

2015 年 11 月 12 日 (木) 実施

アニメーション

アニメーションの基本

アニメーションは少しずつ異なる画像を連続して表示することで実現出来る。ここでは、前回学んだイメージビュー上で、画面に触れる度に表示する画像を入れ替えるという、手動のアニメーションを作成する。

課題

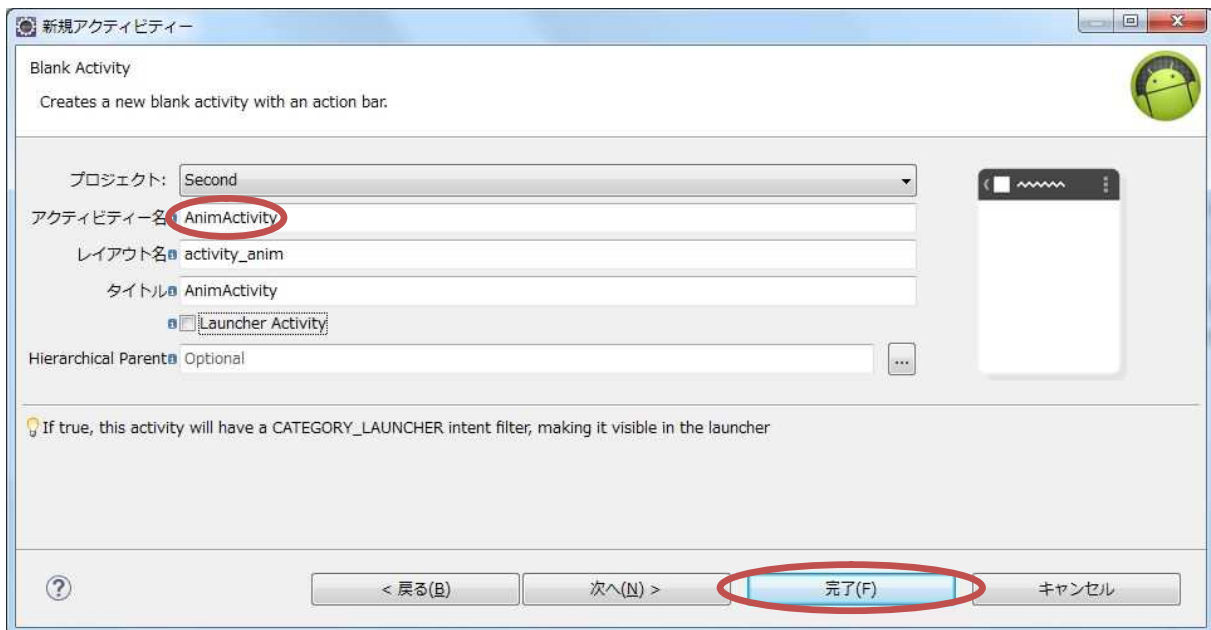
今回は、新規のアクティビティを作成し、イメージビューを用いた画面定義を作成し、アニメーションの基本を学ぶ。

