

# CS 528 Mobile and Ubiquitous Computing

## HW 0 Slides: Introduction to Android

---

**Emmanuel Agu**





# Tutorials 1-5 Overview



# Setting up your Project

- **Tutorials 1 & 2 (Optional):** Installing Java, Android Studio on your own machine
- **Tutorial 3:** Setting up your project
  - How to set up a new Android Project, add new Activity (App screen)
- **Tutorial 4:** Running a Simple App
  - How to select, run app on a virtual device (AVD)
- **Tutorial 5:** Tour of Android Studio Interface
  - Intro to Android Studio menus, toolbars and Drag-and-drop widget palette



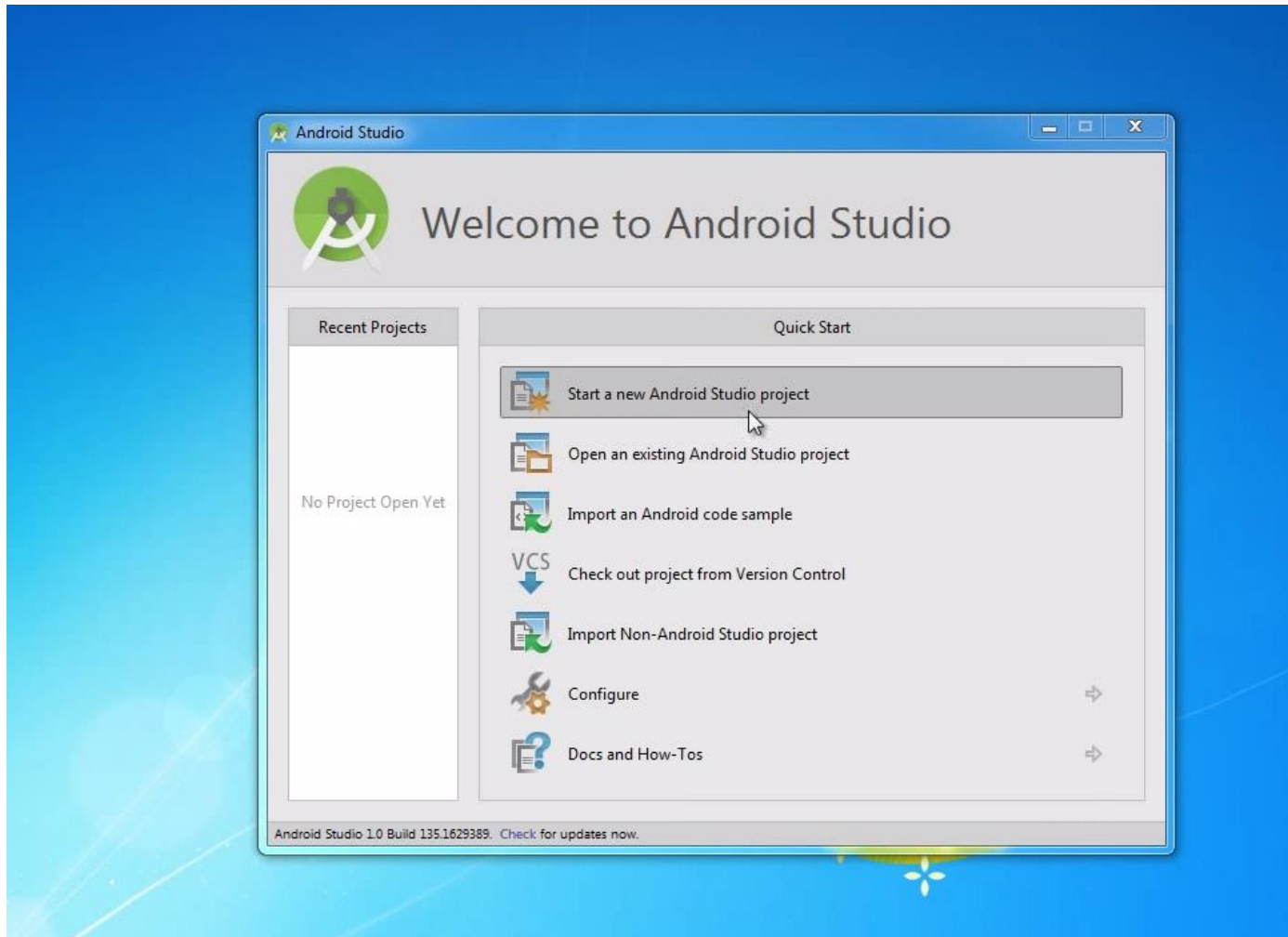
