Android

UI Basics

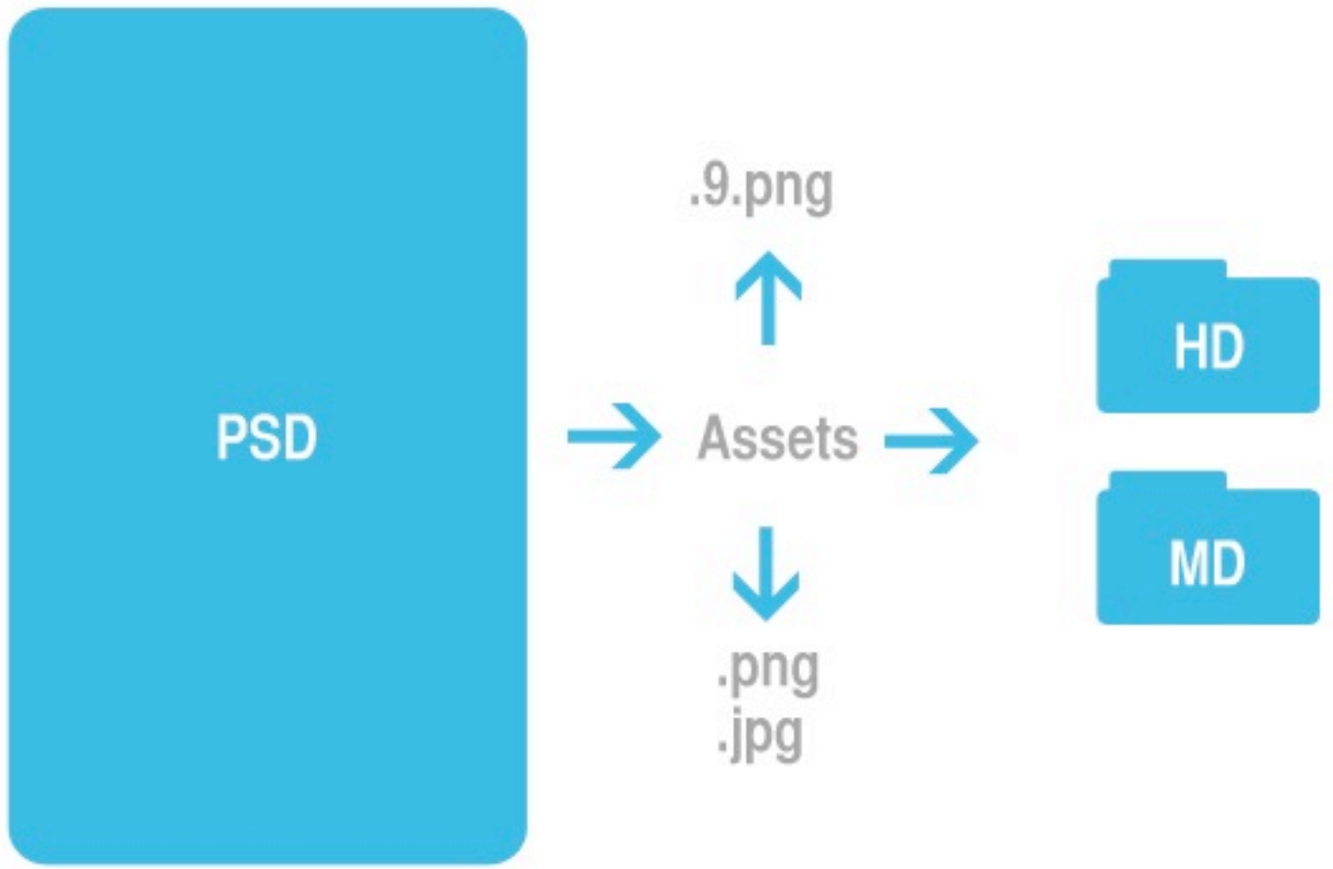
Why split the UI and programing tasks for a Android AP

The most convenient and maintainable way to design application user interfaces is by creating XML layout resources.

