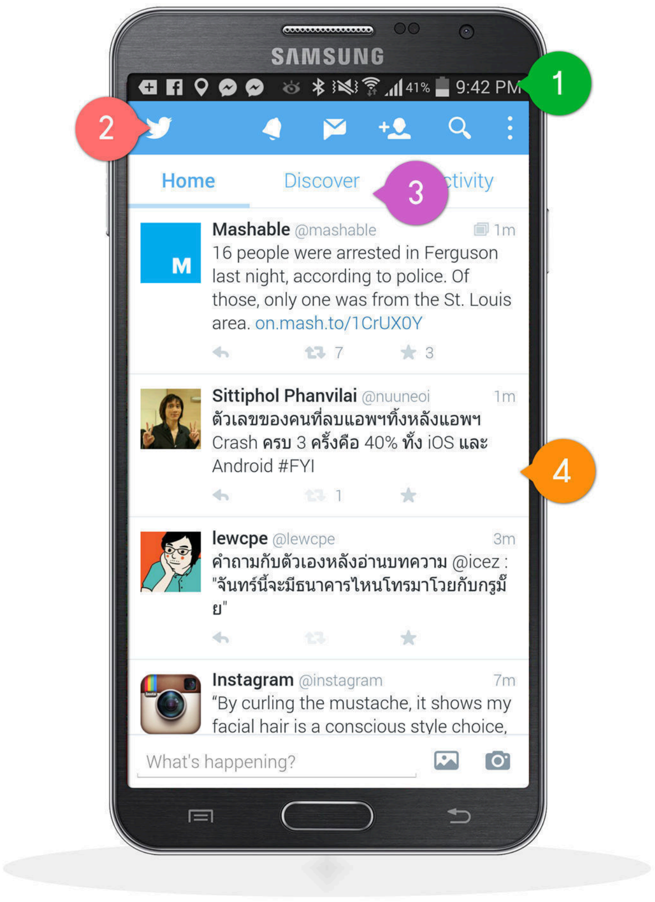
The android Cheese Sheet

by THE CHEESE FACTORY



Mashable @mashable 1m
16 people were arrested in Ferguson last night, according to police. Of those, only one was from the St. Louis area. on.mash.to/1CrUX0Y
7 3

Sittiphol Phanvilai @nuuneoi 1m
ตัวเลขของคนที่ลบแอพฯทิ้งหลังแอพฯ Crash ครบ 3 ครั้งคือ 40% ทั้ง iOS และ Android #FYI
1

lewcpe @lewcpe 3m
คำถามกับตัวเองหลังอ่านบทความ @icez : "จันทร์นี้จะมีธนาคารไหนโทรมาโวยกับกรูมั้ย"

Instagram @instagram 7m
"By curling the mustache, it shows my facial hair is a conscious style choice,

#1 Status Bar

Show Device's Status and Notification
Can be hidden if needed

#2 Action Bar

Up Icon, App Icon and Menu
Standard UX by Android's Guideline

#3 Tab Bar

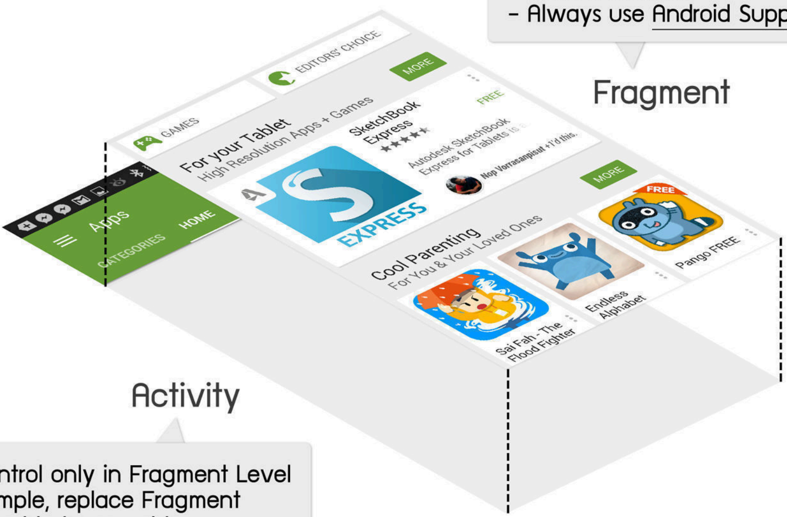
Tab Style Navigation
Action Bar's Tab is deprecated, use SlidingTabLayout instead

#4 Content Area

Show main contents of application
Let it always be scrollable

DESIGN PATTERN

"Fragmentalize Everything"

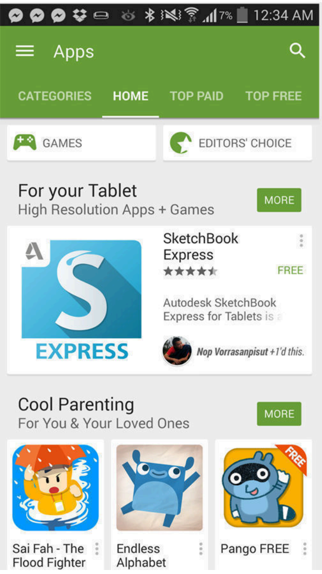


- Put UI and Logic here not Activity
- Make it dependent to Activity
- Make it reusable
- Always use Android Support Library

Fragment

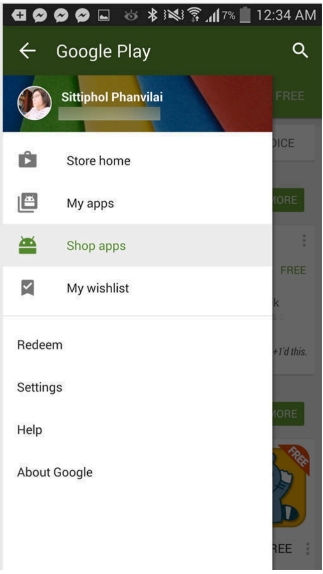
Activity

- Do control only in Fragment Level for example, replace Fragment
- No need to be reusable
- Always use ActionBarActivity

