AAiT
ADDIS ABABA INSTITUTE OF TECHNOLOGY
አዲስ አበባ ቴክኖሎጂ ኢንስቲትዩት
ADDIS ABABA UNIVERSITY
አዲስ አበባ ዩኒቨርሲቲ

CS 2122:- Mobile Programming

By Mesfin Belachew /PhD/

Assistant Professor,

AAiT
ADDIS ABABA INSTITUTE OF TECHNOLOGY
አዲስ አበባ ቴክኖሎጂ ኢንስቲትዩት
ADDIS ABABA UNIVERSITY
አዲስ አበባ ዩኒቨርሲቲ

Android Mobile Programming

1 Android Major Components

– There are four main Android app components:

- **Activities**: entry point for interacting with the user (Ex. To start Camera, read Email, etc.)**,**
- **Services**: general-purpose entry point for keeping an app running in the background (Ex. Play music, notification listeners, screen savers, etc.)**,**
- Content **providers**: share app data that can be stored on file, database, web, or on any storage location (Ex. contact information, SQLite database, etc.), and
- Broadcast **receivers**: enables the system to deliver events to the app outside of a regular user flow (Ex. alarm to post a notification, alert, etc.).

– Whenever you create or use any of them, you must include elements in the project manifest .

Android Mobile Programming …..

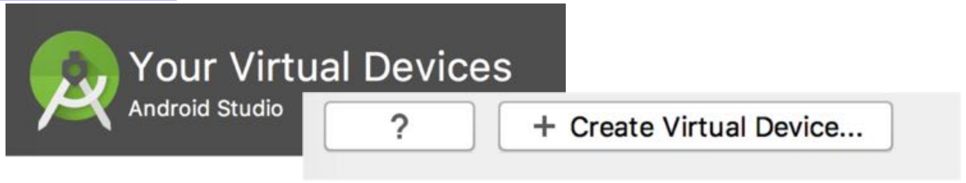
- Components are declared in the manifest, namely in the file name “AndroidManifest.xml”
- The main components that are required for the created App will be generated by default using the Android Studio
- each components has its own syntax to be followed

```
<manifest ... >
   ...
   <application ... >
      <activity android:name=”name_of_activity">
              ……
      </activity>
      <receiver android:name=”name_of_the_receiver">
              ……
      </receiver>
      <service android:name="name_of_the_service">
              ……
      </service>
      <provider  android:name="name_of_the_ provider">
              ……
      </provider>
   </application>
</manifest>
```

Android Mobile Programming …..

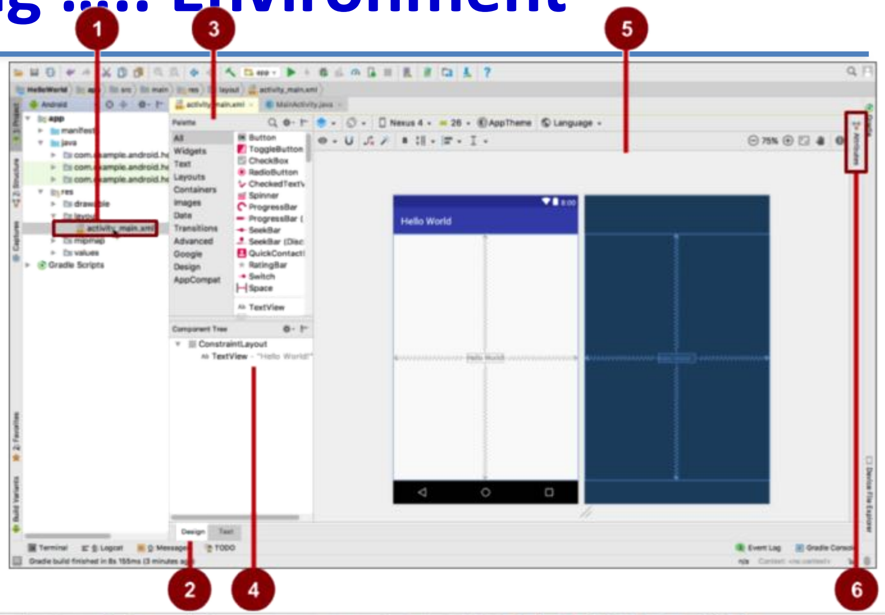
2 Android Execution Environment

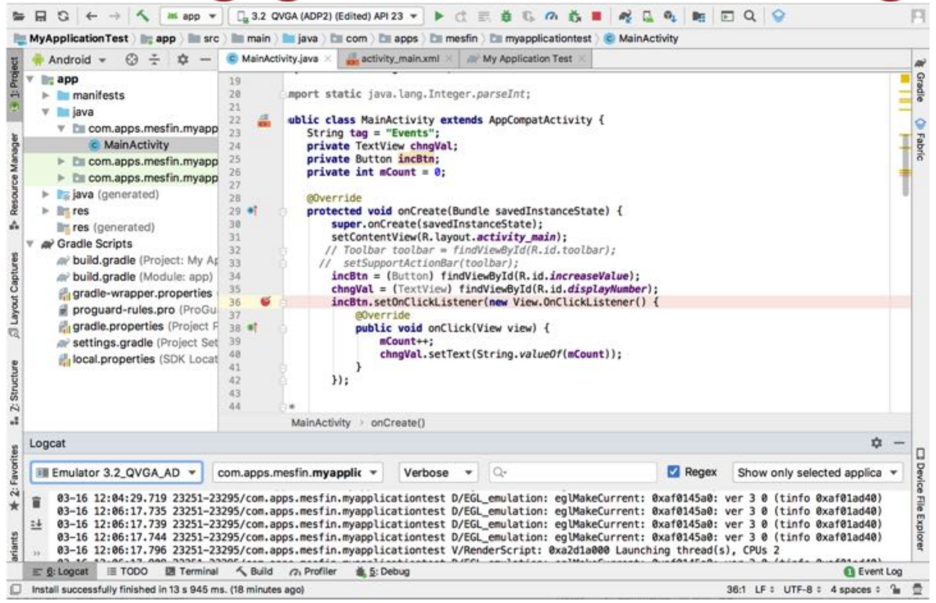
- Set Up Android Environment
  - need an android SDK and Java Development Kit (JDK)
  - Java syntax is used for writing code,
  - run on Dalvik Virtual Machine,
- Android Development Tools,
  - Android SDK (https://developer.android.com/studio)
  - Android Virtual Device (AVD), once installed,
- HelloWorld: First Android Application,



Android Mobile Programming ….. Environment

- Android Studio Layout Editor (intro)
  - First window after an app is created/opened using the studio,
  - 1 – Folder section, to select one item,
  - 2- to switch between design & text (xml form) mode
  - 3- Palette – drag & drop elements,
  - 4- component tree to see the hierarchy of the components,
  - 5 – view window to display xml, java, design view
  - 6- Preview button to show some preview, if applicable





```
mport static java.lang.Integer.parseInt;

ublic class MainActivity extends AppCompatActivity {
    String tag = "Events";
    private TextView chngVal;
    private Button incBtn;
    private int mCount = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
      // Toolbar toolbar = findViewById(R.id.toolbar);
     //  setSupportActionBar(toolbar);
        incBtn = (Button) findViewById(R.id.increaseValue);
        chngVal = (TextView) findViewById(R.id.displayNumber);
        incBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                mCount++;
                chngVal.setText(String.valueOf(mCount));
            }
        });
```

