

# **Android 250 - Lecture 1**

## **Review & Intro**

Margaret Maynard-Reid  
March 30, 2015

# Agenda

- Overview
- Review
- User Interface Design
  - Guidelines & Patterns
  - Fundamentals
  - Tools
  - Terminologies

# Introductions

- Instructor:  
Margaret Maynard-Reid
- Teaching Assistant:  
Chuwuezugo Nwosu

# Course Overview

- Course portal on Catalyst
- Sample code on both GitHub & Catalyst
- Video recording available after class

# Attendance

UW PCE Policy states you can't miss more than 80% of classes

- Online versus In-Class
- Show Up! It should be fun.

# Grading

- Submit on Catalyst Dropbox
- Total of 100 points:
  - 3 homework assignments - 50 points
  - 1 (group) project - 45 points
  - Participation - 5 points
- You need at least 80 points to pass

# Final Project

- Choose an app of your interest
- Same app or a different one
- Form team & idea by 4/20
- Submit project by 6/8
- Present on 6/8 via recording or in person

# Course Materials

Official reference:

- The Android Developer Online Documentation

Recommended

- The Busy Coder's Guide to Android Development, by Mark Murphy

Optional Reading:

- Android Programming: The Big Nerd Ranch Guide, by Bill Phillips and Brian Hardy
- Learn Java for Android Development 3rd Edition, by Jeff Friesen



# Resources

Aside from classroom learning & Catalyst,

- Online documentation
- Textbooks
- Android SDK sample code
- Stackoverflow
- Course discussion forum
- Preferably no email unless personal

# Android Stories

- [New Lollipop “on-body” mode keeps your phone unlocked as long as you’re carrying it](#)
- [Amazon wants to give paid apps away for free](#)
- <http://www.talkandroid.com/243111-google-now-cards-on-their-way-to-chrome-launcher-2-0/>

# Break

# Review - Android Basics

- What is the Manifest?
- What are the 4 primary elements of an Android application?
- Give an example of a permission?
- What is an Intent?
- What makes up an Android app?

# Review - UI Layout

- When do we use a `LinearLayout`?
- When do we use a `RelativeLayout`?
- When do we use a `FrameLayout`?
- The difference between a margin vs. padding
- Can you describe Gravity vs. Weight?

# Review - View

- What is an EditText?
- Can you interact with a Toast?
- What is a Spinner?
- What is an Adapter?

# Review - UI

- What is a ListView?
- What is a Fragment?
- How do you add a fragment?
- What is an ActionBar?

# Review - Storage

- What is stored on internal storage?
- What is stored on external storage?
- What are SharedPreferences?
- Can you access the files on the internal storage via DDMS/File Explorer?
- Are files secured on the external storage?



# Review - Storage & DB

- What are 3 ways you can persist data locally?
- SharedPreferences vs. Database
- Where is the sqlite db stored?
- What is the best way to create a SQLite DB?

# Review - Intent

- What is an Explicit intent?
- What is an Implicit intent?
- An example of how IntentFilter is used?
- What is a BroadcastReceiver?
- An example of Android system broadcast?
- How do you register a BroadcastReceiver?

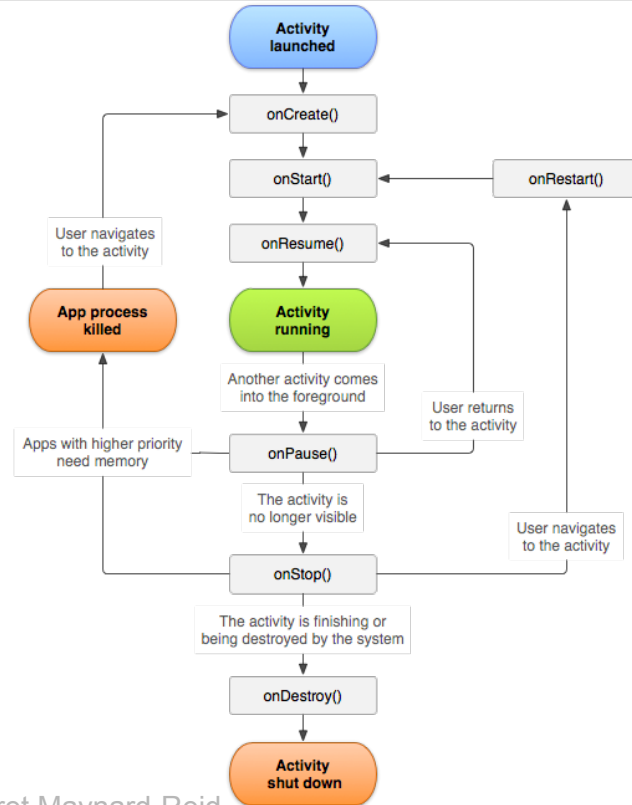
# Review - Threading

- On which thread are UI updates performed?
- Can you update UI from a non-UI thread?
- Give two examples of how to create a new thread, then perform UI updates?
- What is an AsyncTask?

