CS 683 Emerging Technologies
Fall Semester, 2008
Doc 16 Android
Nov 13 2008

References

Google Android Documentation, http://code.google.com/android/documentation.html

Other references given per slide

Mobile Phone Market 2007

3.3 Billion mobile accounts

90% of world population has cell coverage

Source http://www.networkworld.com/community/node/22410

SmartPhone

"mobile phone offering advanced capabilities beyond a typical mobile phone, often with PC-like functionality"

Wikipedia

1992	Simon (IBM)
1996	Nokia 9000 Nokia Communicator series
2001	BlackBerry (RIM) BREW (Qualcomm)
2002	Windows Mobile
2007	iPhone
2008	Android

Smartphone Market

39.9 million sold Q3 2008

13% of mobile phone market

28% increase

Source: http://www.informationweek.com/blog/main/archives/2008/11/apple_beats_rim.html

Mobile Phone OS & Market Share

	Q4 2007	Q2 2008	Q3 2008
Symbian (Nokia)	65%	57.1%	46.6%
iPhone	7%	2.8%	17.3%
BlackBerry	11%	17.4%	15.2%
Windows Mobile	12%	12.0%	13.6%
Linux	5%	7.3%	5.1%
Palm OS		2.3%	
Brew			
Android			

Source http://en.wikipedia.org/wiki/Smartphone

Jobs for mobile phone developers

Work for mobile phone company

- Qualcomm
- Motorolla
- Nokia
- etc

Third party developer

- Hard to make money

And then came iPhone App Store

Trism

- iPhone game
- Costs $5.00
- Two month revenues $250,000

2across

- iPhone crossword puzzle application
- Costs $6.00
- Earned $2,000 per day for a while

Exceptional cases get press

Not clear on revenues over longer time

Source: http://blog.wired.com/gadgets/2008/09/indie-developer.html, http://blogs.computerworld.com/apples_biggest_innovation_for_2008_the_apps_store

Android

Googles mobile phone OS and SDK

Java only

- Special VM
- Nonstandard byte code

Eclipse is development IDE

Linux

Application framework

2D & 3D graphics

Audio, video and still image support

SQLite database

Embeddable web browser

Hardware dependent

GSM

Bluetooth, EDGE, 3G, WIFI

Camera, GPS, compass

accelerometer

Android SDK

http://code.google.com/android/documentation.html

See Getting Started at Android Docs

Supported OS

- Windows XP, Vista
- Mac OS X 10.4.8 or later (intel processor only)
- Linux (Tested on Ubuntu Dapper Drake)