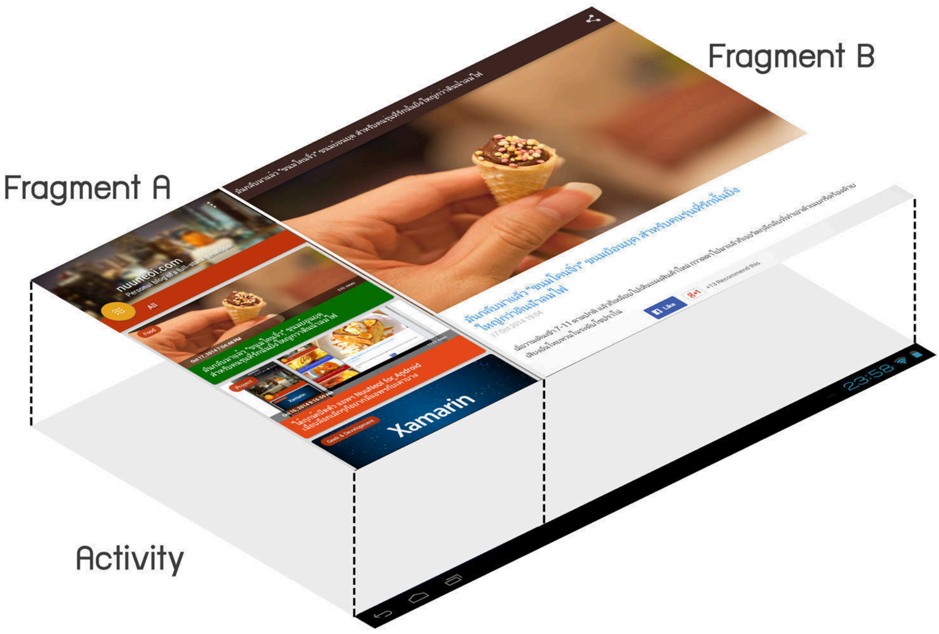

12:34 AM
Apps
CATEGORIES
HOME
TOP PAID
TOP FREE
GAMES
EDITORS' CHOICE
For your Tablet
High Resolution Apps + Games
MORE
SketchBook Express
FREE
Autodesk SketchBook Express for Tablets is
Nop Vorrasanpisut +1'd this.
Cool Parenting
For You & Your Loved Ones
MORE
Sai Fah - The Flood Fighter
Endless Alphabet
Pango FREE

Put frequently used Fragment in Tab Bar (SlidingTabLayout)

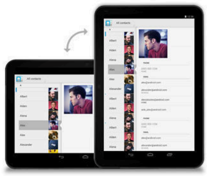
Put occasionally used Fragment in Drawer Menu (DrawerLayout)

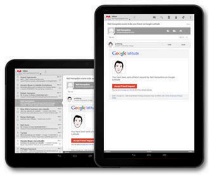


DESIGN PATTERN - TABLET



4 WAYS TO ROTATE UI





DIMENSION

DP unit

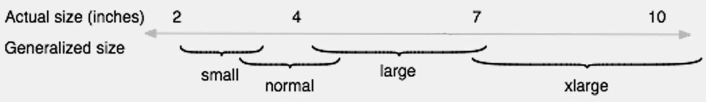
$$px = dp * (dpi / 160)$$

$$px = dp * scaleFactor$$

Densities

ldpi (low) ~120dpi	x0.75
mdpi (medium) ~160dpi	x1
hdpi (high) ~240dpi	x1.5
xhdpi (extra-high) ~320dpi	x2
xxhdpi (extra-extra-high) ~480dpi	x3
xxxhdpi (extra-extra-extra-high) ~640dpi	x4

Screen Size

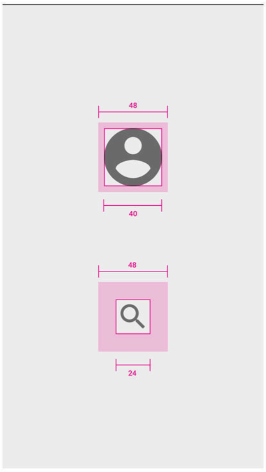


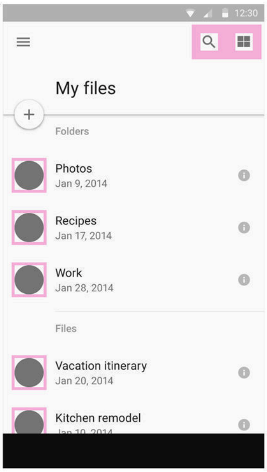
~~Never use px~~ Always use dp for width/height Use sp for Font Size

MAGIC NUMBER

Title Bar's Height = 56dp Smallest Width 600dp = Tablet

TOUCH TARGET SIZE





My files

Folders

Photos
Jan 9, 2014

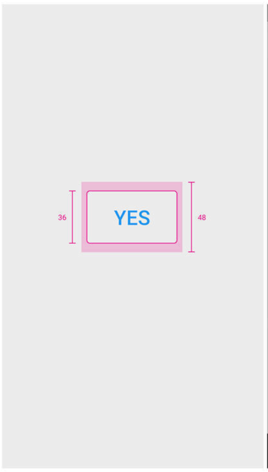
Recipes
Jan 17, 2014

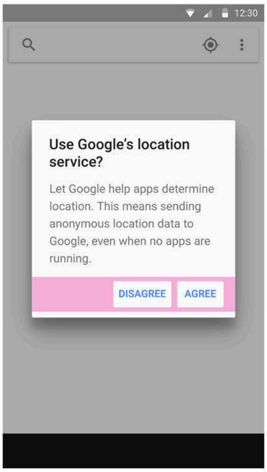
Work
Jan 28, 2014

Files

Vacation itinerary
Jan 20, 2014

Kitchen remodel
Jan 10, 2014



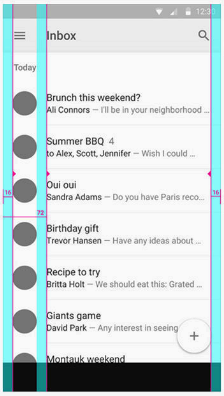


Use Google's location service?

Let Google help apps determine location. This means sending anonymous location data to Google, even when no apps are running.

DISAGREE AGREE

KEYLINE & METRIC

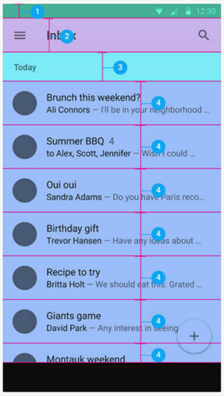


Brunch this weekend?
Ali Connors — I'll be in your neighborhood ...
Summer BBQ 4
to Alex, Scott, Jennifer — Wish I could ...
Oui oui
Sandra Adams — Do you have Paris reco...
Birthday gift
Trevor Hansen — Have any ideas about ...
Recipe to try
Britta Holt — We should eat this: Grated ...
Giants game
David Park — Any interest in seeing ...
Montauk weekend

Vertical keylines and horizontal margins

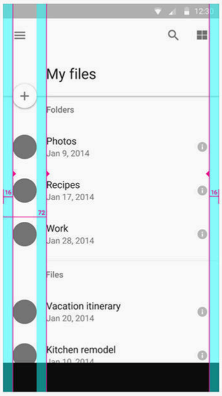
Vertical keyline at 16dp from the left and right edges. Content associated with an icon or avatar aligns 72dp from the left edge.

16dp horizontal margins on mobile.



Vertical spacing

1. 24dp
2. 56dp
3. 48dp
4. 72dp

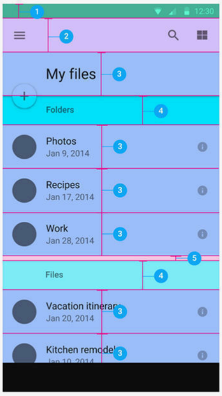


Photos
Jan 9, 2014
Recipes
Jan 17, 2014
Work
Jan 28, 2014
Vacation itinerary
Jan 20, 2014
Kitchen remodel
Jan 10, 2014

Vertical keylines and horizontal margins

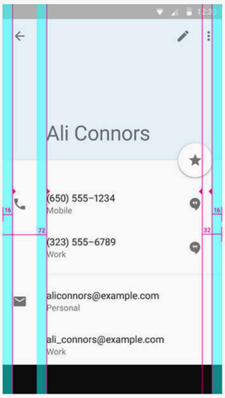
Vertical keyline for icons at 16dp from the left and right edges. Content associated with an icon or avatar aligns 72dp from the left edge.

