# INTENT FROM PEERLIST TO MESSAGING



PeerList

Select a peer  
Send Intent →



Messaging



# WORKING IN BACKGROUND

## ○ Services

- NO GUI, **higher** priority than inactive Activities
- Usage:
  - responding to events, polling for data, updating Content Providers.
- However, all in the main thread

## ○ Background threads



# SERVICE

- Service class
  - public class MyService extends Service
  - public void onStart() {...}
- Manifest.xml
  - `<service android:enabled="true" android:name=".MyService"></service>`
- Control
  - startService
  - stopService
- Communication
  - Bind service with activity: use public method and properties
  - Intent



# THREADING

## ○ *Being Responsive (1sec)*

- Respond to any user action within 5 seconds.
- A Broadcast Receiver must return within 10 seconds.

## ○ What to thread?

- Network, file IO, Complex processing

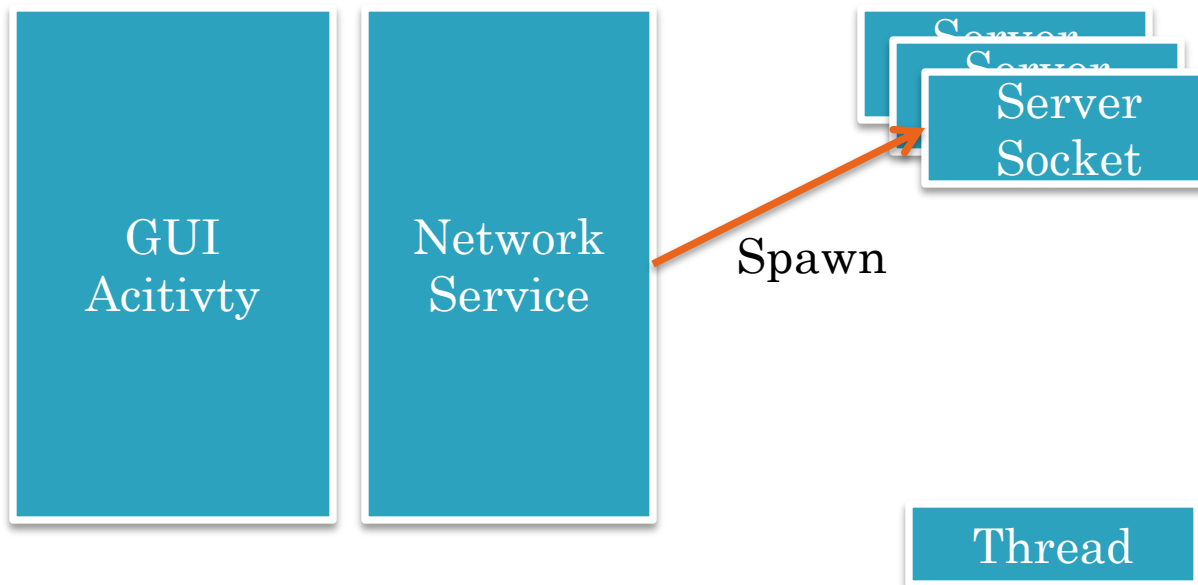
## ○ How?

- New Thread
- Synchronize threads
  - `Handler.post()`



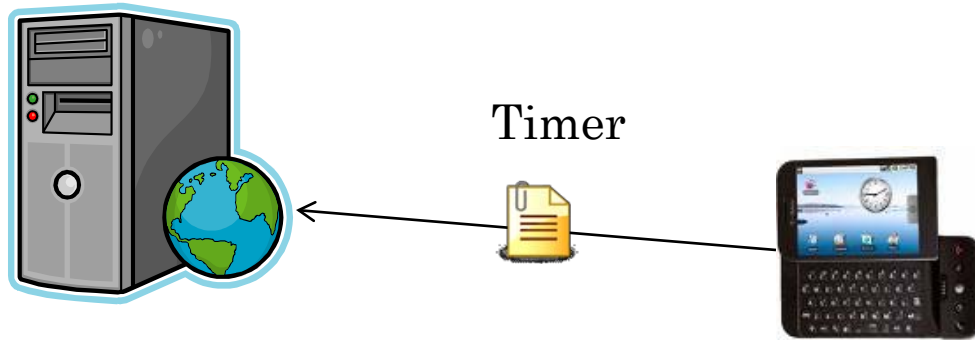
# MP1 THREADING STRUCTURE

- Is it enough?



