

Introduction

AN INTRODUCTION TO ANDROID FOR DEVELOPERS



Introduction

INTRODUCTION GOALS

Introduction Goals

- Get you Started with Android Development
- Get the Environment Set Up and Working
- Create Some Demo Apps (Tutorials)
- Demonstrate the Tools / Environment
- Introduction to the Documentation
 - (Which is changing...)
- Build Enthusiasm (you can do it)

Introduction Goals

- Differences from Other Environments
 - UI - Declarative XML Layout
 - Activities
 - Intents / Intent Receivers
 - Services
 - Content Providers
 - Application Life Cycle
- Project Structure
 - Files, Resources, Building

Tools

- SDK
 - Command line tools (adb, aidl, etc.)
 - Supporting Libraries
- IDE (We will use Eclipse)
 - Eclipse Plugin
 - Included:
 - Debugger
 - Profiler
 - Resource Building
 - Deployment

Not Covered

- Java Development Basics
- Similarities to Other Environments
- Parts that Aren't Ready
 - Syncing etc.
- Anything We Can't Get to in time!
- Get you going, not teach you everything

Learn

GUI CREATION / LAYOUTS

GUI Creation

- Different from
 - Java Swing
 - Java ME
- Layouts
 - res/layout - XML Files Go Here
 - Layouts - Can be Nested
- Strings / i18n
 - res/values/strings.xml
 - Deployment

GUI Creation

- IDs / Lookup
 - Used to Bridge Views / Java Code
 - @+id/myname Syntax
- Resource Building
 - Eclipse Plugin Builds into R.java
 - Efficient Resource Creation / Representation
 - Less Chance of Programatic Errors (Intellisense)
- XML Declarative Faster to Develop

Layout Basics

- Views
 - Basic Building Blocks
 - TextView, EditText, Button, ImageView, Checkbox, Lists, etc
- Layouts
 - FrameLayout : Each Child a Layer
 - LinearLayout : Single Row / Column
 - RelativeLayout : Relative to Parent / Other Views
 - TableLayout : Rows and Columns - HTML like
 - AbsoluteLayout : $\langle x, y \rangle$ Coords - Discouraged
- Layouts can be Nested

Layout Parameters

- Parameters Control Many Aspects
- Some are More Common:
 - `<android:layout_width>` and `<android:layout_height>`
 - “wrap_content”, “fill_parent”, values...
 - `<android:layout_weight>`
 - Relative amount of available space to use
- Most are in the Docs
 - Class Reference documentation most useful

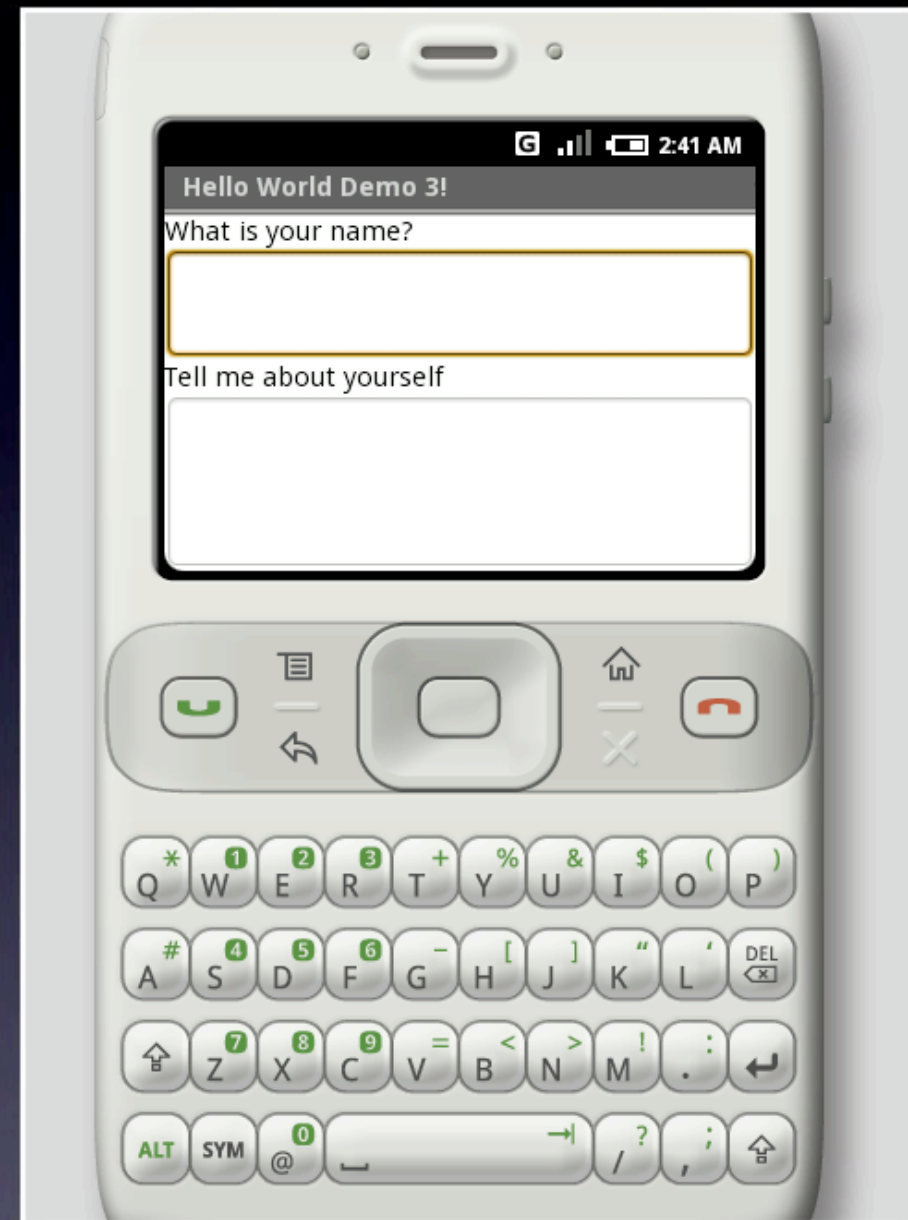
When Things Go Wrong

- Android is still early-release software
- Most problems fall within two areas
 - Build Problems
 - R class not updated or running old code
 - Look at console and problems pane
 - Clean Build
 - Communication breakdown to emulator
 - Code not deploying, errors, debugger failure
 - Use DDMS Reset ADB option
 - Or: quit eclipse and emulator, adb kill-server

