

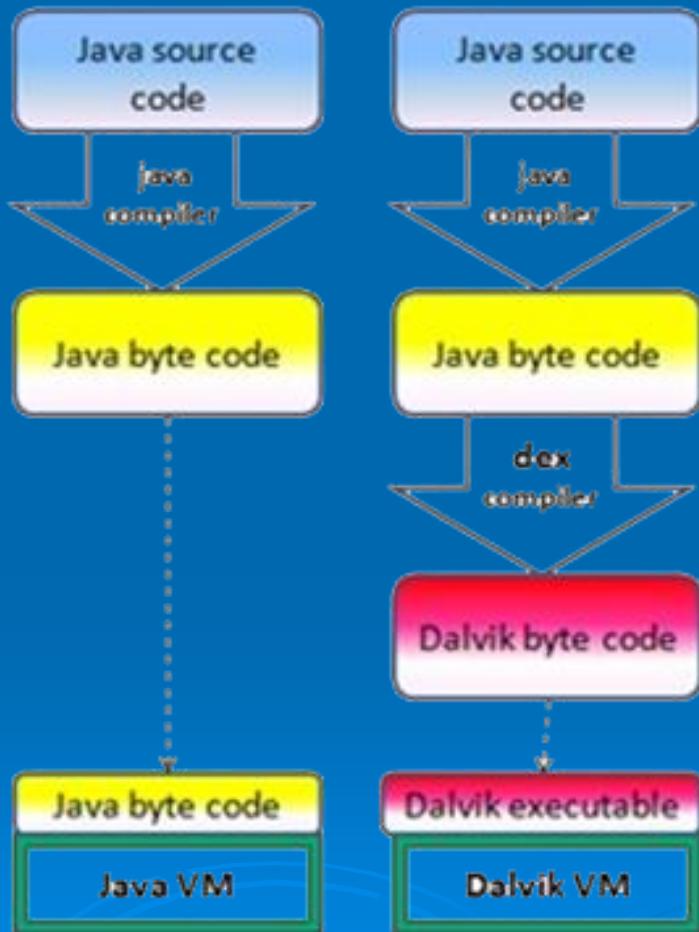
COMP08076

Programming Native App Interaction

- Module coordinator: John Nixon
 - john.nixon@uws.ac.uk, room E259, x3617
- Lecture 2
 - A bit of Android background
 - Components, UI, widgets etc.
 - Layout

A bit more on Android

- Dalvik Java virtual machine
 - A JVM optimised for limited capacity environments
 - .dex (Dalvik Executable) file format (smaller than equivalent multiple .class files)
 - .apk package format is more compact than a compressed .jar file
- Dalvik is host for all Android applications
 - Will not run standard Java .class files
 - Compile multiple class files into single .dex file
- Android Runtime (ART)
 - Replaces Dalvik partially Android 4.4 and completely in Android 5.0



