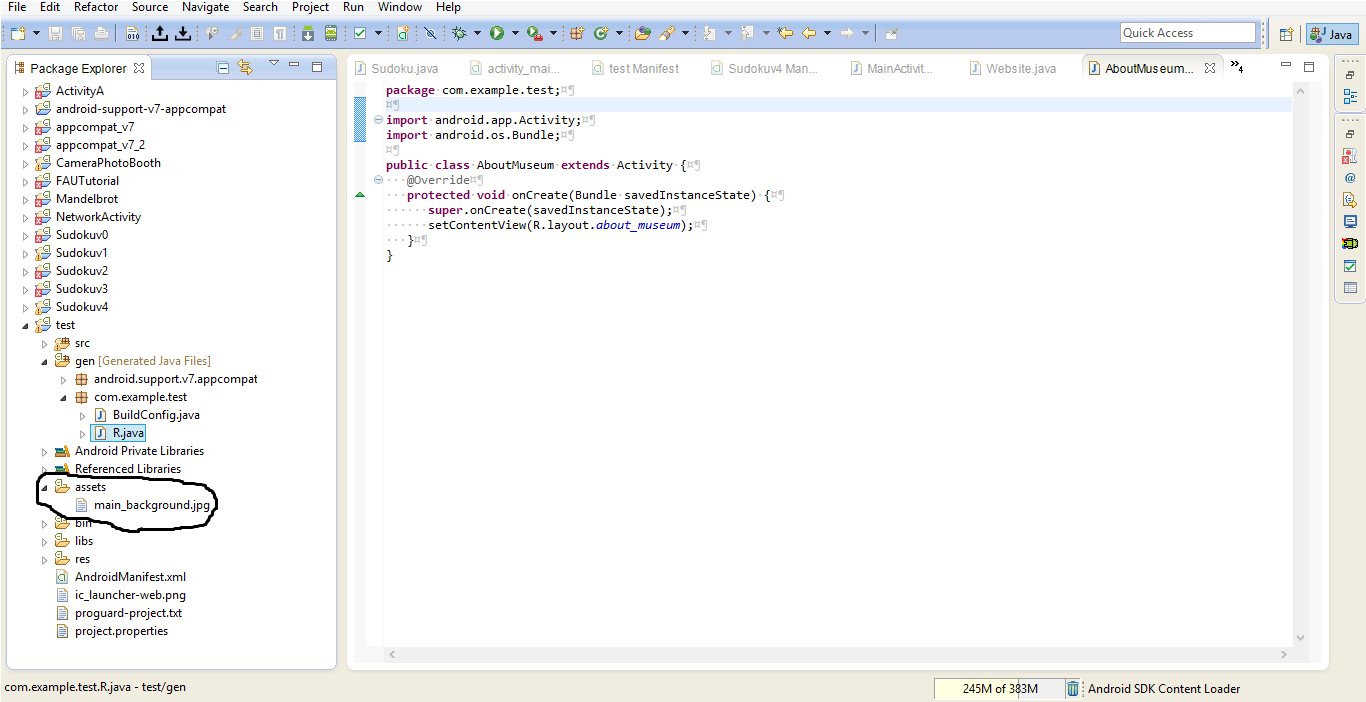
**How to create scrollable and zoomable images**

1. Create a new android project, with any name that you choose.

2. Put the image you want to be zoomable and scrollable under the assets folder in your project:



3. In your *main\_activity.xml*, (or the layout xml file that you want the picture to be in), insert the following code:

<?xml version="1.0" encoding="utf-8"?>

<WebView xmlns:android="http://schemas.android.com/apk/res/android"

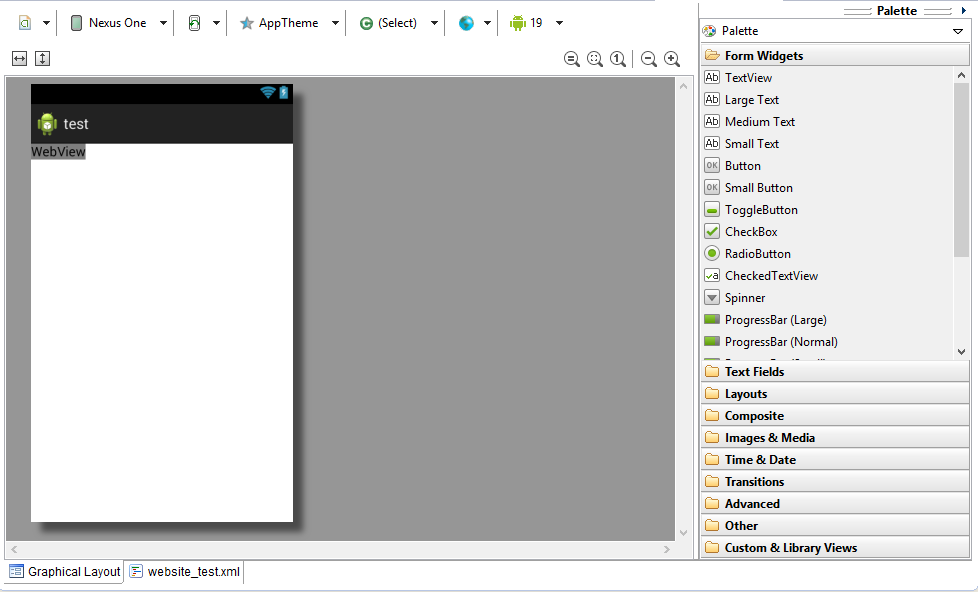
android:id="@+id/webview"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" >

</WebView>

So your xml file should look something like this:



4. Add this import statement to your .java file:

import android.webkit.\*;

5. In your .java file, insert the following code in the method onCreate:

WebView myWebView = (WebView) findViewById(R.id.webview);

WebSettings webSettings = myWebView.getSettings();

webSettings.setBuiltInZoomControls(true);

myWebView.loadUrl("file:///android\_asset/*your\_picture*");

(for example, my picture is named *main\_background.jpg*, so I would use myWebView.loadUrl("file:///android\_asset/main\_background.jpg"));

6. Run the app, and try zooming in on your custom image or scrolling it.

**NOTE:** If you did not create the webview under your main\_activity.xml file, and instead you created it under a different xml file, you have to make sure that the activity you use to access that new xml file is declared in your AndroidManifest.xml file.