Hello World Demo

First Project with Eclipse

Layout Experimentation

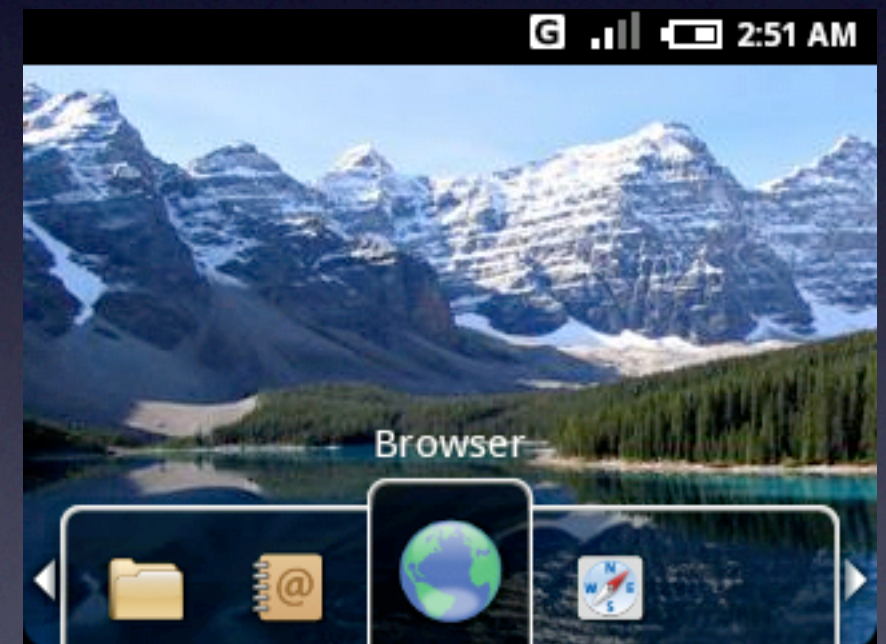


Learn

ANDROID CONCEPTS

Activities

- Typically corresponds to one screen in the UI
 - Can be faceless
 - Can be in a floating window
 - Can return a value
 - Can be embedded



Intents & IntentFilters

- Intents: description of what you want done
- IntentFilter: what an Activity or IntentReceiver can do
- Activities publish their IntentFilters in `AndroidManifest.xml`

Intents & IntentFilters

- Forward navigation is accomplished by resolving Intents
 - Caller calls `startActivity(intent)` (or `startSubActivity...`)
 - System picks Activity whose IntentFilter best matches intent
 - New Activity is informed of the Intent

IntentReceivers

- Respond to alarms and notifications
 - Including those originating externally
- Will wake up your process if necessary
- System can broadcast intents: data connection, phone state changed, etc
- Apps can invent and broadcast their own intents

IntentReceivers

- IntentReceivers can (should) start Services for lengthy tasks (e.g. downloading new data)
- IntentReceivers can put up UI notifications
- Register IntentReceivers in AndroidManifest.xml
- Can also attach IntentReceivers to other objects so they can receive notifications (Activities, Views, etc.)

Services

- Faceless classes that run in the background
 - Music player, network download, etc.
- Services run in your application's process or their own process
- Your code can bind to Services in your process or another process
- Once bound, you communicate with Services using a remotable interface defined in IDL

ContentProviders

- Enable data sharing across applications
- Provide uniform APIs to:
 - query data (returns a Cursor)
 - delete, update, and insert rows
- Hide underlying implementation
- Work across processes

ContentProviders

- All content is represented by URIs
 - Convenience methods mean clients don't need to know syntax
- ContentProviders own URIs based on authority, e.g. `content://contacts/...`
- ContentProviders are responsible for mapping URIs they own to a MIME type