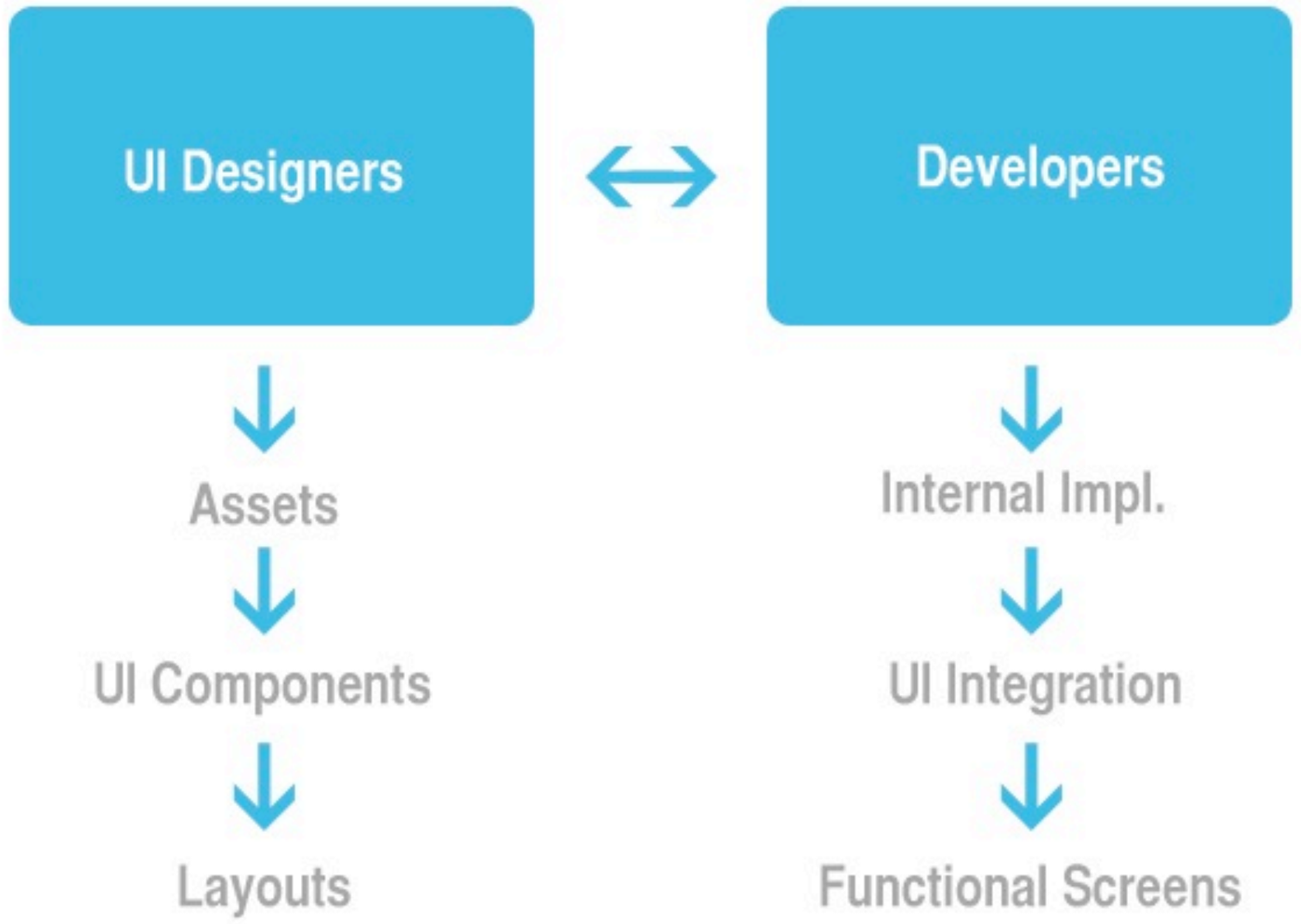
This method greatly simplifies the UI design process, moving much of the static creation and layout of user interface controls and definition of control attributes, to the XML, instead of littering the code. It creates a potential distinction between UI designers (who concern themselves more with layout) and developers (who know Java and implement application functionality).

