

[ 1 ] MyTeppou.java

/\*

-----  
下手な鉄砲も数撃ちゃ当たる！  
Android 4.4 (Kit Kat)  
Copyright(C) K.Niwa 2019.12.13  
-----

\*/

package jp.kiyo.wuena.myteppou;

import android.content.Context;  
import android.graphics.Canvas;  
import android.graphics.Color;  
import android.graphics.Paint;  
import android.graphics.Rect;  
import android.graphics.RectF;  
import android.util.AttributeSet;  
import android.view.MotionEvent;  
import android.view.View;

public class MyTeppou extends View {

int px,py; //玉の位置  
double i; //的の描写に使用  
int flag=0; //自動識別子 (flag==1:自動 start、flag==2:自動 stop、  
flag==0:発射回数・当たり回数・実験回数等の初期化)  
int ct1=0; //発射回数  
int ct2=0; //当たり回数  
int ct3=0; //実験回数  
int seikou=0; //成功回数  
float ritu=0; //成功率  
double r1; //動径の長さ 0 以上 1 0 0 未満 (乱数)  
double r2; //偏角の大きさ 0 以上 2 π 未満 (乱数)

public MyTeppou(Context context, AttributeSet attrs, int defStyle) {  
super(context, attrs, defStyle);  
// TODO 自動生成されたコンストラクター・スタブ  
init(context);  
}

```

public MyTeppou(Context context, AttributeSet attrs) {
    super(context, attrs);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

public MyTeppou(Context context) {
    super(context);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

@Override
protected void onDraw(Canvas canvas) {
    // TODO 自動生成されたメソッド・スタブ
    super.onDraw(canvas);
    canvas.drawColor(Color.WHITE);
    Paint paint = new Paint();
    paint.setColor(Color.BLUE);
    paint.setAlpha(50);
    canvas.drawRect((getWidth()/2-240)+10,(getHeight()/2-343)+10,(getWidth()/2-240)
+470,(getHeight()/2-343)+675, paint);

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

    for (int i=0;i<2;i++) {
        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+10+i,(getWidth
()/2-240)+10+i,(getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+675-i,(getWidth
()/2-240)+470-i,(getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+675-i,(getWidth
()/2-240)+470-i,(getHeight()/2-343)+10+i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+10+i,(getWidth
()/2-240)+10+i,(getHeight()/2-343)+10+i, paint);
    }

    paint.setColor(Color.BLUE);
    paint.setTextSize(23.0f);
}

```

```

        canvas.drawText("【下手な鉄砲も数撃ちゃ当たる！】", (getWidth()/2-240)+80-24,
(getHeight()/2-343)+70, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright(C) K.Niwa 2019.11", (getWidth()/2-240)+120,
(getHeight()/2-343)+630, paint);

        //枠の作成
        paint.setColor(Color.BLACK);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+90, (getWidth()
/2-240)+390, (getHeight()/2-343)+90, paint);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+90, (getWidth()
/2-240)+90, (getHeight()/2-343)+390, paint);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+390, (getWidth()
/2-240)+390, (getHeight()/2-343)+390, paint);
        canvas.drawLine((getWidth()/2-240)+390, (getHeight()/2-343)+90, (getWidth()
/2-240)+390, (getHeight()/2-343)+390, paint);
        canvas.drawLine((getWidth()/2-240)+91, (getHeight()/2-343)+91, (getWidth()
/2-240)+389, (getHeight()/2-343)+91, paint);
        canvas.drawLine((getWidth()/2-240)+91, (getHeight()/2-343)+91, (getWidth()
/2-240)+91, (getHeight()/2-343)+389, paint);
        canvas.drawLine((getWidth()/2-240)+91, (getHeight()/2-343)+389, (getWidth()
/2-240)+389, (getHeight()/2-343)+389, paint);
        canvas.drawLine((getWidth()/2-240)+389, (getHeight()/2-343)+91, (getWidth()
/2-240)+389, (getHeight()/2-343)+389, paint);

        //的の描写
        paint.setColor(Color.WHITE);
        canvas.drawCircle((getWidth()/2-240)+240, (getHeight()/2-343)+240, 10, paint);

        //発射回数を1～20回に設定する
        ct1=ct1 % 20;
        if (ct1==0) { //発射回数が0回になったら、当たり回数を0回にし、実験回
数を1回増やす
            ct2=0;
            ct3++;
        }

        if (ct1<20) { //発射回数が19回目までのとき

