ANDROID

Graduate presentation for CSCI 5448

By

Janakiram Vantipalli

(Janakiram.vantipalli@colorado.edu)

Content

- What is Android ??
- Versions and statistics
- Android Architecture
- Application Components
- Inter Application Communication
- Manifest File
- Activity life cycle
- Media playback
- Android sensors
- Android package (APK)
- Best Practices

Smart Phone

- Smart phones are high specification phones that operate like basic level computers. They can usually do tasks like connect to the internet and receive emails .They have various components like camera, light sensors etc.
- Operating system(OS) is a software stack operating on a hardware
- An application is a software executable designed to run on the OS platform by help of a middleware

What is Android

- Android is a platform and Operating system developed by Google based on Linux Kernel.

- Android eco system consists of

  a) Hardware manufacturers - partners in Open handset alliance(Samsung ,Asus ,Sony, HTC etc)

  b)Android Operating system-Developed and supported by Google

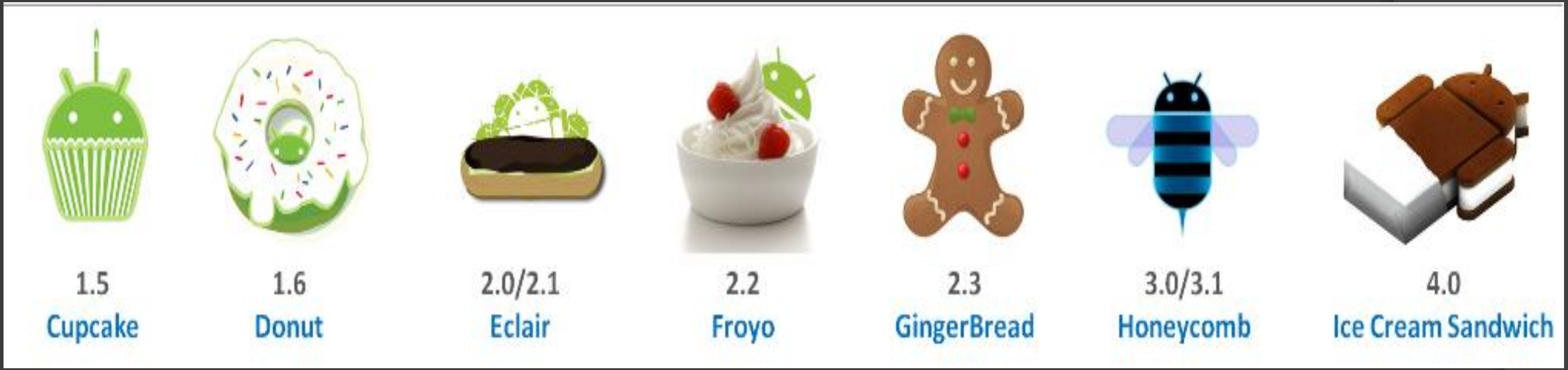
  c)Applications-Developed and supported by Google and respective application owners

Evolution of Android

- Android is continuously evolving from it's initial stage.
- Google does this by releasing the updates to the operating system , alphabetically after tasty treats ☺
- With Android updates, Original equipment manufacturers (OEM) implement new hardware like front camera, Near field communication(NFC) etc
- Also OEM's can choose to customize Android components developed by Google .