# PERIODICAL REGISTER WITH SERVER

- Every 15 seconds



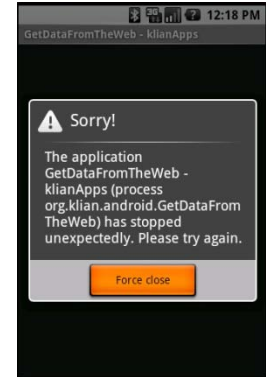
- How to update PeerList on PeerList GUI?
  - Intent



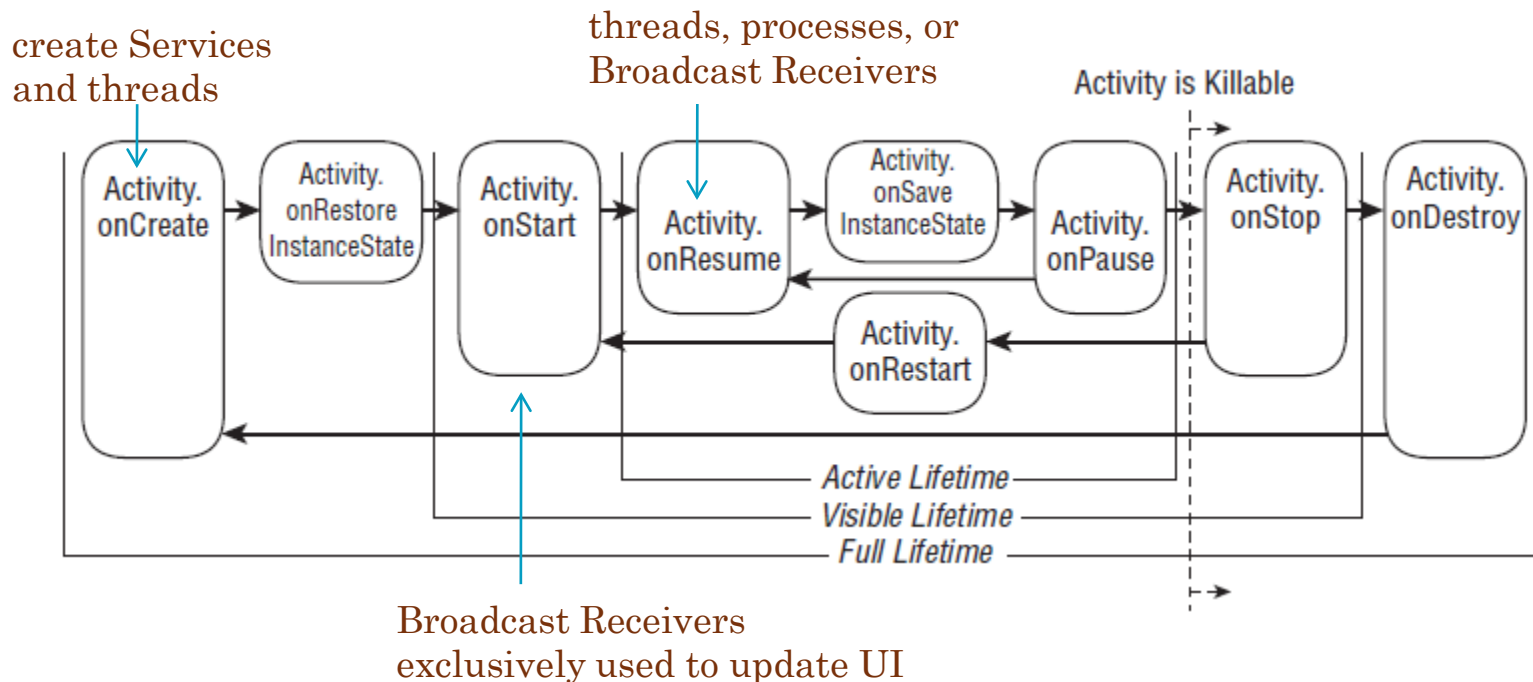


# ACTIVITY LIFETIME

- Android apps do not control their lifetime.