So for example, if I have a custom activity I created that uses *website.xml* which contains my webview and image, and I defined a custom class called *Website.java* that uses all of the code above, then in my AndroidManifest.xml, I have to do this:

<application android:icon="@drawable/icon"

android:label="@string/app\_name">

**<activity android:name=".MainActivity"**

**android:label="@string/app\_name">**

**<intent-filter>**

**<action android:name="android.intent.action.MAIN" />**

**<category android:name="android.intent.category.LAUNCHER" />**

**</intent-filter>**

**</activity>**

The bold in above represents the initial activity that runs when you app first starts, in order word the main menu screen.

**<activity android:name=".Website">**

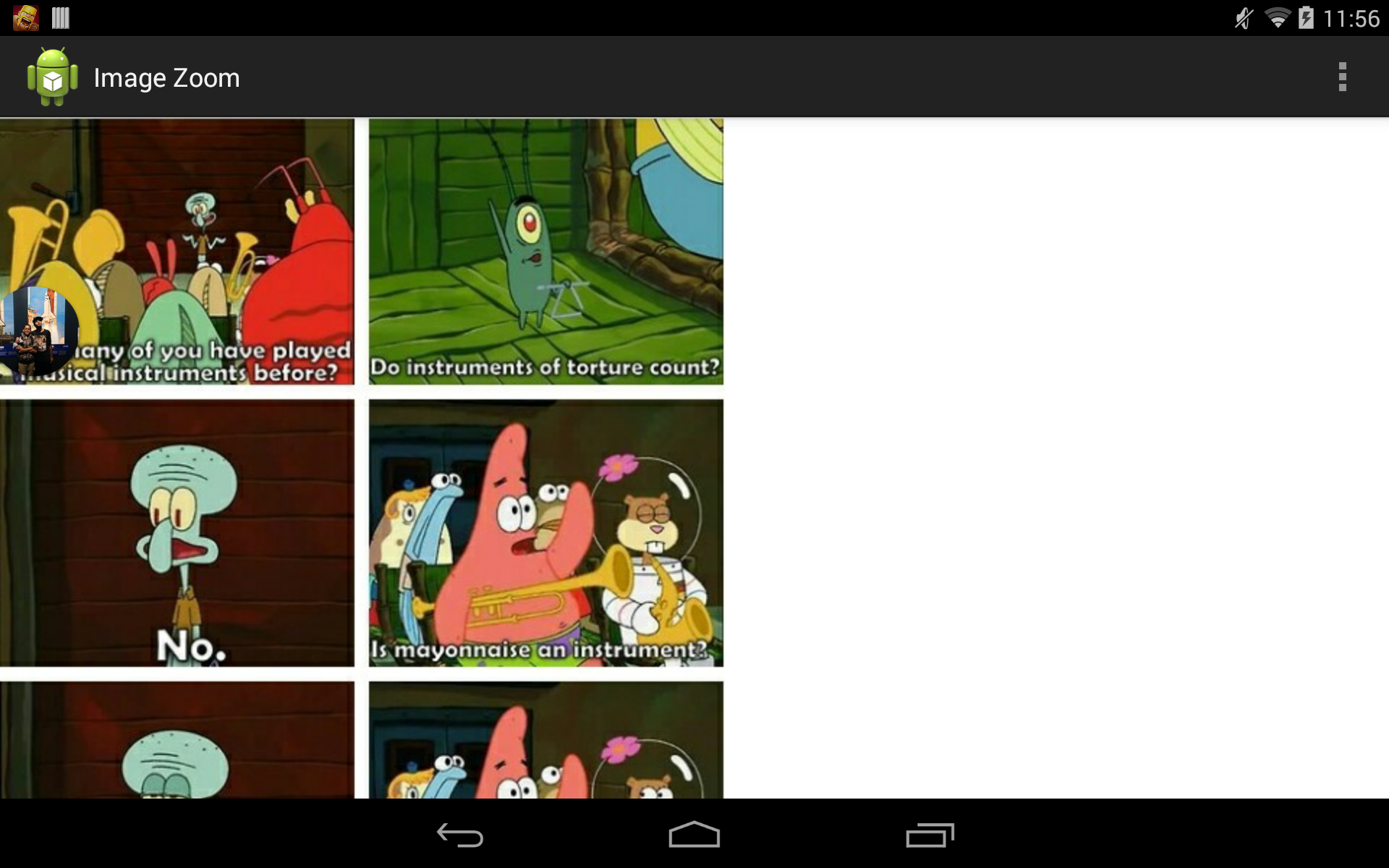
**</activity>**

The bold above represents your custom activity. You have to put .YourClassName after android:name=. So because my custom class name was Website.java, I have to put android:name=".Website".

</application>

Here's an example:

Original:



Zoomed-in Picture:

