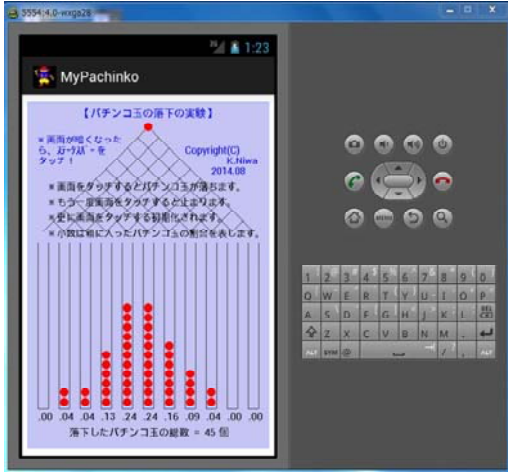
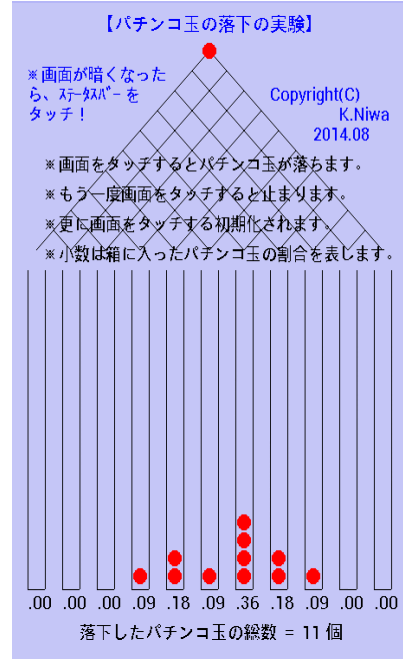


```
//-----
//
//          パチンコ玉の落下の実験
//          Ver9
//          Copyright(C) K.Niwa 2014.08.06
//
//-----
```



【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】

パチンコ玉の落下の様子が観察できます。  
 パチンコ玉は左右に半々の確率で分かれながら落下し、  
 11本の容器のいづれか1つの中に入ります。  
 容器に入るまでに、左右の分岐は10回あります。  
 どの容器にパチンコ玉は入りやすいでしょうか。  
 また、11本のそれぞれの容器について、パチンコ玉  
 が入る確率を数学的に求めてみましょう。

