

# Bust the Android Fragmentation Myth

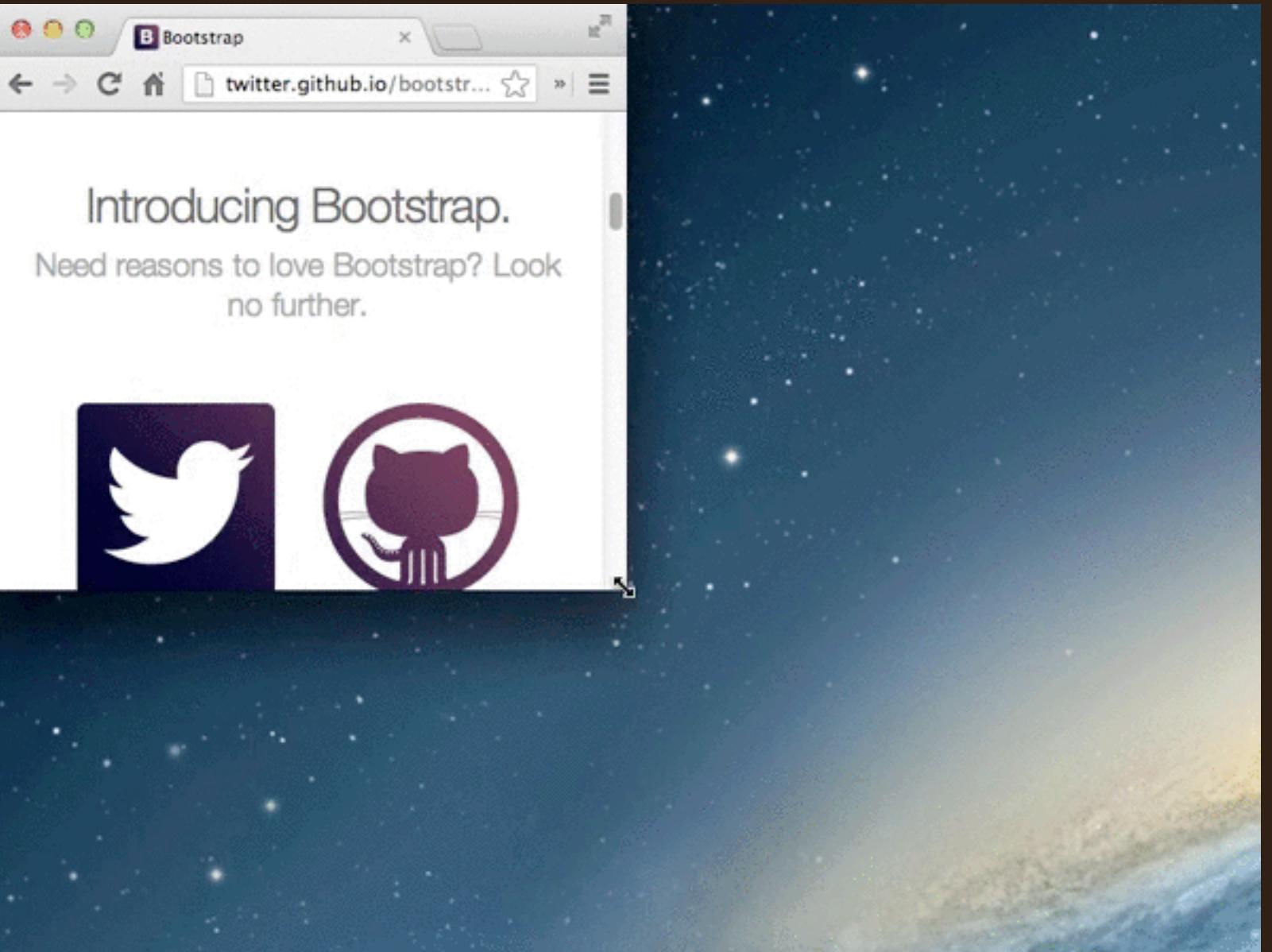
Chiu-Ki Chan  
@chiuki

# So many devices!



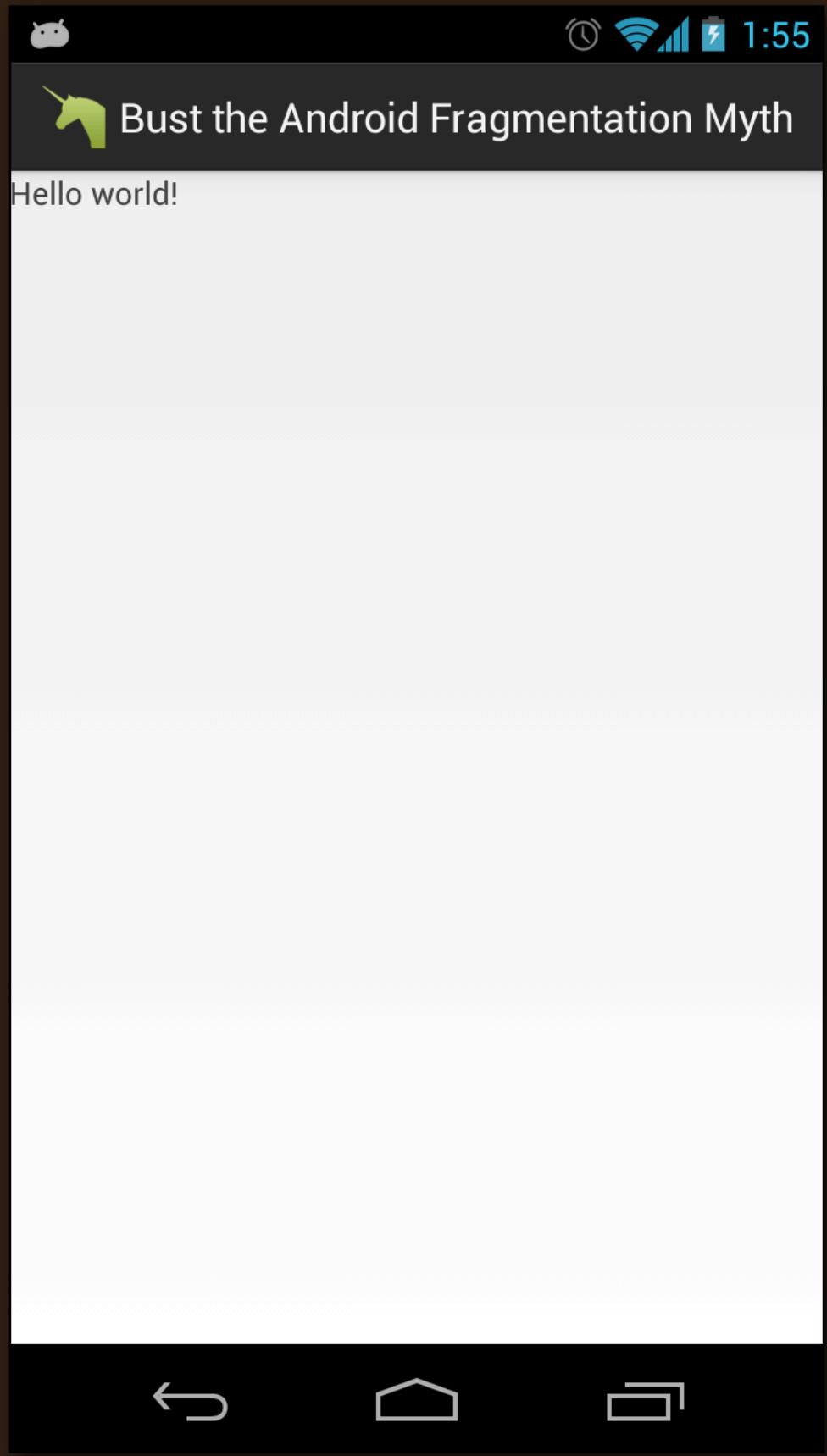
@chiuki

# Infinite screen sizes!



Responsive  
+  
Progressive

# Hello World



@chiuki

The screenshot shows the Android Studio interface with the project 'bust' open. The left pane displays the project structure, and the right pane shows the code editor for `MainActivity.java`.

`MainActivity.java` content:

```
1 package com.sqisland.android.bust;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.TextView;
6
7 public class MainActivity extends Activity {
8     private TextView textView;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        textView = (TextView) findViewById(R.id.text);
16        textView.setText(R.string.hello_world);
17    }
18}
```

@chiuki

The screenshot shows the Android Studio interface with the project 'bust' open. The left pane displays the project structure, including the .idea folder, Bust module (with build, libs, and src folders), and various resource and configuration files like build.gradle, Bust.iml, gradle, gradlew, and gradlew.bat. The right pane shows the code editor with MainActivity.java and activity\_main.xml tabs. The Java code defines a MainActivity that extends Activity, imports android.os.Bundle, android.app.Activity, and android.widget.TextView. It overrides onCreate to set the content view to R.layout.activity\_main, finds a TextView by ID, and sets its text to "hello world". The XML file activity\_main.xml is shown below, featuring a single TextView with the id "text".

```
1 package com.sqisland.android.bust;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.TextView;
6
7 public class MainActivity extends Activity {
8     private TextView textView;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        textView = (TextView) findViewById(R.id.text);
16        textView.setText(R.string.hello_world);
17    }
18}
```

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >
    <TextView
        android:id="@+id/text"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello world" />
</LinearLayout>
```

@chiuki

The screenshot shows the Android Studio interface with the project 'bust' open. The left pane displays the project structure, and the right pane shows the code editor for `MainActivity.java`.

**Project Structure:**

- bust (~/projects/android-studio/bust)**
  - .idea
  - Bust
    - build
    - libs
    - src
      - main
        - java
          - com.sqisland.android.bust
        - res
          - drawable-hdpi
          - drawable-mdpi
          - drawable-xhdpi
          - drawable-xxhdpi
          - layout
            - activity\_main.xml
        - values
          - strings.xml
          - styles.xml
        - values-v11
        - values-v14
    - build.gradle
    - Bust.iml
    - gradle
    - build.gradle
    - bust.iml
    - gradlew
    - gradlew.bat

**MainActivity.java (Code Editor):**

```
1 package com.sqisland.android.bust;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.TextView;
6
7 public class MainActivity extends Activity {
8     private TextView textView;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        textView = (TextView) findViewById(R.id.text);
16        textView.setText(R.string.hello_world);
17    }
18}
```

@chiuki

The screenshot shows the Android Studio interface with the following details:

- Project View:** On the left, the project structure for the "bust" application is displayed. It includes the .idea folder, a Bust module with build, libs, and src folders containing main, java, and res subfolders, and various resource files like strings.xml and styles.xml. Build files like build.gradle and bust.iml are also shown.
- Code Editors:** There are two tabs open in the main area:
  - MainActivity.java**: Shows a single TextView definition with attributes: xmlns:android, android:id, android:layout\_width, and android:layout\_height.
  - activity\_main.xml**: Shows the XML layout file for the activity, which contains the TextView defined in MainActivity.java.

@chiuki

The screenshot shows the Android Studio interface with the following details:

- Project View:** On the left, the project structure for the "bust" application is displayed. It includes the .idea folder, the Bust module with its build, libs, and src folders containing main/java and res (drawable-hdpi, drawable-mdpi, drawable-xhdpi, drawable-xxhdpi, layout, values, values-v11, values-v14, and AndroidManifest.xml), and build.gradle and Bust.iml files.
- MainActivity.java:** The code editor tab for MainActivity.java is open, showing the following Java code:

```
1 <TextView
2     xmlns:android="http://schemas.android.com/apk/res/android"
3     android:id="@+id/text"
4     android:layout_width="wrap_content"
5     android:layout_height="wrap_content" />
```
- activity\_main.xml:** The XML layout file for the activity is also visible in the code editor tab, showing the same TextView definition.

@chiuki

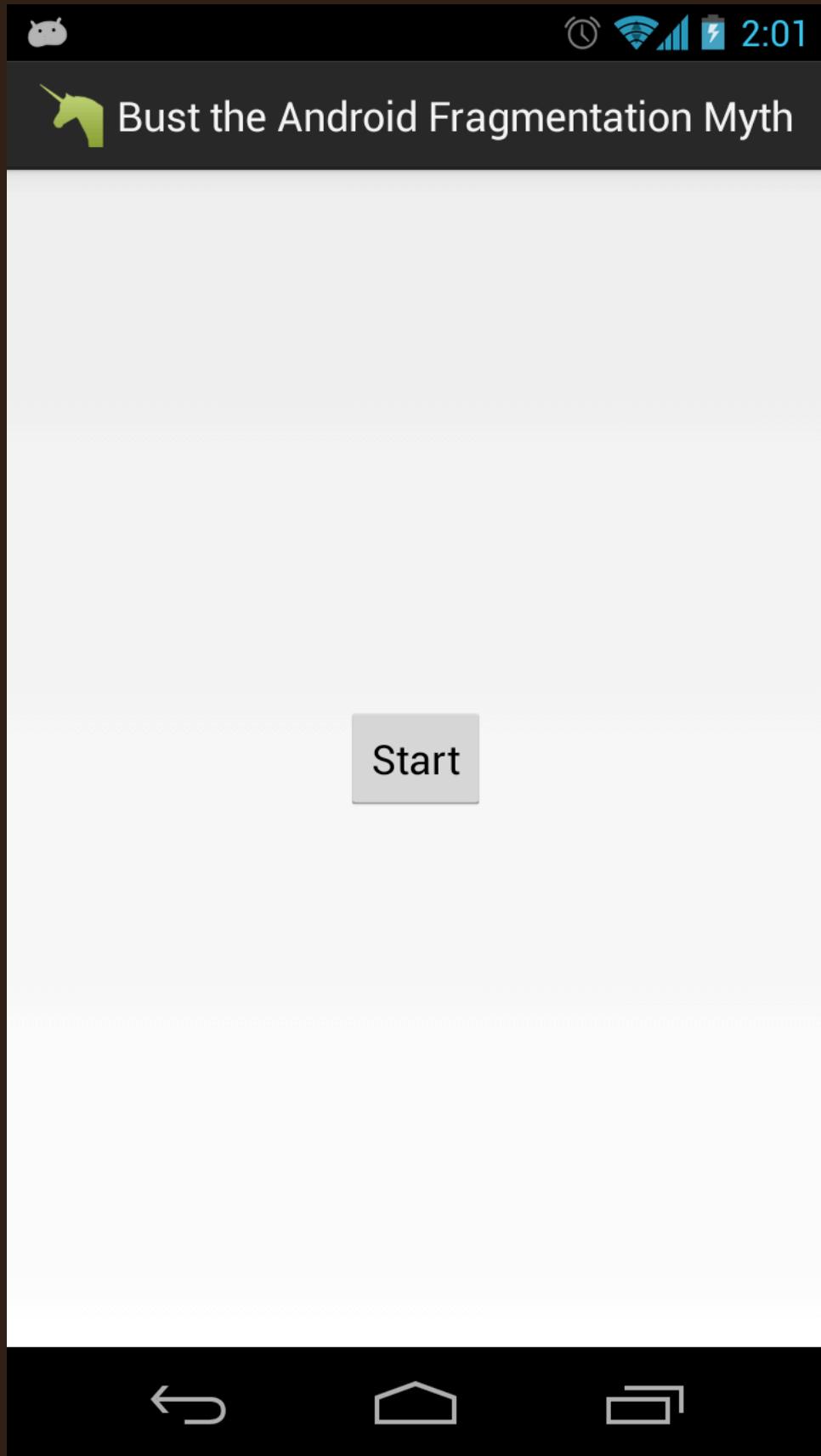
The screenshot shows the Android Studio interface with the project 'bust' open. The left pane displays the project structure, including the .idea folder, Bust module (build, libs, src), and various resource and configuration files like build.gradle, Bust.iml, gradle, gradlew, and gradlew.bat. The right pane shows the code editor with MainActivity.java and activity\_main.xml tabs. The code in MainActivity.java is as follows:

```
1 package com.sqisland.android.bust;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.TextView;
6
7 public class MainActivity extends Activity {
8     private TextView textView;
9
10    @Override
11    protected void onCreate(Bundle savedInstanceState) {
12        super.onCreate(savedInstanceState);
13        setContentView(R.layout.activity_main);
14
15        textView = (TextView) findViewById(R.id.text);
16        textView.setText(R.string.hello_world);
17    }
18}
```

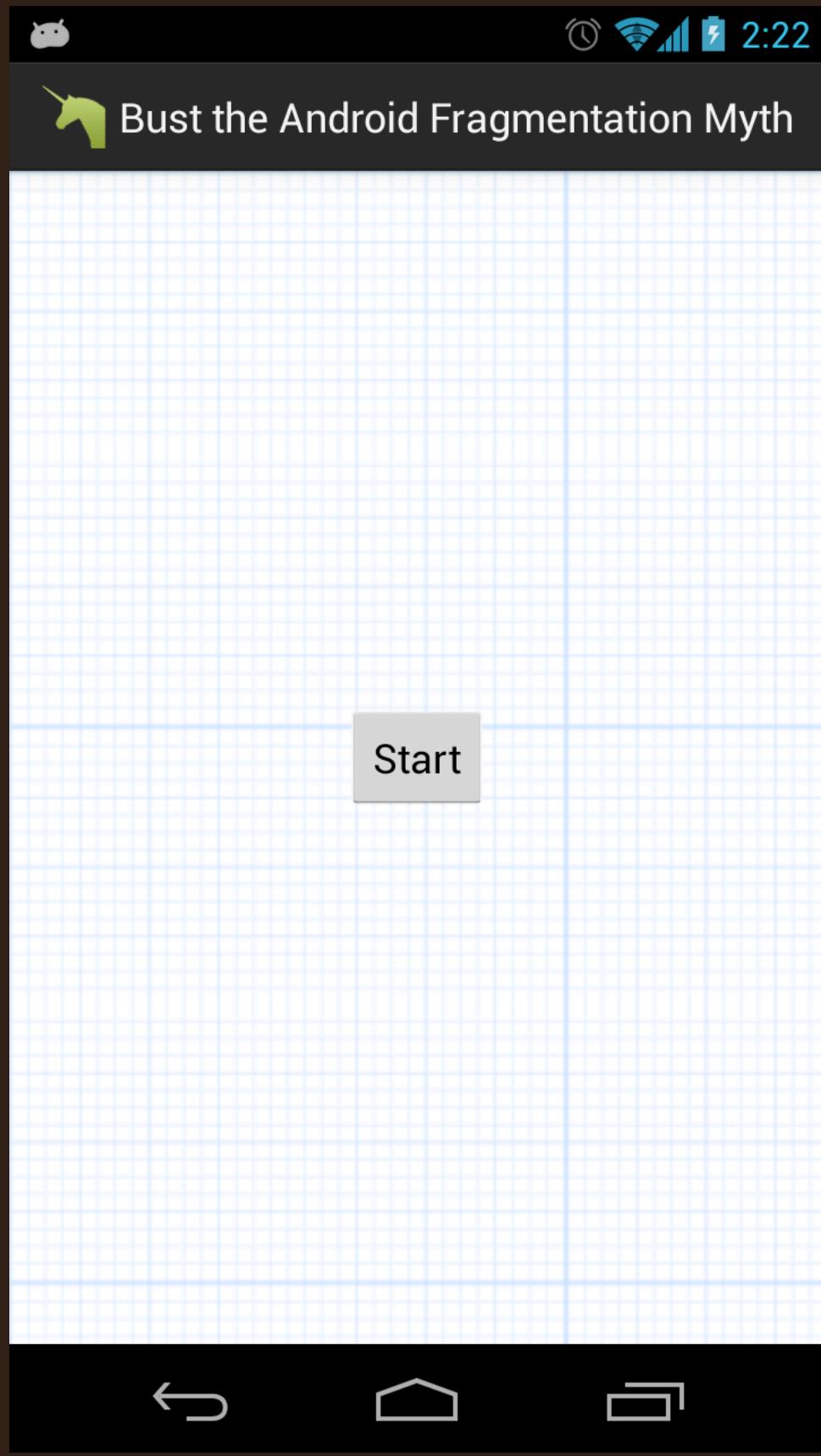
The line `textView = (TextView) findViewById(R.id.text);` is highlighted with a red rectangle.

@chiuki

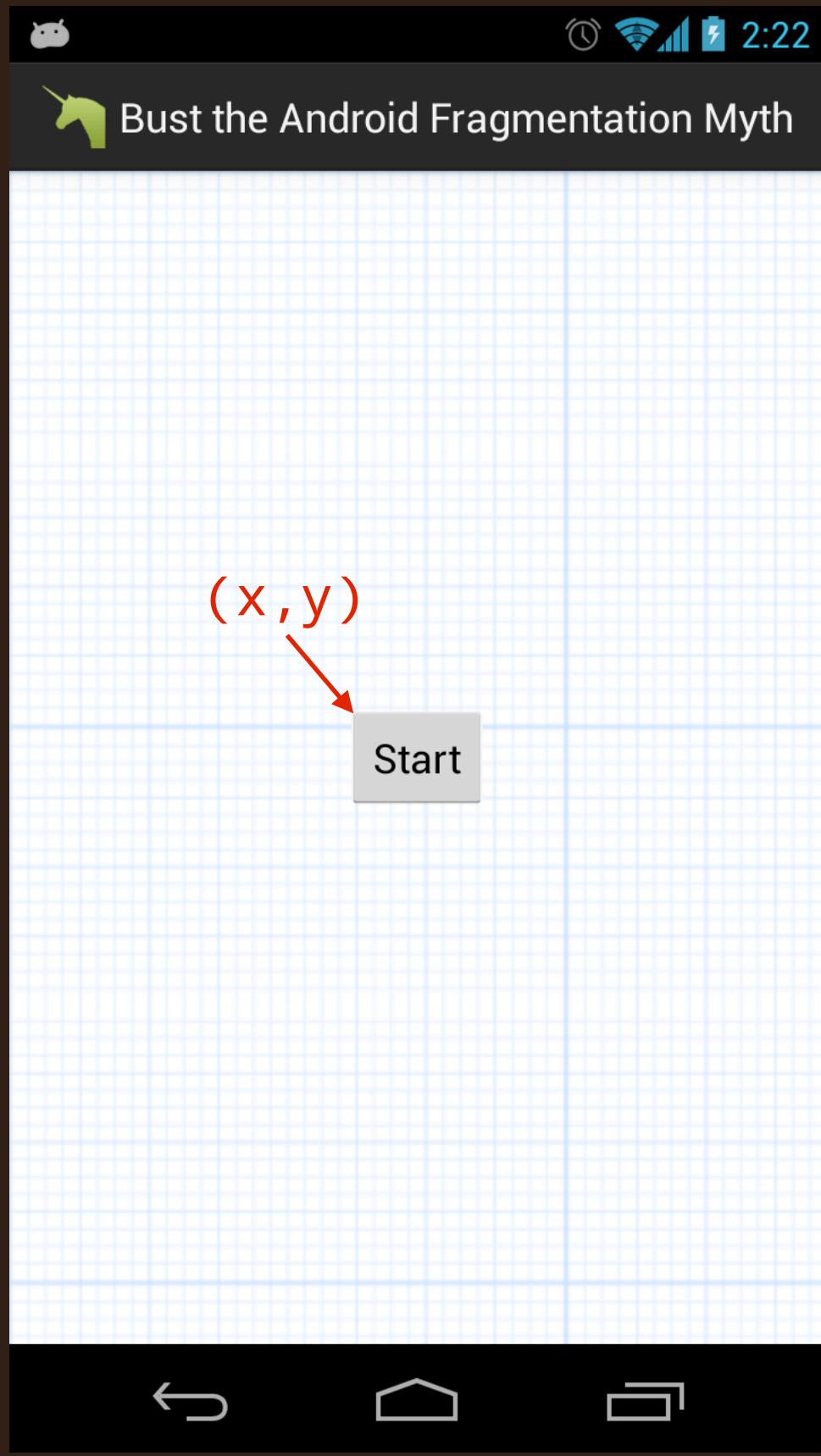
# Declarative Layout



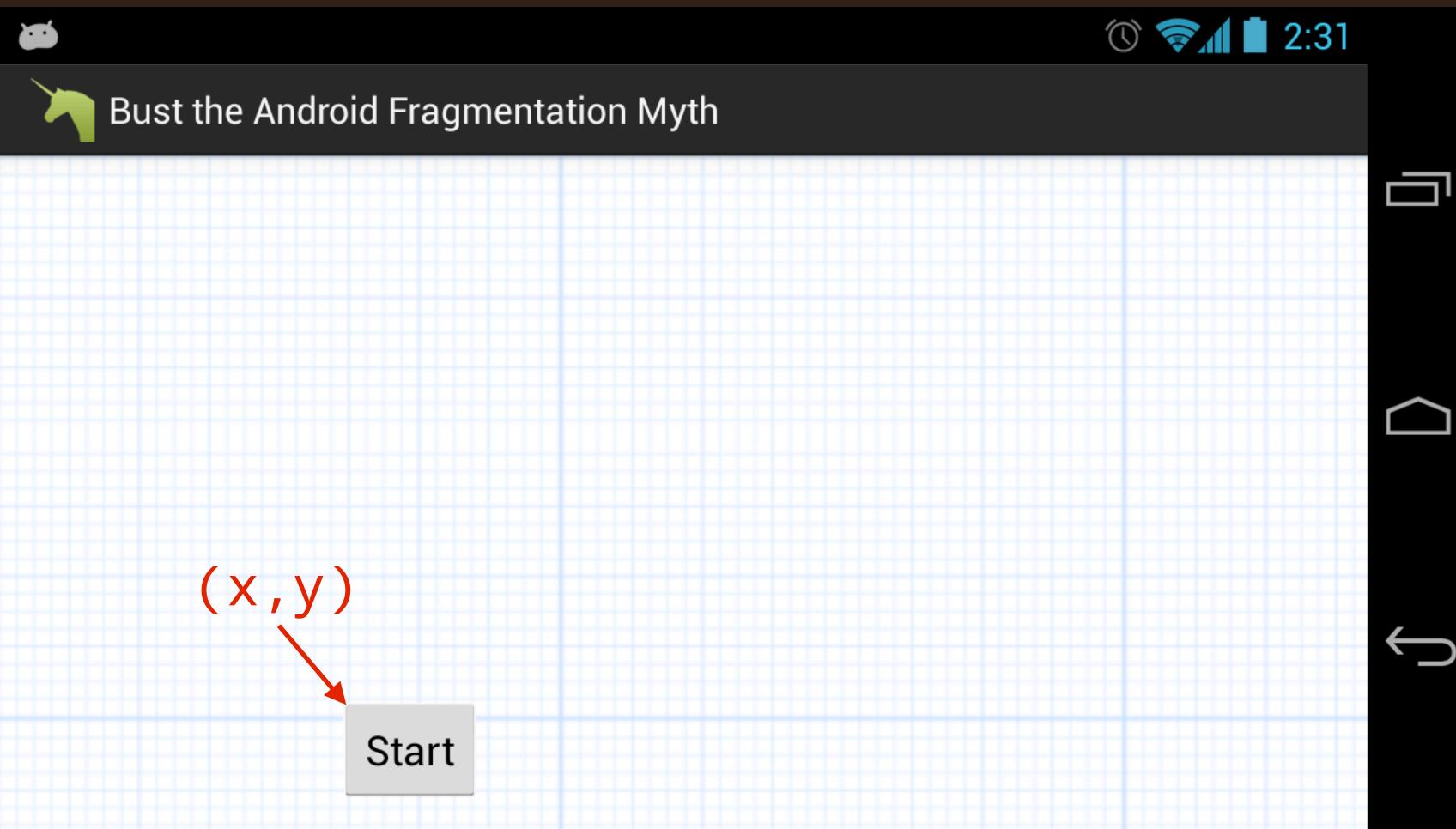
@chiuki



