

My Collage App

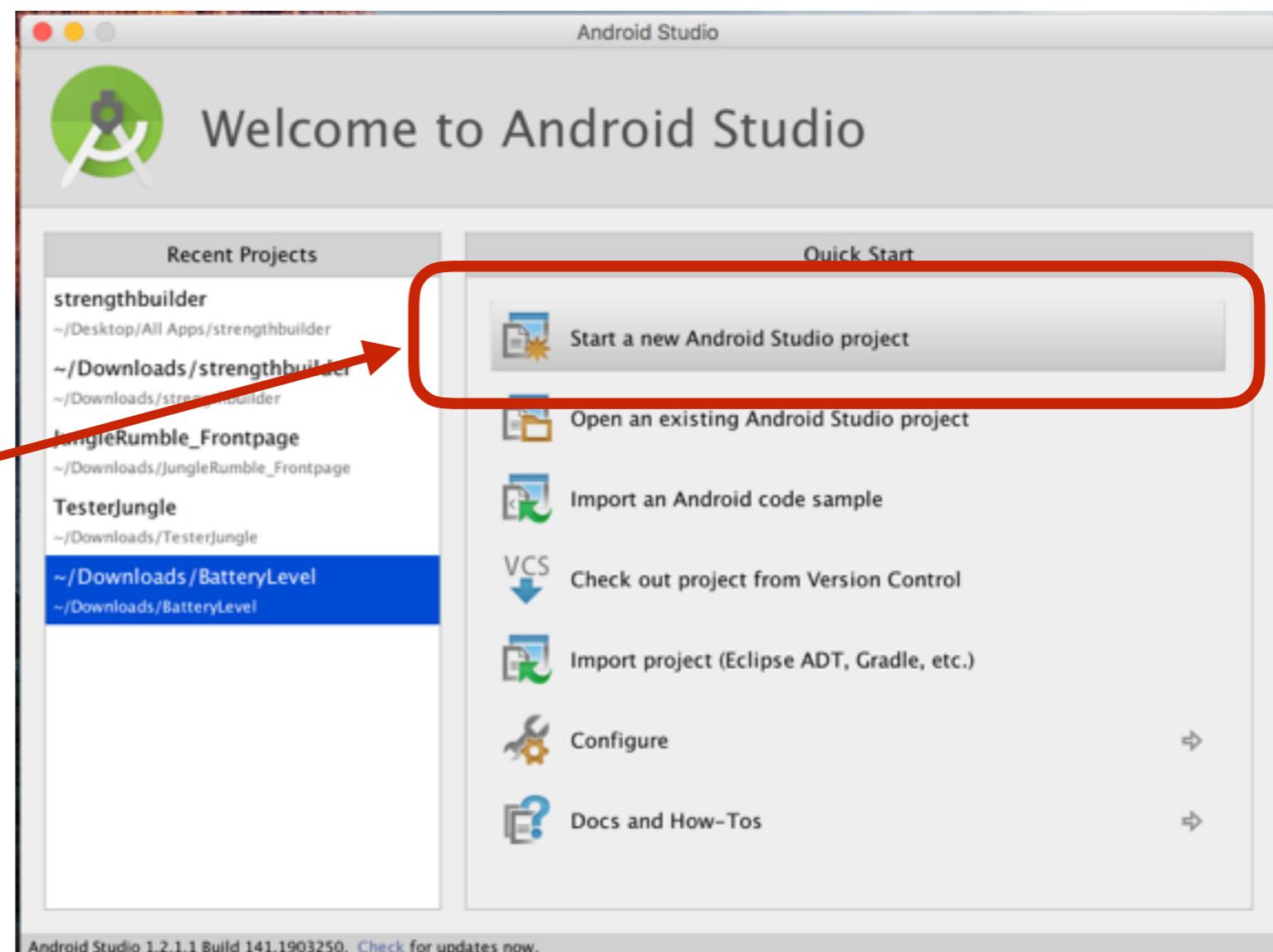
Android Studio

Let's build an app together

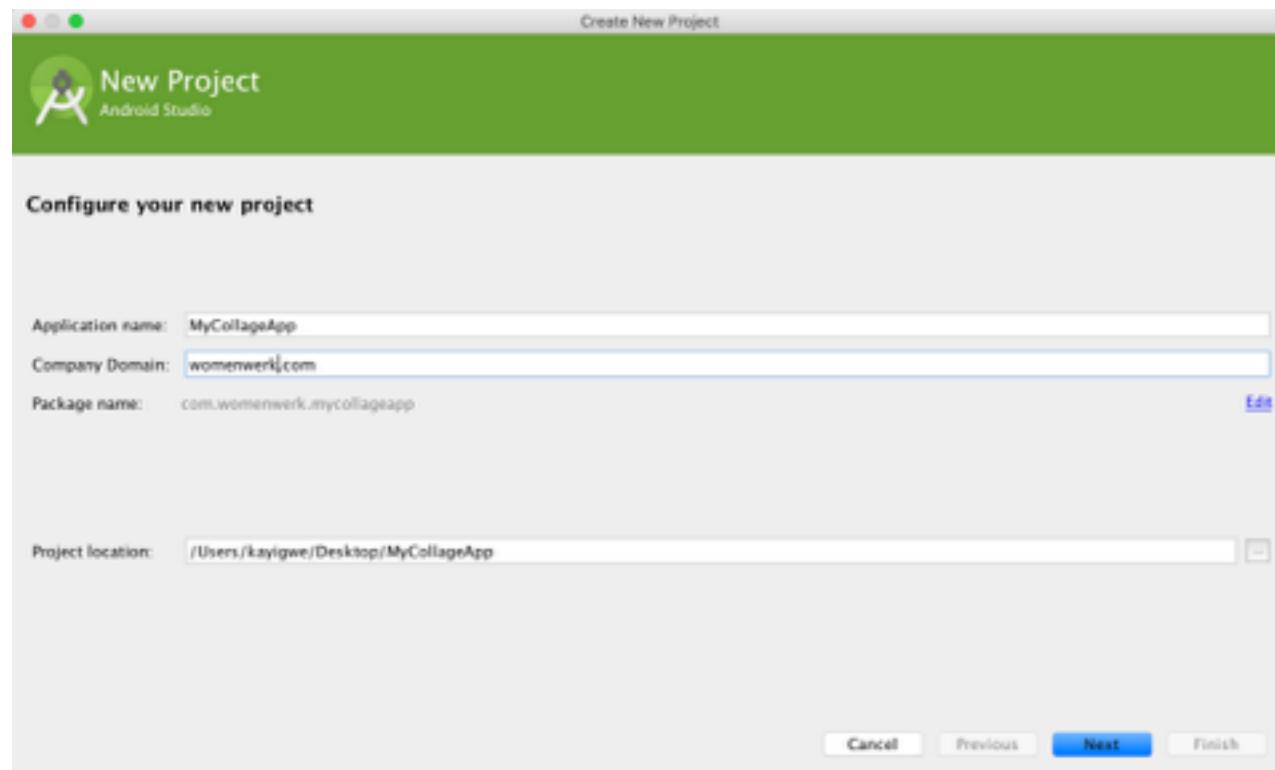


Android Studio

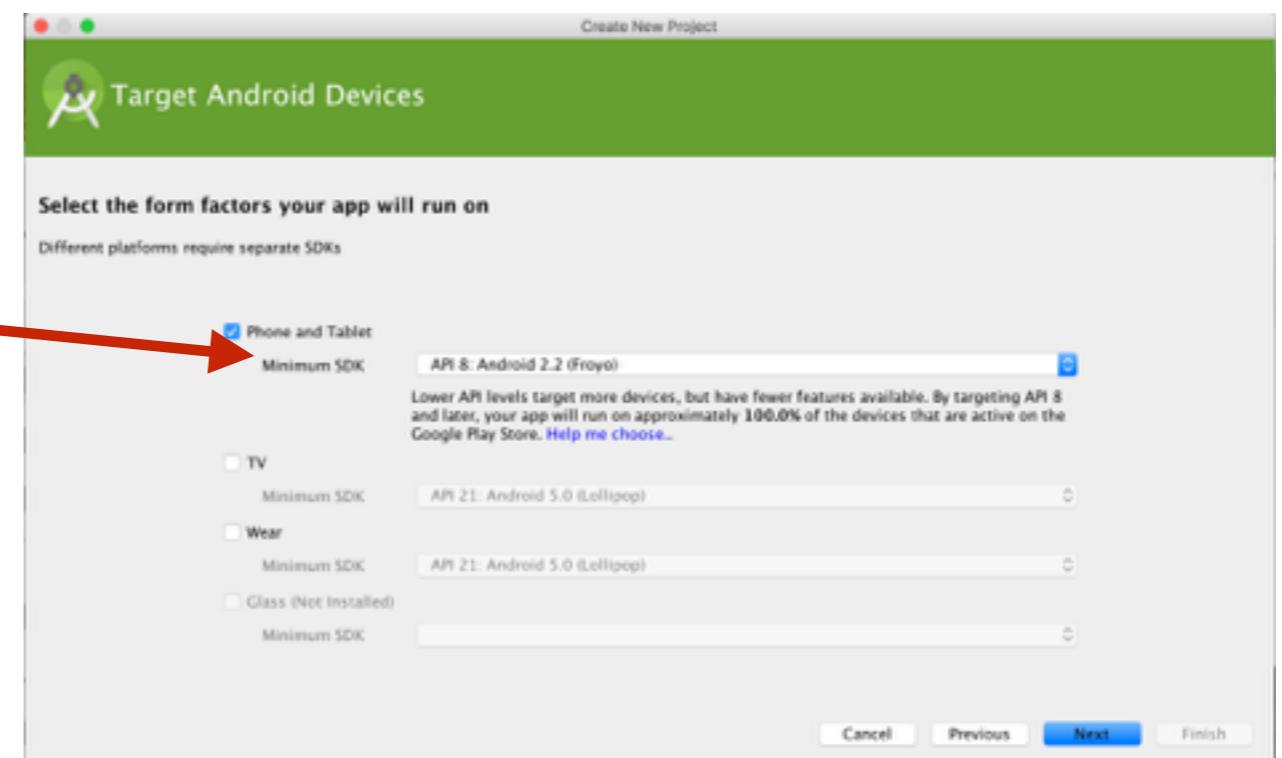
- Open Android Studio
- You should see the screen on the right
- Click the “Start a new Android Studio project” button



