




```
//-----
//
//          3人のじゃんけん
//          Ver7
//      Copyright(C) K.Niwa 2014.08.09
//
//-----
```



【エミュレータ画面例】

【3人のじゃんけん】

 A君
  B君
  C君

あいこの回数 = 157 (0.3007663)
 A君だけが勝った回数 = 74 (0.14176245)
 B君だけが勝った回数 = 75 (0.14367816)
 C君だけが勝った回数 = 52 (0.099616855)
 A君だけが負けた回数 = 57 (0.109195404)
 B君だけが負けた回数 = 57 (0.109195404)
 C君だけが負けた回数 = 50 (0.09578544)
実験回数 = 522

- ※ 画面を5回タッチすると自動になります。
- ※ 画面をタッチすると自動が止まります。
- ※ 更に画面をタッチすると初期化されます。
- ※ 画面が暗くなったらステータスバーをタッチ！

Copyright(C) K.Niwa 2014.08.09

【スマートフォン画面例】
(Android 4.0)

【アプリの概要】

A君、B君、C君の3人がじゃんけんするときの様子が観察できます。

1回のじゃんけんで、A君一人だけが勝つ場合、A君一人だけが負ける場合、あいこになる場合、どの場合が起こりやすいのでしょうか。

また、A君一人だけが勝つ確率、A君一人だけが負ける確率、あいこになる確率、一人だけが勝つ確率、一人だけが負ける確率、これらの確率を求めてみましょう。

【1】 Zyannken.java

```
package jp.seitoku.zyannken;
```

```
import android.content.Context;
import android.content.res.Resources;
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;
```

```
public class Zyannken extends View {
```

```
    private Bitmap bitmap1 = null;
    private Bitmap bitmap2 = null;
    private Bitmap bitmap3 = null;
```

```
    int ct=0;
    int r1=1, r2=2, r3=3;
```

はパー

```
    int d1=0, d2=0, d3=0, d4=0, d5=0, d6=0, d7=0; //勝ち負けのカウンター
    int flag=0;
    int syoki=0;
```

```
//実験回数カウンター
//A君はグー、B君はチョキ、C君
```

```
//自動識別子
//初期化識別子
```

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

```
    float a=0;
```

```

float b=0;

public Zyannken(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 Zyannken(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 Zyannken(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.guu);
    bitmap2 = BitmapFactory.decodeResource(res, R.drawable.pii);
    bitmap3 = BitmapFactory.decodeResource(res, R.drawable.paa);

    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;

    ct++;

    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(30.0f);
    canvas.drawText("【3人のじゃんけん】", (getWidth()/2-240)+120-30, (getHeight()/2-343)+110, paint);
}

```

```

paint.setColor(Color.BLACK);
paint.setTextSize(20.0f);
canvas.drawText("A君", (getWidth()/2-240)+123, (getHeight()/2-343)+210, paint);
canvas.drawText("B君", (getWidth()/2-240)+223, (getHeight()/2-343)+210, paint);
canvas.drawText("C君", (getWidth()/2-240)+323, (getHeight()/2-343)+210, paint);

if (ZyannkenActivity.ritsu != 0) {
    a=(float)0.9*320/ZyannkenActivity.ritsu; //----- <画像の
    拡大・縮小の横の倍率を指定する>
    b=(float)0.9*320/ZyannkenActivity.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 //-----***
); //-----***

//Mat.postScale(a, b); //-----***
Bitmap bitmap22 = Bitmap.createBitmap( //-----***
    bitmap2,0,0, //-----***
    bitmap2.getWidth(), //-----***
    bitmap2.getHeight(), //-----***
    Mat,true //-----***
); //-----***

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

if (bitmap11 != null && bitmap22 != null && bitmap33 != null) {

    //A君のじゃんけんの判断
    r1=(int) (1+3*Math.random());
    if (r1==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-240)+128, (getHeight()/2-343)+15
0, paint);
    }
    else if (r1==2) {
        canvas.drawBitmap(bitmap22, (getWidth()/2-240)+128, (getHeight()/2-343)+15
0, paint);
    }
    else if (r1==3) {
        canvas.drawBitmap(bitmap33, (getWidth()/2-240)+128, (getHeight()/2-343)+15
0, paint);
    }

    //B君のじゃんけんの判断
    r2=(int) (1+3*Math.random());
    if (r2==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-240)+228, (getHeight()/2-343)+15
0, paint);
    }
    else if (r2==2) {

