

[1]MyRandomwalksEng.java

/*

ランダムウォーク(英語版)
Android 4.1 (Jelly Bean)
Copyright (C) K. Niwa 2021. 9. 25

*/

package jp.kiyo.wuena.myrandomwalkseng;

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 MyRandomwalksEng **extends** View {

private Bitmap **bitmap1** = **null**; //酔っぱらいの画像を宣言し初期化する

int **x=240+120,y=100**; //, oldx, oldy; //酔っぱらいの位置

int **xx**; //座標軸に使用

int **k=0**; //ふらつき回数

int **flag=0**; //停止識別子

int **ct=1**; //実験回数

int[] **cy** = **new int**[6]; //ふらつき30回目の位置のカウンター(配列)

double **r**; //右にふれるか左にふれるかの識別子(乱数)

int **i**; //ループカウンタ

int **first=0**; //最初識別子

int **pause=0**; //終了識別子

int **syoki=0**; //初期化識別子

```
int width;  
int height;
```

```
public MyRandomwalksEng(Context context) {  
    super(context);  
    init(context);  
}
```

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

```
public MyRandomwalksEng(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.hito1); //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;
```

```
int p=120; // y 軸方向の移動幅
```

```
int q=120; // x 軸方向の移動幅
```

```
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(45.0f);
    canvas.drawText(" Random Walk ", (getWidth()/2-360)+15+130+35, (getHeight()/2-
600)+80, paint);

    paint.setColor(Color.BLACK);
    canvas.drawLine((getWidth()/2-360)+60, (getHeight()/2-600)+270, (getWidth()/2-
360)+660, (getHeight()/2-600)+270, paint);    //数直線 (上)
    for (xx=80;xx<=640;xx=xx+20) {
        canvas.drawLine((getWidth()/2-360)+xx, (getHeight()/2-600)+148+p, (getWidth()/2-
360)+xx, (getHeight()/2-600)+152+p, paint);    //メモリ線
    }
    paint.setTextSize(23.0f);
    canvas.drawText("0", (getWidth()/2-360)+360-5, (getHeight()/2-600)+170+p, paint);
//メモリ値
    canvas.drawText("8", (getWidth()/2-360)+436, (getHeight()/2-600)+170+p, paint);
    canvas.drawText("16", (getWidth()/2-360)+516, (getHeight()/2-600)+170+p, paint);