Android Studio



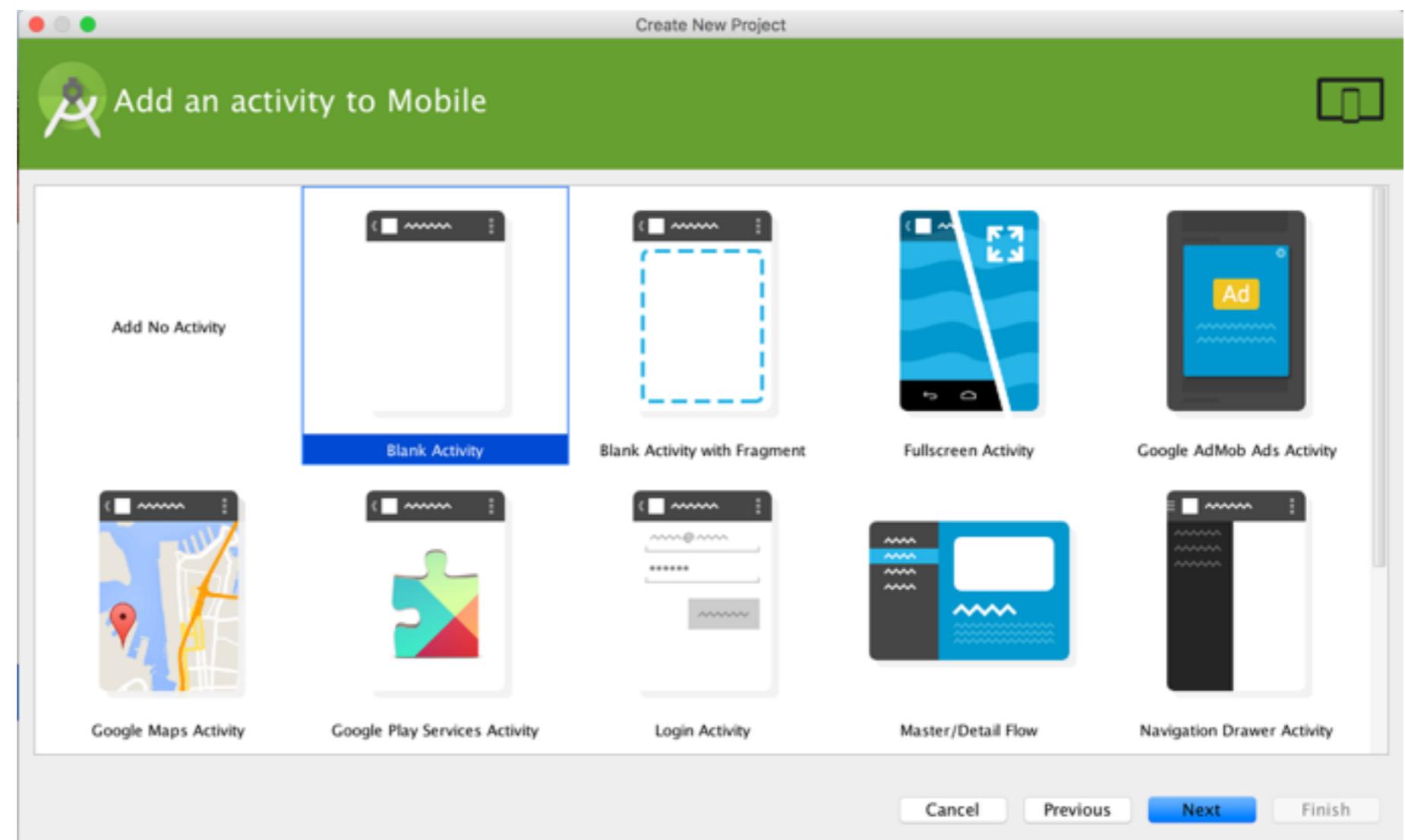
- Type “MyCollageApp” into the Application name area
- In the Company Details section, type “womenwerk.com”
- If you are ok with the file path, leave the 3rd section as is
- Click Next



