

```
[1]MyGoma2.java
```

```
/*
```

```
-----  
    ゴマをまいて円周率  $\pi$  を求める  
    Android 4.1 (Jelly Bean)  
    Copyright (C) K. Niwa 2021. 2. 7  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mygoma2;
```

```
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 MyGoma2 extends View {
```

```
    int i; //変数宣言と初期化
```

```
    int sum=0; //円に入ったゴマの個数
```

```
    int px; //ゴマの位置のx座標を整数型にしたもの
```

```
    int py; //ゴマの位置のy座標を整数型にしたもの
```

```
    int N=9999; //実験回数の最大値
```

```
    int flag=0; //ゴマをまくか(1)、否か(0)、初期化する(2)の識別子
```

```
    double pai;
```

```
    int k=0; //まいたゴマの個数
```

```
    double[] x=new double[10001];
```

```
    double[] y=new double[10001];
```

```
    private Canvas canvas;
```

```
    public MyGoma2(Context context) {
```

```
    super(context);  
}
```

```
public MyGoma2(Context context, AttributeSet attrs) {  
    super(context, attrs);  
}
```

```
public MyGoma2(Context context, AttributeSet attrs, int defStyle) {  
    super(context, attrs, defStyle);  
}
```

```
//onDraw メソッド-----
```

```
-----  
@Override
```

```
protected void onDraw(Canvas canvas) {  
    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.BLACK); //実 験枠の描画  
    canvas.drawRect((getWidth()/2-360)+90+120, (getHeight()/2-600)+100, (getWidth()/2-
```

```

360)+390+120, (getHeight()/2-600)+400, paint);
    paint.setColor(Color.WHITE);
    canvas.drawRect((getWidth()/2-360)+91+120, (getHeight()/2-600)+101, (getWidth()/2-
360)+389+120, (getHeight()/2-600)+399, paint);
    paint.setColor(Color.BLACK); //扇形の描画
    RectF rect1 = new RectF((getWidth()/2-360)+90-300+120, (getHeight()/2-600)+100-
300, (getWidth()/2-360)+390+120, (getHeight()/2-600)+400);
    canvas.drawArc(rect1, 0f, 90f, true, paint); paint.setColor(Color.WHITE);
    RectF rect2 = new RectF((getWidth()/2-360)+90-300+120, (getHeight()/2-600)+100-
300, (getWidth()/2-360)+390-1+120, (getHeight()/2-600)+400-1);
    canvas.drawArc(rect2, 0f, 90f, true, paint);
    paint.setColor(Color.BLACK); //実 験 枠 の 描 画
    canvas.drawLine((getWidth()/2-360)+90+120, (getHeight()/2-600)+100, (getWidth()/2-
360)+90+120, (getHeight()/2-600)+400, paint);
    canvas.drawLine((getWidth()/2-360)+90+120, (getHeight()/2-600)+100, (getWidth()/2-
360)+390+120, (getHeight()/2-600)+100, paint);
    paint.setColor(Color.BLUE);
    //表題の表示
    paint.setTextSize(40.0f);
    canvas.drawText("【ゴマをまいて円周率πを求める】", (getWidth()/2-360)+25+40,
(getHeight()
        /2-600)+65, paint);
    paint.setColor(Color.BLACK); //説明の表示
    paint.setTextSize(30.0f);
    canvas.drawText("※ 画面をタッチするとゴマを自動でまきます。", (getWidth()/2-360)+50,
(getHeight()/2-600)+95, paint);
    canvas.drawText("※ もう一度タッチすると止まります。", (getWidth()/2-360)+50,
(getHeight()/2-600)+99, paint);
    canvas.drawText("※ 更にタッチすると初期化されます。", (getWidth()/2-360)+50,
(getHeight()/2-600)+103, paint);
    canvas.drawText("※ 画面が暗くなったらタイトルバーをタッチ!", (getWidth()/2-360)+50,
(getHeight()/2-600)+107, paint);
    //if (k==0) {
        canvas.drawText("■ (円に入ったゴマの個数)を(まいたゴマの総数)", (getWidth()/2-
360)+50, (getHeight()
            /2-600)+140+600+60, paint);