Android アプリの作成

Eclipse を起動し、『ファイル』→『新規』→『Android アクティビティ』と選択して、『次へ』をクリックする。

『Blank Activity』を選択して、『次へ』をクリックする。

『アクティビティ名』を「AnimActivity」に書き換え、『完了』をクリックする。

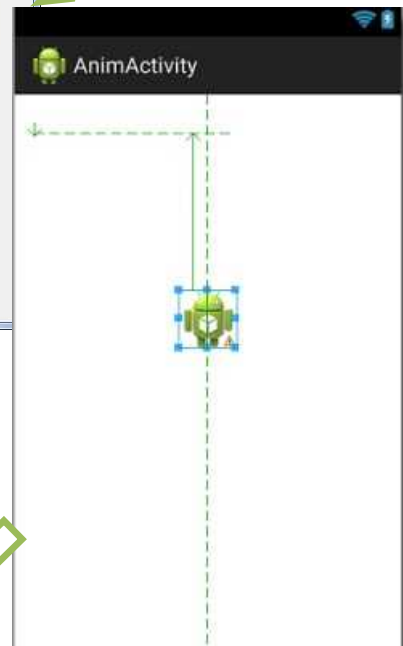
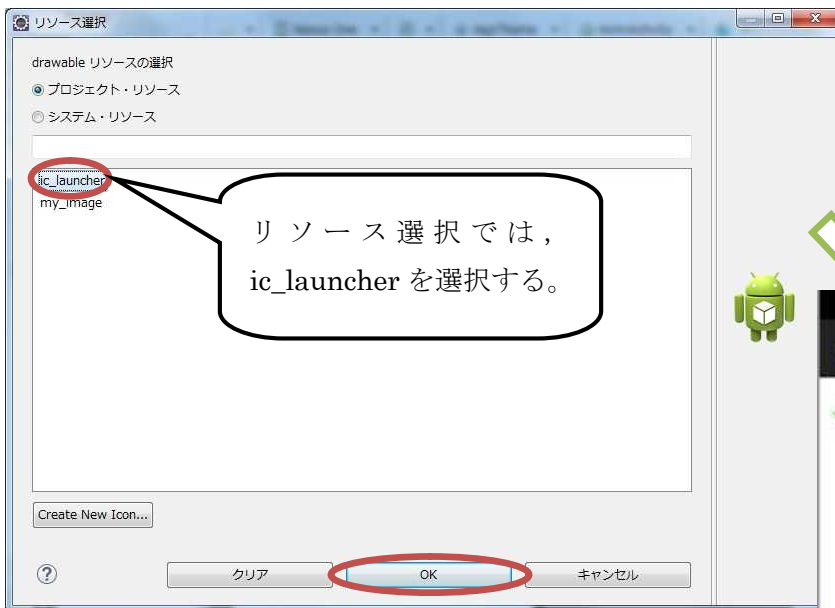


テキストビューの『Properties』で『Id』の右側のテキストボックスに「@+id/animtv1」と入力し、『Text』欄を空にする。

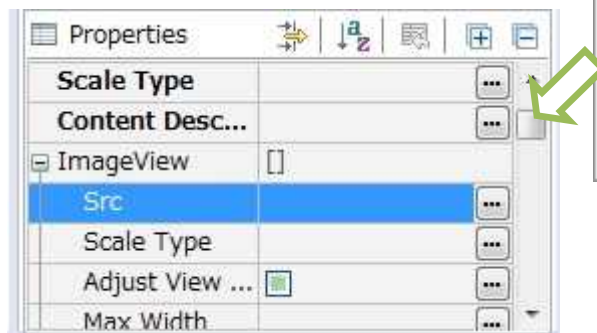
(図は次ページ)



activity_anim.xml にイメージビューを配置する。



イメージビューの『Properties』で『Src』欄を空にする。



H ドライブの¥workspace¥Second¥res¥drawable-hdpi の位置に、ダウンロードした **anim1.png**～**anim5.png** をコピーして貼り付ける。