16dp horizontal margins on mobile.



Vertical spacing

1. 24dp
2. 56dp
3. 72dp
4. 48dp
5. 8dp

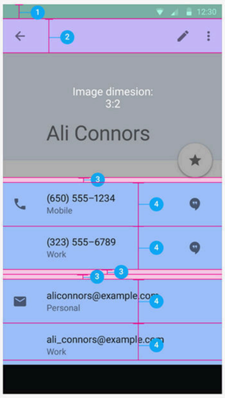


Ali Connors
(650) 555-1234
Mobile
(323) 555-6789
Work
aliconnors@example.com
Personal
ali_connors@example.com
Work

Vertical keylines and horizontal margins

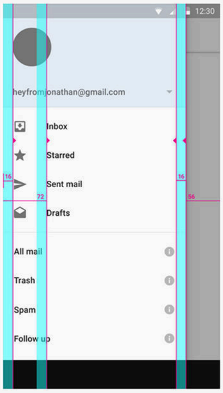
Vertical keyline for icons at 16dp from the left edge. Content associated with an icon or avatar aligns 72dp from the left edge. An extra keyline is added 32dp from the right edge to allow the floating action button to align with the icons below.

16dp horizontal margins on mobile.



Vertical Spacing

1. 24dp
2. 56dp
3. 8dp
4. 72dp

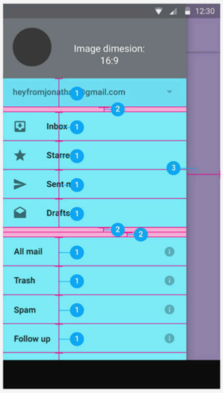


Vertical keylines and horizontal margins

Vertical keylines for icons at 16dp from the left and right edges of the side nav.

Content associated with an icon or avatar aligns 72dp from the left edge of the side nav.

The width of the side nav is equal to the width of the screen minus the height of the action bar, or in this case 56dp from the right edge of the screen.



