GDG DevFest
2013 Season

year.mm.dd Meeting Name @ **Where**

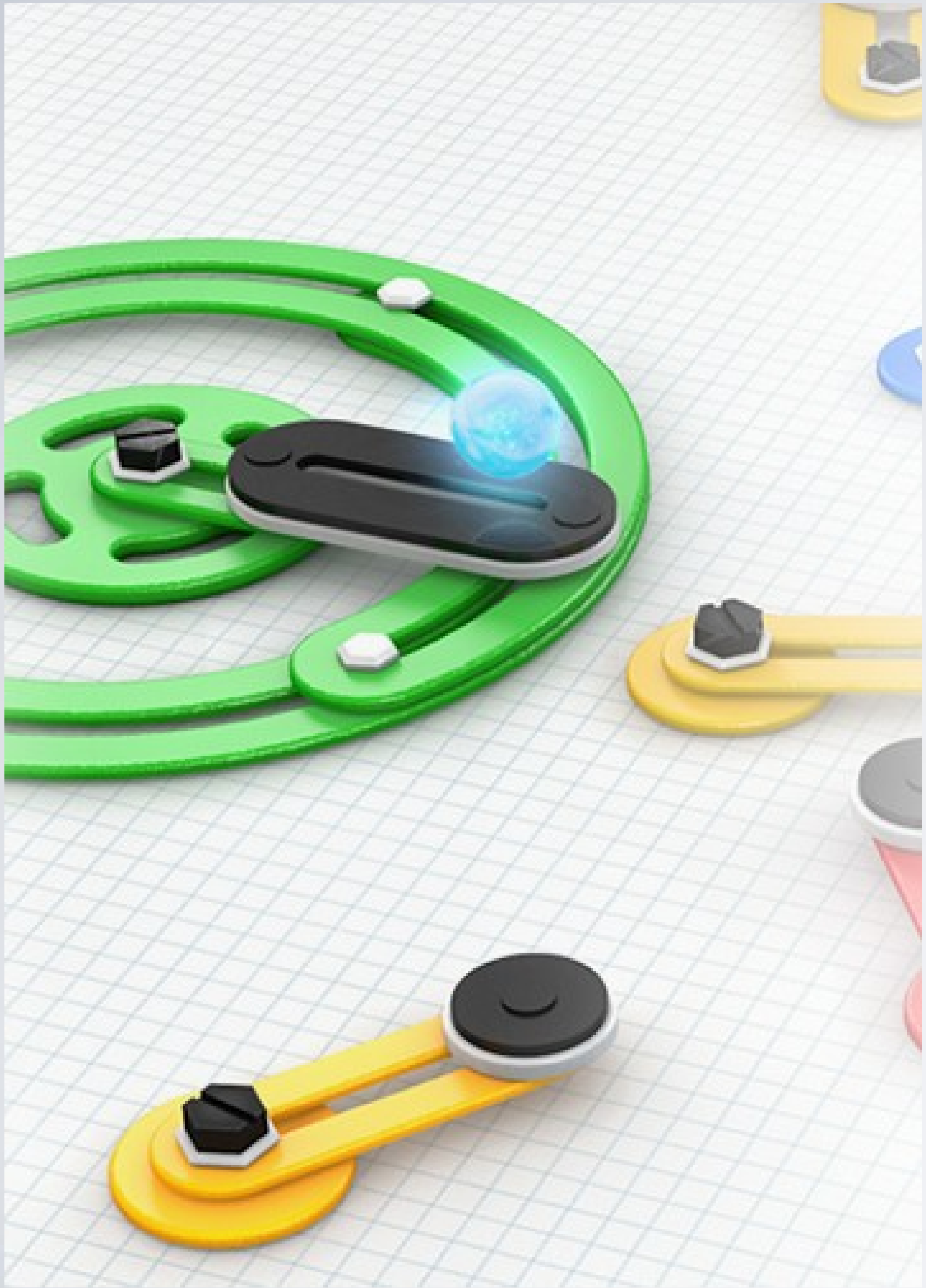
Give us feedback!

Fill out this survey!

http://goo.gl/B3pJeR

Lesson 1:

Describing the Android Platform



Describing the Android Development Environment

Android Studio

Android Studio

=

Android development environment based on IntelliJ IDEA

To help you build, test, debug, and package your Android apps

Android Studio was in **early access preview**, as of Sept. 2013.

Android Studio

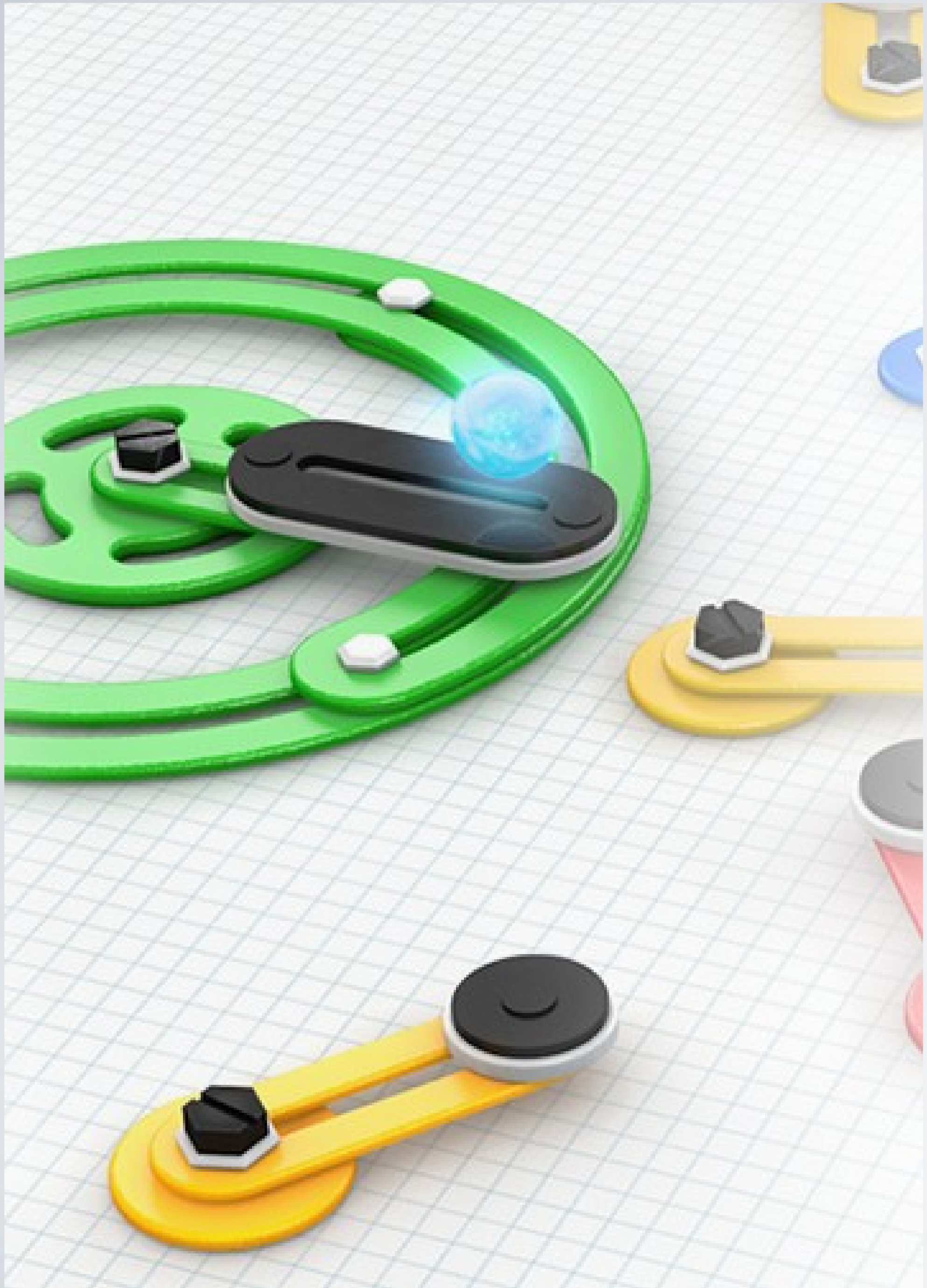
What You Get with Android Studio Based on IntelliJ IDEA

Highlights of IntelliJ

- Full Java IDE
- Graphical UI Builders
- Powerful Debugging

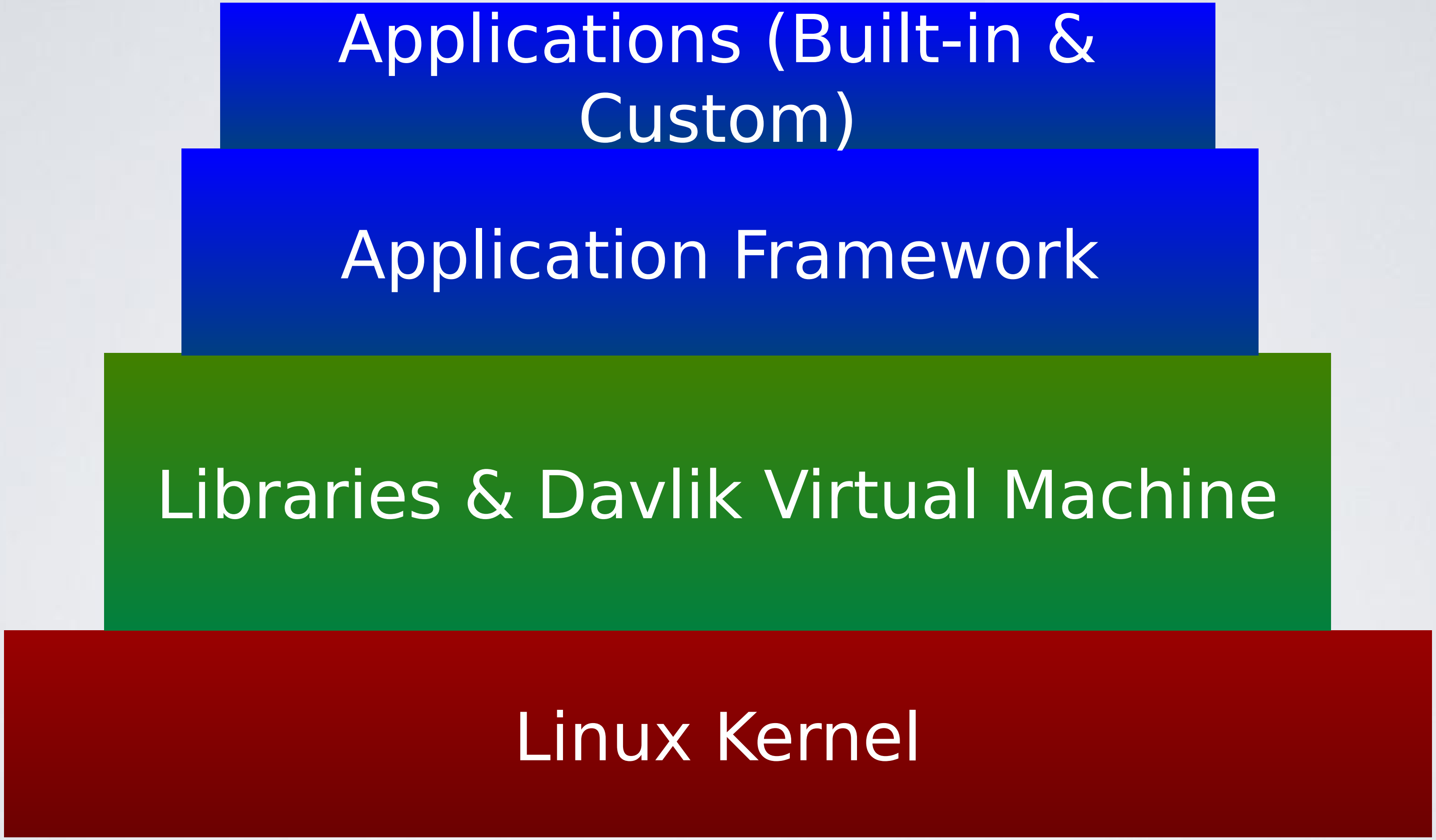
Highlights of Android Studio

- On-device Developer Options
- Develop on Hardware Devices
- Develop on Virtual Devices
- Native Development
- Testing



Anatomy of an Android app

Anatomy of an Android app



Applications (Built-in & Custom)

Application Framework

Libraries & Davlik Virtual Machine

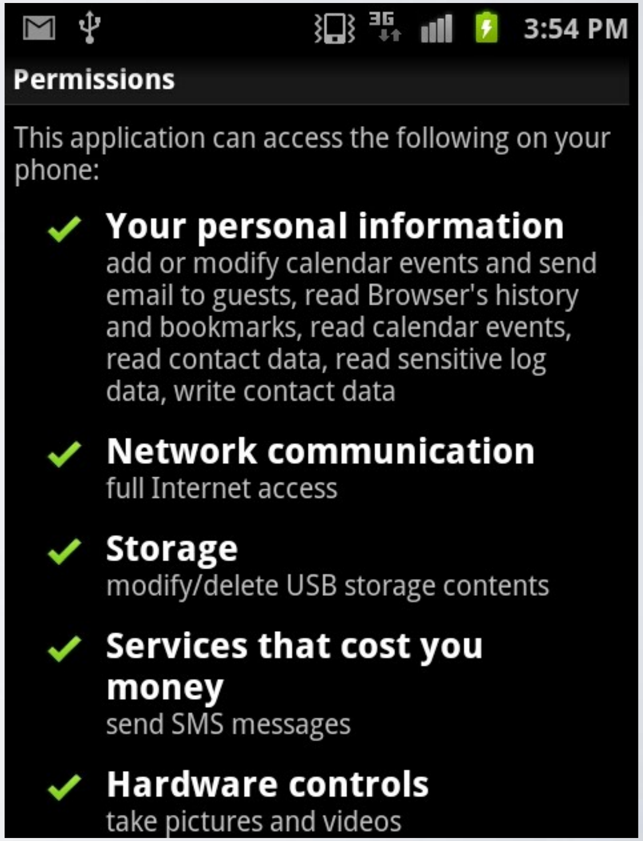
Linux Kernel

Anatomy of an Android app

Anatomy of an Android app

Android App Security

- Runs inside a sandbox as a separate UID (Linux User ID).
- Framework restricts access.
- Privileges can be requested for additional access



Permissions

This application can access the following on your phone:

✓ **Your personal information**
add or modify calendar events and send email to guests, read Browser's history and bookmarks, read calendar events, read contact data, read sensitive log data, write contact data