Eclipse のパッケージ・エクスプローラーで、『drawable-hdpi』を右クリックして、リフレッシュを適用する。



AnimActivity.java に下の図の色付きの枠で囲った箇所を付け加える。

```

1 package jp.ac.cuc.jimbo.second;
2
3 import android.app.Activity;
4 import android.os.Bundle;
5 import android.view.Menu;
6 import android.view.MenuItem;
7 import android.view.MotionEvent;
8 import android.widget.ImageView;
9 import android.widget.TextView;
10
11 public class AnimActivity extends Activity {
12     private TextView tv;
13     private ImageView iv;
14     private int n = 0;
15
16     @Override
17     protected void onCreate(Bundle savedInstanceState) {
18         super.onCreate(savedInstanceState);
19         setContentView(R.layout.activity_anim);
20
21         tv = (TextView)findViewById(R.id.anim_tv);
22         iv = (ImageView)findViewById(R.id.imageView);
23     }
24
25     @Override
26     public boolean onTouchEvent(MotionEvent event) {
27         if (event.getAction() == MotionEvent.ACTION_DOWN) {
28             n++;
29             tv.setText("タッチ回数:" + n + "回");
30             switch(n%5){
31                 case 1:
32                     iv.setImageResource(R.drawable.anim1);
33                     break;
34                 case 2:
35                     iv.setImageResource(R.drawable.anim2);
36                     break;
37                 case 3:
38                     iv.setImageResource(R.drawable.anim3);
39                     break;
40                 case 4:
41                     iv.setImageResource(R.drawable.anim4);
42                     break;
43                 case 0:
44                     iv.setImageResource(R.drawable.anim5);
45                     break;
46             }
47         }
48         return false;
49     }
50
51     @Override
52     public boolean onCreateOptionsMenu(Menu menu) {

```

- 1) 橙色の枠内 **AnimActivity** クラスのフィールド・・・クラス内の複数のメソッドで使用。
TextView のインスタンス tv, ImageView のインスタンス iv, 整数変数 n (カウンタとして用いるので, 0 に初期化)
- 2) 紫色の枠内 **activity_anim.xml** の animtv1 を tv, imageView1 を iv にセット。
- 3) 赤色の枠内 **onTouchEvent** を用いて, 画面がタッチされた際のイベントを記述。
 - ① カウンタ n を増分演算子++で 1 だけ進める。
 - ② tv に **setText** メソッドを適用して, タッチ回数を表示する。
 - ③ **switch** 文で n を 5 で割った余り (n%5) を評価して, その値が 1, 2, 3, 4, 0 (5 で割り切れる場合) のそれぞれに対して, **setImageResource** で **anim1.png~anim5.png** を iv に表示するようにセットする。

それぞれの色の枠内を次に示す。

```
private TextView tv;
private ImageView iv;
private int n = 0;
```

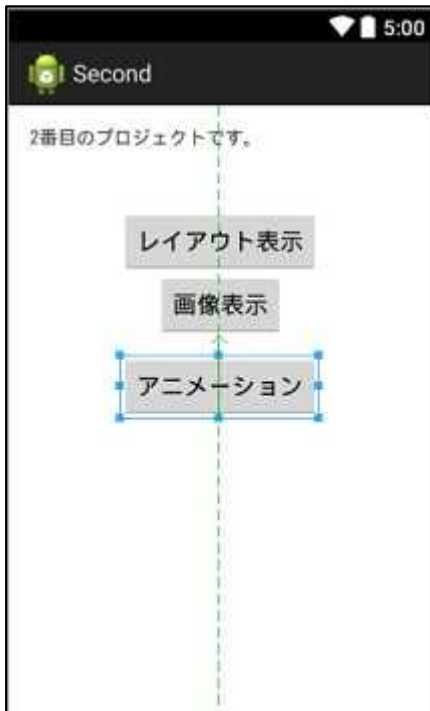
```
tv = (TextView)findViewById(R.id.animtv1);
iv = (ImageView)findViewById(R.id.imageView1);
```