  Eg: HTC sense UI and Samsung TouchWiz UI

Android Versions

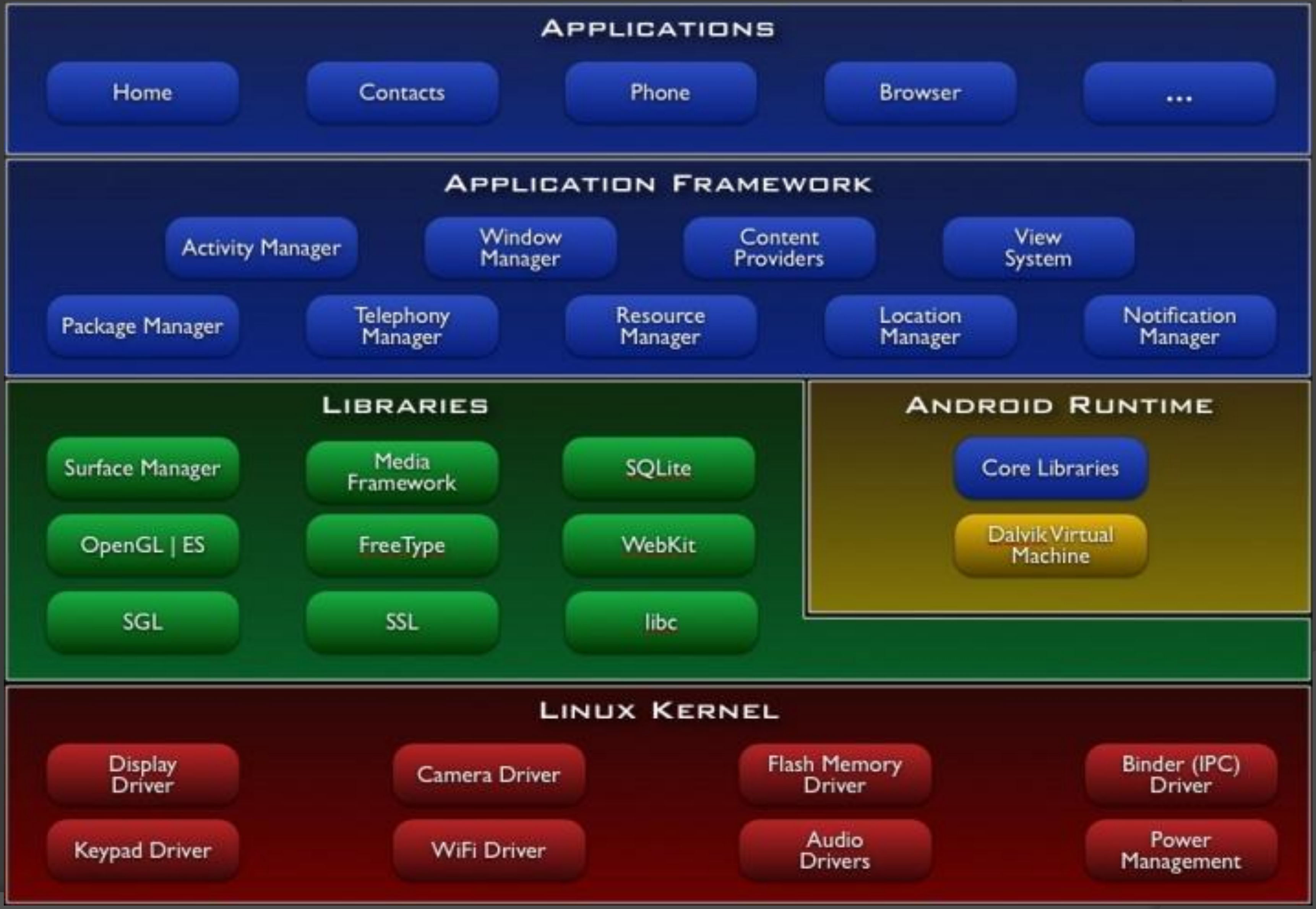


Latest versions are Android 4.1.x and 4.2 Jelly Bean.

Android Statistics

- Every day more than 1 million new Android devices are activated worldwide
- Android Apps in Google play 850,000(as of oct 2012)
- Android users download more than 1.5 billion apps and games from Google Play each month.

Android Architecture (reference wikipedia)



APPLICATIONS

Home
Contacts
Phone
Browser
...