### 【1】 Pachinko.java

```
package jp.seitoku.pachinko;          //パッケージ名を指定

import java.util.concurrent.*;        //ライブラリの読み込み
import android.content.Context;
import android.graphics.*;
import android.util.AttributeSet;
import android.view.*;
import java.text.DecimalFormat;

//-----MySurfaceViewメソッド-----SurfaceHolder.Callbackクラスを実装したSurfaceViewクラスを継承したPachinkoクラス
public class Pachinko extends SurfaceView implements SurfaceHolder.Callback {

    //-----変数宣言-----
    private SurfaceHolder holder = null;
    float lastX, lastY;                //パチンコ玉の位置の x y座標
    double r;                          //乱数
    float r1, r2, r3, r4, r5, r6, r7, r8, r9, r10, r11; //各容器に入ったパチンコ玉の数の割合
    int lastColor;
    int flag, flag1, flag2, flag3, flag4, flag5, flag6, flag7, flag8, flag9, flag10, flag11; //各箱に落下した玉の個数のカウンター
    int s;                              //落下した玉の総数
    int pause=1;                       //停止識別子 → 停止:1, 動く:0
    int fc=0;                          //パチンコ玉の落下回数識別子
    //-----
```

```

public Pachinko(Context context, AttributeSet attrs, int defStyle) { //コンストラクタ
    super(context, attrs, defStyle);
    initSurface(); //MySurfaceViewを初期化するメソッド
}

public Pachinko(Context context, AttributeSet attrs) { //コンストラクタ
    super(context, attrs);
    initSurface(); //MySurfaceViewを初期化するメソッド
}

public Pachinko(Context context) { //コンストラクタ
    super(context);
    initSurface(); //MySurfaceViewを初期化するメソッド
}

public void surfaceCreated(SurfaceHolder holder) { //MSurfaceViewが出来上がったら
    drawSurface(); //MySurfaceViewに描くメソッド
    doAnim(); //繰り返しメソッド
}

public void surfaceDestroyed(SurfaceHolder holder) { //MySurfaceViewが終わったら
}

public void surfaceChanged(SurfaceHolder holder, int fprmat, int width, int height) { //MySurfaceView
// TODO 自動生成されたメソッド・スタブ
}

//----- initSurface() メソッド -----
public void initSurface() {
    lastColor = Color.WHYTE;
    holder = this.getHolder();
    holder.addCallback(this);
    holder.setFixedSize(getWidth(), getHeight());
}

//-----drawSurface() メソッド-----パチンコ玉の落下経路、箱、箱に溜まったパチンコ玉、落下前のパチン
コ玉、落下中のパチンコ玉を描く
public void drawSurface() {
    Canvas canvas = holder.lockCanvas(); //ロックして描写を非表示にする
    canvas.drawColor(Color.WHYTE); //背景を白色にする
    Paint paint = new Paint();
    paint.setColor(Color.BLUE); //描く色を青にする
    paint.setAlpha(50);
    canvas.drawRect((getWidth()/2-240)+10, (getHeight()/2-343)+10, (getWidth()/2-240)+470, (getH
eight()/2-343)+675, paint); //長方形を描く

    paint.setColor(Color.BLUE); //描く色を青にする
    paint.setAlpha(10000);

    for (int i=0;i<2;i++) {
        canvas.drawLine((getWidth()/2-240)+10+i, (getHeight()/2-343)+10+i, (getWidth()/2-24
0)+10+i, (getHeight()/2-343)+675-i, paint); //長方形を描く
        canvas.drawLine((getWidth()/2-240)+10+i, (getHeight()/2-343)+675-i, (getWidth()/2-2
40)+470-i, (getHeight()/2-343)+675-i, paint); //長方形を描く
        canvas.drawLine((getWidth()/2-240)+470-i, (getHeight()/2-343)+675-i, (getWidth()/2-
240)+470-i, (getHeight()/2-343)+10+i, paint); //長方形を描く
        canvas.drawLine((getWidth()/2-240)+470-i, (getHeight()/2-343)+10+i, (getWidth()/2-2
40)+10+i, (getHeight()/2-343)+10+i, paint); //長方形を描く
    }

    paint.setColor(Color.BLACK); //描く色を黒にする
    //paint.setAlpha(2000);

    for(int x = (getWidth()/2-240)+30; x<= (getWidth()/2-240)+450;x=x+20) { //箱
を描く