# Tutorial 3



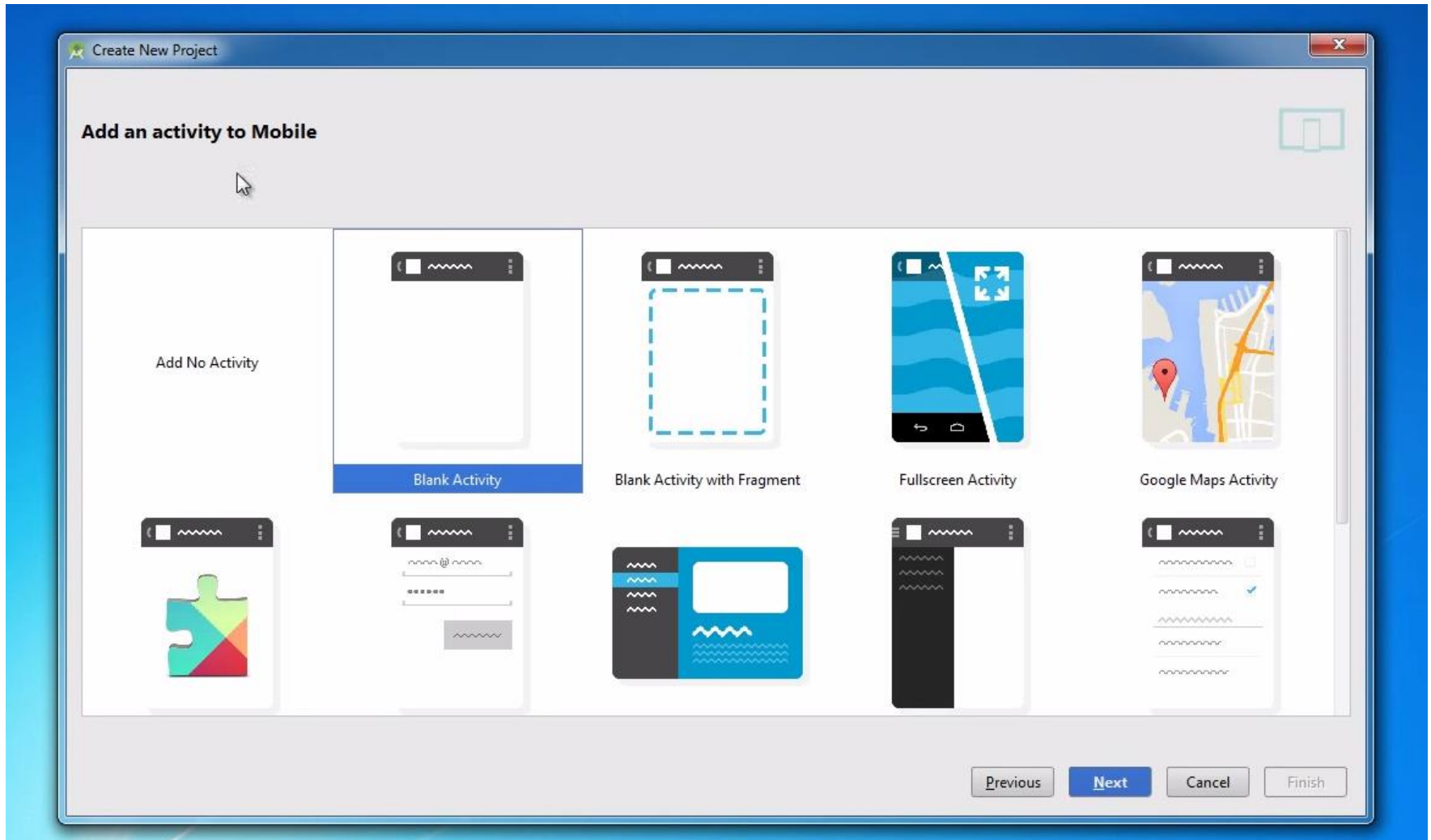
# Setting up your Project

- **Tutorial 3: Android App Development for Beginners - 3**
  - Setting up your project by Bucky Roberts (thenewboston)
    - <https://www.youtube.com/watch?v=r4olez0sfvY>
- Main steps to set up Android Project
  - Start a new Android Project
  - Configure new Android Project (select app name, domain name, etc)
  - Set platform and minimum SDK
  - Add an Activity

# Start a new Android Project



# Add an Activity (Blank Activity is Simplest)





# Tutorial 4

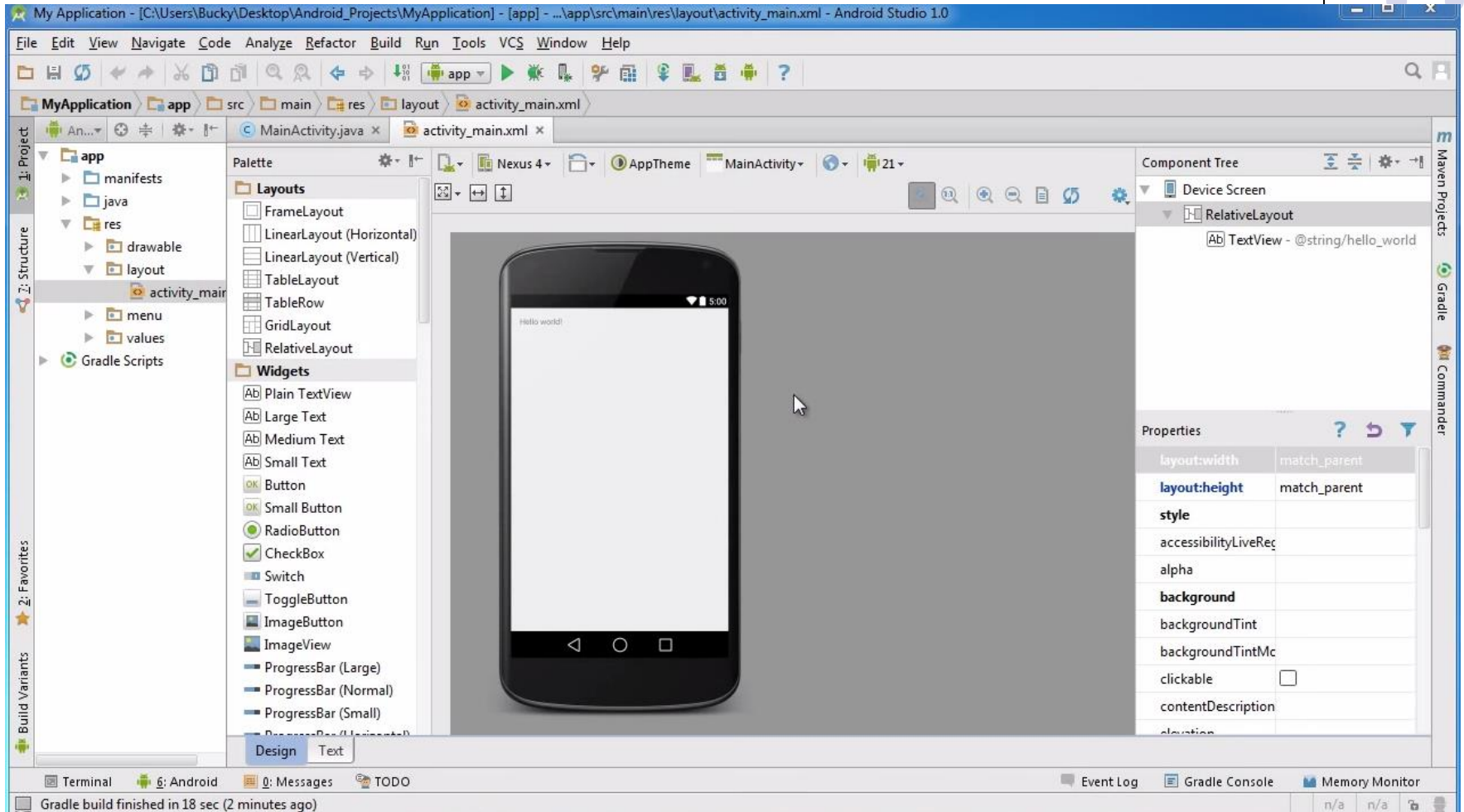
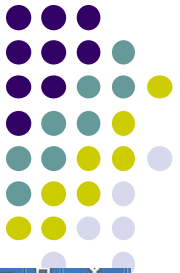