✓ **Network communication**
full Internet access

✓ **Storage**
modify/delete USB storage contents

✓ **Services that cost you money**
send SMS messages

✓ **Hardware controls**
take pictures and videos

Android project folder structure

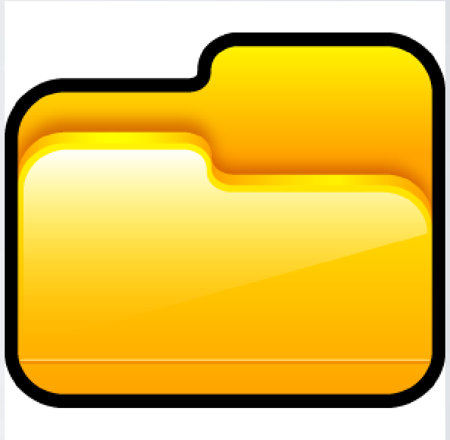
Android Project Folder Structure

- AndroidManifest.xml - Fundamental characteristics

  of your app

- src/main/res - Directory for your app's main source files

- src/res/ - Contains several sub-directories for app resources
  - drawable-hdpi/ - Directory for drawable objects, designed for a specific screen
  - layout/ - Directory for files that define your app's user interface
  - values/ - Directory for other XML files that contain a



Android project folder structure

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ...>
    <uses-feature ... />
    <uses-permission ... />
    <uses-sdk  android:minSdkVersion="3" android:targetSdkVersion="5" android:maxSdkVersion="5" />
    <application ...>
        <activity ...>
            ...
        </activity>
        <service ...>
          ...
        </service>
        <provider ...>
           ...
        </provider>