Developers can still alter the content of a screen programmatically when necessary. Complex controls, like ListView or GridView, are usually populated with data programmatically.

Preparation of Assets - Why?



Who does what?



What is a Layout?

A type of resource that defines what is drawn on the screen. Layout resources are stored as XML files in the /res/layout resource directory for the application. A layout resource is simply a template for a user interface screen, or portion of a screen, and contain.

A type of View class whose primary purpose is to organize other controls. These layout classes (LinearLayout, RelativeLayout, TableLayout, etc.) are used to display child controls, such as text controls or buttons or images on the screen.

Is the interface finished?

- Identify Complex components
- Adjust Interface components

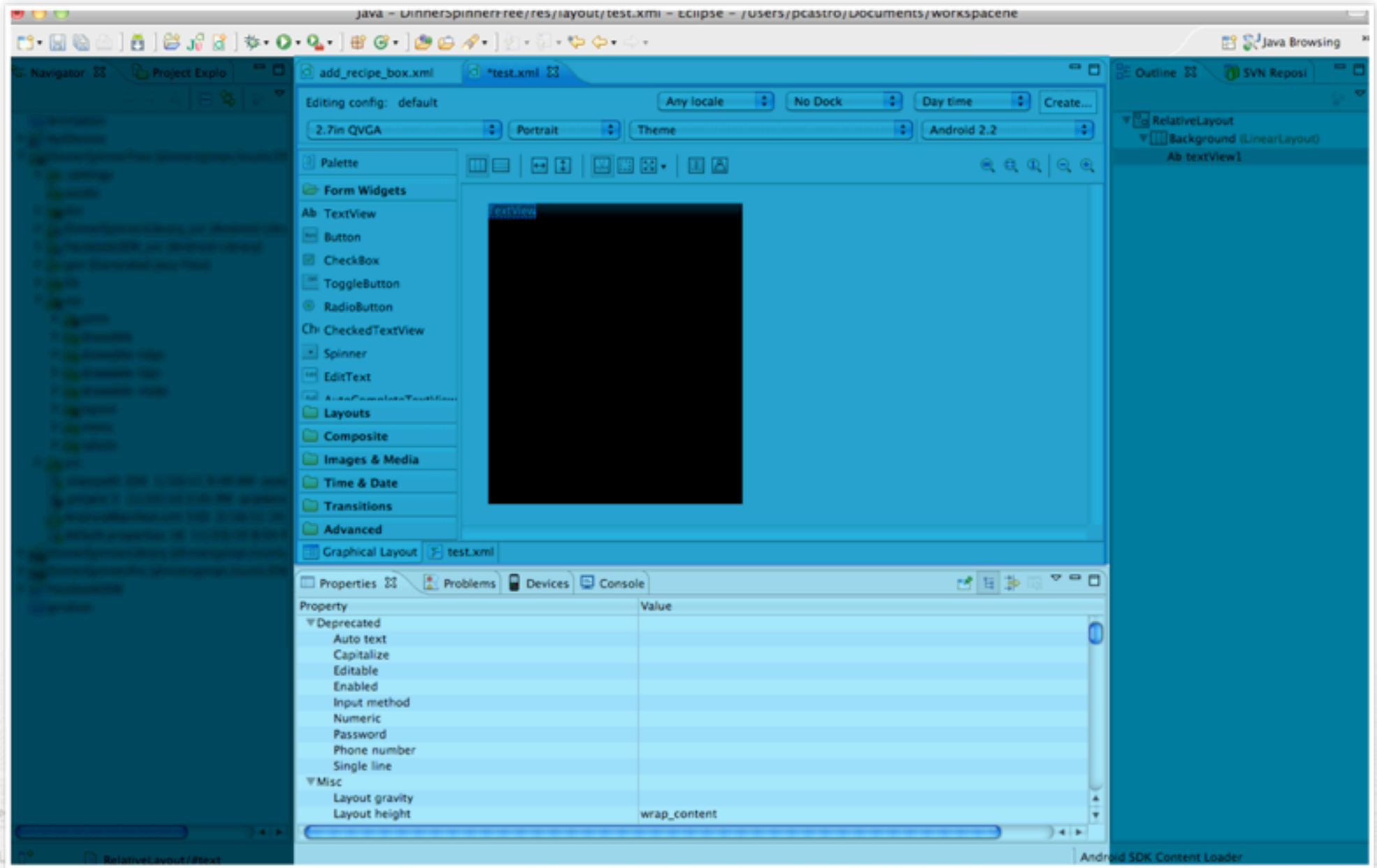
Is the interface finished?