- Leave this screen as is
- Click Next

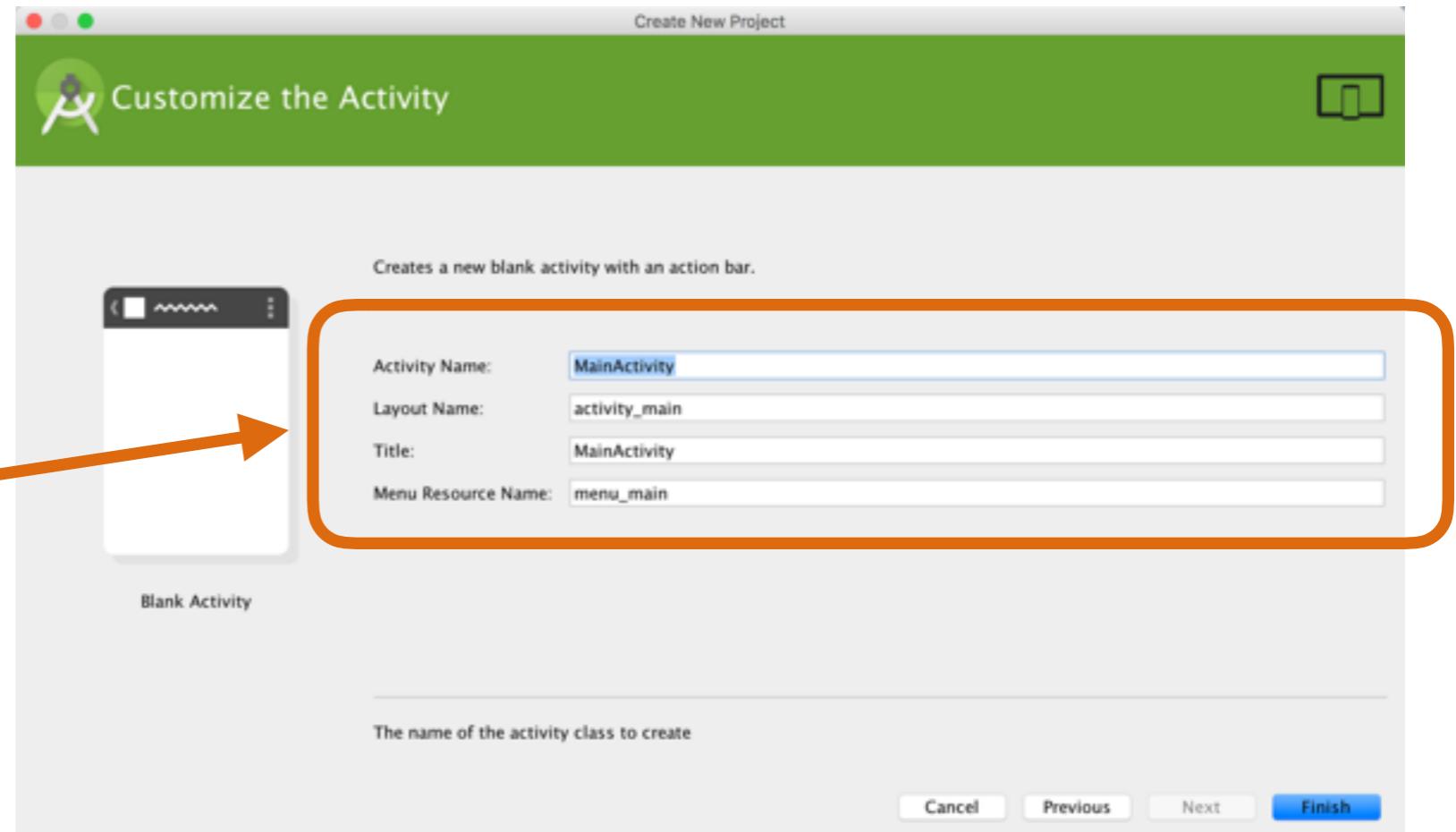
Android Studio

- Click the “Blank Activity” option
- Click Next



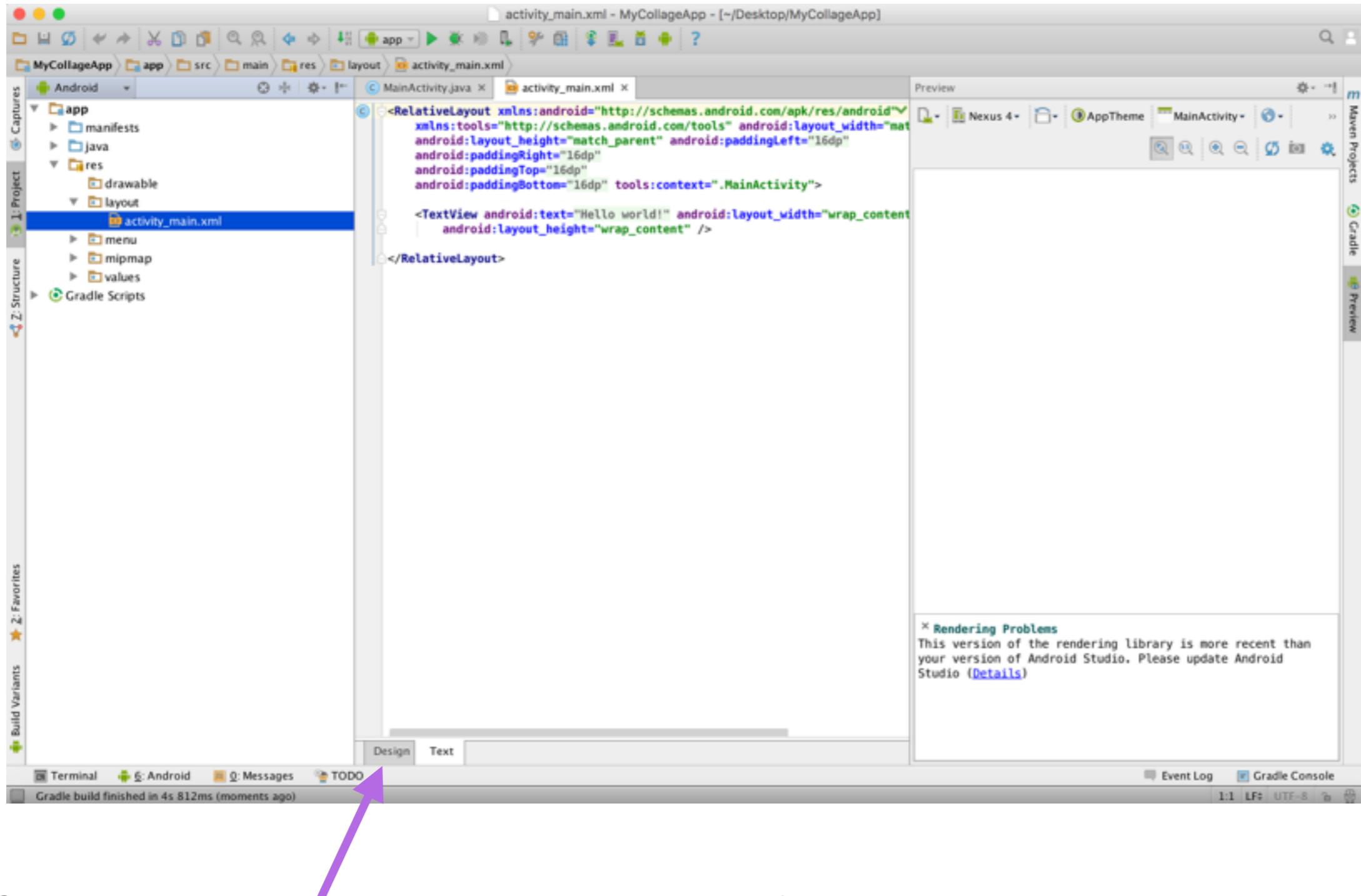
Android Studio

- Once you arrive to this screen, make sure everything is as follows...
- Click the Finish button



Android Studio

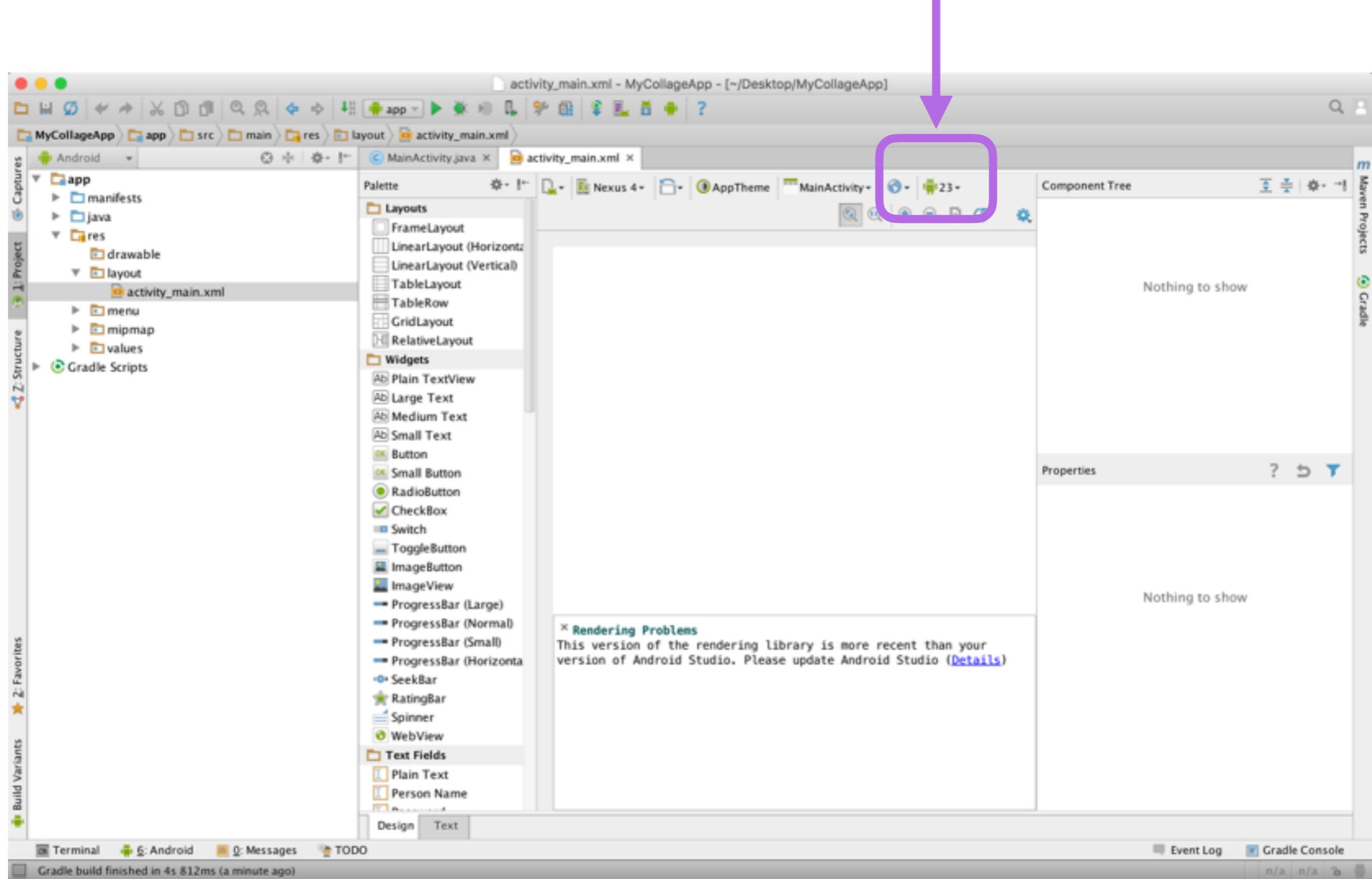
- Your screen should now look like or similar to the screen below.



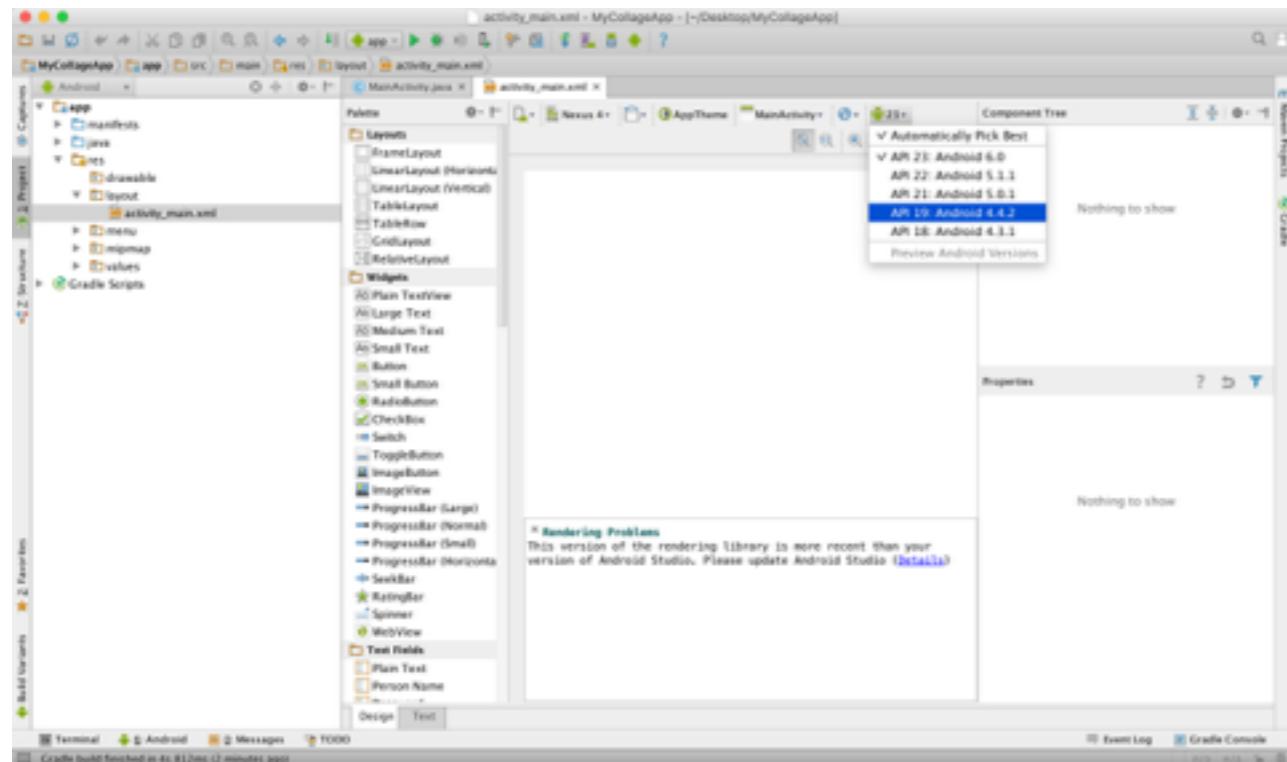
- Click the “Design” tab at the bottom of the screen

Android Studio

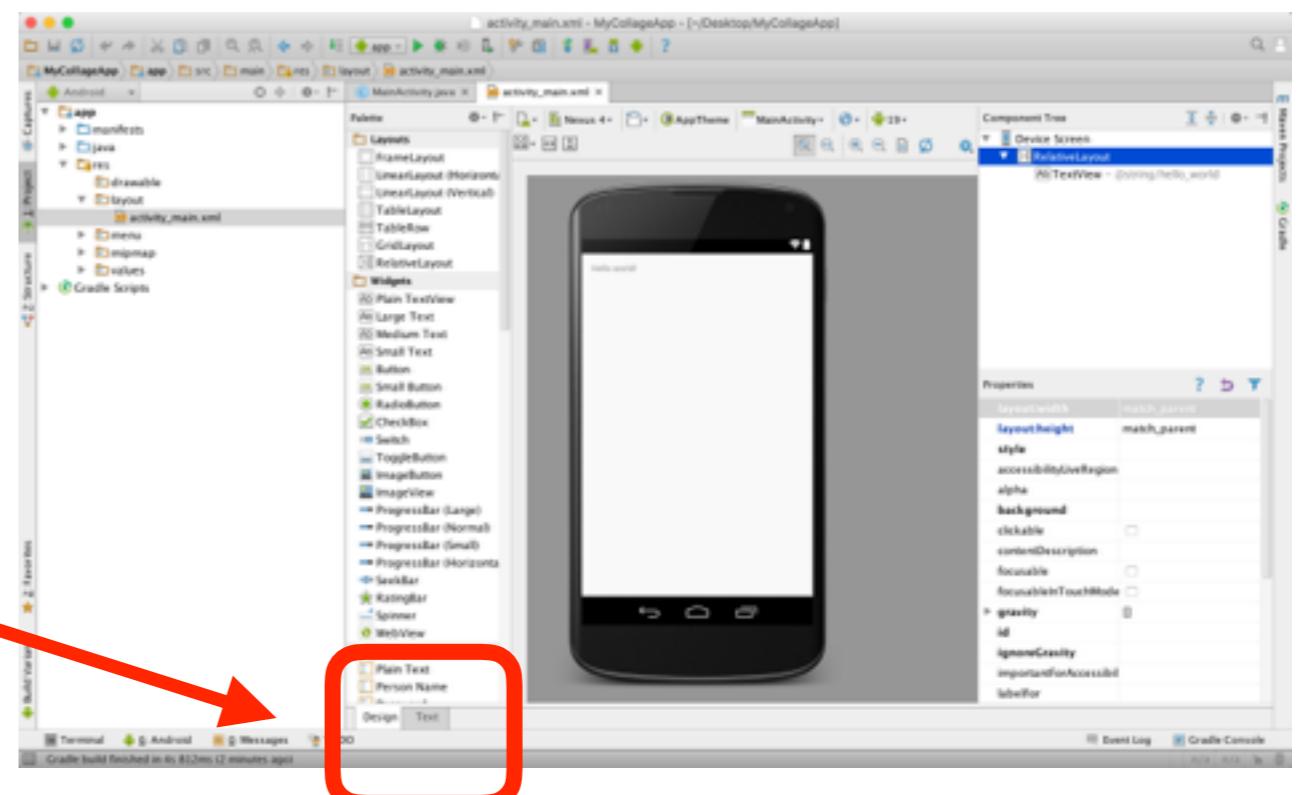
- If you see this render error, click on the Android Version button



Android Studio



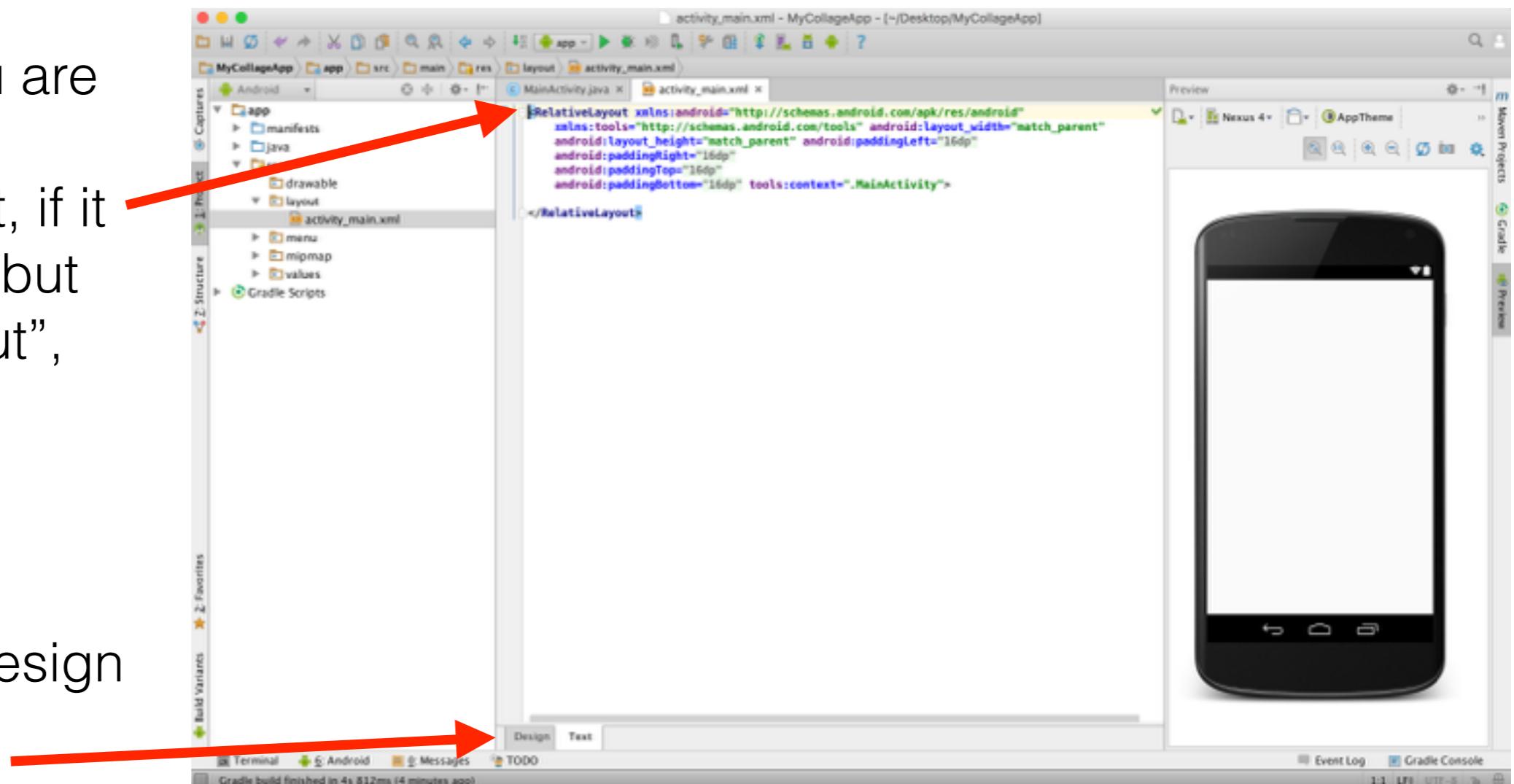
- Click on API 19



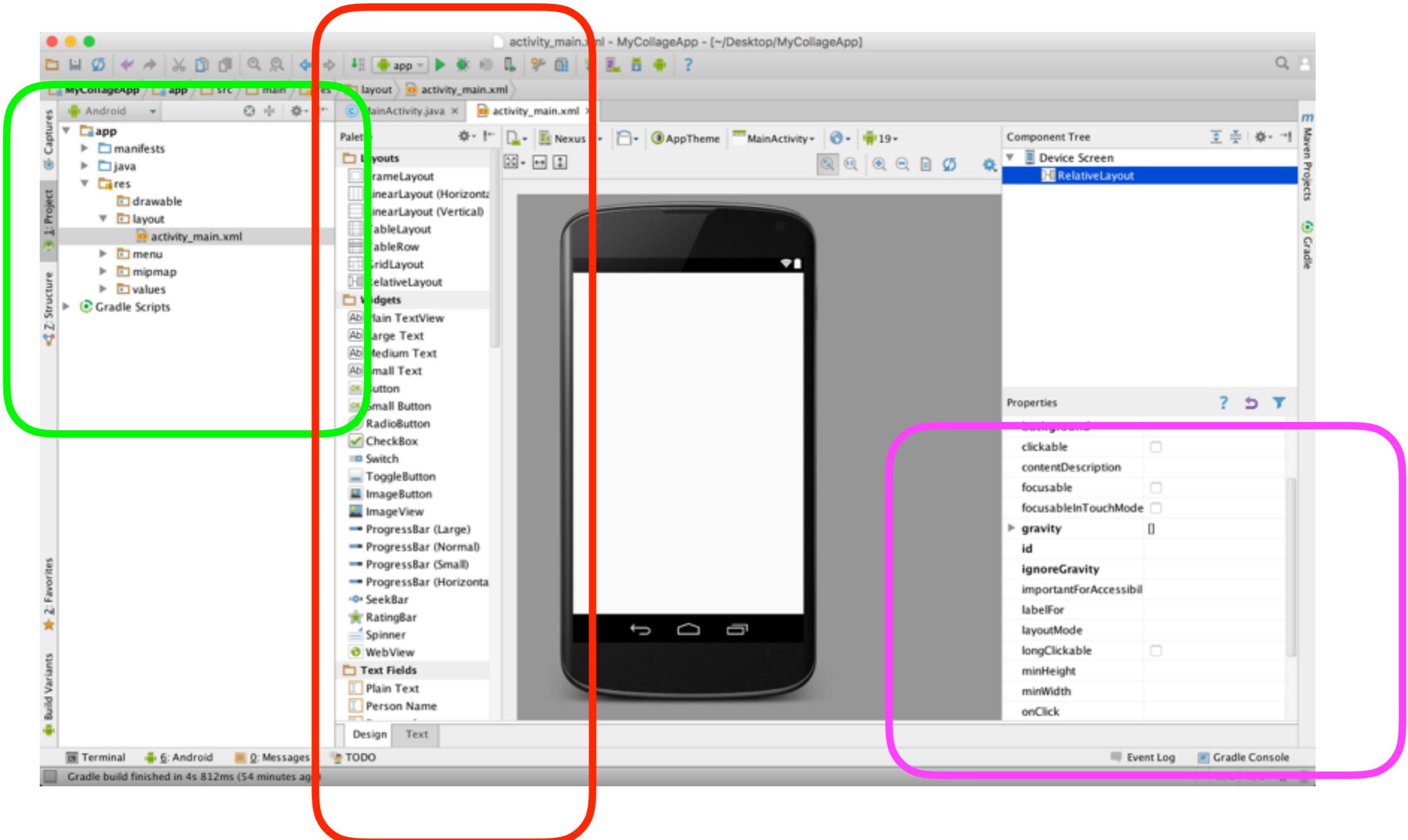
- The picture of the phone should render now
- Click on the “Text” button