Android Architecture

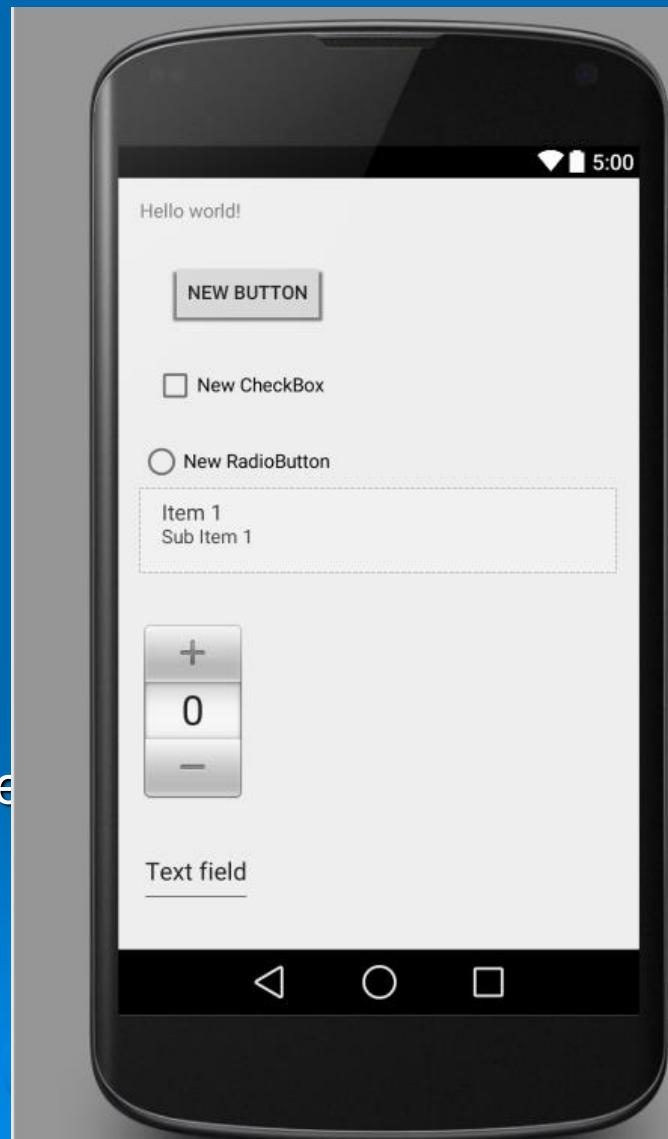


<http://www.edureka.in/blog/beginners-guide-android-architecture/>

- Assignment 1 requires the design and implementation of a user interface to meet a given specification using components into which data can be entered and selections made resulting in appropriate actions taken.
- Decide on UI components required
 - Know range of components (useXML)
- Lay UI components out as a user interface
 - Layout, containers (use XML again)
- “wire up” UI components so they work
- Process input
- Generate output

