

```
[1]MyCointossEng.java
```

```
/*
```

```
-----  
    2個のコイントス(英語版)  
    Android 4.1 (Jelly Bean)  
    Copyright (C) K.Niwa 2021. 9. 26  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mycointosseng;
```

```
import android.content.Context;
```

```
import android.graphics.Canvas;
```

```
import android.graphics.Color;
```

```
import android.graphics.Paint;
```

```
import android.graphics.Rect;
```

```
import android.util.AttributeSet;
```

```
import android.view.View;
```

```
import android.content.res.Resources; //画像用
```

```
import android.graphics.*;
```

```
import android.view.*;
```

```
public class MyCointossEng extends View {
```

```
    private Bitmap bitmap1 = null;
```

```
    private Bitmap bitmap2 = null;
```

```
    int ct=0; //実験回数カウンター
```

```
    int r1,r2; //コイン1、コイン2 のそれぞれの表裏の識別子(乱数)
```

```
    int d1=0, d2=0, d3=0, d4=0; //[表表][表裏][裏表][裏裏]のカウンター
```

```
    int flag=0; //自動識別子
```

```
    int syoki=0; //初期化識別子
```

```
    int width;
```

```
    int height;
```

```

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

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

public MyCointossEng(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    init(context);
}

private void init(Context context) {
    // TODO 自動生成されたメソッド・スタブ
    Resources res = context.getResources();
    bitmap1 = BitmapFactory.decodeResource(res, R.drawable.coin1);
    bitmap2 = BitmapFactory.decodeResource(res, R.drawable.coin2);

    //WindowManager wm = (WindowManager)context.getSystemService(Context.WINDOW_SERVICE);
    //Display disp = wm.getDefaultDisplay();
    //width = disp.getWidth();
    //height = disp.getHeight();
}

@Override
protected void onDraw(Canvas canvas) {
    // TODO 自動生成されたメソッド・スタブ

    float a=0;
    float b=0;

    super.onDraw(canvas);
    canvas.drawColor(Color.WHITE);
}

```

```

Paint paint = new Paint();
paint.setColor(Color.BLUE);
paint.setAlpha(50);
canvas.drawRect((getWidth()/2-360)+20, (getHeight()/2-600)+10, (getWidth()/2-
360)+700, (getHeight()/2-600)+1190, paint);

paint.setAlpha(10000);
paint.setColor(Color.BLUE);

for (int i=0;i<2;i++) {
    canvas.drawLine((getWidth()/2-360)+20+i, (getHeight()/2-600)+10+i, (getWidth()/2-
360)+20+i, (getHeight()/2-600)+1190-i, paint);
    canvas.drawLine((getWidth()/2-360)+20+i, (getHeight()/2-600)+1190-i, (getWidth()/2-
360)+700-i, (getHeight()/2-600)+1190-i, paint);
    canvas.drawLine((getWidth()/2-360)+700-i, (getHeight()/2-600)+1190-i, (getWidth()/2-
360)+700-i, (getHeight()/2-600)+10+i, paint);
    canvas.drawLine((getWidth()/2-360)+700-i, (getHeight()/2-600)+10+i, (getWidth()/2-
360)+20+i, (getHeight()/2-600)+10+i, paint);
}

paint.setColor(Color.BLUE);
paint.setTextSize(45.0f);
canvas.drawText("【Two Coin Toss】", (getWidth()/2-360)+145-20+20+25, (getHeight()/2-
600)+80, paint);

paint.setColor(Color.BLACK);
paint.setTextSize(25.0f);
canvas.drawText("coin 1", (getWidth()/2-360)+145+120+10, (getHeight()/2-600)+185+30,
paint);
canvas.drawText("coin 2", (getWidth()/2-360)+265+120+10, (getHeight()/2-600)+185+30,
paint);

if (MainActivity.ritsu != 0) {
    a=(float)1.0*320/MainActivity.ritsu; //----- <画像の拡大・縮小の横の倍率を指定する>
    b=(float)1.0*320/MainActivity.ritsu; //----- <画像の拡大・縮小の縦の倍率を指定する>
}