```

```

canvas.drawText("24", (getWidth()/2-360)+596, (getHeight()/2-600)+170+p, paint);
canvas.drawText("-8", (getWidth()/2-360)+272, (getHeight()/2-600)+170+p, paint);
canvas.drawText("-16", (getWidth()/2-360)+192, (getHeight()/2-600)+170+p, paint);
canvas.drawText("-24", (getWidth()/2-360)+112, (getHeight()/2-600)+170+p, paint);

paint.setColor(Color.BLACK);
canvas.drawLine((getWidth()/2-360)+60, (getHeight()/2-600)+510, (getWidth()/2-
360)+660, (getHeight()/2-600)+510, paint); //数直線(下)
for (xx=80;xx<=640;xx=xx+20) {
    canvas.drawLine((getWidth()/2-360)+xx, (getHeight()/2-600)+388+p, (getWidth()/2-
360)+xx, (getHeight()/2-600)+392+p, paint); //メモリ線
}
paint.setTextSize(23.0f);
canvas.drawText("0", (getWidth()/2-360)+360-5, (getHeight()/2-600)+410+p, paint);
//メモリ値
canvas.drawText("8", (getWidth()/2-360)+436, (getHeight()/2-600)+410+p, paint);
canvas.drawText("16", (getWidth()/2-360)+516, (getHeight()/2-600)+410+p, paint);
canvas.drawText("24", (getWidth()/2-360)+596, (getHeight()/2-600)+410+p, paint);
canvas.drawText("-8", (getWidth()/2-360)+272, (getHeight()/2-600)+410+p, paint);
canvas.drawText("-16", (getWidth()/2-360)+192, (getHeight()/2-600)+410+p, paint);
canvas.drawText("-24", (getWidth()/2-360)+112, (getHeight()/2-600)+410+p, paint);

paint.setColor(Color.BLACK);
paint.setTextSize(30.0f);
canvas.drawText("■A drunken man starts from the origin 0", (getWidth()/2-360)+50,
(getHeight()/2-600)+950-90, paint);
canvas.drawText(" on the number line.", (getWidth()/2-360)+50, (getHeight()/2-
600)+950-90+30, paint);

canvas.drawText("Touch the screen to move the drunken man.", (getWidth()/2-360)+50,
(getHeight()/2-600)+950, paint);
canvas.drawText("Touch the screen again to stop him.", (getWidth()/2-360)+50,
(getHeight()/2-600)+990, paint);
canvas.drawText("If you touch it five times, 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("◎ The bar graph shows", (getWidth()/2-360)+50+20+10, (getHeight()/2-
600)+600+100, paint);
    canvas.drawText("the position of the 30th staggers.", (getWidth()/2-360)+50+100+20+10,
(getHeight()/2-600)+600+100+30, paint);

    paint.setColor(Color.BLUE);
    paint.setTextSize(30.0f);
    canvas.drawText("Copyright(C) Sohun 2021. 9. 25", (getWidth()/2-360)+150,
(getHeight()/2-600)+1130, paint);

    if (k<29) { //ふらつき29回目以下のとき
        if (first>0) {
            k++;
        }

        paint.setColor(Color.BLUE);
        paint.setTextSize(35.0f);
        if (first==0) {
            canvas.drawText("Number of staggers = ", (getWidth()/2-360)+165,
(getHeight()/2-600)+210+p, paint);
        }
        else if (first>0) {
            canvas.drawText("Number of staggers = "+(k+1), (getWidth()/2-360)+165,
(getHeight()/2-600)+210+p, paint);
        }

        paint.setColor(Color.BLUE);
        paint.setTextSize(35.0f);
        if (first==0) {
            canvas.drawText("Number of experiments = ", (getWidth()/2-360)+140,
(getHeight()/2-600)+450+p, paint);
        }
    }

```

```

else if (first>0) {
    canvas.drawText("Number of experiments = "+ct+", (getWidth()/2-360)+140,
(getHeight()/2-600)+450+p, paint);
}

r = Math.random();
if (r<0.5) { //左に動く
    x=x-10;
}
else if (r>=0.5) { //右に動く
    x=x+10;
}

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 bitmap2 = Bitmap.createBitmap( //-----***
    bitmap1, 0, 0, //-----***
    bitmap1.getWidth(), //-----***
    bitmap1.getHeight(), //-----***
    Mat, true //-----***
); //-----***

if (bitmap2 != null) {
    canvas.drawBitmap(bitmap2, (getWidth()/2-360)+x-35, (getHeight()/2-600)+y+p-20-
20-10, paint); //酔っぱらいを描写