        <receiver ...>
          ...
        </receiver>
    </application>
</manifest>
```

Android project folder structure

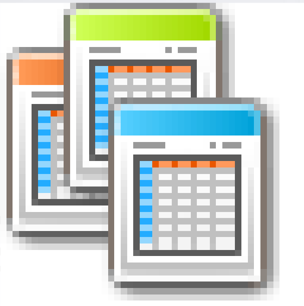
App Resources

res/	
drawable	Drawable XML
drawable-xhdpi drawable-hdpi drawable-mdpi	PNGs, 9-patch PNGs, optimized for multiple densities
layout layout-land layout-large layout-large-land	Layout XML optimized for physical screen size and orientation
values	Strings, styles, themes, etc.
values-v11 values-v14	Styles, themes varying by API level
values-en values-fr values-ja	Strings XML localized for your target regions

Key components of an Android app

Activities & Services

Activities



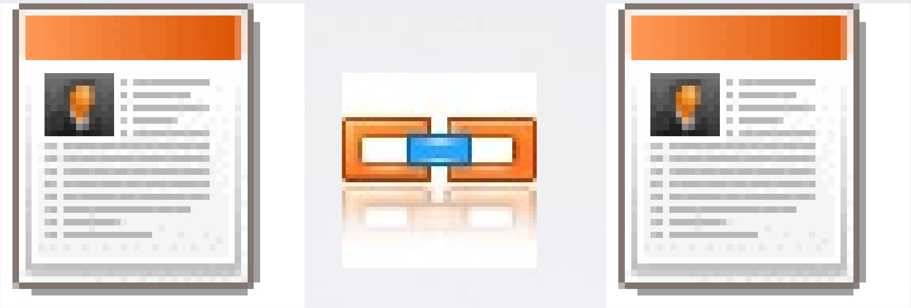
Manag creen the user interacts with

Services

Perfor round operations for your app

Key components of an Android app

Intents



Provide the “links” between your classes

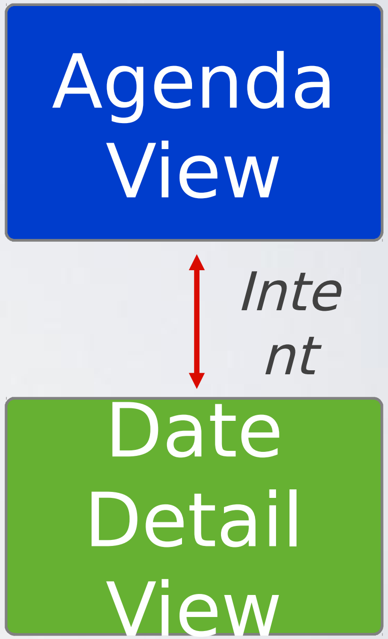
Key components of an Android app

Apps communicate with each other by providing and consuming each other's Intents.

Agenda View

Key components of an Android app

Apps communicate with each other by providing and consuming each other’s Intents.



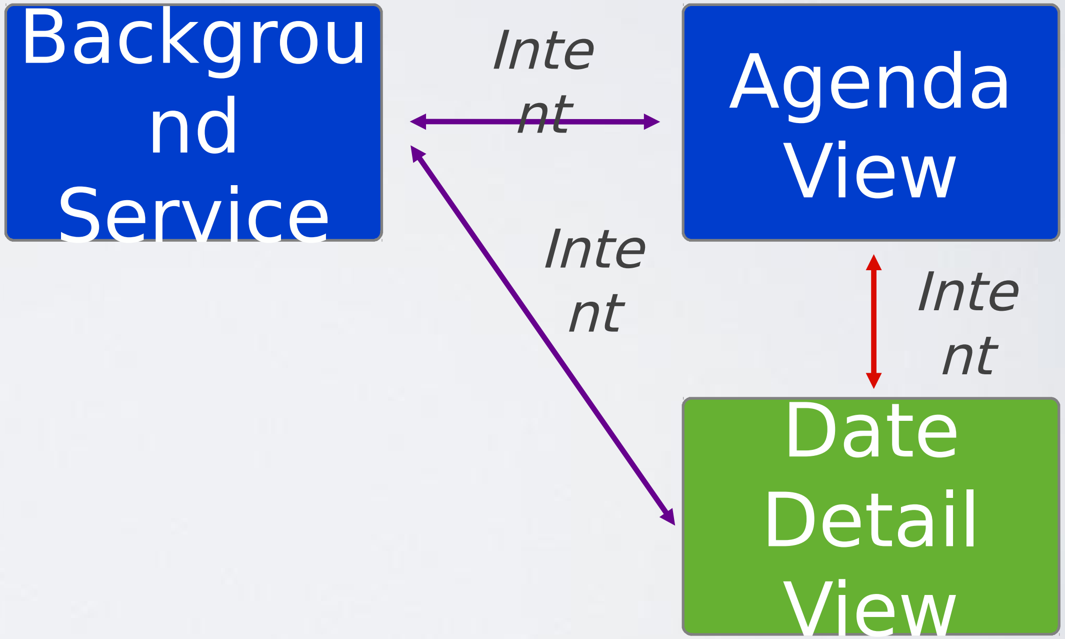
Agenda View

Inte
nt

Date Detail View

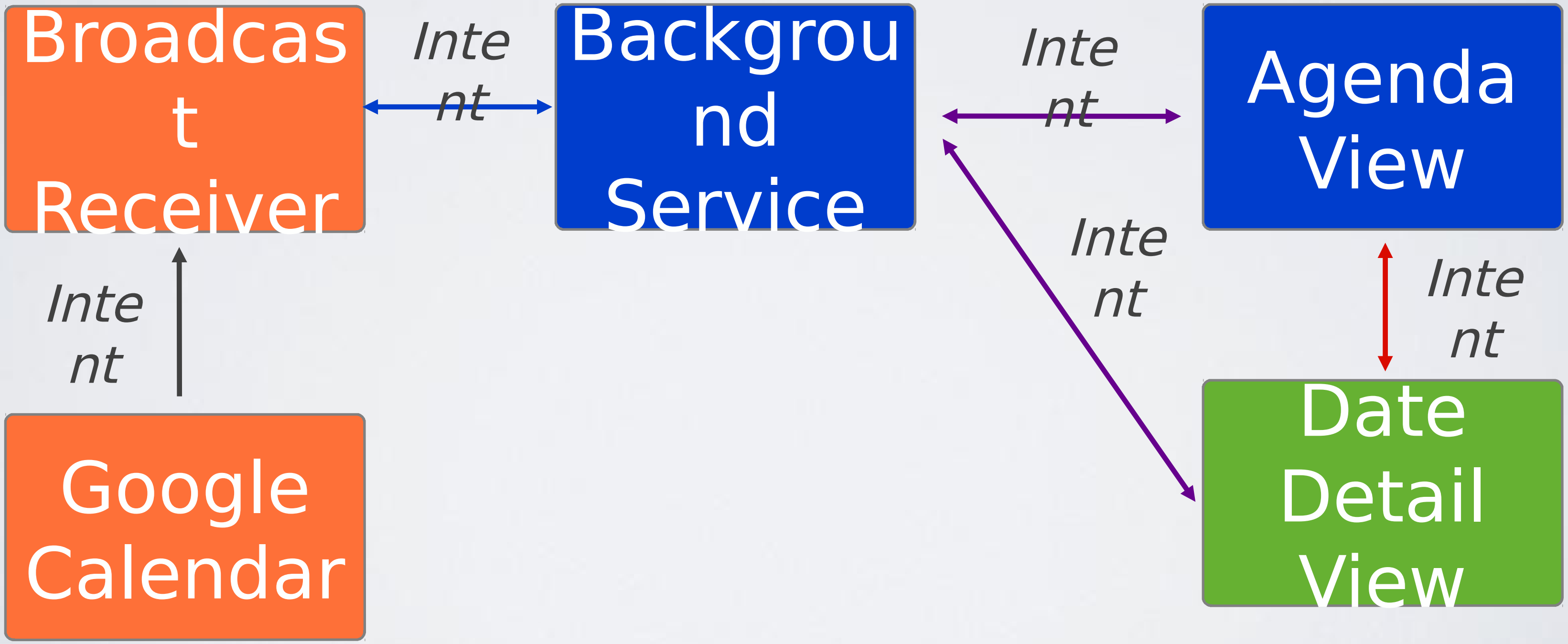
Key components of an Android app

Apps communicate with each other by providing and consuming each other's Intents.



Key components of an Android app

Apps communicate with each other by providing and consuming each other's Intents.



Lab Exercise 1.1

Creating Android Projects

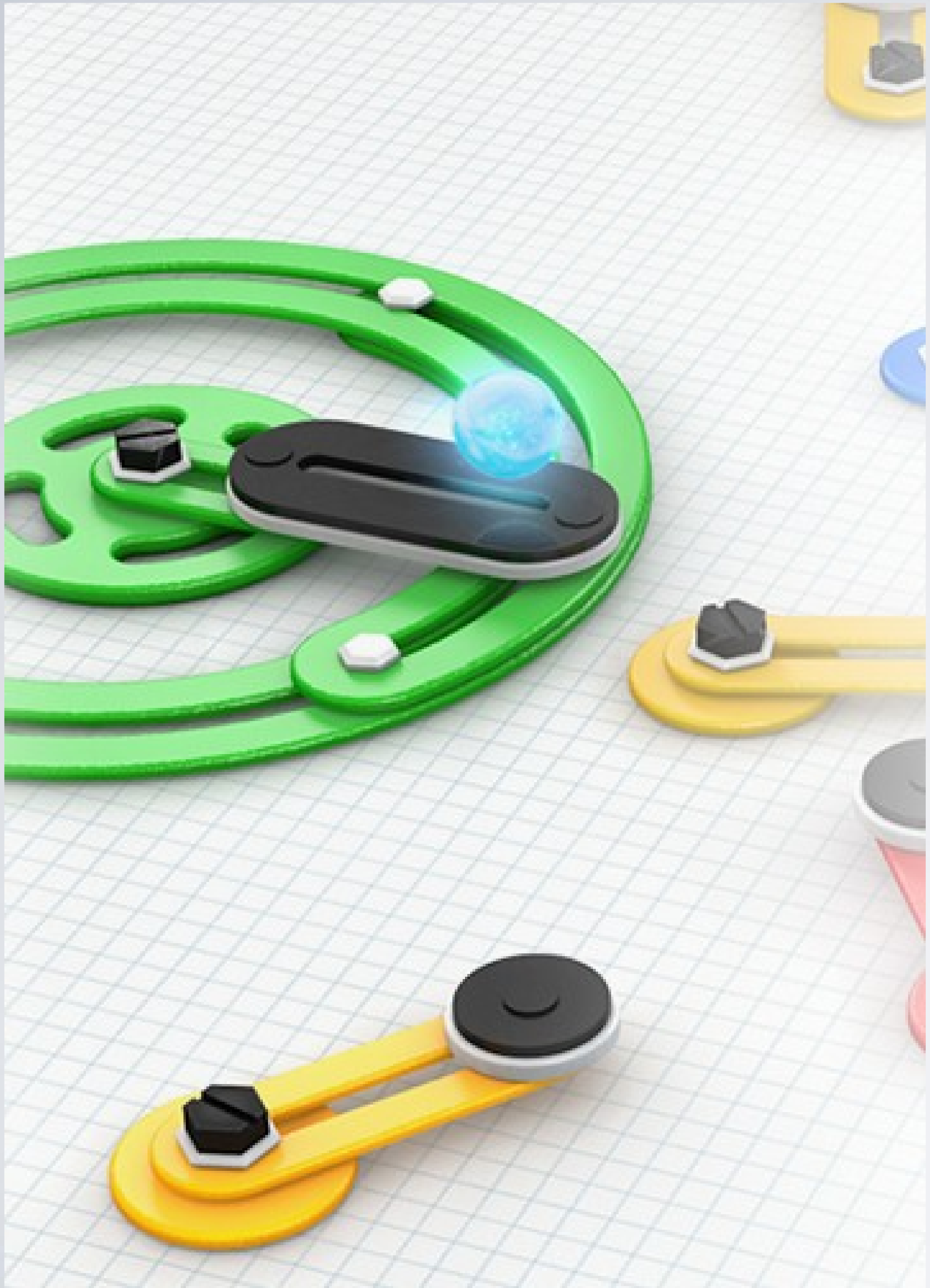
Lab Exercise 1.1

Exploring Android Studio

- Task 1: Verify proper installation of Android Studio and the Android SDK.
- Task 2: Create a new Android Studio project.
- Task 3: Navigate the Android Studio project explorer and identify source and resource files.
- Task 4: Use intelligent features of the code editor, including code-complete and refactoring.
- Task 5: Change the Android Studio skin (optional).
- Task 6: Connect a hardware device.
- Task 7: Create and start a virtual device.
- Task 8: Run a project on a virtual and hardware device.
- Task 9: View the project structure.

Lesson 2:

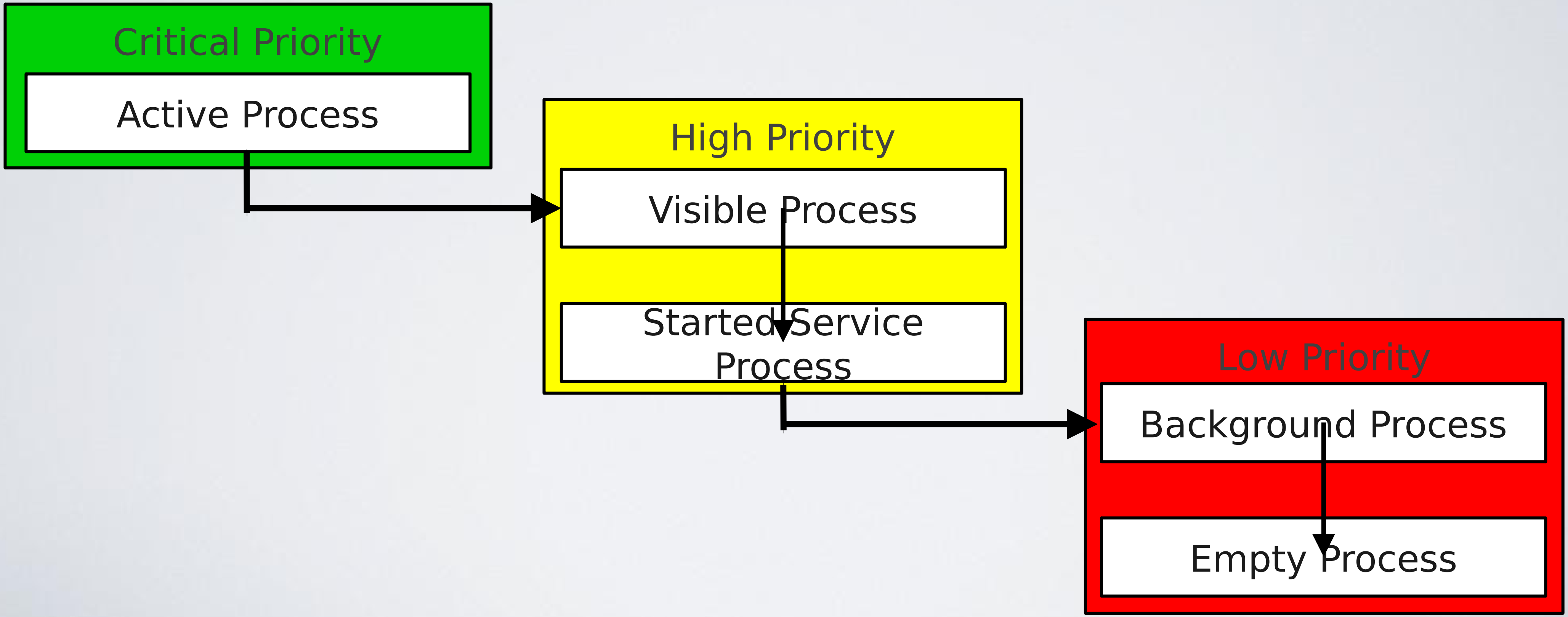
Android Application Types & Fundamental Classes

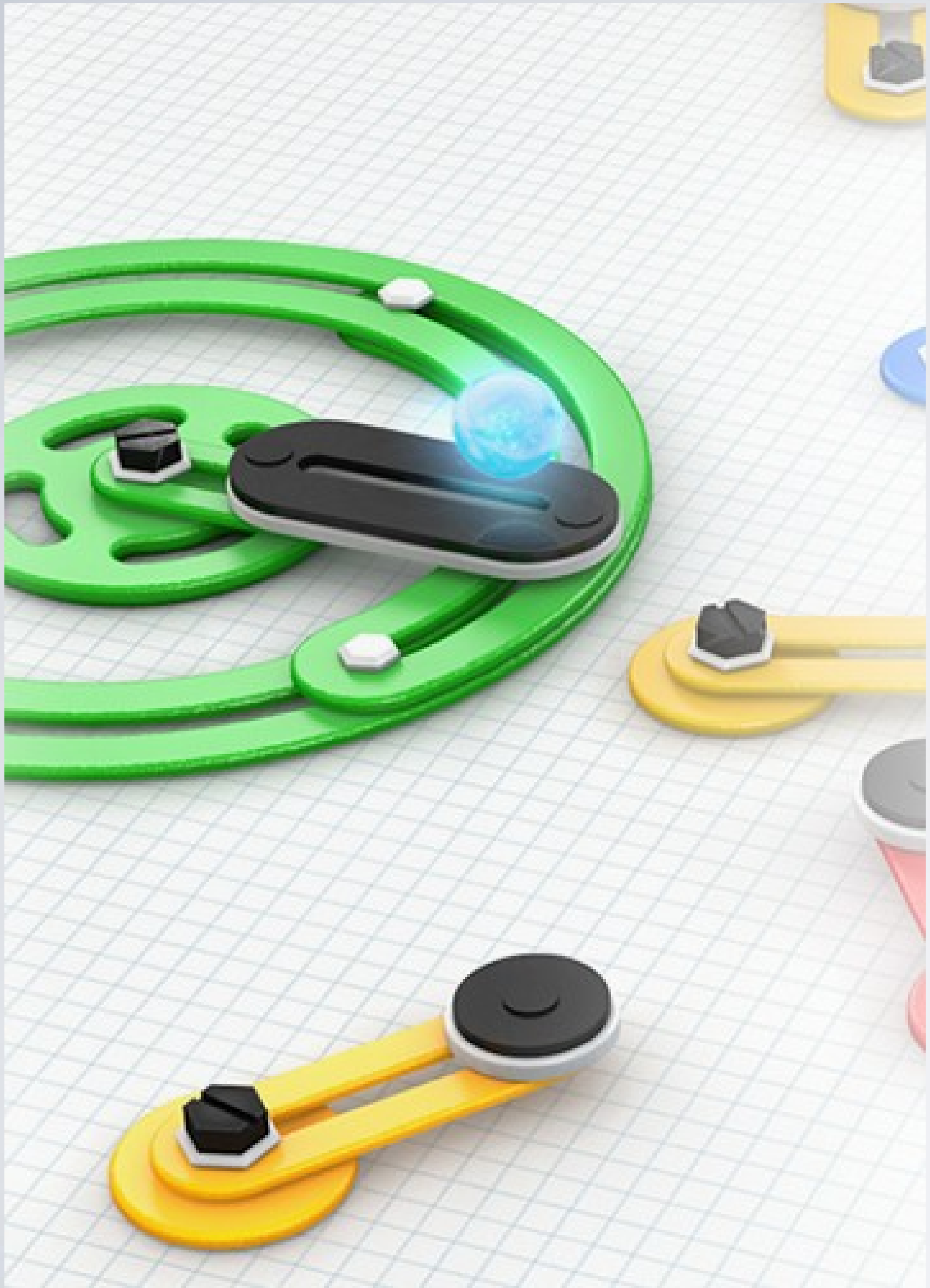


Describing the Application Lifecycle

Describing the Application Lifecycle

Memory and Process Management





Describing the key Android classes and how they relate

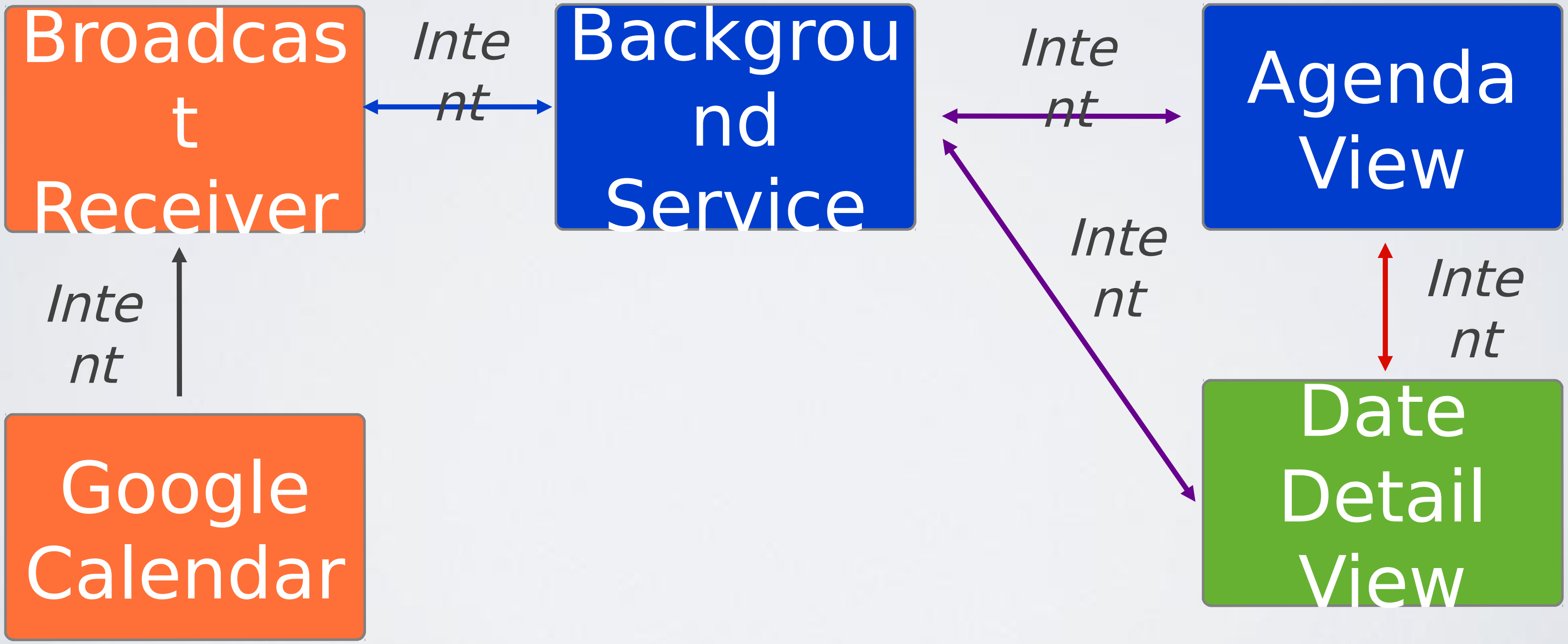
Describing the Fundamental Android Components