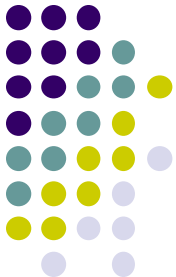
# Running a Simple App

- Tutorial 4: Android App Development for Beginners - 4 – Running a Simple App [10:48 mins] by Bucky Roberts
  - <https://www.youtube.com/watch?v=qKRWC3Q8wRw>
- Main steps
  - Run Android Studio
  - Fix any remaining issues
  - Run AVD, select virtual device
  - Run App on selected virtual device

# Open Android Studio



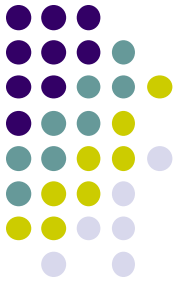
# Run AVD Manager



The screenshot shows the AVD Manager window in Android Studio. The window title is "AVD Manager" and it features a green header with the text "Your Virtual Devices" and the Android Studio logo. Below the header is a table listing virtual devices. At the bottom of the window, there is a "Create Virtual Device..." button and "OK" and "Cancel" buttons.

Type	Name	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Buckys Phone	1080 × 1920: xxhdpi	21	Android 5.0.1	arm	1 GB	
	Nexus 5 API 21 x86	1080 × 1920: xxhdpi	21	Google APIs	x86	1 GB	

# How to Run the App?



Click here to run the app

The screenshot displays the Android Studio interface. On the left, the Project view shows the file structure for 'OMGAndroid', with 'activity\_my.xml' selected in the 'layout' folder. The central editor shows the XML code for 'activity\_my.xml':

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingRight="64dp"
    android:paddingLeft="64dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp"
    tools:context=".MainActivity">

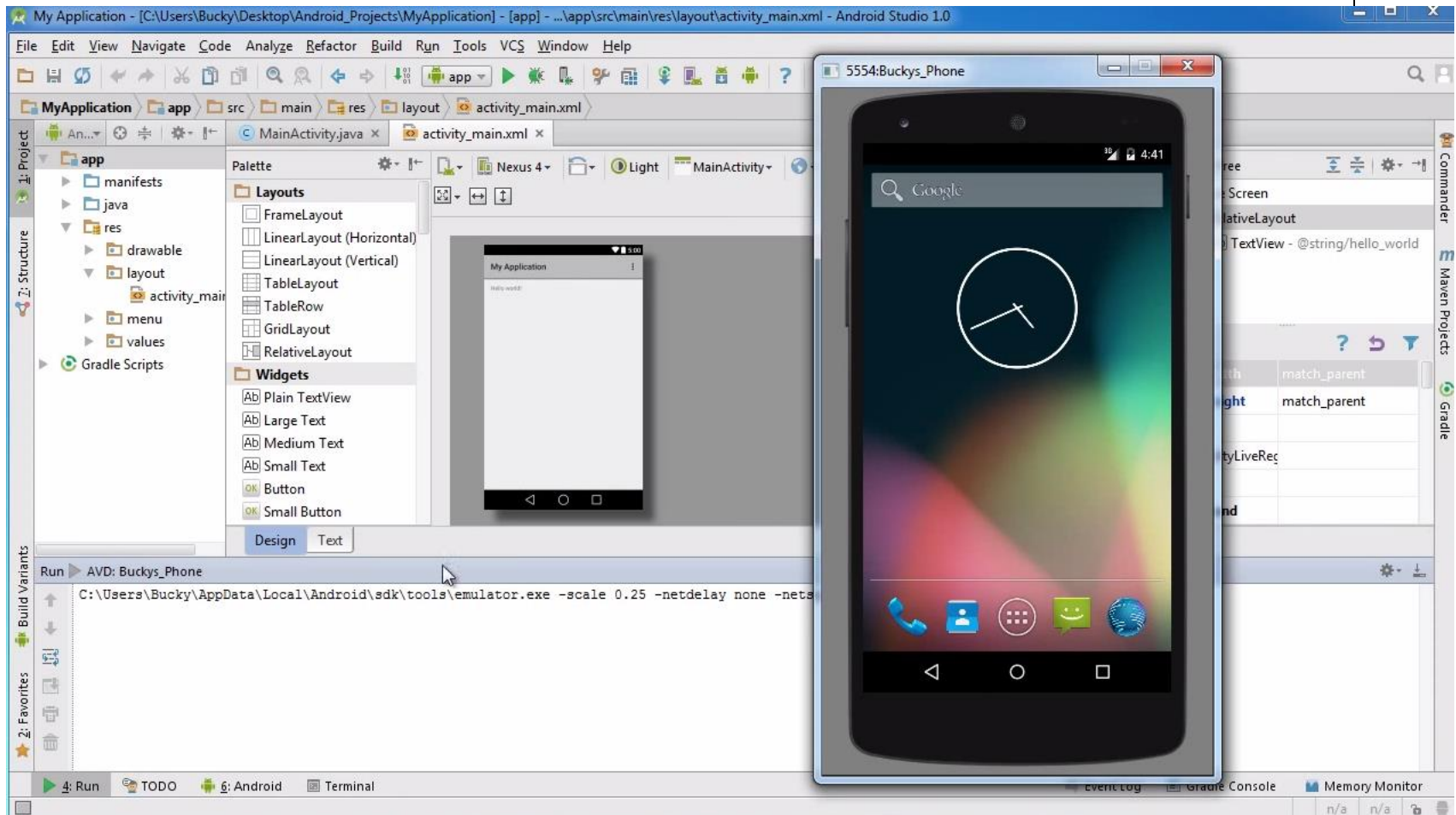
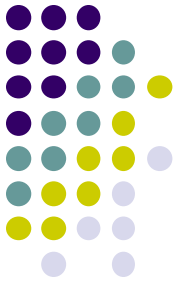
    <TextView
        android:text="Hello world!"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

On the right, the Preview window shows a Nexus 4 device with the app running. The app's title bar is 'OMG Android' and the main content area displays 'Hello world!'. A red circle and arrow point to the 'Run' button (a green play icon) in the top toolbar.