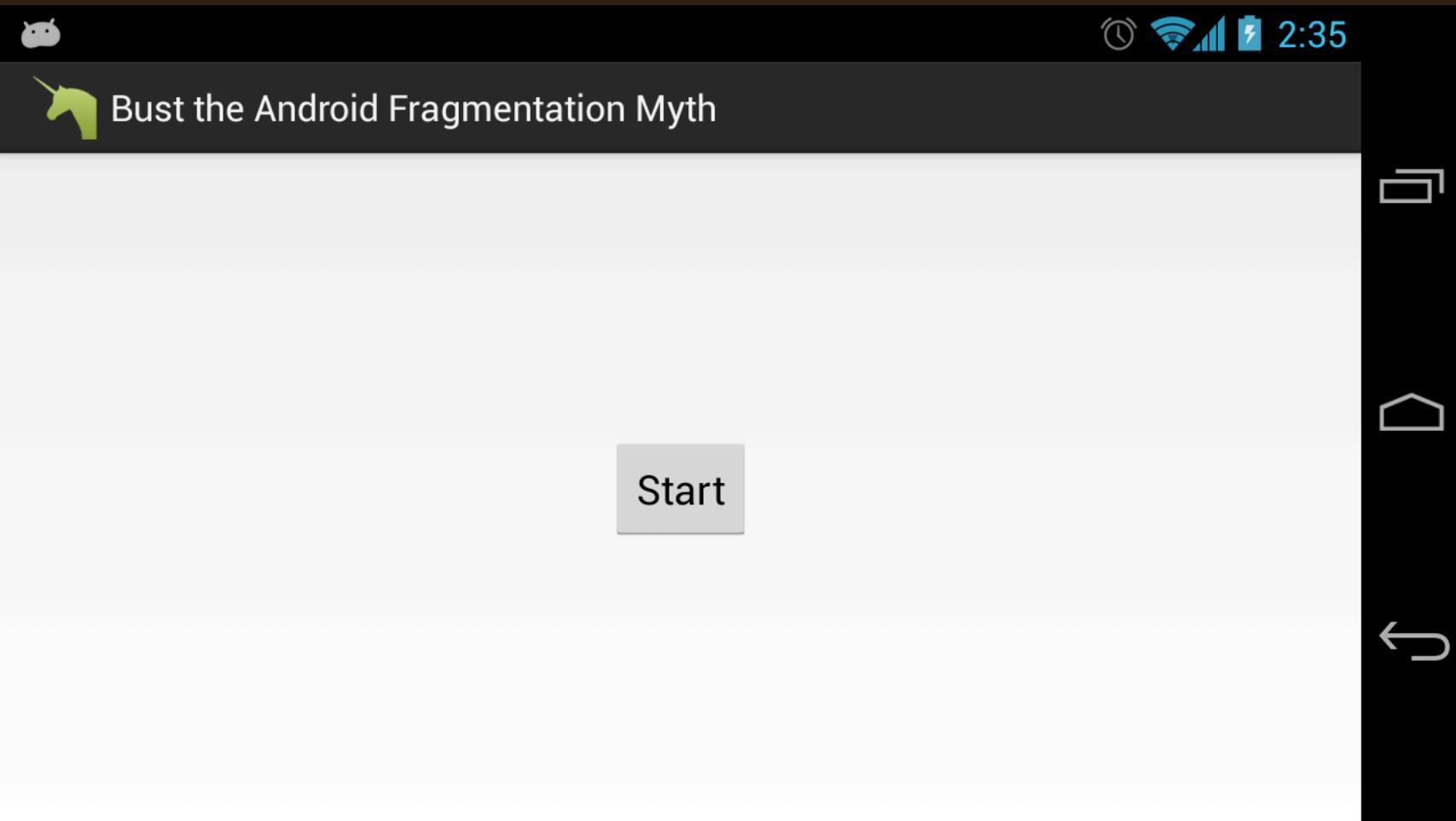
@chiuki



@chiuki



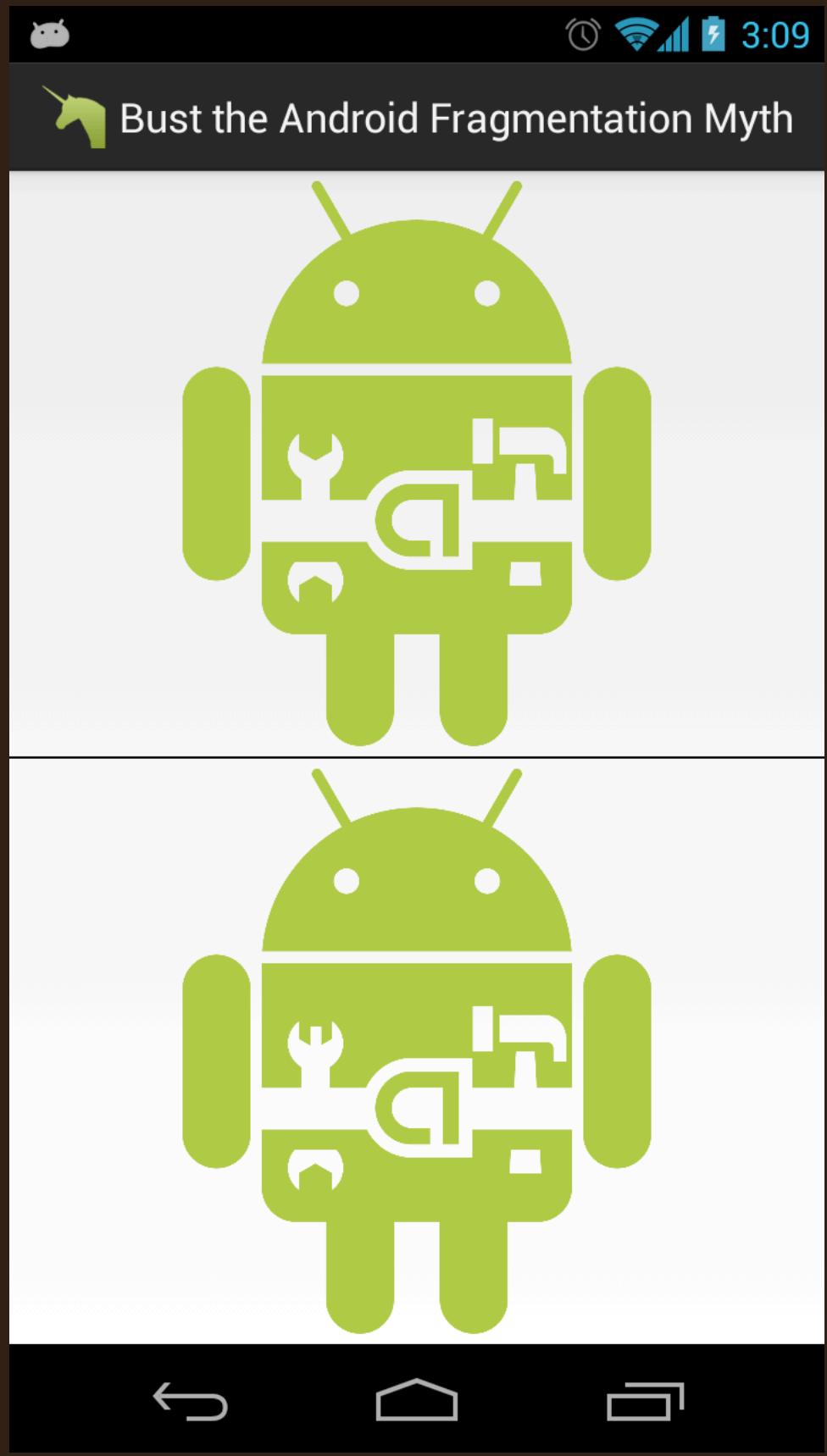
@chiuki



Center

# Center

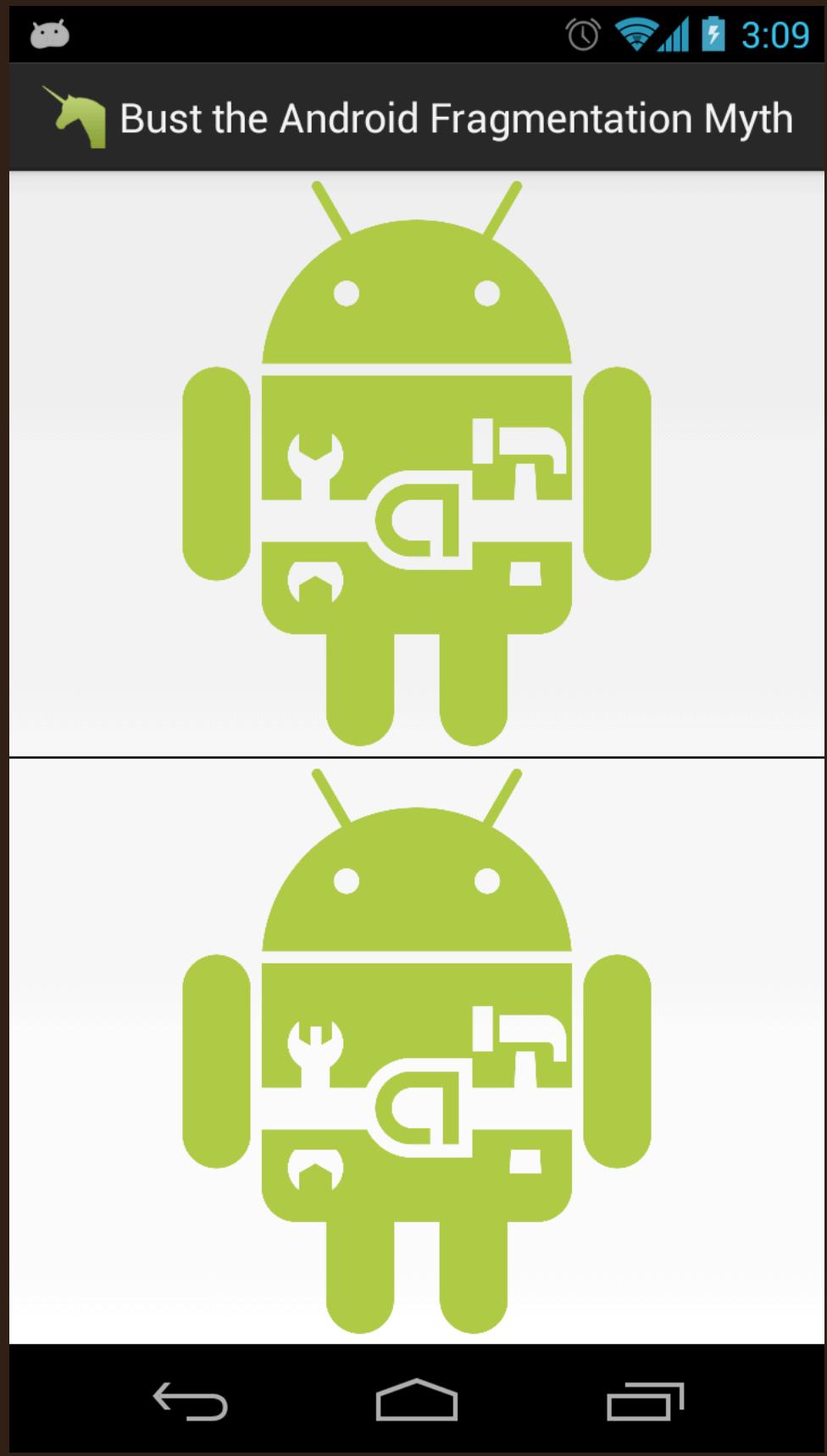
```
<Button  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:text="@string/start" />
```



@chiuki

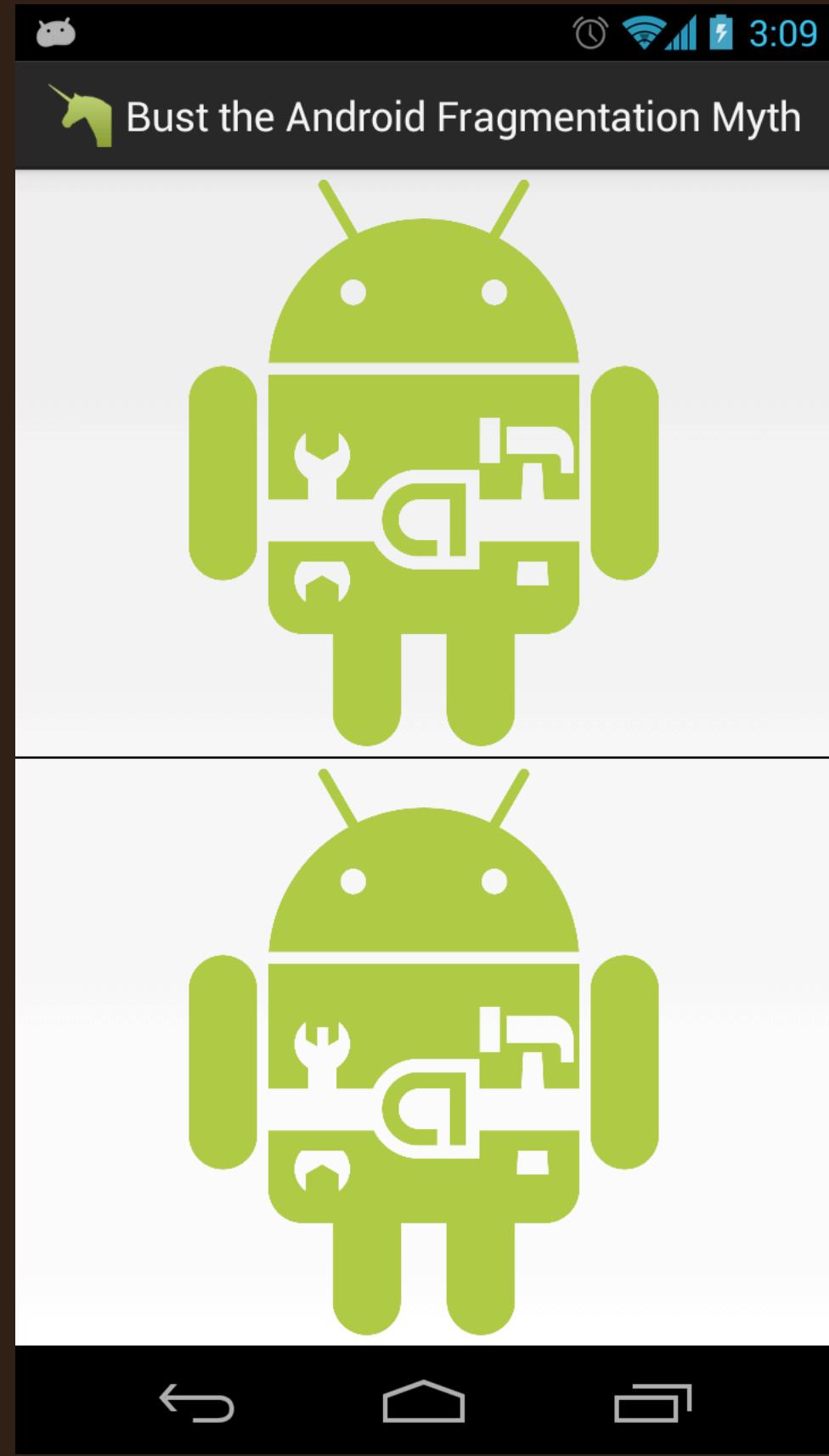
# Proportional Layout

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="vertical" >  
  
    <ImageView  
        android:id="@+id/image1"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_weight="1" />  
  
    <ImageView  
        android:id="@+id/image2"  
        android:layout_width="match_parent"  
        android:layout_height="0dp"  
        android:layout_weight="1" />  
  
</LinearLayout>
```