Not Yet!

- Identify Complex components
- Adjust Interface components

Eclipse to Design Layout Resources

Package Explorer	Outline
Layout/ XML View	Properties



Project Folder Structure

Some tips for working with the layout resource designer in Eclipse:

- Use the Outline panel to Add and Remove controls to your layout resource.
- Select a specific control (either in the Preview or the Outline) and use the Property panel to adjust a specific control's attributes.
- Use the XML tab to edit the XML definition directly.

It's important to remember that the Eclipse layout resource designer preview can't replicate exactly how the layout will appear to end users.

Also, certain "complex" controls, including tabs or video viewers, cannot be previewed within Eclipse.

XML Layout Resource

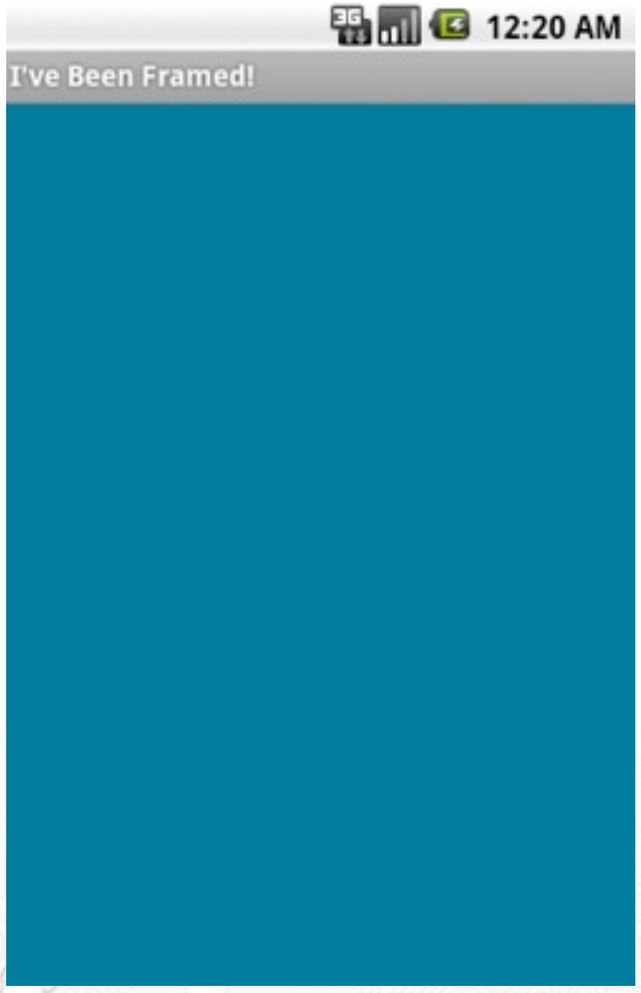
XML layout resources must be stored in the /res/layout project directory (or, in the case of alternative resources, in a specially named sub-directory).

```
1.  <?xml version="1.0" encoding="utf-8"?>
2.  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3.      android:orientation="vertical"
4.      android:layout_width="fill_parent"
5.      android:layout_height="fill_parent"
6.      android:gravity="center">
7.      <TextView
8.          android:layout_width="fill_parent"
9.          android:id="@+id/PhotoLabel"
10.         android:layout_height="wrap_content"
11.         android:text="@string/my_text_label"
12.         android:gravity="center_horizontal"
13.         android:textSize="20dp" />
14.     <ImageView
15.         android:layout_width="wrap_content"
16.         android:layout_height="wrap_content"
17.         android:src="@drawable/matterhorn"
18.         android:adjustViewBounds="true"
19.         android:scaleType="fitXY"
20.         android:maxHeight="250dp"
21.         android:maxWidth="250dp"
22.         android:id="@+id/Photo" />
23. </LinearLayout>
```