APPLICATION FRAMEWORK

Activity Manager
Window Manager
Content Providers
View System
Package Manager
Telephony Manager
Resource Manager
Location Manager
Notification Manager

LIBRARIES

Surface Manager
Media Framework
SQLite
OpenGL | ES
FreeType
WebKit
SGL
SSL
libc

ANDROID RUNTIME

Core Libraries
Dalvik Virtual Machine

LINUX KERNEL

Display Driver
Camera Driver
Flash Memory Driver
Binder (IPC) Driver
Keypad Driver
WiFi Driver
Audio Drivers
Power Management

Android Architecture

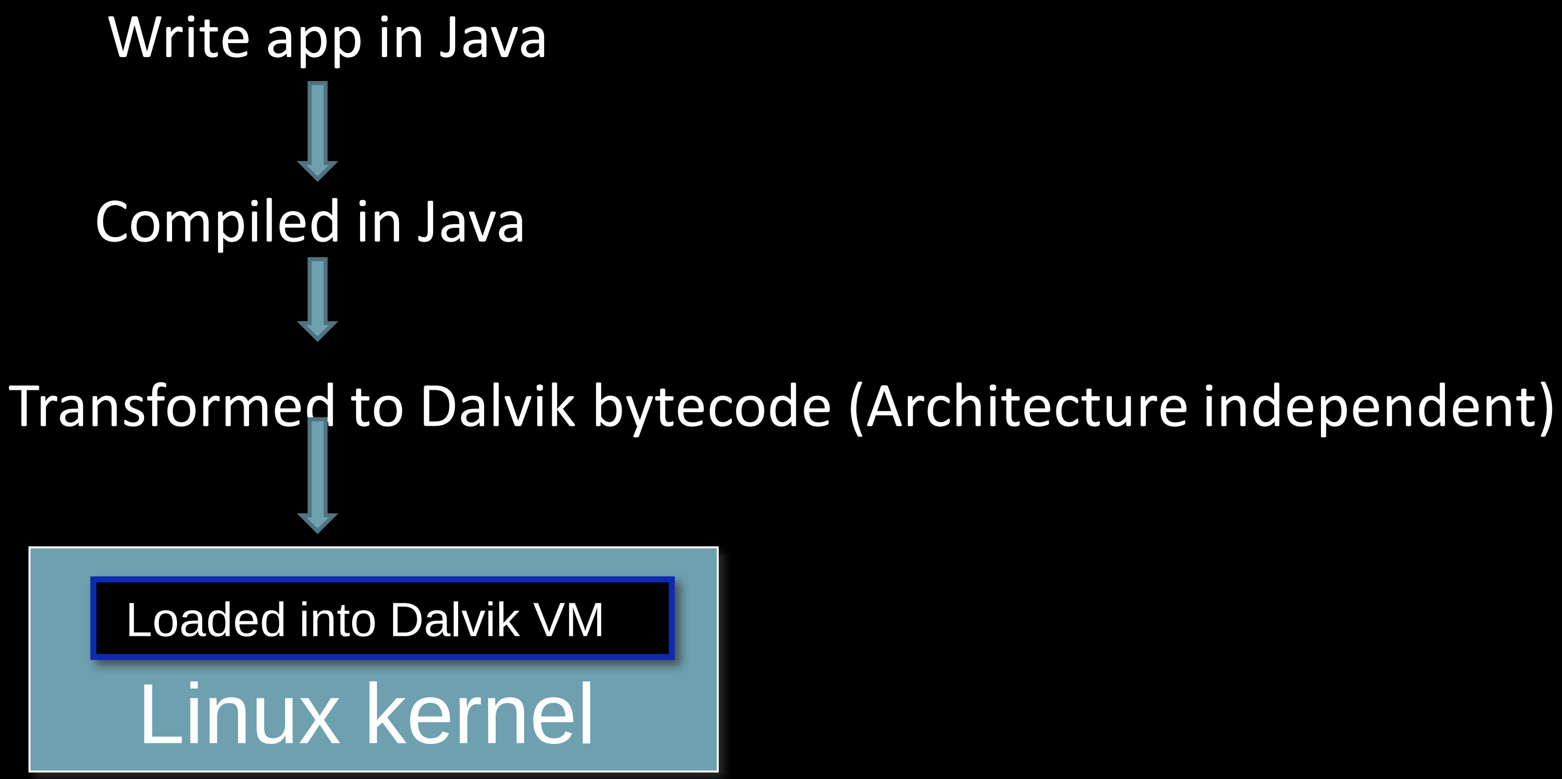
Generally Android OS has Four layers

- Applications-Developed in Java, executed in dex code
- Framework services and Libraries-Implemented in java.