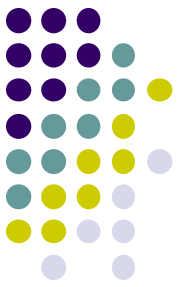
At the bottom of the IDE, a status bar indicates 'Gradle build finished in 13 sec (3 minutes ago)'. The system tray shows '1:1 LF UTF-8' and other icons.

# Run App on Virtual Device (Phone)





# Tutorial 5



# Tour of Android Studio Interface

- Tutorial 5: Tour of Android Studio Interface [6:01 mins]
  - <https://www.youtube.com/watch?v=-pdTqBq2TFQ>
- Quick overview of main sections of Android Studio
  - Windows menu bar
  - Android tool bar
  - Project window
  - Editor Window
  - Palette for Drag-and-Drop Design of Android buttons
- More detailed coverage of specific UI aspects later

# Typical Windows Menu Bar (File, edit, etc)



File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Layouts

- FrameLayout
- LinearLayout (Horizontal)
- LinearLayout (Vertical)
- TableLayout
- TableRow
- GridLayout
- RelativeLayout

Widgets

- Plain TextView
- Large Text
- Medium Text
- Small Text
- Button
- Small Button
- RadioButton
- CheckBox
- Switch
- ToggleButton
- ImageButton
- ImageView
- ProgressBar (Large)
- ProgressBar (Normal)
- ProgressBar (Small)

Component Tree

- Device Screen
  - RelativeLayout
    - TextView - @string/hello\_world

Properties

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchMode	<input type="checkbox"/>

Run AVD: Nexus\_5\_API21\_x86

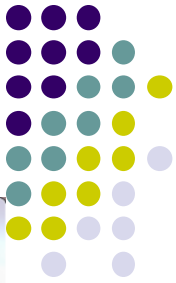
```
Inode size: 256
Journal blocks: 1024
Label:
Blocks: 16896
Block groups: 1
Reserved block group size: 7
Created filesystem with 11/4224 inodes and 1302/16896 blocks
```

Event Log | Gradle Console | Memory Monitor

21:1 n/a n/a



# Tool Bar: Shortcuts to Frequently used Android-specific Functions (E.g. One-click access to SDK manager)



The screenshot displays the Android Studio 1.0.1 IDE. A red box highlights the toolbar, which contains icons for file operations (New, Open, Save, Copy, Paste), navigation (Back, Forward), and development (Run, Stop, Refresh, etc.). The main workspace shows a 'Hello world' app running on a Nexus 5 virtual device. The interface includes a Project view on the left, a Palette of layouts and widgets in the center, a Component Tree on the right, and a Properties panel at the bottom right. The Run/Log panel at the bottom shows the output of the application.

Property	Value
layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveReg	
alpha	
background	
backgroundTint	
backgroundTintMoc	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchM	<input type="checkbox"/>

## Path to Current File in IDE Window (Clickable)

The screenshot displays the Android Studio IDE interface. A red box highlights the breadcrumb path in the top toolbar: **Helloworld** > **app** > **src** > **main** > **res** > **layout** > **activity\_main.xml**. A red arrow points from the text above to this breadcrumb path. The main editor shows a preview of the app with the text "Hello world!". The left sidebar shows the Project view with the file structure. The right sidebar shows the Component Tree and Properties panels.

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Helloworld > app > src > main > res > layout > activity\_main.xml

Project

- Android
  - app
    - manifests
    - java
    - res
      - drawable
      - layout
        - activity\_main.xml
      - menu
      - values
    - Gradle Scripts

Palette

- Layouts
  - FrameLayout
  - LinearLayout (Horizontal)
  - LinearLayout (Vertical)
  - TableLayout
  - TableRow
  - GridLayout
  - RelativeLayout
- Widgets
  - Plain TextView
  - Large Text
  - Medium Text
  - Small Text
  - Button
  - Small Button
  - RadioButton
  - CheckBox
  - Switch
  - ToggleButton
  - ImageButton
  - ImageView
  - ProgressBar (Large)
  - ProgressBar (Normal)
  - ProgressBar (Small)

Component Tree

- Device Screen
  - RelativeLayout
    - TextView - @string/hello\_world

Properties

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
alpha	
background	
backgroundTint	
backgroundTintMode	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchMode	<input type="checkbox"/>

Run ▶ AVD: Nexus\_5\_API\_21\_x86

```
Created filesystem with 11/4224 inodes and 1302/16896 blocks
```

Event Log | Gradle Console | Memory Monitor

21:1 n/a n/a

# Editor Window (Allows editing of current file we are working on)



The screenshot displays the Android Studio IDE interface. The main window is titled "Hello world - [C:\temp\android\_studio\_projects\Helloworld] - [app] - ...\app\src\main\res\layout\activity\_main.xml - Android Studio 1.0.1". The interface is divided into several panels:

- Project Structure:** Shows the project hierarchy with folders like manifests, java, res, and layout. The file `activity_main.xml` is selected.
- Palette:** Contains a list of UI components under "Layouts" (e.g., FrameLayout, LinearLayout) and "Widgets" (e.g., TextView, Button).
- Editor Window:** The central area showing the design view of the app. It displays a mobile screen with the text "Hello world" and "Hello world!". A red box highlights this design view, and a red arrow points to the `activity_main.xml` tab above it.
- Component Tree:** Shows the hierarchy of UI components, including `RelativeLayout` and `TextView`.
- Properties:** A table showing the properties of the selected `TextView` widget.
- Run Console:** Shows the output of the application run, including system logs.