```

```
    }  
}
```

```
else if (k==29) { //ふらつき30回目のとき  
    switch (x) {  
        case 360: //パチンコ玉の落下した位置の x 座標  
            cy[0]=cy[0]+1; //この位置に落下したパチンコ玉のカウンター  
            if (cy[0]>10) {  
                pause=1; //自動描画を止める識別子  
            }  
            break;  
        case 370: //パチンコ玉の落下した位置の x 座標  
            cy[1]=cy[1]+1; //この位置に落下したパチンコ玉のカウンター  
            if (cy[1]>10) {  
                pause=1; //自動描画を止める識別子  
            }  
            break;  
        case 380: //パチンコ玉の落下した位置の x 座標  
            cy[2]=cy[2]+1; //この位置に落下したパチンコ玉のカウンター  
            if (cy[2]>10) {  
                pause=1; //自動描画を止める識別子  
            }  
            break;  
        case 390: //パチンコ玉の落下した位置の x 座標  
            cy[3]=cy[3]+1; //この位置に落下したパチンコ玉のカウンター  
            if (cy[3]>10) {  
                pause=1; //自動描画を止める識別子  
            }  
            break;  
        case 400: //パチンコ玉の落下した位置の x 座標  
            cy[4]=cy[4]+1; //この位置に落下したパチンコ玉のカウンター  
            if (cy[4]>10) {  
                pause=1; //自動描画を止める識別子  
            }  
            break;  
        case 410: //パチンコ玉の落下した位置の x 座標
```

```

    cy[5]=cy[5]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[5]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 420: //パチンコ玉の落下した位置の x 座標
    cy[6]=cy[6]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[6]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 430: //パチンコ玉の落下した位置の x 座標
    cy[7]=cy[7]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[7]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 440: //パチンコ玉の落下した位置の x 座標
    cy[8]=cy[8]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[8]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 450: //パチンコ玉の落下した位置の x 座標
    cy[9]=cy[9]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[9]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 460: //パチンコ玉の落下した位置の x 座標
    cy[10]=cy[10]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[10]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 470: //パチンコ玉の落下した位置の x 座標

```



```
    cy[11]=cy[11]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[11]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 480: //パチンコ玉の落下した位置の x 座標
    cy[12]=cy[12]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[12]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 490: //パチンコ玉の落下した位置の x 座標
    cy[13]=cy[13]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[13]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 500: //パチンコ玉の落下した位置の x 座標
    cy[14]=cy[14]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[14]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 510: //パチンコ玉の落下した位置の x 座標
    cy[15]=cy[15]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[15]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 520: //パチンコ玉の落下した位置の x 座標
    cy[16]=cy[16]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[16]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 530: //パチンコ玉の落下した位置の x 座標
```

```

    cy[17]=cy[17]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[17]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 540: //パチンコ玉の落下した位置の x 座標
    cy[18]=cy[18]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[18]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 550: //パチンコ玉の落下した位置の x 座標
    cy[19]=cy[19]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[19]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 560: //パチンコ玉の落下した位置の x 座標
    cy[20]=cy[20]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[20]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 570: //パチンコ玉の落下した位置の x 座標
    cy[21]=cy[21]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[21]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 580: //パチンコ玉の落下した位置の x 座標
    cy[22]=cy[22]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[22]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 590: //パチンコ玉の落下した位置の x 座標

```

```

    cy[23]=cy[23]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[23]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 600: //パチンコ玉の落下した位置の x 座標
    cy[24]=cy[24]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[24]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 610: //パチンコ玉の落下した位置の x 座標
    cy[25]=cy[25]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[25]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 620: //パチンコ玉の落下した位置の x 座標
    cy[26]=cy[26]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[26]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 630: //パチンコ玉の落下した位置の x 座標
    cy[27]=cy[27]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[27]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 640: //パチンコ玉の落下した位置の x 座標
    cy[28]=cy[28]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[28]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 650: //パチンコ玉の落下した位置の x 座標

```

```

    cy[29]=cy[29]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[29]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 350: //パチンコ玉の落下した位置の x 座標
    cy[31]=cy[31]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[31]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 340: //パチンコ玉の落下した位置の x 座標
    cy[32]=cy[32]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[32]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 330: //パチンコ玉の落下した位置の x 座標
    cy[33]=cy[33]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[33]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 320: //パチンコ玉の落下した位置の x 座標
    cy[34]=cy[34]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[34]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 310: //パチンコ玉の落下した位置の x 座標
    cy[35]=cy[35]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[35]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 300: //パチンコ玉の落下した位置の x 座標