@chiuki

Divider



# Divider

```
<LinearLayout>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
</LinearLayout>
```

# Divider

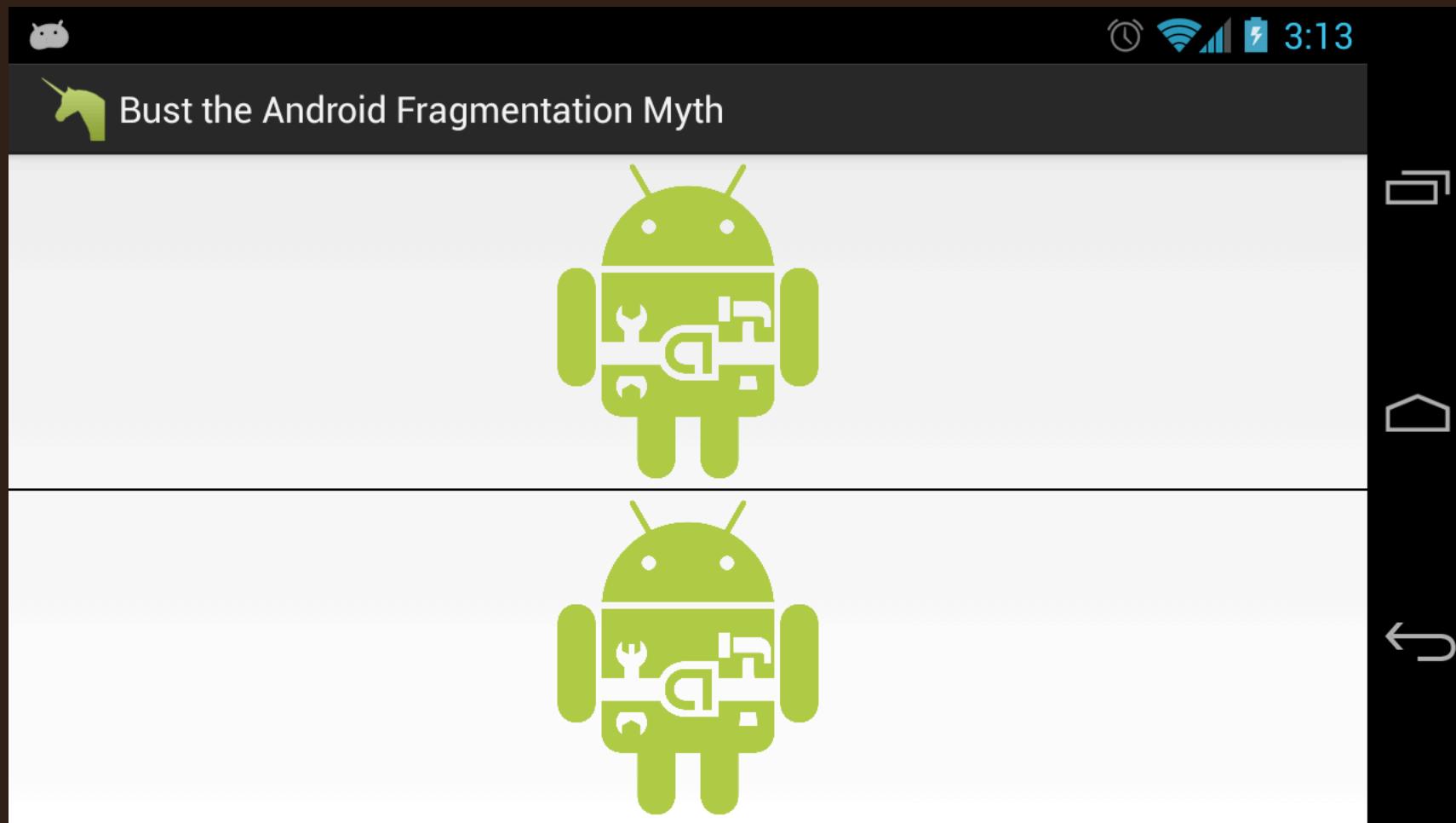
```
<LinearLayout>
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
    <View
        android:layout_width="match_parent"
        android:layout_height="1dp"
        android:background="@android:color/black" />
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="0dp"
        android:layout_weight="1" />
</LinearLayout>
```

dp

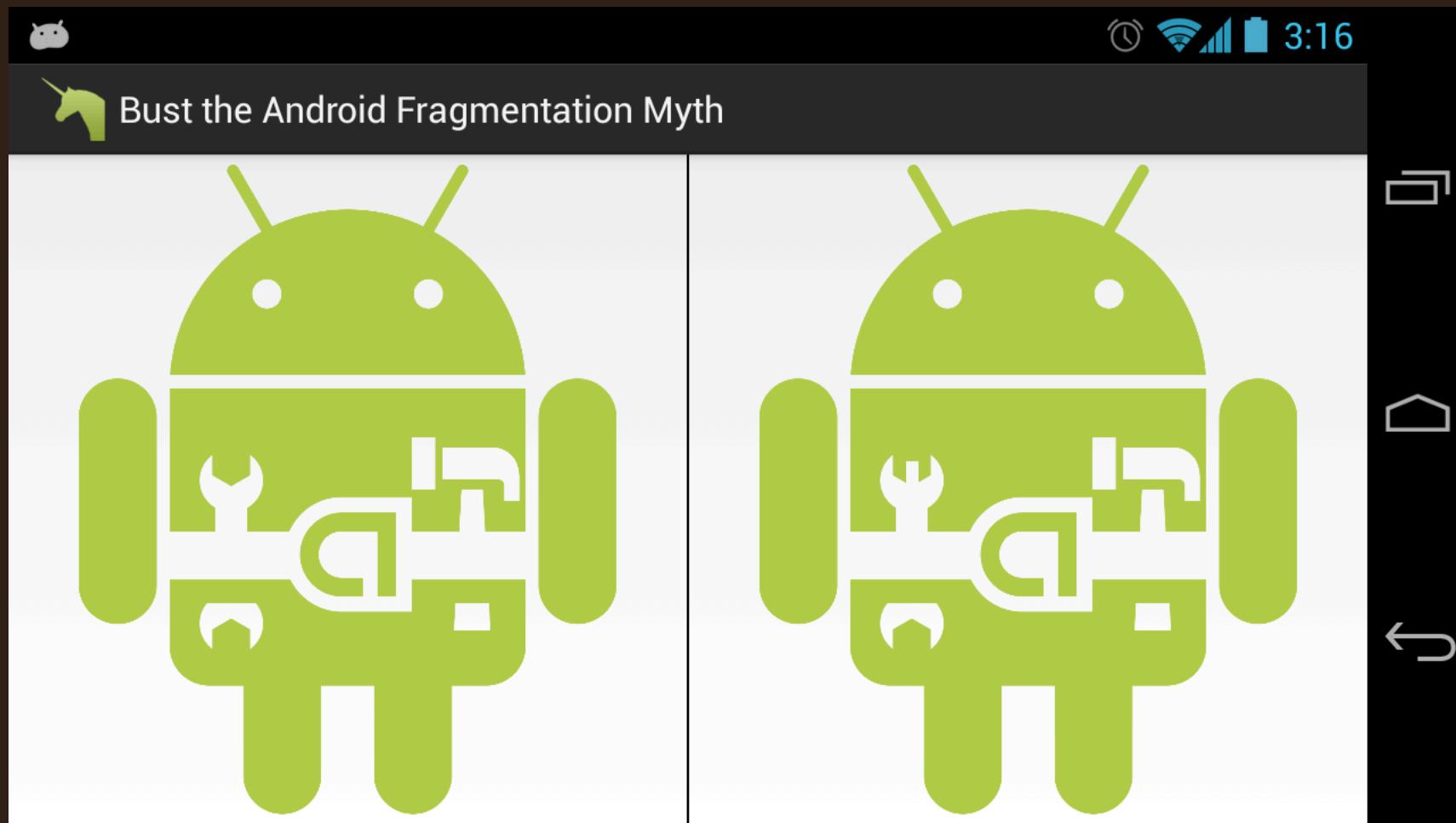
@chiuki

dp

Density-independent pixel



@chiuki



@chiuki

# res/layout-land

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal" >  
  
    <ImageView  
        android:id="@+id/image1"  
        android:layout_width="0dp"  
        android:layout_height="match_parent"  
        android:layout_weight="1" />  
  
    <ImageView  
        android:id="@+id/image2"  
        android:layout_width="0dp"  
        android:layout_height="match_parent"  
        android:layout_weight="1" />  
  
</LinearLayout>
```

The screenshot shows the Android Studio interface with the project 'bust' open. The left pane displays the project structure, and the right pane shows the code editor for `MainActivity.java`.

**Project Structure:**

- bust (~/projects/android-studio/bust)**
  - .idea
  - Bust
    - build
    - libs
    - src
      - main
        - java
          - com.sqisland.android.bust
        - res
          - drawable-hdpi
          - drawable-mdpi
          - drawable-xhdpi
          - drawable-xxhdpi
          - layout
            - activity\_main.xml
            - layout-land
              - activity\_main.xml
          - values
            - strings.xml
            - styles.xml
          - values-v11
          - values-v14
    - build.gradle
    - Bust.iml
    - gradle
    - build.gradle
    - bust.iml

**MainActivity.java:**