FrameLayout

Designed to display a stack of child View controls. Multiple view controls can be added to this layout. This can be used to show multiple controls within the same screen space.

Generally used to display only one view, or views which overlap.

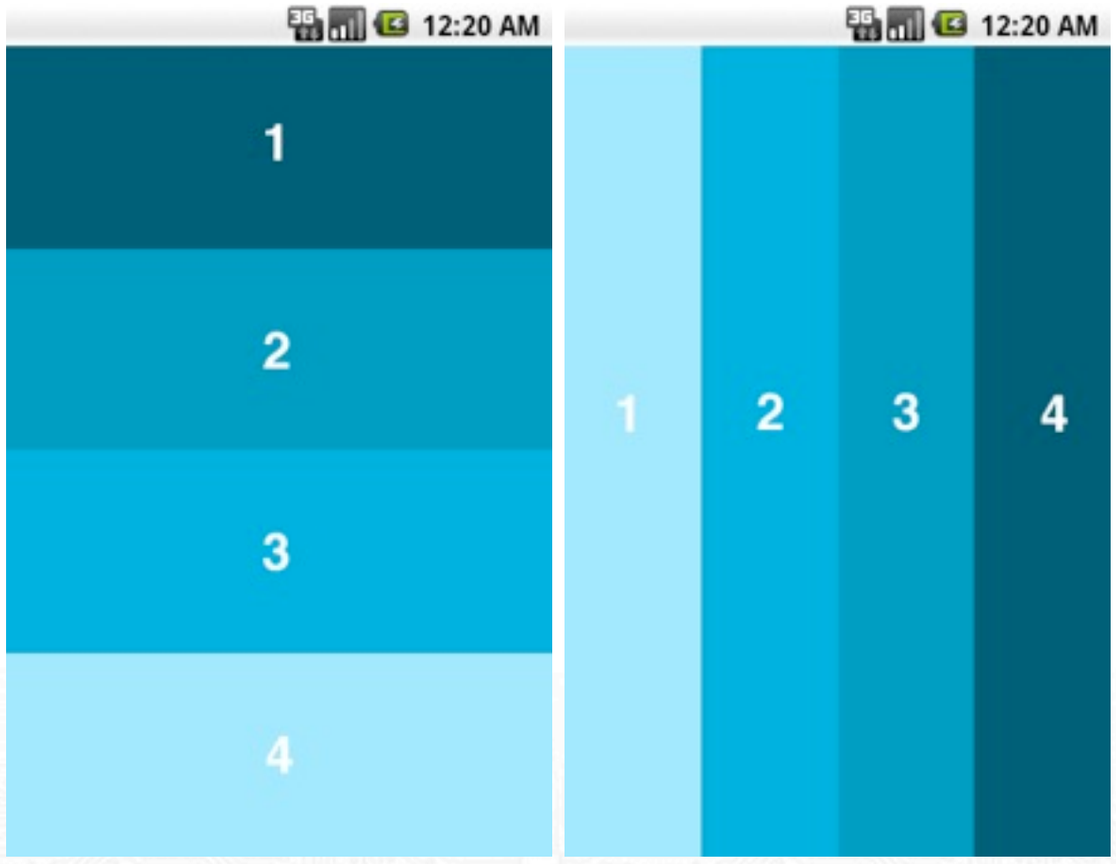



```xml
<?xml version="1.0" encoding="utf-8"?>
<FrameLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <ImageView
        android:id="@+id/ImageView01"
        android:layout_height="fill_parent"
        android:layout_width="fill_parent"
        android:src="@drawable/lake"
        android:scaleType="matrix"></ImageView>
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#000"
        android:textSize="40dp"
        android:text="@string/top_text" />
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/bottom_text"
        android:layout_gravity="bottom"
        android:gravity="right"
        android:textColor="#fff"
        android:textSize="50dp" />
</FrameLayout>
```

Linear Layout

The linear layout works much as its name implies: it organizes controls linearly in either a vertical or horizontal.

