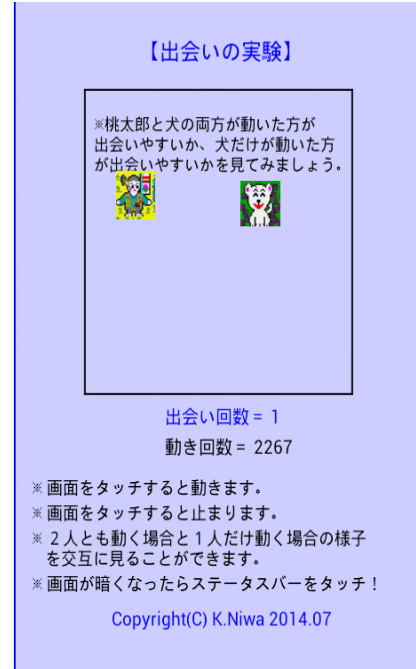


```
//-----
//
//      出会いの実験
//      Ver5
//      Copyright(C) K.Niwa 2014.07.21
//
//-----
```



【エミュレータ画面例】



【スマートフォン画面例】
(Android 4.0)

【アプリの概要】

ポロ博士も人の子、女性とデートすることもあります。ところがこの間は待ち合わせの場所が混雑していた上、暗くなりかけていたので彼女が博士を見つけるまでに1時間もかかってしまい大喧嘩になりました。

実は、かなりの近視の上に乱視のポロ博士としては、うろろう探し回るより、じっとしている方がいいと思って、「この辺で待ってて」と指定された範囲の中の一定の場所を動かずにいたのですが、彼女にはそれが気に喰わなかったのです。

彼女はお互いが相手を探して動き回る方が早く相手を見つけることができるはずだと主張します。もし、両方とも動かなかつたらいつまでも相手を見つけられないだろうというのがその根拠です。

もちろんポロ博士は、「山で遭難したら一歩も動かずに助けを待て」と言うとおりに、数学的には自分は絶対に正しいのだと思っはいるのですが、彼女の剣幕に少々自信がぐらつてきました。君ならどちらの味方をしますか。また、それは何故ですか。その理由を数学的に説明できますか。

(第4回日本数学コークール問題より)

このアプリでは、彼女を「犬」に、ポロ博士を「桃太郎」にして作成してあります。

【1】 Deaino.java

```
package jp.seitoku.deaino;

import android.content.Context;
import android.content.res.Resources; //画像用として追加
import android.graphics.*; //※に修正
import android.util.AttributeSet;
import android.view.*; //※に修正

public class Deaino 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,my=220; //桃太郎の位置
    int ix=265,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 Deaino(Context context, AttributeSet attrs, int defStyle) {
    super(context, attrs, defStyle);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

public Deaino(Context context, AttributeSet attrs) {
    super(context, attrs);
    // TODO 自動生成されたコンストラクター・スタブ
    init(context);
}

public Deaino(Context context) {
    super(context);
    // TODO 自動生成されたコンストラクター・スタブ
    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
    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);
    }
    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, (getHei
ght()/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, (getH
eight()/2-343)+160, paint);
            canvas.drawText("が出会いやすいかを見てください。", (getWidth()/2-240)+100, (ge
tHeight()/2-343)+180, paint);
        }

        paint.setColor(Color.BLACK);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()/2-240)+390, (get
Height()/2-343)+100, paint);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+100, (getWidth()/2-240)+90, (getH
eight()/2-343)+400, paint);
        canvas.drawLine((getWidth()/2-240)+90, (getHeight()/2-343)+400, (getWidth()/2-240)+390, (get
Height()/2-343)+400, paint);
        canvas.drawLine((getWidth()/2-240)+390, (getHeight()/2-343)+400, (getWidth()/2-240)+390, (ge
tHeight()/2-343)+100, paint);
        canvas.drawLine((getWidth()/2-240)+89, (getHeight()/2-343)+99, (getWidth()/2-240)+391, (getH
eight()/2-343)+99, paint);
        canvas.drawLine((getWidth()/2-240)+89, (getHeight()/2-343)+99, (getWidth()/2-240)+89, (getHe
ight()/2-343)+401, paint);
        canvas.drawLine((getWidth()/2-240)+89, (getHeight()/2-343)+401, (getWidth()/2-240)+391, (get
Height()/2-343)+401, paint);
        canvas.drawLine((getWidth()/2-240)+391, (getHeight()/2-343)+401, (getWidth()/2-240)+391, (ge
tHeight()/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()/2-3
43)+570, paint);
        canvas.drawText("※ 画面が暗くなったらステータスバーをタッチ!", (getWidth()/2-240)+50-2
0, (getHeight()/2-343)+595, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright (C) K.Niwa 2014.07", (getWidth()/2-240)+120, (getHeight()/2-34
3)+630, paint);

        if (bitmap1 != null && bitmap2 != null) {
            canvas.drawBitmap(bitmap1, (getWidth()/2-240)+mx, (getHeight()/2-343)+my, paint);
            canvas.drawBitmap(bitmap2, (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);

    } //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; //出会いカウンターの初期化
            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] main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:weightSum="1" >

    <jp.seitoku.deaino.Deaino
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:id="@+id/myview1" >

    </jp.seitoku.deaino.Deaino>
</LinearLayout>

```

[3] DeainoActivity.java

```

package jp.seitoku.deaino;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class DeainoActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}

```