```

```

    cy[36]=cy[36]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[36]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 290: //パチンコ玉の落下した位置の x 座標
    cy[37]=cy[37]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[37]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 280: //パチンコ玉の落下した位置の x 座標
    cy[38]=cy[38]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[38]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 270: //パチンコ玉の落下した位置の x 座標
    cy[39]=cy[39]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[39]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 260: //パチンコ玉の落下した位置の x 座標
    cy[40]=cy[40]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[40]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 250: //パチンコ玉の落下した位置の x 座標
    cy[41]=cy[41]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[41]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 240: //パチンコ玉の落下した位置の x 座標

```

```

    cy[42]=cy[42]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[42]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 230: //パチンコ玉の落下した位置の x 座標
    cy[43]=cy[43]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[43]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 220: //パチンコ玉の落下した位置の x 座標
    cy[44]=cy[44]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[44]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 210: //パチンコ玉の落下した位置の x 座標
    cy[45]=cy[45]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[45]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 200: //パチンコ玉の落下した位置の x 座標
    cy[46]=cy[46]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[46]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 190: //パチンコ玉の落下した位置の x 座標
    cy[47]=cy[47]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[47]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 180: //パチンコ玉の落下した位置の x 座標

```

```

    cy[48]=cy[48]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[48]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 170: //パチンコ玉の落下した位置の x 座標
    cy[49]=cy[49]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[49]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 160: //パチンコ玉の落下した位置の x 座標
    cy[50]=cy[50]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[50]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 150: //パチンコ玉の落下した位置の x 座標
    cy[51]=cy[51]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[51]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 140: //パチンコ玉の落下した位置の x 座標
    cy[52]=cy[52]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[52]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 130: //パチンコ玉の落下した位置の x 座標
    cy[53]=cy[53]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[53]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 120: //パチンコ玉の落下した位置の x 座標

```

```

    cy[54]=cy[54]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[54]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 110: //パチンコ玉の落下した位置の x 座標
    cy[55]=cy[55]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[55]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 100: //パチンコ玉の落下した位置の x 座標
    cy[56]=cy[56]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[56]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 90: //パチンコ玉の落下した位置の x 座標
    cy[57]=cy[57]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[57]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 80: //パチンコ玉の落下した位置の x 座標
    cy[58]=cy[58]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[58]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
case 70: //パチンコ玉の落下した位置の x 座標
    cy[59]=cy[59]+1; //この位置に落下したパチンコ玉のカウンター
    if (cy[59]>10) {
        pause=1; //自動描画を止める識別子
    }
    break;
}

```



```

    paint.setColor(Color.BLUE);
    paint.setTextSize(35.0f);
    canvas.drawText("Number of staggers = "+(k+1), (getWidth()/2-360)+165,
(getHeight()/2-600)+210+p, paint);

    k=0;
    x=240+120;
    y=100;
    ct++;

    paint.setColor(Color.BLUE);
    paint.setTextSize(35.0f);
    canvas.drawText("Number of experiments = "+ct+"", (getWidth()/2-360)+140,
(getHeight()/2-600)+450+p, paint);

    r = Math.random();
    if (r<0.5) { //左に移動
        x=x-10;
    }
    else if (r>=0.5) { //右に移動
        x=x+10;
    }

    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 bitmap2 = Bitmap.createBitmap( //-----***
    bitmap1, 0, 0, //-----***
    bitmap1.getWidth(), //-----***
    bitmap1.getHeight(), //-----***
    Mat.TRUE //-----***
); //-----***

if (bitmap2 != null) {
    canvas.drawBitmap(bitmap2, (getWidth()/2-360)+x-35, (getHeight()/2-600)+y+p-20-
20-10, paint); //酔っぱらい画像の描写
}
}

paint.setColor(Color.BLUE);
canvas.drawRect((getWidth()/2-360)+356, (getHeight()/2-600)+390-
cy[0]*14+p, (getWidth()/2-360)+364, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+366, (getHeight()/2-600)+390-
cy[1]*14+p, (getWidth()/2-360)+374, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+376, (getHeight()/2-600)+390-
cy[2]*14+p, (getWidth()/2-360)+384, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+386, (getHeight()/2-600)+390-
cy[3]*14+p, (getWidth()/2-360)+394, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+396, (getHeight()/2-600)+390-
cy[4]*14+p, (getWidth()/2-360)+404, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+406, (getHeight()/2-600)+390-
cy[5]*14+p, (getWidth()/2-360)+414, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+416, (getHeight()/2-600)+390-
cy[6]*14+p, (getWidth()/2-360)+424, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+426, (getHeight()/2-600)+390-
cy[7]*14+p, (getWidth()/2-360)+434, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+436, (getHeight()/2-600)+390-
cy[8]*14+p, (getWidth()/2-360)+444, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+446, (getHeight()/2-600)+390-
cy[9]*14+p, (getWidth()/2-360)+454, (getHeight()/2-600)+390+p, paint);
canvas.drawRect((getWidth()/2-360)+456, (getHeight()/2-600)+390-

