

```
[1]MyKinjiofp6Eng. java
```

```
/*
```

```
-----  
無限級数による $\pi$ の近似6 (英語版)  
Android 4.1 (Jelly Bean)  
Copyright (C) K. Niwa 2021. 9. 10  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mykinjiofp6eng;
```

```
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 MyKinjiofp6Eng extends View {
```

```
    private Bitmap bitmap1 = null;
```

```
    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
```

```
    double pai; // $\pi$ の近似値
```

```
    double s; // $\pi/8$ を求める過程での無限級数
```

```
    //*****
```

```
    //double rx; //x座標の画面の幅に対する比率
```

```
    //double ry; //y座標の画面の高さに対する比率
```

```
    //int x1, y1, x2, y2; //キャスト後のx y座標に使用
```

```

//*****

public MyKinjiopf6Eng(Context context) {
    super(context);
    init(context);
}

public MyKinjiopf6Eng(Context context, AttributeSet attrs) {
    super(context, attrs);
    init(context);
}

public MyKinjiopf6Eng(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);

//*****

```

```

//rx=super.getWidth()/480; //x座標の画面の幅に対する比率
//ry=super.getHeight()/690; //y座標の画面の高さに対する比率
//x1=(int) ((int)10*rx);
//y1=(int) ((int)10*ry);
//x2=(int) ((int)470*rx);
//y2=(int) ((int)675*ry);

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);
}

paint.setTextSize(35.0f);

//x1=(int) ((int)100*rx);
//y1=(int) ((int)120*ry);
canvas.drawText(" Approximation 6 of Pi by Infinite Series ", (getWidth()/2-360)+25,
(getHeight()/2-600)+80, paint);

paint.setTextSize(35.0f);

//x1=(int) ((int)110*rx);
//y1=(int) ((int)150*ry);
canvas.drawText(" (Find an approximation of pi) ", (getWidth()/2-360)+105,
(getHeight()/2-600)+130, paint);

paint.setColor(Color.BLUE);
paint.setTextSize(30.0f);

//x1=(int) ((int)120*rx);
//y1=(int) ((int)600*ry);
canvas.drawText(" Copyright(C) Sohun 2021. 9. 10", (getWidth()/2-360)+150,
(getHeight()/2-600)+1130, paint);
```

```
//----- 計算部始まり -----
```

```
ct++;  
s = s + (double) 1 / ((4 * ct - 3) * (4 * ct - 1));  
  
pai = (double) 8 * s;
```

```
//----- 計算部終わり -----
```

```
paint.setColor(Color.BLACK);  
paint.setTextSize(40.0f);  
  
//x1=(int) ((int)40*rx);  
//y1=(int) ((int)250*ry);  
canvas.drawText("Number of terms = "+ct+" ", (getWidth()/2-360)+40, (getHeight()/2-  
600)+510, paint);  
//canvas.drawText("x 率=" +getWidth() , x1, y1, paint);  
//canvas.drawText("y 率=" +getHeight() , x1, y1, paint);  
  
//x1=(int) ((int)40*rx);  
//y1=(int) ((int)300*ry);  
canvas.drawText("Approximation of pi", (getWidth()/2-360)+40, (getHeight()/2-600)+590,  
paint);  
  
//x1=(int) ((int)60*rx);  
//y1=(int) ((int)340*ry);  
canvas.drawText("=8{1/(1·3)+1/(5·7)+1/(9·11)+···}", (getWidth()/2-360)+60,  
(getHeight()/2-600)+640, paint);  
  
paint.setColor(Color.BLUE);  
  
//x1=(int) ((int)60*rx);  
//y1=(int) ((int)380*ry);  
canvas.drawText("="+pai , (getWidth()/2-360)+60, (getHeight()/2-600)+690, paint);
```

```

paint.setColor(Color.BLACK);
canvas.drawText("Pi π", (getWidth()/2-360)+40, (getHeight()/2-600)+790, paint);
canvas.drawText("=3.1415926535897932...", (getWidth()/2-360)+60, (getHeight()/2-
600)+840, paint);

paint.setTextSize(30.0f);

//x1=(int) ((int)50*rx);
//y1=(int) ((int)470*ry);
canvas.drawText("Touch the screen to activate.", (getWidth()/2-360)+50,
(getHeight()/2-600)+950, paint);

//x1=(int) ((int)50*rx);
//y1=(int) ((int)500*ry);
canvas.drawText("Touch the screen again to stop the auto.", (getWidth()/2-360)+50,
(getHeight()/2-600)+990, paint);

//x1=(int) ((int)50*rx);
//y1=(int) ((int)530*ry);
canvas.drawText("If you touch it further, it will be initialized.", (getWidth()/2-
360)+50, (getHeight()/2-600)+1030, paint);

//x1=(int) ((int)50*rx);
//y1=(int) ((int)560*ry);
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=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 MyPai6 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.mykinjiofp6eng.MyKinjiofp6Eng
    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6 (英語版)
```

```
Android 4.1 (Jelly Bean)
```

```
Copyright (C) K. Niwa 2021. 9. 10  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mykinjiofp6eng;
```

```
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;  
  }  
}
```