Overview of Android Fundamental Components

- *Activities* - Manage the screen the user interacts with
- *Intents* - Provide the “links” between your classes
- *Services* - Perform background operations for your app
- *BroadcastReceiver* - Receives Intents from other apps
- *ContentProvider* - Connects data between

Key components of an Android app

Apps communicate with each other by providing and consuming each other's Intents.



Describing the Fundamental Android Components

Activity

- An Activity provides a screen with which users can interact.

- An Activity uses a Window to draw its user interface.

- An application consists of multiple Activities loosely bound to each other.

Describing the Fundamental Android components

Intent

Passive data structure holds an abstract description of an operation to be performed.

Messages facilitate late run-time binding between components in the same or different applications.

An Intent object is passed to an Activity, Service, or set of broadcast receivers.

Describing the Fundamental Android Components

Service

Does not provide a user interface.

Can perform long-running operations in the background.

Continues to run even if the user switches to another app.

Can bind to a service to interact with it and even perform interprocess communication (IPC).

Describing the Fundamental Android Components

Service - Started vs. Bound

Started

Once started, a service can run in the background indefinitely.

Bound

Runs only as long as another application component is bound to it.

Describing the Fundamental Android Components

BroadcastReceiver

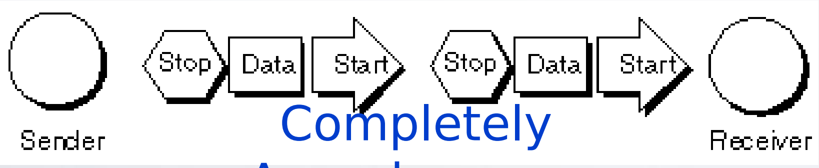
- You can dynamically register an instance of this class.
- You can statically publish an implementation.
- It is an important part of an application's overall lifecycle.

Describing the Fundamental Android Components

BroadcastReceiver - Broadcast Types

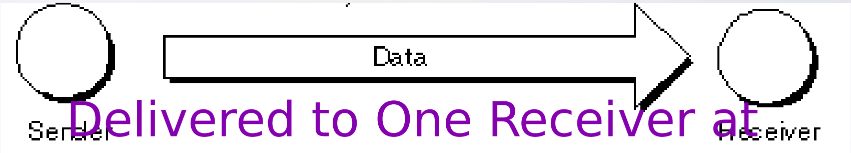
Two major classes of broadcasts can be received:

Normal Broadcasts



Completely Asynchronous

Ordered Broadcasts

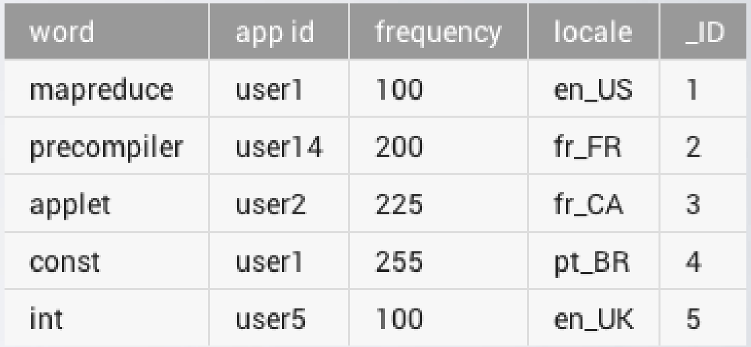


Delivered to One Receiver at a Time

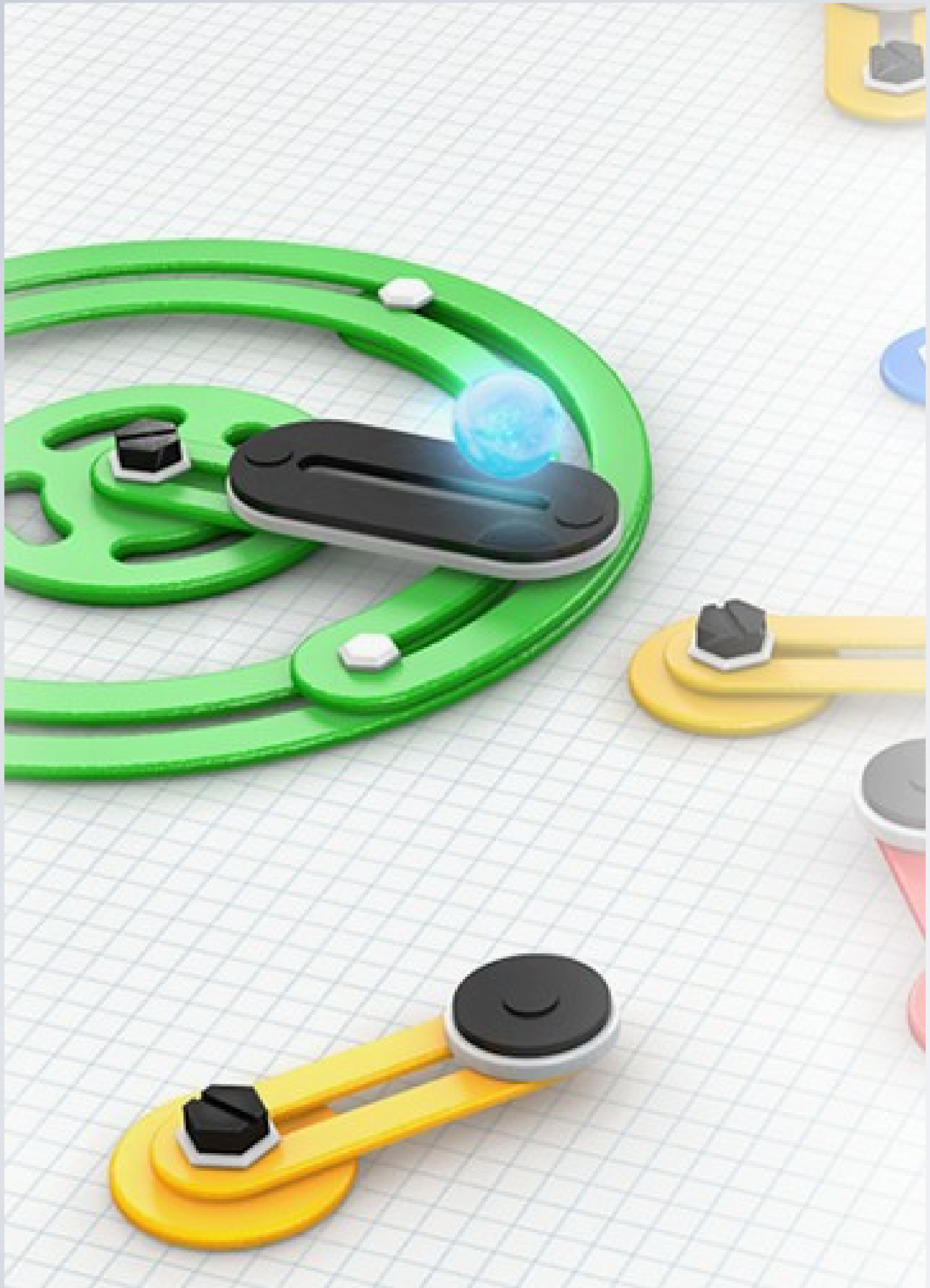
Describing the Fundamental Android Components

ContentProvider

- Manages access to a structured set of data.
- Encapsulates the data and provides mechanisms
  for defining data security.
- Is a standard interface that connects data in one process with code running in another process.
- You don't need to develop your own provider
  if you don't intend to share your data with other applications.



