

2016 年 11 月 24 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、最も単純な手動のアニメーションを作成する。

## 必要となる Java の基礎知識 (第 3 回教材の続き)

### 2 分岐のプログラム

#### if 文

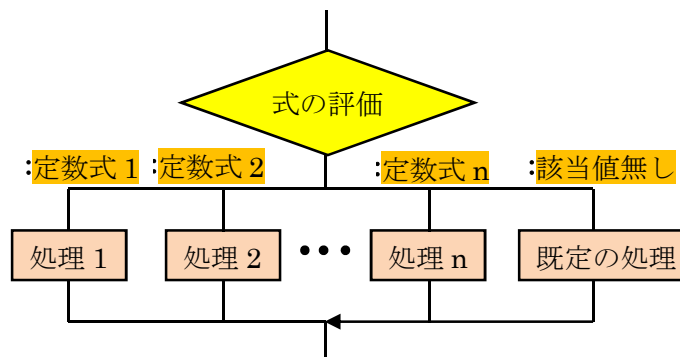
Java 言語で 2 分岐のプログラムを実現するための文として、if 文が用意されている。if 文の構文は次のようになる。

```
if ( 条件式 ) 真文 [ else 偽文 ]
```

ここで、[]内は省略可能であり、省略時は条件式が偽のときは何もしない場合を表す。また、真文または偽文を{}で囲んだブロックとして、複数の文で構成することが出来る。

### 多分岐のプログラム

多段階の 2 分岐を組み合わせることによって 3 種類以上の場合分けを実現出来るが、式の値の評価によって、一度に多種類の場合分けを行う多分岐の利用によって見通しのよいプログラムを作成出来る場合がある。(流れ図は右図)



#### switch 文

Java 言語で多分岐のプログラムを実現するための文として、switch 文が用意されている。switch 文の構文は次のようになる。

```
switch ( 制御式 ) {
    case 定数式 1: [文 1-1] [文 1-2] . . . [break;]
    case 定数式 2: [文 2-1] [文 2-2] . . . [break;]
    . . .
    case 定数式 n: [文 n-1] [文 n-2] . . . [break;]
    [default: [既定文 1] [既定文 2] . . . ]
}
```

ここで、[]内は省略可能である。switch 文の機能は、制御式を評価してその値が定数式(例えば、10, 'a' の様なリテラルや 1+2+3 の様な値の定まった式)の何れかの値に一致したとき、その case ラベルに続く文を break 文に出会うまで実行する。break 文に到達すると switch 文から抜け出す。もし、一致した case ラベルに続く文で break 文が省略されている場合には、その下の case ラベルに続く文も break 文に出会うまで実行する。

また、switch 文では、制御式を評価してその値が定数式の何れの値にも一致しないときは、default ラベルが書かれていれば、それに続く文を実行する。

## 授業の準備

### 1) Android Studio の初期設定

Android Studio を起動し、『Configure』→『設定のインポート』を選択し、第 3 回の教材の p.5 に従って設定をインポートする。

### 2) プロジェクトの新規作成

『Application name』(アプリ名)を「Prog\_7th」(先頭は大文字,「\_」は下線),『Company Domain』を「b6a0xxx.cuc.ac.jp」に書き換え、『Project Location』の先頭の「C:¥Users¥ b6a0xxx」を『H:』に書き換えて、『次へ』ボタンを押す。

第 1 回と同様に『Minimum SDK』では『API 22』を選択する(第 1 回教材 p.7)。

『Activity name』は「MainActivity7」とする。

### 3) AVD の設定

第 1 回の授業で作成した AVD の設定は H ドライブにあって残るが、SDK のシステムイメージは C ドライブにあるので、消失している。そこで、『Download』をクリックして、インストールし直す(第 2 回教材 p.4)。

## 課題

今回は、イメージビューを用いて、アニメーションの基本を学ぶ。

### Android アプリの作成

activity\_main7.xml のタブを開き、「Hello World!」と表示されているテキストビューをダブルクリックし、『id:』の右側のテキストボックスに「animTV」と入力し、『text:』の右側のテキストボックスを空にする。



『Palette』の『Widgets』からイメージビューを配置する。

H ドライブの H:¥AndroidStudioProjects¥Prog\_7th¥app¥src¥main¥res¥drawable 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。