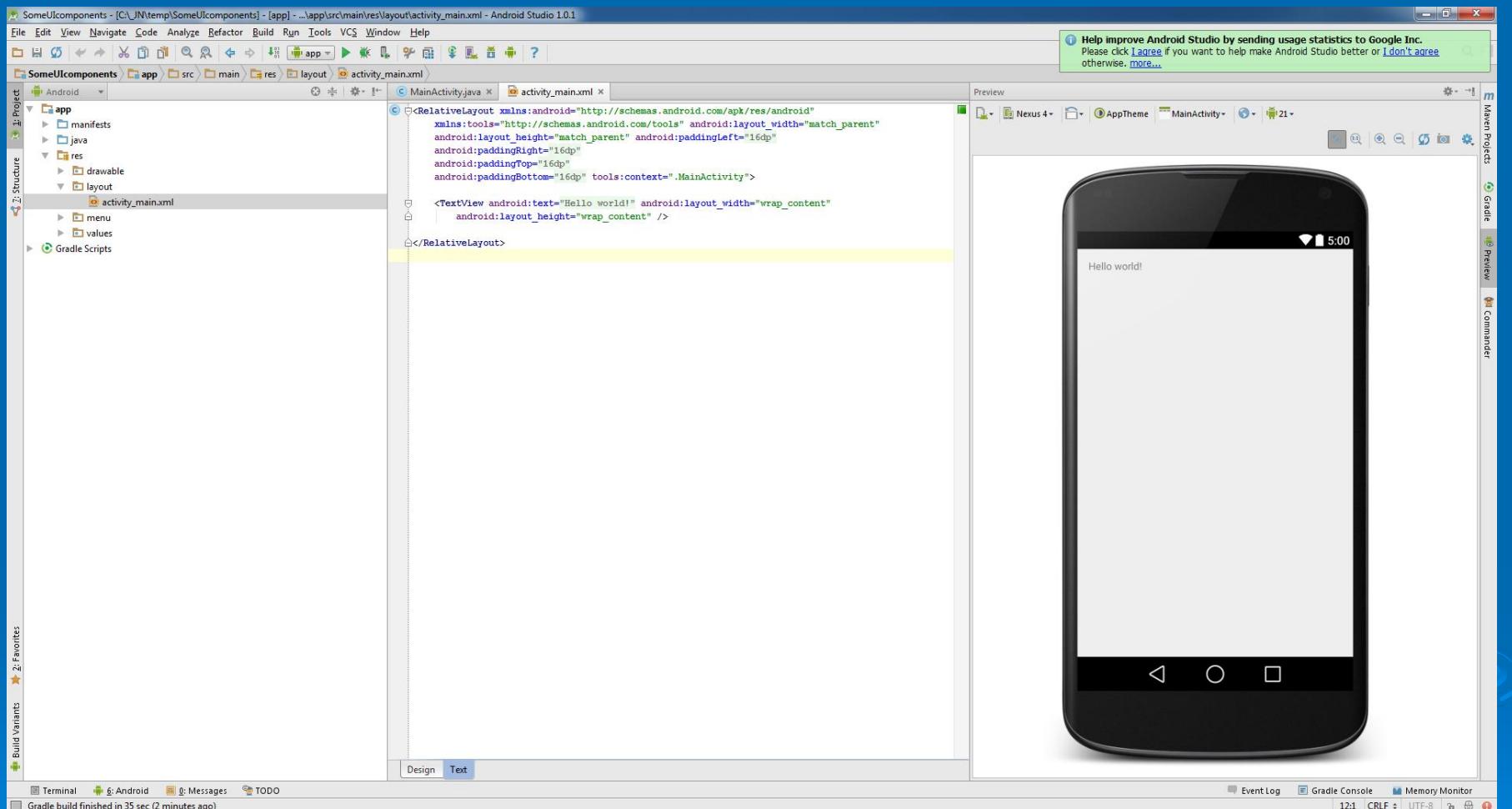
UI Components or Widgets

- TextView – displays text on screen
- Button
- Checkbox
- Radio button – group
- Drop down menu
 - spinner
- Number picker
- Text fields – EditText
 - Many input types, e.g. password, number
- etc.



- To allow user to enter name?
- To let user say whether they wish to bring a pet?
- To allow user to enter email address?
- To allow user to enter number of vouchers required?
- To allow user to select cabin style?