```

```

        canvas.drawText(" で割った値の4倍が円周率πになっているこ", (getWidth()/2-
360)+50, (getHeight
        ()/2-600)+170+600+60+5, paint);
        canvas.drawText(" とを観察してみましょう。", (getWidth()/2-360)+50, (getHeight
        ()/2-600)+200+600+60+10, paint);

//
paint.setColor(Color.BLUE);
paint.setTextSize(30.0f);
canvas.drawText("Copyright(C) K.Niwa 2021.2.7", (getWidth()/2-360)+150,
(getHeight()/2-600)+1130, paint); //作者・作成年月の表示
k=k+1;
//まくゴマの個数を1個増やす
x[k]=Math.random(); //落ちるゴマの位置(x, y)を乱数で求める
y[k]=Math.random();
if (x[k]*x[k]+y[k]*y[k]<=1) { //ゴマが円に入った場合
    sum++; //円に入ったゴマの個数を1つ増やす
}
for (i=1; i<=k; i++) {
    if (x[i]*x[i]+y[i]*y[i]<=1) { //ゴマが円に入った場合
        px=(int)(300*x[i])+(getWidth()/2-360)+90+120;
        //倍精度型変数を整数型変数にキャストする
        py=(int)(300*y[i])+(getHeight()/2-600)+100;
        paint.setColor(Color.RED);
        //円の色を赤にする
        canvas.drawCircle(px, py, 1, paint); //円に入ったゴマを描く
        canvas.drawCircle(px, py, 2, paint); //円に入ったゴマを描く
    }
    else if (x[i]*x[i]+y[i]*y[i]>1) { //円に入らなかった場合
        px=(int)(300*x[i])+(getWidth()/2-360)+90+120;
        //倍精度型変数を整数型変数にキャストする
        py=(int)(300*y[i])+(getHeight()/2-600)+100;
        paint.setColor(Color.GREEN); //円の色を緑にする
        canvas.drawCircle(px, py, 1, paint); //円に入らなかったゴマを描く
        canvas.drawCircle(px, py, 2, paint); //円に入ったゴマを描く
    }
}
}

```

```

    pai=(double)4*sum/k; //πの近似値の計算
    paint.setColor(Color.BLUE);
    paint.setTextSize(40.0f);
    canvas.drawText("円周率π="+pai, (getWidth()/2-360)+80, (getHeight()/2-600)+510,
paint); //テキストと数値を絵として描く
    //paint.setTextSize(20.0f);
    paint.setColor(Color.BLACK);
    canvas.drawText("円に入ったゴマの個数="+sum+" 個", (getWidth()/2-360)+80,
(getHeight()/2-600)+570, paint); //テキストと数値を絵として描く
    //paint.setTextSize(20.0f);
    canvas.drawText("まいたゴマの総数="+k+" 個", (getWidth()/2-360)+80, (getHeight()
/2-600)+630, paint); //テキストと数値を絵として描く
    if (k<=N && flag==1) { //まく最大のゴマの個数以下で、かつゴマをまく識別子が1の場合
        invalidate(); //再描画する (clear & goto onDraw)
    }
    if (k==N) {
        flag=2; //ゴマをまく識別子が2 (初期化) の場合
    }
} //protected void onDraw(Canvas canvas)

```

//画面にタッチしたときのイベント処理-----

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    flag=flag+1; //flagに1を加える
    flag=flag % 3; //flagに1、2、0を代入する
    if (flag==0) { //ゴマをまく識別子が0 (停止) の場合
        sum=0; //円に入ったゴマの個数を0にする
        k=0; //まいたゴマの個数を0にする
    }
    invalidate(); //再描画する (clear & goto onDraw)
    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.mygoma2.MyGoma2
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

</androidx.constraintlayout.widget.ConstraintLayout>
```

[3]MainActivity.java

```
/*
```

```
-----
    ゴマをまいて円周率 $\pi$ を求める
    Android 4.1 (Jelly Bean)
    Copyright (C) K.Niwa 2021. 2. 7
    -----
```

```
*/
```

```
package jp.kiyo.wuena.mygoma2;

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);
    }
}
```