word	app id	frequency	locale	_ID
mapreduce	user1	100	en_US	1
precompiler	user14	200	fr_FR	2
applet	user2	225	fr_CA	3
const	user1	255	pt_BR	4
int	user5	100	en_UK	5



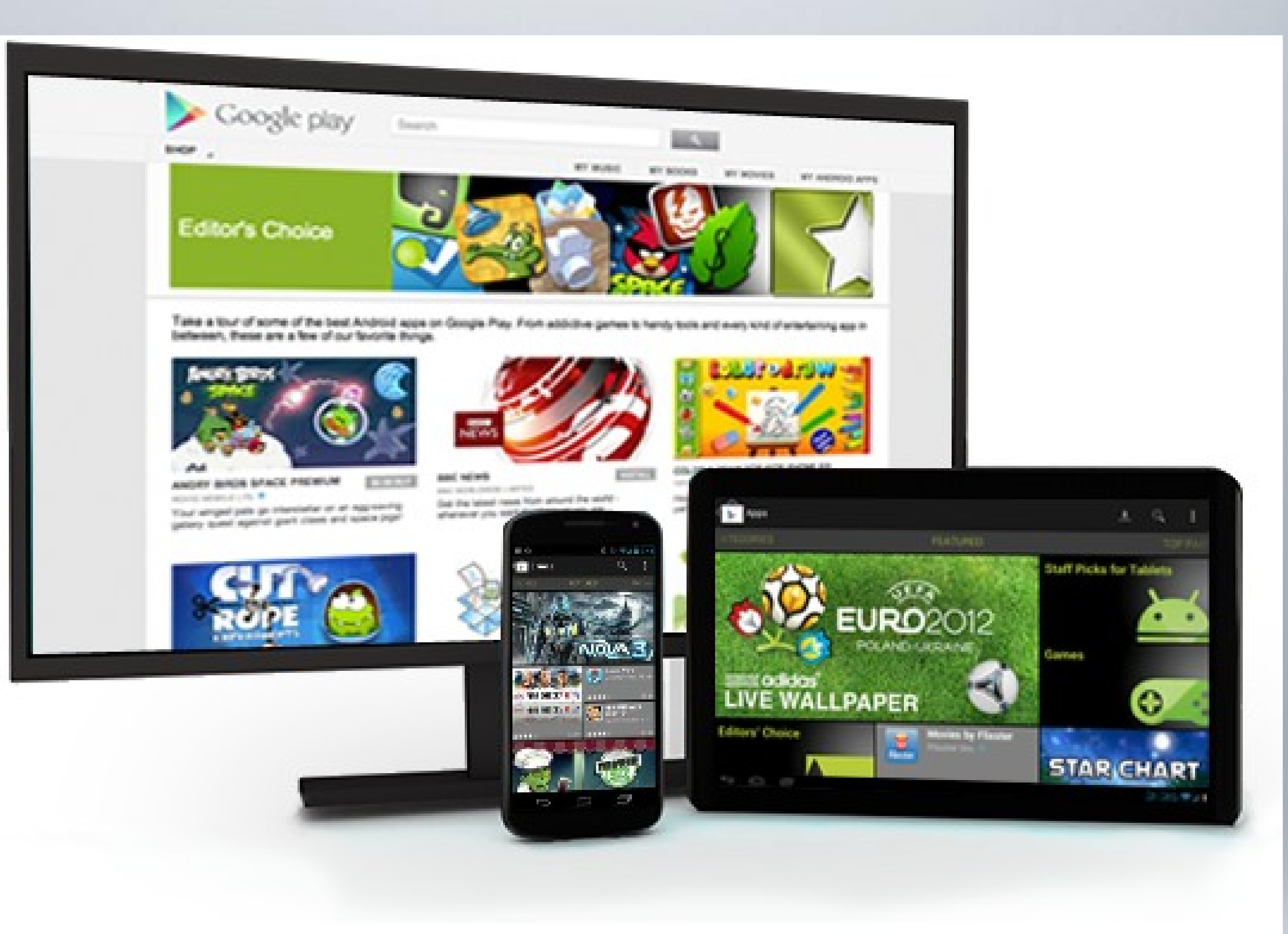
Types of Android Applications

Lab Exercise 2.1

Types of Android Applications

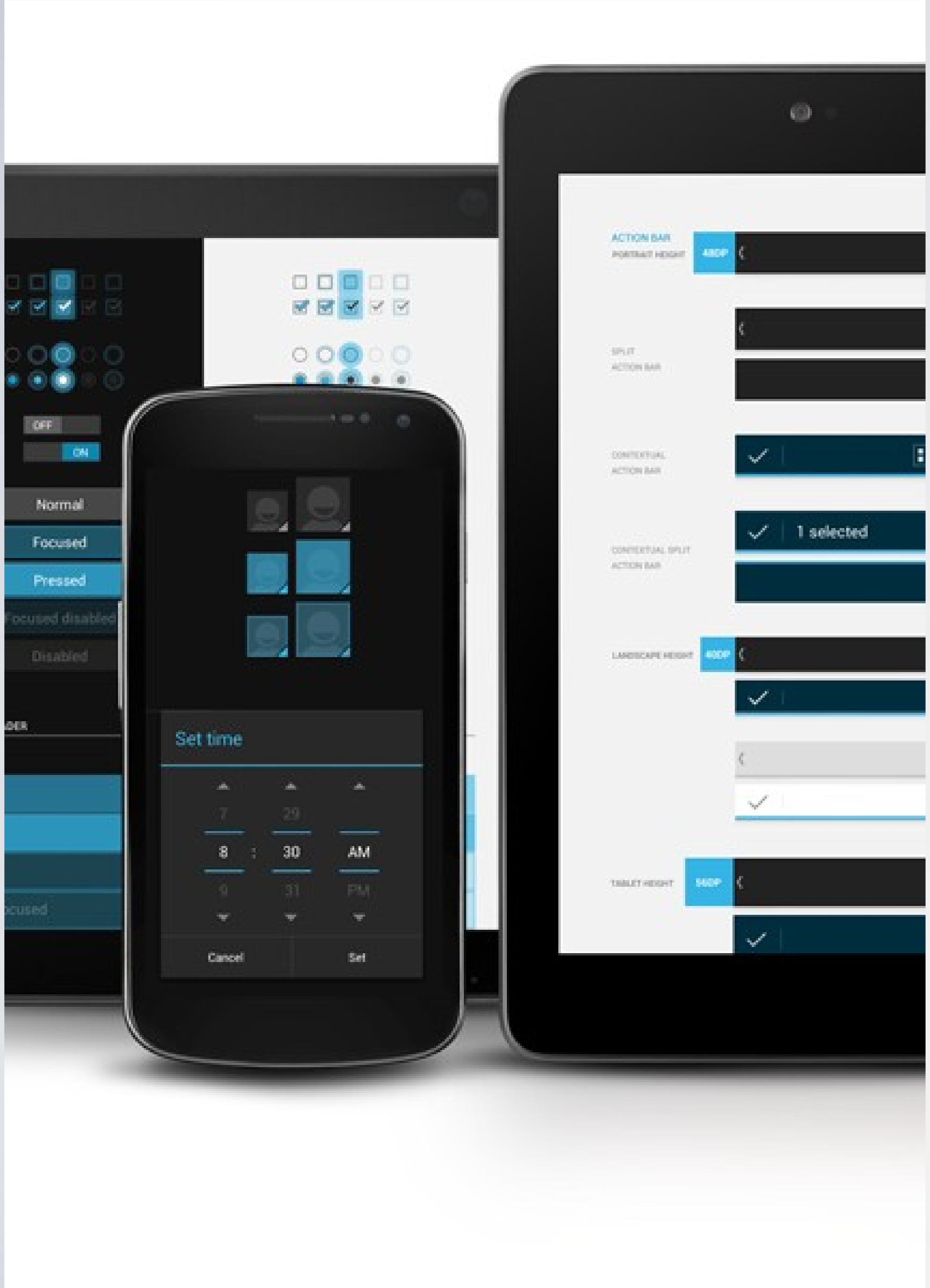
Lab Exercise 2.1

- App Analyze!
- In this lab you examine a popular app from the Play Store and analyze its fundamental Android components.



Lesson 3:

Android User Interface



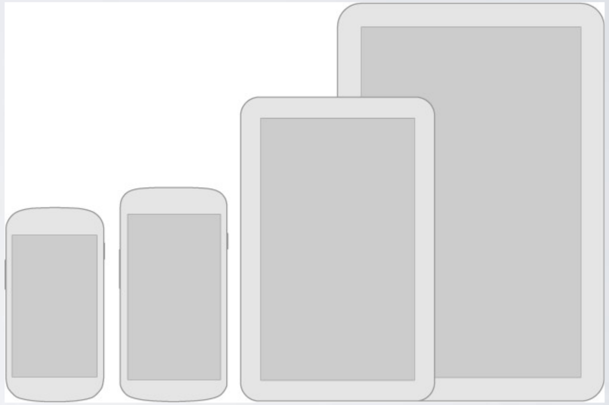
BUILDING AN ANDROID USER INTERFACE: GETTING STARTED

UI design and
the mobile touch environment

Considerations When Designing for Android

Touch

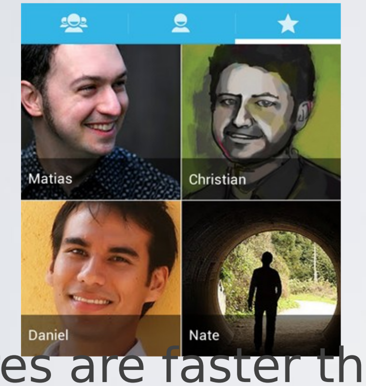
Mobility



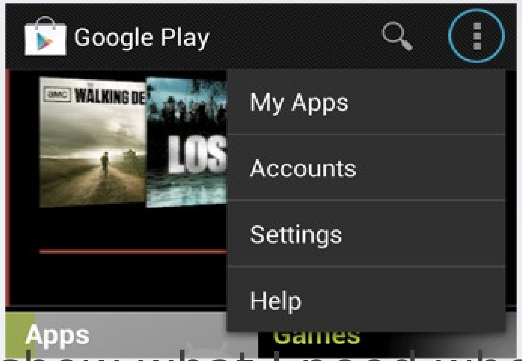
Heterogeneity

UI design and the mobile touch environment

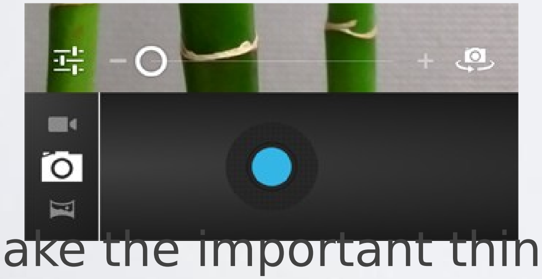
Key Principles of Android UI Design



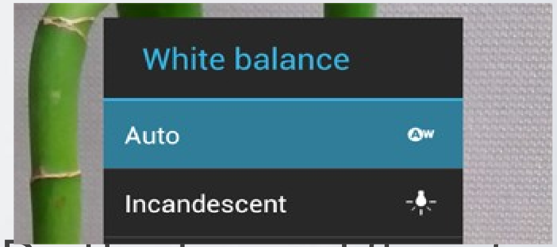
“Pictures are faster than words.”



“Only show what I need when I need it.”



“Make the important things fast.”



“Do the heavy lifting for me.”

Creating a wireframe

Why Create Wireframes?

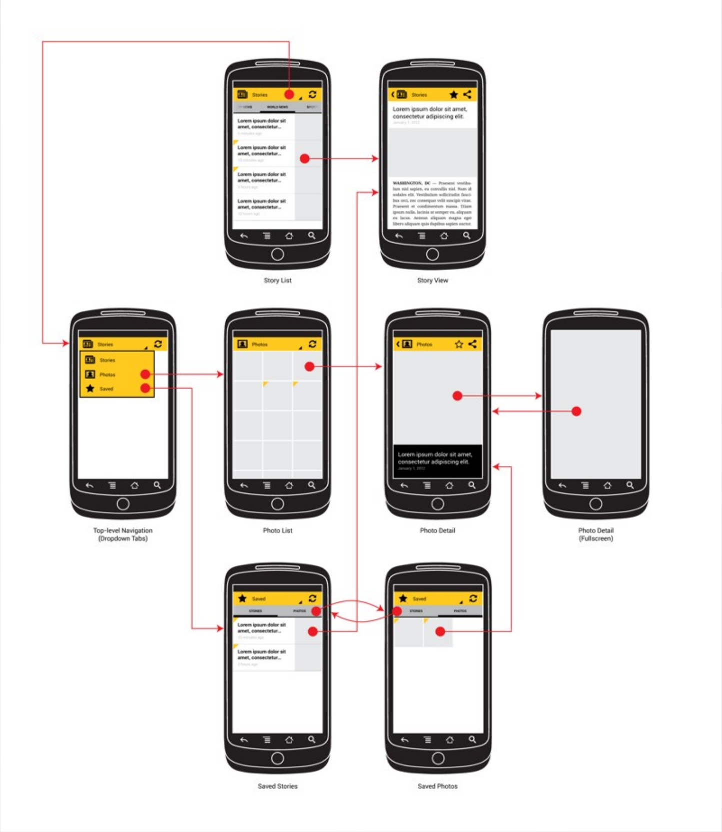
Wireframing before coding saves you time.

Always start with pencil and paper (or a whiteboard).

Creating a wireframe

Wireframe Examples

Wireframe Sketches



Digital Wireframe

BUILDING AN ANDROID USER INTERFACE: GETTING STARTED

Quiz Questions

Wireframes help you:

- Record your ideas
- Assess your app from a high-level user point of view
- Save you a lot of time
- All of the above
- None of the above

Which of the following are considerations when designing for Android?

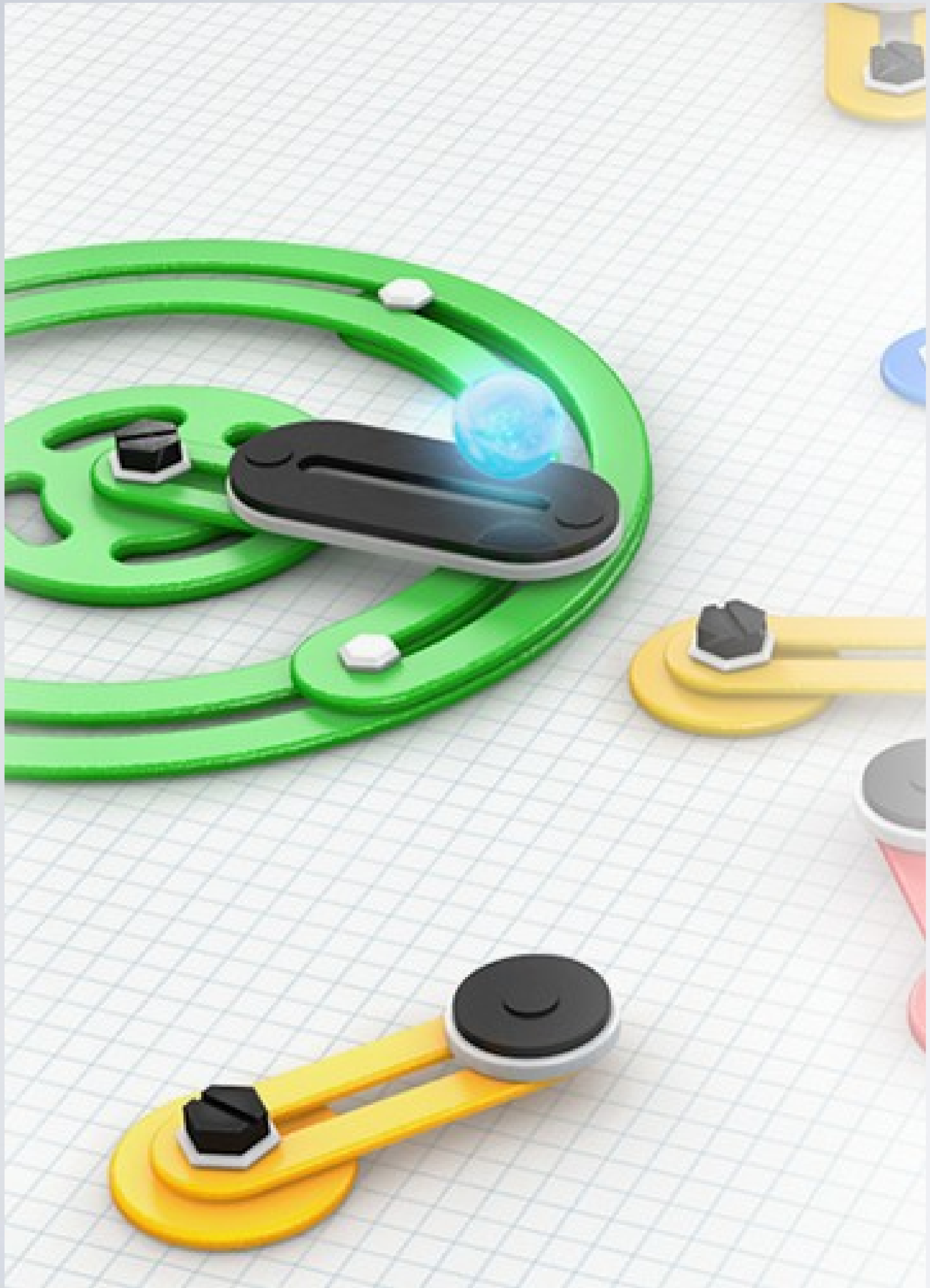
- Mobile
- Heterogeneity
- Touch
- All of the above
- None of the above

You should use your wireframes to...

- Re-arrange, add, and remove interactions quickly
- Scope out UI complexity
- Both of the above
- None of the above

You should start drawing your wireframes using Keynote or Powerpoint.

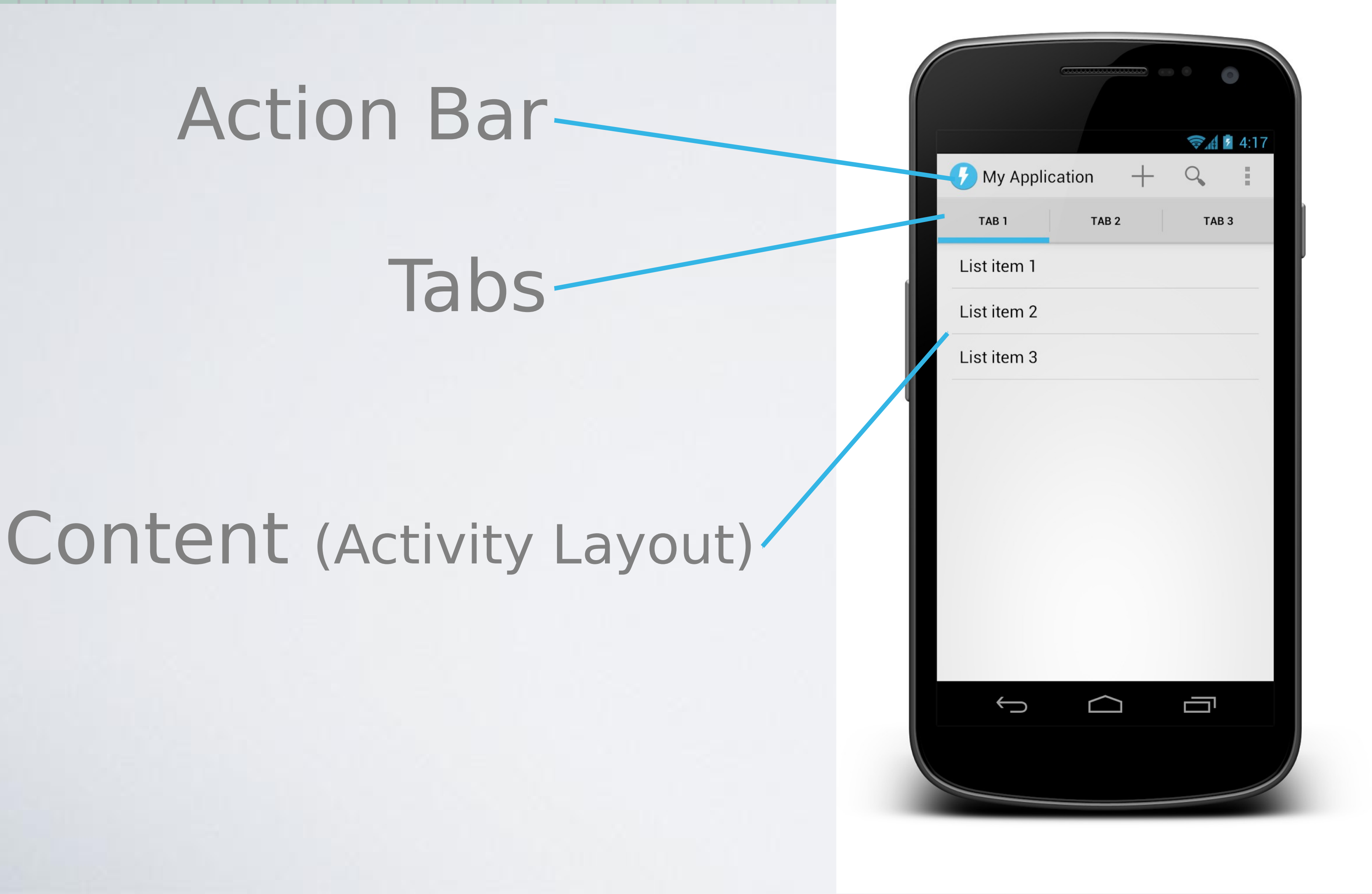
- True
- False



Understanding the Android user interface XML

Understanding the Activity layout structure

Overview of the Activity Layout Structure



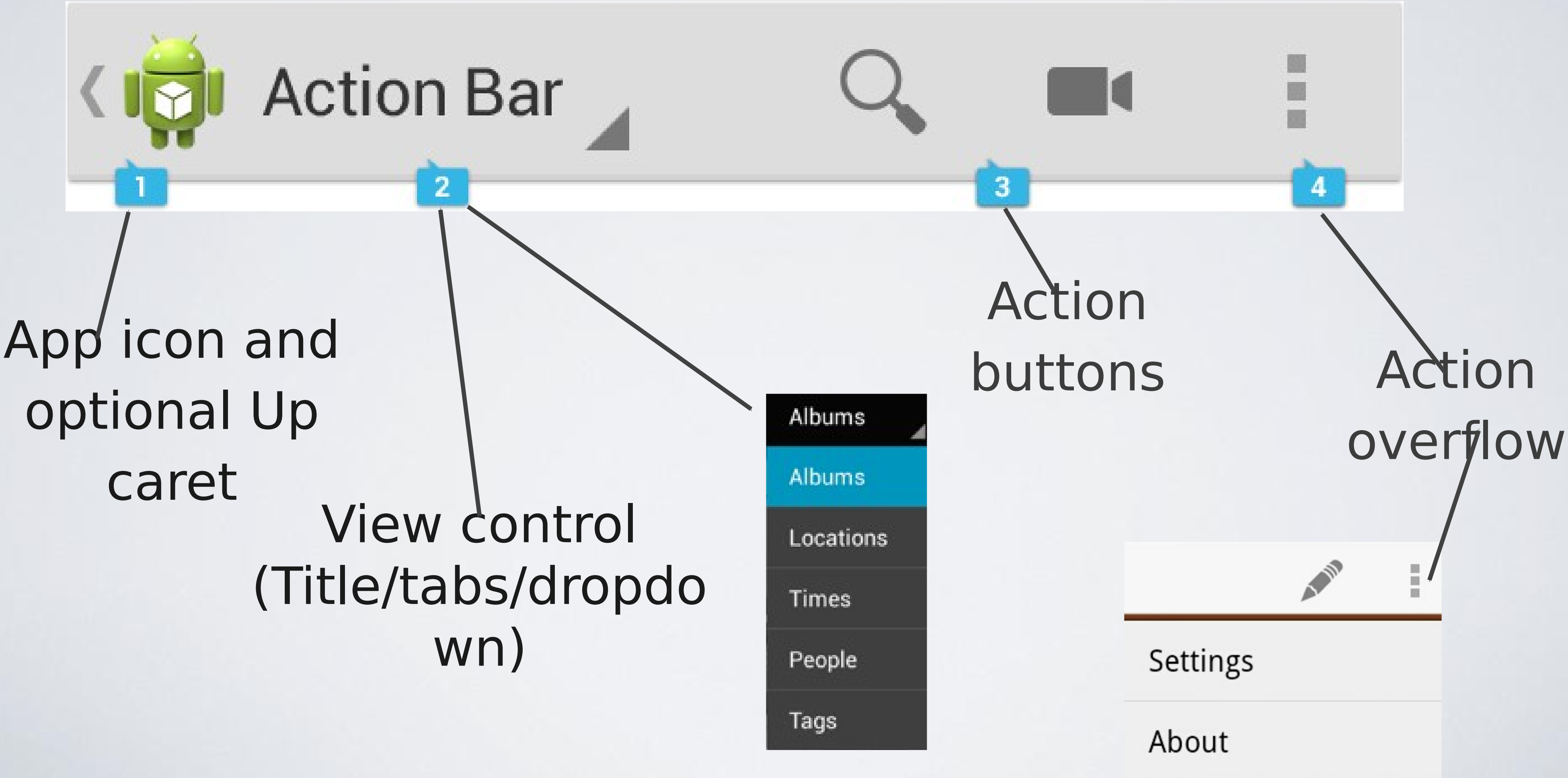
Action Bar

Tabs

Content (Activity Layout)

Understanding the Activity layout structure

Understanding the Action Bar Element



App icon and optional Up caret

View control (Title/tabs/dropdo wn)

Action buttons

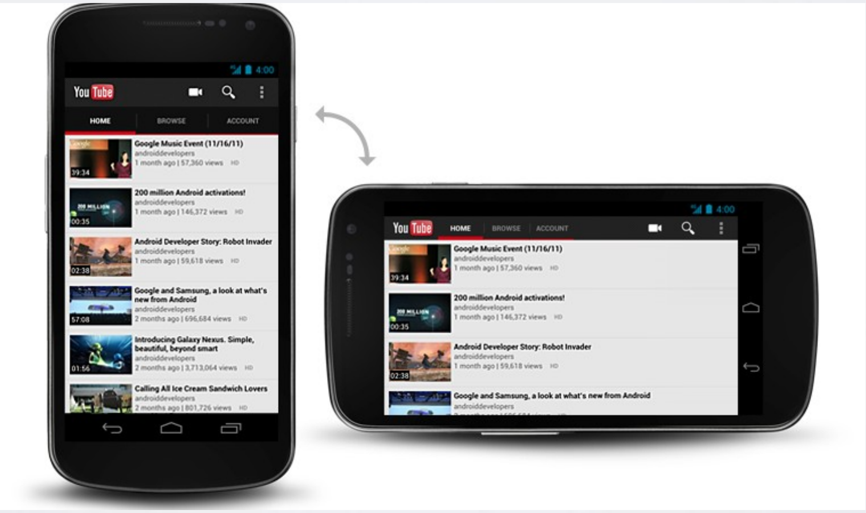
Action overflow

Understanding the Activity layout structure

Understanding the Tab Element