```
1 package com.sqisland.android.bust;
2
3 import android.os.Bundle;
4 import android.app.Activity;
5 import android.widget.ImageView;
6
7 public class MainActivity extends Activity {
8     private ImageView image1;
9     private ImageView image2;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
14         setContentView(R.layout.activity_main);
15
16         image1 = (ImageView) findViewById(R.id.image1);
17         image2 = (ImageView) findViewById(R.id.image2);
18
19         image1.setImageResource(R.drawable.android_tools_1);
20         image2.setImageResource(R.drawable.android_tools_2);
21     }
22 }
```

@chiuki

# Resource Folders

# Resource Folders

<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values drawable menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi Platform version: v4, v11, v14 UI mode: car, desk, television, appliance

# Resource Folders

<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values drawable menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi Platform version: v4, v11, v14 UI mode: car, desk, television, appliance

res/layout-land/activity\_main.xml

# Resource Folders

<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values <b>drawable</b> menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, <b>hdpi</b> , xhdpi, nodpi, tvdpi Platform version: v4, v11, v14 UI mode: car, desk, television, appliance

res/drawable-hdpi/ic\_launcher.png

# Resource Folders

<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values <b>drawable</b> menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi Platform version: v4, v11, v14 UI mode: car, desk, television, appliance

res/drawable-large-land/splash.png



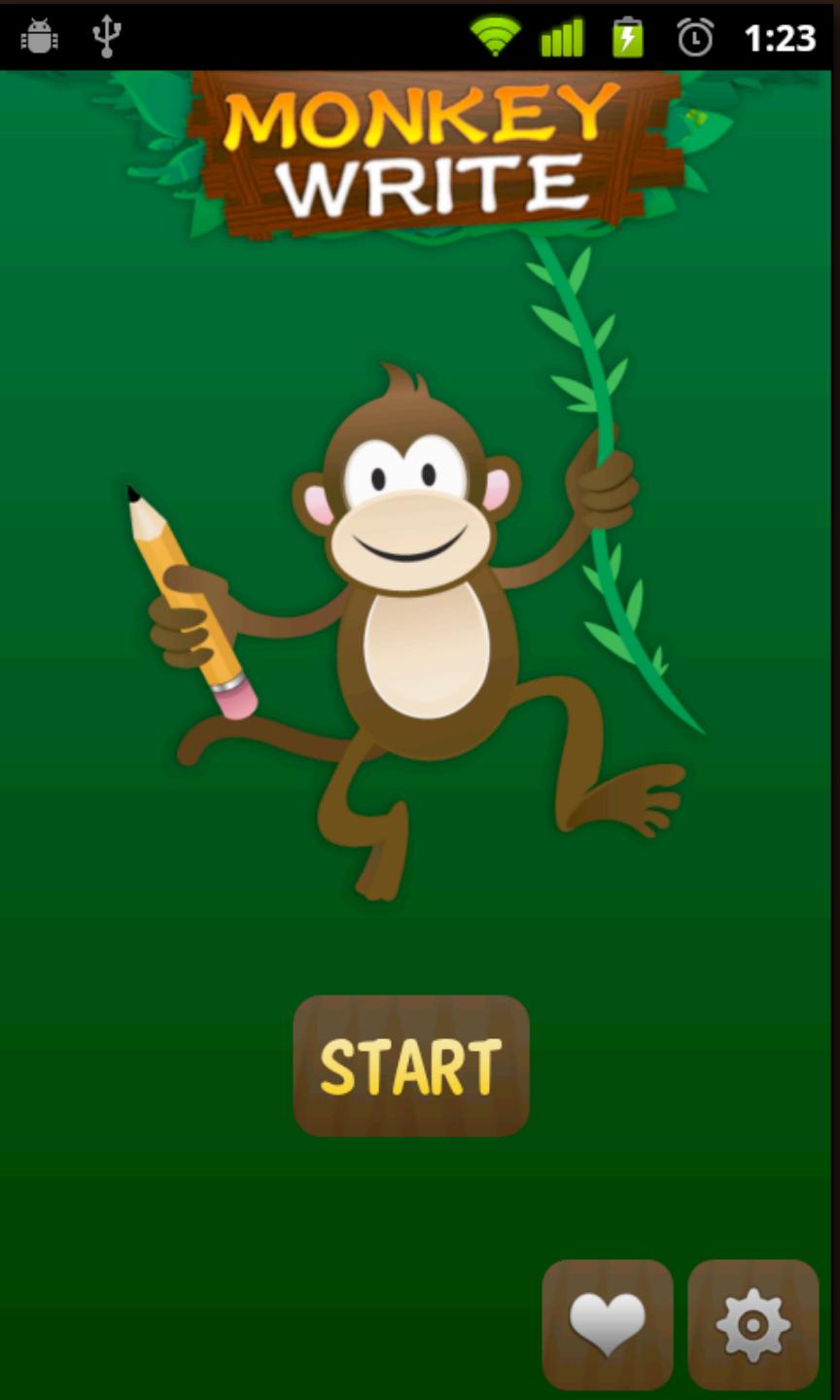
res/drawable-large-land/splash.png

@chiuki



res/drawable/splash.xml

@chiuki



res/drawable/splash.xml

# XML drawable

# XML shape

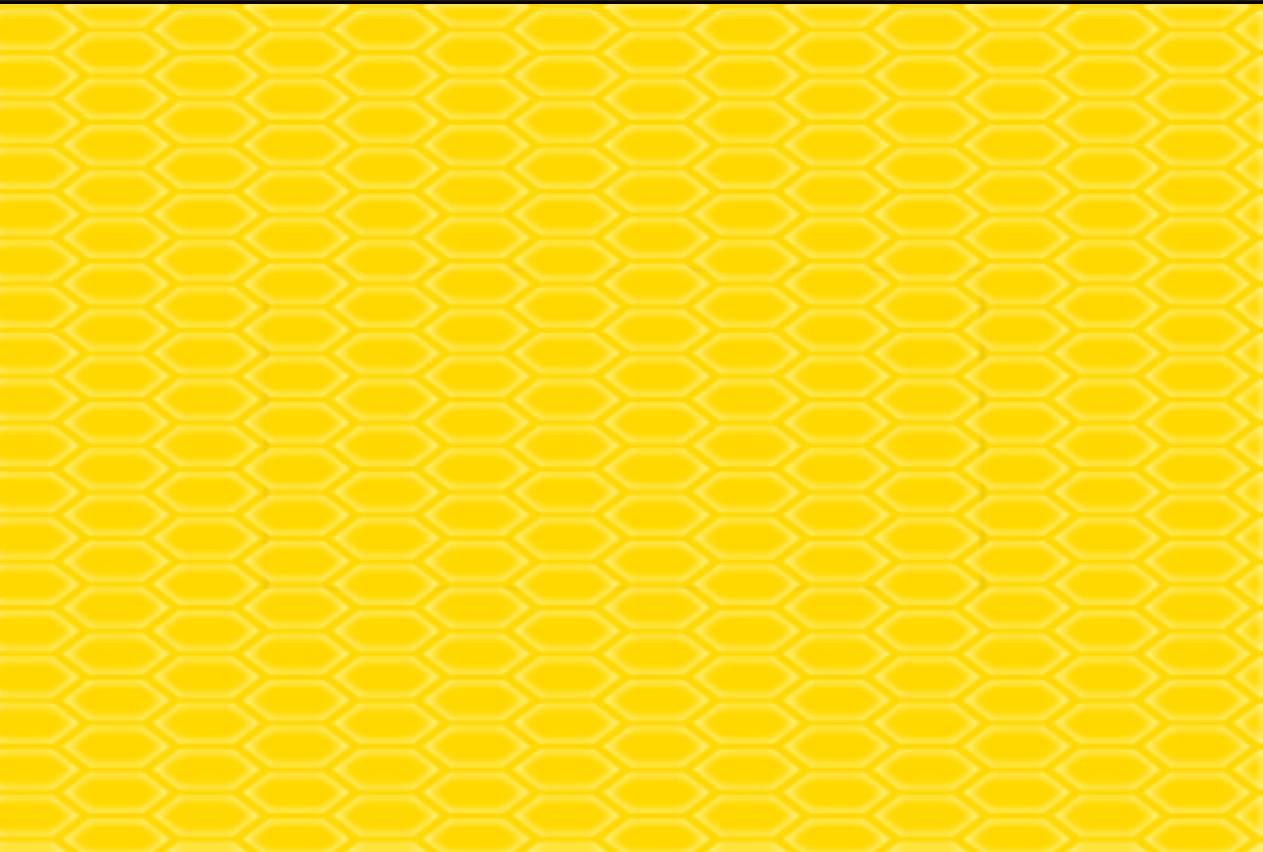
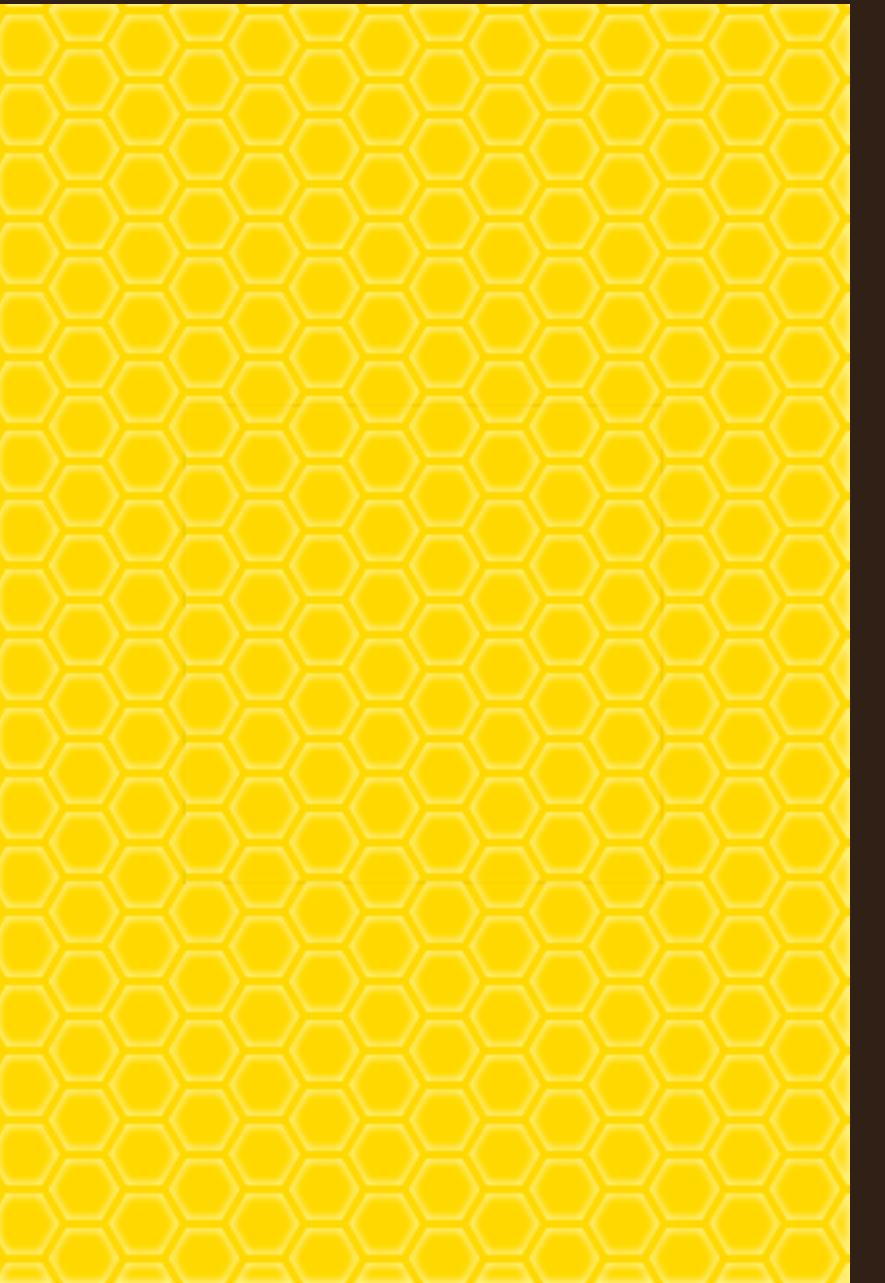
```
<shape android:shape="rectangle" >  
    <gradient  
        android:startColor="#063"  
        android:endColor="#030"  
        android:angle="270" />  
</shape>
```

# Gradient

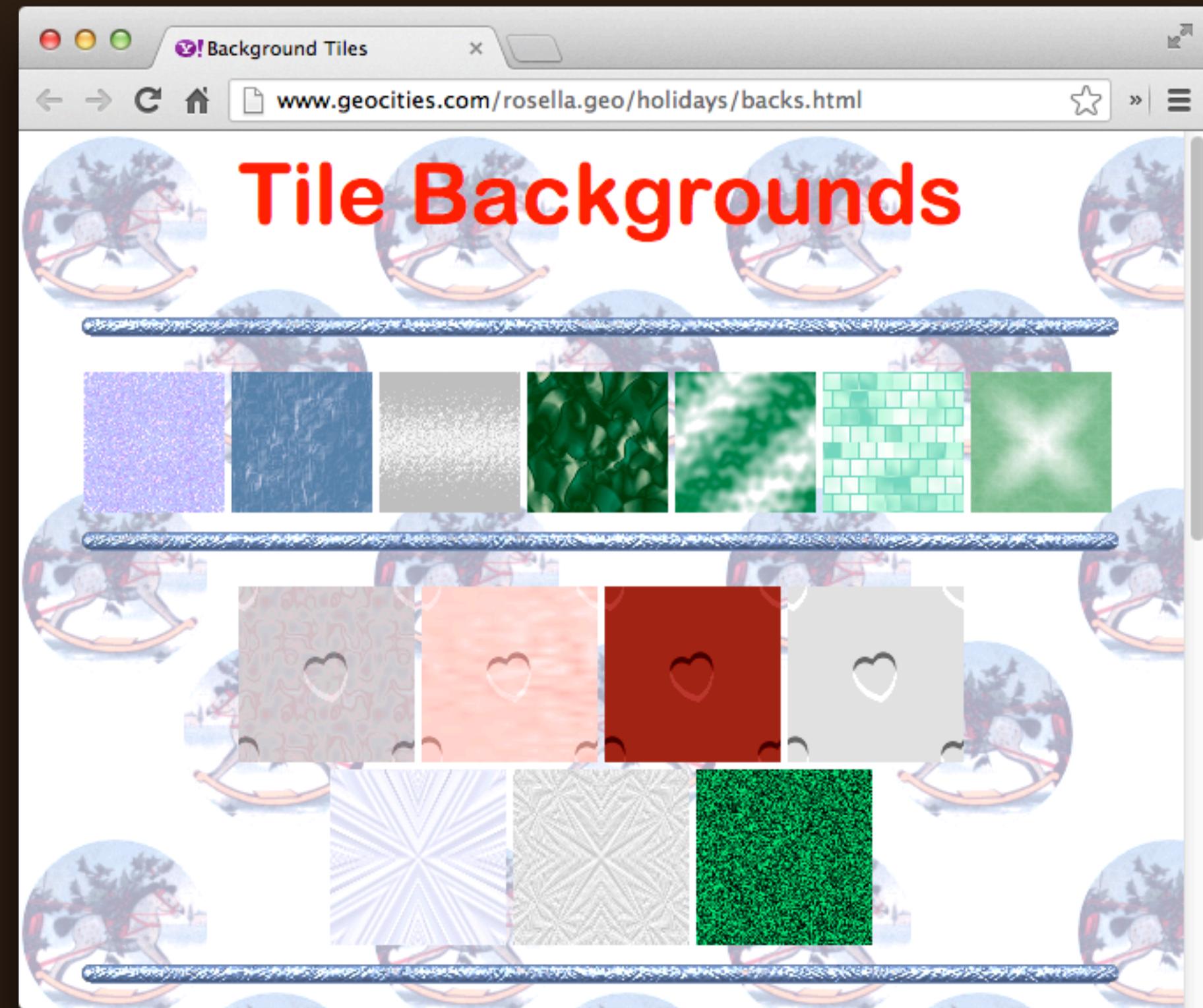


# Gradient

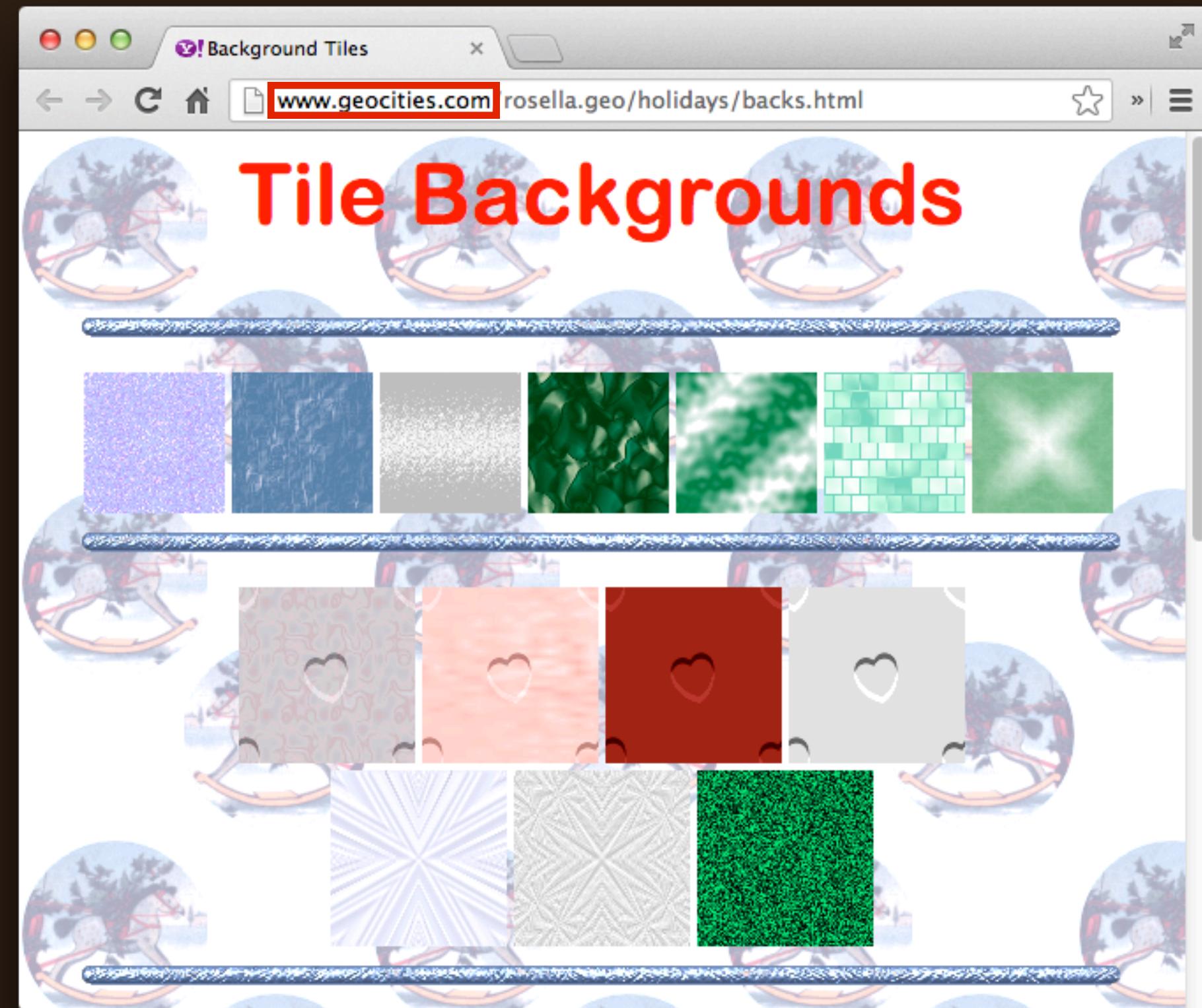
# Pattern



# Tiles



# Tiles



# Tiling background

