Android Project Documentation For M04_GUI_01

Author: Isaac Lohnes
Date: Oct 1 / 2018

Table of Contents

1. Project Overview

Create second activity displaying various information about device, and practice working with a second gui/guis in android studio.

2. Project Requirements

Make one Android Application that supports the 4 different Android screen densities....(hdpi, mdpi, xhdpi, xxhdpi).

Mix up the testing on emulators as follows:

-test on 3 different screen sizes
-2 different API-levels (23, 22, 21, ...any 2 different)
-at least one is Landscape, at least one is Portrait.
-You decide exactly which sizes and API levels to use, but it needs to be detectable per the next section.

...with the following features...

-Use a button on a "1st activity" to launch a "2nd activity" (similar to earlier assignment).
-In the 2nd activity, display the particulars of the screen size, platform, and API versions (dpi, API version, ...).
-For one screen size, detect the size, and display text "This is Screen Size of Activity #1".
-For the other screen size, detect the size, display the text "This is Screen Size of Activity #2".
-Create a graphic (to display on the 1st activity) with your name that changes density to match the screen (similar to my example).
-Create your own mipmap/launcher.png file to match the different screen densities.
Note: as implied, you make your own graphic and launcher, and make your own project from scratch (i.e. use my sample project as a guide, but make your own artifacts, code, graphics, ...).

2.1. Derived Requirements

Student has a deeper understanding of how auto generated XML works and activities in android development. Student also is required to have more hands on time with IDE, to better understand concepts presented.

3. Android Design Plans

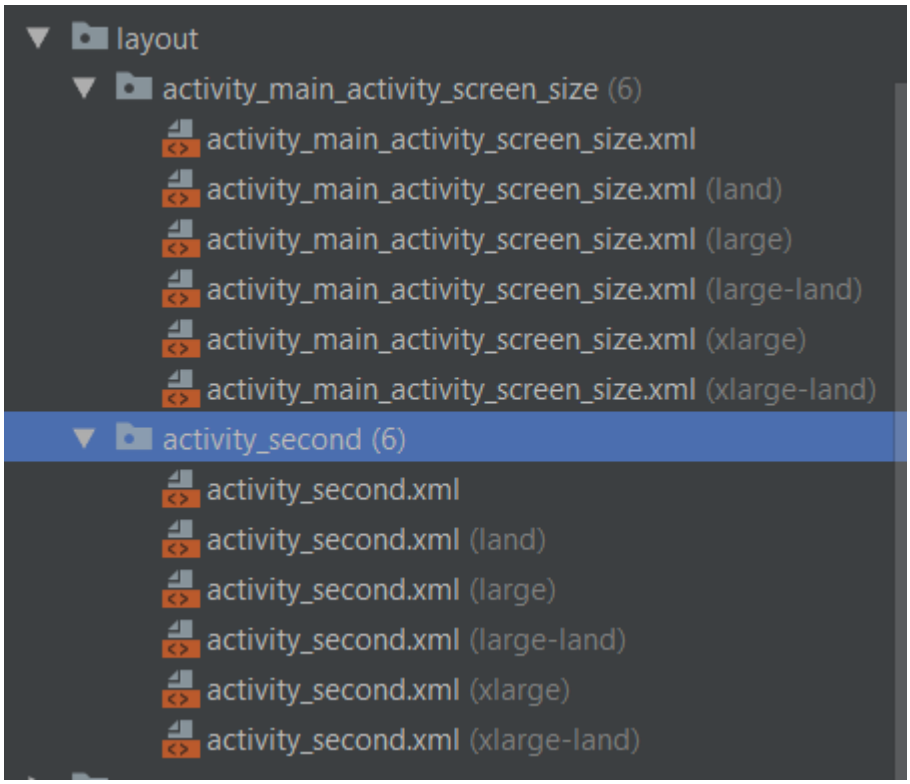
3.1. Manifest

No changes in manifest will be required for this project.

3.1.1. XML

XML will need to be generated in a similar pattern to activity_main, for our second activity.

