

[1]MyGrerai2. java

/\*

-----  
グレゴリー・ライプニッツの公式  
Android 4.1 (Jelly Bean)  
Copyright (C) K. Niwa 2021. 1. 23  
-----

\*/

**package** jp.kiyo.wuena.mygrerai2;

**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** MyGrerai2 **extends** View {

**private** Bitmap **bitmap1** = **null**; //画像用

**int** **flag**=0; //自動識別子

**int** **ct**=0; //分子・分母の項の数

**int** **count**; //ループカウンター

**double** **pai**; //πの近似値

**double** **s**; //π/4を求める過程での無限級数

**public** MyGrerai2(Context context) {

**super**(context);

init(context);

}

```

public MyGreraI2(Context context, AttributeSet attrs) {
    super(context, attrs);
    init(context);
}

public MyGreraI2(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)+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<3; 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);
    }

    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)+245, (getHeight()/2-600)+150, paint);
//画像用
    }

```

```

        paint.setTextSize(35.0f);
        canvas.drawText("【グレゴリー・ライプニッツの公式】", (getWidth()/2-360)+80,
(getHeight()/2-600)+80, paint);
        paint.setTextSize(35.0f);
        canvas.drawText(" (πの近似値を求める) ", (getWidth()/2-360)+185, (getHeight()/2-
600)+130, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0f);
        canvas.drawText("Copyright(C) K.Niwa 2021.1.23", (getWidth()/2-360)+150,
(getHeight()/2-600)+1130, paint);

//----- 計算部始まり -----

        ct++;
        if (ct % 2 == 1) {
            s = s + (double) 1 / (2 * ct - 1);
        } else if (ct % 2 == 0) {
            s = s - (double) 1 / (2 * ct - 1);
        }
        }

        pai = (double) 4 * s;

//----- 計算部終わり -----

        paint.setColor(Color.BLACK);
        paint.setTextSize(40.0f);
        canvas.drawText("項数 "+ct+" のとき", (getWidth()/2-360)+40, (getHeight()/2-600)+510,
paint);

        canvas.drawText("円周率πの近似値", (getWidth()/2-360)+40, (getHeight()/2-600)+590,
paint);
        canvas.drawText("= 4 (1-1/3+1/5-1/7+1/9 ...)", (getWidth()/2-360)+100,
(getHeight()/2-600)+640, paint);

        paint.setColor(Color.BLUE);

```

```

canvas.drawText("= "+pai , (getWidth()/2-360)+100, (getHeight()/2-600)+690, paint);

paint.setColor(Color.BLACK);
canvas.drawText("円周率π", (getWidth()/2-360)+40, (getHeight()/2-600)+790, paint);
canvas.drawText("= 3.1415926535897932...", (getWidth()/2-360)+100, (getHeight()/2-
600)+840, paint);

//paint.setColor(Color.BLACK);
paint.setTextSize(30.0f);
canvas.drawText("※ 画面をタッチすると自動になります。", (getWidth()/2-360)+50,
(getHeight()/2-600)+950, paint);
canvas.drawText("※ 画面をタッチすると自動が止まります。", (getWidth()/2-360)+50,
(getHeight()/2-600)+990, paint);
canvas.drawText("※ 更に画面をタッチすると初期化されます。", (getWidth()/2-360)+50,
(getHeight()/2-600)+1030, 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)+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; //分子・分母の項の数
        s=0; //πを求める過程で使用
    }

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

```

```
        } //public boolean onTouchEvent(MotionEvent event)

    } //public class MyGregory 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.mygrerai2.MyGrerai2
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>
```

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

[3]MainActivity.java

```
/*
```

---

グレゴリー・ライプニッツの公式

Android 4.1 (Jelly Bean)  
Copyright (C) K. Niwa 2021. 1. 23

---

\*/

```
package jp.kiyo.wuena.mygrerai2;

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;
    }
}
```