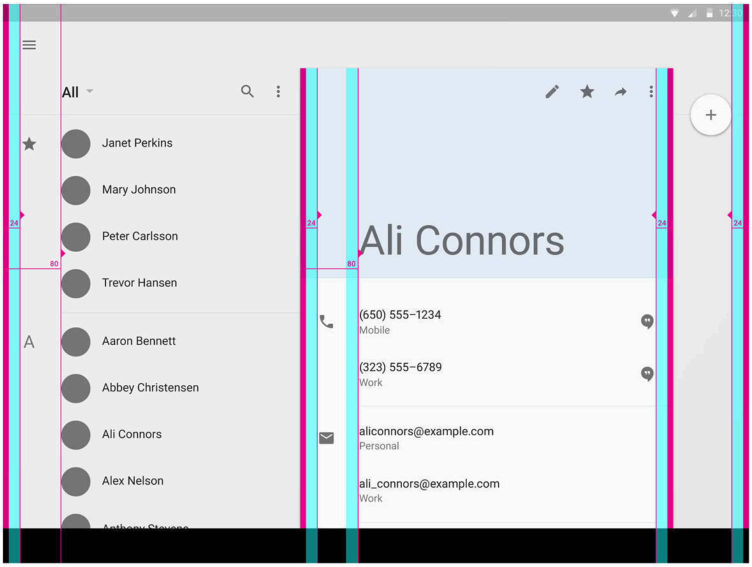
Vertical spacing

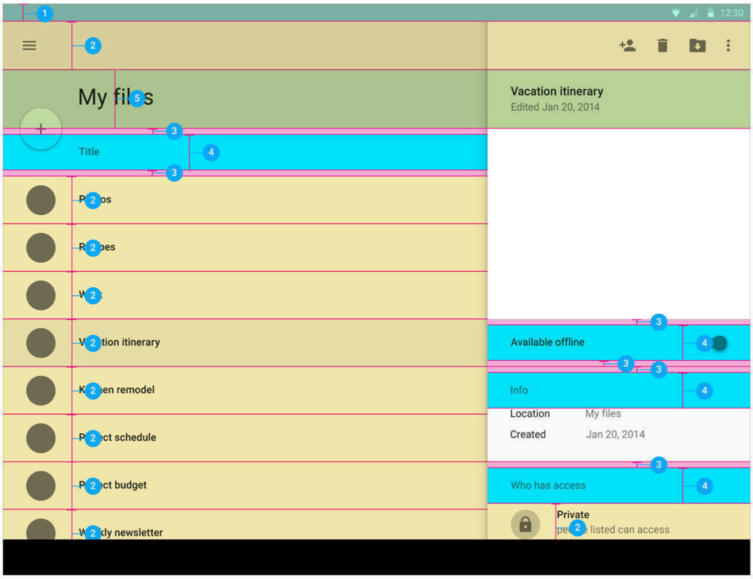
1. 48dp
2. 8dp
3. 56dp

Material Design

More in http://www.google.com/design/spec/layout/metrics-keylines.html

KEYLINE & METRIC: TABLET





1. 24dp
2. 64dp
3. 8dp
4. 48dp
5. 80dp

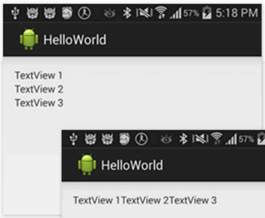
Material Design

More in http://www.google.com/design/spec/layout/metrics-keylines.html

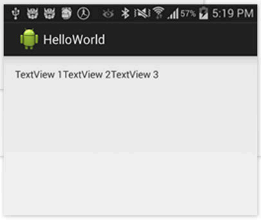
LAYOUT

LinearLayout

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="[horizontal|vertical]"
    >
</LinearLayout>
```



TextView 1
TextView 2
TextView 3

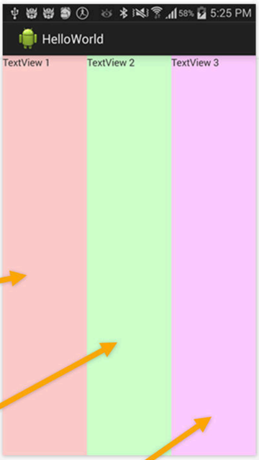


TextView 1TextView 2TextView 3

LinearLayout with weight

Help you distribute views by %

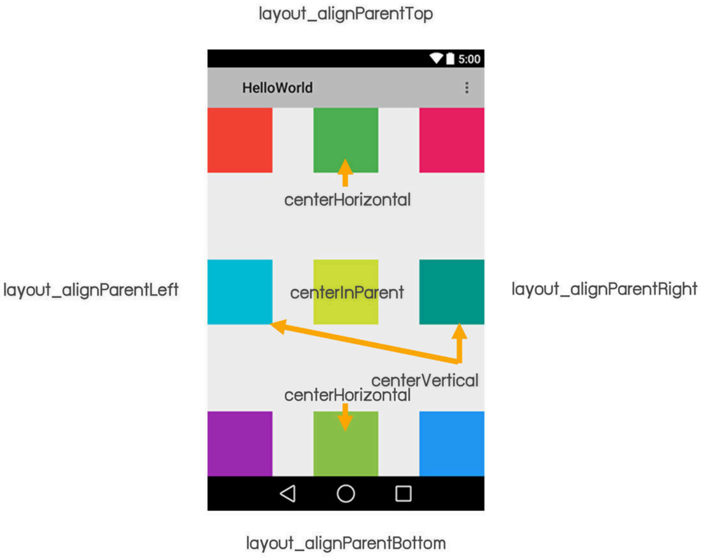
```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    >
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="TextView 1"
        android:background="#ffcccc"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="TextView 2"
        android:background="#ccffcc"/>
    <TextView
        android:layout_width="0dp"
        android:layout_height="match_parent"
        android:layout_weight="1"
        android:text="TextView 3"
        android:background="#ffccff"/>
</LinearLayout>
```



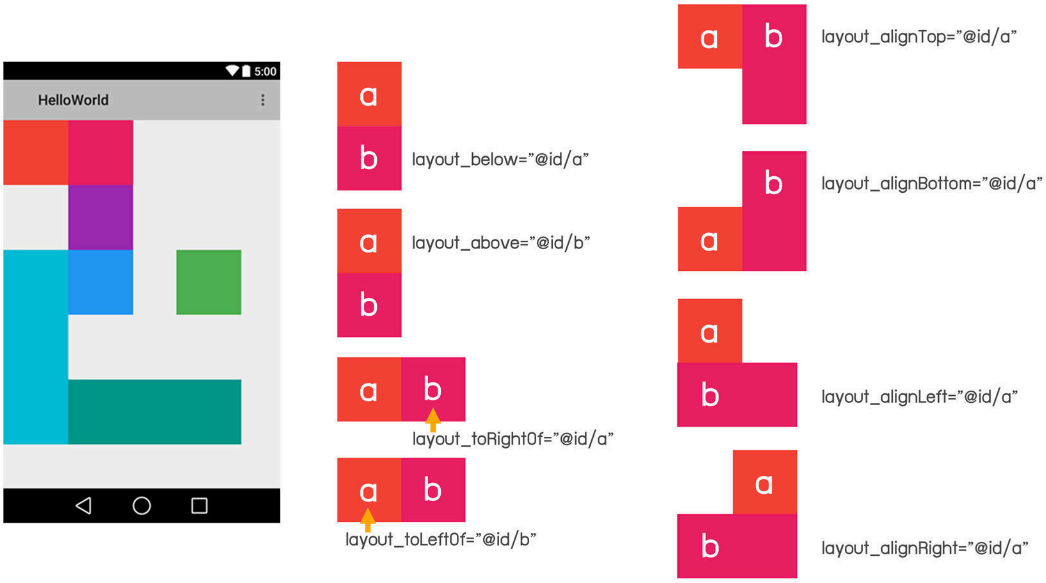
TextView 1 TextView 2 TextView 3

LAYOUT

RelativeLayout: Align to Parent



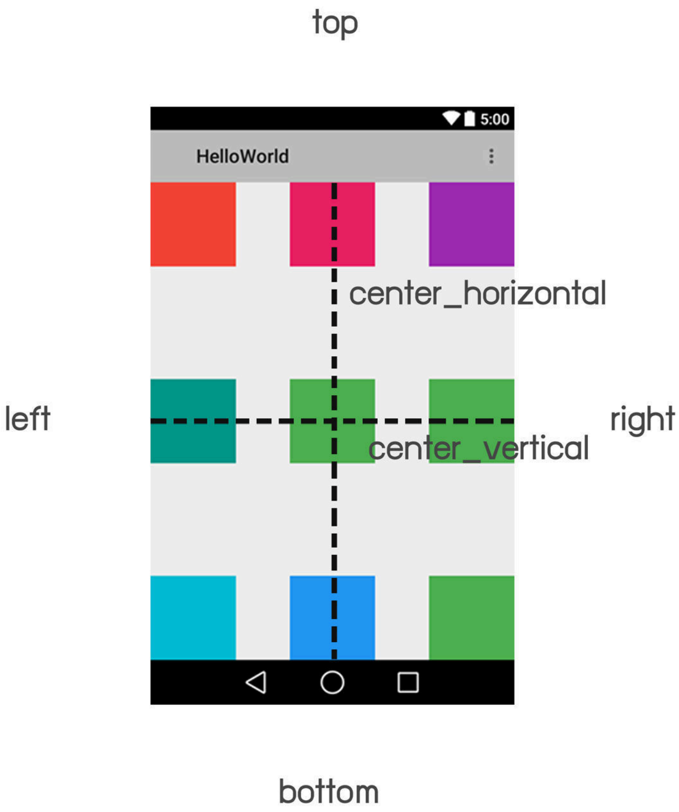
RelativeLayout: Align to Sibling View(s)



LAYOUT

FrameLayout

layout_gravity

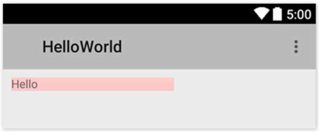


combine with or (|)
for example,

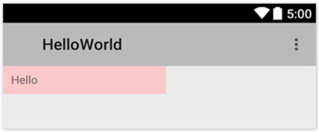
```
layout_gravity="center_vertical|right"
```

LAYOUT

margin & padding

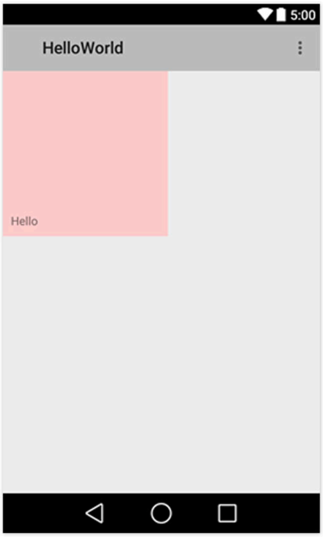


android:layout_margin="10dp"

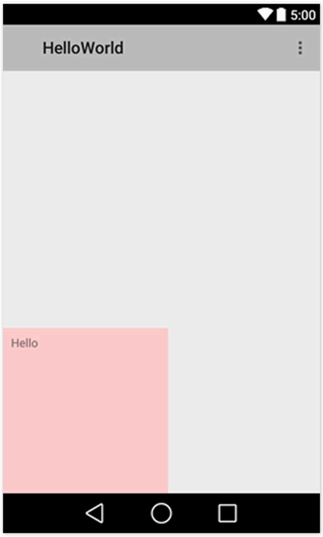


android:padding="10dp"

gravity vs layout_gravity



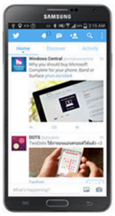
android:gravity="bottom"

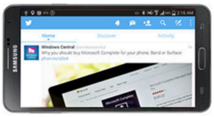


android:layout_gravity="bottom"

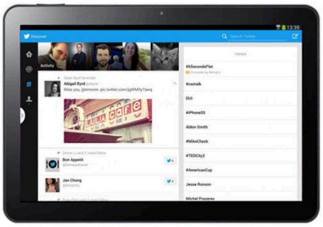
FRAGMENTATION

4 TYPES OF SCREEN YOU HAVE TO DEAL WITH


Mobile Portrait


Mobile Landscape

Tablet Portrait


Tablet Landscape

3 TYPES OF MOBILE PHONES

Low End

Works
Fine
::
- UI Fit -

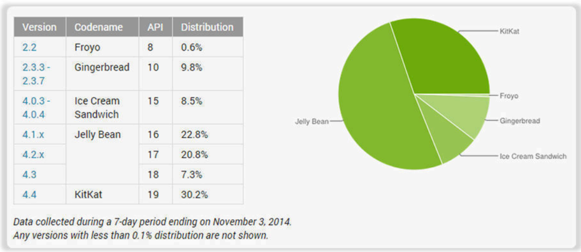
Mid End

Works
Perfectly
::
- UI Fit -
- Fluid -
- Fast -

High End

OS VERSION

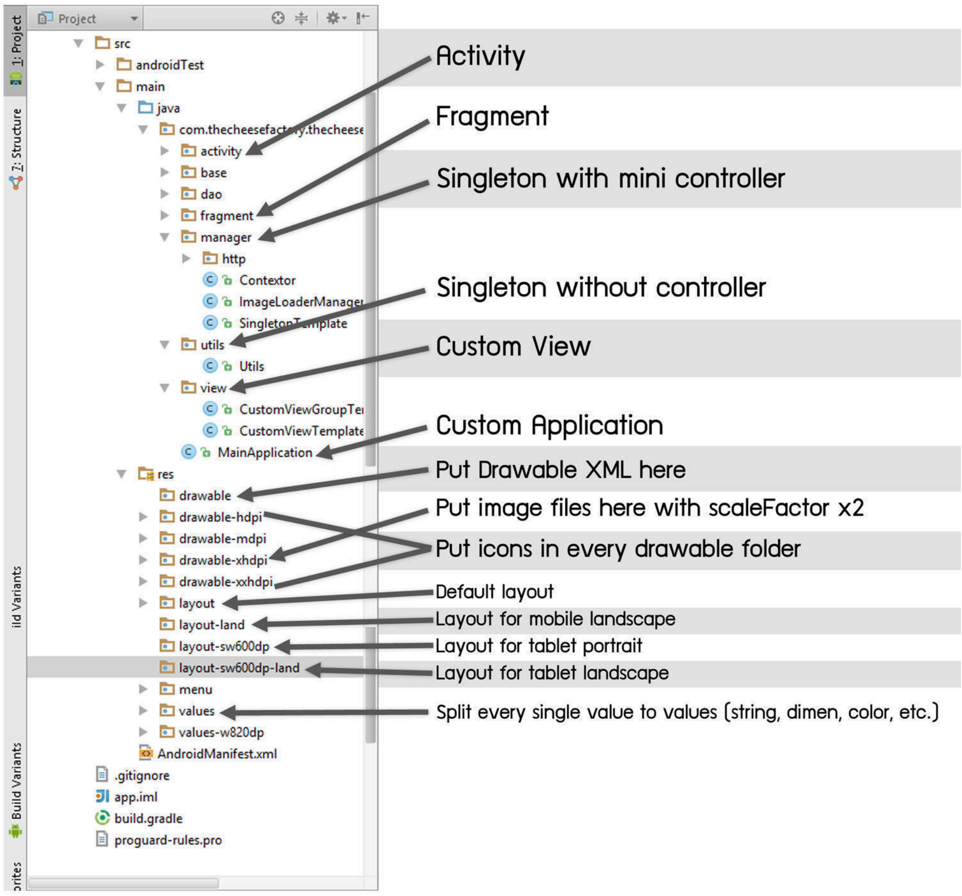
minSdkVersion = 14



Version	Codename	API	Distribution
2.2	Froyo	8	0.6%
2.3.3 - 2.3.7	Gingerbread	10	9.8%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	8.5%
4.1.x	Jelly Bean	16	22.8%
4.2.x		17	20.8%
4.3		18	7.3%
4.4	KitKat	19	30.2%

Data collected during a 7-day period ending on November 3, 2014.
Any versions with less than 0.1% distribution are not shown.

CODE STRUCTURE


1: Project
7: Structure
ild Variants
Build Variants
orites
Project
Activity
Fragment
Singleton with mini controller
Singleton without controller
Custom View
Custom Application
Put Drawable XML here
Put image files here with scaleFactor x2
Put icons in every drawable folder
Default layout
Layout for mobile landscape
Layout for tablet portrait
Layout for tablet landscape
Split every single value to values (string, dimen, color, etc.)