Below is activity_second XML files, which show the same support for landscape and size as their activity_main counterparts.

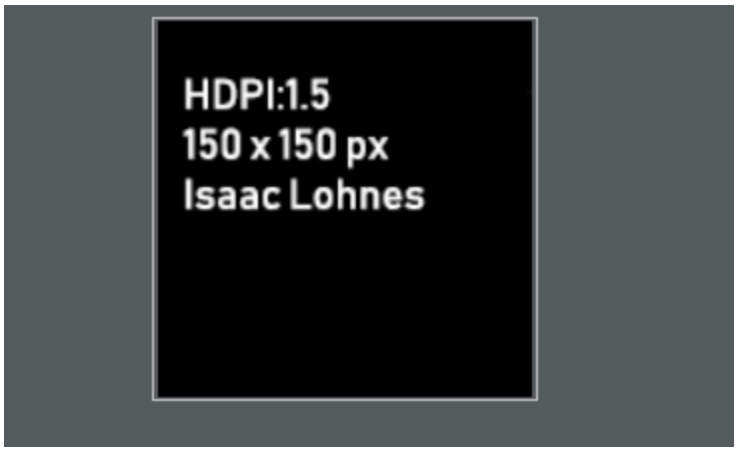


3.1.2. Mipmap and Drawable.

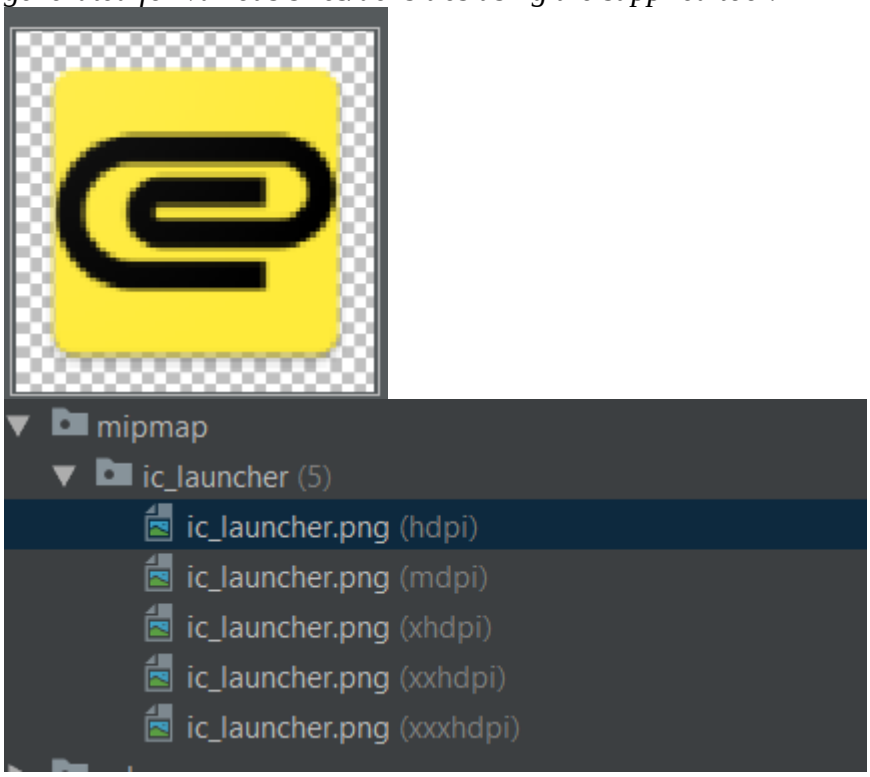
New launcher.png files will need to be generated using supplied auto generator. As well, current drawable files will need to be updated.

Below is an example of the new drawable file:

Subject



As well, shown below is the new ic_launcher.png, and a file structure showing it has been generated for various sizes/densities using the supplied tool.



3.2. Java Design

A second activity is required to meet project specifications, and this activity requires methods to populate a text field with information, as well as a separate function to get the dimensions to populate into the text field.