Android Studio

- Make sure you are working with a `RelativeLayout`, if it says anything but “`RelativeLayout`”, type it in



Android Studio Design Anatomy

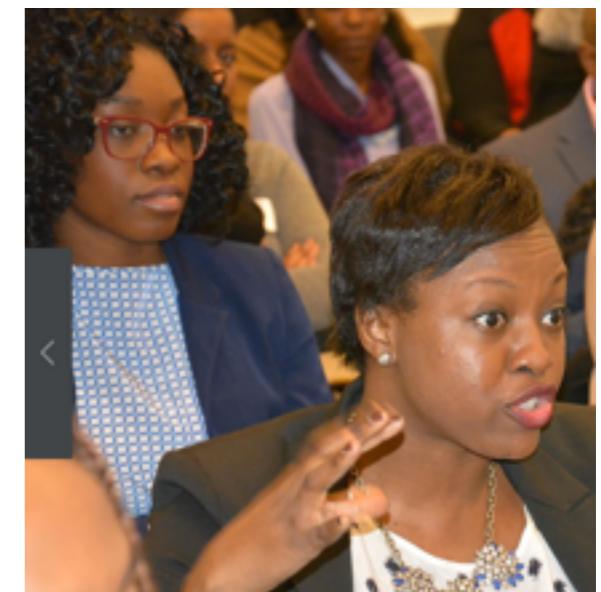


Android Studio Images



- MyCollageApp

- app
 - src
 - main
 - res

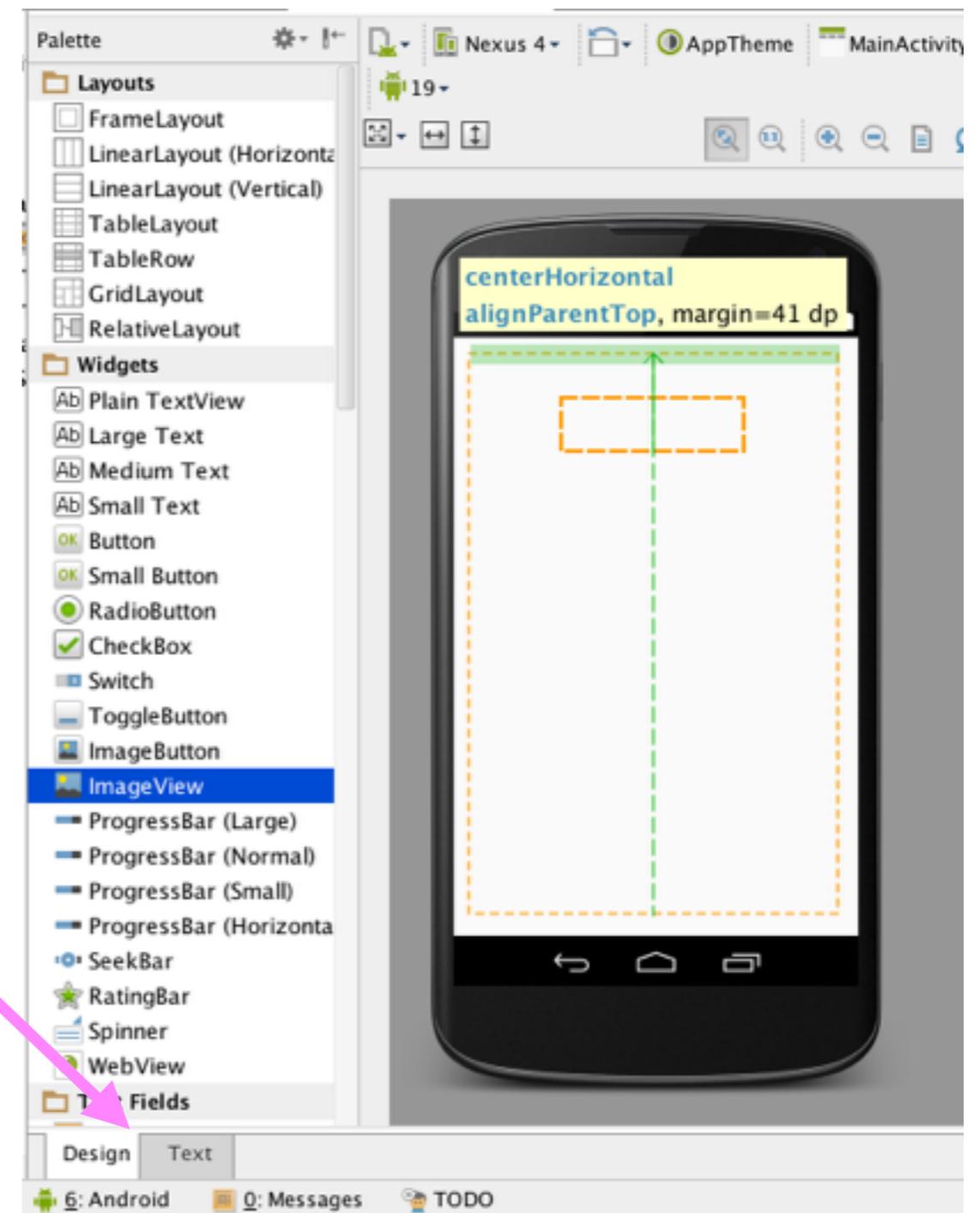


- drawable (place images in the drawable folder)



Android Studio Design, xml

- Drag an ImageView onto the Android device
- Place the ImageView on the device by letting go of the mouse (or clicking on the desired placement area)
- Now click on the “Text tab at the bottom of the screen”



Android Studio Design, xml coding

```
<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">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/imageView"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="41dp" />
</RelativeLayout>
```

- Change **wrap_content** to **match_parent** in the **android:layout_width** section
- Second give the View an id, change it to: **topImage**
- Next, add an image src by typing:
android:background="@drawable/pictureone"
- Delete all instances of padding and margin in order to make images flush with the edges of the screen
- You can change the height of your picture by changing, in the **android:layout_height** section, the **wrap_content**, to "120dp" or any height you would like.

Android Studio Design, code so far...

```
<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" tools:context=".MainActivity">

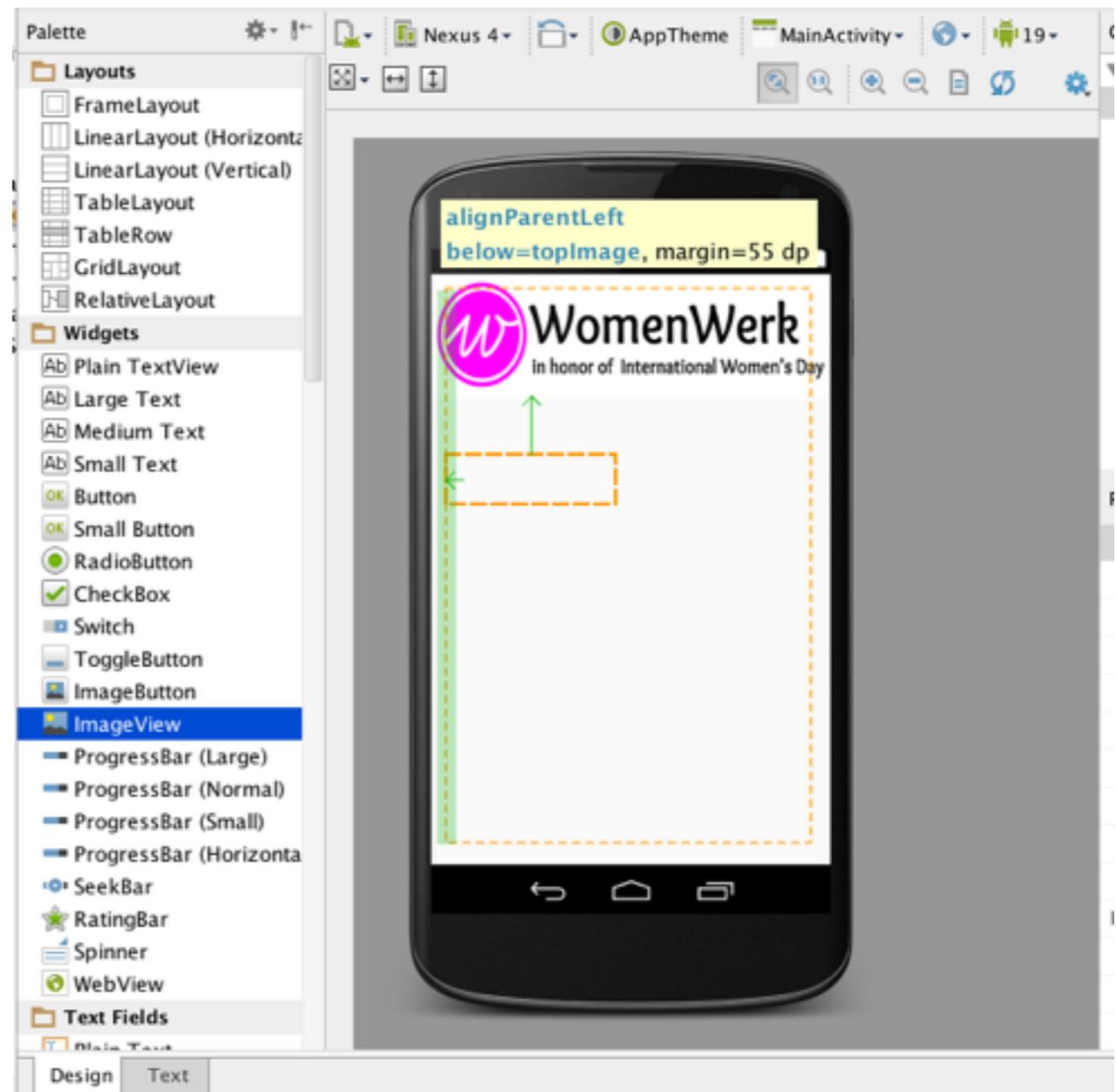
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="120dp"
        android:id="@+id/topImage"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:background="@drawable/pictureone"/>
</RelativeLayout>
```

