

[1] MyDeaino.java

/*

 出会いの実験
 Android 4.4 (Kit Kat)
 Copyright(C) K.Niwa 2019.12.13

*/

package jp.kiyo.wuena.mydeaino;

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 MyDeaino extends View {

 private Bitmap bitmap1 = null; //画像型として宣言し、初期化する
 private Bitmap bitmap2 = null; //画像型として宣言し、初期化する

 int mx=185-10,my=220; //桃太郎の位置
 int ix=265-10,iy=220; //犬の位置
 int oldmx,oldmy; //桃太郎の直前の位置
 int oldix,oldiy; //犬の直前の位置
 int flag=0; //自動識別子
 int k; //動き回数カウン

ター

 double r1,r2; //桃太郎、犬の動く方向（上下左右）
識別子（乱数）
 int ct=0; //出会い回数カウンター
 int count; //カウンター
 int syoki=0; //初期化識別子

```

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

```

```

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

```

```

public MyDeaino (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.momota);
    bitmap2 = BitmapFactory.decodeResource (res, R.drawable.inu);
}

```

@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;
    float c=0;
    float d=0;

    super.onDraw (canvas);
    canvas.drawColor (Color.WHITE);
    Paint paint = new Paint ();
    paint.setColor (Color.BLUE);
    paint.setAlpha (50);
    canvas.drawRect ((getWidth () /2-240) +10, (getHeight () /2-343) +10, (getWidth () /2-240)
+470, (getHeight () /2-343) +675, paint);

    paint.setAlpha (10000);
    paint.setColor (Color.BLUE);
}

```

```

    for (int i=0;i<2;i++) {
        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+10+i,(getWidth()
()/2-240)+10+i,(getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+10+i,(getHeight()/2-343)+675-i,(getWidth()
()/2-240)+470-i,(getHeight()/2-343)+675-i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+675-i,(getWidth()
()/2-240)+470-i,(getHeight()/2-343)+10+i, paint);
        canvas.drawLine((getWidth()/2-240)+470-i,(getHeight()/2-343)+10+i,(getWidth()
()/2-240)+10+i,(getHeight()/2-343)+10+i, paint);
    }

```

```

    if (MainActivity.ritsu != 0) {
        a=(float) 0.7*320/MainActivity.ritsu;
        b=(float) 0.7*320/MainActivity.ritsu;
        c=(float) 0.7*320/MainActivity.ritsu;
        d=(float) 0.7*320/MainActivity.ritsu;
    }

```

```

    else {
        a=(float) 1.0;
        b=(float) 1.0;
        c=(float) 1.0;
        d=(float) 1.0;
    }

```

```

Matrix Mat1 = new Matrix();
Mat1.postScale(a,b);
Bitmap bitmap11 = Bitmap.createBitmap(
    bitmap1,0,0,
    bitmap1.getWidth(),
    bitmap1.getHeight(),
    Mat1,true
);

```

```

Matrix Mat2 = new Matrix();
Mat2.postScale(c,d);
Bitmap bitmap22 = Bitmap.createBitmap(
    bitmap2,0,0,
    bitmap2.getWidth(),

```

```

        bitmap2.getHeight(),
        Mat2,true

    );

    paint.setColor(Color.BLUE);
    paint.setTextSize(23.0f);
    canvas.drawText("【出会いの実験】", (getWidth()/2-240)+175-24, (getHeight()/2-343)+70, paint);

    paint.setColor(Color.BLUE);
    paint.setTextSize(19.0f);
    canvas.drawText("出会い回数 = "+ct, (getWidth()/2-240)+180, (getHeight()/2-343)+430, paint);
    paint.setColor(Color.BLACK);
    canvas.drawText("動き回数 = "+k, (getWidth()/2-240)+180, (getHeight()/2-343)+460, paint);

    paint.setColor(Color.BLACK);
    paint.setTextSize(17.0f);

