[1]MyKinjiofp13Eng.java

```
/*
-------------------------------------
    無限級数によるπの近似１３(英語版)
        Android 4.1 (Jelly Bean)
     Copyright(C) K.Niwa 2021.9.17
-------------------------------------
*/

package jp.kiyo.wuena.mykinjiofp13eng;

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 MyKinjiofp13Eng extends View {

    private Bitmap bitmap1 = null;

    int flag=0;        //自動識別子
    double ct=0;   //項の数 int 型にするとうまく計算結果が得られない！
    int n=0;       //項数
    double t=1;        //πの近似値を求める過程で使用
    int i;        //ループカウンター
    double pai;        //πの近似値
    double s=1;        //π／８を求める過程での無限級数

    public MyKinjiofp13Eng(Context context) {
        super(context);
```

```
        init(context);
    }

    public MyKinjiofp13Eng(Context context, AttributeSet attrs) {
        super(context,attrs);
        init(context);
    }

    public MyKinjiofp13Eng(Context context, AttributeSet attrs,int defStyle) {
        super(context,attrs,defStyle);
        init(context);
    }

    private void init(Context context) {
        Resources res = context.getResources();     //画像用
        bitmap1 = BitmapFactory.decodeResource(res, R.drawable.gregory);    //画像用
    }

    @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)+15,(getHeight()/2-600)+10,(getWidth()/2-
360)+705,(getHeight()/2-600)+1190, paint);

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

```
        for (int i=0;i<3;i++) {
            canvas.drawLine((getWidth()/2-360)+15+i,(getHeight()/2-600)+10+i,(getWidth()/2-
360)+15+i,(getHeight()/2-600)+1190-i, paint);
            canvas.drawLine((getWidth()/2-360)+15+i,(getHeight()/2-600)+1190-i,(getWidth()/2-
360)+705-i,(getHeight()/2-600)+1190-i, paint);
            canvas.drawLine((getWidth()/2-360)+705-i,(getHeight()/2-600)+1190-i,(getWidth()/2-
360)+705-i,(getHeight()/2-600)+10+i, paint);
            canvas.drawLine((getWidth()/2-360)+705-i,(getHeight()/2-600)+10+i,(getWidth()/2-
360)+15+i,(getHeight()/2-600)+10+i, 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>
        }
        else {
            a=(float) 1.0;
            b=(float) 1.0;
        }

        Matrix Mat = new Matrix(); //---------------------------------- <画像を拡大・縮小す
る>
        Mat.postScale(a, b);   //---------------------------------------
        Bitmap bitmap2 = Bitmap.createBitmap(  //-----------------------
                bitmap1,0,0,    //---------------------------------------
                bitmap1.getWidth(),     //-----------------------------------
                bitmap1.getHeight(),    //-----------------------------------
                Mat,true    //---------------------------------------
        ); //-------------------------------------------------

        if (bitmap2 != null) {
            canvas.drawBitmap(bitmap2, (getWidth()/2-360)+240,(getHeight()/2-600)+150, paint);
        }
```

```
        paint.setTextSize(35.0f);
        canvas.drawText("【Approximation 13 of Pi by Infinite Series】", (getWidth()/2-
360)+15, (getHeight()/2-600)+80, paint);
        paint.setTextSize(35.0f);
        canvas.drawText(" (Find an approximation of pi) ", (getWidth()/2-360)+105,
(getHeight()/2-600)+130, paint);
        paint.setColor(Color.BLACK);
        paint.setTextSize(30.0f);
        canvas.drawText("＜Convergence is fast＞", (getWidth()/2-360)+165+100-10-65,
(getHeight()/2-600)+120+290+5, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0f);
        canvas.drawText("Copyright(C) Sohun 2021.9.17", (getWidth()/2-360)+150+10,
(getHeight()/2-600)+1130, paint);

//---------- 計算部始まり ----------

        ct++;
        n++;

        t=1;
        for (i=1;i<=ct;i++){
            t=(t/3);
        }

        if (ct % 2 == 1){
            s=s-(1/(2*ct+1))*t;
        }
        else {
            s=s+(1/(2*ct+1))*t;
        }
        pai=s*2*Math.sqrt(3);

//---------- 計算部終わり ----------
```

```
        paint.setColor(Color.BLACK);
        paint.setTextSize(40.0f);
        canvas.drawText("Number of terms = "+(n+1)+"" ,(getWidth()/2-360)+40-5,
(getHeight()/2-600)+510, paint);

        canvas.drawText("Approximation of pi", (getWidth()/2-360)+40-5, (getHeight()/2-
600)+590, paint);
        canvas.drawText("=2√3{1-(1/3)・(1/3^1)+(1/5)・(1/3^2)", (getWidth()/2-360)+50-15-5,
(getHeight()/2-600)+650-10, paint);
        canvas.drawLine((getWidth()/2-360)+111,(getHeight()/2-600)+352+267-10,(getWidth()/2-
360)+138,(getHeight()/2-600)+352+267-10,paint);
        canvas.drawLine((getWidth()/2-360)+111,(getHeight()/2-600)+351+267-10,(getWidth()/2-
360)+138,(getHeight()/2-600)+351+267-10,paint);
        canvas.drawLine((getWidth()/2-360)+111,(getHeight()/2-600)+351+267-10,(getWidth()/2-
360)+111-1,(getHeight()/2-600)+351+267+5-10,paint);
        canvas.drawLine((getWidth()/2-360)+111+1,(getHeight()/2-600)+351+267-10,(getWidth()/2-
360)+111-1+1,(getHeight()/2-600)+351+267+5-10,paint);
        canvas.drawLine((getWidth()/2-360)+111+2,(getHeight()/2-600)+351+267-10,(getWidth()/2-
360)+111-1+2,(getHeight()/2-600)+351+267+5-10,paint);

        canvas.drawText("-(1/7)・(1/3^3)+(1/9)・(1/3^4) ‥ }", (getWidth()/2-360)+50+40,
(getHeight()/2-600)+710-20, paint);

        paint.setColor(Color.BLUE);
        canvas.drawText("="+pai ,(getWidth()/2-360)+50-15-5, (getHeight()/2-600)+740, paint);

        paint.setColor(Color.BLACK);
        canvas.drawText("Pi π", (getWidth()/2-360)+40-5, (getHeight()/2-600)+840, paint);
        canvas.drawText("=3.14159265358979323‥", (getWidth()/2-360)+50-15-5, (getHeight()/2-
600)+890, paint);

        paint.setTextSize(30.0f);
        canvas.drawText("Touch the screen 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);

        if (flag==1) {    //flag=1 で自動になる     flag=2 で自動が止まる     flag=0 で初期化する
            invalidate();  //表示を更新する
        }

    }//protected void onDraw(Canvas canvas)

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        flag++;
        flag = flag % 3;
        if (flag==0) {
            ct=0;     //項数
            s=1;      //πを求める過程で使用
            n=0;      //項数
        }

        invalidate();  //表示を更新する
        return false;

    }//public boolean onTouchEvent(MotionEvent event)

}//public class MyPai13 extends View
```

[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.mykinjiofp13eng.MyKinjiofp13Eng
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3]MainActivity.java

```
/*
-------------------------------------
        無限級数によるπの近似１３(英語版)
            Android 4.1 (Jelly Bean)
          Copyright(C) K.Niwa 2021.2.4
-------------------------------------
*/

package jp.kiyo.wuena.mykinjiofp13eng;

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;
    }
}
```