Layout file – activity_main.xml



➤ Text

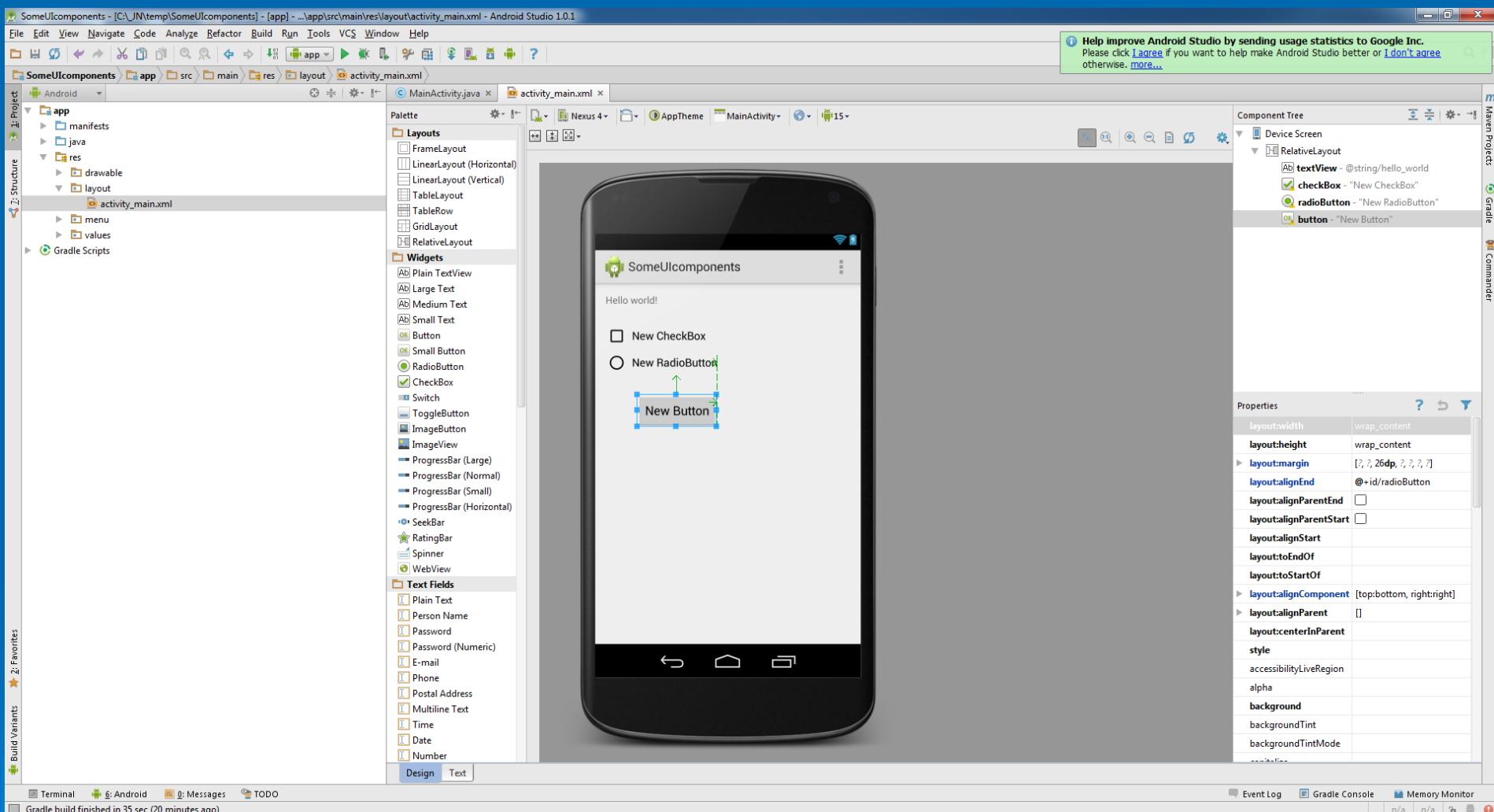
Layout file – activity_main.xml

- <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:paddingLeft="@dimen/activity_horizontal_margin"
 android:paddingRight="@dimen/activity_horizontal_margin"
 android:paddingTop="@dimen/activity_vertical_margin"
 android:paddingBottom="@dimen/activity_vertical_margin"
 tools:context=".MainActivity">

- <TextView android:text="@string/hello_world"
 android:layout_width="wrap_content"
 android:layout_height="wrap_content" />

- </RelativeLayout>

Layout file – activity_main.xml



➤ Design

Layout file – activity_main.xml

SomeUIcomponents - [C:\JN\temp\SomeUIcomponents] - [app] - ...app\src\main\res\layout\activity_main.xml - Android Studio 1.0.1

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

SomeUIcomponents app src main res layout activity_main.xml

MainActivity.java x activity_main.xml x

Help improve Android Studio by sending usage statistics to Google Inc.
Please click Agree if you want to help make Android Studio better or I don't agree otherwise. more...

Project

Android

app

manifests

java

res

drawable

layout

activity_main.xml

menu

values

Gradle Scripts

Structure

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)
- ProgressBar (Horizontal)
- SeekBar
- RatingBar
- Spinner
- WebView

Text Fields

- Plain Text
- Person Name
- Password
- Password (Numeric)
- E-mail
- Phone
- Postal Address
- Multiline Text
- Time
- Date
- Number

Device Screen

Nexus 4

AppTheme

MainActivity

15+

Component Tree

RelativeLayout

- textView - @+id/hello_world
- checkbox - "New CheckBox"
- radioButton - "New RadioButton"
- button - "New Button"

Properties

layoutwidth wrap_content

layoutheight wrap_content

layoutmargin [2, 26dp, 2, 2, 2]

layoutalignEnd @+id/radioButton

layoutalignParentEnd

layoutalignParentStart

layoutalignStart

layouttoEndOf

layouttoStartOf

layoutalignComponent [top:bottom, right:right]

layoutalignParent []

layoutcenterInParent

style

accessibilityLiveRegion

alpha

background

backgroundTint

backgroundTintMode

Code

activity_main.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:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New CheckBox"
        android:id="@+id/checkBox"
        android:layout_marginTop="24dp"
        android:layout_below="@+id/textView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New RadioButton"
        android:id="@+id/radioButton"
        android:layout_below="@+id/checkBox"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button"
        android:layout_below="@+id/radioButton"
        android:layout_alignRight="@+id/radioButton"
        android:layout_alignEnd="@+id/radioButton"
        android:layout_marginTop="26dp" />
```

Design Text

Terminal

Android Messages

Messages

TODO

Gradle build finished in 35 sec (20 minutes ago)

n/a n/a

Event Log Gradle Console Memory Monitor

Design