# Activity Lifecycle

## Main Cases

- App Starts
- App to Background
- App to Foreground
- App Killed



# Break

# User Interface Design

Create beautiful UI and avoid design mistakes...

- Do research or lean on the research of others...
- Get inspiration from others' design
- Follow UI Guidelines and Patterns!

# User Interface

## Guidelines & Patterns

# Android UI Design

- The official Android UI design guideline  
<http://developer.android.com/design/index.html>
- Official Android UI design patterns <http://developer.android.com/design/patterns/navigation.html>



# Mobile UI Patterns

- Android UX - <http://androidux.com/>
- Android Patterns - <http://www.androidpatterns.com/>
- Mobile Patterns (Android & iOS) <http://www.mobile-patterns.com/>
- AndroidPptrns - <http://androidpptrns.com/>
- Pptrns (Android & iOS) - <http://pptrns.com/>

# Get inspiration

Android Niceties - a collection of screenshots of beautiful Android apps...

<http://androidniceties.tumblr.com/>

# Material Design

Unified experience across platforms and device sizes...

- Introduced in Android 5.0 Lollipop
- A new theme
- New widgets for complex views
- New APIs for custom shadows and animations

# User Interface

## Tools

# Design Tools

- WireframeSketcher  
<http://wireframesketcher.com/>
- Fluid UI  
<https://www.fluidui.com/>
- Pencil  
<http://pencil.evolus.vn/>
- MockFlow  
<http://www.mockflow.com/>

# Content Tool

## Raster

- Microsoft Paint
- GIMP
- Corel PaintShop Pro / Painter
- Adobe Photoshop

## Vector

- Inkscape
- Xara
- Corel Draw
- Adobe Illustrator

# Other Tools

- Android Asset Studio - <http://romannurik.github.io/AndroidAssetStudio/>
- Droid@Screen - project your device screen  
<http://droid-at-screen.ribomation.com/>
- DroidDraw - <http://droiddraw.org/>
- Android Layout Finder  
<http://www.buzzingandroid.com/tools/android-layout-finder/>
- AppInventor - <http://appinventorededu.mit.edu/>

# User Interface

## Fundamentals



# Android Screens

There are various screen sizes:

- Phones & Tablets
- Wearables
- TV
- Auto

# Screen Terminology

- Size
- Resolution
- Rate
- Orientation
- Density

Most Android devices have touch screens.

# Screen Size

The actual diagonal dimensions of your screen

- 5" Phone
- 10.1" Tablet
- 17" Laptop
- 22" Monitor
- 55" Television



# Screen Resolution

The count of pixels on the screen

- $640 \times 480 = 307200$
- $1024 \times 768 = 786432$
- $1920 \times 1080 = 2073600$



# Screen Rate

The times per second the screen is refreshed

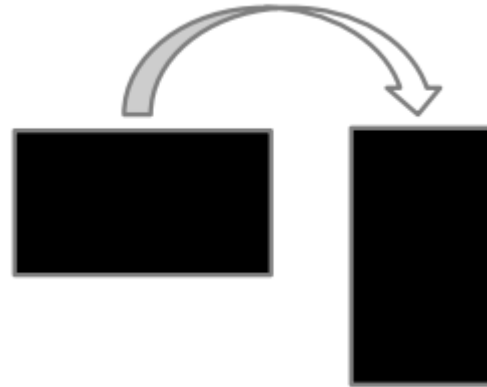
- Frame or Refresh Rate
- Varies significantly by technology