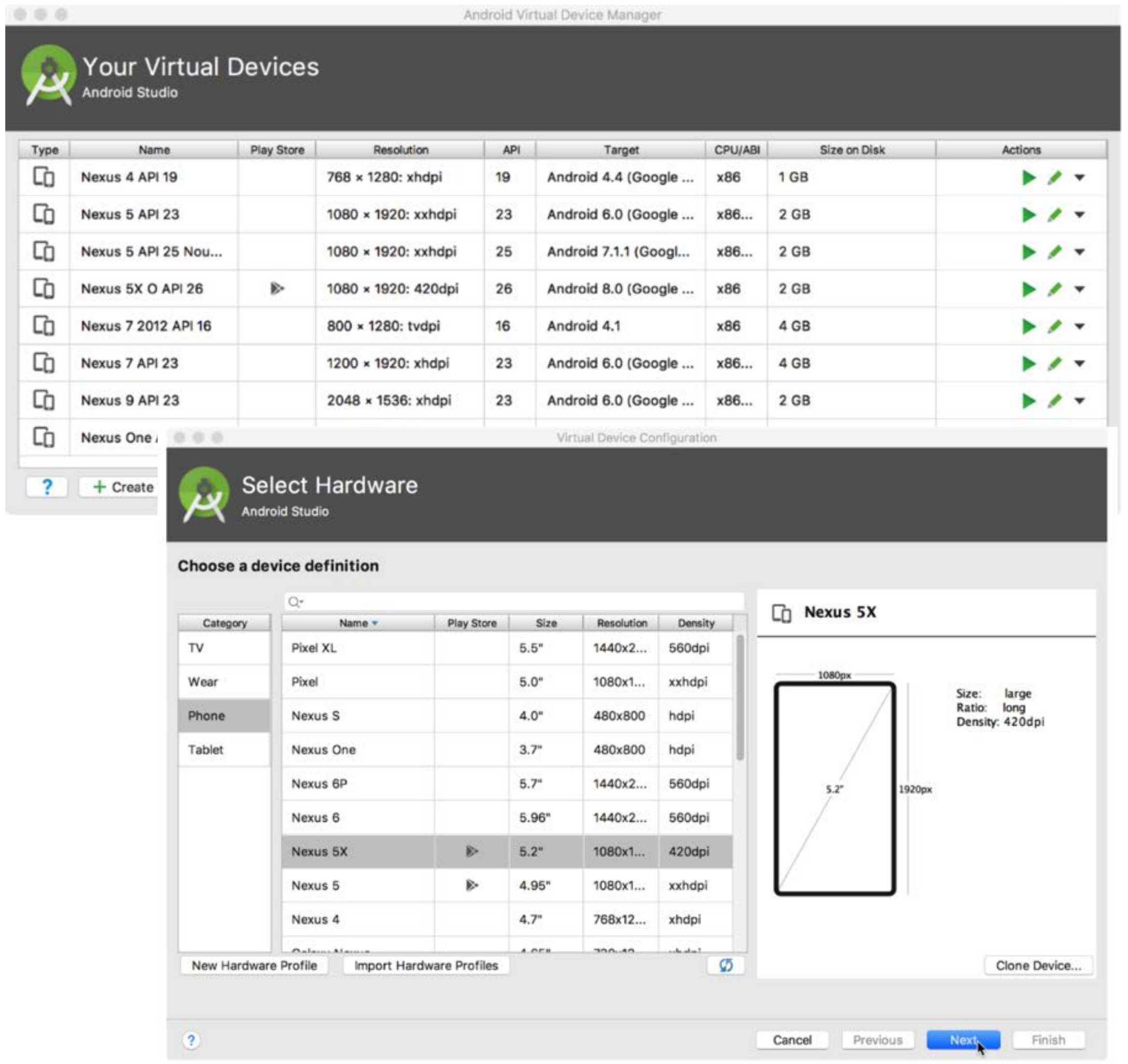
```
03-16 12:04:29.719 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
03-16 12:06:17.735 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
03-16 12:06:17.739 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
03-16 12:06:17.744 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
03-16 12:06:17.796 23251-23295/com.apps.mesfin.myapplicationtest V/RenderScript: 0xa2d1a000 Launching thread(s), CPUs 2
```

Install successfully finished in 13 s 945 ms. (18 minutes ago)

Android Mobile Programming ….. Environment

- Create Virtual Device (Emulator)

- In Android Studio, select Tools > Android > AVD Manager,
- then +Create Virtual Device,
- Set display size (Size), API, pixels (Resolution & Density), choose a device, etc.
- Once it is completed, new emulator will listed and ready for use,
- While running the App, you need to select emulator



Android Virtual Device Manager

Your Virtual Devices
Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk	Actions
	Nexus 4 API 19		768 × 1280: xhdpi	19	Android 4.4 (Google ...	x86	1 GB	
	Nexus 5 API 23		1080 × 1920: xxhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus 5 API 25 Nou...		1080 × 1920: xxhdpi	25	Android 7.1.1 (Googl...	x86...	2 GB	
	Nexus 5X O API 26	▶	1080 × 1920: 420dpi	26	Android 8.0 (Google ...	x86	2 GB	
	Nexus 7 2012 API 16		800 × 1280: tvdpi	16	Android 4.1	x86	4 GB	
	Nexus 7 API 23		1200 × 1920: xhdpi	23	Android 6.0 (Google ...	x86...	4 GB	
	Nexus 9 API 23		2048 × 1536: xhdpi	23	Android 6.0 (Google ...	x86...	2 GB	
	Nexus One							

Virtual Device Configuration

Select Hardware
Android Studio

Choose a device definition

Category
TV
Wear
Phone
Tablet

Name	Play Store	Size	Resolution	Density
Pixel XL		5.5"	1440x2...	560dpi
Pixel		5.0"	1080x1...	xxhdpi
Nexus S		4.0"	480x800	hdpi
Nexus One		3.7"	480x800	hdpi
Nexus 6P		5.7"	1440x2...	560dpi
Nexus 6		5.96"	1440x2...	560dpi
Nexus 5X	▶	5.2"	1080x1...	420dpi
Nexus 5	▶	4.95"	1080x1...	xxhdpi
Nexus 4		4.7"	768x12...	xhdpi

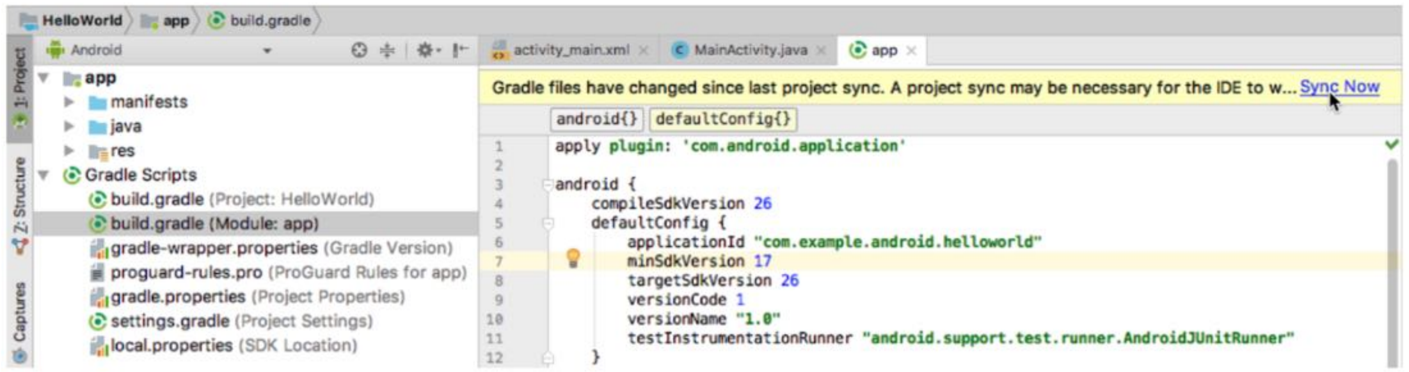
Nexus 5X

Size: large
Ratio: long
Density: 420dpi

You can also use your own physical device, to be connected through USB cable, set the developer options in settings app

Android Mobile Programming ….. Environment

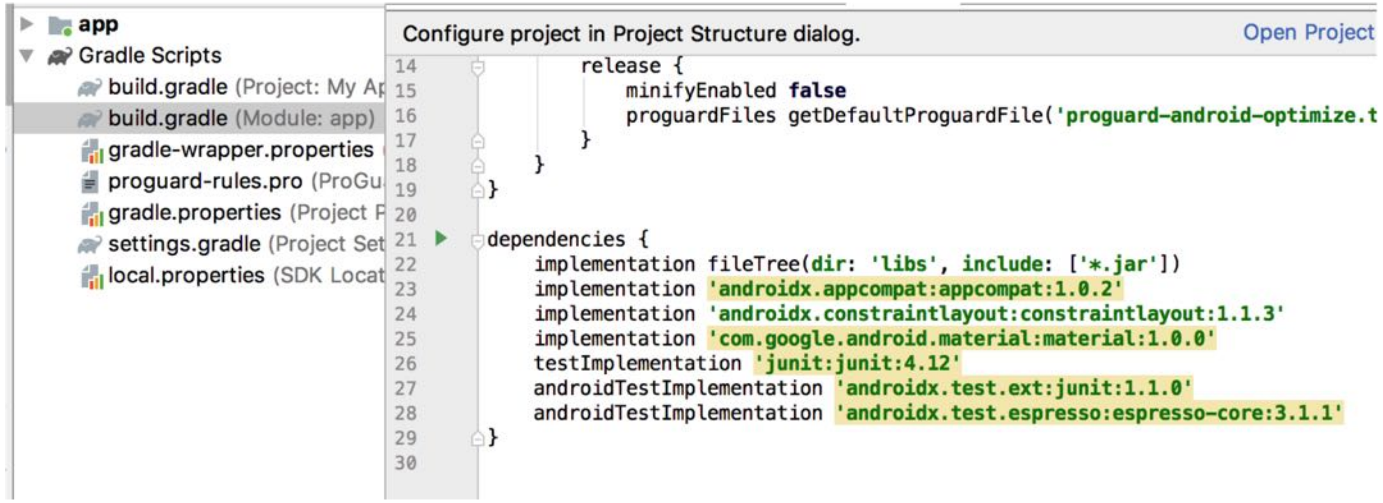
- Gradle configuration
- Is used to adjust some settings about your mobile App, like version, libraries,
- Located in the build.gradle(Module:app),
- Used to define some dependencies to be used,
- Version will be set here when some major updates are happening


HelloWorld
app
build.gradle
Android
app
manifests
java
res
Gradle Scripts
build.gradle (Project: HelloWorld)
build.gradle (Module: app)
gradle-wrapper.properties (Gradle Version)
proguard-rules.pro (ProGuard Rules for app)
gradle.properties (Project Properties)
settings.gradle (Project Settings)
local.properties (SDK Location)
activity_main.xml
MainActivity.java
app
Sync Now
android{}
defaultConfig{}

Gradle files have changed since last project sync. A project sync may be necessary for the IDE to w...