```

```

        canvas.drawLine(x, (getHeight()/2-343)+280, x, (getHeight()/2-343)+600, paint);
    }
    for(int x = (getWidth()/2-240)+30; x<= (getWidth()/2-240)+430;x=x+40) { // 箱
        canvas.drawLine(x, (getHeight()/2-343)+600, x+20, (getHeight()/2-343)+600, paint);
    }
    for(int x = 0; x<= 180;x=x+20) { //パチンコ玉が落ちる経路を描く
        canvas.drawLine((getWidth()/2-240)+240+x, (getHeight()/2-343)+x+60, (getWidth()/2-2
40)+40+2*x, (getHeight()/2-343)+260, paint);
        canvas.drawLine((getWidth()/2-240)+240-x, (getHeight()/2-343)+x+60, (getWidth()/2-2
40)+440-2*x, (getHeight()/2-343)+260, paint);
    }

    paint.setColor(Color.BLUE);
    paint.setTextSize(20.0f);
    //paint.setAlpha(2000);

    canvas.drawText("【パチンコ玉の落下の実験】", (getWidth()/2-240)+130-20, (getHeight()/2-3
43)+40, paint);

    paint.setTextSize(18.0f);
    paint.setColor(Color.BLUE);
    //paint.setAlpha(2000);

    canvas.drawText("※ 画面が暗くなった", (getWidth()/2-240)+30, (getHeight()/2-343)+90, pai
nt);
    canvas.drawText("ら、ステータスバーを", (getWidth()/2-240)+30, (getHeight()/2-343)+110, paint)
;
    canvas.drawText("タッチ!", (getWidth()/2-240)+30, (getHeight()/2-343)+130, paint);

    paint.setColor(Color.BLACK);
    //paint.setAlpha(2000);

    if (fc==0) {
        canvas.drawText("※ 画面をタッチするとパチンコ玉が落ちます。", (getWidth()/2-240)
+50, (getHeight()/2-343)+180, paint);
        canvas.drawText("※ もう一度画面をタッチすると止まります。", (getWidth()/2-240)+5
0, (getHeight()/2-343)+210, paint);
        canvas.drawText("※ 更に画面をタッチする初期化されます。", (getWidth()/2-240)+50,
(getHeight()/2-343)+240, paint);
        canvas.drawText("※ 小数は箱に入ったパチンコ玉の割合を表します。", (getWidth()/2-
240)+50, (getHeight()/2-343)+270, paint);
    }

    paint.setColor(Color.BLUE);
    paint.setTextSize(18.0f);
    //paint.setAlpha(2000);

    canvas.drawText("Copyright (C)", (getWidth()/2-240)+310, (getHeight()/2-343)+110, paint);
    canvas.drawText("K. Niwa", (getWidth()/2-240)+390, (getHeight()/2-343)+130, paint);
    canvas.drawText("2014.08", (getWidth()/2-240)+360, (getHeight()/2-343)+150, paint);

    s=flag1+flag2+flag3+flag4+flag5+flag6+flag7+flag8+flag9+flag10+flag11;

    r1=(float) flag1/s;
    r2=(float) flag2/s;
    r3=(float) flag3/s;
    r4=(float) flag4/s;
    r5=(float) flag5/s;
    r6=(float) flag6/s;
    r7=(float) flag7/s;
    r8=(float) flag8/s;
    r9=(float) flag9/s;
    r10=(float) flag10/s;
    r11=(float) flag11/s;

    DecimalFormat fmt = new DecimalFormat("#.00");

```

```

paint.setTextSize(19.0f);
paint.setColor(Color.BLACK);
//paint.setAlpha(2000);

if(s!=0) {
    canvas.drawText(""+fmt.format(r1), (getWidth()/2-240)+40-10, (getHeight()/2-343)+
620, paint);
    canvas.drawText(""+fmt.format(r2), (getWidth()/2-240)+80-10, (getHeight()/2-343)+
620, paint);
    canvas.drawText(""+fmt.format(r3), (getWidth()/2-240)+120-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r4), (getWidth()/2-240)+160-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r5), (getWidth()/2-240)+200-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r6), (getWidth()/2-240)+240-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r7), (getWidth()/2-240)+280-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r8), (getWidth()/2-240)+320-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r9), (getWidth()/2-240)+360-10, (getHeight()/2-343)
+620, paint);
    canvas.drawText(""+fmt.format(r10), (getWidth()/2-240)+400-10, (getHeight()/2-34
3)+620, paint);
    canvas.drawText(""+fmt.format(r11), (getWidth()/2-240)+440-10, (getHeight()/2-34
3)+620, paint);
    canvas.drawText("落下したパチンコ玉の総数 = "+s+" 個", (getWidth()/2-240)+90, (g
etHeight()/2-343)+650, paint);
}

//落下前と落下中のパチンコ玉を描く
if(flag==0) { //パチンコ玉が停止している場合
    paint.setColor(Color.RED); //描く色を赤にする
    for(int i=1;i<=4;i++) { //落下する前のパチンコ玉を描く
        canvas.drawCircle((getWidth()/2-240)+240, (getHeight()/2-343)+60, 2*i, pain
t);
    }
}
else if(flag==1) { //パチンコ玉が動いている場合
    if(flag1<17 && flag2<17 && flag3<17 && flag4<17 && flag5<17 && flag6<17 && flag7<
17 && flag8<17 && flag9<17 && flag10<17 && flag11<17) {
        if(lastY <= (getHeight()/2-343)+260) {
            paint.setColor(Color.RED); //描く色を赤にする
            for(int i=1;i<=4;i++) { //落下中のパチンコ玉を描く
                canvas.drawCircle(lastX, lastY, 2*i, paint);
            }
        }
    }
}
}

//箱に溜まっているパチンコ玉を描く
if(flag==0 || flag==1) { //パチンコ玉が停止しているかまたは動いている場合
    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag1;k++) { //箱1に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++) {
            canvas.drawCircle((getWidth()/2-240)+40, (getHeight()/2-343)+605-
18*k, 2*i, paint);
        }
    }
}

paint.setColor(Color.RED); //描く色を赤にする
for(int k=1;k<=flag2;k++) { //箱2に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++) {
        canvas.drawCircle((getWidth()/2-240)+80, (getHeight()/2-343)+605-
18*k, 2*i, paint);
    }
}

