

```
[1]MySaicoroEng.java
```

```
/*
```

```
-----  
    積が奇数の2個のさいころ(英語版)  
    Android 4.1 (Jelly Bean)  
    Copyright (C) K.Niwa 2021. 9. 27  
-----
```

```
*/
```

```
package jp.kiyo.wuena.mysaicoroeng;
```

```
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 MySaicoroEng extends View {
```

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

```
    private Bitmap bitmap2 = null;
```

```
    private Bitmap bitmap3 = null;
```

```
    private Bitmap bitmap4 = null;
```

```
    private Bitmap bitmap5 = null;
```

```
    private Bitmap bitmap6 = null;
```

```
    int flag=0; //自動識別子
```

```
    int syoki=0; //初期化識別子
```

```
    int r1,r2; //さいころ1、さいころ2の目の識別子(乱数)
```

```
    int ct1=0; //実験回数
```

```
    int d11=0, d21=0, d31=0, d41=0, d51=0, d61=0; //度数 d31とはさいころ1の目が3で、さいころ
```

2の目が1の度数

```
int d12=0, d22=0, d32=0, d42=0, d52=0, d62=0;
int d13=0, d23=0, d33=0, d43=0, d53=0, d63=0;
int d14=0, d24=0, d34=0, d44=0, d54=0, d64=0;
int d15=0, d25=0, d35=0, d45=0, d55=0, d65=0;
int d16=0, d26=0, d36=0, d46=0, d56=0, d66=0;

int d=0; //積が奇数の度数
float ritu; //積が奇数の割合
int yy, xx; //枠に使用したループカウンター

int width;
int height;

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

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

public MySaicoroEng(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    init(context);
}

private void init(Context context) {
    Resources res = context.getResources();
    bitmap1 = BitmapFactory.decodeResource(res, R.drawable.sai1);
    bitmap2 = BitmapFactory.decodeResource(res, R.drawable.sai2);
    bitmap3 = BitmapFactory.decodeResource(res, R.drawable.sai3);
    bitmap4 = BitmapFactory.decodeResource(res, R.drawable.sai4);
    bitmap5 = BitmapFactory.decodeResource(res, R.drawable.sai5);
    bitmap6 = BitmapFactory.decodeResource(res, R.drawable.sai6);
}
```

```

//WindowManager wm = (WindowManager)context.getSystemService(Context.WINDOW_SERVICE);
//Display disp = wm.getDefaultDisplay();
//width = disp.getWidth();
//height = disp.getHeight();
}

@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<2;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);
    }
}

```

```

    paint.setColor(Color.BLUE);
    paint.setTextSize(40.0f);
    canvas.drawText("【Two Dice with an Odd Product】", (getWidth()/2-360)+60-10,
(getHeight()/2-600)+80, paint);

    paint.setColor(Color.BLACK);
    paint.setTextSize(30.0f);
    canvas.drawText("Dice I", (getWidth()/2-360)+135+10+110+10, (getHeight()/2-
600)+188+30, paint);
    canvas.drawText("Dice II", (getWidth()/2-360)+255+10+110+10, (getHeight()/2-
600)+188+30, 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(); //-----***
    Mat.postScale(a, b); //-----***
    Bitmap bitmap11 = Bitmap.createBitmap( //-----***
        bitmap1, 0, 0, //-----***
        bitmap1.getWidth(), //-----***
        bitmap1.getHeight(), //-----***
        Mat, true //-----***
    ); //-----***

    Bitmap bitmap22 = Bitmap.createBitmap( //-----***
        bitmap2, 0, 0, //-----***
        bitmap2.getWidth(), //-----***
        bitmap2.getHeight(), //-----***

```

```
        Mat, true //-----***
); //-----***
```

```
Bitmap bitmap33 = Bitmap.createBitmap( //-----***
    bitmap3, 0, 0, //-----***
    bitmap3.getWidth(), //-----***
    bitmap3.getHeight(), //-----***
    Mat, true //-----***
); //-----***
```

```
Bitmap bitmap44 = Bitmap.createBitmap( //-----***
    bitmap4, 0, 0, //-----***
    bitmap4.getWidth(), //-----***
    bitmap4.getHeight(), //-----***
    Mat, true //-----***
); //-----***
```

```
Bitmap bitmap55 = Bitmap.createBitmap( //-----***
    bitmap5, 0, 0, //-----***
    bitmap5.getWidth(), //-----***
    bitmap5.getHeight(), //-----***
    Mat, true //-----***
); //-----***
```

```
Bitmap bitmap66 = Bitmap.createBitmap( //-----***
    bitmap6, 0, 0, //-----***
    bitmap6.getWidth(), //-----***
    bitmap6.getHeight(), //-----***
    Mat, true //-----***
); //-----***
```

```
if (bitmap11 != null && bitmap22 != null && bitmap33 != null && bitmap44 != null &&
    bitmap55 != null && bitmap66 != null) {
```

```
    ct1++;
```

```

    r1=(int) (1+6*Math.random());
    if (r1==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r1==2) {
        canvas.drawBitmap(bitmap22, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r1==3) {
        canvas.drawBitmap(bitmap33, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r1==4) {
        canvas.drawBitmap(bitmap44, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r1==5) {
        canvas.drawBitmap(bitmap55, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }
    else if (r1==6) {
        canvas.drawBitmap(bitmap66, (getWidth()/2-360)+160+120-10, (getHeight()/2-
600)+130-5, paint);
    }

    r2=(int) (1+6*Math.random());
    if (r2==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
    }