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    defaultConfig {
        applicationId "com.example.android.helloworld"
        minSdkVersion 17
        targetSdkVersion 26
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
    }
```


app
Gradle Scripts
build.gradle (Project: My Ap
build.gradle (Module: app)
gradle-wrapper.properties
proguard-rules.pro (ProGu
gradle.properties (Project P
settings.gradle (Project Set
local.properties (SDK Locat
Open Project

Configure project in Project Structure dialog.

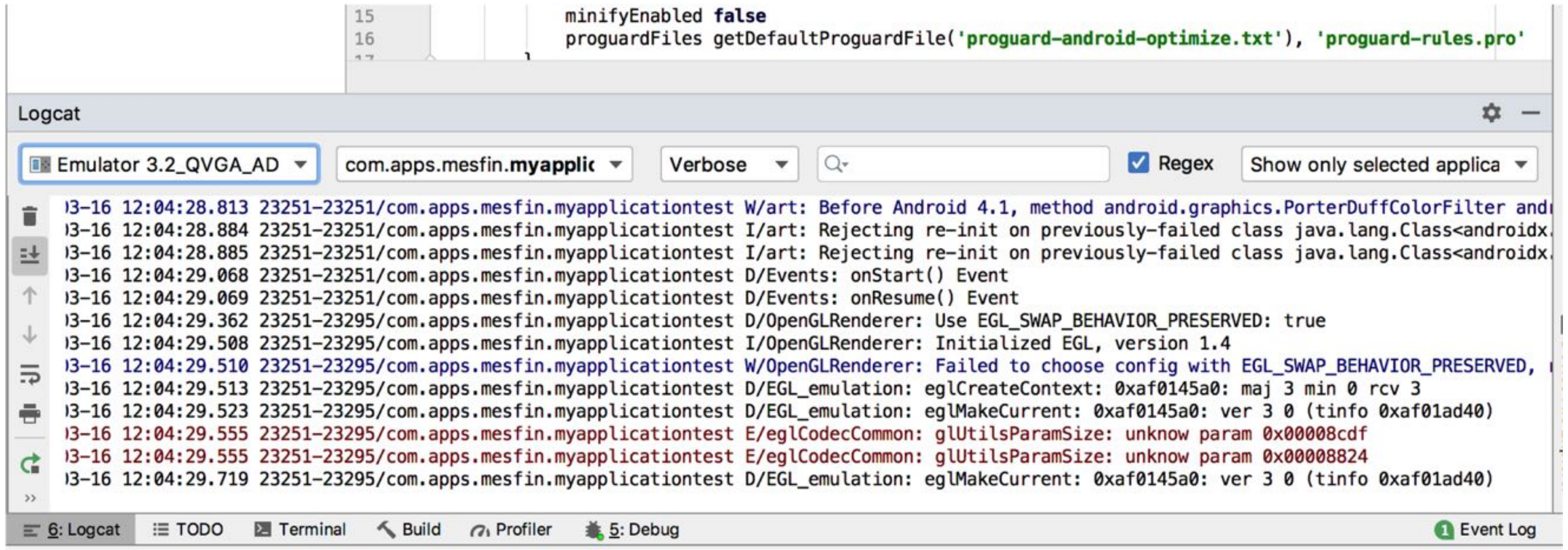
```
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android-optimize.t
        }
    }
}

dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.0.2'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'com.google.android.material:material:1.0.0'
    testImplementation 'junit:junit:4.12'
    androidTestImplementation 'androidx.test.ext:junit:1.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.1'
}
```

Android Mobile Programming ….. Environment

- Log Statements

- Logcat pan is used to display messages, errors, warnings, etc. while compiling, debugging and running the App,
- The window/pan has different options to display variety of information about the running App,

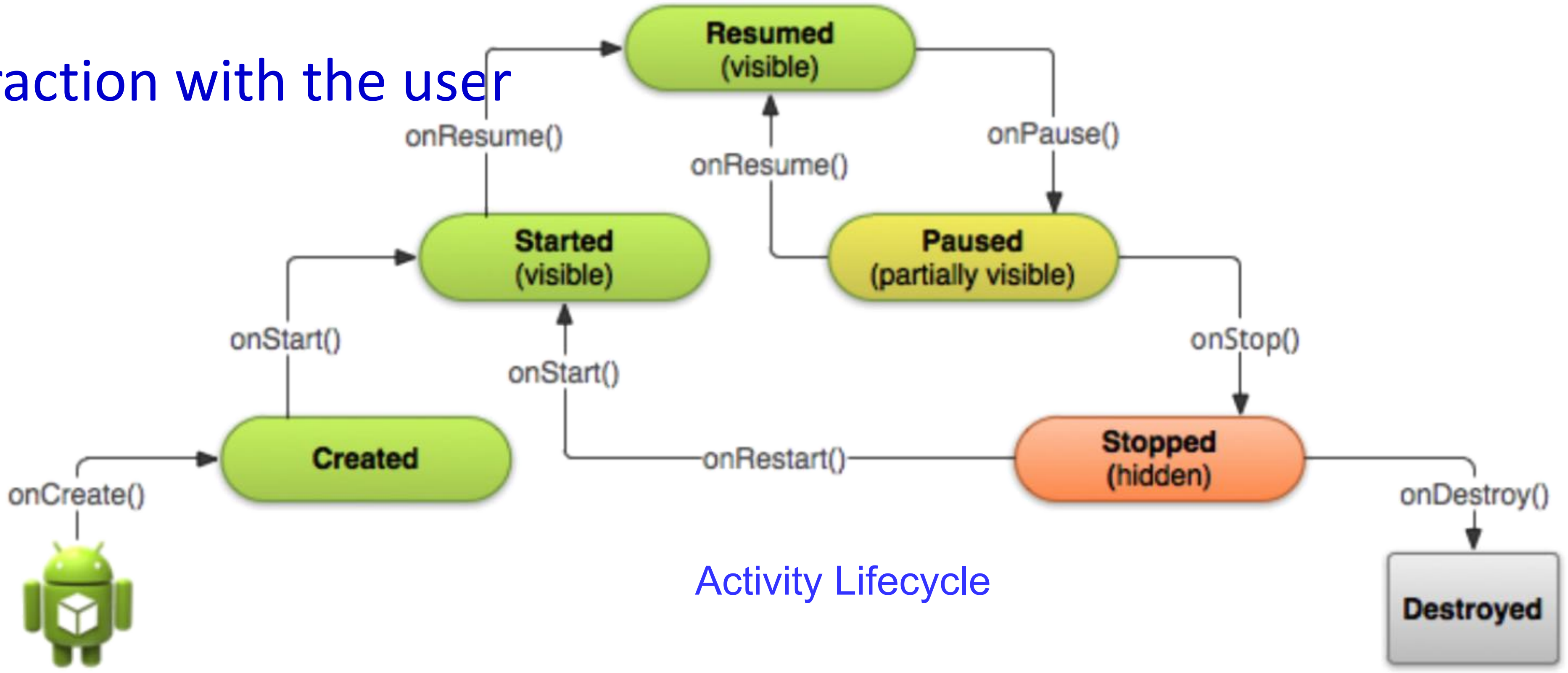