```

```

        canvas.drawBitmap(bitmap22, (getWidth()/2-240)+228, (getHeight()/2-343)+15
0, paint);
    }
    else if (r2==3) {
        canvas.drawBitmap(bitmap33, (getWidth()/2-240)+228, (getHeight()/2-343)+15
0, paint);
    }

    //C君のじゃんけんの判断
    r3=(int) (1+3*Math.random());
    if (r3==1) {
        canvas.drawBitmap(bitmap11, (getWidth()/2-240)+328, (getHeight()/2-343)+15
0, paint);
    }
    else if (r3==2) {
        canvas.drawBitmap(bitmap22, (getWidth()/2-240)+328, (getHeight()/2-343)+15
0, paint);
    }
    else if (r3==3) {
        canvas.drawBitmap(bitmap33, (getWidth()/2-240)+328, (getHeight()/2-343)+15
0, paint);
    }
}

//勝ち負けの判断
//あいこ
    if (r1==1 && r2==1 && r3==1) {
        d1++;
    }
    else if (r1==2 && r2==2 && r3==2) {
        d1++;
    }
    else if (r1==3 && r2==3 && r3==3) {
        d1++;
    }
    else if (r1==1 && r2==2 && r3==3) {
        d1++;
    }
    else if (r1==1 && r2==3 && r3==2) {
        d1++;
    }
    else if (r1==2 && r2==1 && r3==3) {
        d1++;
    }
    else if (r1==2 && r2==3 && r3==1) {
        d1++;
    }
    else if (r1==3 && r2==1 && r3==2) {
        d1++;
    }
    else if (r1==3 && r2==2 && r3==1) {
        d1++;
    }
    //A君だけが勝つ
    if (r1==1 && r2==2 && r3==2) {
        d2++;
    }
    else if (r1==2 && r2==3 && r3==3) {
        d2++;
    }
    else if (r1==3 && r2==1 && r3==1) {
        d2++;
    }
    //B君だけが勝つ
    if (r1==2 && r2==1 && r3==2) {
        d3++;
    }
    else if (r1==3 && r2==2 && r3==3) {

```

```

        d3++;
    }
    else if (r1==1 && r2==3 && r3==1) {
        d3++;
    }
    //C君だけが勝つ
    if (r1==2 && r2==2 && r3==1) {
        d4++;
    }
    else if (r1==3 && r2==3 && r3==2) {
        d4++;
    }
    else if (r1==1 && r2==1 && r3==3) {
        d4++;
    }
    //A君だけが負ける
    if (r1==1 && r2==3 && r3==3) {
        d5++;
    }
    else if (r1==2 && r2==1 && r3==1) {
        d5++;
    }
    else if (r1==3 && r2==2 && r3==2) {
        d5++;
    }
    //B君だけが負ける
    if (r1==3 && r2==1 && r3==3) {
        d6++;
    }
    else if (r1==1 && r2==2 && r3==1) {
        d6++;
    }
    else if (r1==2 && r2==3 && r3==2) {
        d6++;
    }
    //C君だけが負ける
    if (r1==3 && r2==3 && r3==1) {
        d7++;
    }
    else if (r1==1 && r2==1 && r3==2) {
        d7++;
    }
    else if (r1==2 && r2==2 && r3==3) {
        d7++;
    }
}

paint.setColor(Color.BLACK);
paint.setTextSize(20.0f);
canvas.drawText("あいこの回数="+d1+" ( "+((float)(d1)/(float)(ct))+")
", (getWidth()/2-240)+50, (getHeight()/2-343)+260, paint);
canvas.drawText("A君だけが勝った回数="+d2+" ( "+((float)(d2)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+290, paint);
canvas.drawText("B君だけが勝った回数="+d3+" ( "+((float)(d3)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+320, paint);
canvas.drawText("C君だけが勝った回数="+d4+" ( "+((float)(d4)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+350, paint);
canvas.drawText("A君だけが負けた回数="+d5+" ( "+((float)(d5)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+380, paint);
canvas.drawText("B君だけが負けた回数="+d6+" ( "+((float)(d6)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+410, paint);
canvas.drawText("C君だけが負けた回数="+d7+" ( "+((float)(d7)/(float)(c
t))+")", (getWidth()/2-240)+50, (getHeight()/2-343)+440, paint);
paint.setColor(Color.BLUE);
canvas.drawText("実験回数="+ct, (getWidth()/2-240)+50, (getHeight()/2-3
43)+470, paint);

paint.setColor(Color.BLACK);
paint.setTextSize(19.0f);

```

```

        canvas.drawText("※ 画面を5回タッチすると自動になります。", (getWidth()
/2-240)+50-20, (getHeight()/2-343)+510, paint);
        canvas.drawText("※ 画面をタッチすると自動が止まります。", (getWidth()/2
-240)+50-20, (getHeight()/2-343)+540, paint);
        canvas.drawText("※ 更に画面をタッチすると初期化されます。", (getWidth()
/2-240)+50-20, (getHeight()/2-343)+570, paint);
        canvas.drawText("※ 画面が暗くなったらステータスバーをタッチ!", (getWid
th()/2-240)+50-20, (getHeight()/2-343)+600, paint);

        paint.setColor(Color.BLUE);
        paint.setTextSize(19.0f);
        canvas.drawText("Copyright (C) K.Niwa 2014. 08. 09", (getWidth()/2-240)+120
-20, (getHeight()/2-343)+630+10, paint);

        if (flag>=5) {
            invalidate();
        }
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        //invalidate();
        flag=flag+1;
        flag=flag % 6;

        syoki=syoki+1;
        if (syoki > 6) {
            ct=0; //実験回数カ
            d1=0;d2=0;d3=0;d4=0;d5=0;d6=0;d7=0; //勝ち負けのカウンター
            flag=0; //自動識別子
            syoki=0; //初期化識別子
        }

        invalidate();
        return false;
    }
}

```

ウンター

[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.zyannken.Zyannken
        android:layout_height="match_parent"
        android:layout_width="match_parent"
        android:id="@+id/myview1" >
    </jp.seitoku.zyannken.Zyannken>
</LinearLayout>

```

[3] ZannkenActivity.java

```

package jp.seitoku.zyannken;

```

```

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

import android.util.DisplayMetrics;           //----- <画像の拡大・縮小に必要なライブ
ラリ>

public class ZyannkenActivity extends Activity {

    static int ritsu;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.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; // -----これで値が取り出せ
た!
    }

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

```