Quick Dial Code Walkthrough

Eclipse Import + Code
Walkthrough



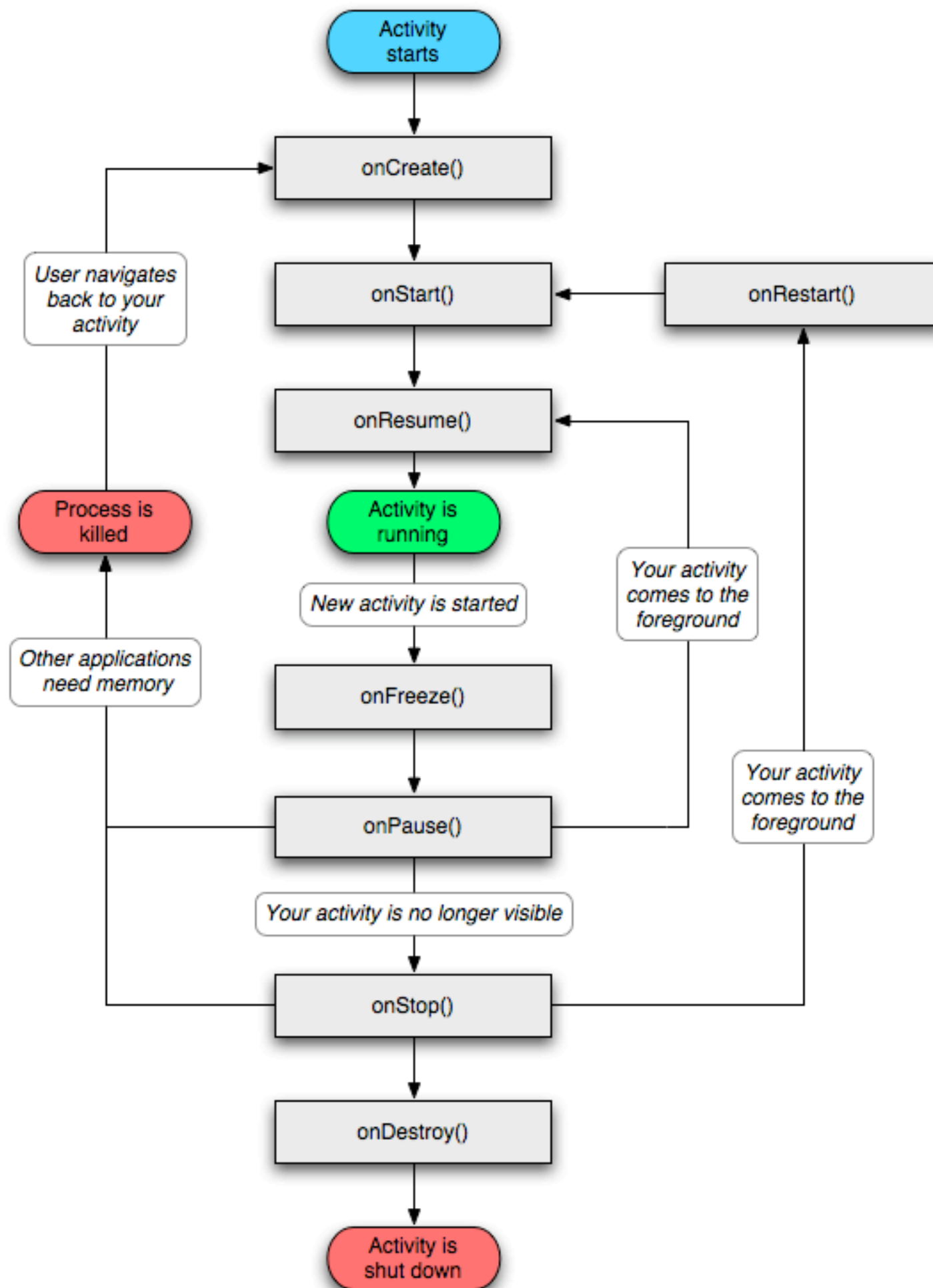
U
n
i
v
e
r
s
i
t
y

LIFE CYCLE & BUNDLES

Application Lifecycle

- Applications run in their own processes
 - Many Activities, Services, etc. can run in the same process
- Processes are started and stopped as needed to run an application's components
- Processes killed to reclaim resources

Life Cycle of an Application



Activities

- Designed to be reused and replaced
 - Your application can invoke Activities from another application
 - Another application can invoke your Activities
- Existing Activities can be replaced with a new Activity that fulfills the same contract

Activity Lifecycle

- An Application can have Several Activities
- Activities can be started with
 - `startActivity()` - Synchronous
 - `startSubActivity()` - Asynchronous, with handler callback
- Each Activity has its Own Life Cycle
- Messages can be Passed with Bundles
- Like Services on a Bus

Bundles

- Similar in concept to a simple Map
- Can put data into Bundle under a name
 - `bundle.putString("name", <data object>)`
- Can also be retrieved by name
 - `bundle.getString("name")`
- Bundles used to pass messages/store state
- `onFreeze()` has outState bundle passed in
- Data stored in outstate handed back to `onCreate` as the icle bundle