```
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
```

```
3-16 12:04:28.813 23251-23251/com.apps.mesfin.myapplicationtest W/art: Before Android 4.1, method android.graphics.PorterDuffColorFilter and
3-16 12:04:28.884 23251-23251/com.apps.mesfin.myapplicationtest I/art: Rejecting re-init on previously-failed class java.lang.Class<androidx.
3-16 12:04:28.885 23251-23251/com.apps.mesfin.myapplicationtest I/art: Rejecting re-init on previously-failed class java.lang.Class<androidx.
3-16 12:04:29.068 23251-23251/com.apps.mesfin.myapplicationtest D/Events: onStart() Event
3-16 12:04:29.069 23251-23251/com.apps.mesfin.myapplicationtest D/Events: onResume() Event
3-16 12:04:29.362 23251-23295/com.apps.mesfin.myapplicationtest D/OpenGLRenderer: Use EGL_SWAP_BEHAVIOR_PRESERVED: true
3-16 12:04:29.508 23251-23295/com.apps.mesfin.myapplicationtest I/OpenGLRenderer: Initialized EGL, version 1.4
3-16 12:04:29.510 23251-23295/com.apps.mesfin.myapplicationtest W/OpenGLRenderer: Failed to choose config with EGL_SWAP_BEHAVIOR_PRESERVED,
3-16 12:04:29.513 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglCreateContext: 0xaf0145a0: maj 3 min 0 rcv 3
3-16 12:04:29.523 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
3-16 12:04:29.555 23251-23295/com.apps.mesfin.myapplicationtest E/eglCodecCommon: glUtilsParamSize: unknow param 0x00008cdf
3-16 12:04:29.555 23251-23295/com.apps.mesfin.myapplicationtest E/eglCodecCommon: glUtilsParamSize: unknow param 0x00008824
3-16 12:04:29.719 23251-23295/com.apps.mesfin.myapplicationtest D/EGL_emulation: eglMakeCurrent: 0xaf0145a0: ver 3 0 (tinfo 0xaf01ad40)
```

3 Android Activities

- Activity Lifecycle indicate the path of activity from it's creation up to it's destroyed
- Shows the interaction with the user
  - onCreate():
  - onStart():
  - onResume():
  - onRestart():
  - onPause()
  - onStop()
  - onDestroy(),



Activity Lifecycle

Android Mobile Programming ….. Activities

- Android Activities

```
package com.Debosmita.Activitytest;
import android.os.Bundle;
import android.app.Activity;
import android.util.Log;

public class MainActivity extends Activity {
    String tag = "Events";
    @Override
    protected void onCreate(Bundle
      savedInstanceState) {
      super.onCreate(savedInstanceState);
     setContentView(R.layout.activity_main);
      Log.d(tag, "onCreate() Event");
   }
```

```
@Override
public void onStart() {
      super.onStart();
      Log.d(tag, "onStart() Event");

  }

@Override
public void onRestart() {
      super.onRestart();
      Log.d(tag, "onRestart() Event");
}
@Override
public void onResume() {
      super.onResume();
      Log.d(tag, "onResume() Event");
}
```

```
@Override
public void onPause() {
      super.onPause();
      Log.d(tag, "onPause() Event");
}
@Override
public void onStop() {
      super.onStop();
      Log.d(tag, "onStop() Event");
}
@Override
public void onDestroy() {
      super.onDestroy();
      Log.d(tag, "onDestroy() Event");
}