```

```

        else if (r2==2) {
            canvas.drawBitmap(bitmap22, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
        }
        else if (r2==3) {
            canvas.drawBitmap(bitmap33, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
        }
        else if (r2==4) {
            canvas.drawBitmap(bitmap44, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
        }
        else if (r2==5) {
            canvas.drawBitmap(bitmap55, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
        }
        else if (r2==6) {
            canvas.drawBitmap(bitmap66, (getWidth()/2-360)+280+120-10, (getHeight()/2-
600)+130-5, paint);
        }
    } //if (bitmap1 != null && ...

    if (r1==1 && r2==1) {
        d11++;
    }
    else if (r1==2 && r2==1) {
        d21++;
    }
    else if (r1==3 && r2==1) {
        d31++;
    }
}

```

```
else if (r1==4 && r2==1) {
    d41++;
}
else if (r1==5 && r2==1) {
    d51++;
}
else if (r1==6 && r2==1) {
    d61++;
}
else if (r1==1 && r2==2) {
    d12++;
}
else if (r1==2 && r2==2) {
    d22++;
}
else if (r1==3 && r2==2) {
    d32++;
}
else if (r1==4 && r2==2) {
    d42++;
}
else if (r1==5 && r2==2) {
    d52++;
}
else if (r1==6 && r2==2) {
    d62++;
}
else if (r1==1 && r2==3) {
    d13++;
}
else if (r1==2 && r2==3) {
    d23++;
}
else if (r1==3 && r2==3) {
    d33++;
}
```



```
else if (r1==4 && r2==3) {
    d43++;
}
else if (r1==5 && r2==3) {
    d53++;
}
else if (r1==6 && r2==3) {
    d63++;
}
else if (r1==1 && r2==4) {
    d14++;
}
else if (r1==2 && r2==4) {
    d24++;
}
else if (r1==3 && r2==4) {
    d34++;
}
else if (r1==4 && r2==4) {
    d44++;
}
else if (r1==5 && r2==4) {
    d54++;
}
else if (r1==6 && r2==4) {
    d64++;
}
else if (r1==1 && r2==5) {
    d15++;
}
else if (r1==2 && r2==5) {
    d25++;
}
else if (r1==3 && r2==5) {
    d35++;
}
```

```

else if (r1==4 && r2==5) {
    d45++;
}
else if (r1==5 && r2==5) {
    d55++;
}
else if (r1==6 && r2==5) {
    d65++;
}
else if (r1==1 && r2==6) {
    d16++;
}
else if (r1==2 && r2==6) {
    d26++;
}
else if (r1==3 && r2==6) {
    d36++;
}
else if (r1==4 && r2==6) {
    d46++;
}
else if (r1==5 && r2==6) {
    d56++;
}
else if (r1==6 && r2==6) {
    d66++;
}

d=d11+d13+d15+d31+d33+d35+d51+d53+d55; //積が奇数の度数
paint.setColor(Color.BLACK);
paint.setTextSize(40.0f);
canvas.drawText("Number of times", (getWidth()/2-360)+100-20-10, (getHeight()/2-
600)+640, paint);
canvas.drawText("the product is odd = "+d, (getWidth()/2-360)+100-20+130-20-20+8-10,
(getHeight()/2-600)+675, paint);