```
getActionBar().setNavigationMode(NAVIGATION_MODE_TABS);

ActionBar.Tab tab = actionBar.newTab();
tab.setText(“Tab 1”);
tab.setTabListener(this);
getActionBar().addTab(tab);
```


4:00
You Tube
HOME
BROWSE
ACCOUNT
Google Music Event (11/16/11)
androiddevelopers
1 month ago | 57,360 views HD
39:34
200 million Android activations!
androiddevelopers
1 month ago | 146,372 views HD
00:35
Android Developer Story: Robot Invader
androiddevelopers
1 month ago | 59,618 views HD
02:38
Google and Samsung, a look at what's new from Android
androiddevelopers
2 months ago | 696,684 views HD
57:08
Introducing Galaxy Nexus. Simple, beautiful, beyond smart
androiddevelopers
2 months ago | 3,713,064 views HD
01:56
Calling All Ice Cream Sandwich Lovers
androiddevelopers
2 months ago | 801,726 views HD

Understanding Views and ViewGroups

Views

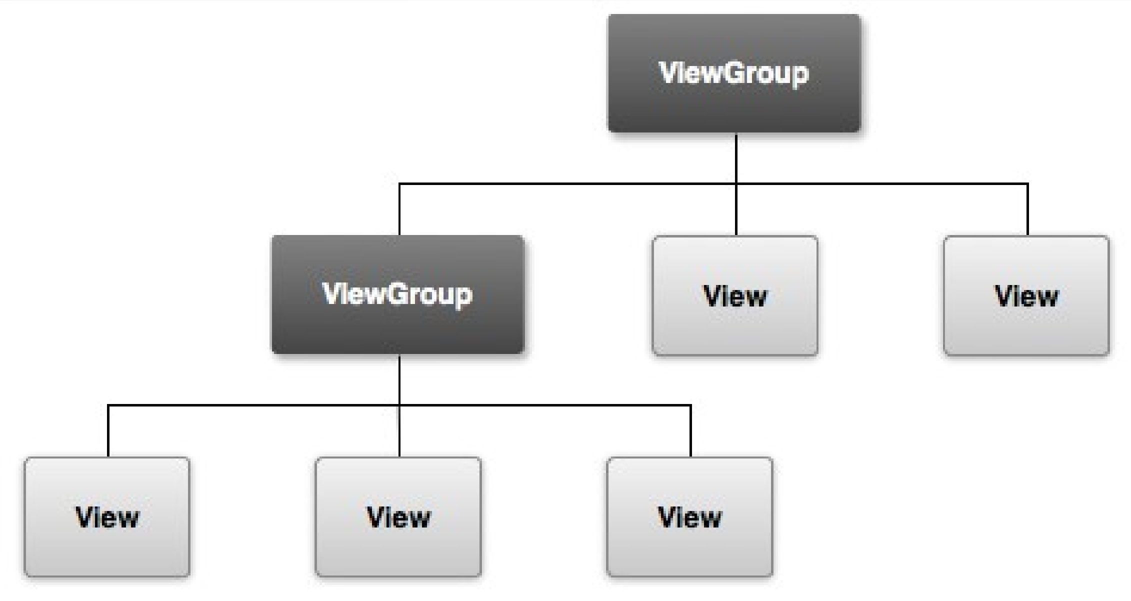
- Reusable individual UI components
- Optionally interactive (clickable/focusable/etc.)
- Bare minimum functionality
  is to draw themselves

ViewGroups

- Ordered list of Views and ViewGroups
- Positions and sizes child views and layouts

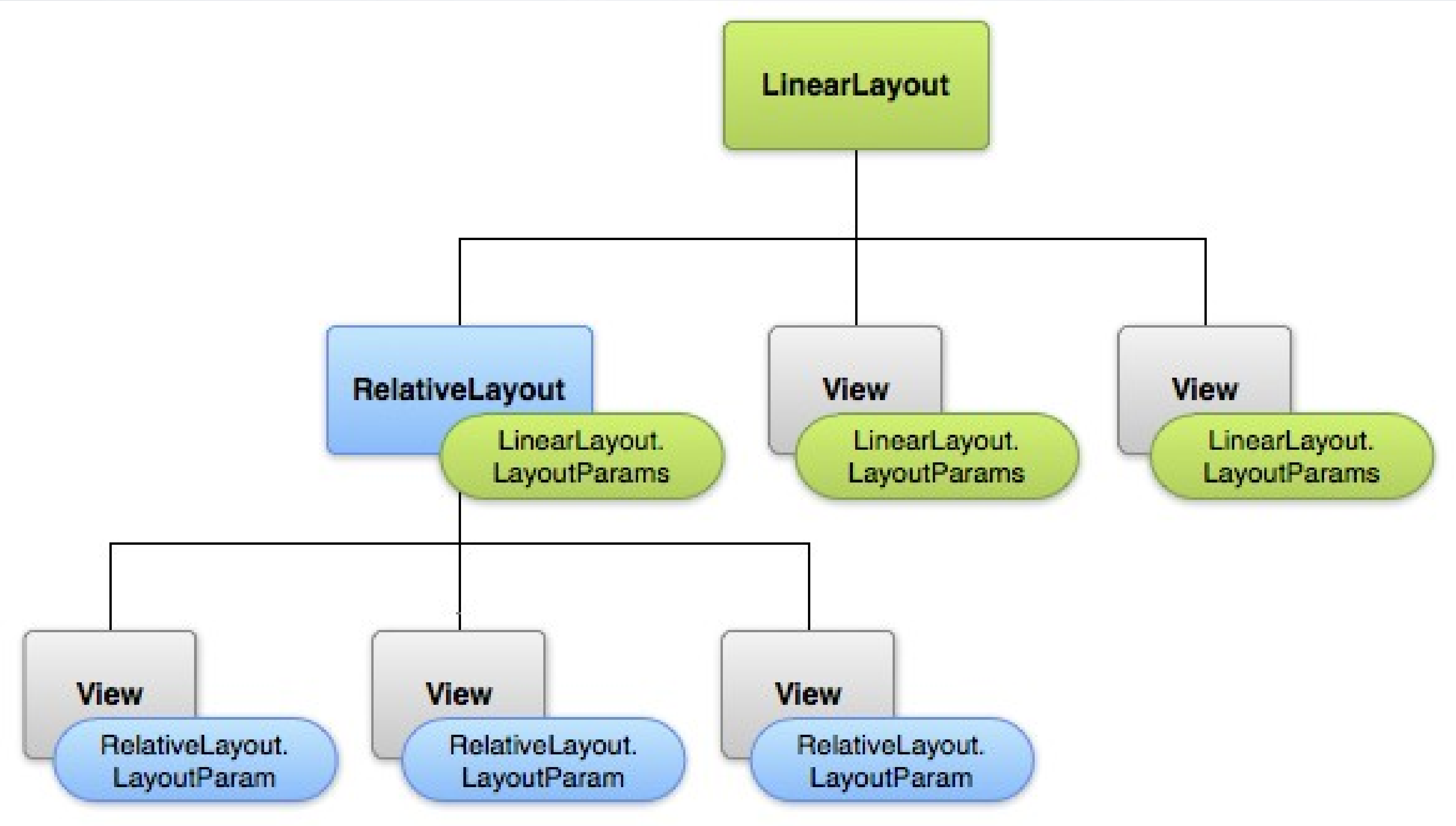
Understanding Views and ViewGroups

How Views and ViewGroups Apply to Activities



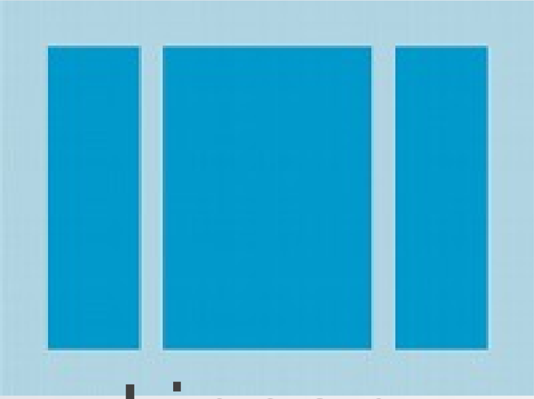
```
<view group>
    <view group>
        <view>
    <view group>
        <view>
        <view>
```

Understanding Views and ViewGroups

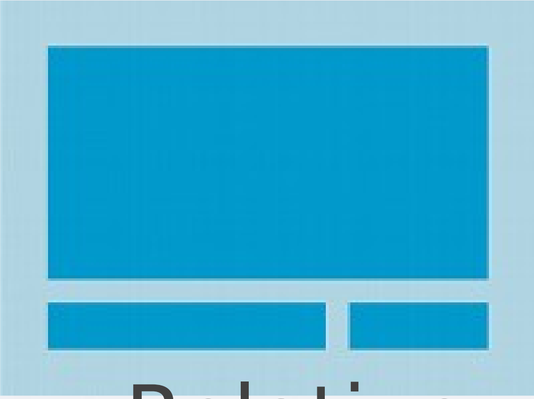

LinearLayout
RelativeLayout
LinearLayout. LayoutParams
View
LinearLayout. LayoutParams
View
LinearLayout. LayoutParams
View
RelativeLayout. LayoutParam
View
RelativeLayout. LayoutParam
View
RelativeLayout. LayoutParam

Understanding Views and ViewGroups

Common Layouts



Linear Layout



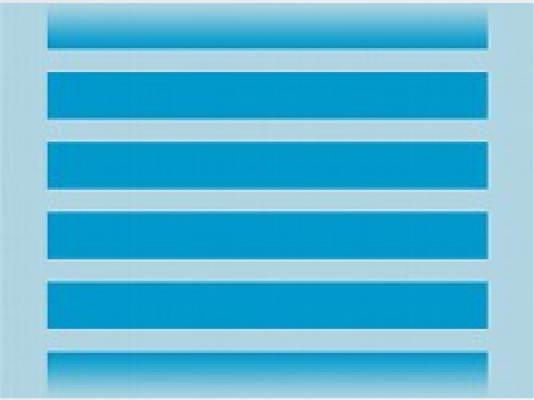
Relative Layout

<html>
&lt;!-- web page --&gt;
</html>

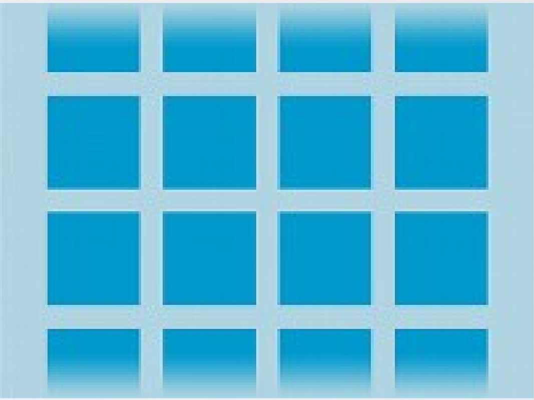
Web View

Understanding Views and ViewGroups

AdapterView

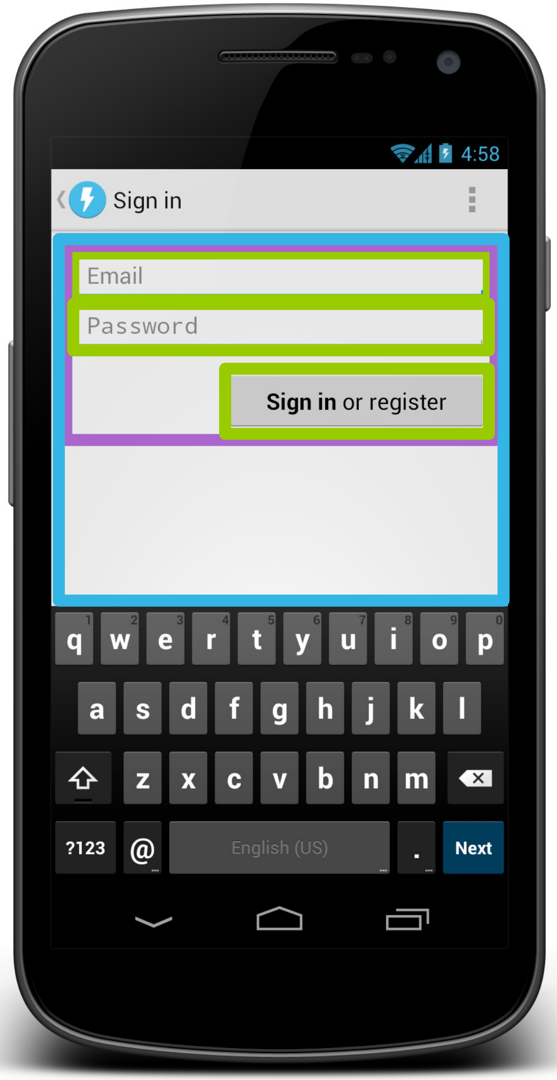


List View



Grid View

The Android user interface XML and resources



