**JumpingBeans project (in Android Studio)**

* File – New – New Project.(JumpingBeansLoading)

Content\_main.xml

*<!--  
 ~ Copyright 2015 Frakbot (Sebastiano Poggi and Francesco Pontillo)  
 ~  
 ~ Licensed under the Apache License, Version 2.0 (the "License");  
 ~ you may not use this file except in compliance with the License.  
 ~ You may obtain a copy of the License at  
 ~  
 ~ http://www.apache.org/licenses/LICENSE-2.0  
 ~  
 ~ Unless required by applicable law or agreed to in writing, software  
 ~ distributed under the License is distributed on an "AS IS" BASIS,  
 ~ WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
 ~ See the License for the specific language governing permissions and  
 ~ limitations under the License.  
 -->*<**LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:paddingBottom="@dimen/activity\_vertical\_margin"  
 android:paddingLeft="@dimen/activity\_horizontal\_margin"  
 android:paddingRight="@dimen/activity\_horizontal\_margin"  
 android:paddingTop="@dimen/activity\_vertical\_margin"  
 tools:context=".MainActivity"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="8dp"  
 android:text="@string/label\_1"  
 android:textAppearance="@style/TextAppearance.AppCompat.Display1"** />  
  
 <**TextView  
 android:id="@+id/jumping\_text\_1"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="24dp"  
 android:text="@string/message"  
 android:textAppearance="@style/AlertDialog.AppCompat.Light"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="@string/label\_2"  
 android:textAppearance="@style/TextAppearance.AppCompat.Display1"** />  
  
 *<!-- This TextView has 8dp of top padding to avoid clipping the ascenders  
 of the first word when animating - also, the TextView above this doesn't  
 have any margin, to create the same spacing that we have between the  
 first couple of TextViews. -->* <**TextView  
 android:id="@+id/jumping\_text\_2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:paddingTop="8dp"  
 android:text="@string/message"  
 android:textAppearance="@style/Base.AlertDialog.AppCompat.Light"** />  
  
</**LinearLayout**>

Activity\_main.xml

*<?***xml version="1.0" encoding="utf-8"***?>*<**android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:fitsSystemWindows="true"  
 tools:context=".MainActivity"**>  
  
 <**android.support.design.widget.AppBarLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:theme="@style/AppTheme.AppBarOverlay"**>  
  
 <**android.support.v7.widget.Toolbar  
 android:id="@+id/toolbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="?attr/actionBarSize"  
 android:background="?attr/colorPrimary"  
 app:popupTheme="@style/AppTheme.PopupOverlay"** />  
  
 </**android.support.design.widget.AppBarLayout**>  
  
 <**include layout="@layout/content\_main"** />  
  
 <**android.support.design.widget.FloatingActionButton  
 android:id="@+id/fab"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="bottom|end"  
 android:layout\_margin="@dimen/fab\_margin"  
 android:src="@android:drawable/ic\_dialog\_email"** />  
  
</**android.support.design.widget.CoordinatorLayout**>

* Copy and paste all the folders inside libs folder.
* After copying folders inside libs folder, add following lines in build.gradle (Module:app).

compile **'net.frakbot:jumpingbeans:1.3.0'**

MainActivity.java

**package** com.example.crossoovernepal.jumpingbeansloading;  
  
**import** android.os.Bundle;  
**import** android.support.design.widget.FloatingActionButton;  
**import** android.support.design.widget.Snackbar;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.support.v7.widget.Toolbar;  
**import** android.view.View;  
**import** android.view.Menu;  
**import** android.view.MenuItem;  
**import** android.widget.TextView;  
  
**import** net.frakbot.jumpingbeans.JumpingBeans;  
  
**public class** MainActivity **extends** AppCompatActivity {  
 **private** JumpingBeans **jumpingBeans1**, **jumpingBeans2**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 Toolbar toolbar = (Toolbar) findViewById(R.id.***toolbar***);  
 setSupportActionBar(toolbar);  
  
 FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.***fab***);  
 fab.setOnClickListener(**new** View.OnClickListener() {  
 @Override  
 **public void** onClick(View view) {  
 Snackbar.*make*(view, **"Replace with your own action"**, Snackbar.***LENGTH\_LONG***)  
 .setAction(**"Action"**, **null**).show();  
 }  
 });  
 }  
  
 @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* **if** (id == R.id.***action\_settings***) {  
 **return true**;  
 }  
  
 **return super**.onOptionsItemSelected(item);  
 }  
  
 @Override  
 **protected void** onResume() {  
 **super**.onResume();  
  
 *// Here you can see that we don't duplicate trailing dots on the text (we reuse  
 // them or, if it's an ellipsis character, replace it with three dots and animate  
 // those instead)* **final** TextView textView1 = (TextView) findViewById(R.id.***jumping\_text\_1***);  
 **jumpingBeans1** = JumpingBeans.*with*(textView1)  
 .appendJumpingDots()  
 .build();  
  
 *// Note that, even though we access textView2's text when starting to work on  
 // the animation builder, we don't alter it in any way, so we're ok* **final** TextView textView2 = (TextView) findViewById(R.id.***jumping\_text\_2***);  
 **jumpingBeans2** = JumpingBeans.*with*(textView2)  
 .makeTextJump(0, textView2.getText().toString().indexOf(**' '**))  
 .setIsWave(**false**)  
 .setLoopDuration(1000)  
 .build();  
 }  
  
 @Override  
 **protected void** onPause() {  
 **super**.onPause();  
 **jumpingBeans1**.stopJumping();  
 **jumpingBeans2**.stopJumping();  
 }  
  
}