When the layout's orientation is set to vertical, all child controls within it are organized in a single column; when the layout's orientation is set to horizontal, all child controls within it are organized in a single row.

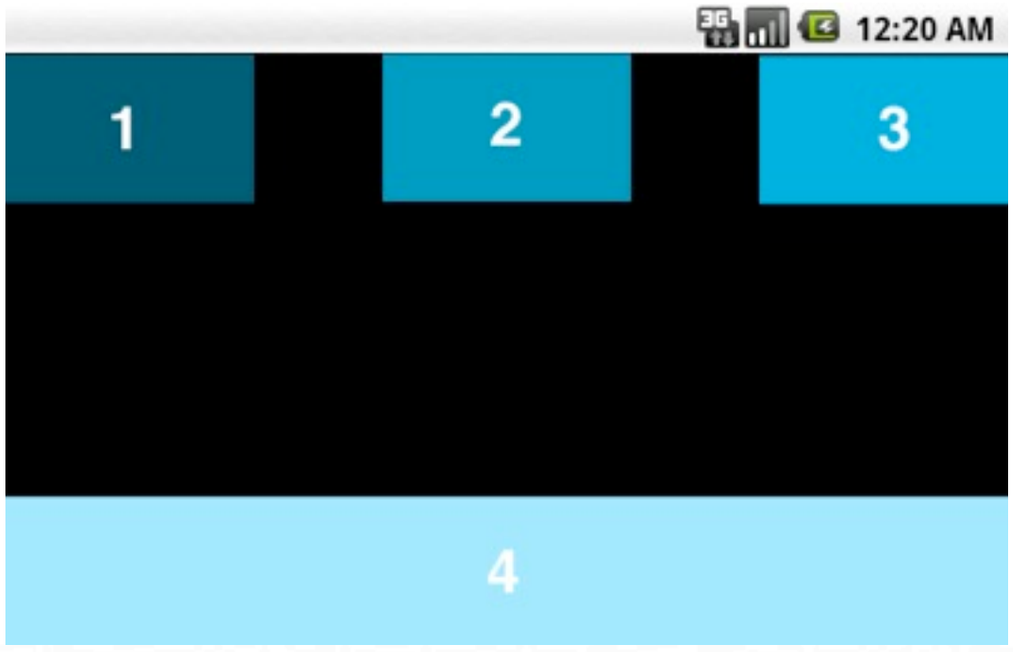



```xml
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent" android:layout_height="fill_parent"
    android:orientation="vertical">
    <TextView android:text="RED" android:id="@+id/TextView01"
        android:layout_height="wrap_content" android:background="#f00"
        android:layout_width="fill_parent" android:layout_weight=".14"
        android:gravity="center" android:textColor="#000"></TextView>
    <TextView android:text="ORANGE" android:id="@+id/TextView02"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".15" android:background="#ffa500"
        android:gravity="center" android:textColor="#000"></TextView>
    <TextView android:text="YELLOW" android:id="@+id/TextView03"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".14" android:background="#ffff00"
        android:gravity="center" android:textColor="#000"></TextView>
    <TextView android:text="GREEN" android:id="@+id/TextView04"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".15" android:background="#0f0" android:gravity="center"
        android:textColor="#000"></TextView>
    <TextView android:text="BLUE" android:id="@+id/TextView05"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".14" android:background="#00f" android:gravity="center"
        android:textColor="#fff"></TextView>
    <TextView android:text="INDIGO" android:id="@+id/TextView06"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".14" android:background="#4b0082"
        android:gravity="center" android:textColor="#fff"></TextView>
    <TextView android:text="VIOLET" android:id="@+id/TextView07"
        android:layout_height="wrap_content" android:layout_width="fill_parent"
        android:layout_weight=".14" android:background="#ee82ee"
        android:gravity="center" android:textColor="#000"></TextView>
</LinearLayout>
```

Relative Layout

The relative layout works much as its name implies: it organizes controls relative to one another, or to the parent control itself.

