[1]MyNumberofeEng.java

```java
/*
------------------------------------
            オイラー数e(英語版)
          Android 4.1 (Jelly Bean)
      Copyright(C) K.Niwa 2021.9.1
------------------------------------
*/

package jp.kiyo.wuena.mynumberofeeng;

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 MyNumberofeEng extends View {

    private Bitmap bitmap1 = null; //画像用

    int flag=0;     //自動識別子
    int count; //ループカウンター
    long n;        //e=(1+1/n)^n
    double tte;     //eの近似値
    double s;  //e=(1+1/n)^nを求めるのに使用
    double k;  //e=(1+1/n)^nを求めるのに使用

    public MyNumberofeEng(Context context) {
        super(context);
        init(context);
    }

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

    public MyNumberofeEng(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)+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+10,(getHeight()/2-
600)+150,paint);     //画像用
        }
        paint.setAlpha(10000);
        paint.setColor(Color.BLUE);
        paint.setTextSize(45.0f);
        canvas.drawText("【Napier`s Constant】", (getWidth()/2-360)+190-60, (getHeight()/2-600)+80, paint);
        paint.setTextSize(35.0f);
        canvas.drawText("（the base of natural logarithm ｅ）", (getWidth()/2-360)+50+20, (getHeight()/2-600)+130, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0f);
        canvas.drawText("Copyright(C) Sohun 2021.9.1",(getWidth()/2-360)+150, (getHeight()/2-600)+1130, paint);

        n++;
        s=1;

        for (count=1;count<=n;count++) {
            k=(double)1+(double)1/(double)n;
            s=s*k;
        }
        tte=s;

        paint.setColor(Color.BLACK);

        paint.setTextSize(40.0f);
        canvas.drawText("ｎ＝ "+n+"", (getWidth()/2-360)+40, (getHeight()/2-600)+510-30, paint);
        canvas.drawText("Approximaion of Napier`s constant", (getWidth()/2-360)+40, (getHeight()/2-600)+590-10-30, paint);
        canvas.drawText("＝ｌｉｍ（１＋１／ｎ）", (getWidth()/2-360)+130, (getHeight()/2-600)+650-10-30, paint);
        canvas.drawText("n", (getWidth()/2-360)+405+170-30, (getHeight()/2-600)+710-80-10-30, paint);
        canvas.drawText("ｎ→∞ ", (getWidth()/2-360)+160+25-10, (getHeight()/2-600)+690-10-30, paint);
        paint.setColor(Color.BLUE);
        canvas.drawText("= "+tte, (getWidth()/2-360)+130, (getHeight()/2-600)+750-20-30,
paint);

        paint.setColor(Color.BLACK);
        canvas.drawText("Napier`s constant", (getWidth()/2-360)+40, (getHeight()/2-600)+830-
30, paint);
        canvas.drawText("= 2.7182818284590452・・・", (getWidth()/2-360)+130, (getHeight()/2-
600)+880-30, 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) {
            invalidate();
        }

    }//protected void onDraw(Canvas canvas)


    @Override
    public boolean onTouchEvent(MotionEvent event) {
        flag++;
        flag = flag % 3;
        if (flag==0) {
            n=0;       //e=(1+1/n)^n
            tte=0; //eの近似値
            s=1;   //e=(1+1/n)^nを求めるのに使用
            k=0;   //e=(1+1/n)^nを求めるのに使用
        }

        invalidate();
        return false;
    }
}//public class MyE 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.mynumberofeeng.MyNumberofeEng
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3]MainActivity.java

```java
/*
----------------------------------------
            オイラー数 e（英語版）
         Android 4.1 (Jelly Bean)
     Copyright(C) K.Niwa 2021.9.1
----------------------------------------
*/

package jp.kiyo.wuena.mynumberofeeng;

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;
    }
}
```