```

大・縮小の縦の倍率を指定する>

```
    }
    else {
        a=(float) 1.0;
        b=(float) 1.0;
    }

    Matrix Mat = new Matrix(); //-----***
    Mat.postScale(a, b); //-----***

    Bitmap bitmap11 = Bitmap.createBitmap( //-----***
        bitmap1, 0, 0, //-----***
        bitmap1.getWidth(), //-----***
        bitmap1.getHeight(), //-----***
        Mat, true //-----***
    ); //-----***

    //Matrix Mat = new Matrix(); //-----***
    //Mat.postScale(a, b); //-----***
    Bitmap bitmap22 = Bitmap.createBitmap( //-----***
        bitmap2, 0, 0, //-----***
        bitmap2.getWidth(), //-----***
        bitmap2.getHeight(), //-----***
        Mat, true //-----***
    ); //-----***

    if (bitmap11 != null && bitmap22 != null) {
        ct++;

        r1=(int) (1+2*Math.random());
        if (r1==1) {
            canvas.drawBitmap(bitmap11, (getWidth()/2-360)+160-4+120, (getHeight()/2-
600)+130-5, paint);
        }
        else if (r1==2) {
```

```

        canvas.drawBitmap(bitmap22, (getWidth()/2-360)+160-4+120, (getHeight()/2-
600)+130-5, paint);
    }

    r2=(int) (1+2*Math.random());
    if (r2==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-360)+280-4+120, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r2==2) {
        canvas.drawBitmap(bitmap22, (getWidth()/2-360)+280-4+120, (getHeight()/2-
600)+130-5, paint);
    }
}

if (r1==1 && r2==1) {
    d1++;
}
else if (r1==1 && r2==2) {
    d2++;
}
else if (r1==2 && r2==1) {
    d3++;
}
else if (r1==2 && r2==2) {
    d4++;
}

//碎
paint.setColor(Color.BLACK);
canvas.drawLine((getWidth()/2-360)+130, (getHeight()/2-600)+240, (getWidth()/2-
360)+590, (getHeight()/2-600)+240, paint);
canvas.drawLine((getWidth()/2-360)+130, (getHeight()/2-600)+280, (getWidth()/2-
360)+590, (getHeight()/2-600)+280, paint);
canvas.drawLine((getWidth()/2-360)+130, (getHeight()/2-600)+320, (getWidth()/2-
360)+590, (getHeight()/2-600)+320, paint);

```

```

        canvas.drawLine((getWidth()/2-360)+130, (getHeight()/2-600)+360, (getWidth()/2-
360)+590, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+130, (getHeight()/2-600)+240, (getWidth()/2-
360)+130, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+270, (getHeight()/2-600)+240, (getWidth()/2-
360)+270, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+350, (getHeight()/2-600)+240, (getWidth()/2-
360)+350, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+430, (getHeight()/2-600)+240, (getWidth()/2-
360)+430, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+510, (getHeight()/2-600)+240, (getWidth()/2-
360)+510, (getHeight()/2-600)+360, paint);
        canvas.drawLine((getWidth()/2-360)+590, (getHeight()/2-600)+240, (getWidth()/2-
360)+590, (getHeight()/2-600)+360, paint);

```

```

        paint.setColor(Color.BLACK);
        paint.setTextSize(30.0f);
        canvas.drawText("coin 1", (getWidth()/2-360)+150+10, (getHeight()/2-600)+270, paint);
        canvas.drawText("coin 2", (getWidth()/2-360)+150+10, (getHeight()/2-600)+310, paint);
        canvas.drawText("frequency", (getWidth()/2-360)+150-17, (getHeight()/2-600)+350,
paint);

```

```

        canvas.drawText("table", (getWidth()/2-360)+295-20, (getHeight()/2-600)+270, paint);
        canvas.drawText("table", (getWidth()/2-360)+295-20, (getHeight()/2-600)+310, paint);
        canvas.drawText("table", (getWidth()/2-360)+375-20, (getHeight()/2-600)+270, paint);
        canvas.drawText("back", (getWidth()/2-360)+375-20, (getHeight()/2-600)+310, paint);
        canvas.drawText("back", (getWidth()/2-360)+455-20, (getHeight()/2-600)+270, paint);
        canvas.drawText("table", (getWidth()/2-360)+455-20, (getHeight()/2-600)+310, paint);
        canvas.drawText("back", (getWidth()/2-360)+535-20, (getHeight()/2-600)+270, paint);
        canvas.drawText("back", (getWidth()/2-360)+535-20, (getHeight()/2-600)+310, paint);

```

```

        paint.setColor(Color.BLUE);
        paint.setTextSize(34.0f);
        canvas.drawText(""+d1, (getWidth()/2-360)+275-3, (getHeight()/2-600)+350+3, paint);
        canvas.drawText(""+d2, (getWidth()/2-360)+355-3, (getHeight()/2-600)+350+3, paint);
        canvas.drawText(""+d3, (getWidth()/2-360)+435-3, (getHeight()/2-600)+350+3, paint);