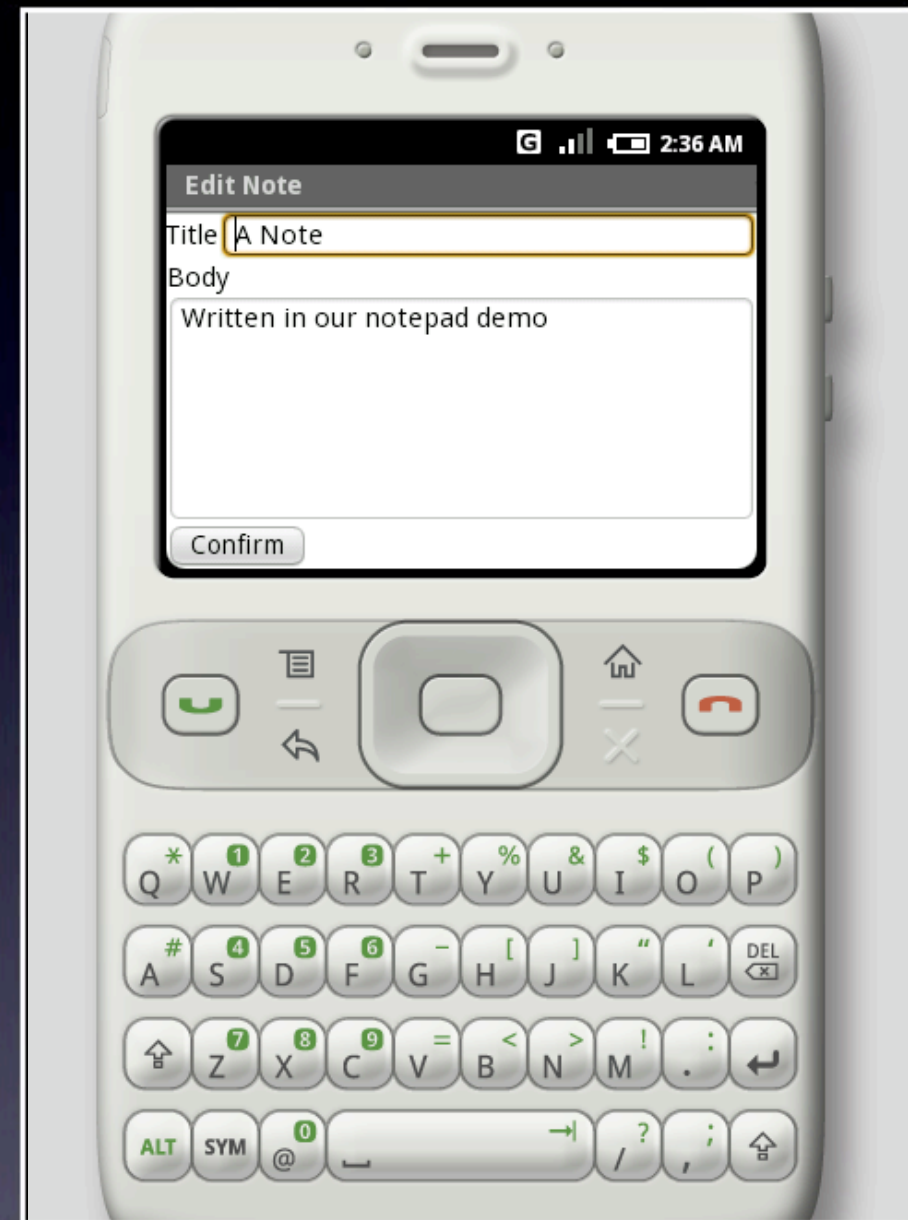
Storage / Persistence

- Normally handled by `ContentProvider`
 - Front end to File, DB, Online Service, etc.
- SQL Lite is Available to Developers
 - Simple SQL DB
- Can also access flat files
- Online communication yet to be formalized

Note Pad Example

See the Tutorial in the SDK

Multi-stage exercise to illustrate the basics.