```
▼ src
  ▶ androidTest
  ▼ main
    ▼ java
      ▼ com.thecheesefactory.thecheese
        ▶ activity
        ▶ base
        ▶ dao
        ▶ fragment
        ▼ manager
          ▶ http
            Contextor
            ImageLoaderManager
            SingletonTemplate
        ▼ utils
            Utils
        ▼ view
            CustomViewGroupTe
            CustomViewTemplate
          MainApplication
    ▼ res
        drawable
      ▶ drawable-hdpi
      ▶ drawable-mdpi
      ▶ drawable-xhdpi
      ▶ drawable-xxhdpi
      ▶ layout
        layout-land
        layout-sw600dp
        layout-sw600dp-land
      ▶ menu
      ▶ values
      ▶ values-w820dp
      AndroidManifest.xml
  .gitignore
  app.iml
  build.gradle
  proguard-rules.pro
```

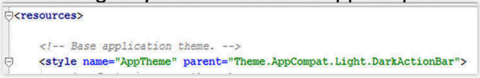
ACTIVITY

Template

Use ActionBarActivity from Android Support Library v7

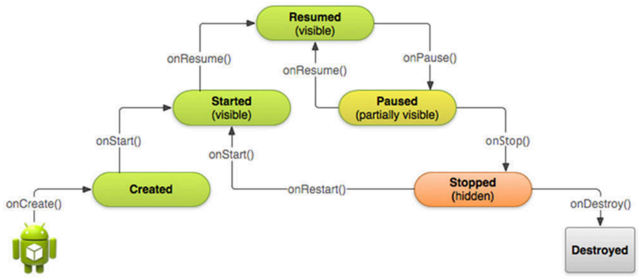
```
public class ActivityTemplate extends ActionBarActivity {

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

        initInstances();
    }

    private void initInstances() {
        // init instance with findViewById here
    }

    @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
        switch (id) {
            case R.id.action_settings:
                return true;
            default:
                break;
        }
        return super.onOptionsItemSelected(item);
    }
}
```

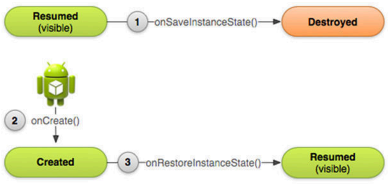
Change styles.xml to Theme.AppCompat.*



```
<resources>

    <!-- Base application theme. -->
    <style name="AppTheme" parent="Theme.AppCompat.Light.DarkActionBar">
```

ACTIVITY - LIFECYCLE





Template

```
...