```

```

    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag3;k++){ //箱 3 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+120, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag4;k++){ //箱 4 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+160, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag5;k++){ //箱 5 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+200, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag6;k++){ //箱 6 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+240, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag7;k++){ //箱 7 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+280, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag8;k++){ //箱 8 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+320, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag9;k++){ //箱 9 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+360, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag10;k++){ //箱 1 0 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+400, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

paint.setColor(Color. RED); //描く色を赤にする
for(int k=1;k<=flag11;k++){ //箱 1 1 に落下したパチンコ玉を描く
    for(int i=1;i<=4;i++){
        canvas.drawCircle((getWidth()/2-240)+440, (getHeight()/2-343)+605
-18*k, 2*i, paint);
    }
}

```

```

    }
}

//箱に溜まっているパチンコ玉を描く
else if(flag==2){ //一つの箱にパチンコ玉が17個以上溜まったので停止の場合
    flag=3;
    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag1;k++){ //箱1に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+40, (getHeight()/2-343)+605-
18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag2;k++){ //箱2に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+80, (getHeight()/2-343)+605-
18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag3;k++){ //箱3に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+120, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag4;k++){ //箱4に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+160, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag5;k++){ //箱5に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+200, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag6;k++){ //箱6に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+240, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag7;k++){ //箱7に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+280, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする
    for(int k=1;k<=flag8;k++){ //箱8に落下したパチンコ玉を描く
        for(int i=1;i<=4;i++){
            canvas.drawCircle((getWidth()/2-240)+320, (getHeight()/2-343)+605-
-18*k, 2*i, paint);
        }
    }

    paint.setColor(Color.RED); //描く色を赤にする

```

```

        for(int k=1;k<=flag9;k++){ //箱 9 に落下したパチンコ玉を描く
            for(int i=1;i<=4;i++){
                canvas.drawCircle((getWidth()/2-240)+360, (getHeight()/2-343)+605
-18*k, 2*i, paint);
            }
        }

        paint.setColor(Color. RED); //描く色を赤にする
        for(int k=1;k<=flag10;k++){ //箱 1 0 に落下したパチンコ玉を描く
            for(int i=1;i<=4;i++){
                canvas.drawCircle((getWidth()/2-240)+400, (getHeight()/2-343)+605
-18*k, 2*i, paint);
            }
        }

        paint.setColor(Color. RED); //描く色を赤にする
        for(int k=1;k<=flag11;k++){ //箱 1 1 に落下したパチンコ玉を描く
            for(int i=1;i<=4;i++){
                canvas.drawCircle((getWidth()/2-240)+440, (getHeight()/2-343)+605
-18*k, 2*i, paint);
            }
        }
    }

    holder.unlockCanvasAndPost(canvas); //アンロックして描写を表示にする
} //public void drawSurface() {

private float abs(float f) {
    // TODO 自動生成されたメソッド・スタブ
    return 0;
}

//-----doAnim()メソッド-----繰り返しメソッド
public void doAnim() {
    ScheduledExecutorService executor = Executors.newSingleThreadScheduledExecutor(); Executo
r();
    executor.scheduleAtFixedRate(new Runnable() {

        public void run() {
            r=Math.random(); //乱数
            if(r<0.5 && lastY<(getHeight()/2-343)+260) { //右
                lastX=lastX+20;
            }
            else if(r>=0.5 && lastY<(getHeight()/2-343)+260) { //左へ落下する
                lastX=lastX-20;
            }
            lastY=lastY+20;
            lastX=Math.abs(lastX); //パチンコ玉の位置の x 座標を整
            lastY=Math.abs(lastY); //パチンコ玉の位置の y 座標を整
        }

        if (pause==0) { //パチンコ玉が動いている場合
            if(lastX==(getWidth()/2-240)+40.0f && lastY>(getHeight()/2-343)+
260.0f) { //箱 1 にパチンコ玉が落下したら
                lastX = (getWidth()/2-240)+240.0f;
                //パチンコ玉の位置を最初の地点に戻す
                lastY = (getHeight()/2-343)+60.0f;
                flag1=flag1+1; //箱
            }
            if(lastX==(getWidth()/2-240)+80.0f && lastY>(getHeight()/2-343)+
260.0f) { //箱 2 にパチンコ玉が落下したら
                lastX = (getWidth()/2-240)+240.0f;
                //パチンコ玉の位置を最初の地点に戻す
            }
        }
    });
}

