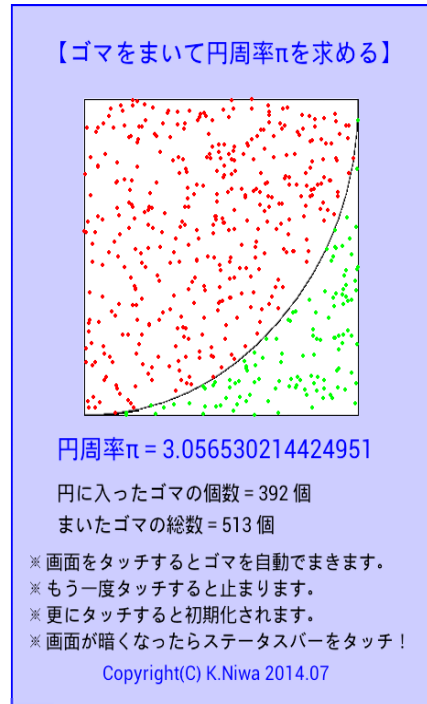


```
//-----
//
//      ゴマをまいて円周率 $\pi$ を求める
//      (モンテカル口法)
//      Ver10
//      Copyright(C) K.Niwa 2014.07.20
//
//-----
```



【エミュレータ画面例】



【スマートフォン画面例】
(Android 4.0)

【アプリの概要】

ゴマをまいて円周率 π の近似値を求めます。
 正方形とそれに内接する円を描いておいて、その上からゴマを無作為にまきます。
 正方形の中に入ったゴマの個数と円の中に入ったゴマの個数の割合から、円周率 π の近似値が求まります。
 その理由を数学的に考えてみましょう。

【 1 】 Goma. java

```
package jp.seitoku.goma; //パッケージ名の指定

import android.content.Context; //ライブラリを読み込む
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;

public class Goma extends View { //Viewクラスを継承したMyMonteクラス

    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=1999; //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[2001];
    double[] y=new double[2001];

    public Goma(Context context, AttributeSet attrs, int defStyle) { //コンストラクタ
        super(context, attrs, defStyle);
    }

    public Goma(Context context, AttributeSet attrs) { //コンストラクタ
        super(context, attrs);
    }
}
```

```

    }

    public Goma(Context context) { //コンストラクタ
        super(context);
    }

//onDrawメソッド-----
@Override
protected void onDraw(Canvas canvas) {

    super.onDraw(canvas);
    canvas.drawColor(Color.WHYTE);
    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.BLACK); //実
    canvas.drawRect((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()/2-240)+390, (getHeight()/2-343)+400, paint);
    paint.setColor(Color.WHYTE);
    canvas.drawRect((getWidth()/2-240)+91, (getHeight()/2-343)+101, (getWidth()/2-240)+389, (getHeight()/2-343)+399, paint);

    paint.setColor(Color.BLACK); //扇
    RectF rect1 = new RectF((getWidth()/2-240)+90-300, (getHeight()/2-343)+100-300, (getWidth()/2-240)+390, (getHeight()/2-343)+400);
    canvas.drawArc(rect1, 0f, 90f, true, paint);
    paint.setColor(Color.WHYTE);
    RectF rect2 = new RectF((getWidth()/2-240)+90-300, (getHeight()/2-343)+100-300, (getWidth()/2-240)+390-1, (getHeight()/2-343)+400-1);
    canvas.drawArc(rect2, 0f, 90f, true, paint);

    paint.setColor(Color.BLACK); //実
    canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()/2-240)+90, (getHeight()/2-343)+400, paint);
    canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()/2-240)+390, (getHeight()/2-343)+100, paint);

    //表題の表示
    paint.setColor(Color.BLUE);
    paint.setTextSize(25.0f);
    canvas.drawText("【ゴマをまいて円周率πを求める】", (getWidth()/2-240)+60-12, (getHeight()/2-343)+65, paint);

    paint.setColor(Color.BLACK); //説
    paint.setTextSize(19.0f);
    canvas.drawText("※ 画面をタッチするとゴマを自動でまきます。", (getWidth()/2-240)+30, (getHeight()/2-343)+545, paint);

```

```

        canvas.drawText("※ もう一度タッチすると止まります。", (getWidth()/2-240)+30, (getHeight()
/2-343)+570, paint);
        canvas.drawText("※ 更にタッチすると初期化されます。", (getWidth()/2-240)+30, (getHeight()
/2-343)+595, paint);
        canvas.drawText("※ 画面が暗くなったらステータスバーをタッチ!", (getWidth()/2-240)+30,
(getHeight()/2-343)+620, paint);

        if (k==0) {
            canvas.drawText("※ (円に入ったゴマの個数)を", (getWidth()/2-240)+100, (getHeight
()/2-343)+140, paint);
            canvas.drawText(" (まいたゴマの総数)で割った", (getWidth()/2-240)+100, (getHeight
()/2-343)+160, paint);
            canvas.drawText(" 値の4倍が円周率 $\pi$ になって", (getWidth()/2-240)+100, (getHeight
()/2-343)+180, paint);
            canvas.drawText(" いることを観察してみよう。", (getWidth()/2-240)+100, (getHeight
()/2-343)+200, paint);
        }

        paint.setColor(Color.BLUE);
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright (C) K.Niwa 2014.07", (getWidth()/2-240)+110, (getHeight()/2-34
3)+650, 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()/2-240)+90;
                    //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()/2-343)+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()/2-240)+90;
                    //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()/2-343)+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; // $\pi$ の近似値の計算
        paint.setColor(Color.BLUE);
        paint.setTextSize(25.0f);
        canvas.drawText("円周率 $\pi$ ="+pai, (getWidth()/2-240)+60, (getHeight()/2-343)+440,
        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()/2-240)+60, (getHeight()/2-343)+480, 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()/2-240)+60, (getHeight()/2-343)+510, 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] 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="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1" >
    <jp.seitoku.goma.Goma
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:id="@+id/myview1" >
    </jp.seitoku.goma.Goma>
</LinearLayout>

```

[3] GomaActivity.java

```

package jp.seitoku.goma;

import android.os.Bundle;
import android.app.Activity;

```

```
import android.view.Menu;

public class GomaActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
```