# Screen Orientation

The relative position of a screen to a user

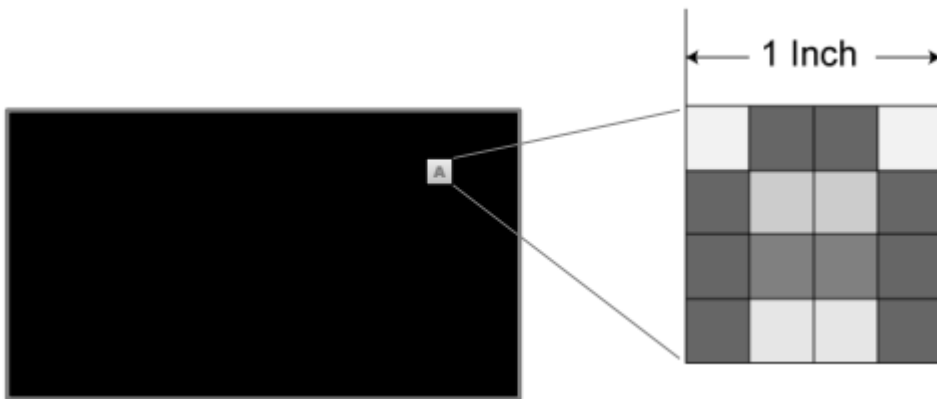
- Landscape
- Portrait



# Screen Density

The amount of pixels in an area of your screen

- Measured as dots in a line in an inch
- Pixels or Dots Per Inch (PPI/DPI)



# Density-Independent Pixels

## Density-independent Pixels (***dp*** or ***dip***)

- Physical Pixels  $px = dp * (dpi / 160)$
- “An abstract unit that is based on the physical density of the screen.”
- Equivalent to a 160 dpi screen - MDPI as the baseline density, so 160dp is always one inch regardless of the screen density.
- Android handles scaling of the dp units at runtime.



# Scale-independent Pixels (sp)

- Related to dp, but scaled by the font size preference.
- Scale factor depends on user settings
- Use ***sp*** when specifying font sizes

# 48dp Rhythm

- In general, a touchable UI element (Button or EditText for example) should be at least 48dp



- Android ActionBar has height of 48dp

# User Interface

## Definitions

# Android Screen Size Definitions

Newer Sizes are defined via Numeric selectors

- Available Width (in dp)
- Available Height (in dp)
- Smallest Width (in dp)

Older Generalized Sizes (Screen Pixel Density)

- small (426x320 dp)
- normal (470x320 dp)
- large (640x480 dp)
- xlarge (960x720 dp)

# Android Density Definitions

- *dpi (low)* ~120dpi
- *mdpi* (medium) ~160dpi ← baseline
- *hdpi* (high) ~240dpi
- *xhdpi* (extra-high) ~320dpi
- *xxhdpi* (extra-extra-high) ~480dpi
- *xxxhdpi* (extra-extra-extra-high) ~640dpi

# Density Independence

Depending on the current screen density

Android will:

- Scale Drawable resources to the right size
- Scale DIP units as appropriate

# Android Orientation

Orientation is an indication if the device is in a landscape or portrait configuration