    if (flag==1 || flag==3) {
        canvas.drawText("                ", (getWidth()/2-240)+100, (getHeight()/2-343)+140, paint);
        canvas.drawText("                ", (getWidth()/2-240)+100, (getHeight()/2-343)+160, paint);
        canvas.drawText("                ", (getWidth()/2-240)+100, (getHeight()/2-343)+180, paint);
    }
    else {
        canvas.drawText("※桃太郎と犬の両方が動いた方が", (getWidth()/2-240)+100, (getHeight()/2-343)+140, paint);
        canvas.drawText("出会いやすいか、犬だけが動いた方", (getWidth()/2-240)+100, (getHeight()/2-343)+160, paint);
        canvas.drawText("が出会いやすいかを見てみましょう。", (getWidth()/2-240)+100, (getHeight()/2-343)+180, paint);
    }

    paint.setColor(Color.BLACK);
    canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()

```

```

/2-240)+390,(getHeight()/2-343)+100, paint);
        canvas.drawLine( (getWidth()/2-240)+90,(getHeight()/2-343)+100,(getWidth()/
/2-240)+90,(getHeight()/2-343)+400, paint);
        canvas.drawLine( (getWidth()/2-240)+90,(getHeight()/2-343)+400,(getWidth()/
/2-240)+390,(getHeight()/2-343)+400, paint);
        canvas.drawLine( (getWidth()/2-240)+390,(getHeight()/2-343)+400,(getWidth()/
/2-240)+390,(getHeight()/2-343)+100, paint);
        canvas.drawLine( (getWidth()/2-240)+89,(getHeight()/2-343)+99,(getWidth()/2-240)
+391,(getHeight()/2-343)+99, paint);
        canvas.drawLine( (getWidth()/2-240)+89,(getHeight()/2-343)+99,(getWidth()/2-240)
+89,(getHeight()/2-343)+401, paint);
        canvas.drawLine( (getWidth()/2-240)+89,(getHeight()/2-343)+401,(getWidth()/
/2-240)+391,(getHeight()/2-343)+401, paint);
        canvas.drawLine( (getWidth()/2-240)+391,(getHeight()/2-343)+401,(getWidth()/
/2-240)+391,(getHeight()/2-343)+99, paint);

        paint.setColor(Color.BLACK);
        paint.setTextSize(18.0f);
        canvas.drawText("※ 画面をタッチすると動きます。", (getWidth()/2-240)+50-20,
(getHeight()/2-343)+500, paint);
        canvas.drawText("※ 画面をタッチすると止まります。", (getWidth()/2-240)
+50-20, (getHeight()/2-343)+525, paint);
        canvas.drawText("※ 2人とも動く場合と1人だけ動く場合の様子", (getWidth()/
/2-240)+50-20, (getHeight()/2-343)+550, paint);
        canvas.drawText("を交互に見ることができます。", (getWidth()/2-240)+50-5,
(getHeight()/2-343)+570, paint);
        canvas.drawText("※ 画面が暗くなったらタイトルバーをタッチ!", (getWidth()/
/2-240)+50-20, (getHeight()/2-343)+595, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright(C) K.Niwa 2019.11", (getWidth()/2-240)+120,
(getHeight()/2-343)+630, paint);

        if (bitmap11 != null && bitmap22 != null) {
            canvas.drawBitmap(bitmap11, (getWidth()/2-240)+mx,(getHeight()/2-343)+my,
paint);
            canvas.drawBitmap(bitmap22, (getWidth()/2-240)+ix,(getHeight()/2-343)+iy,
paint);

```

```

k++;

if (flag==1) {
    r1=Math.random();           //桃太郎の動きの判断
    if (r1<0.25) {
        if (mx<320) {
            oldmx=mx;
            oldmy=my;
            mx=mx+10;           //右
            my=my;
        }
    }
    else if (r1<0.5) {
        if (my<320) {
            oldmx=mx;
            oldmy=my;
            mx=mx;
            my=my+10;           //下
        }
    }
    else if (r1<0.75) {
        if (mx>120) {
            oldmx=mx;
            oldmy=my;
            mx=mx-10;           //左
            my=my;
        }
    }
    else if (r1<1) {
        if (my>120) {
            oldmx=mx;
            oldmy=my;
            mx=mx;
            my=my-10;           //上
        }
    }
}