```

```

    if (ct1 != 0) {
        ritu = (float)d/(float)ct1;    //積が奇数の割合
        paint.setColor(Color.BLUE);
        paint.setTextSize(40.0f);
        canvas.drawText("Percentage", (getWidth()/2-360)+100-20-10, (getHeight()/2-
600)+735, paint);
        canvas.drawText("of odd products = "+ritu, (getWidth()/2-360)+100-20+30-10,
(getHeight()/2-600)+735+35, paint);
    }

    paint.setColor(Color.BLACK);
    paint.setTextSize(40.0f);
    canvas.drawText("Number of experiments = "+ct1, (getWidth()/2-360)+100-20-10,
(getHeight()/2-600)+580, paint);

    //表の枠
    paint.setColor(Color.BLACK);
    canvas.drawLine((getWidth()/2-360)+120, (getHeight()/2-600)+260, (getWidth()/2-
360)+580, (getHeight()/2-600)+260, paint);
    canvas.drawLine((getWidth()/2-360)+120, (getHeight()/2-600)+290, (getWidth()/2-
360)+210, (getHeight()/2-600)+290, paint);
    canvas.drawLine((getWidth()/2-360)+210, (getHeight()/2-600)+290, (getWidth()/2-
360)+220, (getHeight()/2-600)+320, paint);

    //paint.setColor(Color.BLACK);
    for (yy=320;yy<=500;yy=yy+30) {
        canvas.drawLine((getWidth()/2-360)+120, (getHeight()/2-600)+yy, (getWidth()/2-
360)+580, (getHeight()/2-600)+yy, paint);
    }

    //paint.setColor(Color.RED);
    for (xx=220;xx<=580;xx=xx+60) {
        canvas.drawLine((getWidth()/2-360)+xx, (getHeight()/2-600)+260, (getWidth()/2-
360)+xx, (getHeight()/2-600)+500, paint);
    }
    canvas.drawLine((getWidth()/2-360)+120, (getHeight()/2-600)+260, (getWidth()/2-

```

```
360)+120, (getHeight()/2-600)+500, paint);
```

```
//表の文字
```

```
paint.setTextSize(19.0F);
```

```
canvas.drawText("Dice II", (getWidth()/2-360)+120+5+20, (getHeight()/2-600)+260+20, paint);
```

```
paint.setTextSize(25.0F);
```

```
canvas.drawText("1", (getWidth()/2-360)+240, (getHeight()/2-600)+280+20, paint);
```

```
canvas.drawText("2", (getWidth()/2-360)+300, (getHeight()/2-600)+280+20, paint);
```

```
canvas.drawText("3", (getWidth()/2-360)+360, (getHeight()/2-600)+300, paint);
```

```
canvas.drawText("4", (getWidth()/2-360)+420, (getHeight()/2-600)+300, paint);
```

```
canvas.drawText("5", (getWidth()/2-360)+480, (getHeight()/2-600)+300, paint);
```

```
canvas.drawText("6", (getWidth()/2-360)+540, (getHeight()/2-600)+300, paint);
```

```
paint.setTextSize(18.0F);
```

```
canvas.drawText("Dice I", (getWidth()/2-360)+125+20, (getHeight()/2-600)+290+25, paint);
```

```
paint.setTextSize(25.0F);
```

```
canvas.drawText("1", (getWidth()/2-360)+125+35, (getHeight()/2-600)+305+40, paint);
```

```
canvas.drawText("2", (getWidth()/2-360)+125+35, (getHeight()/2-600)+335+40, paint);
```

```
canvas.drawText("3", (getWidth()/2-360)+125+35, (getHeight()/2-600)+365+40, paint);
```

```
canvas.drawText("4", (getWidth()/2-360)+125+35, (getHeight()/2-600)+395+40, paint);
```

```
canvas.drawText("5", (getWidth()/2-360)+125+35, (getHeight()/2-600)+425+40, paint);
```

```
canvas.drawText("6", (getWidth()/2-360)+125+35, (getHeight()/2-600)+455+40, paint);
```

```
//表の度数
```

```
paint.setTextSize(30.0F);
```

```
paint.setColor(Color.BLUE); canvas.drawText(""+d11, (getWidth()/2-360)+225, (getHeight()/2-600)+345, paint);
```

```
paint.setColor(Color.BLACK); canvas.drawText(""+d12, (getWidth()/2-360)+285, (getHeight()/2-600)+345, paint);
```

```
paint.setColor(Color.BLACK); canvas.drawText(""+d21, (getWidth()/2-360)+225, (getHeight()/2-600)+375, paint);
```

```
paint.setColor(Color.BLACK); canvas.drawText(""+d22, (getWidth()/2-360)+285, (getHeight()/2-600)+375, paint);
```

```
paint.setColor(Color.BLUE); canvas.drawText(""+d31, (getWidth()/2-360)+225,
```

```
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d32, (getWidth()/2-360)+285,
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d41, (getWidth()/2-360)+225,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d42, (getWidth()/2-360)+285,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d51, (getWidth()/2-360)+225,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d52, (getWidth()/2-360)+285,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d61, (getWidth()/2-360)+225,
(getHeight()/2-600)+495, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d62, (getWidth()/2-360)+285,
(getHeight()/2-600)+495, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d13, (getWidth()/2-360)+345,
(getHeight()/2-600)+345, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d14, (getWidth()/2-360)+405,
(getHeight()/2-600)+345, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d23, (getWidth()/2-360)+345,
(getHeight()/2-600)+375, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d24, (getWidth()/2-360)+405,
(getHeight()/2-600)+375, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d33, (getWidth()/2-360)+345,
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d34, (getWidth()/2-360)+405,
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d43, (getWidth()/2-360)+345,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d44, (getWidth()/2-360)+405,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d53, (getWidth()/2-360)+345,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d54, (getWidth()/2-360)+405,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d63, (getWidth()/2-360)+345,
```

```

(getHeight()/2-600)+495, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d64, (getWidth()/2-360)+405,
(getHeight()/2-600)+495, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d15, (getWidth()/2-360)+465,
(getHeight()/2-600)+345, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d16, (getWidth()/2-360)+525,
(getHeight()/2-600)+345, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d25, (getWidth()/2-360)+465,
(getHeight()/2-600)+375, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d26, (getWidth()/2-360)+525,
(getHeight()/2-600)+375, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d35, (getWidth()/2-360)+465,
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d36, (getWidth()/2-360)+525,
(getHeight()/2-600)+405, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d45, (getWidth()/2-360)+465,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d46, (getWidth()/2-360)+525,
(getHeight()/2-600)+435, paint);
    paint.setColor(Color.BLUE);canvas.drawText(""+d55, (getWidth()/2-360)+465,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d56, (getWidth()/2-360)+525,
(getHeight()/2-600)+465, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d65, (getWidth()/2-360)+465,
(getHeight()/2-600)+495, paint);
    paint.setColor(Color.BLACK);canvas.drawText(""+d66, (getWidth()/2-360)+525,
(getHeight()/2-600)+495, paint);

    paint.setColor(Color.BLUE);
    paint.setTextSize(35.0F);
    canvas.drawText("■Let's observe how the odd product", (getWidth()/2-360)+50+20,
(getHeight()/2-600)+800+50, paint);
    canvas.drawText(" ratio approaches 0.25 (9/36) .", (getWidth()/2-360)+50+20,
(getHeight()/2-600)+830+50, paint);

    paint.setColor(Color.BLACK);

