- **Layouts** The Layouts are the main xml files, that contain the Android xml code with which we are going to develop, how will our application views look like.
- **Values** The Layouts are the main xml files, that contain the Android xml code with which we are going to develop, how will our application views look like.
  - Animation Resources
  - Color State List Resource
  - Drawable Resources
  - Layout Resource
  - Menu Resource
  - String Resources
  - Style Resource
- **Drawables** A drawable resource is a general concept for a graphic that can be drawn to the screen. There are several different types of drawables:
  - Bitmap File A bitmap graphic file (.png, .jpg, or .gif). Creates a BitmapDrawable.
  - Nine-Patch File A PNG file with stretchable regions to allow image resizing based on content (.9.png). Creates a NinePatchDrawable.
  - Layer List A Drawable that manages an array of other Drawables. These are drawn in array order, so the element with the largest index is be drawn on top. Creates a LayerDrawable.
  - State List An XML file that references different bitmap graphics for different states (for example, to use a different image when a button is pressed). Creates a StateListDrawable.
  - Level List An XML file that defines a drawable that manages a number of alternate Drawables, each assigned a maximum numerical value. Creates a LevelListDrawable.
  - Transition Drawable An XML file that defines a drawable that can cross-fade between two drawable resources. Creates a TransitionDrawable.
  - Inset Drawable An XML file that defines a drawable that insets another drawable by a specified distance. This is useful when a View needs a background drawble that is smaller than the View's actual bounds.
  - Clip Drawable An XML file that defines a drawable that clips another Drawable based on this Drawable's current level value. Creates a ClipDrawable.
  - Scale Drawable An XML file that defines a drawable that changes the size of another Drawable based on its current level value. Creates a ScaleDrawable
  - Shape Drawable An XML file that defines a geometric shape, including colors and gradients. Creates a ShapeDrawable.

Once our app is ready, we will use a build tool to compile all the project files and package them together into an .apk file that you can run on Android devices and/or submit to Google Play.

1.6 Create "Hello Android World" application

1.6.1 Create a New Android Studio Project

Open Android Studio and choose Start a new Android Studio Project in the welcome screen.