RESOURCES, PACKAGING AND SECURITY

Resources

- Added under the res/ folder
 - layout, drawable, anim, values, xml, raw
 - layout for declarative layout files
 - drawable/anim for bitmaps, transitions, etc.
 - values includes strings, arrays, colors, dimensions, styles and more
 - xml for general XML files available at runtime
 - raw for binary files (e.g. sound)
- Compiled into the apk
 - Read through API convenience methods
 - e.g. XML access looks like a pull parser

Assets

- Similar to Resources, but...
 - InputStream access to Assets
 - Placed under assets folder
 - Looks like a “root” folder to app
 - Read only access
- Any kind of file
 - Stored on device - watch the size

APK files

- Android Packages
 - All class files and resources needed to run
 - Class files recoded to dex
 - Manifest defines activities and other facets
- DEX
 - Dalvik Executable
 - More compressed form than bytecode
 - Third party libs can be converted to dex
- APK is the application
 - Install - put APK in data/app
 - Uninstall - remove APK

Security

- Unique Users for Apps
 - App completely sandboxed
 - Sharing only via ContentProviders, Activities, IntentReceivers, Services, etc.
 - Strong, linux-backed security
- Access to Restricted Features
 - Must be declared in the Manifest
 - Still working on rest of security model
 - Some kind of trusted authority
 - Advanced users should have fine grained control

Learn

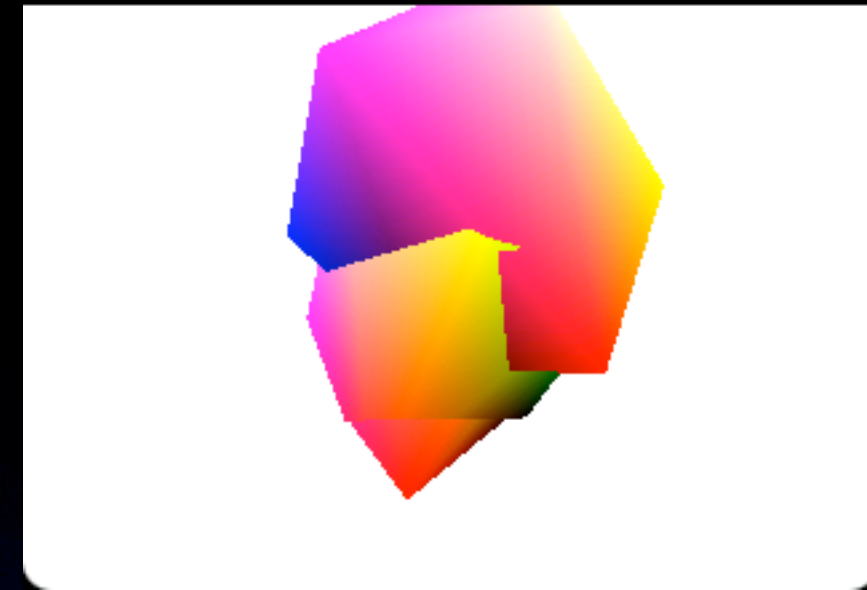
OTHER APIs

2D Graphics



- Similar to the Java 2d API
 - Canvas passed in to graphics methods
 - Drawable Resources
 - Alpha channel available
 - Supports transformations (e.g. Rotate, Scale)
- Custom Graphical Components
 - Extend View
 - Override onDraw()
 - Override onMeasure() - setMeasuredDimension()
 - Other Optional Overrides: onKeyDown(), onKeyUp(), onMotionEvent(), etc.