```

```
cy[10]*14+p, (getWidth()/2-360)+464, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+466, (getHeight()/2-600)+390-
cy[11]*14+p, (getWidth()/2-360)+474, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+476, (getHeight()/2-600)+390-
cy[12]*14+p, (getWidth()/2-360)+484, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+486, (getHeight()/2-600)+390-
cy[13]*14+p, (getWidth()/2-360)+494, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+496, (getHeight()/2-600)+390-
cy[14]*14+p, (getWidth()/2-360)+504, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+506, (getHeight()/2-600)+390-
cy[15]*14+p, (getWidth()/2-360)+514, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+516, (getHeight()/2-600)+390-
cy[16]*14+p, (getWidth()/2-360)+524, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+526, (getHeight()/2-600)+390-
cy[17]*14+p, (getWidth()/2-360)+534, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+536, (getHeight()/2-600)+390-
cy[18]*14+p, (getWidth()/2-360)+544, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+546, (getHeight()/2-600)+390-
cy[19]*14+p, (getWidth()/2-360)+554, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+556, (getHeight()/2-600)+390-
cy[20]*14+p, (getWidth()/2-360)+564, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+566, (getHeight()/2-600)+390-
cy[21]*14+p, (getWidth()/2-360)+574, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+576, (getHeight()/2-600)+390-
cy[22]*14+p, (getWidth()/2-360)+584, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+586, (getHeight()/2-600)+390-
cy[23]*14+p, (getWidth()/2-360)+594, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+596, (getHeight()/2-600)+390-
cy[24]*14+p, (getWidth()/2-360)+604, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+606, (getHeight()/2-600)+390-
cy[25]*14+p, (getWidth()/2-360)+614, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+616, (getHeight()/2-600)+390-
cy[26]*14+p, (getWidth()/2-360)+624, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+626, (getHeight()/2-600)+390-
cy[27]*14+p, (getWidth()/2-360)+634, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+636, (getHeight()/2-600)+390-
```

```
cy[28]*14+p, (getWidth()/2-360)+644, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+646, (getHeight()/2-600)+390-
cy[29]*14+p, (getWidth()/2-360)+654, (getHeight()/2-600)+390+p, paint);

    canvas.drawRect((getWidth()/2-360)+346, (getHeight()/2-600)+390-
cy[31]*14+p, (getWidth()/2-360)+354, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+336, (getHeight()/2-600)+390-
cy[32]*14+p, (getWidth()/2-360)+344, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+326, (getHeight()/2-600)+390-
cy[33]*14+p, (getWidth()/2-360)+334, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+316, (getHeight()/2-600)+390-
cy[34]*14+p, (getWidth()/2-360)+324, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+306, (getHeight()/2-600)+390-
cy[35]*14+p, (getWidth()/2-360)+314, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+296, (getHeight()/2-600)+390-
cy[36]*14+p, (getWidth()/2-360)+304, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+286, (getHeight()/2-600)+390-
cy[37]*14+p, (getWidth()/2-360)+294, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+276, (getHeight()/2-600)+390-
cy[38]*14+p, (getWidth()/2-360)+284, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+266, (getHeight()/2-600)+390-
cy[39]*14+p, (getWidth()/2-360)+274, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+256, (getHeight()/2-600)+390-