@Override
protected void onStart() {
    super.onStart();
}

@Override
protected void onStop() {
    super.onStop();
}

@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
}

@Override
protected void onRestoreInstanceState(Bundle savedInstanceState) {
    super.onRestoreInstanceState(savedInstanceState);
}

...
```

CUSTOM APPLICATION

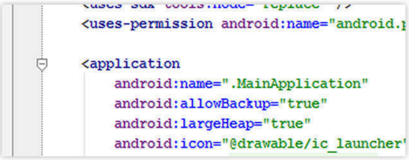
Template

```
public class MainApplication extends Application {
    @Override
    public void onCreate() {
        super.onCreate();
        Contextor.getInstance().init(getApplicationContext());
    }
}
```

Make Context Global

```
public class Contextor {
    private static Contextor instance;

    public static Contextor getInstance() {
        if (instance == null)
            instance = new Contextor();
        return instance;
    }

    private Context mContext;

    public Contextor() {}

    public void init(Context context) {
        mContext = context;
    }

    public Context getContext() {
        return mContext;
    }
}
```

Define in AndroidManifest.xml



```
<uses-sdk tools:node="replace" />
<uses-permission android:name="android.

<application
    android:name=".MainApplication"
    android:allowBackup="true"
    android:largeHeap="true"
    android:icon="@drawable/ic_launcher
```

SINGLETON

Template

```
public class SingletonTemplate {

    private static SingletonTemplate instance;

    public static  SingletonTemplate getInstance() {
        if (instance == null)
            instance = new SingletonTemplate();
        return instance;
    }

    private Context mContext;

    private SingletonTemplate() {
        mContext = Contextor.getInstance().getContext();
    }

}
```

Best Practices:

- Use Singleton as Model in MVC
- Singleton with mini controller inside = Manager
- Singleton without controller = Utils

} Put in separate folder

FRAGMENT

Template

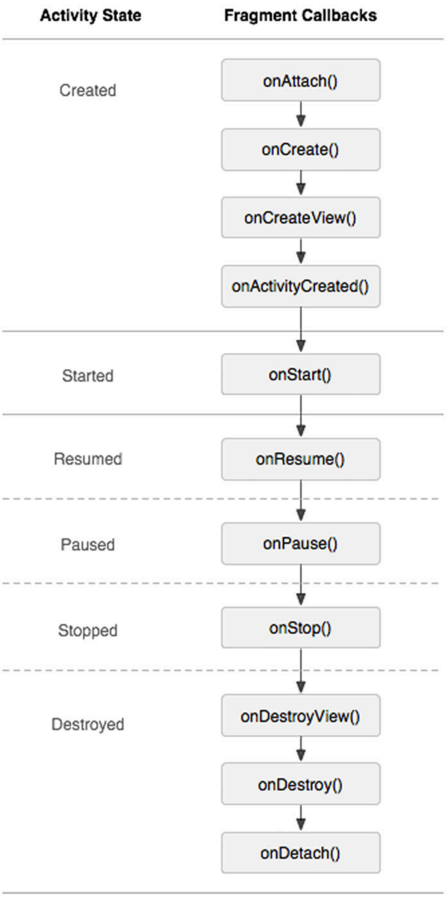
Use Fragment from Android Support Library v4

```
public class FragmentTemplate extends Fragment {
    public FragmentTemplate() {
        super();
    }

    public FragmentTemplate newInstance() {
        FragmentTemplate fragment = new FragmentTemplate();
        Bundle args = new Bundle();
        fragment.setArguments(args);
        return fragment;
    }

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
        View rootView = inflater.inflate(R.layout.fragment_main, container,
                                         false);
        initInstances(rootView);
        return rootView;
    }

    private void initInstances(View rootView) {
        // init instance with rootView.findViewById here
    }

    @Override
    public void onStart() {
        super.onStart();
    }

    @Override
    public void onStop() {
        super.onStop();
    }
}
```

FRAGMENT - LIFECYCLE

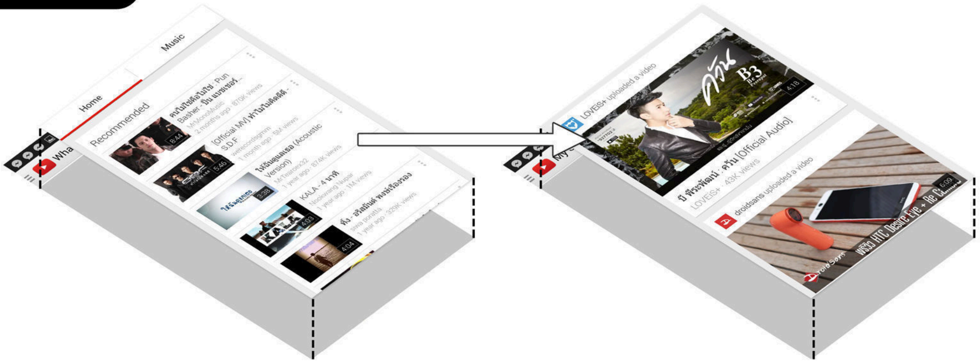
Template

```
...

@Override
public View onCreateView(
           LayoutInflater inflater,
           ViewGroup container,
           Bundle savedInstanceState) {
    View rootView = inflater.inflate(
                R.layout.fragment_main,
                container,
                false);
    initInstances(rootView);
    // Restore Instance State here
    ...
    return rootView;
}

@Override
protected void onStart() {
    super.onStart();
}

@Override
protected void onStop() {
    super.onStop();
}

@Override
public void onSaveInstanceState(
                   Bundle outState) {
    super.onSaveInstanceState(outState);
    // Save Instance State here
}

...
```



USE OF FRAGMENT

1) Replace



1 Declare Container in xml

```
<FrameLayout
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    />
```

2 Add First Fragment in Activity's onCreate

```
if (savedInstanceState == null) { // Check if it is the first launch
    getSupportFragmentManager().beginTransaction()
                               .add(R.id.container, FragmentA.newInstance())
                               .commit();
}
```

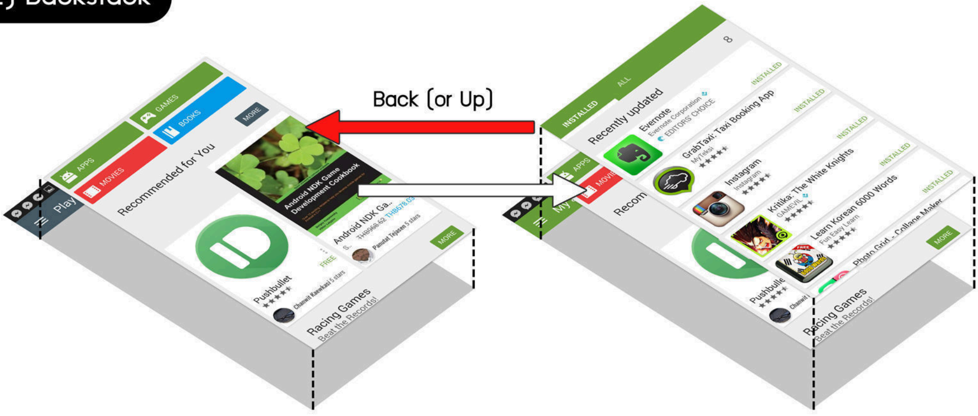
3 Replace new tab with “replace”