Property	Value
<code>layout:width</code>	<code>match_parent</code>
<code>layout:height</code>	<code>match_parent</code>
<b>style</b>	
<code>accessibilityLiveReg</code>	
<code>alpha</code>	
<b>background</b>	
<code>backgroundTint</code>	
<code>backgroundTintMoc</code>	
<code>clickable</code>	<input type="checkbox"/>
<code>contentDescription</code>	
<code>elevation</code>	
<code>focusable</code>	<input type="checkbox"/>
<code>focusableInTouchM</code>	<input type="checkbox"/>

# Clicking on Editor Window Tabs switches between project files



The screenshot shows the Android Studio IDE interface. The top menu bar includes File, Edit, View, Navigate, Code, Analyze, Refactor, Build, Run, Tools, VCS, Window, and Help. The toolbar contains various icons for file operations and development. The breadcrumb navigation shows the path: Helloworld > app > src > main > java > com > example > emmanuel > helloworld > MainActivity. The editor window has two tabs: MainActivity.java (active) and activity\_main.xml. The code in the editor is as follows:

```
package com.example.emmanuel.helloworld;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

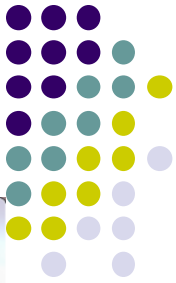
        return super.onOptionsItemSelected(item);
    }
}
```

The bottom of the IDE shows the Run tab with the following output:

```
Run ▶ AVD: Nexus_5_API_21_x86
Inode size: 256
Journal blocks: 1024
Label:
Blocks: 16896
Block groups: 1
Reserved block group size: 7
Created filesystem with 11/4224 inodes and 1302/16896 blocks
```

The status bar at the bottom indicates the current encoding is 1:1 CRLF + UTF-8 + [icon].

# Project Window (Shows project files, packages, etc)



The screenshot displays the Android Studio IDE with the following components:

- Project Window (Left):** A tree view showing the project structure. A red box highlights the 'app' folder, which contains subfolders for 'manifests', 'java', 'res' (with subfolders 'drawable', 'layout', 'menu', 'values'), and 'Gradle Scripts'. A red arrow points from the text above to the 'app' folder.
- Palette (Middle-Left):** A list of UI components categorized into 'Layouts' (FrameLayout, LinearLayout, etc.) and 'Widgets' (TextView, Button, etc.).
- Design View (Center):** A visual representation of the app's main screen, showing a white background with the text 'Hello world!' and an Android logo.
- Component Tree (Middle-Right):** A hierarchical view of the UI components, showing a 'RelativeLayout' containing a 'TextView' with the text '@string/hello\_world'.
- Properties Panel (Bottom-Right):** A table of properties for the selected component. The visible properties are:

Property	Value
layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveReg	
alpha	
background	
backgroundTint	
backgroundTintMoc	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchM	<input type="checkbox"/>

At the bottom of the screen, the 'Run' console shows the output of the AVD creation process, including details about the filesystem and storage allocation.

# Palette of Drag-and-Drop Elements for Designing Interface (Layout, widgets, etc)



The screenshot displays the Android Studio IDE with the following components:

- Project Structure:** Shows the project hierarchy for 'Hello world', including 'app', 'src', 'main', 'res', and 'layout'.
- Palette:** A red box highlights the 'Layouts' and 'Widgets' sections. A red arrow points from the title above to this palette.
- Design View:** Shows a mobile device (Nexus 5) displaying the 'Hello world' app.
- Component Tree:** Shows the hierarchy of the app's UI components, including 'RelativeLayout' and 'TextView'.
- Properties Panel:** Shows the properties of the selected 'TextView' component, such as 'layout:width', 'layout:height', and 'style'.

Property	Value
layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveReg	
alpha	
background	
backgroundTint	
backgroundTintMoc	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchM	<input type="checkbox"/>

# Parameters of Drag-and-Drop Elements for Designing Interface (e.g. colors, dimensions of widgets, etc)



The screenshot shows the Android Studio IDE with the design view of an activity. The central canvas displays a mobile device screen with the text "Hello world!". The left sidebar shows the Project structure, and the bottom shows the Run console output.

The Properties panel for the selected TextView widget is highlighted with a red box. The parameters listed are:

layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveReg	
alpha	
background	
backgroundTint	
backgroundTintMoc	
clickable	<input type="checkbox"/>
contentDescription	
elevation	
focusable	<input type="checkbox"/>
focusableInTouchM	<input type="checkbox"/>