It means that child controls, such as ImageView, TextView ,and Button controls, can be placed above, below, to the left or right, of one another.




```xml
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_height="fill_parent"
    android:layout_width="fill_parent">
    <EditText
        android:id="@+id/EditText01"
        android:hint="Enter some text..."
        android:layout_alignParentLeft="true"
        android:layout_width="fill_parent"
        android:layout_toLeftOf="@+id/Button01"
        android:layout_height="wrap_content"></EditText>
    <Button
        android:id="@+id/Button01"
        android:text="Press Here!"
        android:layout_width="wrap_content"
        android:layout_alignParentRight="true"
        android:layout_height="wrap_content"></Button>
</RelativeLayout>
```

Table Layout

A grid of made up of rows and columns, where a cell can display a view control.


```xml
<?xml version="1.0" encoding="utf-8"?>
<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/tableLayout1"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:shrinkColumns="*"
    android:stretchColumns="*">
    <TableRow
        android:id="@+id/tableRow4"
        android:layout_height="wrap_content"
        android:layout_width="match_parent"
        android:gravity="center_horizontal">
        <TextView
            android:id="@+id/textView9"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:textStyle="bold"
            android:typeface="serif"
            android:textSize="18dp"
            android:text="Weather Table"
            android:gravity="center"
            android:layout_span="6"></TextView>
    </TableRow>
```