```

```

ct1=ct1+1;
r1=Math.random(); //極座標の動径の長さ(0以上100未満を作る)
r2=Math.random(); //極座標の偏角の大きさ(0以上2π未満を作る)
px=(int)(150+90+100*r1*Math.cos(2*Math.PI*r2));
py=(int)(150+90-100*r1*Math.sin(2*Math.PI*r2));

//鉄砲の弾の描写
paint.setColor(Color.RED);
canvas.drawCircle((getWidth()/2-240)+px, (getHeight()/2-343)+py, 3, paint);

if (r1<0.1) {
    ct2=ct2+1; //当たり回数を1つ増やす
}

if (ct1==20 && ct2>0) { //発射20回目で
    当たり回数が1回以上のとき
        seikou=seikou+1; //成功回数を1回増やす
        paint.setColor(Color.BLUE);
        paint.setTextSize(60.0f);
        canvas.drawText("成功!!", (getWidth()/2-240)+100, (getHeight()/2-343)
+180, paint);
    }
    else if (ct1==20 && ct2==0) { //発射20回目で当たり回数が0回
    のとき
        paint.setColor(Color.RED);
        paint.setTextSize(60.0f);
        canvas.drawText("失敗!!", (getWidth()/2-240)+100, (getHeight()/2-343)
+180, paint);
    }
}

paint.setColor(Color.BLACK);
paint.setTextSize(19.0f);
canvas.drawText("発射回数 = "+ct1, (getWidth()/2-240)+190, (getHeight()/2-343)
+120, paint);
canvas.drawText("当たり回数 = "+ct2, (getWidth()/2-240)+180, (getHeight()/2-343)
+375, paint);
canvas.drawText("成功回数／実験回数 = "+seikou+ " / "+ct3, (getWidth()/2-240)
+100, (getHeight()/2-343)+420, paint);

```

```

    ritu=(float) seikou/(float) ct3;
    paint.setColor(Color.BLUE);
    paint.setTextSize(19.0f);
    canvas.drawText("= "+ritu, (getWidth()/2-240)+270,(getHeight()/2-343)+520-20-60,
paint);

```

```

    paint.setColor(Color.BLUE);
    paint.setTextSize(19.0f);
    canvas.drawText("少なくとも1回当たる数学的確率= 0.8784233", (getWidth()
/2-240)+30,(getHeight()/2-343)+470, paint);

```

```

    paint.setColor(Color.BLACK);
    paint.setTextSize(18.0f);
    canvas.drawText("※ 10回に1回当たる鉄砲の腕前の人20回撃つ",
(getWidth()/2-240)+30,(getHeight()/2-343)+500, paint);
    canvas.drawText("とき、少なくとも1回当たる確率を考えます。", (getWidth()
/2-240)+30+10,(getHeight()/2-343)+520, paint);
    canvas.drawText("※ 画面をタッチすると、[自動開始] → [自動停止]", (getWidth
()/2-240)+30,(getHeight()/2-343)+550, paint);
    canvas.drawText(" → [初期化] の順に繰り返します。", (getWidth()/2-240)+30,
(getHeight()/2-343)+570, paint);
    canvas.drawText("※ 画面が暗くなったらタイトルバーをタッチ!", (getWidth()
/2-240)+30,(getHeight()/2-343)+600, paint);

```

```

    if (flag==1) { //flag==1 のとき、自動発射がスタートする。
        invalidate();
    }

```

```

}

```

```

public void init(Context context) {
}

```

```

@Override

```

```

public boolean onTouchEvent(MotionEvent event) {
    flag++; //flag==2 のとき、自動発射がストップする。
    flag = flag % 3; //自動識別子 (flag==1:自動 start、flag==2:自動 stop、flag==0:発
射回数等の初期化)
    invalidate();
}

```

```

        if (flag==0) { //flag==0 のとき、発射回数等の初期化を行う。
            ct1=0; //発射回数
            ct2=0; //当たり回数
            ct3=0; //実験回数
            seikou=0; //成功回数
            ritu=0; //成功率
        }

        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.myteppou.MyTeppou
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

```

</androidx.constraintlayout.widget.ConstraintLayout>

[ 3 ] MainActivity.java

/\*

-----  
下手な鉄砲も数撃ちゃ当たる！  
Android 4.4 (Kit Kat)  
Copyright (C) K.Niwa 2019.12.13  
-----

\*/

package jp.kiyo.wuena.myteppou;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class MainActivity extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

    }

}