IDE

- Eclipse 3.3 or 3.4
- Java JDK 5 or JDK 6

Design Issues for Mobile Phone Apps

Screen Size

User input

Memory constraints

Limited CPU

Battery life

Security

G1 - First Android Phone

Source: T-Mobile Ad

Why Android

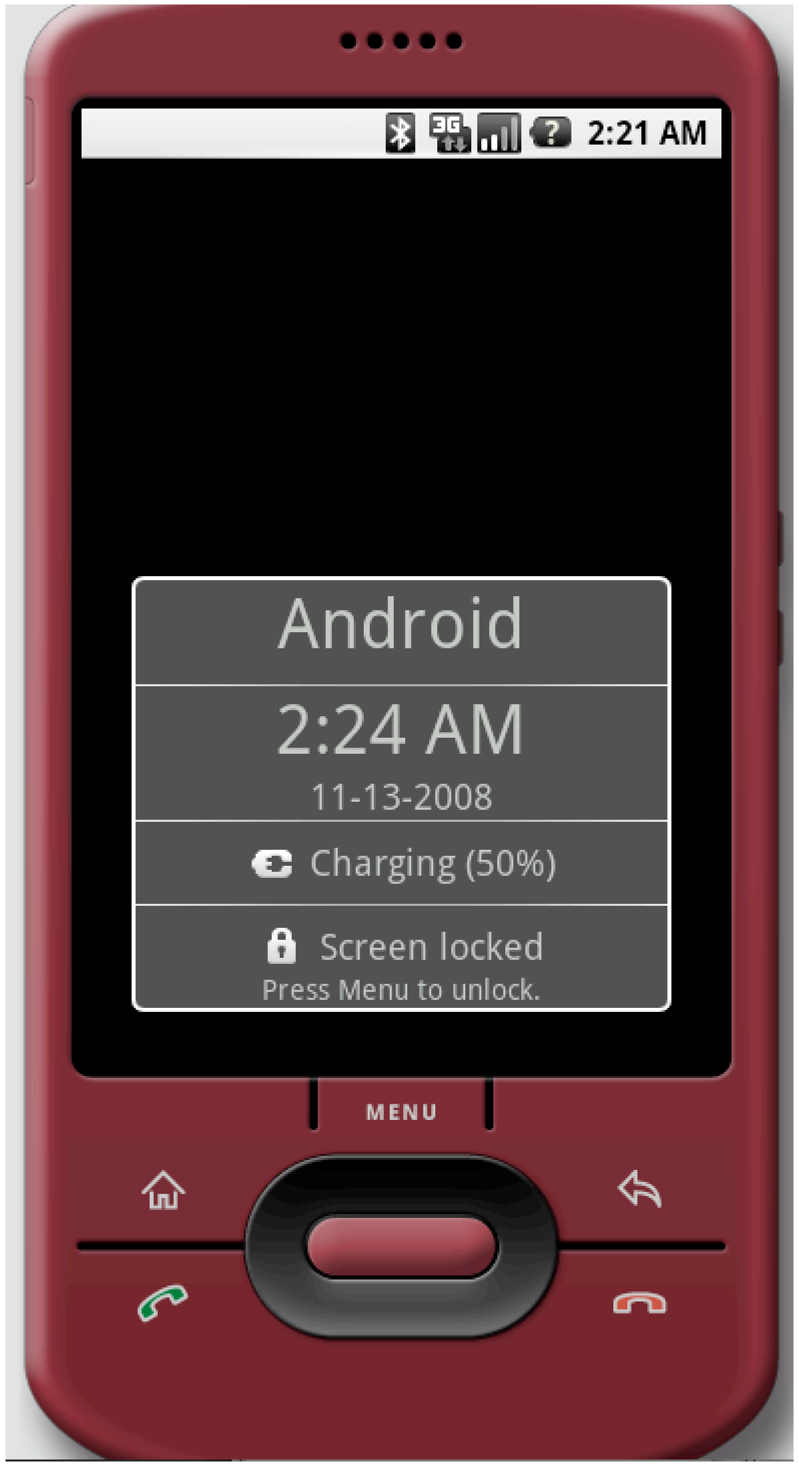
Why did Google create Android

- Google search, maps, talk part of Andoid

Why study Android in this course

- New generation of mobile app development
- Google App store

Emulators



Very useful in developing applications

Not the same as running on real device

- Emulator has bugs
- Device has different bugs
- Device has restriction and limitations
- Device as resources not on your development machine

Eclipse starts emulator when run Android app

- Can recompile and run app without exiting and restarting emulator

Hello World

Following "Hello Android" section of "Getting Started"

Auto generated parts of application

Hello.java

Source code

R.java

Provides access to resources

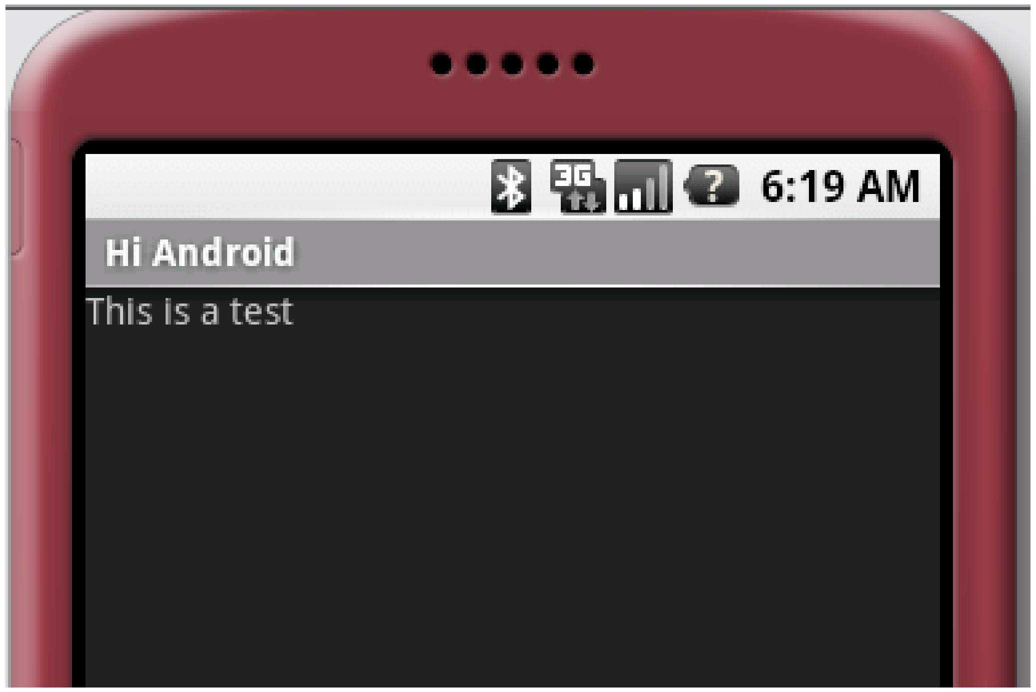
Resources

icon.png	(Application icon)
main.xml	(Optional Layout of application view)
strings.xml	(Allows separation of source code and display text)