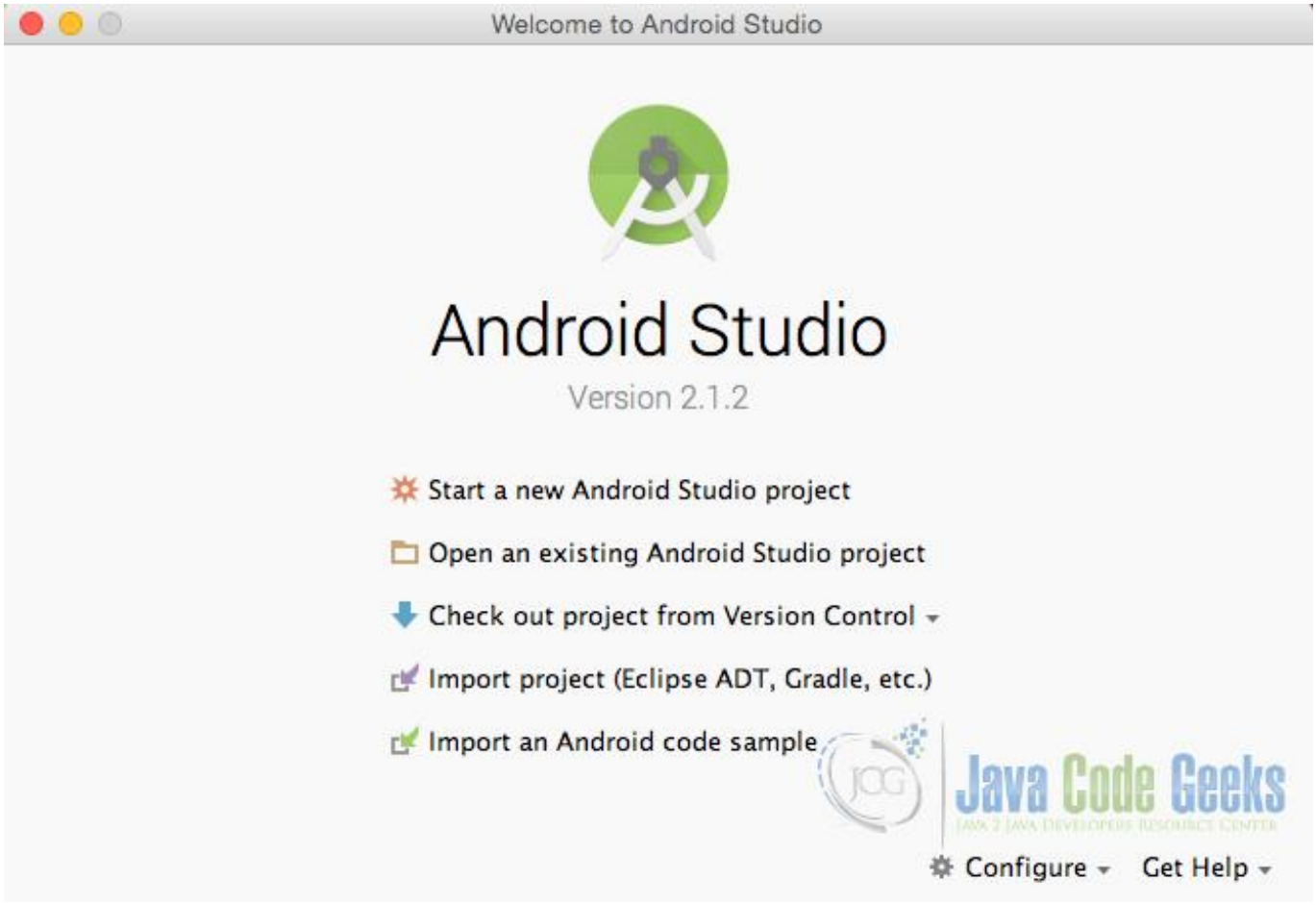

Figure 1.7: Welcome to Android Studio screen. Choose Start a new Android Studio Project.

Specify the name of the application, the project and the package.