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        n++;
        tv.setText("タッチ回数:" + n + "回");
        switch(n%5) {
            case 1:
                iv.setImageResource(R.drawable.anim1);
                break;
            case 2:
                iv.setImageResource(R.drawable.anim2);
                break;
            case 3:
                iv.setImageResource(R.drawable.anim3);
                break;
            case 4:
                iv.setImageResource(R.drawable.anim4);
                break;
            case 0:
                iv.setImageResource(R.drawable.anim5);
                break;
        }
    }
    return false;
}
```

『res』 → 『values』 と展開し、『strings.xml』を開き、『追加』をクリックする。『String』を選択して OK をクリックし、『名前』を「button3_label」、『Value』を「アニメーション」と入力して、保管して閉じる。

名前	button3_label
Value*	アニメーション

activity_main2.xml にボタンを配置する。ボタンの Text には『button3_label』を設定する。



Main2Activity.java を開く。

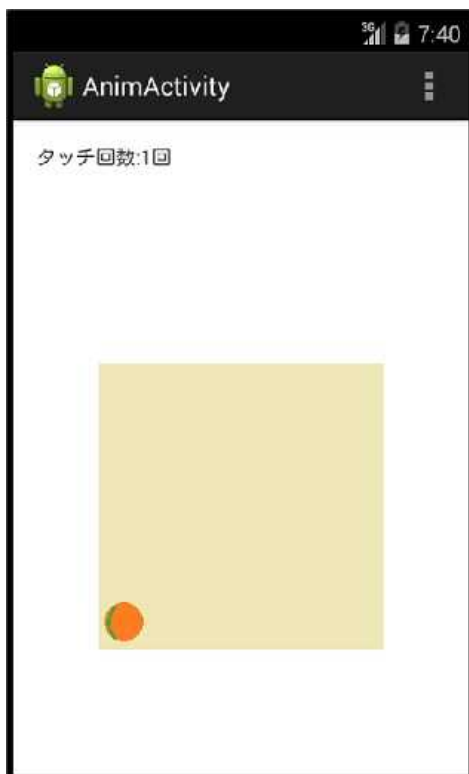
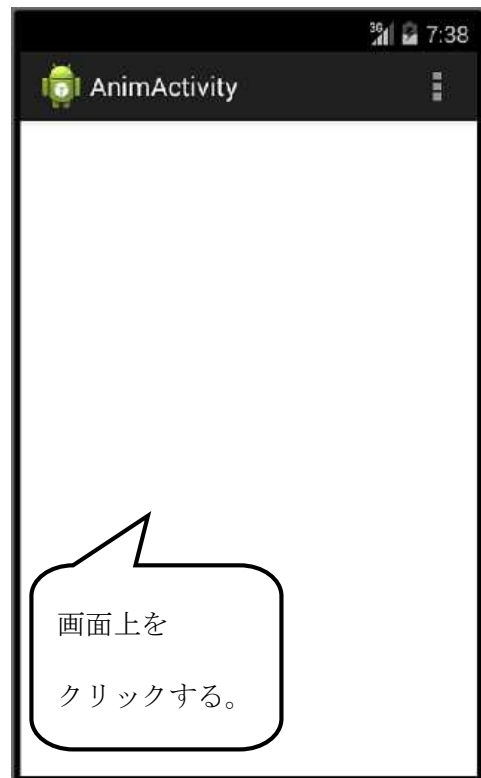
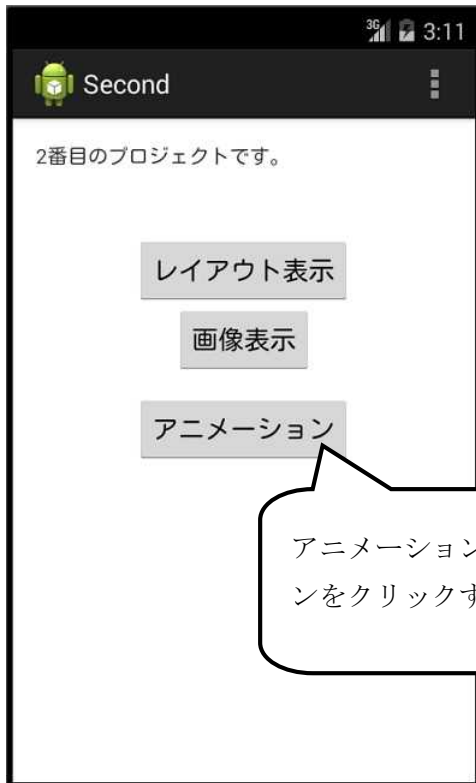
onCreate メソッド中に Button ウィジェット button3 のインスタンス btn3 を作成し、それに働きかけるイベントリスナーを付け加える。イベントリスナーには、インテントを記述する。(intent.setClassName の行は、教材の行幅の都合で折り返している。)