```

```

        paint.setTextSize(30.0F);
        canvas.drawText("Touch the screen five times 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);

        paint.setColor(Color.BLUE);
        paint.setTextSize(30.0F);
        canvas.drawText("Copyright(C) Sohun 2021.9.27", (getWidth()/2-360)+150+5,
(getHeight()/2-600)+1130, paint);    //作者表示

        if (flag >= 5) {
            if (d11<999 && d21<999 && d31<999 && d41<999 && d51<999 && d61<999 && d12<999 &&
d22<999 && d32<999 && d42<999 && d52<999 && d62<999 && d13<999 && d23<999 && d33<999 &&
d43<999 && d53<999 && d63<999 && d14<999 && d24<999 && d34<999 && d44<999 && d54<999 &&
d64<999 && d15<999 && d25<999 && d35<999 && d45<999 && d55<999 && d65<999 && d16<999 &&
d26<999 && d36<999 && d46<999 && d56<999 && d66<999) {
                invalidate();
            }
        }
    } //protected void onDraw(Canvas canvas)

@Override
public boolean onTouchEvent(MotionEvent event) {
    flag++;
    flag = flag % 6;

    syoki++;
    if (syoki > 6) {
        flag=0;                //自動識別子
        syoki=0;                //初期化識別子
    }
}

```

```

        ct1=0; //実験回数
        d11=0;d21=0;d31=0;d41=0;d51=0;d61=0; //度数 d31 とはさいころ 1 の目が 3 で、さい
ころ 2 の目が 1 の度数
        d12=0;d22=0;d32=0;d42=0;d52=0;d62=0;
        d13=0;d23=0;d33=0;d43=0;d53=0;d63=0;
        d14=0;d24=0;d34=0;d44=0;d54=0;d64=0;
        d15=0;d25=0;d35=0;d45=0;d55=0;d65=0;
        d16=0;d26=0;d36=0;d46=0;d56=0;d66=0;
        d=0; //積が奇数の度数
        ritu=0; //積が奇数の割合
    }

    invalidate();
    return false;
}
}

```

[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.mysaicoroeng.MySai coroEng
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3]MainActivity.java

```
/*
```

```
-----
    積が奇数の2個のさいころ(英語版)
    Android 4.1 (Jelly Bean)
    Copyright (C) K. Niwa 2021. 9. 27
-----
```

```
*/
```

```
package jp.kiyo.wuena.mysaicoroeng;

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;
}
}
```