```
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

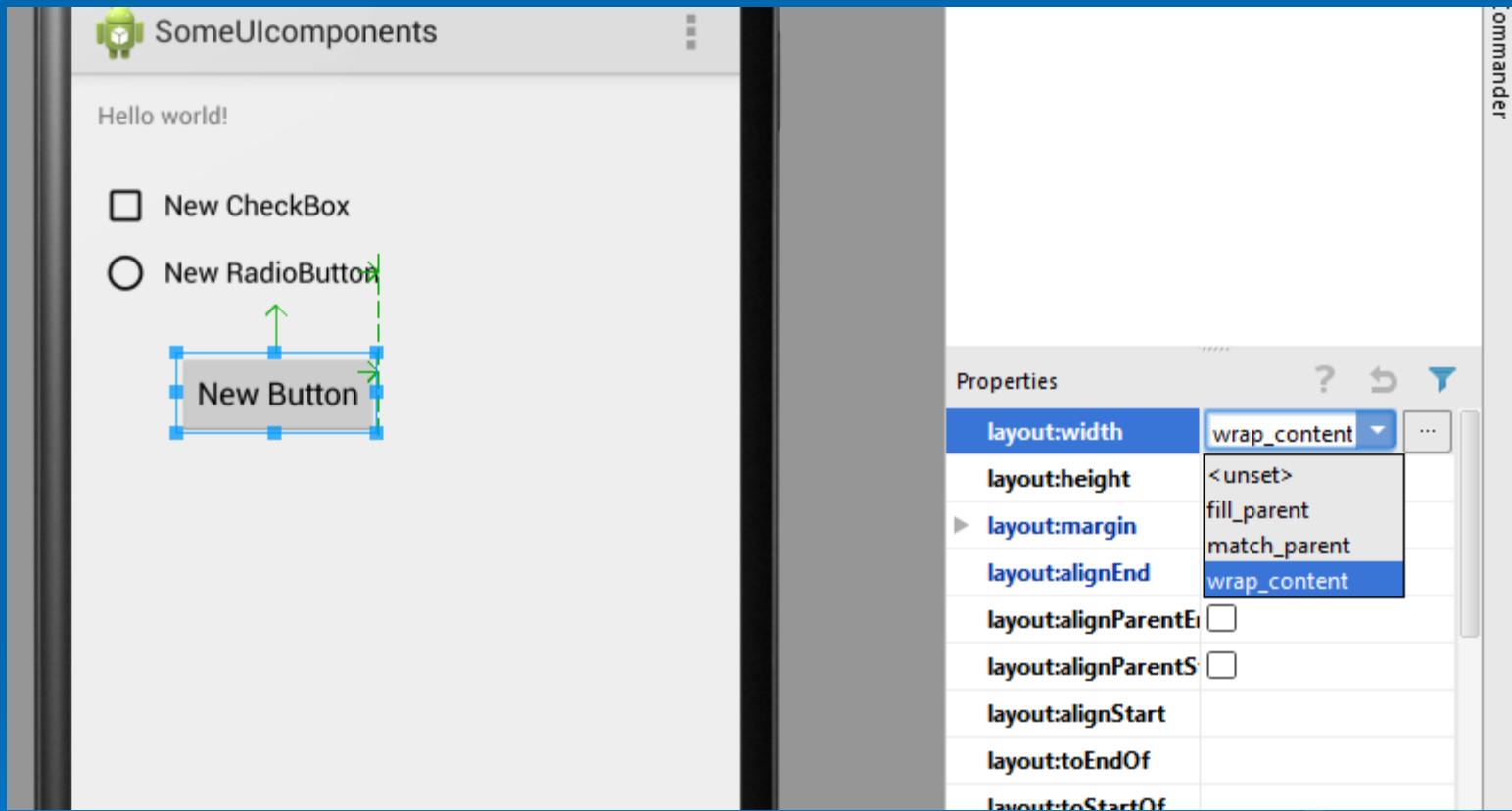
    <TextView android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/textView" />

    <CheckBox
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New CheckBox"
        android:id="@+id/checkBox"
        android:layout_
        android:layout_
        android:layout_
        android:layout_>