```
<ScrollView android:layout_width="match_parent"
   android:layout_height="match_parent">

   <LinearLayout android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:padding="16dp">

      <EditText android:id="@+id/email"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="@string/prompt_email"
         android:inputType="textEmailAddress"
         android:singleLine="true" />

      <EditText android:id="@+id/password"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="@string/prompt_password"
         android:inputType="textPassword”
         android:singleLine="true" />

      <Button android:id="@+id/sign_in_button"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_gravity="right"
         android:layout_marginTop="16dp"
         android:paddingLeft="32dp"
         android:paddingRight="32dp"
         android:text="@string/action_sign_in_register" />

   </LinearLayout>
</ScrollView>
```

The Android user interface XML and resources

App Resources (Review)

res/

drawable	Drawable XML
drawable-xhdpi	PNGs, 9-patch PNGs, optimized for multiple densities
drawable-hdpi	
drawable-mdpi	
layout	Layout XML optimized for physical screen size and orientation
layout-land	
layout-large	
layout-large-land	
values	Strings, styles, themes, etc.
values-v11	Styles, themes varying by API level
values-v14	
values-en	Strings XML localized for your target regions
values-fr	
values-ja	

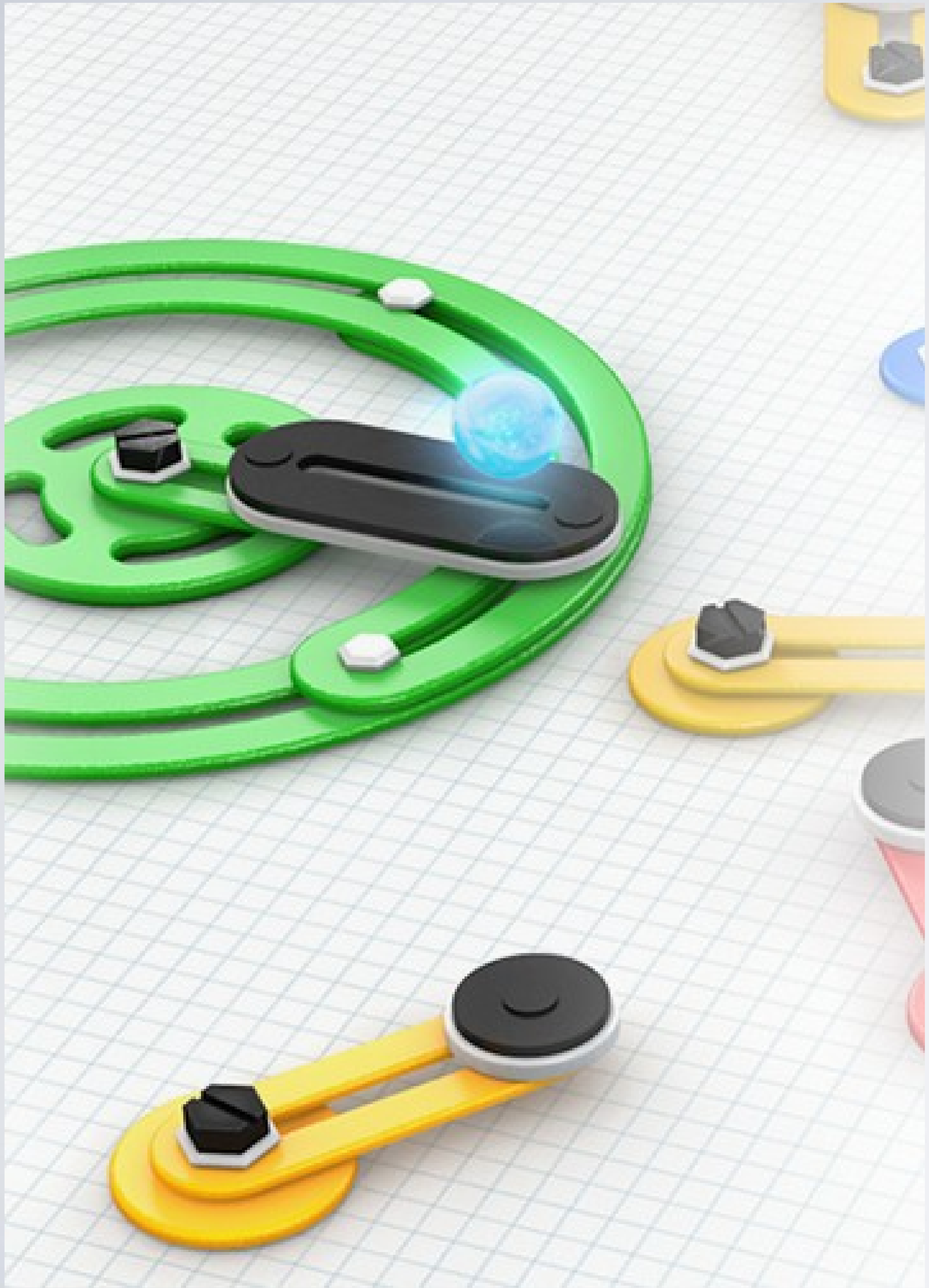
The Android user interface XML and resources

Referencing Resources

Code Review - Breakfast in London

strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">hello</string>
</resources>
```



Understanding Graphical Layout Editor in Android Studio

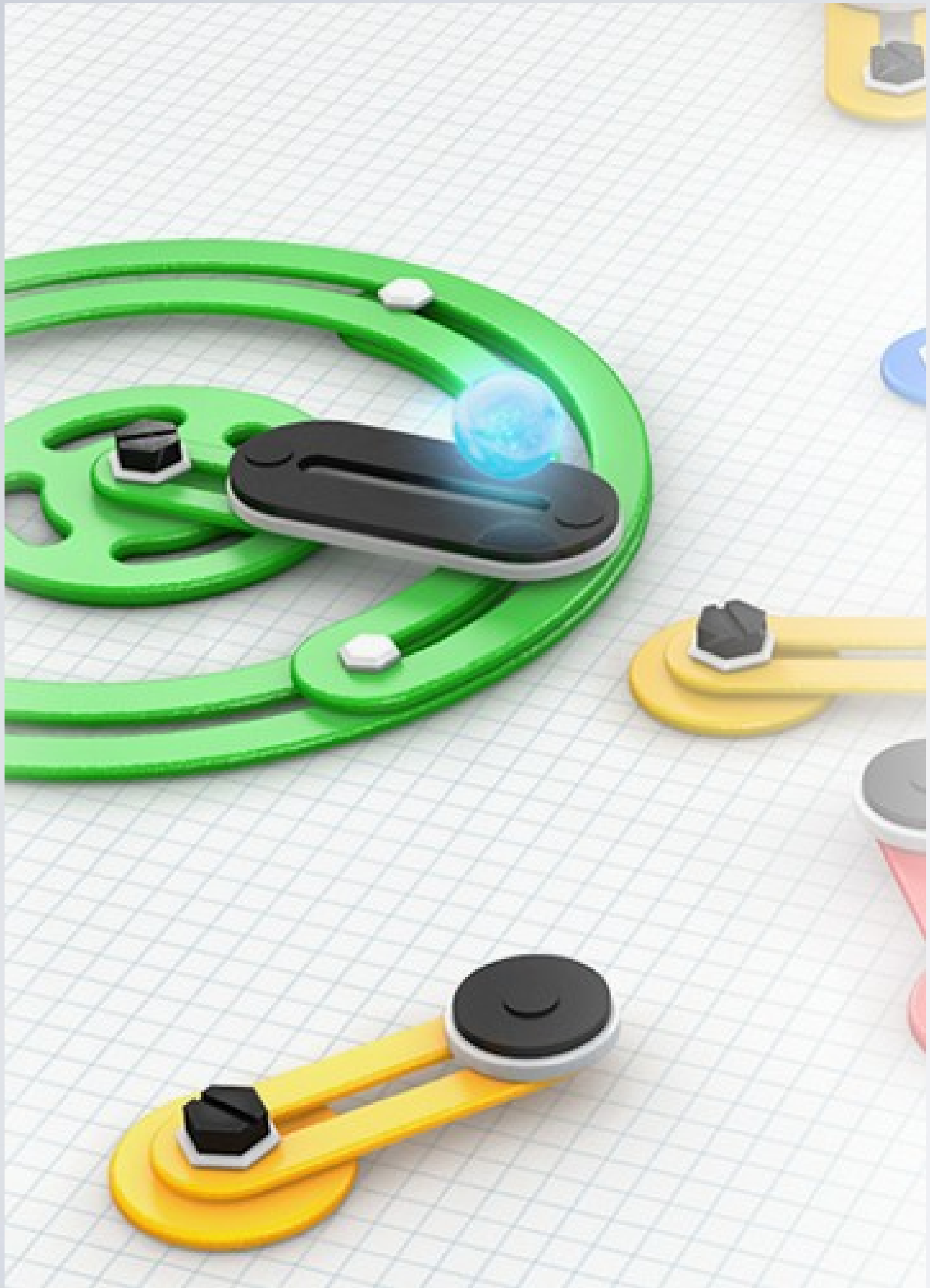
Lab Exercise 3.1

Graphical Layout Editor

Lab Exercise 3.1

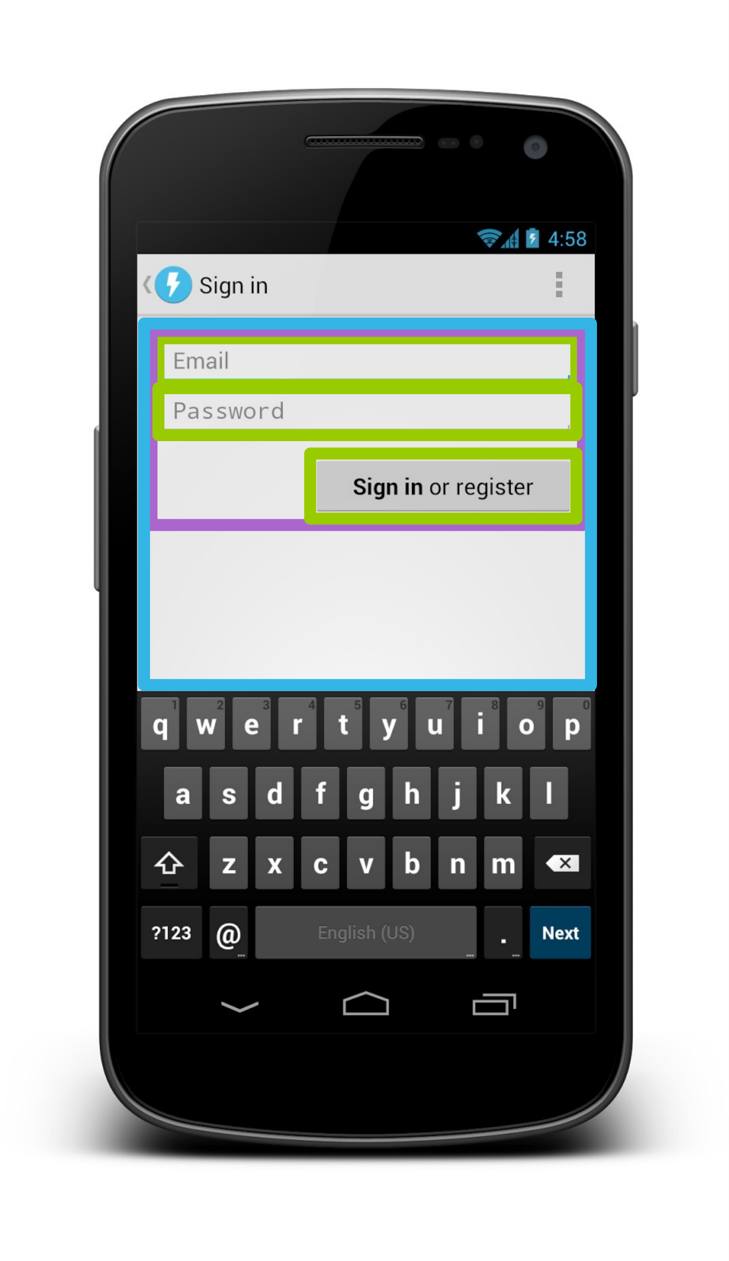
Android Studio - Graphical Layout Editor: Tasks

- Task 1: Install Git > Clone Google I/O App 2013
- Task 2: Navigating the Graphical Layout Editor



Android UI styles, themes, and visual elements

Styling for Android



```
<ScrollView android:layout_width="match_parent"
   android:layout_height="match_parent">

   <LinearLayout android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:orientation="vertical"
      android:padding="16dp">

      <EditText android:id="@+id/email"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="@string/prompt_email"
         android:inputType="textEmailAddress"
         android:singleLine="true" />

      <EditText android:id="@+id/password"
         android:layout_width="match_parent"
         android:layout_height="wrap_content"
         android:hint="@string/prompt_password"
         android:inputType="textPassword”
         android:singleLine="true" />

      <Button android:id="@+id/sign_in_button"
         android:layout_width="wrap_content"
         android:layout_height="wrap_content"
         android:layout_gravity="right"
         android:layout_marginTop="16dp"
         android:paddingLeft="32dp"
         android:paddingRight="32dp"
         android:text="@string/action_sign_in_register" />

   </LinearLayout>
</ScrollView>
```

Styling for Android

Styling in XML

```
<TextView android:layout_width="match_parent"
    android:layout_height="wrap_content" android:padding="4dp" android:text="1" />

<TextView android:layout_width="match_parent"
    android:layout_height="wrap_content" android:padding="4dp" android:text="2" />
```

- OR

-

```
<TextView style="@style/MyText" android:text="1" />
<TextView style="@style/MyText" android:text="2" />
```

+

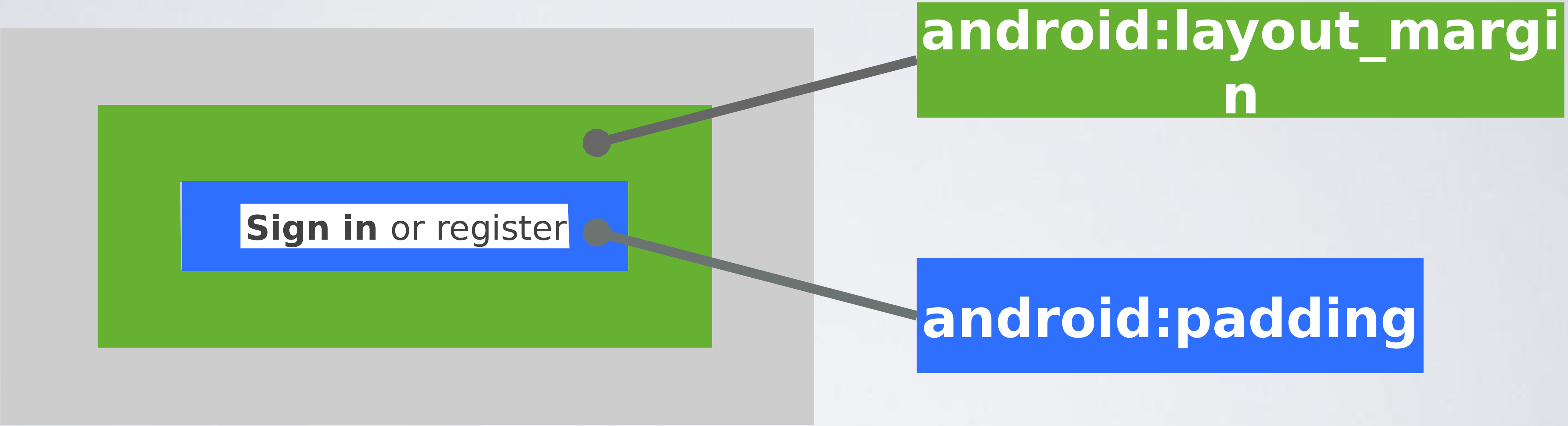
```
<style name="MyText">
    <item name="android:padding">4dp</item>
    <item name="android:layout_width">match_parent</item>
    <item name="android:layout_height">wrap_content</item>
</style>
```

