

● アンドロイドの簡単なアニメーション機能

- ・ 移動
- ・ 透明化
- ・ 回転
- ・ 拡大

の4パターンが紹介されています。各クラスのコンストラクタの引数を任意に変更して、どう動くか、繰り返し試してみてください。

実行時画面



レイアウト定義は以下の通りです。

■ main.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<LinearLayout
```

```
    xmlns:android="http://schemas.android.com/apk/res/android"
```

```

        android:orientation="vertical"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:gravity="center_horizontal | top"
    >
<ImageView
    android:id="@+id/animeview"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="50px"
    >
</ImageView>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_marginTop="30px"
    >
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Translate"
    android:id="@+id/translate"
    >
</Button>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Alpha"
    android:id="@+id/alpha"
    >
</Button>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```

```

        android:text="Rotate"
        android:id="@+id/rotate"
    >
</Button>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Scale"
    android:id="@+id/scale"
    >
</Button>
</LinearLayout>
<LinearLayout
    android:orientation="horizontal"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:layout_marginTop="10px"
    >
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mix1"
    android:id="@+id/mix1"
    >
</Button>
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Mix2"
    android:id="@+id/mix2"
    >
</Button>
</LinearLayout>
</LinearLayout>

```

以下、メインアクティビティ画面の Java コードです。

■MainAnimationActivity.java

```
package sample.animation;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.animation.AlphaAnimation;
import android.view.animation.AnimationSet;
import android.view.animation.CycleInterpolator;
import android.view.animation.RotateAnimation;
import android.view.animation.ScaleAnimation;
import android.view.animation.TranslateAnimation;
import android.widget.Button;
import android.widget.ImageView;

/**
 * 簡易アニメーションクラス一覧の使用例
 */
public class MainAnimationActivity extends Activity implements OnClickListener {
    //画面部品群
    ImageView iv;
    Button btntrans, btnalpha, btnrotate, btnscale, btnmix1, btnmix2;

    //アニメーションクラスの参照変数
    TranslateAnimation translate;
    AlphaAnimation alpha;
    RotateAnimation rotate;
    ScaleAnimation scale;
    AnimationSet mix1, mix2;

    @Override
```

```

public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main);
    iv = (ImageView) findViewById(R.id.animeview);
    // 動画対象の画像をセット
    iv.setImageResource(R.drawable.icon);

    // ボタンの設定
    btntrans = (Button) findViewById(R.id.translate);
    btntrans.setOnClickListener(this);
    btnalpha = (Button) findViewById(R.id.alpha);
    btnalpha.setOnClickListener(this);
    btnrotate = (Button) findViewById(R.id.rotate);
    btnrotate.setOnClickListener(this);
    btnscale = (Button) findViewById(R.id.scale);
    btnscale.setOnClickListener(this);
    btnmix1 = (Button) findViewById(R.id.mix1);
    btnmix1.setOnClickListener(this);
    btnmix2 = (Button) findViewById(R.id.mix2);
    btnmix2.setOnClickListener(this);

    /*
     * 以下、各種アニメーションの設定
     */

    // *** set translate 移動 ***
    // TranslateAnimation(float fromX, float toX, float fromY, float toY)
    // 引数説明 (x 始点、x 終点、y 始点、y 終点)
    translate = new TranslateAnimation(0, 30, 0, 30);
    // set duration to 3000ms
    // 移動時間をミリ秒で設定
    translate.setDuration(3000);
    // repeat 3 times
    // 繰り返し回数を設定
    translate.setInterpolator(new CycleInterpolator(3));

```

```

// *** set alpha 透過 ***
// AlphaAnimation(float fromAlpha, float toAlpha)
// 引数説明 (アニメーション開始時の透明度、終了時の透明度)
// 透明度を 0.0～1.0 の間で指定。0.0 は完全透明、1.0 は完全不透明
alpha = new AlphaAnimation(1, 0);
alpha.setDuration(5000);
alpha.setInterpolator(new CycleInterpolator(3));

// *** set rotate 回転 ***
// RotateAnimation(float fromDegrees, float toDegrees, float pivotX, float pivotY)
// 引数説明 (開始角度、終了角度、回転の X 軸の値、回転の Y 軸の値)
// 開始角度と終了角度は、例えば 0.0 から 360 を設定すると、時計回りに一回転します。

rotate = new RotateAnimation(0, 360, 0, 0);
rotate.setDuration(3000);
rotate.setInterpolator(new CycleInterpolator(1));

// *** set scale 拡大・縮小 ***
// ScaleAnimation(float fromX, float toX, float fromY, float toY, float pivotX, float pivotY)
// 引数説明 (開始時の X のスケール、終了時の X のスケール、開始時の Y スケール、終了時の Y スケール、拡大の X 軸の値、拡大の Y 軸の値)

scale = new ScaleAnimation(1, 2, 1, 2, 150, 150);
scale.setDuration(3000);
scale.setInterpolator(new CycleInterpolator(3));

// *** set mix1 交合パターン 1 ***
// アニメーションセット オブジェクトの生成
mix1 = new AnimationSet(true);
// セットに単体アニメーションを追加
mix1.addAnimation(rotate);
mix1.addAnimation(scale);
mix1.setInterpolator(new CycleInterpolator(2));

// *** set mix2 交合パターン 2 ***
mix2 = new AnimationSet(true);
mix2.addAnimation(rotate);

```

```

        mix2.addAnimation(translate);
        mix2.addAnimation(alpha);
        mix2.setInterpolator(new CycleInterpolator(2));
    }

    //各ボタン押下時にアニメーションを開始する
    public void onClick(View view) {
        switch (view.getId()) {
            case R.id.translate:
                // start translate
                iv.startAnimation(translate);
                break;
            case R.id.alpha:
                // start alpha
                iv.startAnimation(alpha);
                break;
            case R.id.rotate:
                // start rotate
                iv.startAnimation(rotate);
                break;
            case R.id.scale:
                // start scale
                iv.startAnimation(scale);
                break;
            case R.id.mix1:
                iv.startAnimation(mix1);
                break;
            case R.id.mix2:
                iv.startAnimation(mix2);
                break;
            default:
                break;
        }
    }
} // onClick()
} // class_end

```