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="New Button"
        android:id="@+id/button"
        android:layout_below="@+id/radioButton"
        android:layout_alignRight="@+id/radioButton"
        android:layout_alignEnd="@+id/radioButton"
        android:layout_marginTop="26dp" />

    <RadioButton
        android:layout_
        android:layout_
        android:text="N
        android:id="@+i
        android:layout_
        android:layout_
        android:layout_>

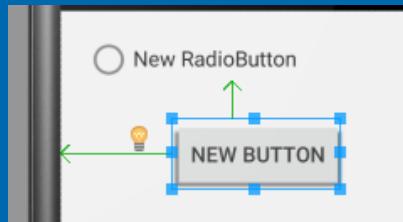
    <Button
        android:layout_
        android:layout_
        android:text="N
        android:id="@+id/button"
        android:layout_below="@+id/radioButton"
        android:layout_alignRight="@+id/radioButton"
        android:layout_alignEnd="@+id/radioButton"
        android:layout_marginTop="26dp" />
```



Layout

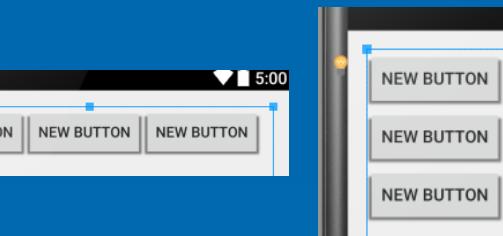
➤ Container

- Hold components together e.g. a Layout, RadioGroup



➤ Relative layout

- The default, limited in use

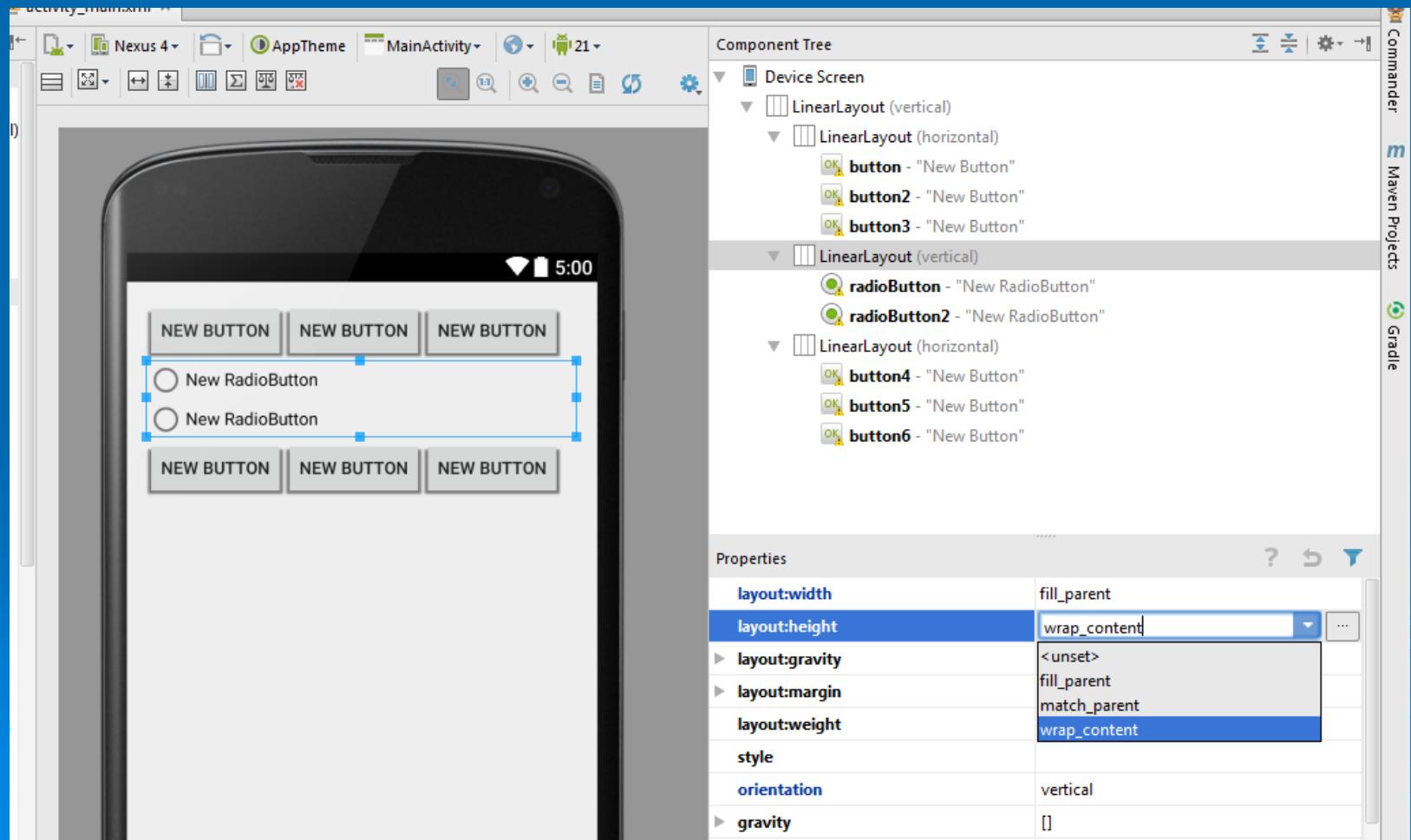


➤ Linear layout

- Horizontal, vertical



➤ Nested layouts



“wire up” UI components so they work

- Android Project Structure provides a framework to work within
- Steps to making components work
 - 1. declare component
 - 2. connect java component to xml component
 - 3. implement a listening interface
 - 4. attach listener to the component
 - 5. provide a method to do the work (the “to do” stuff) when the component is used

➤ Steps to making components work

- 1. declare component
- Button myButton;
- Declare in MainActivity class
- Button default style fixed by system
- Android reference
 - <http://developer.android.com/reference/android/widget/Button.html>

➤ Steps to making components work

- 2. connect java button to xml button
- myButton = (Button)**findViewById(R.id.toastButton);**
- toastButton is the id of the button in the XML or design window
 - android:id="@+id/toastButton"
- **findViewById()**

➤ Steps to making components work

- 3. implement a listening interface
- implements View.OnClickListener
- Placed in class heading
- The word “implements” implies an interface is being used