}
```

Android Mobile Programming ….. Activities

- Android Activities, when you run the App, you will see how the different activities are executed, who is following whom ……

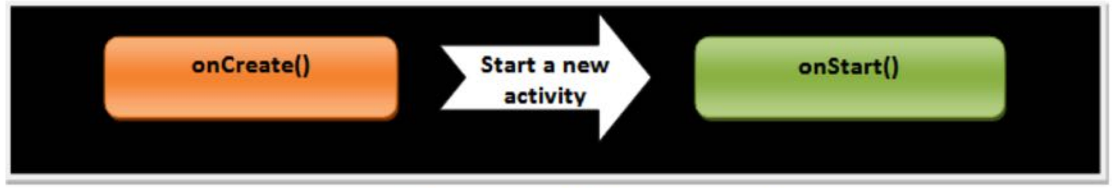


Figure onCrete() followed by onStart()

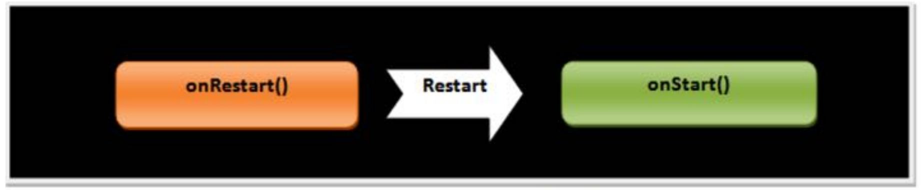


Figure onRestart() is followed by onStart()

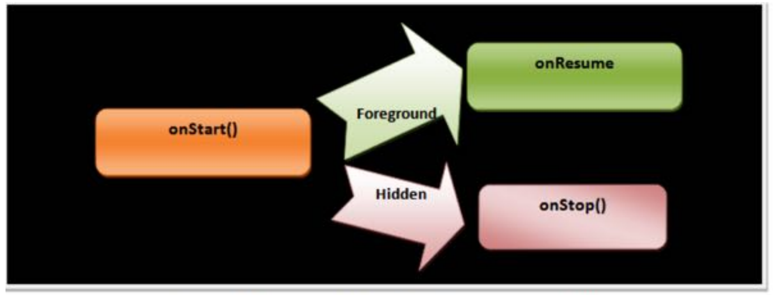


Figure onStart() may be followed by onResume() or onStop()

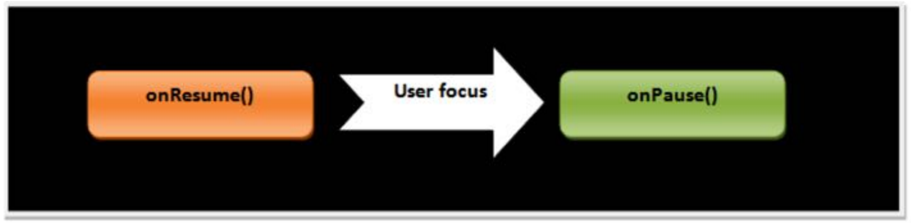


Figure onResume() is followed by onPause()

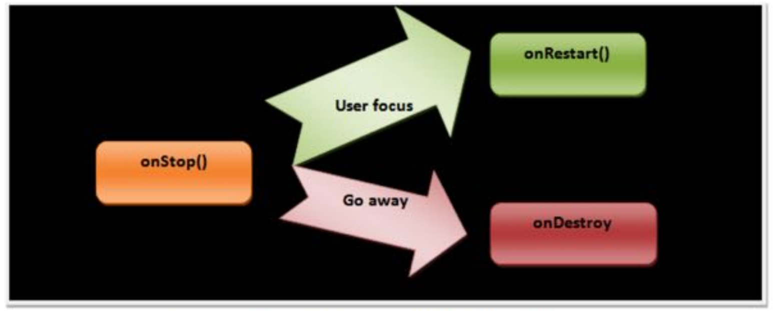


Figure onStop() may be followed by onRestart() or onDestroy().

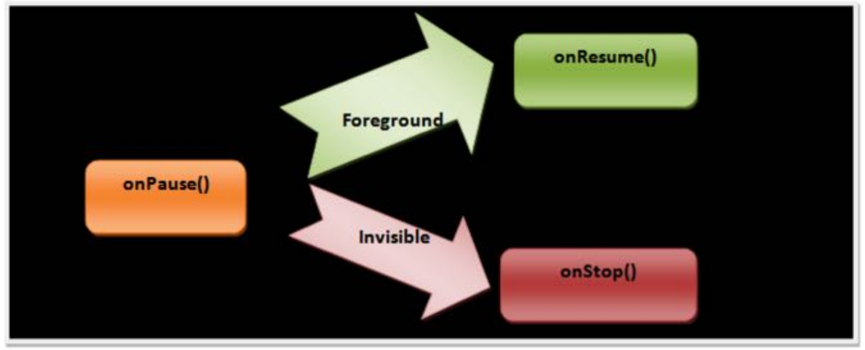


Figure onStop() may be followed by onRestart() or onDestroy().

4 Android View/Layout Editor

- is responsible for the user interface of the Android application,
- is a widget which appears on screen, in XML file
- many view objects in Android, like buttons, text, text edit, image view, check/radio/toggle button, switch, etc.
- There are also view groups to show multiple view objects in specific order/fashion, like Linear, Table, Relative, Frame, Grid and Absolute Layout

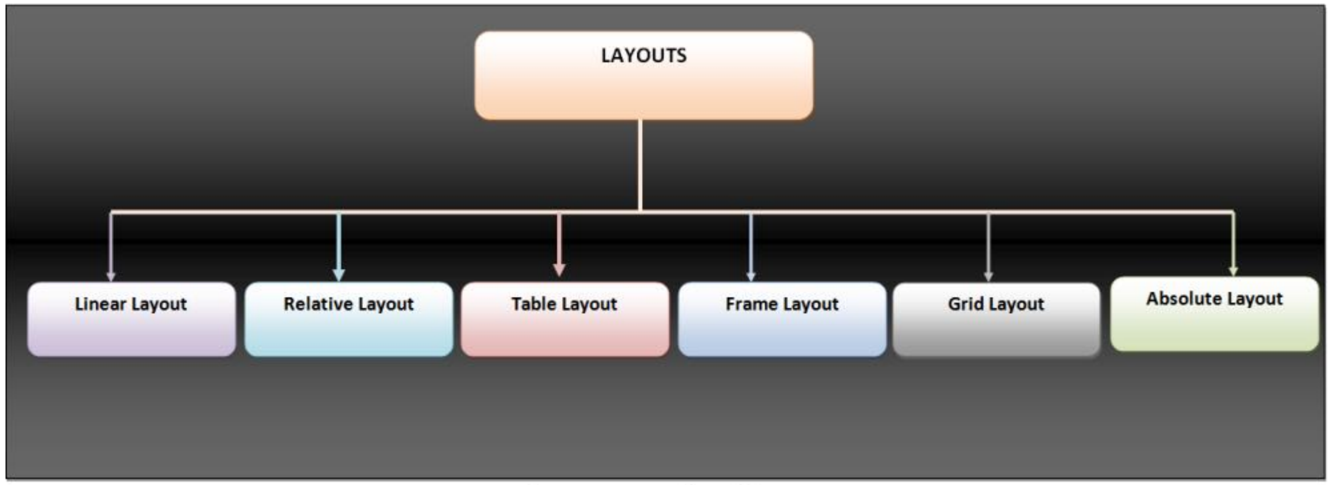
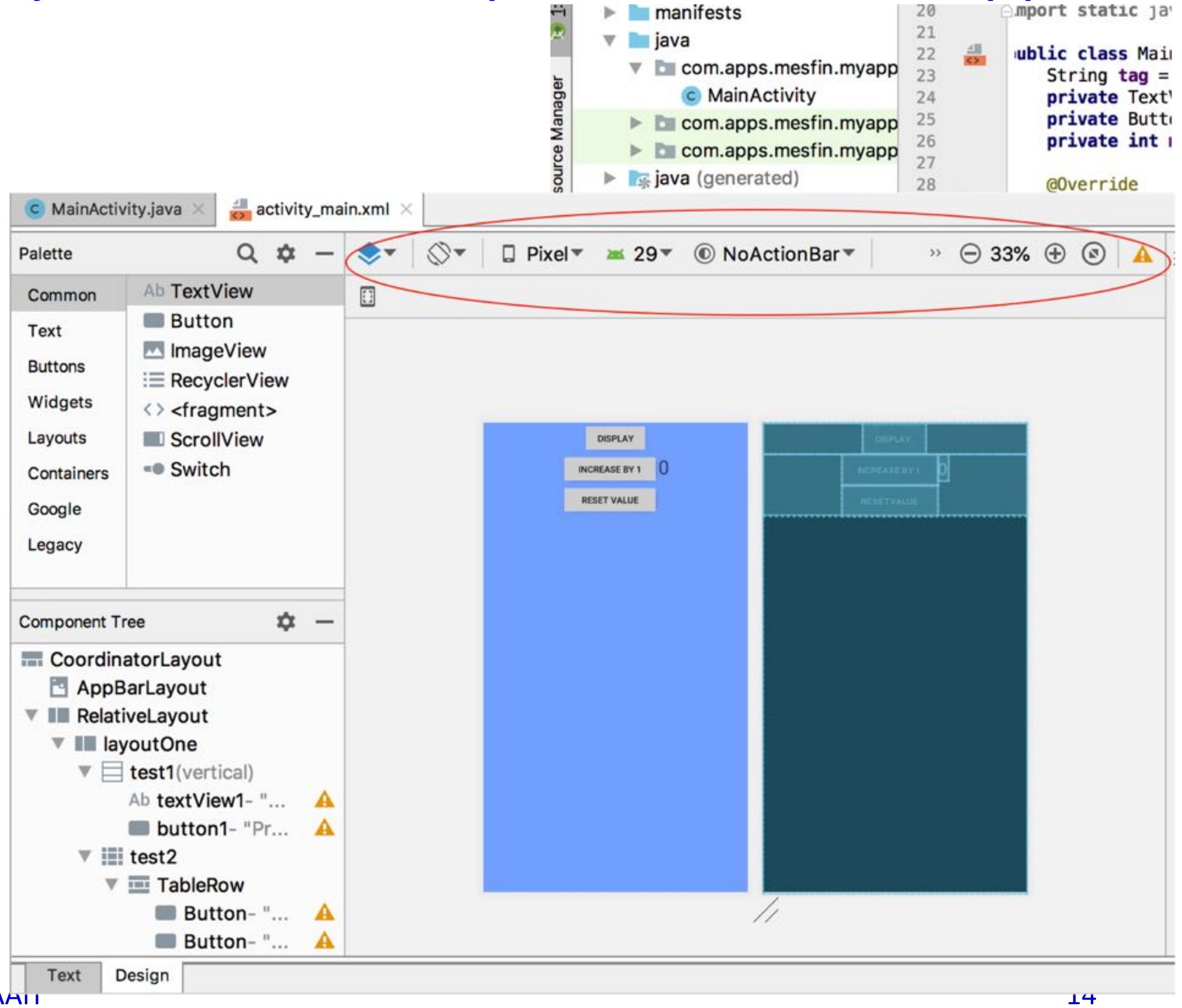


Figure Layouts supported by Android System

Android Mobile Programming ….. View

- Layout Editor is used to edit the layout of the components on the App,
  - Contains all layout xml files
  - To select select design surface, design + blueprint, orientation, phone pixel, API version, action bar type, size of the preview, etc.,
  - Move, position, resize, etc. components in the editor
  - All the above can also be done using xml code,

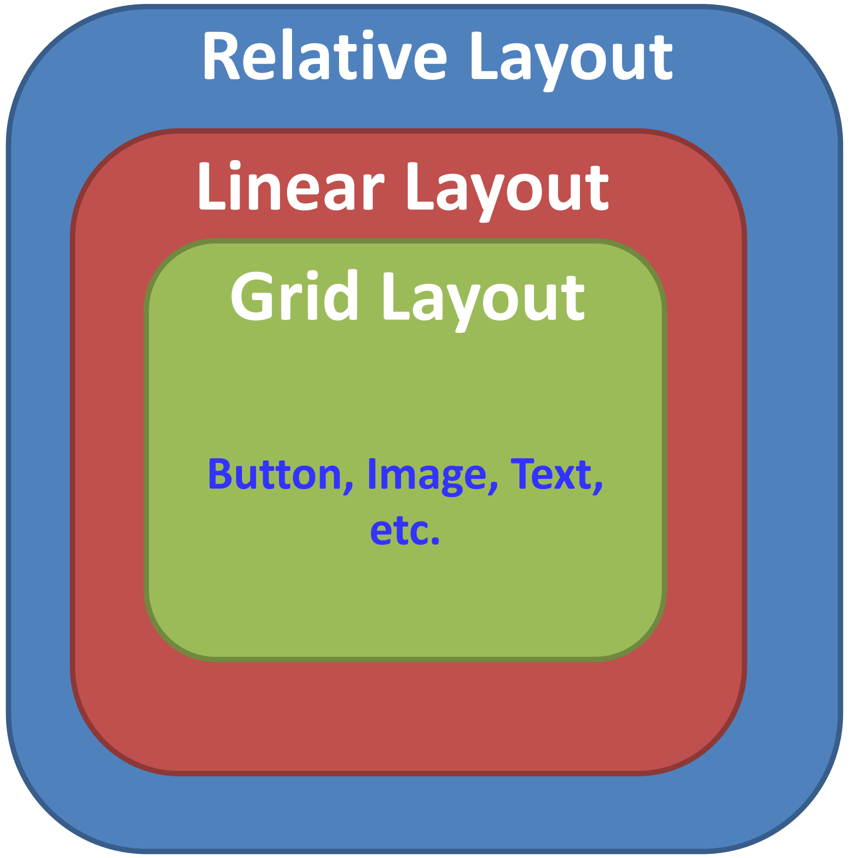


```
mport static ja
ublic class Mai
  String tag =
  private Text
  private Butt
  private int

  @Override
```

Android Mobile Programming ….. View

– View Groups can be also concatenated (linked in chain)

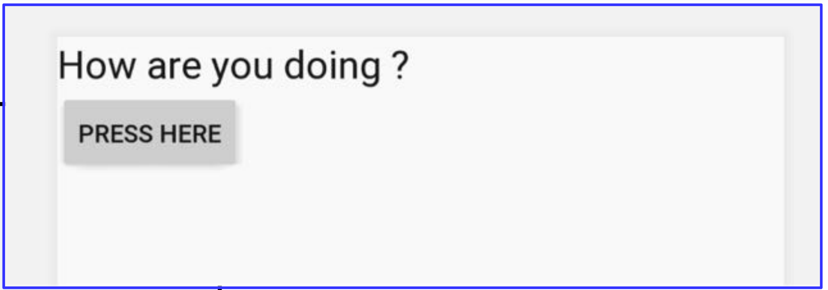