r2=Math.random();             //犬の動きの判断
if (r2<0.25) {
    if (ix<320) {

```

```

        oldix=ix;
        oldiy=iy;
        ix=ix+10;//右
        iy=iy;
    }
}
else if (r2<0.5) {
    if (iy<320) {
        oldix=ix;
        oldiy=iy;
        ix=ix;
        iy=iy+10;//下
    }
}
else if (r2<0.75) {
    if (ix>120) {
        oldix=ix;
        oldiy=iy;
        ix=ix-10; //左
        iy=iy;
    }
}
else if (r2<1) {
    if (iy>120) {
        oldix=ix;
        oldiy=iy;
        ix=ix;
        iy=iy-10; //上
    }
}

if (mx==ix && my==iy) {
    ct=ct+1;
    mx=185;my=220;ix=265;iy=220;
}

paint.setColor(Color.BLUE);
paint.setTextSize(19.0f);
canvas.drawText("出会\回数 = "+ct, (getWidth()/2-240)+180, (getHeight
()/2-343)+430, paint);

```

```
        paint.setColor(Color.BLACK);
        canvas.drawText("動き回数 = "+k, (getWidth()/2-240)+180, (getHeight()
/2-343)+460, paint);
```

```
    } //if (flag==1) {
```

```
    else if (flag==3) {
```

```
        r2=Math.random();           //犬の動きの判断
```

```
        if (r2<0.25) {
```

```
            if (ix<320) {
```

```
                oldix=ix;
```

```
                oldiy=iy;
```

```
                ix=ix+10; //右
```

```
                iy=iy;
```

```
            }
```

```
        }
```

```
        else if (r2<0.5) {
```

```
            if (iy<320) {
```

```
                oldix=ix;
```

```
                oldiy=iy;
```

```
                ix=ix;
```

```
                iy=iy+10; //下
```

```
            }
```

```
        }
```

```
        else if (r2<0.75) {
```

```
            if (ix>120) {
```

```
                oldix=ix;
```

```
                oldiy=iy;
```

```
                ix=ix-10; //左
```

```
                iy=iy;
```

```
            }
```

```
        }
```

```
        else if (r2<1) {
```

```
            if (iy>120) {
```

```
                oldix=ix;
```

```
                oldiy=iy;
```

```
                ix=ix;
```

```
                iy=iy-10; //上
```

```
            }
```

```

    }

    if (mx==ix && my==iy) {
        ct=ct+1;
        mx=185;my=220;ix=265;iy=220;
    }

    paint.setColor(Color.BLUE);
    paint.setTextSize(19.0f);
    canvas.drawText("出会い回数 = "+ct, (getWidth()/2-240)+180, (getHeight()
/2-343)+430, paint);
    paint.setColor(Color.BLACK);
    canvas.drawText("動き回数 = "+k, (getWidth()/2-240)+180, (getHeight()
/2-343)+460, paint);

    }//else if (flag==3) {

    }//if (bitmap1 != null && bitmap2 != null) {

    if (flag==1 || flag==3) {
        invalidate();
    }

} //protected void onDraw(Canvas canvas) {

@Override
public boolean onTouchEvent(MotionEvent event) {

    flag++;
    flag=flag % 4;

    if (flag==1 || flag==3) {
        k=0; //動きカウンターの初期化
        ct=0; //出会いカウンタ
        mx=185;my=220; //桃太郎の位置
        ix=265;iy=220; //犬の位置
    }
    invalidate();
    return false;
}

```

```
}
```

```
}//public class MyDeai 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.mydeaino.MyDeaino
        android:id="@+id/myfview1"
        android:layout_height="match_parent"
        android:layout_width="match_parent"/>

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

[3] MainActivity.java

```
/*
```

```
-----
    出会うの実験
    Android 4.4 (Kit Kat)
```

*/

```
package jp.kiyo.wuena.mydeaino;
```

```
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;
```

```
        }
```

```
    }
```