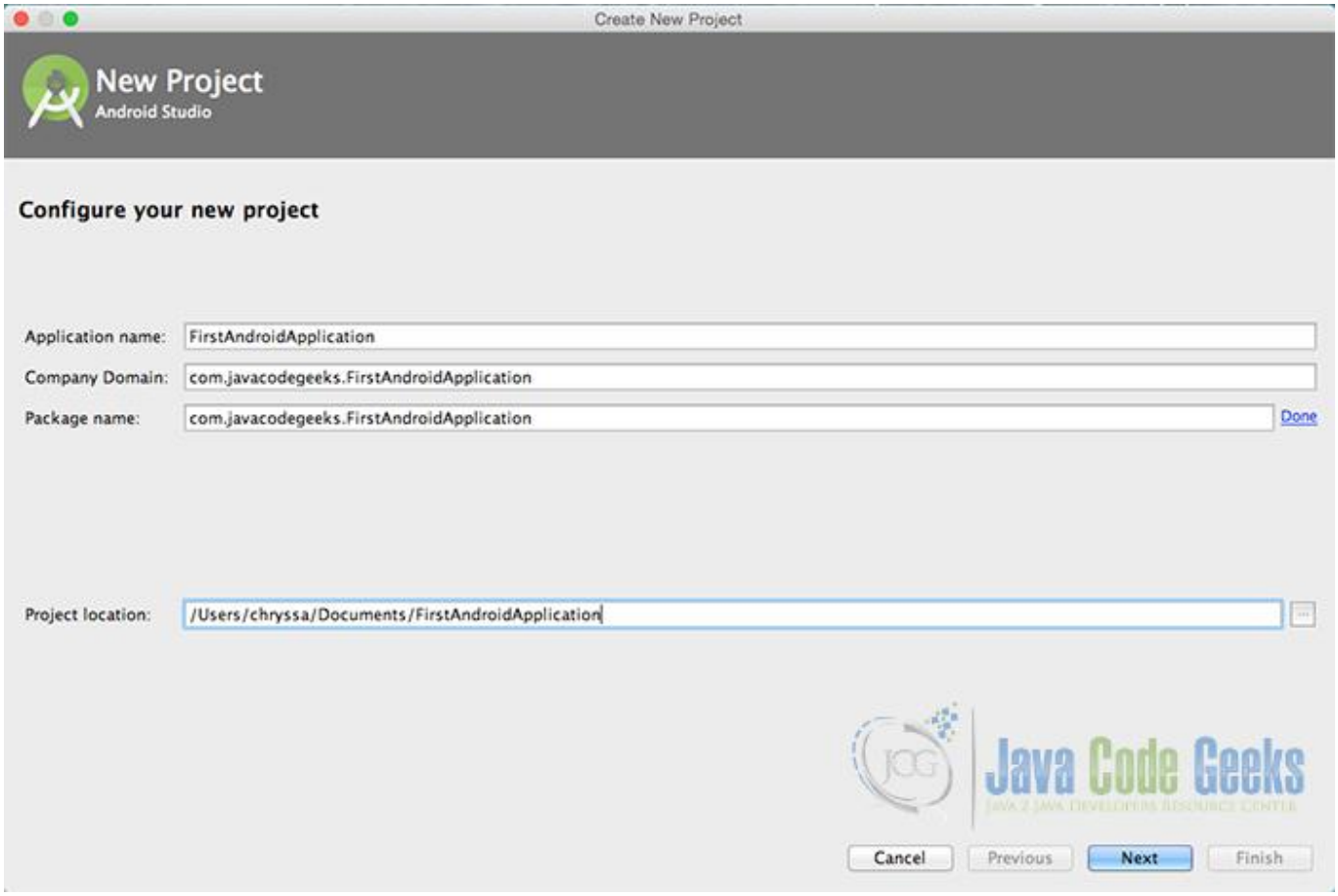

Figure 1.8: Configure your new project screen. Add your application name and the projects package name.

In the next window, select the form factors your app will run on.

Figure 1.9: Target Android Devices screen.

In the next window you should choose Add no activity. In this example, we are going to create our Activity.