```
Button btn3 = (Button) this.findViewById(R.id.button3);
btn3.setOnClickListener(
    new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent();
            intent.setClassName("jp.ac.cuc.jimbo.second",
                "jp.ac.cuc.jimbo.second.AnimActivity");
            startActivity(intent);
        }
    }
);
```

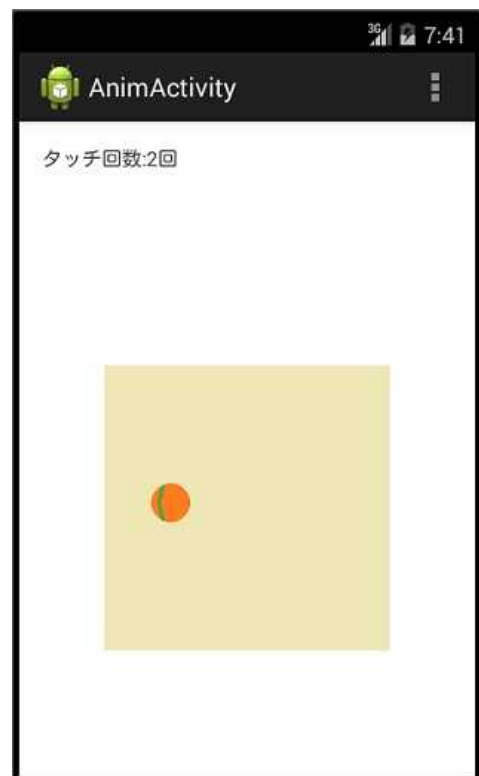
```
1 package jp.ac.cuc.jimbo.second;
2
3 import android.app.Activity;
4 import android.content.Intent;
5 import android.os.Bundle;
6 import android.view.Menu;
7 import android.view.MenuItem;
8 import android.view.View;
9 import android.widget.Button;
10
11 public class Main2Activity extends Activity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_main2);
17         Button btn = (Button) this.findViewById(R.id.button1);
18         btn.setOnClickListener(
19             new View.OnClickListener() {
20
21                 @Override
22                 public void onClick(View v) {
23                     drawLayouts();
24                 }
25             }
26         );
27         Button btn2 = (Button) this.findViewById(R.id.button2);
28         btn2.setOnClickListener(