Do's and Don'ts

Don'ts

Don't create rigid absolute-positioned layouts

Don't use px units use dp or sp por text

Don't use small font sizes

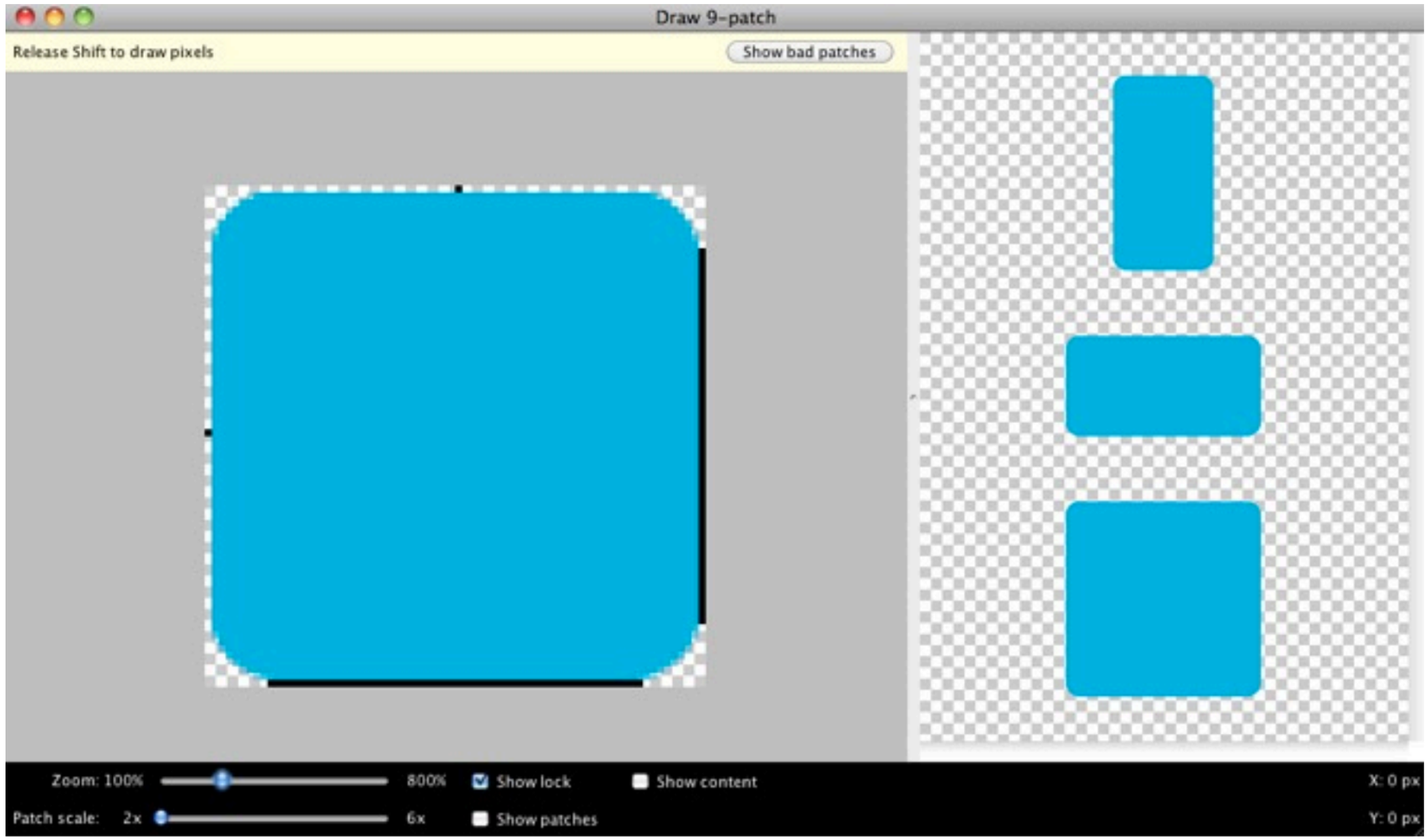
Do's

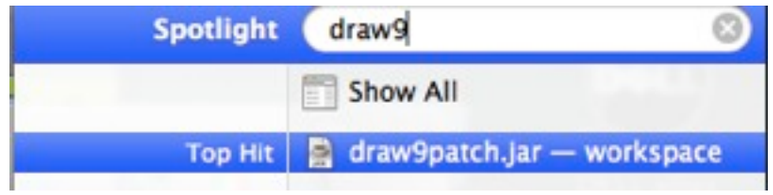
Do create versions of all resources for high density screens

Do support trackball navigation

Do properly handle orientation changes

Nine-Patch


Draw 9-patch
Release Shift to draw pixels
Show bad patches
Zoom: 100%
800%
Show lock
Show content
Patch scale:
2x
6x
Show patches
X: 0 px
Y: 0 px


Spotlight
draw9
Show All
Top Hit
draw9patch.jar — workspace

Buttons

▶ drawable-hdpi

foo_default.png

foo_focused.png

foo_pressed.png

▶ drawable

foo.xml:

```
<selector>
    <item android:drawable="@drawable/foo_disabled"
          android:state_enabled="false" ... />
    <item android:drawable="@drawable/foo_pressed"
          android:state_pressed="true" ... />
    <item android:drawable="@drawable/foo_focused"
          android:state_focused="true" ... />
    <item android:drawable="@drawable/foo_default" />
</selector>
```

Installing Eclipse :)..... :(

1.http://developer.android.com/sdk/installing.html

First application with Android (Hello World)

2.http://www.maestrosdelweb.com/editorial/descubriendo-android-con-el-hello-world/

Some Links

1.http://www.youtube.com/watch?v=oN9b8KP6pVU

2.http://code.google.com/p/android-ui-utils/

Installing Eclipse :)..... :(

1.http://developer.android.com/sdk/installing.html

First application with Android (Hello World)

2.http://www.maestrosdelweb.com/editorial/descubriendo-android-con-el-hello-world/

Some Links

1.http://www.youtube.com/watch?v=oN9b8KP6pVU

2.http://code.google.com/p/android-ui-utils/

Thank you!