- Defines method names and their parameters that must be implemented – but does not say how.
- Not to be confused with a user interface
- ```
public void onClick(View view) {
 // to do stuff here
}
```

## ➤ Steps to making components work

- 4. attach listener to the component
- myButton.**setOnClickListener(this);**
- “this” is the MainActivity
- Want to do this when we start app – so goes in onCreate() method

## ➤ Steps to making components work

- 5. provide a method to do the work (the “to do” stuff) when the component is used
- public void **onClick**(View view) {
- // to do stuff here
- }

# Similary for a Checkbox

## ➤ Steps to making components work

- 1. declare component
- CheckBox cb;
- 2. connect java checkbox to xml checkbox
- cb = (CheckBox)**findViewById**(R.id.checkBox);
- 3. implement a listening interface
- implements CompoundButton.OnCheckedChangeListener
- 4. attach listener to the component
- cb.setOnCheckedChangeListener(this);
- 5. provide a method to do the work (the “to do” stuff)
- public void onCheckedChanged(CompoundButton buttonView, boolean isChecked) {
- .....
- }

# MainActivity.java

- package com.example.pnai.someuicomponents;
- import android.app.Activity; //plus some other imports
  
- public class MainActivity extends Activity implements whatever here{
- **// put your declarations here**
- @Override
- protected void onCreate(Bundle savedInstanceState) {
- super.onCreate(savedInstanceState);
- setContentView(R.layout.activity\_main);
- **// put what you want to do on start up here**
- }
  
- **//put your “to do stuff” methods here**
  
- @Override
- public boolean onCreateOptionsMenu(Menu menu) {...}
  
- @Override
- public boolean onOptionsItemSelected(MenuItem item) {...}
- }

# Typing versus Cut and Paste

- Typing java code can get repetitive and tedious
- Typing java code will lead to errors
  - You can learn from these errors!
- Cut and paste often gives code that works
  - But do you always understand what has been done?
  - Note that quotation marks copied from word do not always work
- You will only really get to understand what is going on when you modify what has been done to try new things for yourself.