AndroidManifest.xml

Describes application contents

Hello.java



6:19 AM

Hi Android

This is a test

```
package edu.sdsu.cs683.hello;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class Hello extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        TextView tv = new TextView(this);
         tv.setText("This is a test");
        setContentView(tv);
    }
}
```

Bold text indicates text added or modified from auto-generated code

Println does not work

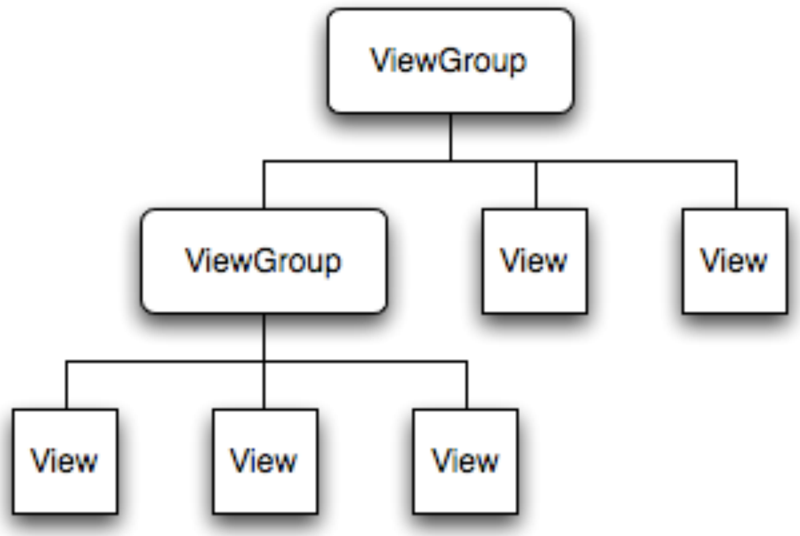
```
package edu.sdsu.cs683.hello;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class Hello extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        System.out.println("I am here"); // no output that I can see
        TextView tv = new TextView(this);
        tv.setText("This is a test");
        setContentView(tv);
    }
}
```

Use the Eclipse debugger

Views

- View
  - Displays content in rectangular area of screen
  - Handles
    - Layout, focus, scrolling
    - Keyboard events
    - Gestures

- ViewGroups
  - Manages set of views and view groups
  - Composite pattern



Some Views

AutoCompleteTextView
Button
CheckBox
CheckedTextView
Chronometer
DatePicker
DigitalClock
EditText
ExpandableListView
Gallery
GridView
ImageButton
ListView
MapView,

MultiAutoCompleteTextView
RadioButton
RatingBar
ScrollView
SeekBar
Spinner
TabHost
TabWidget
TableRow
TimePicker
ToggleButton
TwoLineListItem
VideoView
ViewAnimator
WebView
ZoomButton
ZoomControls