res/values/

styles.xml

Styling for Android

Margins & Padding



Sign in or register

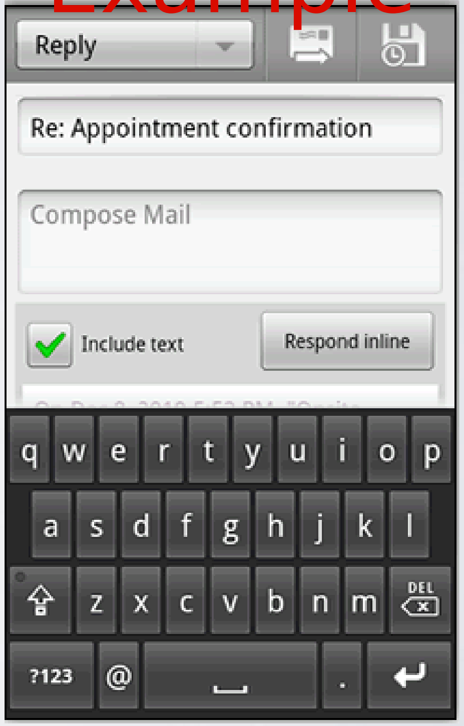
android:layout_margin

android:padding

Overview of Android themes

Holo Visual Language

Gmail Example

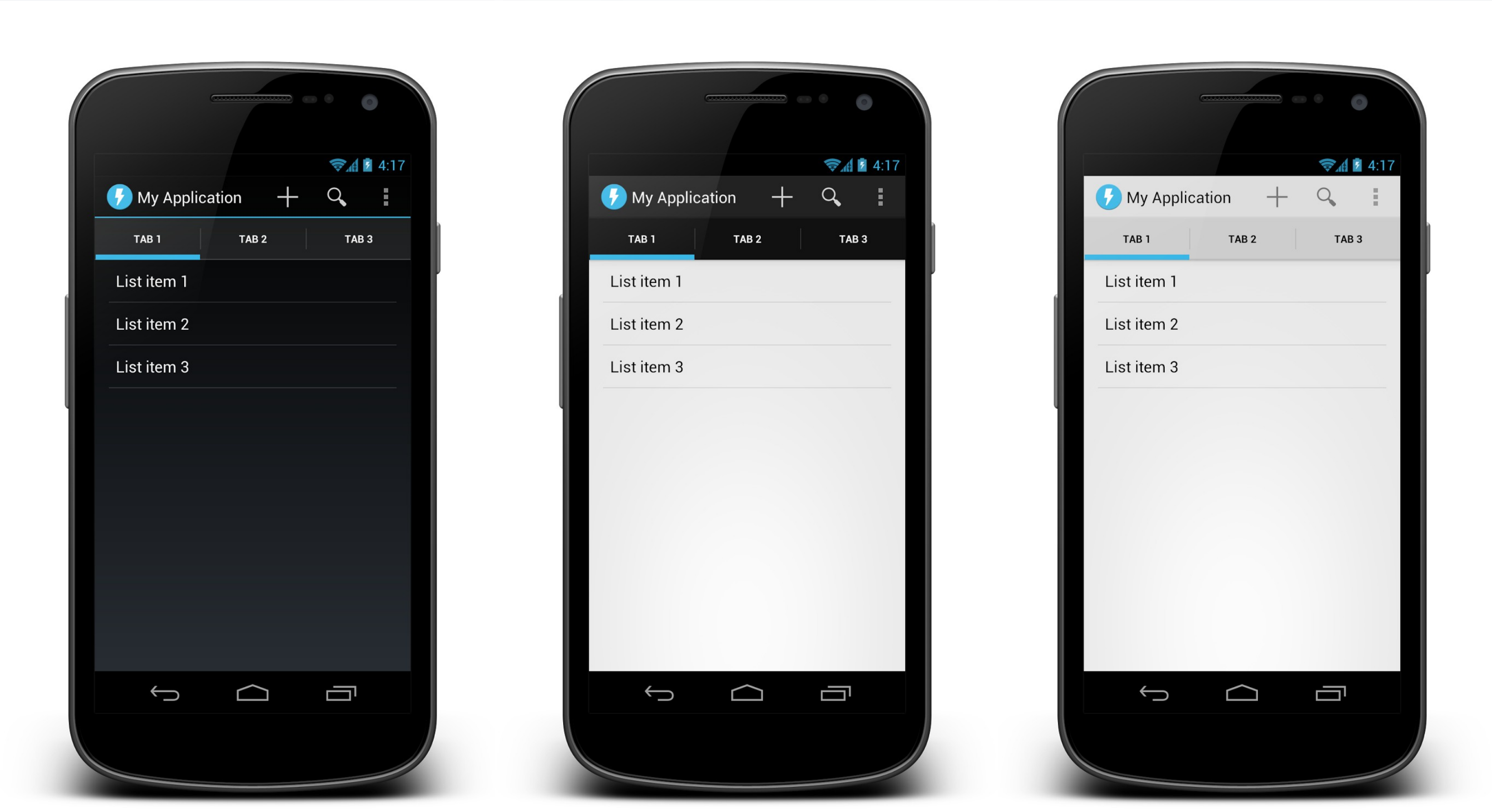

Re: Appointment confirmation

Compose Mail

☑ Include text

Overview of Android themes

Holo Variations



Dark

Dark Action Bar

Light

Overview of Android themes

Applying Themes in XML

```
<application android:theme="@android:style/Theme.Holo">
   ...
</application>
```

```
<style name="MyTheme" parent="@android:style/Theme.Holo">
   ...
</style>
```

Holo @ implementations

Holo Dark - @android:style/Theme.Holo
Holo Dark Action Bar - @android:style/Theme.Holo.Light.DarkActionBar
Holo Light - @android:style/Theme.Holo.Light

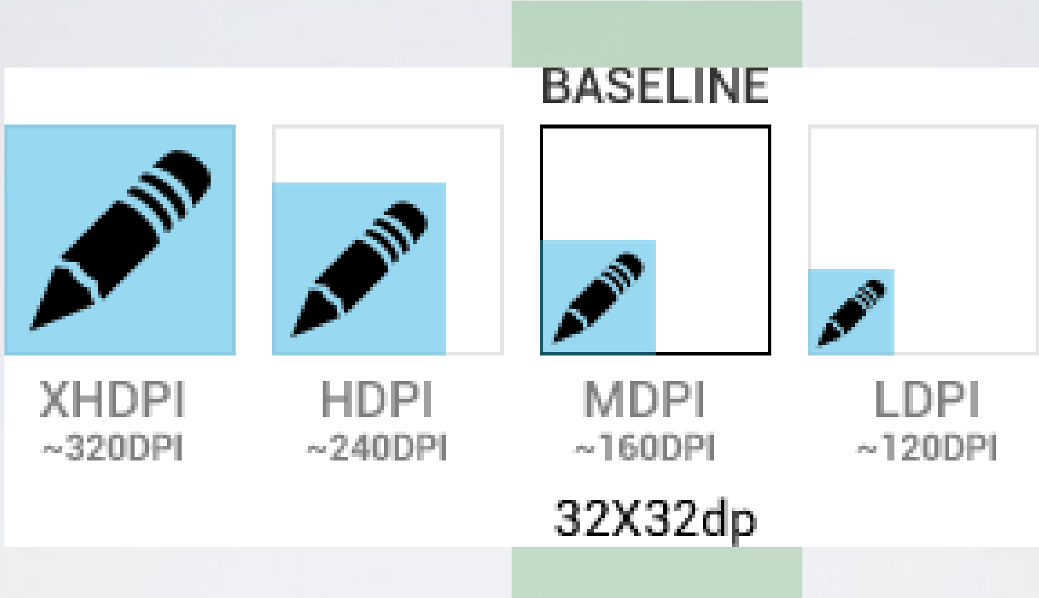
Describing Android layout and visual elements

Defining DIP Units

DIP units keep elements the same physical size across any screen.

Describing Android layout and visual elements

Providing Assets to Support Screen Densities



Icons and other PNG files should generally be provided for multiple densities.

Describing Android layout and visual elements

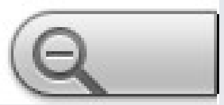
Key Drawable Types

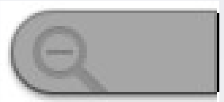
- Bitmaps (.png)
- State Lists (.xml)
- 9-patches (.9.png)

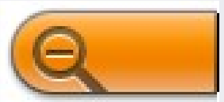
Describing Android layout and visual elements

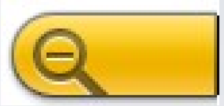
State List Drawables

drawable-mdpi/

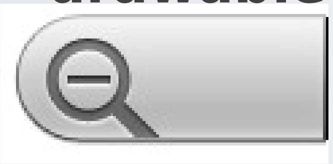

foo_default.png

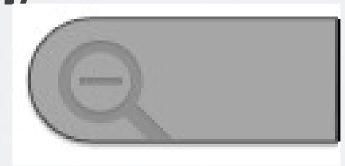

foo_disabled.png

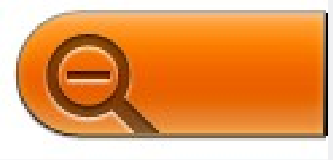

foo_focused.png

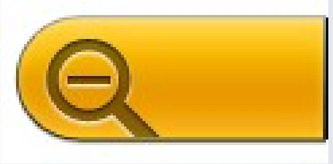

foo_pressed.png

drawable-hdpi/


foo_default.png

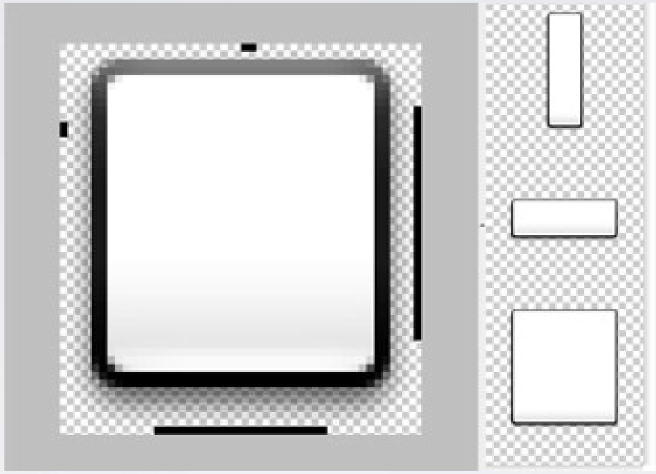

foo_disabled.png


foo_focused.png


foo_pressed.png

Describing Android layout and visual elements

Understanding 9-Patches – foo.9.png

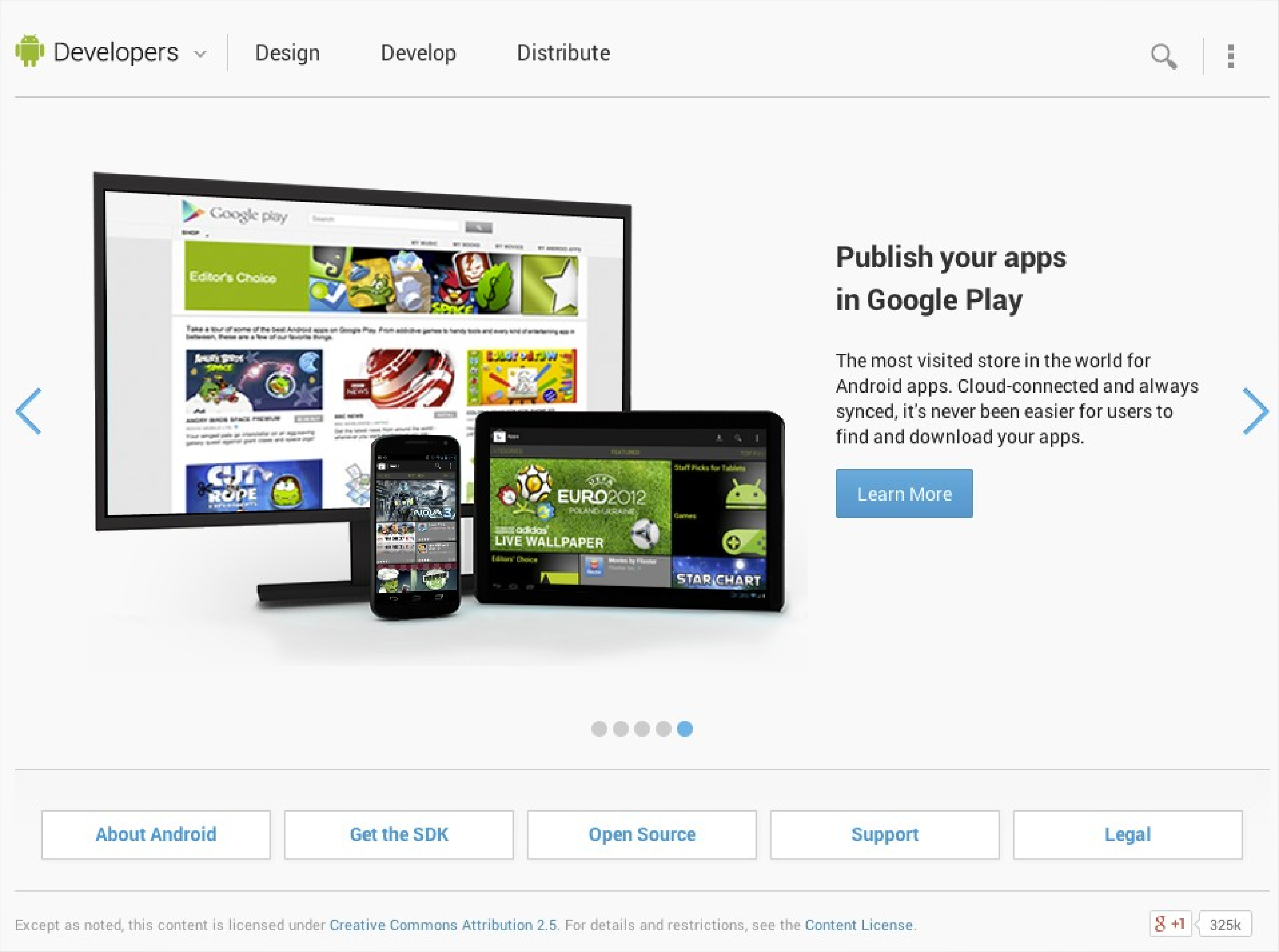


```
<selector>
   <item android:drawable="@drawable/foo_disabled"
        android:state_enabled="false" ... />
   <item android:drawable="@drawable/foo_pressed"
        android:state_pressed="true" ... />
   <item android:drawable="@drawable/foo_focused"
        android:state_focused="true" ... />
   <item android:drawable="@drawable/foo_default" />
</selector>
```

drawable/

foo.xml

More Resources



Publish your apps in Google Play

The most visited store in the world for Android apps. Cloud-connected and always synced, it's never been easier for users to find and download your apps.

Except as noted, this content is licensed under Creative Commons Attribution 2.5. For details and restrictions, see the Content License.

Give us feedback!

Fill out this survey!

http://goo.gl/B3pJeR

Thank You & QA

+Your Name
your_email@your_company.com
your_title, company
@your_twitter
your_URL