  Eg: Media framework has codecs like mp3 to play music.
- Native libraries, daemons and services-Implemented in C/C++ . Eg: Open GL for 3D graphics ,Webkit is engine that powers the browsers
- Linux Kernel-Implemented in C. Contains Device drivers.

Application is everything here…

How App runs on Android



Write app in Java

Compiled in Java

Transformed to Dalvik bytecode (Architecture independent)

Loaded into Dalvik VM

Linux kernel

Each App runs on single Virtual machine as a process.

Application components

- **Activities** - provide user interfaces. All visible portions of applications
- **Services** - run in the background and do not interact with the user Other components can bind to Services.
- **Broadcast Receivers** - receive Intents sent to multiple applications. Receivers are triggered by the receipt of an appropriate Intent and then run in the background to handle the event.
- **Content providers** - databases addressable by their application defined URIs.

Examples of components

- Activity: Compose mail screen of email application
- Service: A music service started by application, still plays music after user navigated to different application.
- Broadcast Receiver: when screen is turned off, a notification is broadcasted to all active application’s broadcast receivers at that time
- Content provider: Contact list provided by Android stores information about contacts can be modified by application that has permission.

Application components

- Each component performs a different role in the overall application behavior, and each one can be activated individually (even by other applications).

  Eg: A picture gallery application can activate compose mail screen of email application to share the picture via email

- Each component is a different point through which the system can enter your application.

- The manifest file must declare all components in the application .

Some UI components

- **Fragments -** components which run in the context of an Activity. Fragment components encapsulate application code so that it is easier to reuse it and to support different sized devices.

- Fragments are optional, we can use Views and ViewGroups directly in an Activity

- **Views and ViewGroups -** Views are user interface widgets, e.g. buttons or text fields and ViewGroups are responsible for arranging other Views

Communication between Apps

- Android provides a sophisticated message passing system .
- An Intent is a asynchronous message that declares a recipient and optionally includes data
- Applications use Intents for both inter-application communication and intra-application communication.
- Except content provider, all the three components can be activated by Intent.

Manifest file

- The manifest file(AndroidManifest.xml) contains essential information about the application so that Android system can run application's code.
- Remember ,all components must be declared in this file.
- Application permissions are to be presented here.
- Permissions that other applications are required to have in order to interact with the application's components are declared here
- Names Java package required by the application

Manifest File contents

<action> - Adds an action to an intent filter .

<activity> - Declares an activity that implements part of the application's visual user interface .

<application> - declaration of the application.

<category> - Adds a category name to an intent filter

<intent-filter>- Specifies the types of intents that an activity, service, or broadcast receiver can respond to

<manifest> - root element of AndroidManifest.xml

<meta-data> - A name-value pair for an item of additional, arbitrary data that can be supplied to the parent component

Manifest File contents

<permission> - Declares a security permission that can be used to limit access to specific components

<provider> - Declares a content provider component

<receiver> - Declares a broadcast receiver as App component

<service> - Declares a service as one of the application's components

<uses-library> - Specifies a shared library that the application must be linked against

<uses-sdk> - minimal SDK version for which application is valid

Activity life cycle

- The Android system supervises the lifecycle of application. At any time the Android system may stop or destroy your application

  e.g. because of an incoming call.

- Pre Defined methods

  - onSaveInstanceState() - called after the Activity is stopped to save data

  - onPause() - always called if the Activity ends

  - onResume() - called if the Activity is re-started, can be used to initialize fields

OO concepts everywhere…

- Unlike a standard Java class, an Android Activity class is instantiated by the system when code creates and starts an Intent object.

- Inheritance: The application activity classes inherit the parent activity class of Android framework.

- Overriding methods: An inheriting class can provide a dedicated implementation of any method in the superclass it is extending.