29             new View.OnClickListener() {
30
31                 @Override
32                 public void onClick(View v) {
33                     setContentView(R.layout.image);
34                 }
35             }
36         );
37         Button btn3 = (Button) this.findViewById(R.id.button3);
38         btn3.setOnClickListener(
39             new View.OnClickListener() {
40
41                 @Override
42                 public void onClick(View v) {
43                     Intent intent = new Intent();
44                     intent.setClassName("jp.ac.cuc.jimbo.second", "jp.ac.cuc.jimbo.second.AnimActivity");
45                     startActivity(intent);
46                 }
47             }
48         );
49     }
50
51     public void drawLayouts() {
52         // ...
53     }
54 }
```

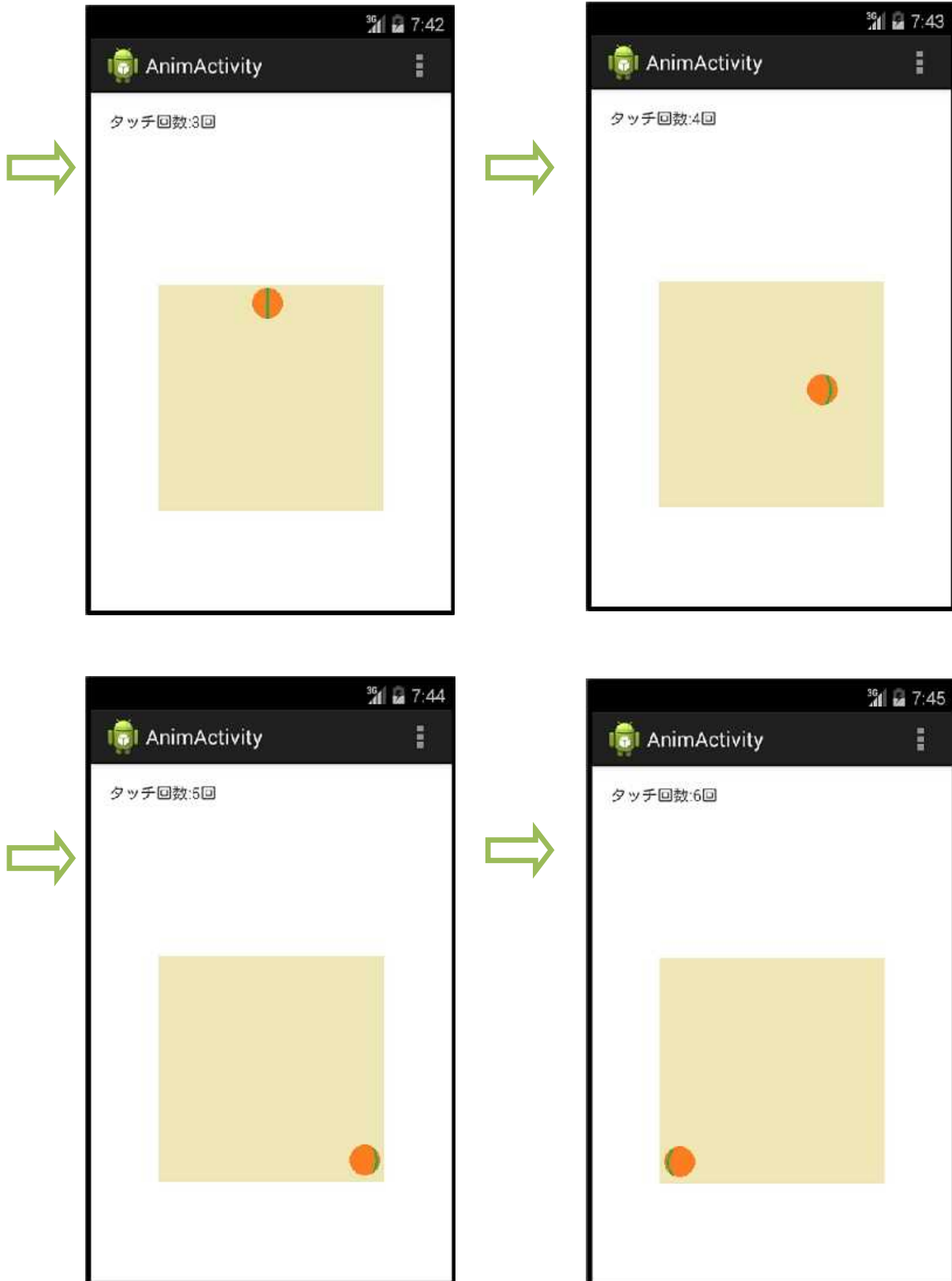
『保管』のアイコンをクリックして、**Main2Activity.java** を上書き保存し、パッケージ・エクスプローラーの『Second』を選択して、実行ボタンをクリックする。

(図は次のページ)



続けて画面上を何度かクリックする。





提出物 :

- 1) 画面のレイアウト設定ファイル `activity_anim.xml`
- 2) 画面のレイアウト設定ファイル `activity_main2.xml`
- 3) `AnimActivity` のソースファイル `AnimActivity.java`
- 4) `Main2Activity` のソースファイル `Main2Activity.java`