```
<bitmap  
    android:src="@drawable/tile"  
    android:tileMode="repeat"  
    android:dither="true" />
```

res/drawable/background.xml

# Tiling background

```
<bitmap  
    android:src="@drawable/tile"       tile.png  
    android:tileMode="repeat"  
    android:dither="true" />
```

res/drawable/background.xml

# Tiling background

```
<bitmap  
    android:src="@drawable/tile"  
    android:tileMode="repeat"  
    android:dither="true" />
```

res/drawable/background.xml

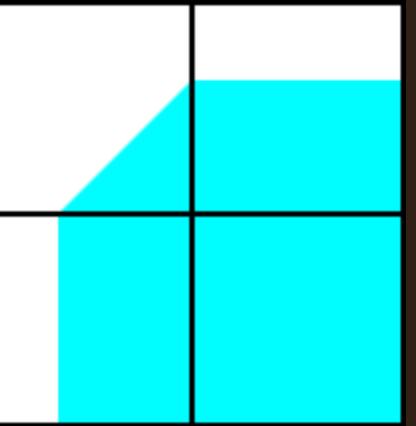
```
<View  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:background="@drawable/background" />
```

res/layout/activity\_main.xml

# Tile mode

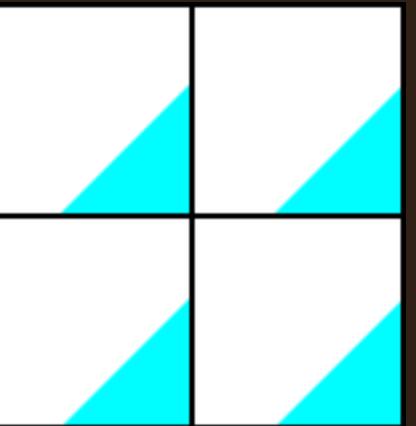
**clamp**

Replicates the edge color



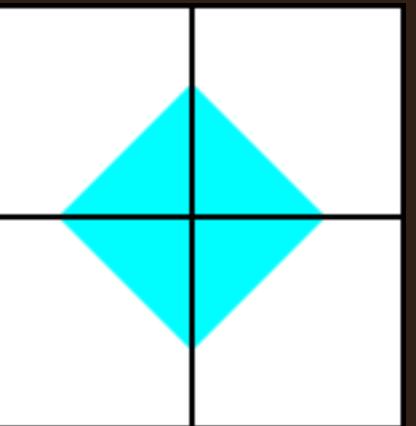
**repeat**

Repeats the bitmap in both direction



**mirror**

Repeats with alternating mirror images



# GridView



# Auto fit columns

```
<GridView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:numColumns="auto_fit"  
    android:columnWidth="@dimen/workbook_column_width"  
    android:padding="@dimen/workbook_padding"  
    android:horizontalSpacing="@dimen/workbook_spacing"  
    android:verticalSpacing="@dimen/workbook_spacing"  
    android:scrollbarStyle="outsideOverlay" />
```

# Auto fit columns

```
<GridView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:numColumns="auto_fit"  
    android:columnWidth="@dimen/workbook_column_width"  
    android:padding="@dimen/workbook_padding"  
    android:horizontalSpacing="@dimen/workbook_spacing"  
    android:verticalSpacing="@dimen/workbook_spacing"  
    android:scrollbarStyle="outsideOverlay" />
```

# Auto fit columns

```
<GridView  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:numColumns="auto_fit"  
    android:columnWidth="@dimen/workbook_column_width"  
    android:padding="@dimen/workbook_padding"  
    android:horizontalSpacing="@dimen/workbook_spacing"  
    android:verticalSpacing="@dimen/workbook_spacing"  
    android:scrollbarStyle="outsideOverlay" />
```

# dimen

