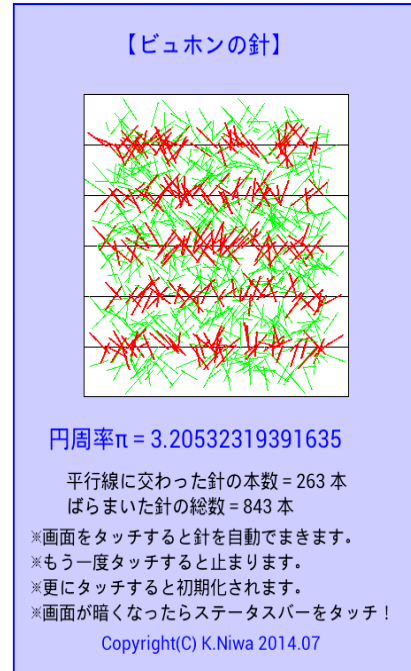


```
//-----
//
//          ビュホンの針
//          Ver8
//      Copyright(C) K.Niwa 2014.07.21
//
//-----
```



【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】

針をまいて円周率  $\pi$  の近似値を求めます。  
 平行線を描いておいて、その上から針を無作為にまきます。ただし、針の長さは平行線の間隔より小さいとします。  
 平行線に交わった針の本数の割合から、円周率  $\pi$  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】 Hari.java

```
package jp.seitoku.hari;          //パッケージ名指定

import android.content.Context; //ライブラリの読み込み
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;

public class Hari extends View { //Viewクラスを継承しMybuffonクラス
    //変数を初期化
    int flag=0;          //針の向き(1)、否(2)、初期化(0)。
    int N=1999;         //まいた針の最大数
    int i;              //for ループで使用
    int sum=0;          //平行線に交わった針の本数
    int px1;            //針の端x座標を初期化したもの
    int py1;            //針の端y座標を初期化したもの
    int px2;            //針の端x座標を初期化したもの
    int py2;            //針の端y座標を初期化したもの
    int yy;             //平行線作成で使用
    int k=0;            //まいた針の本数

    double[] x1=new double[2001]; //針の端x座標
    double[] y1=new double[2001]; //針の端y座標
    double[] x2=new double[2001]; //針の端x座標
    double[] y2=new double[2001]; //針の端y座標
    double pai;          //πの近似値

    public Hari(Context context, AttributeSet attrs, int defStyle) { //コンストラクタ
```

```

        super(context, attrs, defStyle);
    }

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

    public Hari(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.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,(getHei
ght()/2-343)+675-i,paint);

    canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+675-i,(getWidth()/2-240)+470-i,(getH
eight()/2-343)+675-i,paint);

    canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+675-i,(getWidth()/2-240)+470-i,(get
Height()/2-343)+10+i,paint);

    canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+10+i,(getWidth()/2-240)+10+i,(getHe
ight()/2-343)+10+i,paint);
    }

    paint.setColor(Color.BLACK);

//平線初描
    canvas.drawRect((getWidth()/2-240)+90,(getHeight()/2-343)+100,(getWidth()/2-240)+390,(getHeight
()/2-343)+400,paint);
    paint.setColor(Color.WHITE);

    canvas.drawRect((getWidth()/2-240)+91,(getHeight()/2-343)+101,(getWidth()/2-240)+389,(getHeight
()/2-343)+399,paint);

    paint.setColor(Color.BLACK);

//平線再描
    for (yy=150;yy<=350;yy=yy+50) {

    canvas.drawLine((getWidth()/2-240)+90,(getHeight()/2-343)+yy,(getWidth()/2-240)+390,(getHeight()
/2-343)+yy,paint);
    }

    paint.setColor(Color.BLUE);

//表題表示
    paint.setTextSize(25.0f);
    canvas.drawText("【ヒヨ本の針】", (getWidth()/2-240)+150-24,(getHeight()/2-343)+60,
paint);

```

```

        paint.setColor(Color.BLACK);
//表題表示
        paint.setTextSize(18.0f);
        //canvas.drawText(" 平行線に交った針は赤で表示...", (getWidth()/2-240)+80,
(getHeight()/2-343)+90, paint);

        paint.setColor(Color.BLACK);
//説明表示
        paint.setTextSize(19.0f);
        canvas.drawText(" ※画面をタッチすると針を自動まきます。", (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されます。", (getWidth()/2-240)+30,
(getHeight()/2-343)+595, paint);
        canvas.drawText(" ※画面が暗に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(" ※ 平行線に交った針は赤、", (getWidth()/2-240)+100,
(getHeight()/2-343)+120, paint);
            canvas.drawText(" 交らなかった針は緑で表示", (getWidth()/2-240)+100,
(getHeight()/2-343)+140, paint);
            canvas.drawText(" します。", (getWidth()/2-240)+100, (getHeight()/2-343)+160,
paint);
            canvas.drawText(" ※ 「ばらまいた針の総数を、", (getWidth()/2-240)+100,
(getHeight()/2-343)+220, paint);
            canvas.drawText(" 「 平行線に交った針の本数 ", (getWidth()/2-240)+100,
(getHeight()/2-343)+240, paint);
            canvas.drawText(" で割ると、円周率に近づいてい", (getWidth()/2-240)+100,
(getHeight()/2-343)+260, paint);
            canvas.drawText(" ることを観察してみましょう。", (getWidth()/2-240)+100,
(getHeight()/2-343)+280, paint);
            canvas.drawText(" ※ ただし、平行線の間隔は約", (getWidth()/2-240)+100,
(getHeight()/2-343)+370, paint);
            canvas.drawText(" 長さの2倍しています。", (getWidth()/2-240)+100,
(getHeight()/2-343)+390, paint);
        }

        paint.setColor(Color.BLUE);
//著作権表示
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright(C) K.Niwa 2014.07", (getWidth()/2-240)+110,
(getHeight()/2-343)+640+10, paint);

        k=k+1; //また針の本数を増やす
        x1[k]=115+250*Math.random();
//k番目の針の両端位置座標(x1,y1),(x2,y2)を乱数決める
        y1[k]=125+250*Math.random();
        x2[k]=x1[k]+25*Math.cos(2*Math.PI*Math.random());
        y2[k]=y1[k]+25*Math.sin(2*Math.PI*Math.random());

        if (y1[k]>y2[k]) {
//k番目の針が平行線交っているかの判断
            if (y1[k]>150 && y2[k]<150) {
                sum++;
            }
//交った針の本数をカウントする
        }
        else if (y1[k]>200 && y2[k]<200) {
            sum++;
        }
        else if (y1[k]>250 && y2[k]<250) {
            sum++;
        }
    }