```
Fragment fragment = getSupportFragmentManager().findFragmentById(R.id.container);
if (fragment == null || !(fragment instanceof FragmentA))
    getSupportFragmentManager().beginTransaction()
                               .replace(R.id.container, FragmentA.newInstance())
                               .commit();
```

* Or use NonSwipeableViewPager for better experience *

USE OF FRAGMENT

2) Backstack



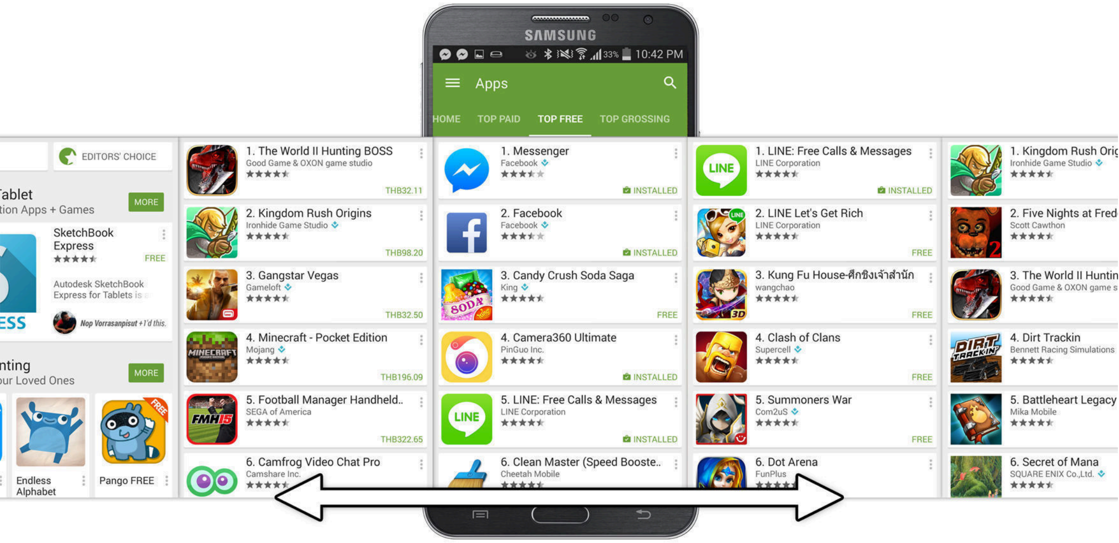
1 Declare fragment in xml

```
<fragment
    android:id="@+id/fragment"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:name="com.thecheesefactory.lab.fragment.FragmentA"
    />
```

2 Add Another Fragment to Backstack

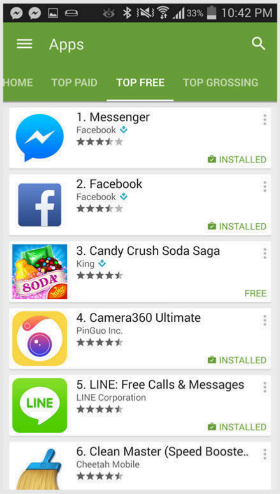
```
Fragment fragment = getSupportFragmentManager().findFragmentById(R.id.fragment);
if (fragment == null || !(fragment instanceof FragmentB))
    getSupportFragmentManager().beginTransaction()
                               .replace(R.id.fragment, FragmentB.newInstance())
                               .addToBackstack(null)
                               .commit();
```

VIEWPAGER



Swipeable

Swipeable - Use as Page (Normal Use)



Define ViewPager and use it in normal way

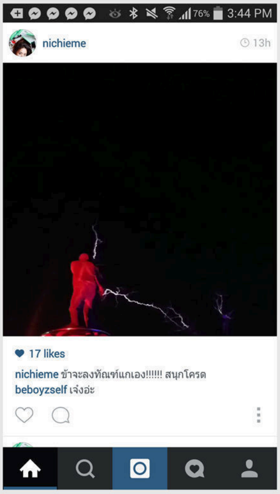
You can use SlidingTabLayout to make Tab Bar looks like this

Nonswipeable - Use as Tab Switching

Override ViewPager to disable swipe (search for Non Swipeable ViewPager) and use that in xml

To change tab, call

```
viewPager.setCurrentItem
          (tabIndex, false);
```



nichieme ข้าจะลงทัณฑ์แกเอง!!!!!! สนุกโครต
beboyzself เจ๋งอ่ะ

VIEWPAGER - USE AS TAB SWITCHING

Define NonSwipeableViewPager

```
public class NonSwipeableViewPager extends ViewPager {

    public NonSwipeableViewPager(Context context) {
        super(context);
    }

    public NonSwipeableViewPager(Context context, AttributeSet attrs) {
        super(context, attrs);
    }

    @Override
    public boolean onInterceptTouchEvent(MotionEvent arg0) {
        // Never allow swiping to switch between pages
        return false;
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        // Never allow swiping to switch between pages
        return false;
    }
}
```

Use it in <layout>.xml instead of normal ViewPager

```
<com.thecheesefactory.lab.view.NonSwipeableViewPager
    android:id=”+@id/viewPager”
    android:layout_width=”match_parent”
    android:layout_height=”match_parent”
    />
```

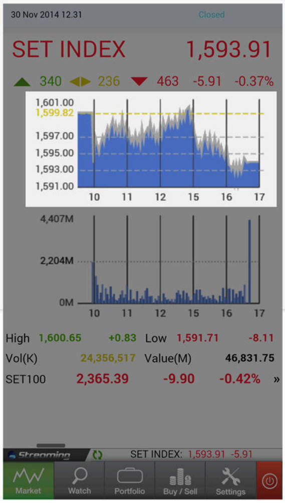
Change Tab without swiping animation via command

```
viewPager.setCurrentItem(tabIndex, false);
```

CUSTOM VIEW

Custom View

```java
public class CustomViewTemplate extends View {

    public CustomViewTemplate(Context context) {
        super(context);
        init();
    }

    public CustomViewTemplate(Context context, AttributeSet attrs) {
        super(context, attrs);
        init();
        initWithAttrs(attrs);
    }

    public CustomViewTemplate(Context context, AttributeSet attrs,
                              int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        initWithAttrs(attrs);
        init();
    }

    private void init() {
        setWillNotDraw(false);
    }

    private void initWithAttrs(AttributeSet attrs) {

    }

    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
    }
}
```



30 Nov 2014 12.31
Closed
SET INDEX 1,593.91
340 236 463 -5.91 -0.37%
High 1,600.65 +0.83 Low 1,591.71 -8.11
Vol(K) 24,356,517 Value(M) 46,831.75
SET100 2,365.39 -9.90 -0.42%
SET INDEX: 1,593.91 -5.91
Market
Watch
Portfolio
Buy / Sell
Settings

Advantage

- Custom Draw
- Custom Input Handling

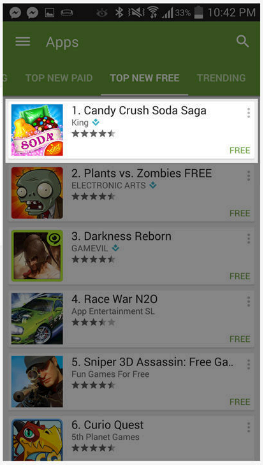
CUSTOM VIEWGROUP

Custom ViewGroup

