

```
[1]MyWourisuEng.java
```

```
/*
```

```
-----  
    ウォーリスの公式(英語版)  
    Android 4.1 (Jelly Bean)  
    Copyright (C) K.Niwa 2021.9.7  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mywourisueng;
```

```
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.*;
```

```
//import jp.kiyo.wuena.mywourisu2.MainActivity;
```

```
public class MyWourisuEng 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 sa=1; //πを求める過程で使用
```

```
    public MyWourisuEng(Context context) {
```

```
        super(context);
```

```
        init(context);
```

```
    }
```

```

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

public MyWourisuEng(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.wallis); //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)+250+10, (getHeight()/2-600)+150,
paint);
    }

```

```

    paint.setAlpha(10000);
    paint.setColor(Color.BLUE);
    paint.setTextSize(45.0f);
    canvas.drawText("【Wallis`s Formula】", (getWidth()/2-360)+120-30+10+95-50+10,
(getHeight()/2-600)+80, paint);
    paint.setTextSize(35.0f);
    canvas.drawText("(Find an approximation of pi)", (getWidth()/2-360)+185-70,
(getHeight()/2-600)+130, paint);

    paint.setColor(Color.BLUE);
    paint.setTextSize(30.0f);
    canvas.drawText("Copyright(C) Sohun 2021.9.7", (getWidth()/2-360)+150, (getHeight()/2-
600)+1130, paint);

    ct++;
    sa = sa * (((double) 2 * ct * 2 * ct) / (((double) (2 * ct - 1) * (2 * ct + 1))));
    pai = (double) 2 * sa;

    paint.setColor(Color.BLACK);
    paint.setTextSize(35.0f);
    canvas.drawText("The number of each terms of"+"", (getWidth()/2-360)+60-20,
(getHeight()/2-600)+310+200-30, paint);
    canvas.drawText("the numerator and denominator = "+2*ct+"", (getWidth()/2-360)+60-20,
(getHeight()/2-600)+310+200+35-30, paint);

    canvas.drawText("Approximation of pi", (getWidth()/2-360)+60-20, (getHeight()/2-
600)+590, paint);
    canvas.drawText("    2 · 2 · 4 · 4 · 6 · 6 · 8 · 8 ···", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+650-10, paint);
    canvas.drawText("= 2", (getWidth()/2-360)+100-30, (getHeight()/2-600)+670-10, paint);
    canvas.drawLine((getWidth()/2-360)+150, (getHeight()/2-600)+350+50+200+60-5-
10, (getWidth()/2-360)+380+50+230, (getHeight()/2-600)+655-10, paint);
    canvas.drawLine((getWidth()/2-360)+150, (getHeight()/2-600)+350+50+200+60-4-
10, (getWidth()/2-360)+380+50+230, (getHeight()/2-600)+656-10, paint);
    canvas.drawText("    1 · 3 · 3 · 5 · 5 · 7 · 7 · 9 ···", (getWidth()/2-360)+100-30,
(getHeight()/2-600)+690-10, paint);

```

```

paint.setColor(Color.BLUE);
canvas.drawText("=π", (getWidth()/2-360)+100-30, (getHeight()/2-600)+730, paint);

paint.setColor(Color.BLACK);
canvas.drawText("Pi π", (getWidth()/2-360)+60-20, (getHeight()/2-600)+830, paint);
canvas.drawText("=3.1415926535897932...", (getWidth()/2-360)+100-30, (getHeight()/2-
600)+880, 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(); ////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
        sa=1; //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 MyWourisu 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.mywourisueng.MyWourisuEng
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3] MainActivity.java

```
/*
```

ウォーリスの公式(英語版)
Android 4.1 (Jelly Bean)
Copyright (C) K. Niwa 2021. 9. 7

*/

```
package jp.kiyo.wuena.mywourisueng;

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;
    }
}
```