MainActivity7.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.b6x0004.prog_7th;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.MotionEvent;
6 import android.widget.ImageView;
7 import android.widget.TextView;
8
9 public class MainActivity7 extends AppCompatActivity {
10     private TextView tv;
11     private ImageView iv;
12     private int n = 0;
13
14     @Override
15     protected void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main7);
18
19         tv = (TextView)findViewById(R.id.animTV);
20         iv = (ImageView)findViewById(R.id.imageView);
21     }
22
23     @Override
24     public boolean onTouchEvent(MotionEvent event) {
25         if (event.getAction() == MotionEvent.ACTION_DOWN) {
26             n++;
27             tv.setText("タップ回数:" + n + "回");
28             switch(n%5){
29                 case 1:
30                     iv.setImageResource(R.drawable.anim1);
31                     break;
32                 case 2:
33                     iv.setImageResource(R.drawable.anim2);
34                     break;
35                 case 3:
36                     iv.setImageResource(R.drawable.anim3);
37                     break;
38                 case 4:
39                     iv.setImageResource(R.drawable.anim4);
40                     break;
41                 case 0:
42                     iv.setImageResource(R.drawable.anim5);
43                     break;
44             }
45         }
46         return false;
47     }
48 }

```

- 1) 橙色の枠内 `MainActivity7` 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_main7.xml` の `animTV` を `tv`, `imageView` を `iv` にセット。
- 3) 赤色の枠内 `onTouchEvent` を用いて, 画面がタップ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, タップ回数を表示する。
  - 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。

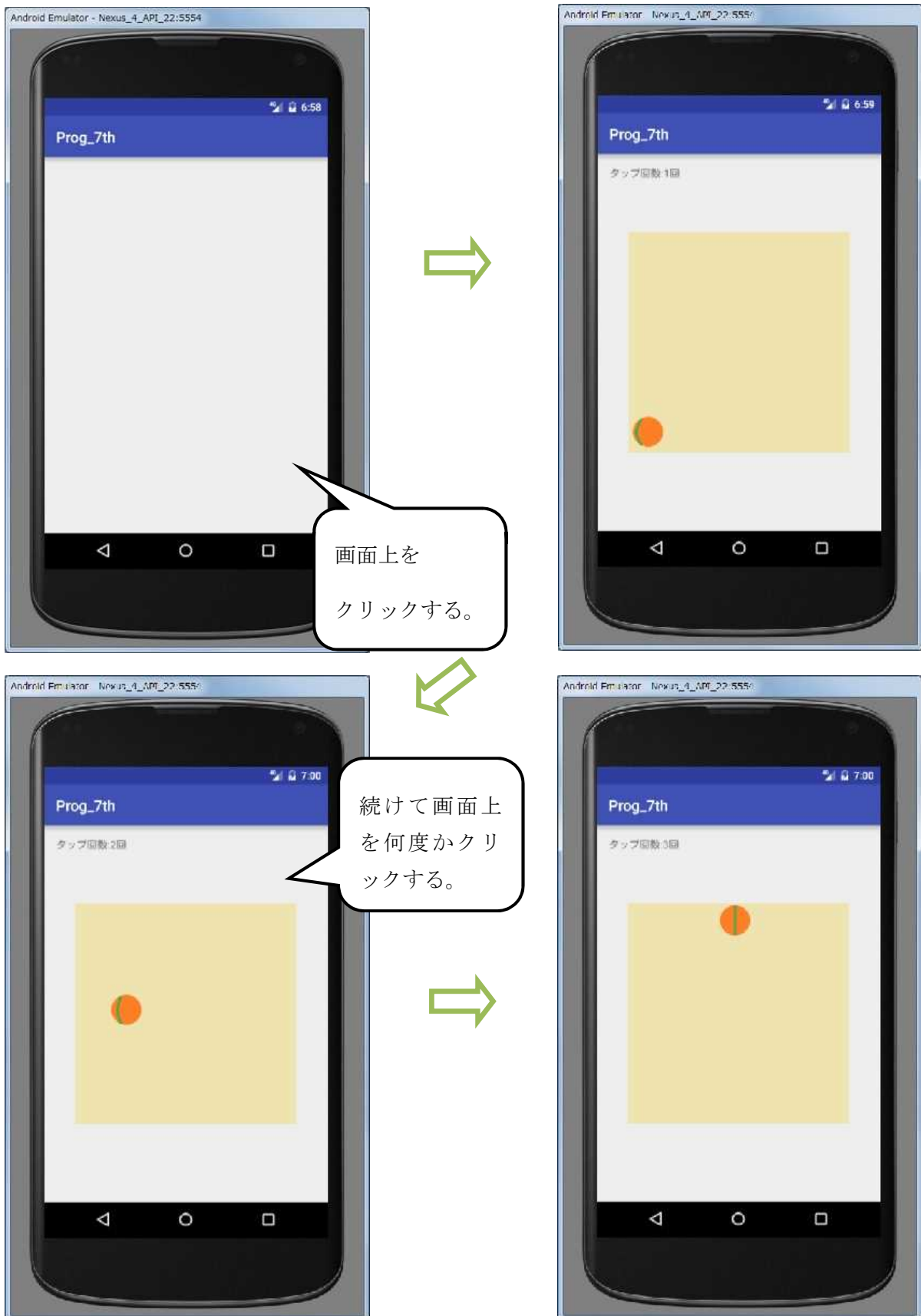
それぞれの色の枠内を次に示す。

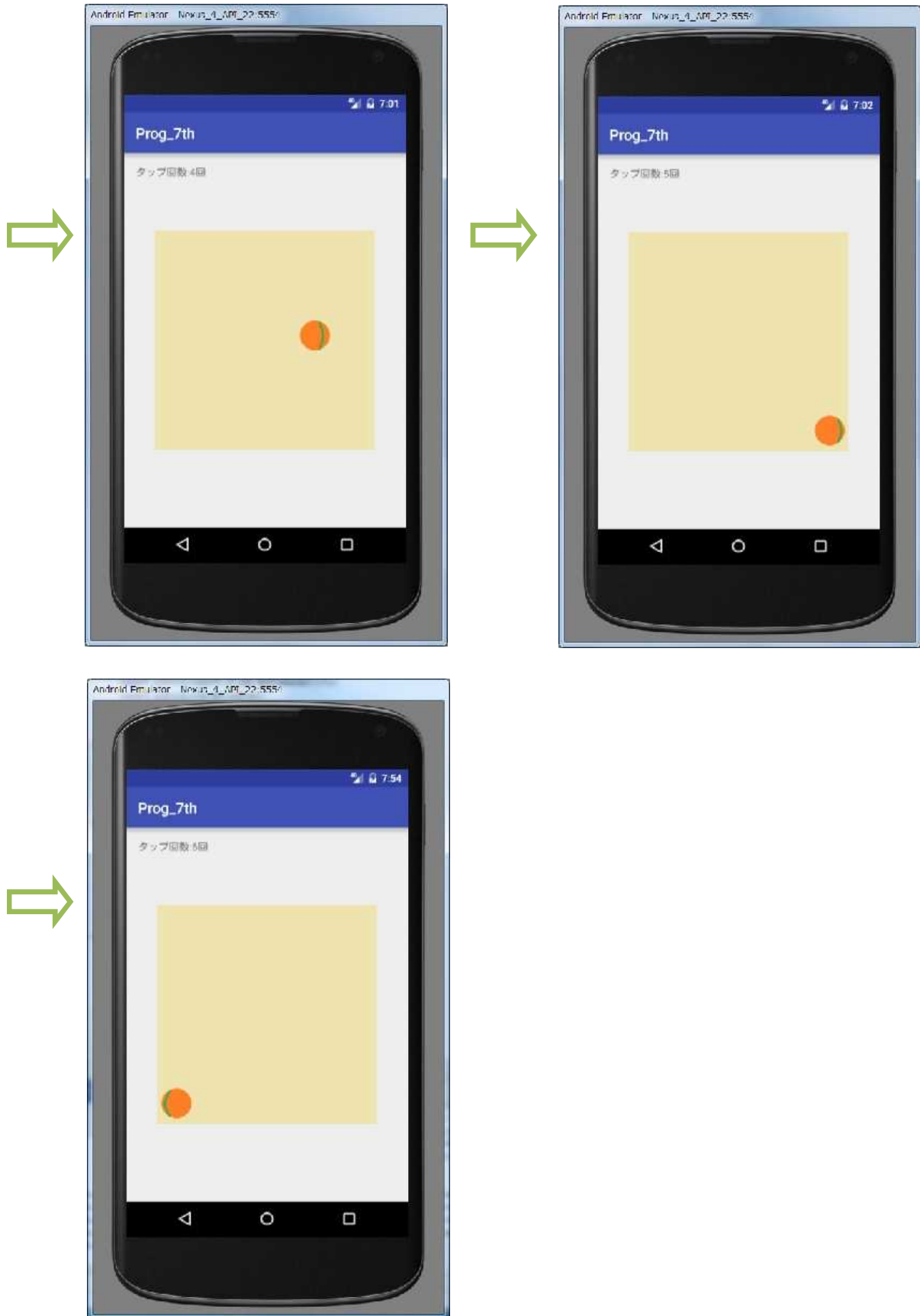
```
private TextView tv;
private ImageView iv;
private int n = 0;
```

```
tv = (TextView)findViewById(R.id.animTV);
iv = (ImageView)findViewById(R.id.imageView);
```

```
@Override
public boolean onTouchEvent(MotionEvent event) {
    if (event.getAction() == MotionEvent.ACTION_DOWN) {
        n++;
        tv.setText("タップ回数:" + 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;
}
```

『保存』のアイコンをクリックして、全てのファイルを上書き保存し、実行ボタンをクリックする。





提出物：

- 1) 画面のレイアウト設定ファイル `activity_main7.xml`
- 2) アクティビティのソースファイル `MainActivity7.java`