Before moving further…

- Till now we saw the basic Application and various components of Application.
- Also we saw how application components and metadata can be represented in manifest file
- Now we will look the structure and design a simple media playback framework on android and also android sensor frame work

Application Development Tools

- To develop an Android application one needs
  - -Android SDK
  - -Eclipse with ADT(Android Development Tools)
- The Android SDK contains three important tools :
  - -Android Emulator that simulates and runs Android virtual device
  - -ADT plug-in for Eclipse IDE
  - -Dalvik Debug Monitor service (DDMS) tool manages processes on an emulator and assisting in debugging.

Media playback frame work

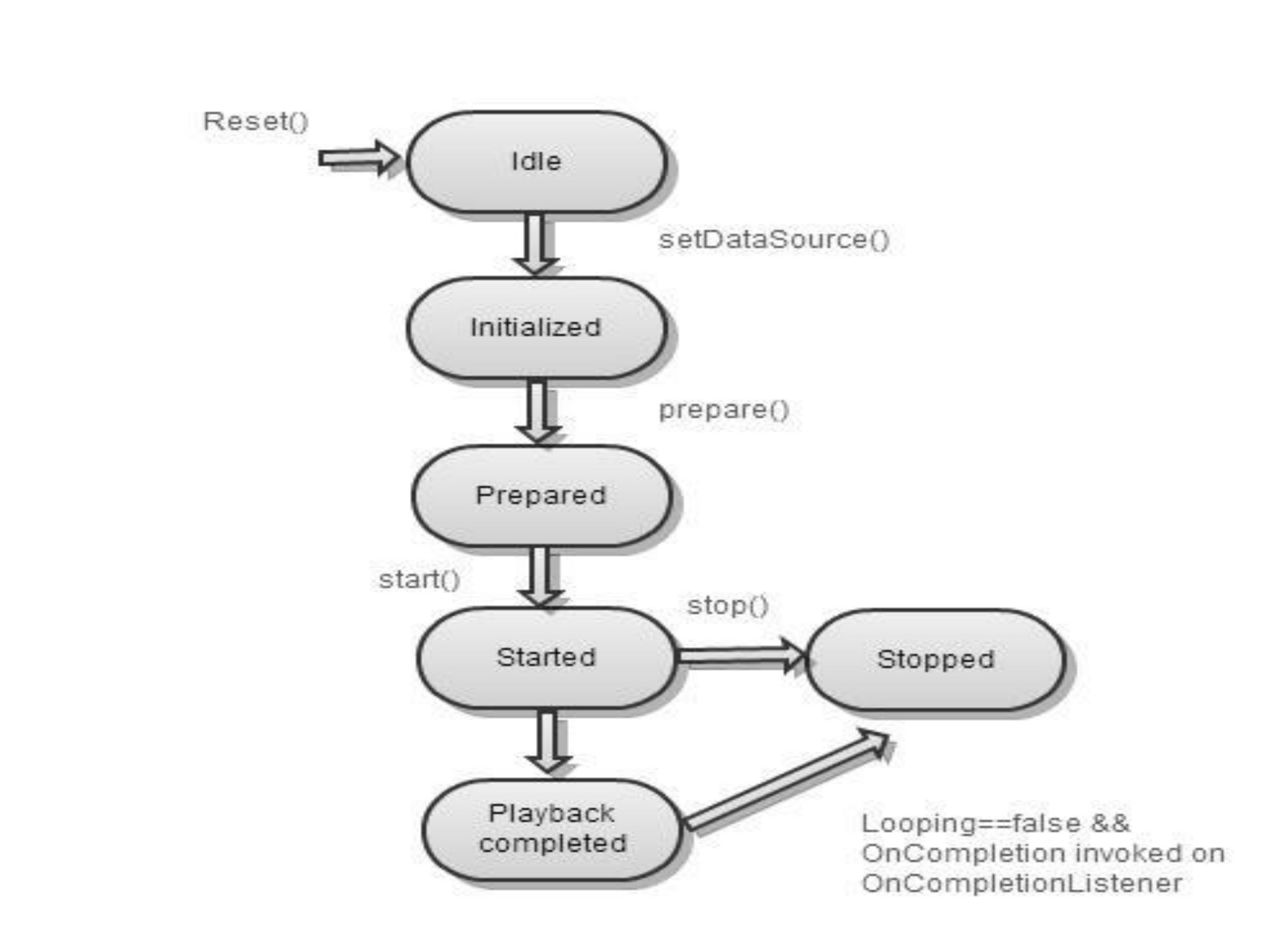
- The following classes are used to play sound and video in the Android framework:

  - MediaPlayer class - primary API for playing sound and video.