- Your Android screen should look like this one



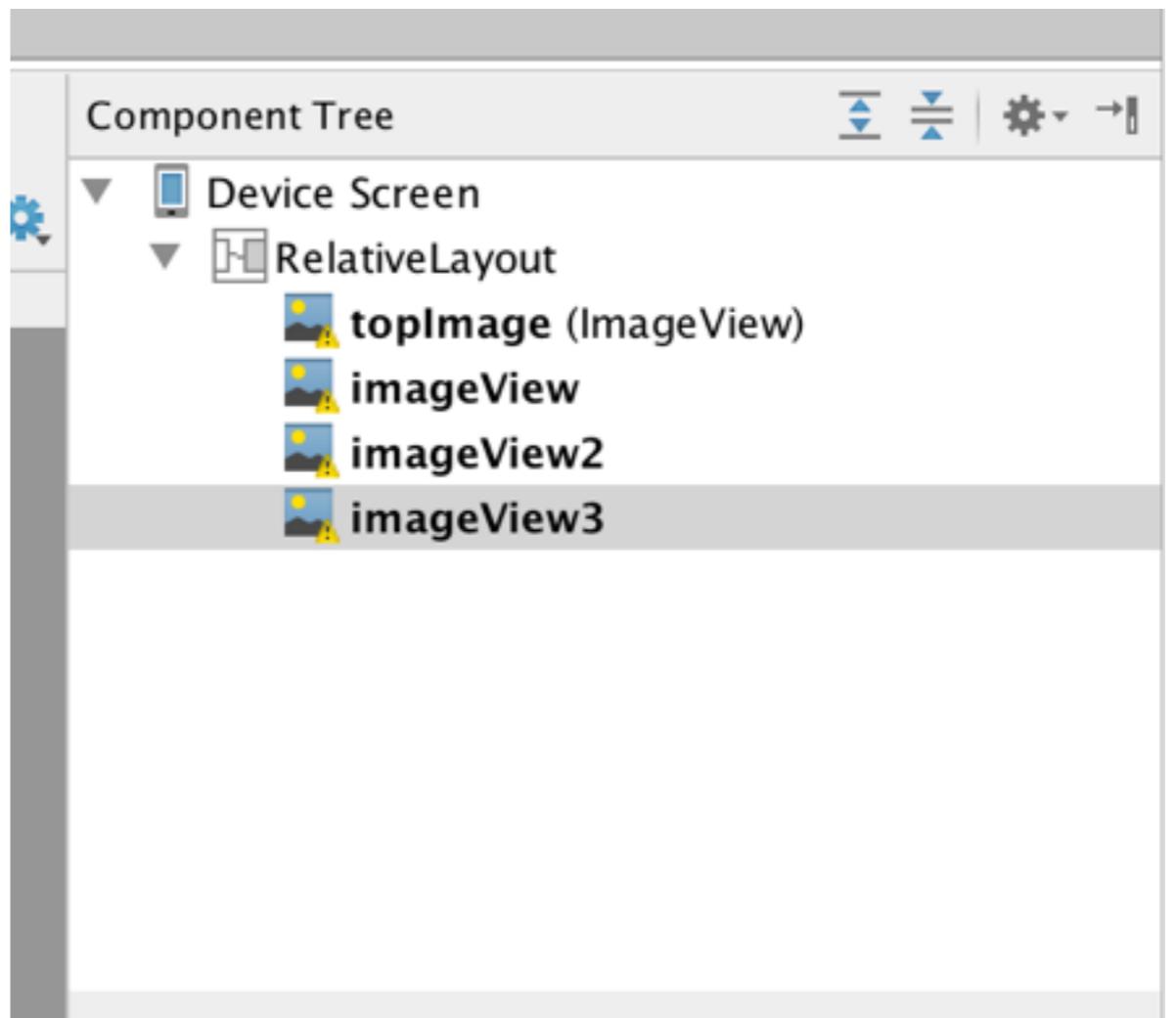
Android Studio Design, xml coding

- Go back to the Design screen and we will repeat the process again
- Drag three ImageView onto the screen.
- Next, go back to the Text screen



Android Studio Design, xml coding

- You will know if you have put 3 ImageViews on the screen if you see 4 imageViews in the screen in the upper right hand panel



Android Studio Design, xml coding

- Change the id to “**secondImage**” **to the first ImageView**
- Change the id to “**thirdImage**” **to the second ImageView**
- Change the id to “**forthImage**” **to the third ImageView**
- Make sure your code looks like the one on the right. When you change the id to secondImage, you will have to change every instance where that ImageView is mentioned

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="120dp"  
    android:id="@+id/topImage"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:background="@drawable/pictureone"/>  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/secondImage"  
    android:layout_below="@+id/topImage"  
    android:layout_alignLeft="@+id/topImage"  
    android:layout_alignStart="@+id/topImage"  
    android:layout_marginTop="67dp" />  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/thridImage"  
    android:layout_alignBottom="@+id/secondImage"  
    android:layout_alignRight="@+id/topImage"  
    android:layout_alignEnd="@+id/topImage" />  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/forthImage"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true" />
```

Android Studio Design, xml coding

- Add...
android:background="@drawable/picturetwo" to the ImageView with the “**secondImage**” id
- Add...
android:background="@drawable/picturethree" to the ImageView with the “**thirdImage**” id
- Add...
android:background="@drawable/picturefour" to the ImageView with the “**forthImage**” id
- Your code should look like the one on the right

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="120dp"  
    android:id="@+id/topImage"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:background="@drawable/pictureone"/>  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/secondImage"  
    android:layout_below="@+id/topImage"  
    android:layout_alignLeft="@+id/topImage"  
    android:layout_alignStart="@+id/topImage"  
    android:layout_marginTop="67dp"  
    android:background="@drawable/picturetwo"/>  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/thridImage"  
    android:layout_alignBottom="@+id/secondImage"  
    android:layout_alignRight="@+id/topImage"  
    android:layout_alignEnd="@+id/topImage"  
    android:background="@drawable/picturethree"/>  
  
<ImageView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:id="@+id/forthImage"  
    android:layout_alignParentBottom="true"  
    android:layout_centerHorizontal="true"  
    android:background="@drawable/picturefour"/>
```

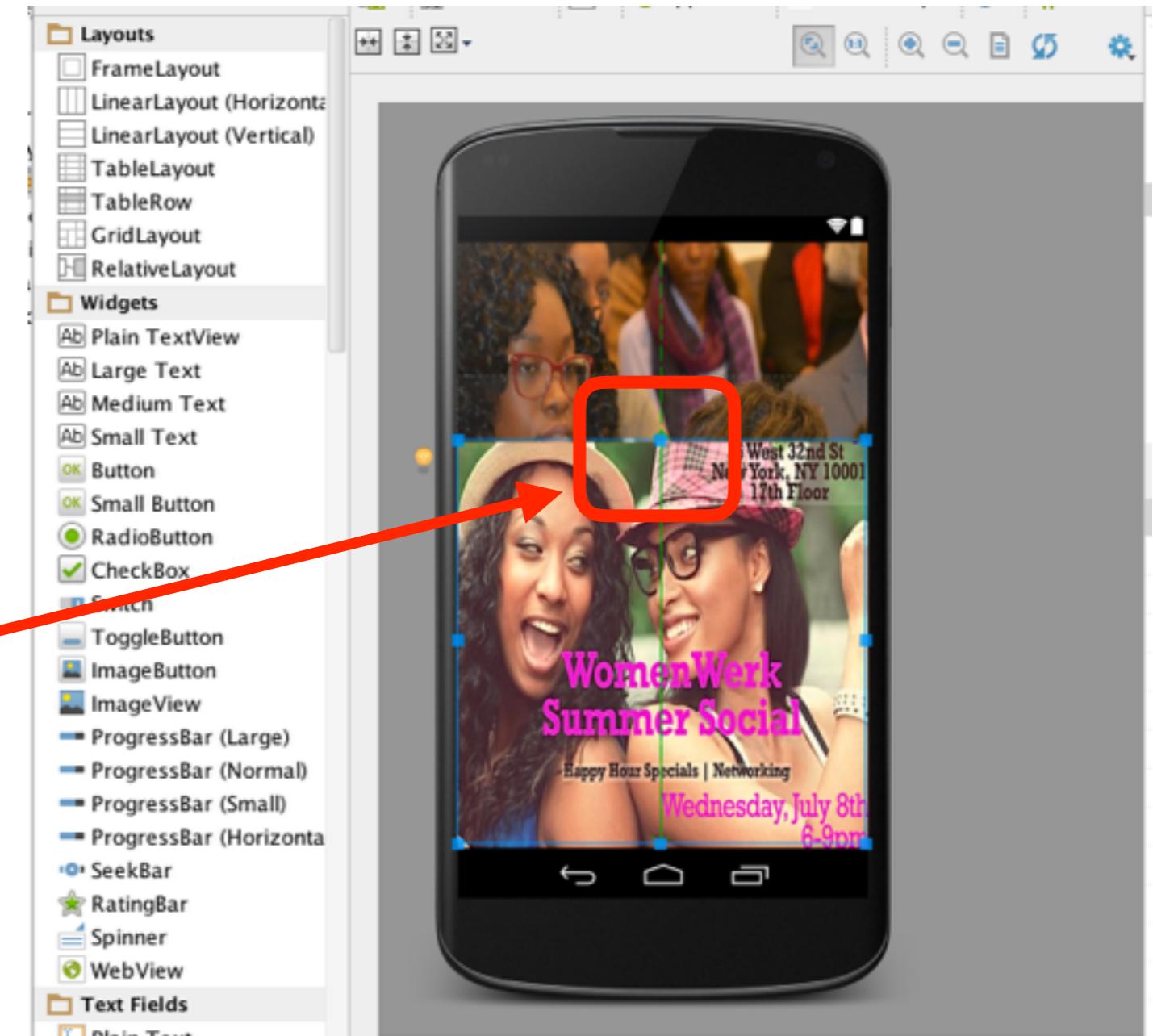
Android Studio Design, xml coding

- Switch back to the Design Screen
- Your screen may look like the one on the right (or some version of craziness), but that is OK, we will fix that next!!!