```
public class CustomViewGroupTemplate extends RelativeLayout {

    private TextView tvName;

    public CustomViewGroupTemplate(Context context) {
        super(context);
        initInflate();
        initInstances();
    }

    public CustomViewGroupTemplate(Context context,
                                   AttributeSet attrs) {
        super(context, attrs);
        initInflate();
        initInstances();
        initWithAttrs(attrs);
    }

    public CustomViewGroupTemplate(Context context, AttributeSet attrs,
                                         int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        initInflate();
        initInstances();
        initWithAttrs(attrs);
    }

    private void initInflate() {
        LayoutInflater inflater = (LayoutInflater)getContext()
                      .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        inflater.inflate(R.layout.blog_list_item, this);
    }

    private void initInstances() {
        tvName = (TextView) findViewById(R.id.tvName);
    }

    private void initWithAttrs(AttributeSet attrs) {
    }
}
```

Advantage

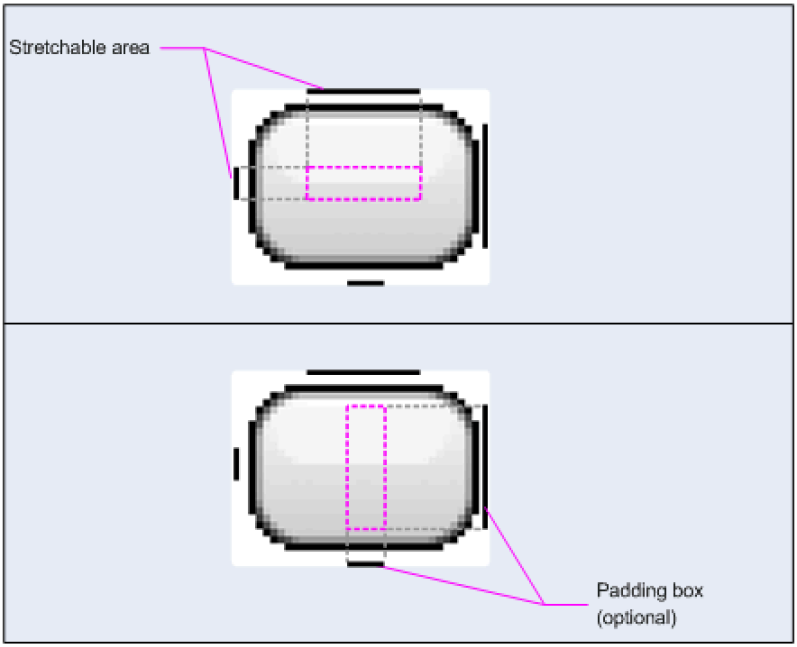
- Layout Grouping
- Reuseable
- Custom Input Handling



9-PATCH

Marker Pattern

* Add a pixel to every single edge. Mark with black (#000000) color. The rest must be transparent (#00000000) pixel.



The way it is scaled is just "streching", so use only plain color or gradient

Filetype

must be a PNG file with .9.png file extension

Result

Button with 8sp textSize

Button with 30sp textSize

Tools

Simple Nine-patch Generator
http://romannurik.github.io/AndroidAssetStudio/nine-patches.html

CODE SNIPPET

Send SMS

```
SmsManager m = SmsManager.getDefault();
String destination = "+66812345678";
String text = "Hello, John!";
m.sendTextMessage(destination, null, text, null, null);
```

Open URL in Browser

```
String url = "http://www.google.com";
Intent browserIntent = new Intent(Intent.ACTION_VIEW, Uri.parse(url));
startActivity(browserIntent);
```

Send Text Content to Another App

```
Intent sendIntent = new Intent();
sendIntent.setAction(Intent.ACTION_SEND);
sendIntent.putExtra(Intent.EXTRA_TEXT, "This is my text to send.");
sendIntent.setType("text/plain");
startActivity(sendIntent);
```

Send Image to Another App

```
Intent shareIntent = new Intent();
shareIntent.setAction(Intent.ACTION_SEND);
shareIntent.putExtra(Intent.EXTRA_STREAM, uriToImage);
shareIntent.setType("image/jpeg");
startActivity(shareIntent);
```

Vibrate

```
(Vibrator)getSystemService(Context.VIBRATOR_SERVICE).vibrate(milliseconds);
```

CODE SNIPPET

Alert Dialog

```
AlertDialog.Builder alert = new AlertDialog.Builder(this);
alert.setTitle(title);
alert.setMessage(message);

// You can set an EditText view to get user input besides
// which button was pressed.
final EditText input = new EditText(this);
alert.setView(input);

alert.setPositiveButton("Ok", new DialogInterface.OnClickListener() {
public void onClick(DialogInterface dialog, int whichButton) {
  String value = input.getText();
  // Do something with value!
  }
});
alert.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
  public void onClick(DialogInterface dialog, int whichButton) {
    // Canceled.
  }
});

alert.show();
```

Enable/Disable WiFi

```
WifiManager wifi = (WifiManager) getSystemService(Context.WIFI_SERVICE);
wifi.setWifiEnabled(enabled);
```

Enable/Disable Ringer

```
AudioManager mAudio = (AudioManager) getSystemService(Activity.AUDIO_SERVICE);
mAudio.setRingerMode(AudioManager.RINGER_MODE_SILENT);
// or...
mAudio.setRingerMode(AudioManager.RINGER_MODE_NORMAL);
```

CODE SNIPPET

HTML in TextView

```
textView.setText(Html.fromHtml("<h2>Title</h2><br><p>Description here</p>"));
```

Take a picture with Intent

```
Intent intent = new Intent("android.media.action.IMAGE_CAPTURE");
startActivityForResult(intent, 123456);
// ...

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data) {
  if (resultCode == Activity.RESULT_OK && requestCode == 123456) {
    String result = data.toURI();
    // ...
  }
}
```

Phone Dial

```
Intent intent = new Intent(Intent.ACTION_DIAL);
intent.setData(Uri.parse(”tel://0812345678”));
startActivity(intent);
```

Make a phone call

```
Intent intent = new Intent(Intent.ACTION_CALL);
intent.setData(Uri.parse(”tel://0812345678”));
startActivity(intent);
```

```
<uses-permission android:name="android.permission.CALL_PHONE"/>
```

More about Intent Action:
http://developer.android.com/reference/android/content/Intent.html