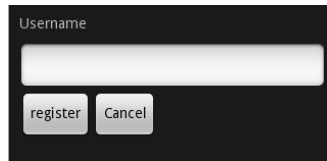
- Active (Foreground) - Paused (FG, lose focus) - Stopped (invisible) – Inactive (kill, exit)



# DECLARATION OF APP – MANIFEST.XML

- Service

- Activity (intent-filter)



- Permission

- Don't forget. Otherwise, your socket programming won't run

```
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    android:versionCode="1" android:versionName="1.0"
    package="com.uiuc.cs425">
    <application android:icon="@drawable/icon" android:label="@string/app_name"
        android:debuggable="true">
        <service android:name=".services.IMService" />
        <activity android:name=".Register">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".PeerList">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.NORMAL" />
            </intent-filter>
        </activity>
        <activity android:name=".Messaging">
            ...
        </activity>
    </application>
    ...
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission
        android:name="android.permission.ACCESS_NETWORK_STATE" />
</manifest>
```

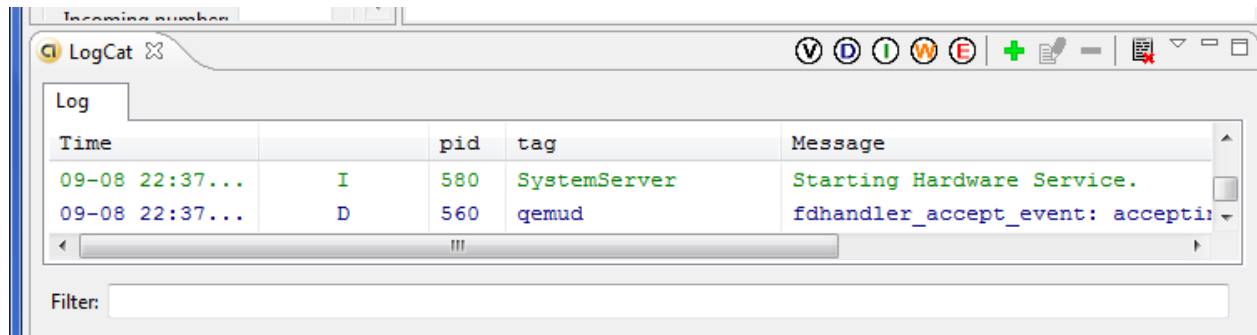
# EXTERNAL RESOURCES

- values/
  - String, color, array, dimension, style theme
- drawables/
  - Image
- layout/
  - screen.xml