# Tutorial 8





# Basic Overview of an App

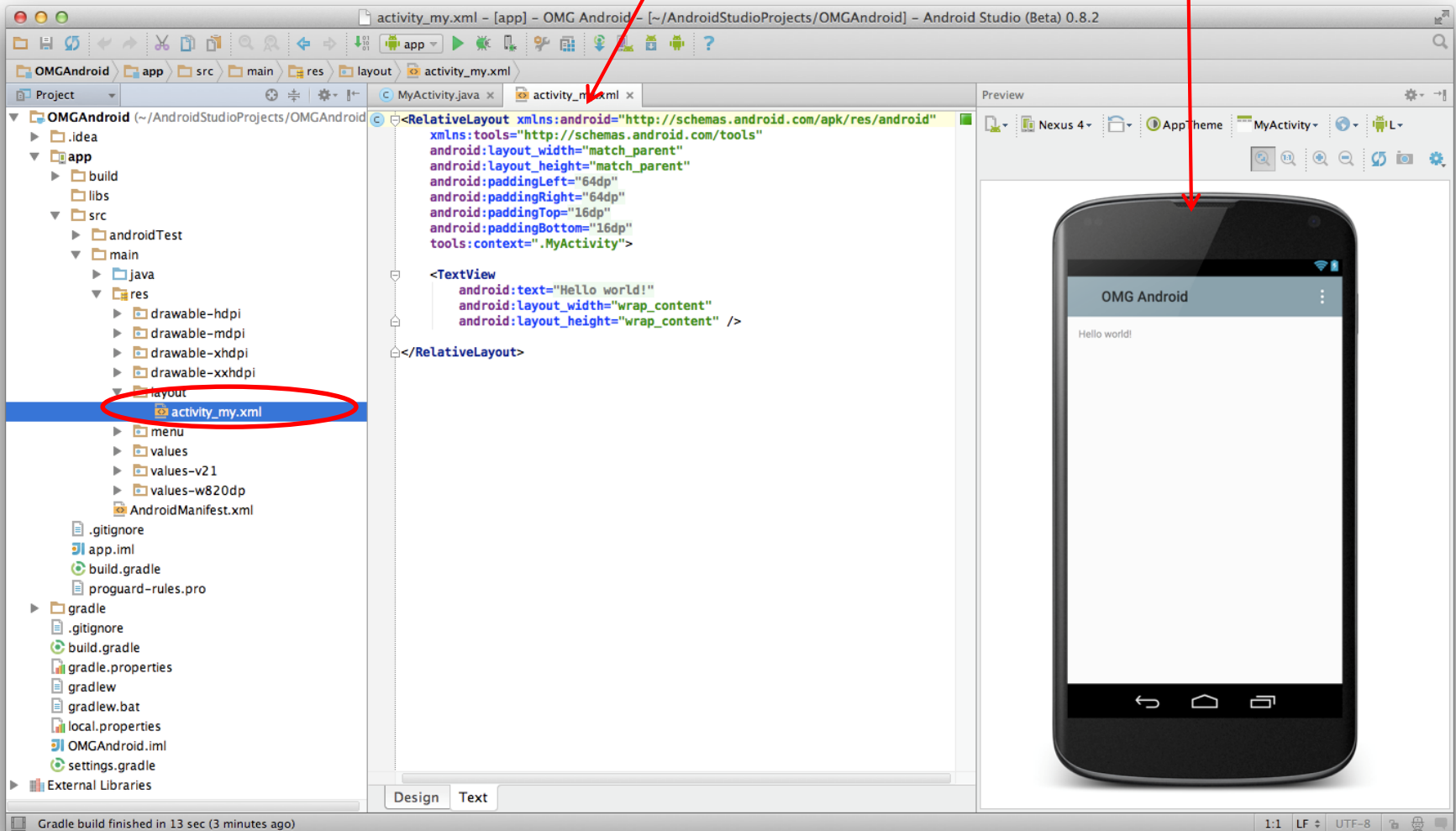
- Tutorial 8: Basic Overview of an App [11:36 mins]
  - <https://www.youtube.com/watch?v=9l1fWAIHPg>
  
- Main topics
  - Introduces main files of Android App
    - Activity\_main.xml
    - MainActivity.java
    - AndroidManifest.xml
  - How to work with these files within Android Studio
  - Editing files using either drag-and-drop interface or XML
  - Flow of basic app

# Editing Android

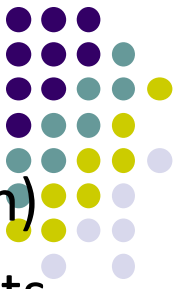
Activity\_my.xml  
(can edit directly)

App running on  
Emulator (can edit  
Text, drag and drop)

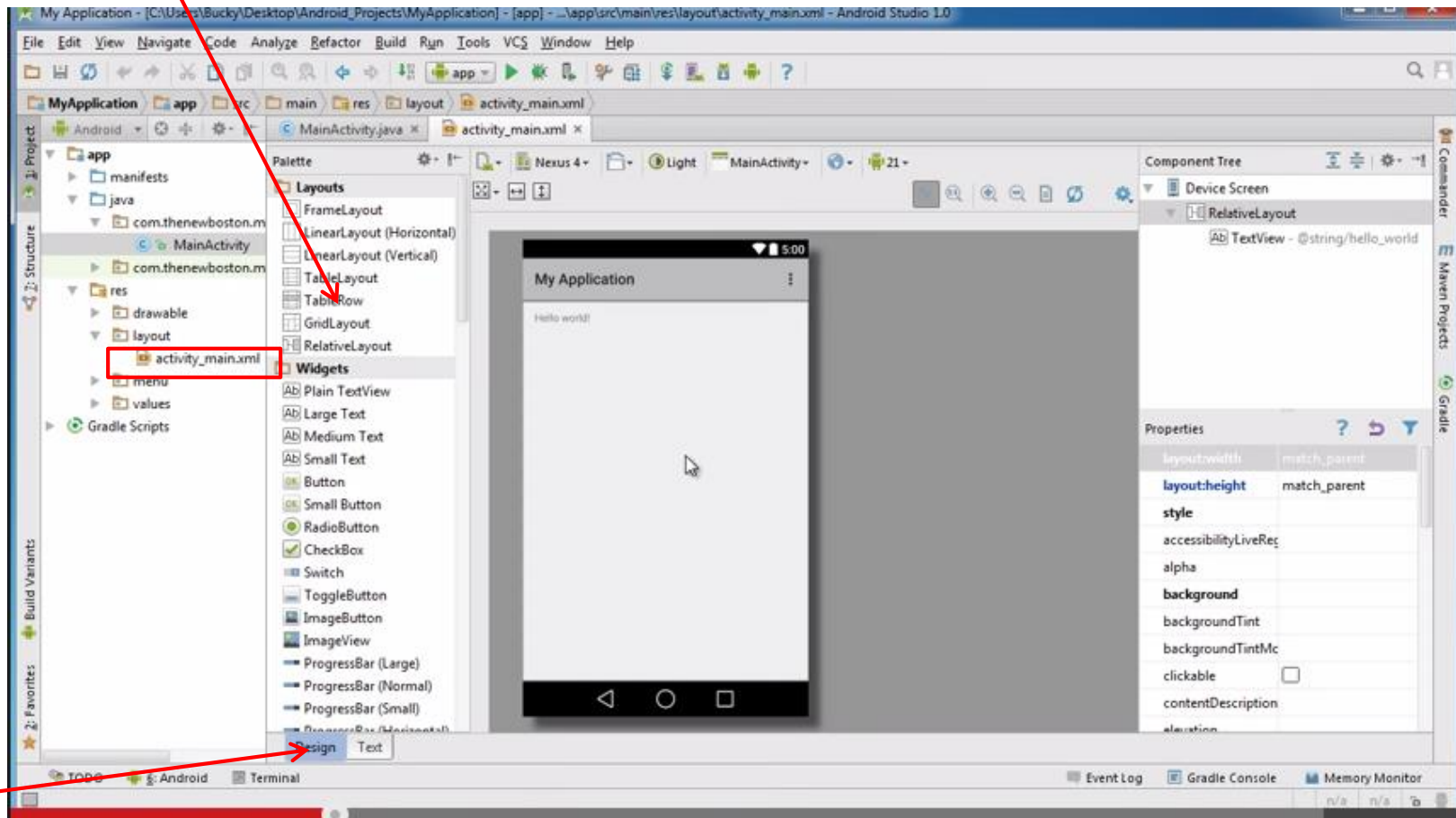
- Activity\_my.xml is XML file specifying screen layout, widgets
- Can edit XML directly or drag and drop