```
<RelativeLayout
  android:layout_width="wrap_content"
  android:layout_height="wrap_content">
    <LinearLayout
       android:layout_width="wrap_content"
       android:layout_height="wrap_content">
       <GridLayout
          android:layout_width="wrap_content"
          android:layout_height="wrap_content">
          <!-- view objects, like buttons, text, text edit,
          image view, -->
       </GridLayout>
    </LinearLayout>
</RelativeLayout>
```

Android Mobile Programming ..... View

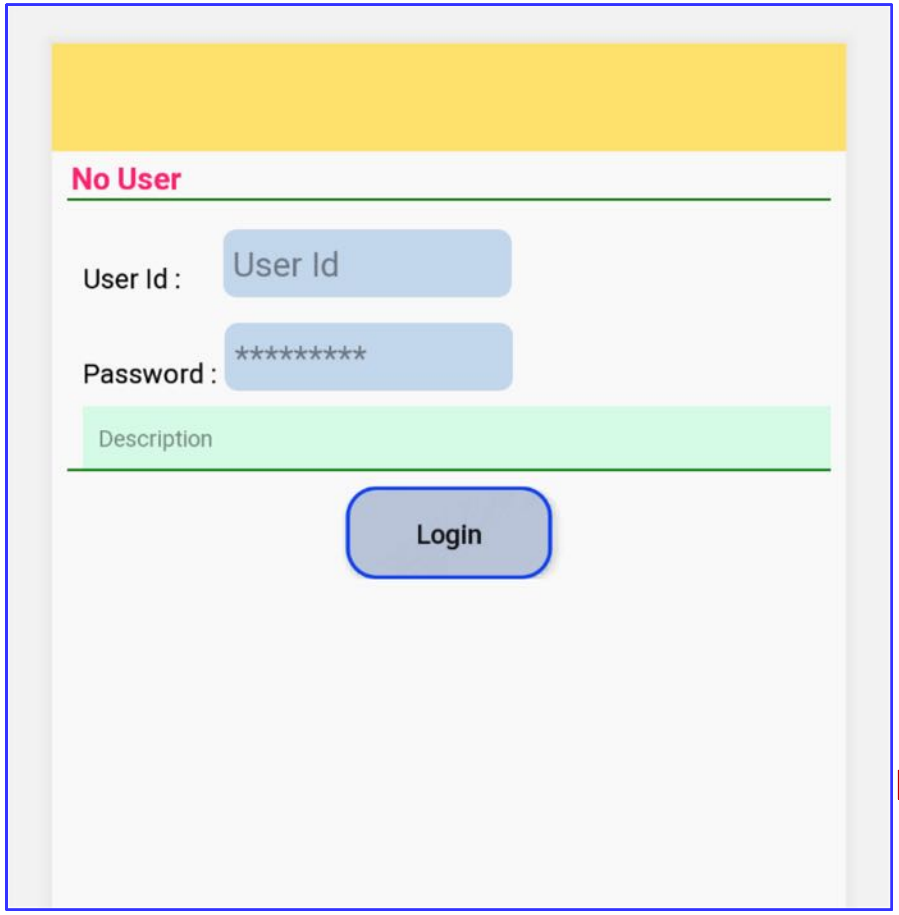
– Example, Linear Layout

```
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height=" wrap_content "
    android:orientation="vertical" >
    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="How are you doing ?"
        android:textAppearance="?android:attr/textAppearanceLarge" />
     <Button
        android:id="@+id/button1"
        android:layout_width="152dp"
        android:layout_height="wrap_content"
        android:text="Press Here" />
  </LinearLayout>
```

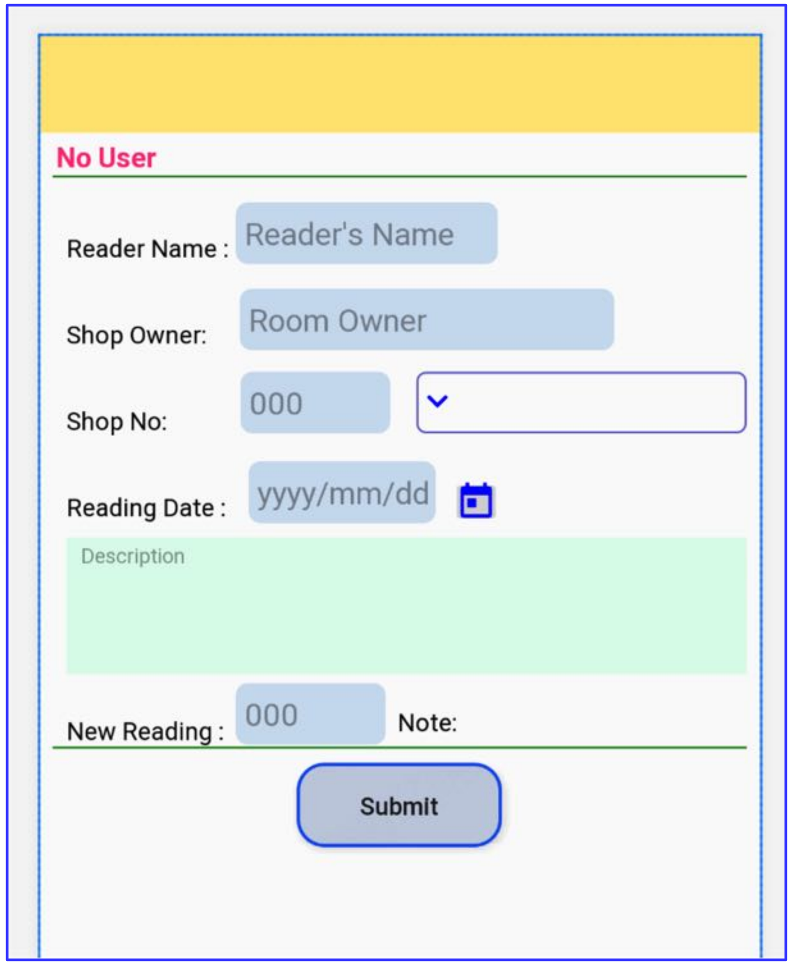

How are you doing ?

Android Mobile Programming ….. View

– For your practice, try to design the screen shot shown below.



Hint: RelativeLayout is best



Android Mobile Programming ….. View

– Table Layout