Android Studio Design, xml coding

- Click on one of the pictures and drag it to the desired size
- Repeat with all of the images until you are satisfied with how your screen looks
- This may not be enough, so switch back to the Text screen



Android Studio Design, xml coding

- Change all of the heights and widths of all of the images until you get the desired arrangement

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="120dp"  
    android:id="@+id/topImage"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:background="@drawable/pictureone"/>
```

```
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:id="@+id/secondImage"  
    android:background="@drawable/picturetwo"  
    android:layout_alignTop="@+id/thridImage"  
    android:layout_alignRight="@+id/topImage"  
    android:layout_toRightOf="@+id/thridImage"  
    android:layout_toEndOf="@+id/thridImage" />
```

```
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:id="@+id/thridImage"  
    android:background="@drawable/picturethree"  
    android:layout_above="@+id/forthImage"  
    android:layout_alignLeft="@+id/topImage"  
    android:layout_alignStart="@+id/topImage" />
```

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="250dp"  
    android:id="@+id/forthImage"  
    android:background="@drawable/picturefour"  
    android:layout_alignParentBottom="true" />
```

Android Studio Design, xml coding Final Screen



Android Studio Design, xml coding

Add Spacing between each image

- In each of the ImageView sections, add a margin to each side
- android:layout_margin=“**anything you would like**”

```
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="100dp"  
    android:id="@+id/topImage"  
    android:layout_alignParentTop="true"  
    android:layout_centerHorizontal="true"  
    android:background="@drawable/pictureone" />  
  
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:id="@+id/secondImage"  
    android:background="@drawable/picturetwo"  
    android:layout_alignTop="@+id/thridImage"  
    android:layout_alignRight="@+id/topImage"  
    android:layout_toRightOf="@+id/thridImage"  
    android:layout_toEndOf="@+id/thridImage"  
    android:layout_marginRight="6dp"  
    android:layout_marginLeft="3dp"/>  
  
<ImageView  
    android:layout_width="200dp"  
    android:layout_height="200dp"  
    android:id="@+id/thridImage"  
    android:background="@drawable/picturethree"  
    android:layout_above="@+id/forthImage"  
    android:layout_alignLeft="@+id/topImage"  
    android:layout_alignStart="@+id/topImage"  
    android:layout_marginLeft="6dp"  
    android:layout_marginRight="3dp"  
    android:layout_marginTop="6dp"/>  
  
<ImageView  
    android:layout_width="match_parent"  
    android:layout_height="250dp"  
    android:id="@+id/forthImage"  
    android:background="@drawable/picturefour"  
    android:layout_margin="6dp"  
    android:layout_alignParentBottom="true" />
```

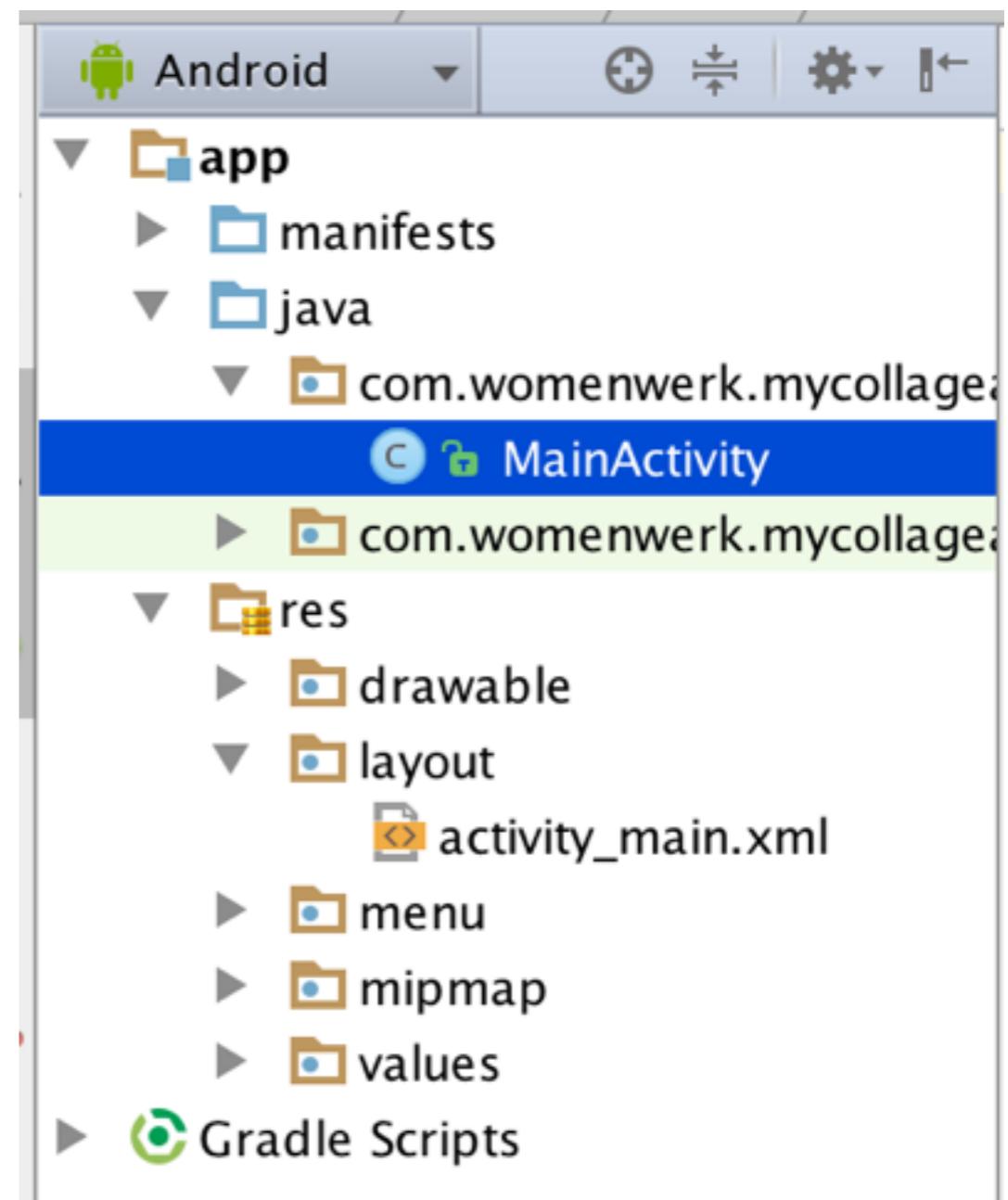
Android Studio Design, xml coding

Final Screen with margins



Now to the Java

- Now that we are done with the design, we are going to move on to the Java
- Navigate to the MainActivity.java file
- You can also click on the tab next to the activity_main.xml tab at the top



MainActivity.java

Your screen should have the code below

```
package com.womenwerk.mycollageapp;

import ...

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

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

Java File Anatomy

- libraries used to enable certain functions

```
package com.womenwerk.mycollageapp;  
  
import android.support.v7.app.ActionBarActivity;  
import android.os.Bundle;  
import android.view.Menu;  
import android.view.MenuItem;
```

- Areas where you will initialize variables, so that everything inside of the class has access to those variables

```
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  
        if (id == R.id.action_settings) {  
            return true;  
        }  
  
        return super.onOptionsItemSelected(item);  
    }  
}
```

- OnCreate Method, executes only once whenever there is a call to the method

- Area where we will be adding our own methods

Java: Initialize the ImageViews

- Add **import android.widget.ImageView;** to the import section
- Add the following code to initialize all of the ImageViews: Add this right above the OnCreate method

```
ImageView imgagViewone;  
ImageView imageViewtwo;  
ImageView imageViewthree;  
ImageView imageViewfour;
```

Java: Initialize the ImageViews

- Inside of the OnCreate method, we are going to assign an image to the ImageView variables we created earlier.

```
@Override  
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.activity_main);  
  
    imageViewone = (ImageView) findViewById(R.id.topImage);  
    imageViewone = (ImageView) findViewById(R.id.secondImage);  
    imageViewthree = (ImageView) findViewById(R.id.thirdImage);  
    imageViewfour = (ImageView) findViewById(R.id.forthImage);  
}
```

Java: Quick Terminology

- OnClickListener: Called when a view has been clicked
- Toast: A popup box that pops up when called

Java: OnClick Listeners

- Set OnClick listeners, so that when you click on an image, a popup window with information about the picture pops up
- Type the following function underneath the variable assignments
- Do this for all 4 variables

```
imageViewone.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);  
        toast.show();  
    }  
});
```

Java: Code so far

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

    imageViewone = (ImageView) findViewById(R.id.topImage);
    imageViewtwo = (ImageView) findViewById(R.id.secondImage);
    imageViewthree = (ImageView) findViewById(R.id.thirdImage);
    imageViewfour = (ImageView) findViewById(R.id.forthImage);

    imageViewone.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);
            toast.show();
        }
    });

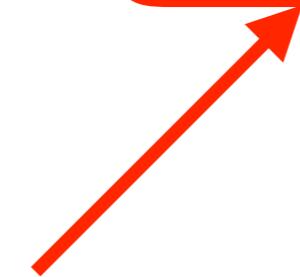
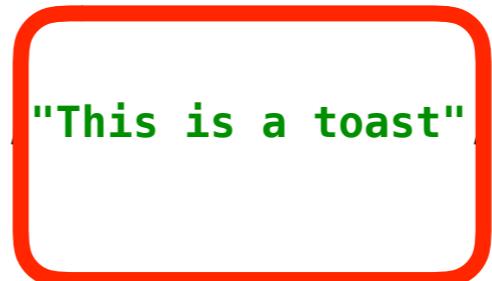
    imageViewtwo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);
            toast.show();
        }
    });

    imageViewthree.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);
            toast.show();
        }
    });

    imageViewfour.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);
            toast.show();
        }
    });
}
```