```

```

else if (y1[k]>300 && y2[k]<300) {
    sum++;
}
else if (y1[k]>350 && y2[k]<350) {
    sum++;
}
else if (y1[k]==150 || y2[k]==150) {
    sum++;
}
else if (y1[k]==200 || y2[k]==200) {
    sum++;
}
else if (y1[k]==250 || y2[k]==250) {
    sum++;
}
else if (y1[k]==300 || y2[k]==300) {
    sum++;
}
else if (y1[k]==350 || y2[k]==350) {
    sum++;
}
}
else if (y1[k]<y2[k]) {
    if (y1[k]<150 && y2[k]>150) {
        sum++;
    }
    else if (y1[k]<200 && y2[k]>200) {
        sum++;
    }
    else if (y1[k]<250 && y2[k]>250) {
        sum++;
    }
    else if (y1[k]<300 && y2[k]>300) {
        sum++;
    }
    else if (y1[k]<350 && y2[k]>350) {
        sum++;
    }
    else if (y1[k]==150 || y2[k]==150) {
        sum++;
    }
    else if (y1[k]==200 || y2[k]==200) {
        sum++;
    }
    else if (y1[k]==250 || y2[k]==250) {
        sum++;
    }
    else if (y1[k]==300 || y2[k]==300) {
        sum++;
    }
    else if (y1[k]==350 || y2[k]==350) {
        sum++;
    }
}
else if (y1[k]==y2[k]) {
    if (y1[k]==150 || y1[k]==200 || y1[k]==250 || y1[k]==300 ||
y1[k]==350) {
        sum++;
    }
}
}

for (i=1;i<=k;i++) { //k本の柱
    px1=(int)(x1[i]); //金の端、座票整理キャストする
    py1=(int)(y1[i]); //金の端、座票整理キャストする
    px2=(int)(x2[i]); //金の端、座票整理キャストする
    py2=(int)(y2[i]); //金の端、座票整理キャストする
    paint.setColor(Color.RED); //金の色にする
}

```

```

        if (y1[i]>y2[i]) {
            if (y1[i]>150 && y2[i]<150) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]>200 && y2[i]<200) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]>250 && y2[i]<250) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]>300 && y2[i]<300) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]>350 && y2[i]<350) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]==150 || y2[i]==150) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]==200 || y2[i]==200) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]==250 || y2[i]==250) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
            }
            else if (y1[i]==300 || y2[i]==300) {

```

```

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]==350 || y2[i]==350) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else {
        paint.setColor(Color.GREEN); //銅色緑す

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint); //銅色
        paint.setColor(Color.RED);
//銅色赤す
    }
    else if (y1[i]<y2[i]) {
        if (y1[i]<150 && y2[i]>150) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]<200 && y2[i]>200) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]<250 && y2[i]>250) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]<300 && y2[i]>300) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]<350 && y2[i]>350) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

canvas.drawLine((getWidth()/2-240)+px1-1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2-1,(getHe
ight()/2-343)+py2, paint);
    }
    else if (y1[i]==150 || y2[i]==150) {

canvas.drawLine((getWidth()/2-240)+px1,(getHeight()/2-343)+py1,(getWidth()/2-240)+px2,(getHeight
()/2-343)+py2, paint);

```





```

        pai=(double)k/sum; //実線から計算倍率にする
    }
    else if (sum==0) { //0で割ったときの処理
        pai=0;
    }

    paint.setColor(Color.BLUE); //実線の結果表示
    paint.setTextSize(25.0f);
    canvas.drawText("円率="+pai, (getWidth()/2-240)+50, (getHeight()/2-343)+450,
paint);

    paint.setColor(Color.BLACK);
    paint.setTextSize(20.0f);
    canvas.drawText("平行線に交った金の本数="+sum+" 本", (getWidth()/2-240)+70,
(getHeight()/2-343)+490, paint);
    canvas.drawText("ばらまいた金の総数="+k+" 本", (getWidth()/2-240)+70,
(getHeight()/2-343)+515, paint);

    if (k<=N && flag==1) {
        invalidate(); //再描画clear & goto onDraw
    }

    if (k==N) { //金を最大枚数にしたとき
        flag=2; //金をまとのを止める
    }

} //protected void onDraw(Canvas canvas) {

//画面タッチしたときのイベント処理-----
@Override
public boolean onTouchEvent(MotionEvent event) {

    flag=flag+1; //金をまか(1)、呑(2)、初期化する(0)
    flag=flag % 3; //flag は1、2、0が来る

    if (flag==0) { //初期化する
        sum=0;
        k=0;
    }

    invalidate(); //再描画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.hari.Hari
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:id="@+id/myview1" >
    </jp.seitoku.hari.Hari>

```



</LinearLayout>

### [3] HariActivity.java

```
package jp.seitoku.hari;

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

public class HariActivity 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;
    }
}
```