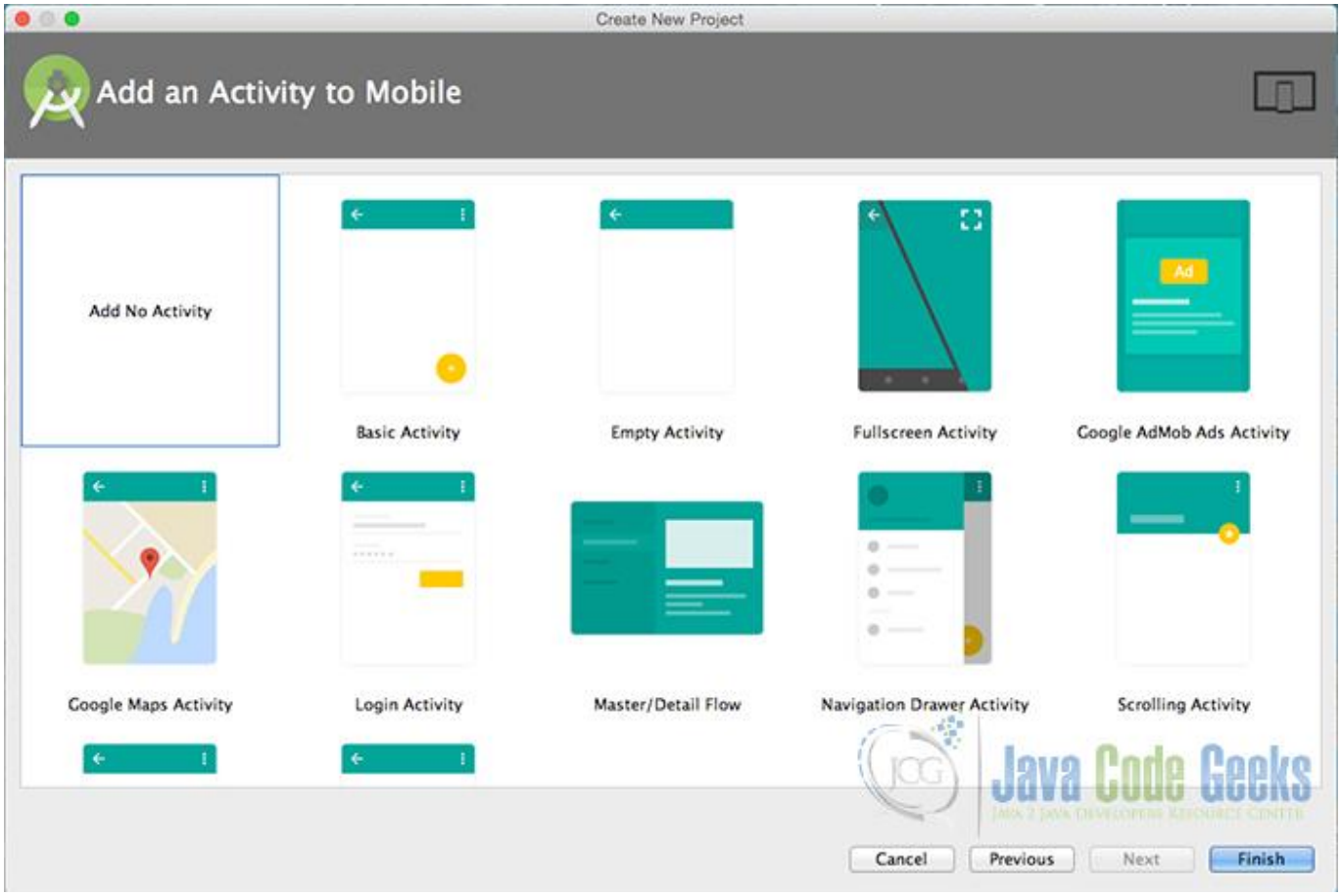

Figure 1.10: Add an activity to Mobile. Choose: Add no activity.

Now, our project has just been created!

1.6.2 Create the source code of a simple FirstAndroidApplication Activity

Add a new Java class Activity inside `src/com.javacodegeeks.FirstAndroidApplication/` so that we are going to have the `src/com.javacodegeeks.FirstAndroidApplication/FirstActivity.java` file and paste the code below.

FirstActivity.java

```
package com.javacodegeeks.FirstAndroidApplication;

import android.app.Activity;
import android.os.Bundle;

public class FirstActivity extends Activity {

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

}
```

1.6.3 Create the layout of the project

Add a new xml file inside `/res/layout` folder, with name `main_layout.xml`. We should have the `/res/layout/main_layout.xml` file and paste the code below.

main_layout.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="https://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#ededed"
    android:gravity="center"
    android:orientation="vertical">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:layout_marginTop="@dimen/textmargin"
        android:gravity="center"
        android:textSize="25dp"
        android:text="@string/helloAndroid" />

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_margin="@dimen/logomargin"
        android:background="@drawable/ic_social_mood" />

</LinearLayout>
```

1.6.4 Android Manifest

Edit the `AndroidManifest.xml` file inside `/app/manifests` folder. The AndroidManifest.xml of our project is simple and should be like this:

AndroidManifest.xml

```
<manifest xmlns:android="https://schemas.android.com/apk/res/android"
    package="com.javacodegeeks.FirstAndroidApplication">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity
            android:name=".FirstActivity"
            android:label="@string/app_name"
            android:screenOrientation="portrait"
            android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

1.6.5 Edit the FirstAndroidApplication dimensions

Add a new xml file inside `/res/values` folder, with name `dimens.xml`. We should have the `/res/values/dimens.xml` file and paste the code below.

dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <dimen name="logomargin">20dp</dimen>
    <dimen name="textmargin">10dp</dimen>
</resources>
```

1.6.6 Edit the FirstAndroidApplication strings

Add a new xml file inside `/res/values` folder, with name `strings.xml`. We should have the `/res/values/strings.xml` file and paste the code below.

strings.xml

```
<resources>
    <string name="app_name">AndroidFirstApplication</string>
    <string name="helloAndroid">Hello Android!</string>
</resources>
```

1.6.7 Add the drawable for every screen density

Inside `/res/values` folder, we should add the folders for each screen dimension we have, and add the specific drawable for each one.

Figure 1.11: Add the drawables for every screen density.

In this way, we are going to have the right drawable dimension for every different screen density.

1.6.8 Build, compile and run

When we are ready, we build our application by pressing the play button in our AndroidStudio main toolbar.

Figure 1.12: Compile and run.

After we build, compile and run our project, the main FirstAndroidApplication application should look like this:

Figure 1.13: This is our FirstAndroidApplication.