```

```

canvas.drawText(""+d4, (getWidth()/2-360)+515-3, (getHeight()/2-600)+350+3, paint);

paint.setColor(Color.BLUE);
paint.setTextSize(40.0f);
canvas.drawText("Number of experiments = "+ct, (getWidth()/2-360)+100-30,
(getHeight()/2-600)+440-10, paint);

paint.setColor(Color.BLACK);
canvas.drawText("Percentage of [table·table]", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+500-10, paint);
canvas.drawText("= "+((float)(d1)/(float)(ct)), (getWidth()/2-360)+100-30+330,
(getHeight()/2-600)+540-10, paint);
canvas.drawText("Percentage of [table·back]", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+600-10-10, paint);
canvas.drawText("= "+((float)(d2)/(float)(ct)), (getWidth()/2-360)+100-30+330,
(getHeight()/2-600)+640-10-10, paint);
canvas.drawText("Percentage of [back·table]", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+700-20-10, paint);
canvas.drawText("= "+((float)(d3)/(float)(ct)), (getWidth()/2-360)+100-30+330,
(getHeight()/2-600)+740-20-10, paint);
canvas.drawText("Percentage of [back·back]", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+800-30-10, paint);
canvas.drawText("= "+((float)(d4)/(float)(ct)), (getWidth()/2-360)+100-30+330,
(getHeight()/2-600)+840-30-10, paint);

paint.setColor(Color.BLUE);
paint.setTextSize(40.0f);
canvas.drawText("■Let's observe how each ratio ", (getWidth()/2-360)+50+20,
(getHeight()/2-600)+850, paint);
canvas.drawText(" approaches 0.25 .", (getWidth()/2-360)+50+20, (getHeight()/2-
600)+890, paint);

paint.setColor(Color.BLACK);
paint.setTextSize(30.0f);
canvas.drawText("Touch the screen five times to activate.", (getWidth()/2-360)+50,
(getHeight()/2-600)+950, paint);

```

```

        canvas.drawText("Touch the screen again to stop the auto.", (getWidth()/2-360)+50,
(getHeight()/2-600)+990, paint);
        canvas.drawText("If you touch it further, it will be initialized.", (getWidth()/2-
360)+50, (getHeight()/2-600)+1030, paint);
        canvas.drawText("When the screen goes dark, touch the title bar !", (getWidth()/2-
360)+50, (getHeight()/2-600)+1070, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0f);
        canvas.drawText("Copyright(C) Sohun 2021.9.26", (getWidth()/2-360)+150+10,
(getHeight()/2-600)+1130, paint);

        if (flag >= 5) {
            if (d1<9998 && d2<9998 && d3<9998 && d4<9998) {
                invalidate();
            }
        }

} //protected void onDraw(Canvas canvas)

@Override
public boolean onTouchEvent(MotionEvent event) {
    flag++;
    flag = flag % 6;

    syoki++;
    if (syoki > 6) {
        ct=0;
        d1=0;d2=0;d3=0;d4=0;
        flag=0;
        syoki=0;
    }

    invalidate();
    return false;
}

```



```
}
```

[2]activity_main.xml

```
<?xml version="1.0" encoding="utf-8" ?>  
<androidx.constraintlayout.widget.ConstraintLayout  
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"  
tools:context=".MainActivity">  
  
    <TextView  
        android:layout_width="wrap_content"  
        android:layout_height="wrap_content"  
        android:text="Hello World!"  
        app:layout_constraintBottom_toBottomOf="parent"  
        app:layout_constraintLeft_toLeftOf="parent"  
        app:layout_constraintRight_toRightOf="parent"  
        app:layout_constraintTop_toTopOf="parent" />  
  
    <jp.kiyo.wuena.mycointosseng.MyCointossEng  
        android:id="@+id/myfview1"  
        android:layout_height="match_parent"  
        android:layout_width="match_parent"/>  
  
</androidx.constraintlayout.widget.ConstraintLayout>
```

[3]MainActivity.java

```
/*
```

*2個のコイントス (英語版)
Android 4.1 (Jelly Bean)*

*/

```
package jp.kiyo.wuena.mycointosseng;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.util.DisplayMetrics;    //<画像の拡大・縮小に必要なライブラリ>
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends AppCompatActivity {

    static int ritsu;

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

        DisplayMetrics metrics = new DisplayMetrics(); //<端末の情報を取得する>
        getWindowManager().getDefaultDisplay().getMetrics(metrics);
        StringBuilder buffer = new StringBuilder();
        buffer.append("densityDpi (ドット数/インチ) : " + String.valueOf(metrics.densityDpi)
+ "\n");
        ritsu=metrics.densityDpi;
    }
}
```