[1]MyKinjiofe2Eng.java

```
/*
-------------------------------------
        自然対数の底ｅの近似２(英語版)
          Android 4.1 (Jelly Bean)
        Copyright(C) K.Niwa 2021.9.5
-------------------------------------
*/

package jp.kiyo.wuena.mykinjiofe2eng;

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 MyKinjiofe2Eng extends View {

    private Bitmap bitmap1 = null;

    int flag=0;       //自動識別子
    int ct=0;     //項数
    int count;    //ループカウンター
    int n;
    double e=0;       //ｅの近似値
    double s=1;       //ｅの近似値を求める過程で〝使用
    double t=1;       //ｅの近似値を求める過程で〝使用

    public MyKinjiofe2Eng(Context context) {
        super(context);
```

```
        init(context);
    }

    public MyKinjiofe2Eng(Context context, AttributeSet attrs) {
        super(context,attrs);
        init(context);
    }

    public MyKinjiofe2Eng(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.euler); //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)+10,(getHeight()/2-600)+10,(getWidth()/2-
360)+710,(getHeight()/2-600)+1190, paint);

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

```
        for (int i=0;i<3;i++) {
            canvas.drawLine((getWidth()/2-360)+10+i,(getHeight()/2-600)+10+i,(getWidth()/2-
360)+10+i,(getHeight()/2-600)+1190-i, paint);
            canvas.drawLine((getWidth()/2-360)+10+i,(getHeight()/2-600)+1190-i,(getWidth()/2-
360)+710-i,(getHeight()/2-600)+1190-i, paint);
            canvas.drawLine((getWidth()/2-360)+710-i,(getHeight()/2-600)+1190-i,(getWidth()/2-
360)+710-i,(getHeight()/2-600)+10+i, paint);
            canvas.drawLine((getWidth()/2-360)+710-i,(getHeight()/2-600)+10+i,(getWidth()/2-
360)+10+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)+250,(getHeight()/2-600)+150, paint);
        }
```

```
        paint.setTextSize(40.0f);
        canvas.drawText("【Approximation 2 of", (getWidth()/2-360)+75, (getHeight()/2-600)+80,
paint);
        canvas.drawText("       Napier`s Constant】", (getWidth()/2-360)+205, (getHeight()/2-
600)+120, paint);
        paint.setTextSize(35.0f);
        //canvas.drawText("（1／e＝"+1/Math.E+"・・・）",(getWidth()/2-360)+130-70,
(getHeight()/2-600)+125, paint);
        paint.setColor(Color.BLACK);
        paint.setTextSize(30.0f);
        canvas.drawText("＜Covergence is fast＞", (getWidth()/2-360)+255-50, (getHeight()/2-
600)+415, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0f);
        canvas.drawText("Copyright(C) Sohun 2021.9.5", (getWidth()/2-360)+165, (getHeight()/2-
600)+1130, paint);

//---------- 計算部始まり ----------

        ct=ct+1;
        double s=1;

        for (n=1;n<=ct;n++) {
            s=s*(double)1/n;
        }

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

        e=t;
```

```
//---------- 計算部終わり ----------

        paint.setColor(Color.BLACK);
        paint.setTextSize(40.0f);
        canvas.drawText("Number of terms = "+(ct+1) +"",(getWidth()/2-360)+40, (getHeight()/2-
600)+510+20, paint);
        canvas.drawText("Approximation of ", (getWidth()/2-360)+40, (getHeight()/2-
600)+580+20, paint);
        canvas.drawText("the reciprocal of Napier`s constant", (getWidth()/2-360)+40,
(getHeight()/2-600)+620+20, paint);
        canvas.drawText("＝1-1/1!+1/2!-1/3!+1/4!-･･･", (getWidth()/2-360)+100-10,
(getHeight()/2-600)+660+20, paint);
        paint.setColor(Color.BLUE);
        canvas.drawText("＝"+e ,(getWidth()/2-360)+100-10, (getHeight()/2-600)+700+20, paint);
        paint.setColor(Color.BLACK);

        canvas.drawText("The reciprocal of Napier`s constant", (getWidth()/2-360)+40,
(getHeight()/2-600)+770+20, paint);
        //canvas.drawText("＝1-1/1!+1/2!-1/3!+1/4!-･･･", (getWidth()/2-360)+100,
(getHeight()/2-600)+650-10, paint);
        canvas.drawText("＝"+1/Math.E+"･･･" ,(getWidth()/2-360)+100-10, (getHeight()/2-
600)+810+20, 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 && ct<16) {     //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
        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) {
            e=0;
            ct=0;     //項数
            s=1;      //ｅを求める過程で使用
            t=1;      //ｅ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 MyOirer1 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.mykinjiofe2eng.MyKinjiofe2Eng
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3]MainActivity.java

```
/*
---------------------------------------
        自然対数の底ｅの近似２(英語版)
          Android 4.1 (Jelly Bean)
        Copyright(C) K.Niwa 2021.9.5
---------------------------------------
*/

package jp.kiyo.wuena.mykinjiofe2eng;

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;
    }
}
```