3D Graphics



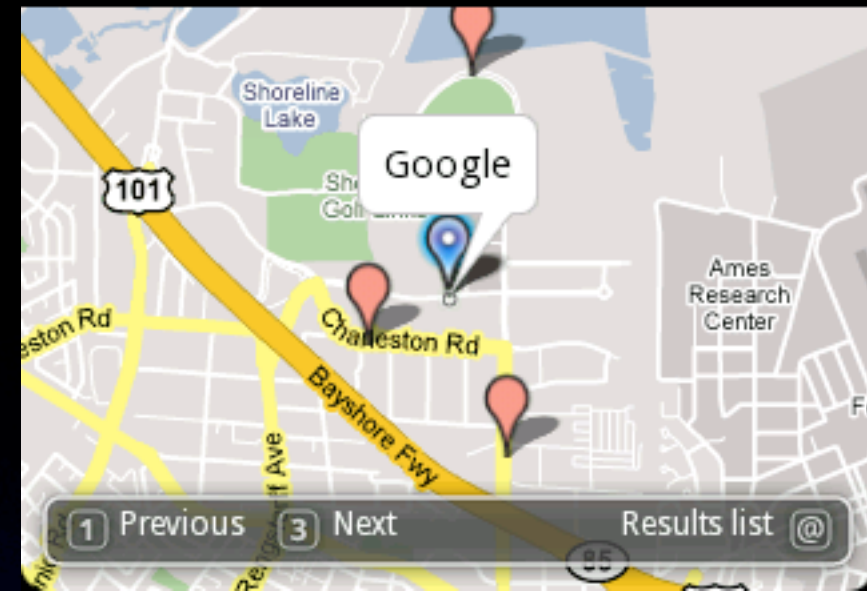
- Optional Hardware Support
- OpenGL ES 1.0 (+ some 1.1 features)
 - Corresponds to OpenGL 1.3 for desktop
- Using the API
 - Extend View
 - Obtain handle to OpenGLContext
 - Obtain handle to GL object in onDraw()

Location Based Services

- Optional Hardware Support for GPS
 - But Cell ID supported on all devices
- LocationProvider
 - Part of the android.location package
 - Use LocationManager to get location and bearing
 - Context.getSystemService
(Context.LOCATION_SERVICE)
- Mock LocationProviders
 - Test in the Emulator
 - /data/misc/location/<provider_name>
 - class, kml, nmea, track file support

Maps

- Natural complement to LBS
- MapView and MapActivity
 - MapActivity easier to use
 - MapView can be embedded in your app
- MapView more flexible
 - Can surround with your own controls
 - And, can trigger events in your own activity
 - But, more involved API
 - Must be created in a MapActivity subclass



Media



- Play back and Record media files
- Audio, Picture and Video
 - CODECs still being decided
- Access through Intents
 - Stream from a URL
 - Set MIME type to help decide action
- Also, class level API
 - Embed playback (e.g. sound effects)
 - Recording

XMPP

- Alternative to SMS for message passing
- Can pass richer content, including bundles
 - Strings only at present
- Can fire Intents
- Server initiated push

Low-Level H/W Access

- APIs not yet available
- Will cover low-level hardware, e.g.
 - Bluetooth
 - WiFi

Learn More

LEARNING MORE



Learning More

- Sample Apps
 - Lunar Lander, Snake - 2d graphics, game
 - Notepad - ContentProvider, Painting override
 - SampleCode - Layouts, Services, Intents & Receivers, much more
- Online
 - groups.google.com/group/android-developers
 - /android-beginners, /android-discuss, /android-internals, /android-challenge
 - <http://android.com>
 - <http://code.google.com/android>

Documentation

- Currently Being Improved
 - Class Documentation Coverage
 - Accessibility
 - Additions - Like Samples, Tutorials, etc.
- Demo
 - Tutorials
 - Class References
 - Including Layout Parameters
 - What's Included in the Class Libs?

Command Line Tools

- ADB

- adb shell - command line into emulator/device
 - Simple unix-like shell
 - Can delete apk files from data/app
- adb install - add packages

- Emulator

- Many command line flags
 - Different resolution, options, etc.

- Other Commands

- See documentation for aidl, aapt, etc.

- Don't have to use Eclipse

Questions?