As well, the main activity needs to be modified to load the second activity correctly. Below is a snip showing the modification.

```
public void sendMessage(View view){
    Log.v( tag: "MainActivityScreenSize",  msg: "Test");
    Intent intent = new Intent( packageContext: this, secondActivity.class);
    Log.v( tag: "MainActivityScreenSize",  msg: "Test");
```

Below is the oncreate method that populates a text view with information for our second activity.

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

    String versionNum = Integer.toString(Build.VERSION.SDK_INT);
    Boolean afterKitKat = (Build.VERSION.SDK_INT >= Build.VERSION_CODES.KITKAT);

    // Put it on the screen
    TextView t = (TextView) findViewById(R.id.textView);
    t.setText(" Version Number is " + versionNum);
    t.append("\n afterKitKat = " + afterKitKat.toString());
    t.append("\n Build.VERSION.RELEASE = " + Build.VERSION.RELEASE);
    t.append("\n Build.VERSION.INCREMENTAL = " + Build.VERSION.INCREMENTAL);
    // dump build and display metrics
    t.append("\n\nThis is the screen size of Activity #2" + this.getSizeName( context: this));
    t.append("\n Build.DISPLAY = " + Build.DISPLAY.toString());
    t.append("\n getDisplayMetrics().densityDpi = " + getResources().getDisplayMetrics().densityDpi);
    t.append("\n getDisplayMetrics().density = " + getResources().getDisplayMetrics().density);
    t.append("\n getDisplayMetrics().xdpi = " + getResources().getDisplayMetrics().xdpi);
    t.append("\n getDisplayMetrics().ydpi = " + getResources().getDisplayMetrics().ydpi);
    t.append("\n getDisplayMetrics().heightPixels = " + getResources().getDisplayMetrics().heightPixels);
    t.append("\n getDisplayMetrics().widthPixels = " + getResources().getDisplayMetrics().widthPixels);
}
```

Also, our second activity has the same method as the first activity to get information about the current device:

Subject

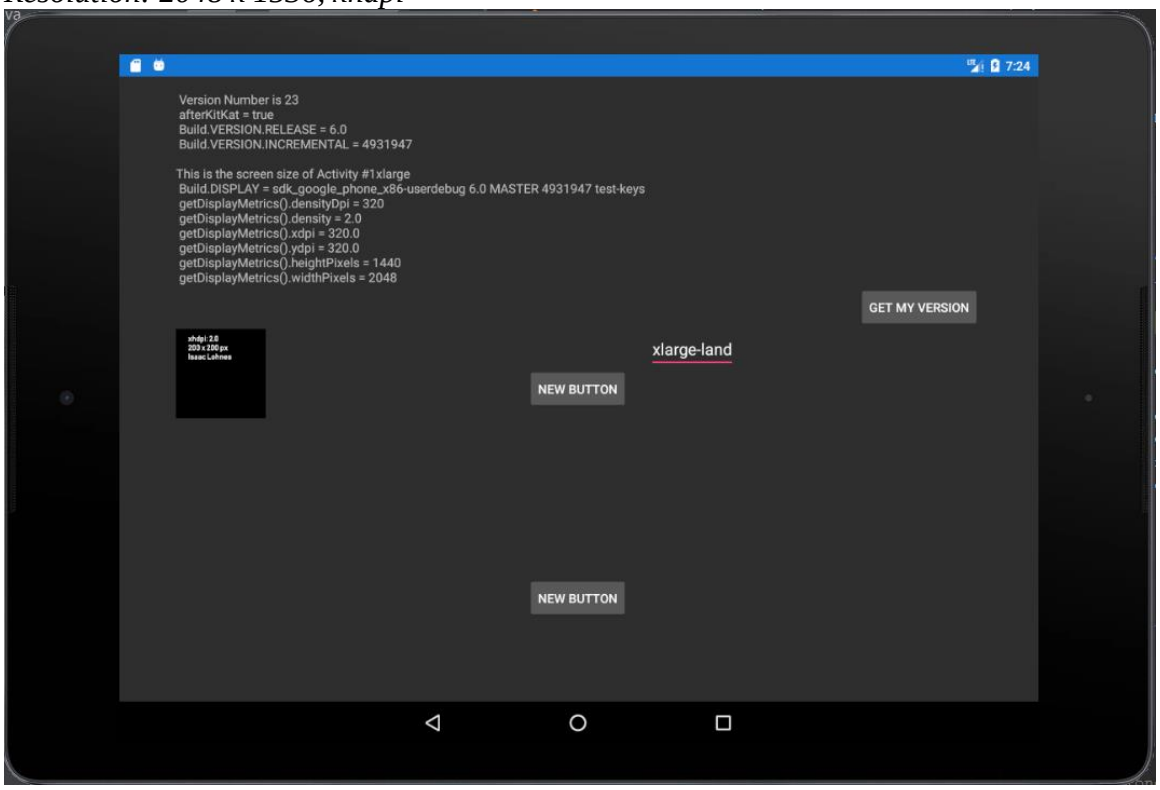
```
private static String getSizeName(Context context) {
    int screenLayout = context.getResources().getConfiguration().screenLayout;
    screenLayout &= Configuration.SCREENLAYOUT_SIZE_MASK;