# Activity\_main.xml



- **Widgets:** elements that can be dragged onto activity (screen)
- **Design View:** Design app screen using Drag-and-drop widgets

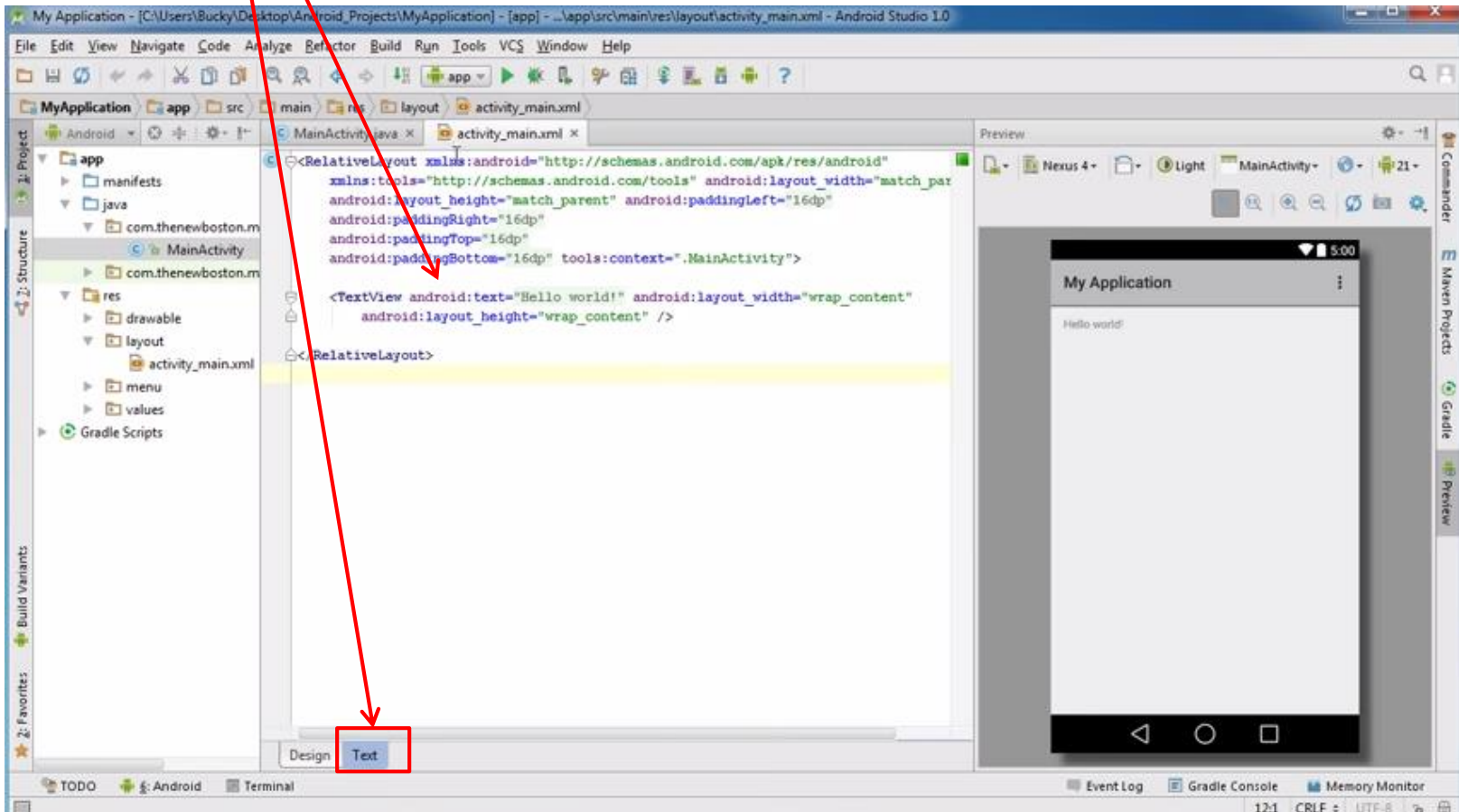


Design view

# Activity\_main.xml: Text View



- **Text view:** Design screen by editing XML file directly
- **Note:** dragging and dropping widgets auto-generates corresponding XML





# MainActivity.java

- Java code, defines actions, handles interaction/put taken (intelligence)
  - E.g. What app will do when button/screen clicked

```
package com.thenewboston.myapplication;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

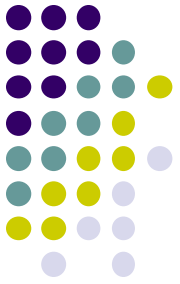
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
    }
}
```

# AndroidManifest.xml

- App's starting point (a bit like main( ) in C)

A screenshot of the Android Studio IDE. The main editor window displays the content of the AndroidManifest.xml file. The file is highlighted in yellow. The code defines an application and a main activity. The activity is named MainActivity and has a label "My Application". It includes an intent filter for the action android.intent.action.MAIN and the category android.intent.category.LAUNCHER. The package name is com.thenewboston.myapplication. The IDE interface includes a menu bar, a toolbar, a project structure view on the left, and a status bar at the bottom.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.thenewboston.myapplication" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="My Application"
        android:theme="@style/AppTheme" >