cy[40]*14+p, (getWidth()/2-360)+264, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+246, (getHeight()/2-600)+390-
cy[41]*14+p, (getWidth()/2-360)+254, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+236, (getHeight()/2-600)+390-
cy[42]*14+p, (getWidth()/2-360)+244, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+226, (getHeight()/2-600)+390-
cy[43]*14+p, (getWidth()/2-360)+234, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+216, (getHeight()/2-600)+390-
cy[44]*14+p, (getWidth()/2-360)+224, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+206, (getHeight()/2-600)+390-
cy[45]*14+p, (getWidth()/2-360)+214, (getHeight()/2-600)+390+p, paint);
    canvas.drawRect((getWidth()/2-360)+196, (getHeight()/2-600)+390-
cy[46]*14+p, (getWidth()/2-360)+204, (getHeight()/2-600)+390+p, paint);
```

```

        canvas.drawRect((getWidth()/2-360)+186, (getHeight()/2-600)+390-
cy[47]*14+p, (getWidth()/2-360)+194, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+176, (getHeight()/2-600)+390-
cy[48]*14+p, (getWidth()/2-360)+184, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+166, (getHeight()/2-600)+390-
cy[49]*14+p, (getWidth()/2-360)+174, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+156, (getHeight()/2-600)+390-
cy[50]*14+p, (getWidth()/2-360)+164, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+146, (getHeight()/2-600)+390-
cy[51]*14+p, (getWidth()/2-360)+154, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+136, (getHeight()/2-600)+390-
cy[52]*14+p, (getWidth()/2-360)+144, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+126, (getHeight()/2-600)+390-
cy[53]*14+p, (getWidth()/2-360)+134, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+116, (getHeight()/2-600)+390-
cy[54]*14+p, (getWidth()/2-360)+124, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+106, (getHeight()/2-600)+390-
cy[55]*14+p, (getWidth()/2-360)+114, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+96, (getHeight()/2-600)+390-
cy[56]*14+p, (getWidth()/2-360)+104, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+86, (getHeight()/2-600)+390-
cy[57]*14+p, (getWidth()/2-360)+94, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+76, (getHeight()/2-600)+390-
cy[58]*14+p, (getWidth()/2-360)+84, (getHeight()/2-600)+390+p, paint);
        canvas.drawRect((getWidth()/2-360)+66, (getHeight()/2-600)+390-
cy[59]*14+p, (getWidth()/2-360)+74, (getHeight()/2-600)+390+p, paint);

```

```

        if (flag==1 && pause==0) {
            invalidate();          //自動描写
        }
    }
}

```

```

@Override
public boolean onTouchEvent(MotionEvent event) {
    flag = flag + 1;
    flag = flag % 2;
}

```

```

        first++;

        syoki = syoki +1;
        if (syoki==5) {
            x=240+120;
            y=100;
            k=0;
            flag=0;
            ct=1;

            for (i=0;i<=59;i++) { //棒グラフの高さを0にする (配列を初期化する)
                cy[i]=0;
            }

            first=0;
            pause=0;
            syoki=0;
        }

        if (pause==0) {
            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.myrandomwalkseng.MyRandomwalksEng

```
    android:id="@+id/myfview1"
    android:layout_height="match_parent"
    android:layout_width="match_parent"/>
```

</androidx.constraintlayout.widget.ConstraintLayout>

[3]MainActivity.java

/*

ランダムウォーク(英語版)
Android 4.1 (Jelly Beans)
Copyright (C) K. Niwa 2021. 9. 25

*/

```
package jp.kiyo.wuena.myrandomwalkseng;
```

```
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;  
    }  
}
```