```
<resources>
    <dimen name="workbook_padding">8dp</dimen>
    <dimen name="workbook_spacing">12dp</dimen>
    <dimen name="workbook_column_width">120dp</dimen>
<resources>
```

res/values/dimens.xml

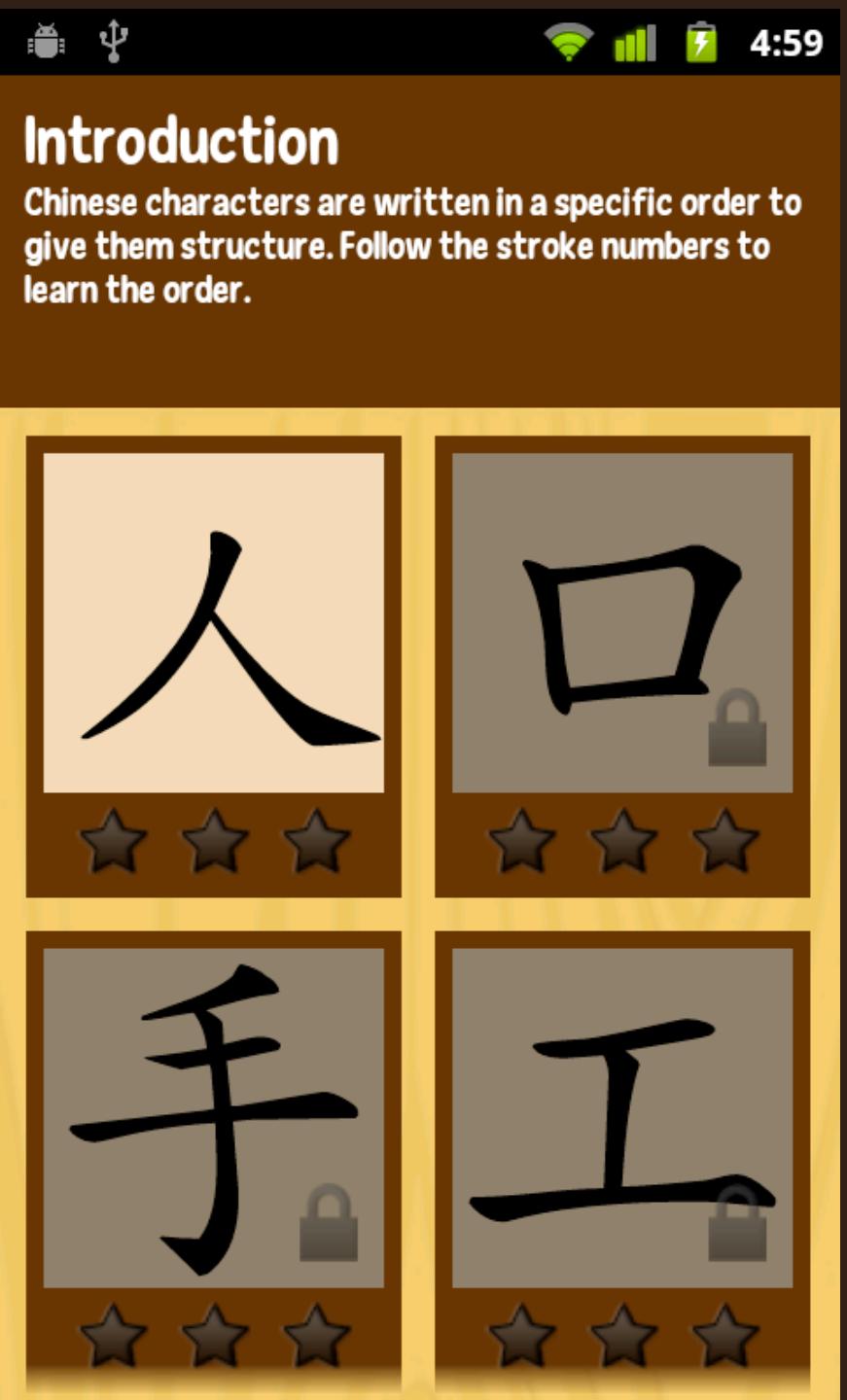
```
<resources>
    <dimen name="workbook_padding">32dp</dimen>
    <dimen name="workbook_spacing">18dp</dimen>
    <dimen name="workbook_column_width">180dp</dimen>
<resources>
```

res/values-large/dimens.xml

# 7-inch



# Phone



# Progressive

# Resource Folders

<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values drawable menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi Platform version: v4, v11, v14 UI mode: car, desk, television, appliance

# Resource Folders

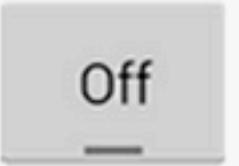
<http://developer.android.com/guide/topics/resources/providing-resources.html>

Type	Variation
layout values drawable menu	Language & Region: en, fr, fr-rCA Screen size: small, large, xlarge Screen orientation: port, land Screen density: ldpi, mdpi, hdpi, xhdpi, nodpi, tvdpi <b>Platform version: v4, v11, v14</b> UI mode: car, desk, television, appliance

# Toggle button

A toggle button allows the user to change a setting between two states.

You can add a basic toggle button to your layout with the [ToggleButton](#) object. Android 4.0 (API level 14) introduces another kind of toggle button called a switch that provides a slider control, which you can add with a [Switch](#) object.



*Toggle buttons*



*Switches (in Android 4.0+)*

The [ToggleButton](#) and [Switch](#) controls are subclasses of [CompoundButton](#) and function in the same manner, so you can implement their behavior the same way.

# layout-v14

```
<ToggleButton  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

res/layout/compound\_button.xml

```
<Switch  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content" />
```

res/layout-v14/compound\_button.xml

# Include

```
<LinearLayout  
    android:layout_width="match_parent"  
    android:layout_height="match_parent"  
    android:orientation="horizontal" >  
    <include  
        android:id="@+id/turbo"  
        layout="@layout/compound_button" />  
    <include  
        android:id="@+id/reset"  
        layout="@layout/compound_button" />  
</LinearLayout>
```

# Cast to superclass

```
CompoundButton turbo = (CompoundButton) findViewById(R.id.turbo);
if (!turbo.isChecked()) {
    // Slow down the computer
}
```

# API level

[android.com/reference/android/app/ActivityManager.html](http://android.com/reference/android/app/ActivityManager.html)

**public int getMemoryClass ()** Added in API level 5

Return the approximate per-application memory class of the current device. This gives you an idea of how hard a memory limit you should impose on your application to let the overall system work best. The returned value is in megabytes; the baseline Android memory class is 16 (which happens to be the Java heap limit of those devices); some device with more memory may return 24 or even higher numbers.

# Detect API level

`android.os.Build.VERSION.SDK`  
String, available from API level 1

`android.os.Build.VERSION.SDK_INT`  
int, available from API level 4

# Progressive

```
// Default value  
int heapSize = 16;  
  
// Detect OS version  
int sdk = Integer.parseInt(Build.VERSION.SDK);  
  
// Progressive enhancement for newer devices  
if (sdk >= Build.VERSION_CODES_ECLAIR) {  
    // Static helper for API level 4 and below  
    heapSize = HeapSizeGetter.getHeapSize(this);  
}
```

# Static helper

```
public abstract class HeapSizeGetter {  
    public static int getHeapSize(Context context) {  
        ActivityManager activityManager = (ActivityManager)  
            context.getSystemService(Context.ACTIVITY_SERVICE);  
        return activityManager.getMemoryClass();  
    }  
}
```

# Progressive

```
// Default value  
int heapSize = 16;  
  
// Detect OS version  
int sdk = Integer.parseInt(Build.VERSION.SDK);  
  
// Progressive enhancement for newer devices  
if (sdk >= Build.VERSION_CODES_ECLAIR) {  
    // Static helper for API level 4 and below  
    heapSize = HeapSizeGetter.getHeapSize(this);  
}
```

# Support libraries

Ship new functionality with your app!

# Official Support Library

Fragment

ViewPager

SlidingPaneLayout

DrawerLayout

Loader

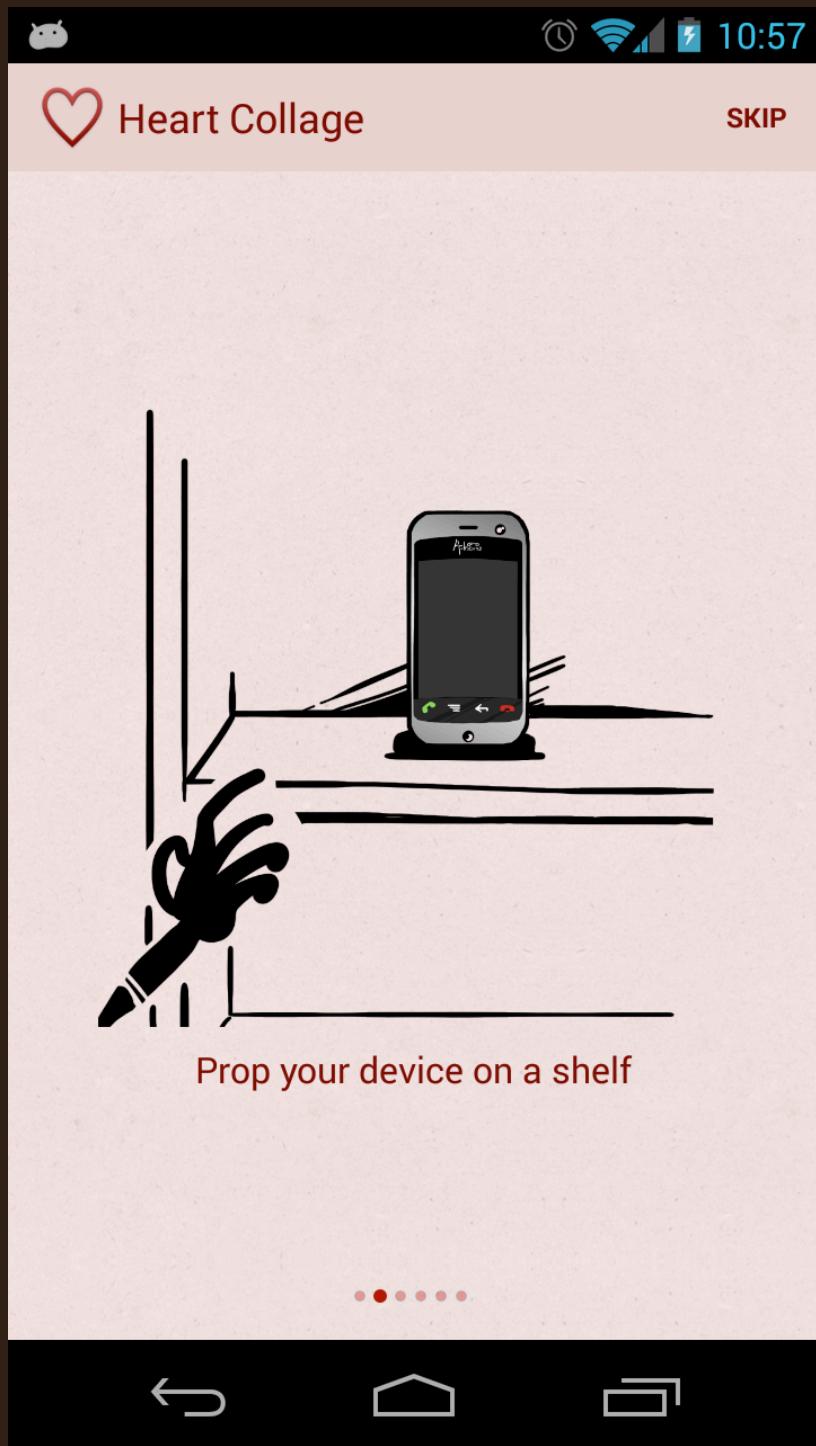
LruCache

TaskStackBuilder

# Community Support Libraries

- Action Bar Sherlock
- View Pager Indicator
- Nine Old Android
- Holo Everywhere

# View Pager Indicator



# Summary

# Summary

Responsive layout

Declarative layout

Density-independent pixels (dp)

Resource folders

XML drawables

Progressive functionality

Resource folders

Version check

Support libraries

# Thank you!

Learn more

<http://is.gd/FluidAndroidLayouts>

<http://is.gd/BeautifulAndroid>

Stay in touch

<http://eepurl.com/lR5uD>

<http://blog.sqisland.com>

<http://twitter.com/chiuki>