```
<TableLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height=" wrap_content ">
        <TableRow >
             <!-- view objects, like buttons, text, text edit,
                              image view, -->
         </TableRow>
         <TableRow >
             <!-- view objects, like buttons, text, text edit,
                              image view, -->
         </TableRow>
  </ TableLayout >
```

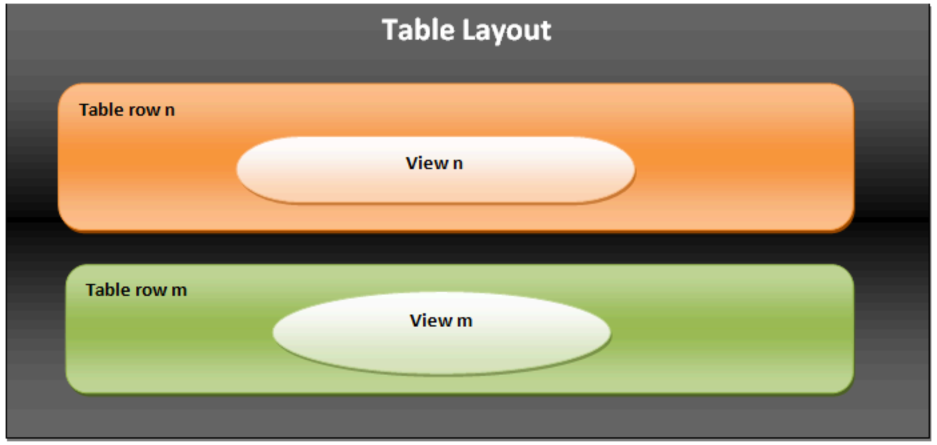
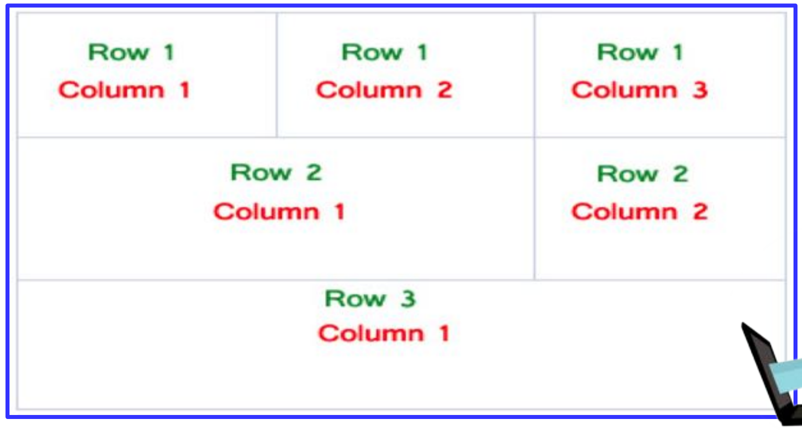


Figure Table Layout



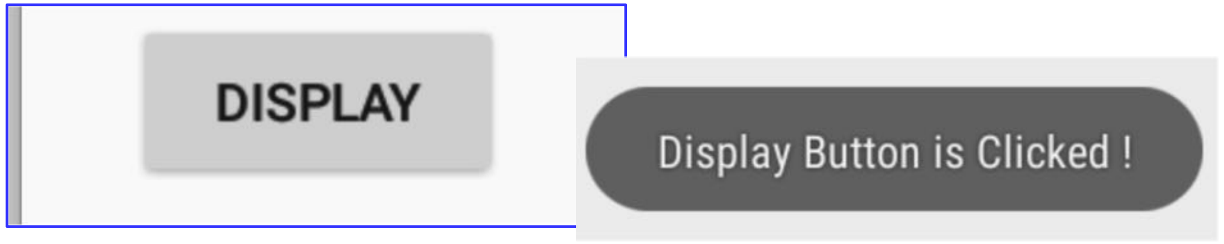
Hint: use layout_span in the table leyout

5 Interactive User Interface

- invoked action when the user clicks or taps on a clickable UI element,
- Called a click handler,
- It can be defined in two ways, in the XML editor or in the main activity using java code (both have the same effect)
- Procedure:-
  - Create the UI component, it can be Button, Image, etc.
  - Assign ID for the UI, make the it Clickable, usually in XML ,
  - Add a method in the main activity to react to the action of the interaction,

Android Mobile Programming ….. Interactive UI

- Simple Exercise (defining the UI interaction in XML)

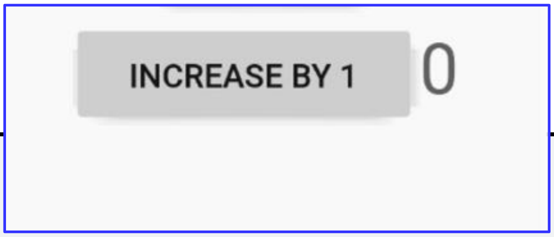


```
<RelativeLayout
  android:layout_width="wrap_content"
  android:layout_height="wrap_content”
  android:gravity="center">
    <Button
      android:id="@+id/displayText"
      android:layout_width="100dp"
      android:layout_height="wrap_content"
      android:text="Display"
      android:onClick="showToast"/>
</RelativeLayout>
```

```
……….
@Override
protected void onCreate(Bundle savedInstanceState) {
  super.onCreate(savedInstanceState);
  setContentView(R.layout.activity_main);
}
public void showToast(View view)
{
  Toast.makeText(getApplicationContext(), "Display
 Button is Clicked !", Toast.LENGTH_SHORT).show();
}
………..
```

Android Mobile Programming ….. Interactive UI

- Simple Exercise (defining the UI interaction in the main activity)



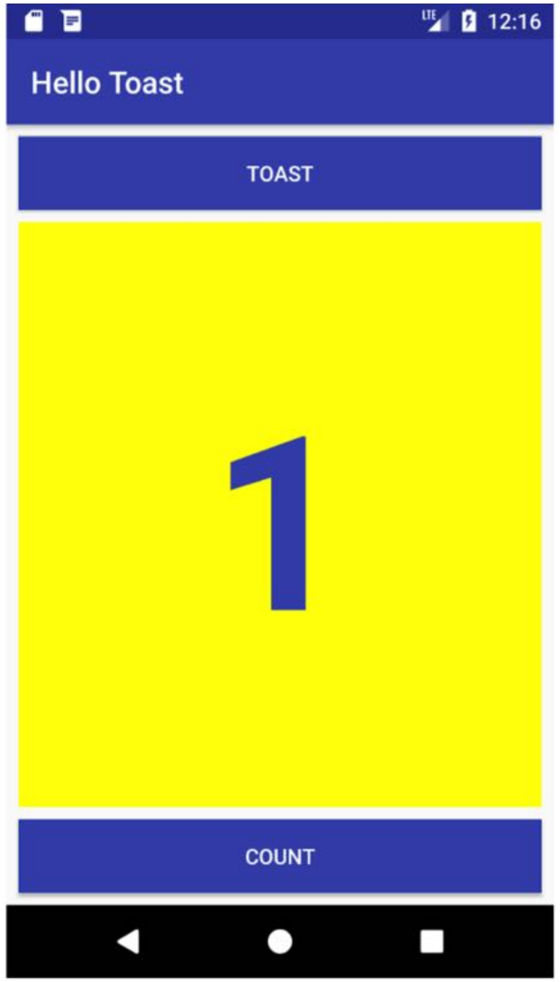
```
……..
<Button
   android:id="@+id/increaseValue"
   android:layout_width="150dp"
   android:layout_height="wrap_content"
   android:text="Increase by 1" />
<TextView
   android:id="@+id/displayNumber"
   android:layout_width="wrap_content"
   android:layout_height="wrap_content"
   android:layout_toRightOf="@id/increaseValue"
   android:textSize="30sp"
   android:text="0"/>
………..
```

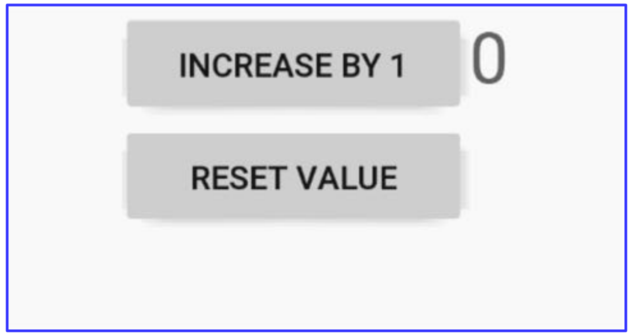
```
……….
public class MainActivity extends AppCompatActivity {
    private int mCount = 0;
    private TextView chngVal;
    private Button incBtn;
………..
```

```
protected void onCreate(Bundle savedInstanceState) {
……….
    incBtn = (Button) findViewById(R.id.increaseValue);
    chngVal = (TextView) findViewById(R.id.displayNumber);
    incBtn.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View view) {
         mCount++;
         chngVal.setText(String.valueOf(mCount));
      } });
    }
………..
```

Android Mobile Programming ….. Interactive UI

- For your exercise





6 Text and Scrolling Views

- TextView – one of widely used component,
- TextView and Scrolling View classes are a subclass of the View class
- Text view - to display single/multiple lines of text
- Scrolling view –to scroll content if the size doesn't fit to the device display,
- Scrolling view – can be applied on single components or even a group of components in specific layout

Android Mobile Programming ….. Layout Editor

- TextView & Scrolling View – Explanations

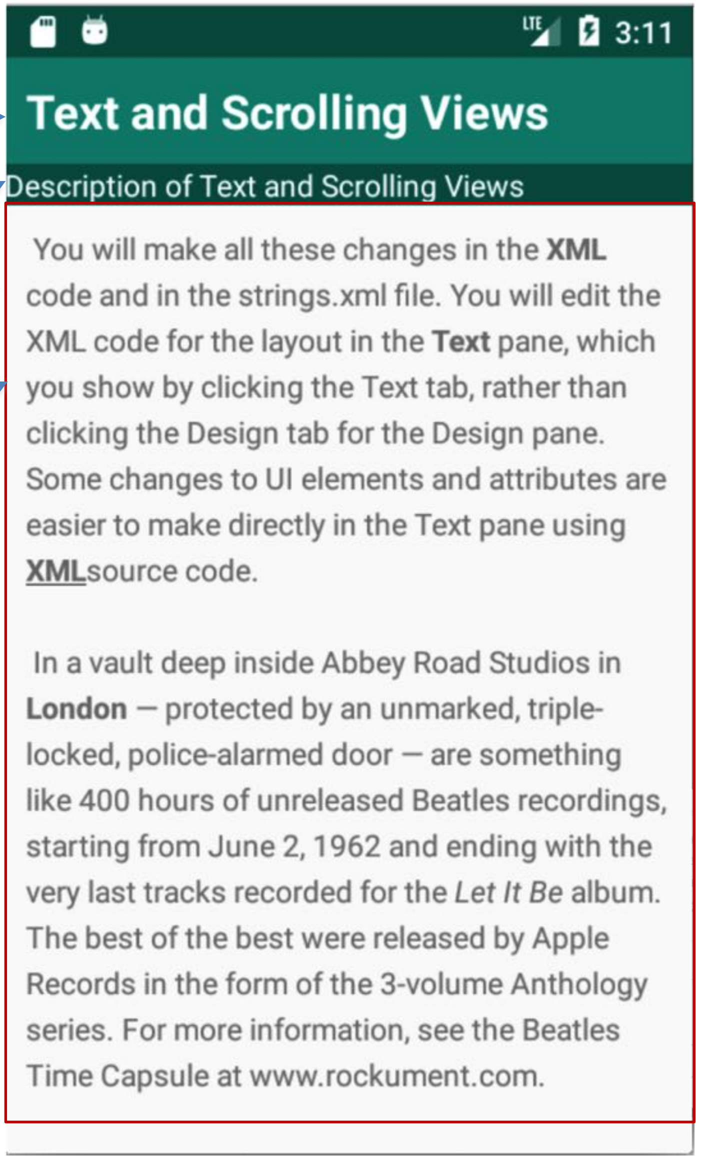
```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height=" wrap_content ">
        <TextView >
            android:id="@+id/article_heading”
            ………../>
         <TextView >
            android:id="@+id/article_subheading”
            ………../>
            <ScrollView
              android:layout_width="wrap_content"
              android:layout_height="wrap_content”>
                  <TextView >
                android:id="@+id/article”
                ………../>
           </ScrollView>
      </ RelativeLayout >
```

TextView 1

TextView 2

TextView 3

Scroll View



Text and Scrolling Views

Description of Text and Scrolling Views

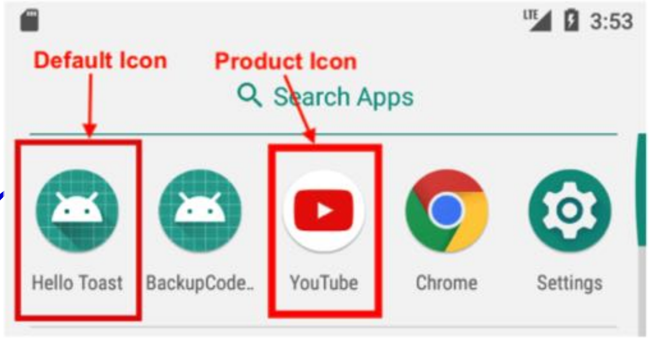
You will make all these changes in the **XML** code and in the strings.xml file. You will edit the XML code for the layout in the **Text** pane, which you show by clicking the Text tab, rather than clicking the Design tab for the Design pane. Some changes to UI elements and attributes are easier to make directly in the Text pane using **XML**source code.

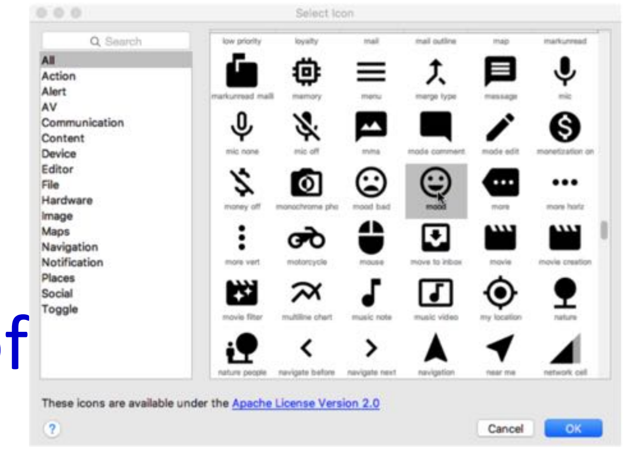
In a vault deep inside Abbey Road Studios in **London** — protected by an unmarked, triple-locked, police-alarmed door — are something like 400 hours of unreleased Beatles recordings, starting from June 2, 1962 and ending with the very last tracks recorded for the *Let It Be* album. The best of the best were released by Apple Records in the form of the 3-volume Anthology series. For more information, see the Beatles Time Capsule at www.rockument.com.

Android Mobile Programming ….. Launcher Icon

7 Launcher Icon

- Each app you create with Android Studio starts wit
  a default launcher icon,
- Launcher icons are also called app icons or product icons,
- You can change the default icon either with image, clipart or text,
- Change also the foreground and background part of icon
- Image- you need to design them with .png format, clipart- are built in symbols to select,

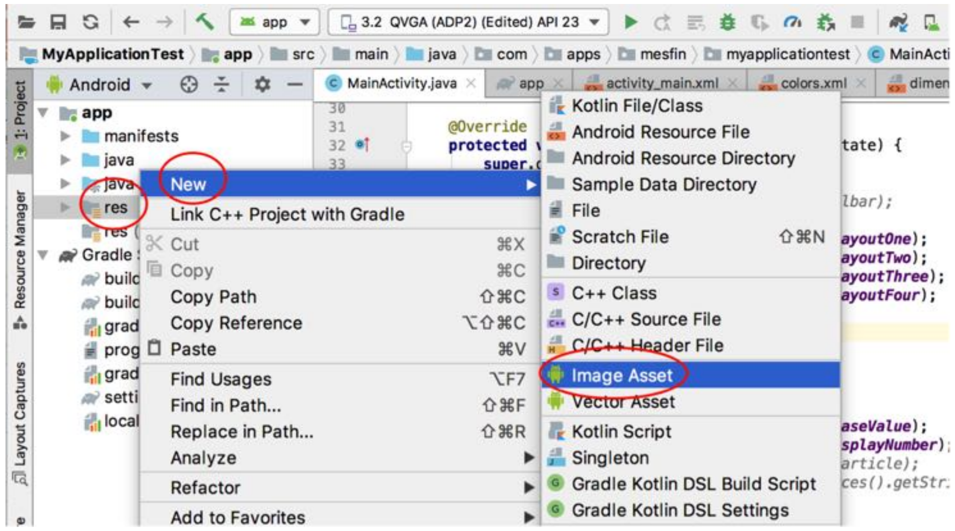


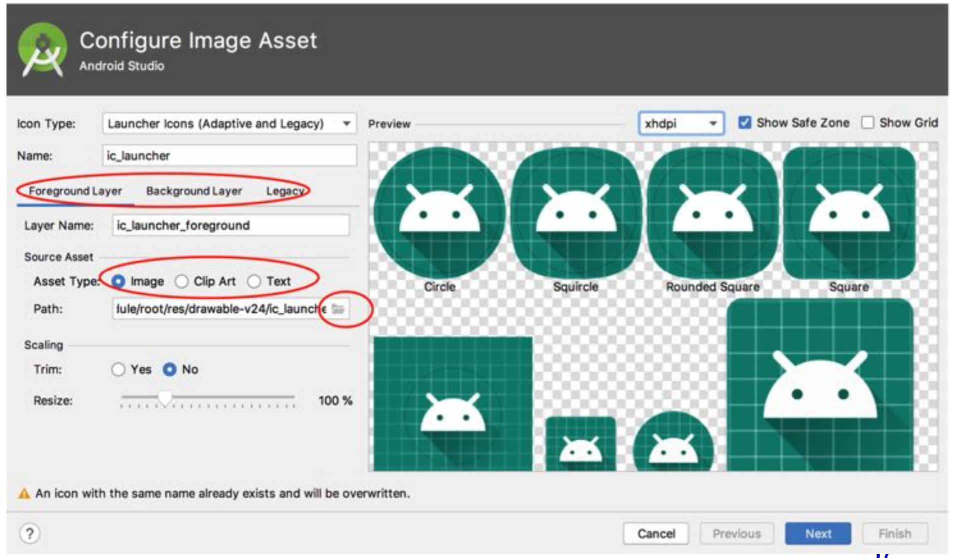


ClipArt

Android Mobile Programming ….. Launcher Icon

- Select res folder from folder section, the select new then image asset, the configure image asset dialog will appears,
- Once the image asset dialog is opened change the different options exist, like rename the icon, select from image or from clipart, resize the icon, change the background color/image, etc.
- Once parameter set, you click next and finalize the icon creation




Configure Image Asset
Android Studio

Icon Type: Launcher Icons (Adaptive and Legacy)
Name: ic_launcher
Foreground Layer | Background Layer | Legacy
Layer Name: ic_launcher_foreground
Source Asset
Asset Type: [✓ Image] Clip Art Text
Path: lule/root/res/drawable-v24/ic_launche
Scaling
Trim: Yes [✓ No]
Resize: 100 %

Preview: xhdpi [✓ Show Safe Zone] Show Grid

An icon with the same name already exists and will be overwritten.

Cancel | Previous | Next | Finish

Android Mobile Programming ….. Launcher Icon

- Change the Launcher Icon (exercise)
  - Design your own .png image for your App,
  - Make the size 512x512 pixel,
  - Follow the procedure how to change launcher icon, and
  - Display the final result