    switch (screenLayout) {
        case Configuration.SCREENLAYOUT_SIZE_SMALL:
            return "small";
        case Configuration.SCREENLAYOUT_SIZE_NORMAL:
            return "normal";
        case Configuration.SCREENLAYOUT_SIZE_LARGE:
            return "large";
        case 4: // Configuration.SCREENLAYOUT_SIZE_XLARGE is API >= 9
            return "xlarge";
        default:
            return "undefined";
    }
}
```

4. Testing

Testing is split up into 3 devices.

4.1. Nexus 9

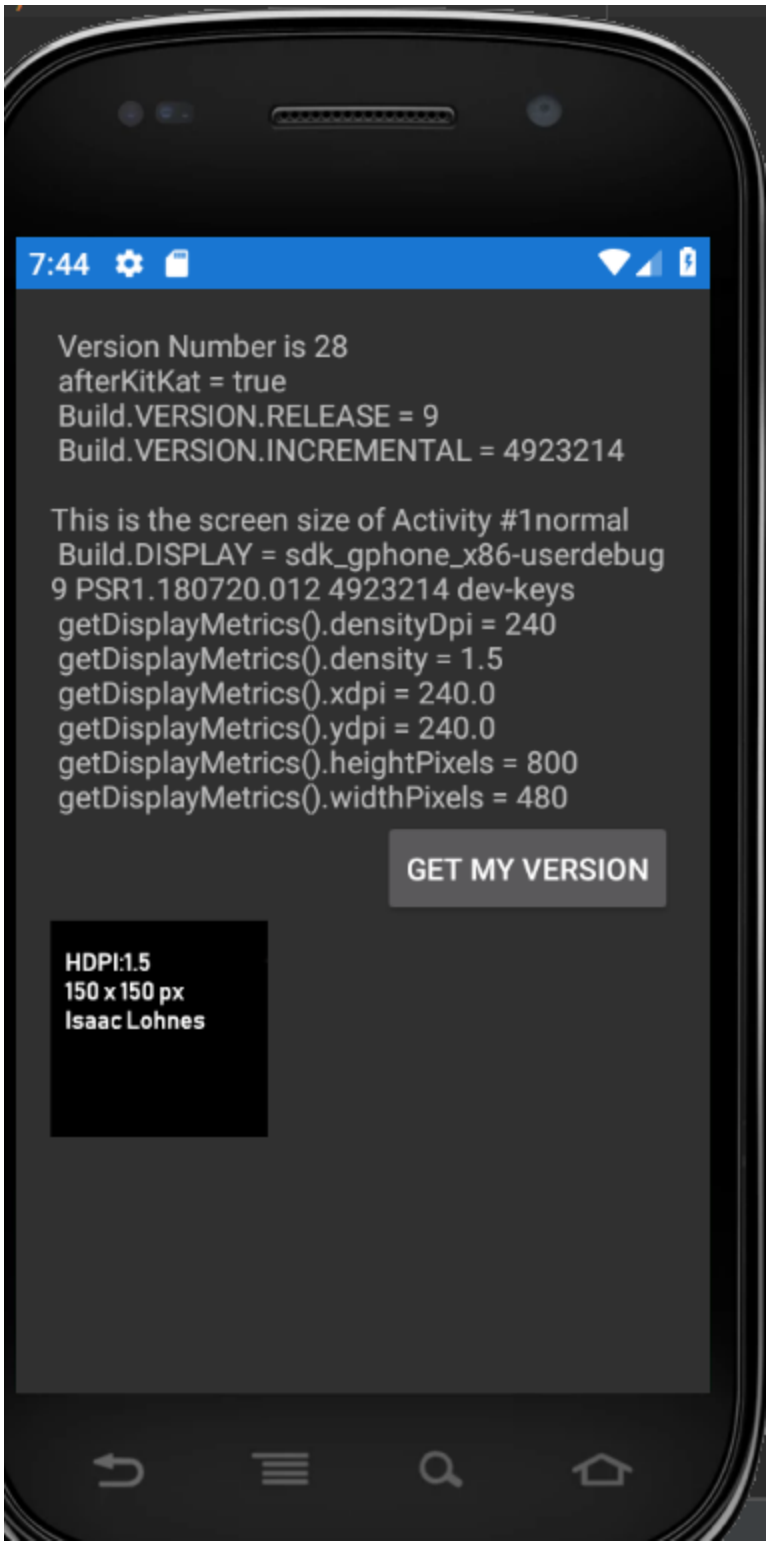
API: 23
Orientation: Landscape
Resolution: 2048 x 1536, xhdpi



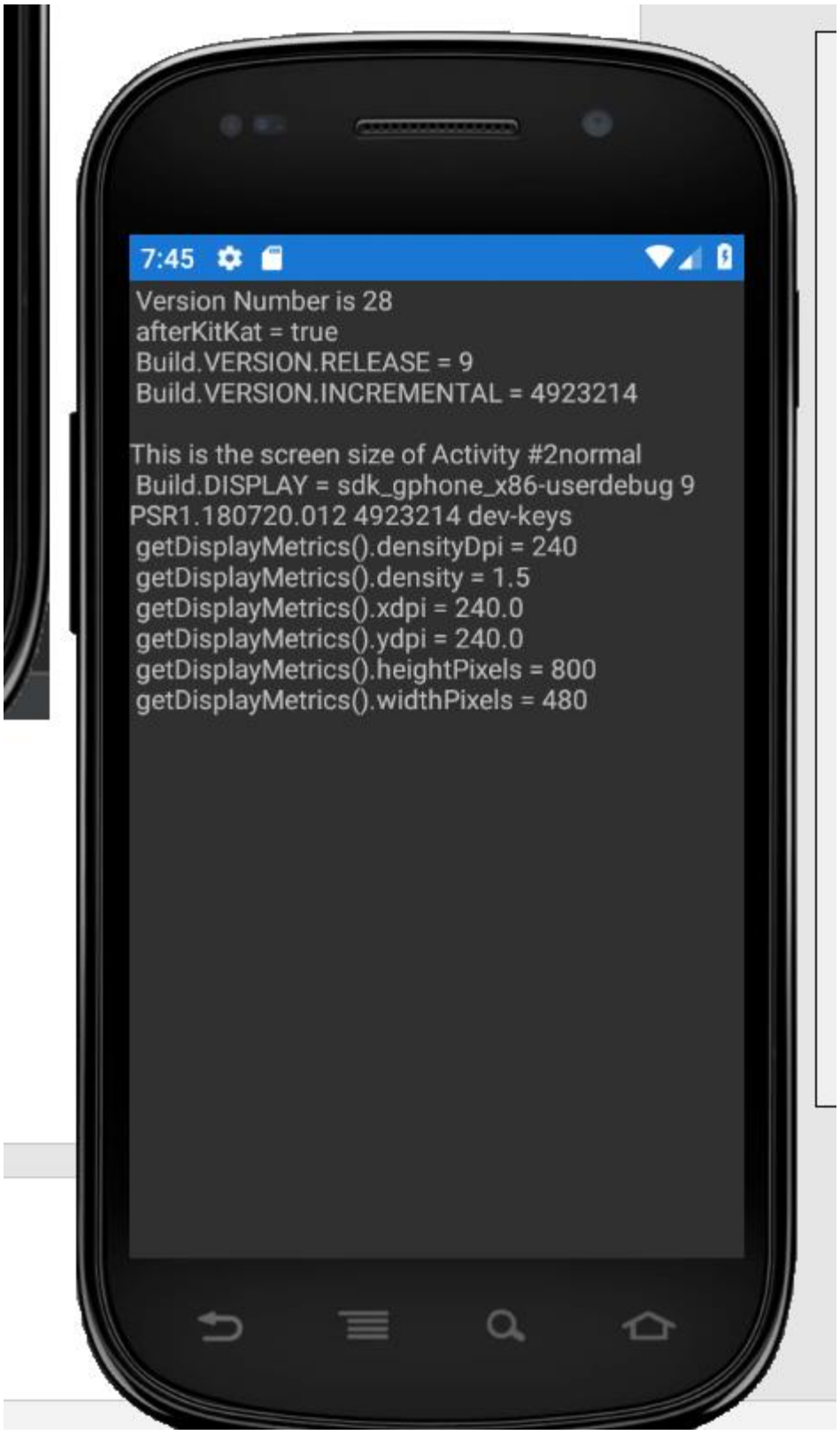
4.2. Nexus S

API : 28
Orientation: Portrait
Resolution: 480 x 800, hdpi

Subject


7:44