        <activity
            android:name=".MainActivity"
            android:label="My Application" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

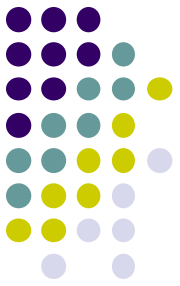
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```



# Android UI Tour

# Home Screen

- First screen, includes **favorites** tray (e.g phone, mail, messaging, web, etc)

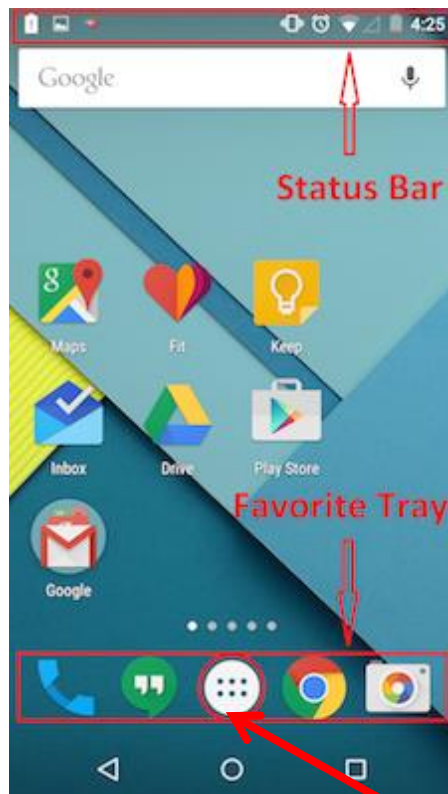






# All Apps Screen

- Accessed by touching **all apps button** in favorites tray
- Can swipe through multiple app screens, customizable



Android 5.0

all apps button

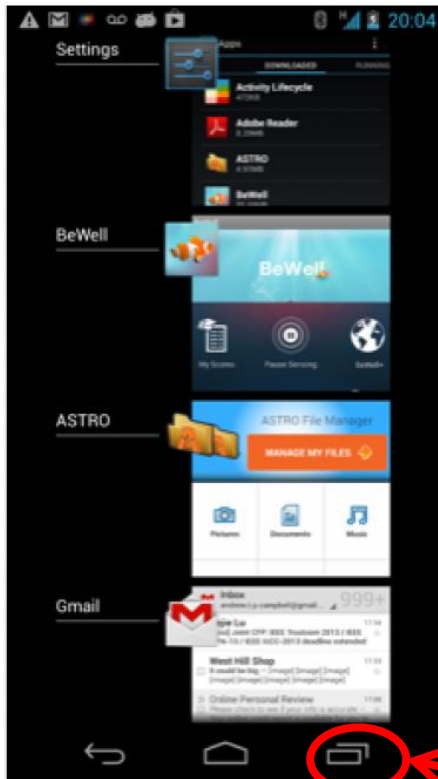


Android 5.0

# Recent Apps Screen



- Accessed by touching **recent apps button**
- Shows recently used apps, touch app to switch to it



recent apps button



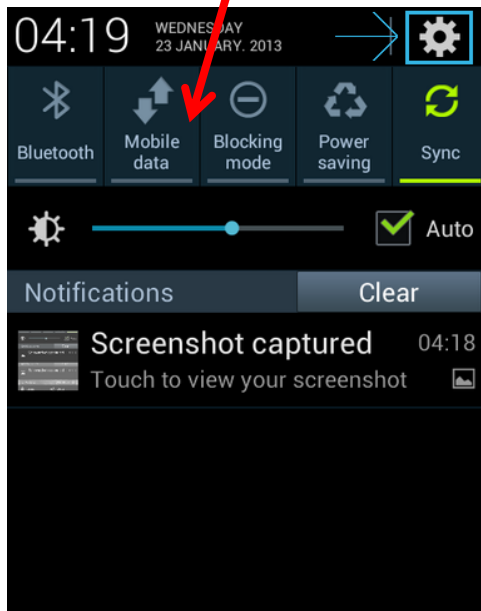
Android 5.0

# Status Bar and Notification Screen

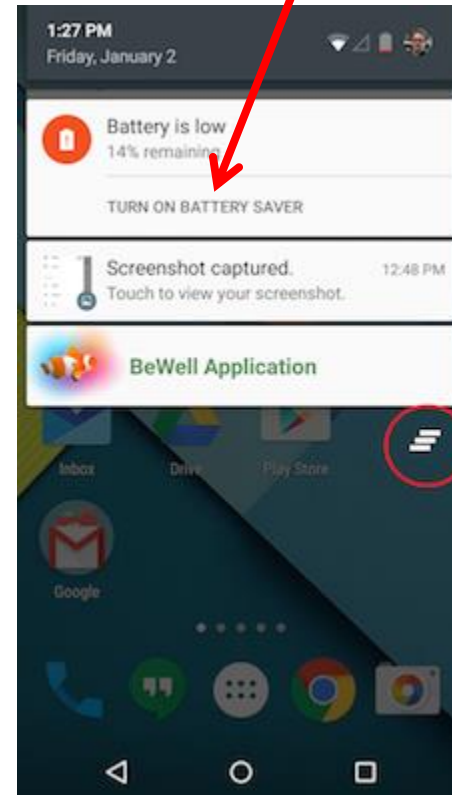


- **Status:** time, battery, cell signal strength, bluetooth enabled, etc
- **Notification:** wifi, mail, bewell, voicemail, usb active, music, etc

Status bar



Notification Screen





# References

- Android App Development for Beginners videos by Bucky Roberts (thenewboston)
- Ask A Dev, Android Wear: What Developers Need to Know, <https://www.youtube.com/watch?v=zTS2NZpLyQg>
- Ask A Dev, Mobile Minute: What to (Android) Wear, [https://www.youtube.com/watch?v=n5Yjzn3b\\_aQ](https://www.youtube.com/watch?v=n5Yjzn3b_aQ)
- Busy Coder's guide to Android version 4.4
- CS 65/165 slides, Dartmouth College, Spring 2014
- CS 371M slides, U of Texas Austin, Spring 2014