Java: Toast Message

```
Toast toast = Toast.makeText(getApplicationContext(), "This is a toast", Toast.LENGTH_LONG);
```



- Type in a description for each of the images
- Remember, this is the description that will pop up every time you press the image

Java: Code

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

    imageViewone = (ImageView) findViewById(R.id.topImage);
    imageViewone = (ImageView) findViewById(R.id.secondImage);
    imageViewthree = (ImageView) findViewById(R.id.thirdImage);
    imageViewfour = (ImageView) findViewById(R.id.forthImage);

    imageViewone.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(),"WomenWerk is a movement that brings women and men together for a modern take on International Women's Day, celebrated annually in March.\n" +
                    "\n" +
                    "WomenWerk was founded to change the way women discuss, perceive, and overcome challenges. Our goal is to provide a platform for women to affirm and celebrate individual and collective success", Toast.LENGTH_LONG);
            toast.show();
        }
    });

    imageViewtwo.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(),"WomenWerk's hallmark event is an annual Forum and Gala held in early March. The next Forum and Gala is set for Saturday, March 12, 2016.", Toast.LENGTH_LONG);
            toast.show();
        }
    });

    imageViewthree.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(),"The WomenWerk Blog is curated for women and men with an international perspective. Our goal is to provide access thought leaders from across the globe on issues relevant to you. ", Toast.LENGTH_LONG);
            toast.show();
        }
    });

    imageViewfour.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast toast = Toast.makeText(getApplicationContext(),"Demi Ajayi and Nekpen Osuan are the founders of Womenwerk, a movement that brings women and men together for a modern take on International Women's Day.", Toast.LENGTH_LONG);
            toast.show();
        }
    });
}
```

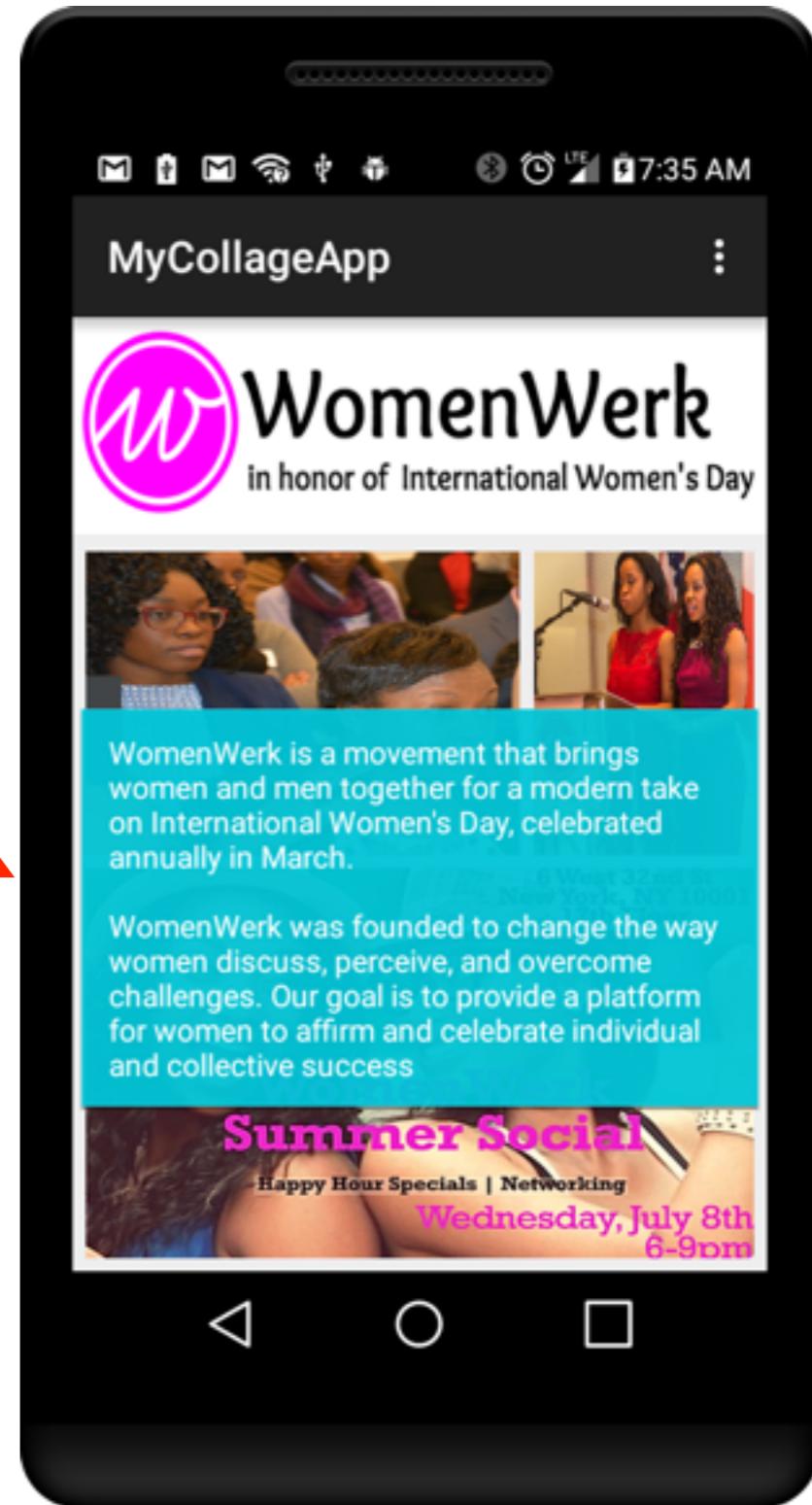
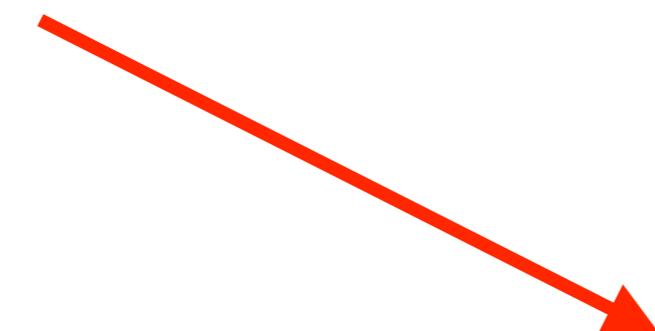
Android Studio: So Far

- Your screen should look like the following



Android Studio: So Far

- When you press a picture, you should get a pop up window with some text
- On my phone, pop up windows are blue



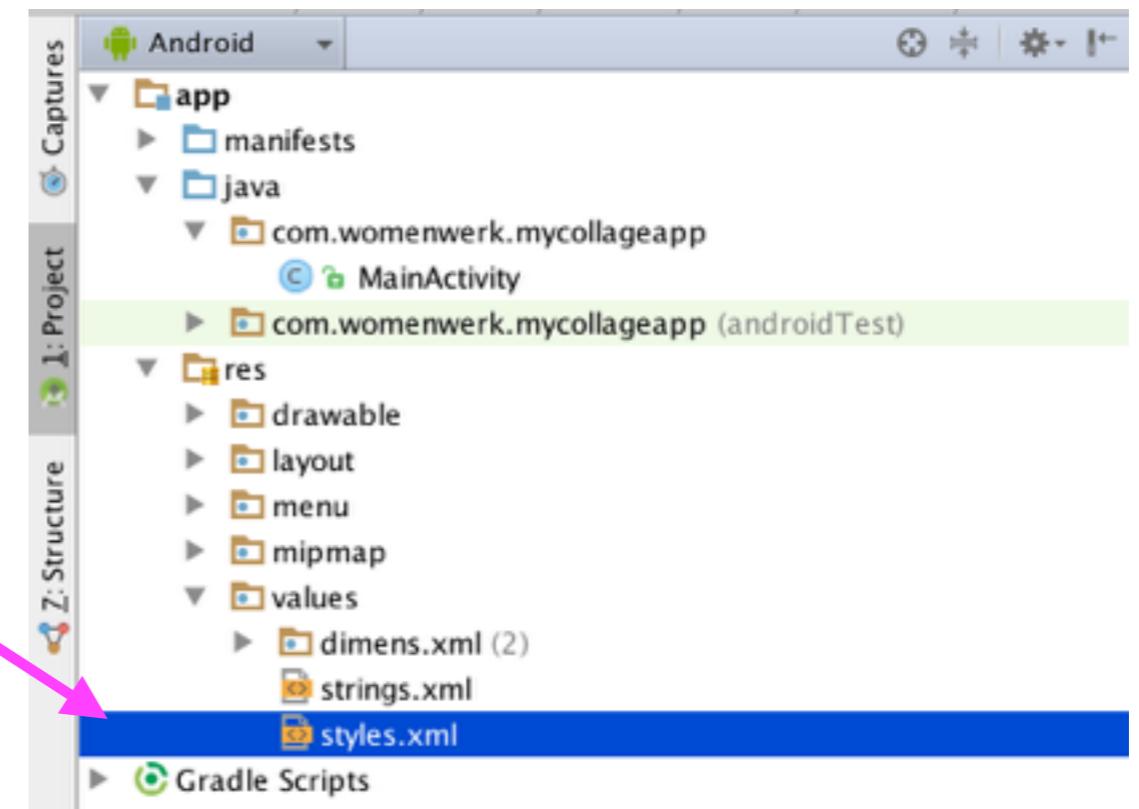
Android Studio: Style

- Let's delete the Action Bar at the top of the screen



Android Studio: Style

- Navigate to the styles.xml file



Android Studio: Style

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
        <!-- Customize your theme here. -->
    </style>

</resources>
```

- Change the “DarkActionBar” text to “NoActionBar”

```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.NoActionBar">
        <!-- Customize your theme here. -->
    </style>

</resources>
```

Android Studio: Finished

- Run the app, it should look like this



Bonus: Android Studio: Enable Portrait Only

- To disable landscape view and only allow for a portrait view...
- Navigate to the AndroidManifest.xml file in the manifest folder
- Add the lines in red inside of the activity

```
<activity
    android:name=".MainActivity"
    android:label="@string/app_name"
    android:configChanges = "orientation"
    android:screenOrientation = "portrait">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```