Can be set via code and the manifest for an Activity

- `setRequestedOrientation()`
- `android:screenOrientation="landscape|portrait"`

# Android Rotation

Rotation is a measure of the difference from the device's "Natural" Orientation

- 0, 90, 180, 270 degrees
- Look to the front camera for a clue



# Android Rotation

Natural Orientation



0



90



180



270

# Coordinate System

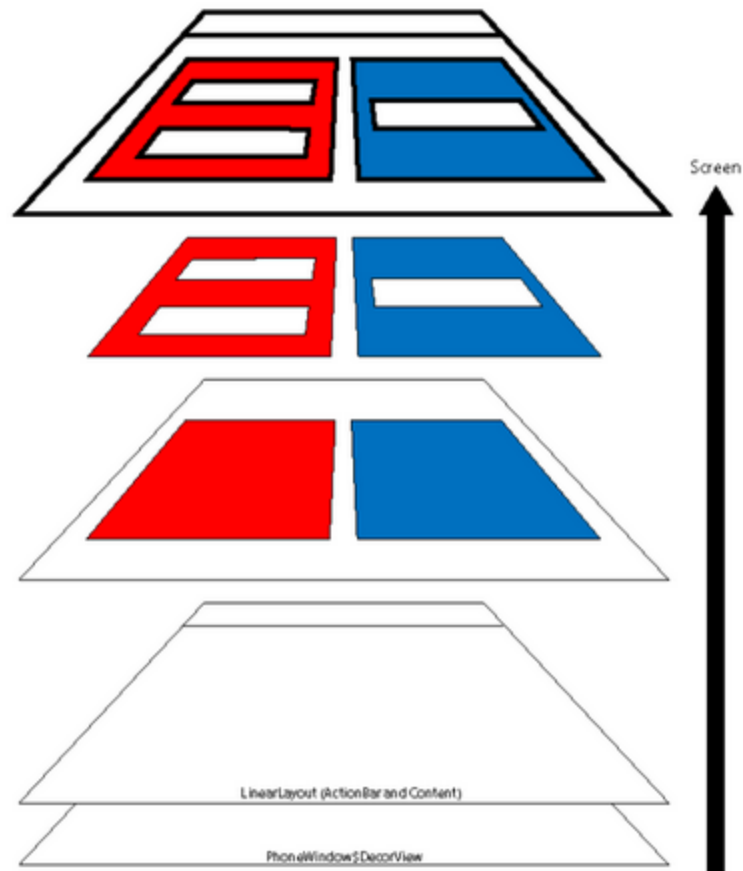
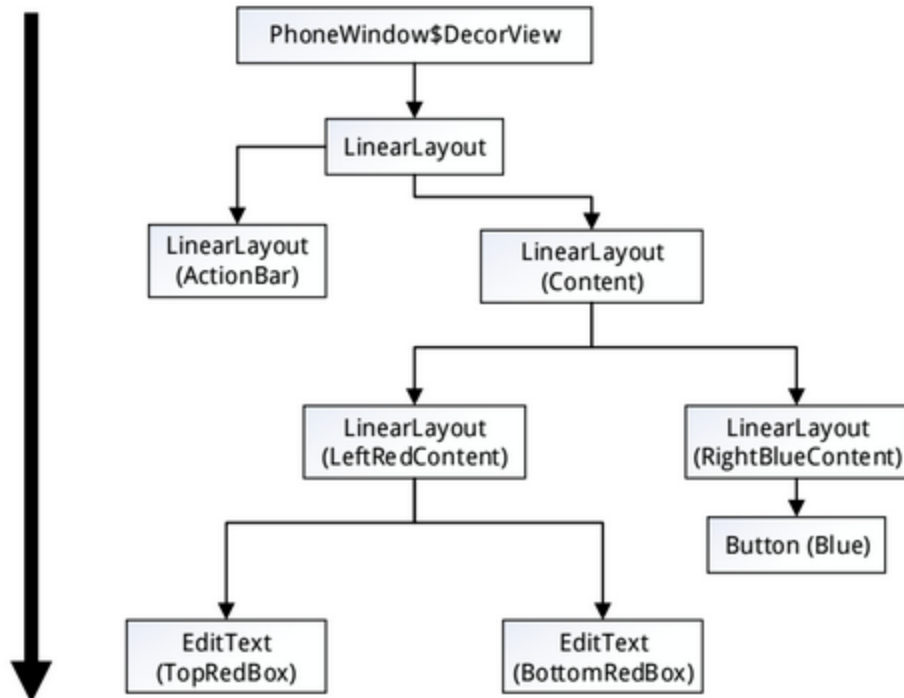
- Identify the unique position of a point in space
- Origin is the Top Left
- Android Views exist in X, Y, Z (depth) dimensions

# Views & ViewGroups

- Views: an UI element
- ViewGroups/Layouts: containers for positioning the views
- Layout Gravity vs. Gravity
- Margin vs. Padding
- View Visibility: Visible, Invisible & Gone

# View Hierarchy

- Tree structure of an application's views
- Every view is a rectangular area responsible for what gets drawn and potentially for what events that occur within that rectangle
- Child nodes are drawn over the parent nodes



Layers are composited from the Root up to the Screen

# Hierarchy Viewer

- Android Studio > Android Device Monitor
- Windows > Open Perspective > Hierarchy View
- Find package name and click on the blue hierarchy icon
- To see performance indicator - highlight root and click icon “Obtain layout times for tree rooted at selected node”

# UI Optimization

- Keep view hierarchy simple:
  - Views < 80
  - Levels < 10
- Use RelativeLayout for complex positioning
- Use Framelayout if there is only one child
- Remove unnecessary layouts
- Use Compound drawable to replace multiple views (i.e An ImageView next to a TextView)

# Next week

- Homework 1 requirements
- We will cover Styles and Themes.