Activity

Single, focused thing that a user can do

Usually each screen has its own activity

An application may have multiple screens, hence multiple activities

An application runs in its own Linux process

Activity Lifecycle

Active

- Running activity in foreground of screen

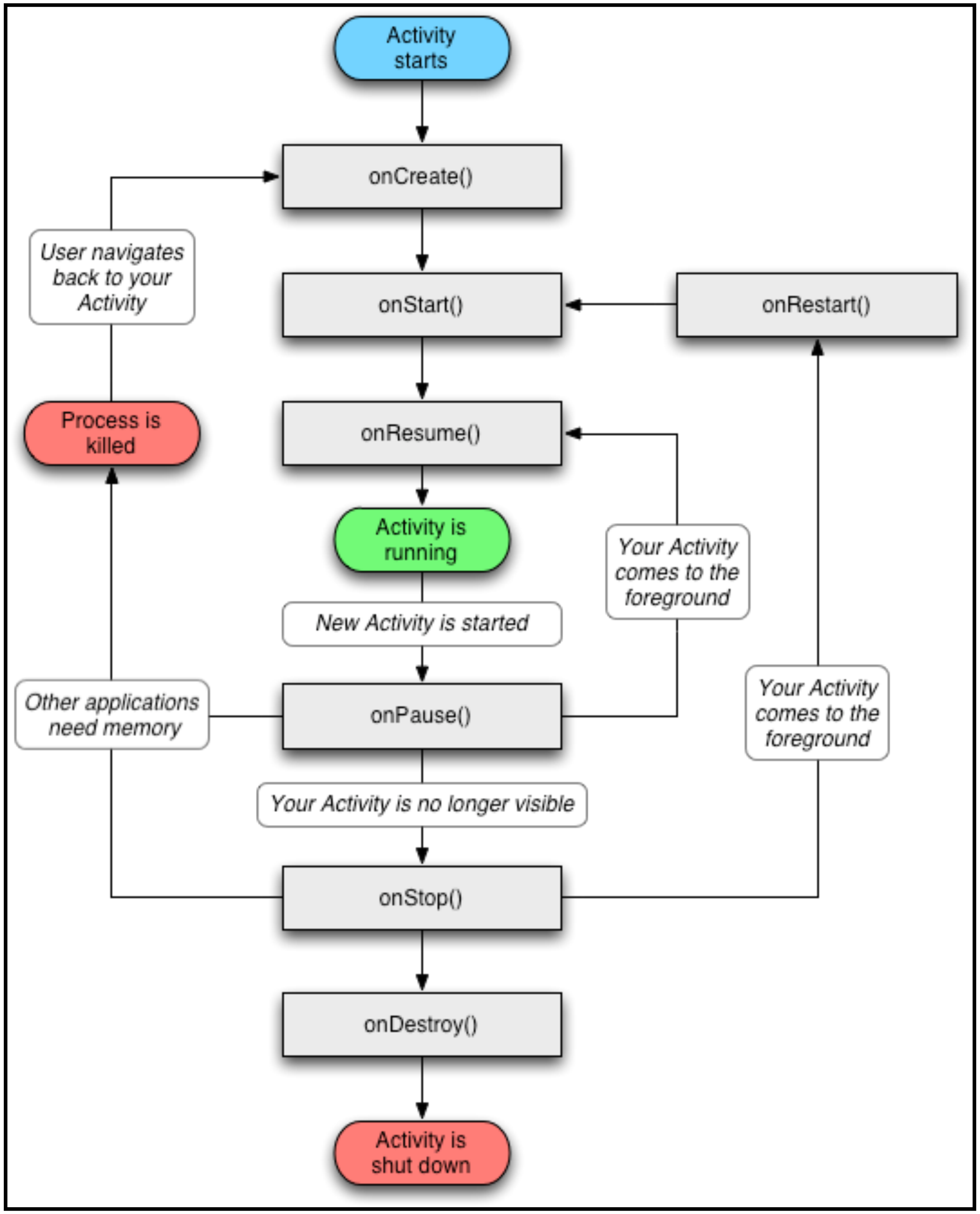
Paused

- Lost focus, but still visible
- Retains all state information
- In extreme memory situations may be killed

Stopped

- Not visible
- Retains all state information
- Often will be killed

Killed


Activity starts
onCreate()
User navigates back to your Activity
onStart()
onRestart()
Process is killed
onResume()
Activity is running
Your Activity comes to the foreground
New Activity is started
Your Activity comes to the foreground
Other applications need memory
onPause()
Your Activity is no longer visible
onStop()
onDestroy()
Activity is shut down

Activity Example

```
package edu.sdsu.cs683;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class CountStates extends Activity {
	int paused = 0;
	int killed = 0;
	int stopped = 0;
	TextView text;
```

Activity Example

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    if (savedInstanceState != null) {
        paused = savedInstanceState.getInt("paused");
        killed = savedInstanceState.getInt("killed");
        stopped = savedInstanceState.getInt("stopped");
    }
    text = new TextView(this);
    text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
            + killed);
    setContentView(text);
}
```

Activity Example

```
protected void onResume() {
        super.onResume();
        text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
                + killed);
    }

    protected void onStart() {
        super.onStart();
        text.setText("Paused: " + paused + " stopped: " + stopped + " killed "
                + killed);
    }

    protected void onStop() {
        stopped++;
        super.onStop();
    }
```

Activity Example

```
	protected void onPause() {
		paused++;
		super.onPause();
	}

	protected void onDestroy() {
		killed++;
		super.onDestroy();
	}

	protected void onSaveInstanceState(Bundle outState) {
		outState.putInt("paused", paused);
		outState.putInt("killed", killed);
		outState.putInt("stopped", stopped);
	}

}
```