  - AudioManager class-manages audio sources and audio output on a device.

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

Media Playback simple state diagram



Media playback

- Permissions-One may need to declare a corresponding WAKE_LOCK permission <uses-permission> element.

- Callbacks -Applications may want to register for informational and error events in order to be informed.

  -setOnPreparedListener(OnPreparedListener)

  -setOnCompletionListener(OnCompletionListener)

  -setOnErrorListener(OnErrorListener)

- In order to receive the respective callback associated with these listeners, applications are required to create MediaPlayer objects on a thread

Android Sensors

- Android-powered devices have built-in sensors that measure motion, orientation, and various environmental conditions.
- Three broad categories of sensors:

  -Motion sensors - accelerometers, gravity sensors

  -Environmental sensors - photometers and thermometers.

  -Position sensors - orientation sensors and magnetometers.

Android sensor framework

- The sensor framework provides several classes and interfaces that help you perform a wide variety of sensor-related tasks.
- Part of Hardware package
- Classes in this package are SensorManager,Sensor and Sensor Event.These classes have API s by which one can obtain raw data from sensors.

```
public class SensorActivity extends Activity implements
SensorEventListener {
        private SensorManager mSensorManager;
        private Sensor mLight; }
```

Android package(.apk)

- The Android SDK contains a tool called *dx* which converts Java class files into a .dex (Dalvik Executable) file. All class files of one application are placed in one compressed .dex file

- The .dex file and the resources of an Android project

- The resulting .apk file contains all necessary data to run the Android application and can be deployed to an Android device via the *adb* tool.

Application storage options…

- Shared Preferences-Store private primitive data in key-value pairs.
- Internal Storage-Store private data on the device memory.
- External Storage-Store public data on the shared external storage.
- SQLite Databases-Store structured data in a private database.
- Network Connection-Store data on the web with own network server.

Apps VS widgets

- As we saw Applications in the earlier, we will see what widgets are now.
- Widgets are an essential aspect of home screen customization.
- Access App or sometime it’s functionality via widget from home screen
- Types of widgets:

  -Information widgets

  -Collective widgets

  -Control widgets

App framework (Tablet vs Smart Phone)

- Initially when Android was introduced only for mobile devices.

- HoneyComb (ver 3.0) support for tablets has been introduced.

- With Android 3.0 (API level 11), Android introduced a new set of framework APIs that allow us to more effectively design activities that take advantage of large screens. eg: the Fragment APIs.

- Android 3.0 also introduced ActionBar.

Best Practices

- Android experience is completely dependent on how user feels comfortable using an application.
- Application developer must design an application

  - better navigation from one activity to other.

  - various screen sizes that Android support today.

  - compatible on multiple Android versions.

Android Future

- Android is powering almost two thirds of smart phones on the planet.
- Many Android tablets are available in the market.
- Recently Samsung has released a Galaxy camera with jelly bean and 4G connectivity.
- Scope for Android powering hybrid devices is very high.

Summary

- Discussed what Android is
- Powerful architecture
- Android and App relationship
- Application fundamentals
- Application development fundamentals
- Rich frame work
- Best practices in App development

Resources & References

- developer.android.com is the popular official source for getting started with Android.
- Google channel on YouTube host many videos about android and Application development
- For more resources, just Google “Android” ☺

Thank You

Any Questions:

janakiram.vantipalli@colorado.edu

Disclaimer: All pictures and content are referred from internet and are used only to better explain Android to the audience of this presentation. Original owners deserve all credit for creating them ☺.