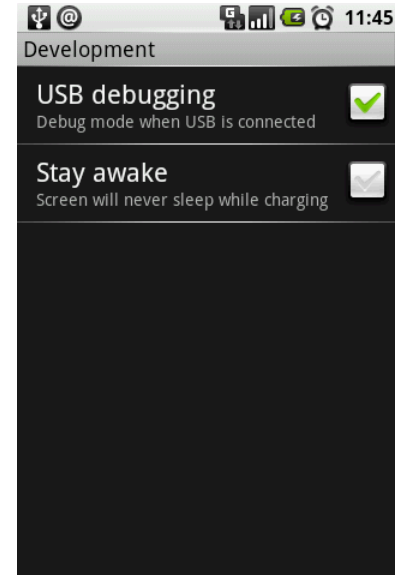
# DEBUG

- ~~System.err.println()~~
- Package - android.util.Log
  
- View results
  - Logcat
  - Eclipse IDE



# DEBUG ON DEVICE

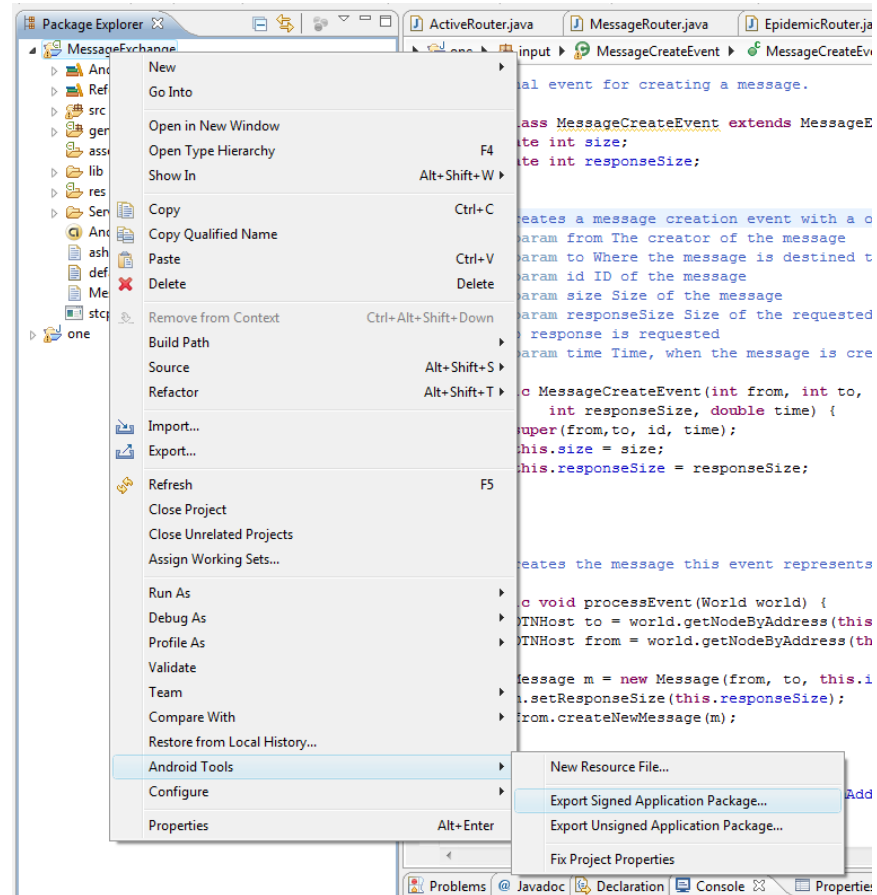
- On device
  - Debug mode
- On desktop
  - Connect your G1 with your PC
  - When it asks for driver location choose
    - For windows, android-sdk-windows-1.5\_r3\usb\_driver\x86\
  - You'll see sth like "HTC Dream Composite ADB Interface" on success
  - (Re)Start Eclipse
  - Your G1 should now be listed in the DDMS-Perspective under Device
- Reference: [[http://www.anddev.org/debugging-installing\\_apps\\_on\\_the\\_g1\\_windows\\_driver-t3236.html](http://www.anddev.org/debugging-installing_apps_on_the_g1_windows_driver-t3236.html)]



# INSTALL PACKAGE TO ANDROID PHONES

- Compile the apk packages in Eclipse
  - Export signed application package

- adb install ...apk
  - Error: uninstall



# OTHER TIPS

- ***Automatic imports of packages***

In Eclipse, cmd-shift-o or ctrl-shift-o

- Start as early as possible

- Possible task partition for a group
  - P1: GUI, activity, intent
  - P2: network service
  - P3: integration



DEMO





# DALVIK DEBUG MONITORING SERVICE

The screenshot shows the Eclipse IDE with the DDMS (Dalvik Debug Monitor Service) interface. The main window title is "DDMS - MessageExchange/src/com/uiuc/cs425/Messaging.java - Eclipse". The menu bar includes File, Edit, Run, Source, Refactor, Navigate, Search, Project, Window, and Help. The toolbar contains various icons for file operations and debugging. The interface is divided into several panels:

- Devices:** Shows a list of virtual devices. The selected device is "emulator-5554", which has a "system\_process" sub-entry.
- Emulator Control:** Contains controls for telephony status, including "Voice" and "Data" dropdown menus set to "home", and "Speed" and "Latency" sliders.
- Threads:** A table showing the file system structure of the selected device.
- File Explorer:** A table showing the file system structure of the selected device.
- LogCat:** A log viewer showing system logs.

Name	Size	Date	Time	Permissions	Info
data		2009-08-13	16:05	drwxrwx--x	
sdcard		1970-01-01	00:00	d---rwxrwx	
system		2009-07-01	00:24	drwxr-xr-x	

Time	pid	tag	Message
09-08 22:37...	I 580	SystemService	Starting Hardware Service.
09-08 22:37...	D 560	gemud	fdhandler_accept_event: accepti

Filter:



# ANDROID DEBUG BRIDGE (ADB)

```
ca. C:\Windows\system32\cmd.exe
C:\Users\Ying>adb help
Android Debug Bridge version 1.0.20

-d          - directs command to the only connected USB device
           - returns an error if more than one USB device is present.
-e          - directs command to the only running emulator.
           - returns an error if more than one emulator is running.
-s <serial number> - directs command to the USB device or emulator with
           - the given serial number
-p <product name or path> - simple product name like 'sooner', or
           - a relative/absolute path to a product
           - out directory like 'out/target/product/sooner'.
           - If -p is not specified, the ANDROID_PRODUCT_OUT
           - environment variable is used, which must
           - be an absolute path.

devices    - list all connected devices

device commands:
adb push <local> <remote> - copy file/dir to device
adb pull <remote> <local> - copy file/dir from device
adb sync [ <directory> ] - copy host->device only if changed
                        - (see 'adb help all')

adb shell - run remote shell interactively
adb shell <command> - run remote shell command
adb emu <command> - run emulator console command
adb logcat [ <filter-spec> ] - View device log
adb forward <local> <remote> - forward socket connections
```