Version Number is 28
afterKitKat = true
Build.VERSION.RELEASE = 9
Build.VERSION.INCREMENTAL = 4923214
This is the screen size of Activity #1normal
Build.DISPLAY = sdk_gphone_x86-userdebug
9 PSR1.180720.012 4923214 dev-keys
getDisplayMetrics().densityDpi = 240
getDisplayMetrics().density = 1.5
getDisplayMetrics().xdpi = 240.0
getDisplayMetrics().ydpi = 240.0
getDisplayMetrics().heightPixels = 800
getDisplayMetrics().widthPixels = 480
GET MY VERSION
HDPI:1.5
150 x 150 px
Isaac Lohnes

Subject


7:45
Version Number is 28
afterKitKat = true
Build.VERSION.RELEASE = 9
Build.VERSION.INCREMENTAL = 4923214
This is the screen size of Activity #2normal
Build.DISPLAY = sdk_gphone_x86-userdebug 9
PSR1.180720.012 4923214 dev-keys
getDisplayMetrics().densityDpi = 240
getDisplayMetrics().density = 1.5
getDisplayMetrics().xdpi = 240.0
getDisplayMetrics().ydpi = 240.0
getDisplayMetrics().heightPixels = 800
getDisplayMetrics().widthPixels = 480

4.3. Pixel 2

API: 25
Orientation: Portrait;
Resolution: 1080 x 1920, 420dpi

Subject


7:38
Version Number is 25
afterKitKat = true
Build.VERSION.RELEASE = 7.1.1
Build.VERSION.INCREMENTAL = 4729347
This is the screen size of Activity #2normal
Build.DISPLAY = sdk_google_phone_x86-userdebug 7.1.1
NYC 4729347 test-keys
getDisplayMetrics().densityDpi = 420
getDisplayMetrics().density = 2.625
getDisplayMetrics().xdpi = 420.0
getDisplayMetrics().ydpi = 420.0
getDisplayMetrics().heightPixels = 1794
getDisplayMetrics().widthPixels = 1080