```



```

                lastY = (getHeight()/2-343)+60.0f;
                flag2=flag2+1; // 箱
2に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱3にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+120.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag3=flag3+1; // 箱
3に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱4にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+160.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag4=flag4+1; // 箱
4に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱5にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+200.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag5=flag5+1; // 箱
5に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱6にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+240.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag6=flag6+1; // 箱
6に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱7にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+280.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag7=flag7+1; // 箱
7に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱8にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+320.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag8=flag8+1; // 箱
8に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱9にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+360.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag9=flag9+1; // 箱
9に落下したパチンコ玉の数を1個増やす
            }
+260.0f) { //箱10にパチンコ玉が落下したら
                if(lastX==(getWidth()/2-240)+400.0f && lastY>(getHeight()/2-343)
                    lastX = (getWidth()/2-240)+240.0f;
                    //パチンコ玉の位置を最初の地点に戻す
                    lastY = (getHeight()/2-343)+60.0f;
                    flag10=flag10+1; // 箱10に落
下したパチンコ玉の数を1個増やす
            }
}

```



```

        if(lastX==(getWidth()/2-240)+440.0f && lastY>(getHeight()/2-343)
+260.0f) { //箱 1 1 にパチンコ玉が落下したら
            lastX = (getWidth()/2-240)+240.0f;
            //パチンコ玉の位置を最初の地点に戻す
            lastY = (getHeight()/2-343)+60.0f;
            flag11=flag11+1; //箱 1 1 に落
下したパチンコ玉の数を1個増やす
        }
    } //if (pause==0) {

        if(flag1<17 && flag2<17 && flag3<17 && flag4<17 && flag5<17 && flag6<17
&& flag7<17 && flag8<17 && flag9<17 && flag10<17 && flag11<17) {
            drawSurface(); //パチンコ玉の落下経路、箱、箱に溜まったパチン
コ玉、落下前のパチンコ玉、落下中のパチンコ玉を描く
        }
        else{
            if(flag==1) { //パチンコ玉が動いている場合
                flag=2; //パチンコ玉を停止する
                drawSurface(); //パチンコ玉の落下経路、箱、箱に溜まっ
たパチンコ玉、落下前のパチンコ玉、落下中のパチンコ玉を描く
            }
        }

    }}, 0, 100, TimeUnit.MILLISECONDS);
}

private void Executor() {
    // TODO 自動生成されたメソッド・スタブ
}

//----- 画面をタッチしたら -----
@Override
public boolean onTouchEvent(MotionEvent event) {

    fc=fc+1; //パチンコ玉の落下回数を1回増やす
    pause=pause+1;
    pause=pause % 2; //pauseには0、1が入る

    if(pause==1) { //パチンコ玉が停止の場合
        lastX = (getWidth()/2-240)+240; //パチンコ玉の初期位置の座標
        lastY = (getHeight()/2-343)+60;
        flag=0; //パチンコ玉を停止
        fc=0; //パチンコ玉の落下回数を初期化する
    }
    else if (pause==0) { //パチンコ玉が動いている場合
        lastX = (getWidth()/2-240)+240;
        lastY = (getHeight()/2-343)+60;
        flag=1; //パチンコ玉を動かせる
        //箱に溜まったパチンコ玉を空にして
        flag1=0;flag2=0;flag3=0;flag4=0;flag5=0;flag6=0;flag7=0;flag8=0;flag9=0;flag10=0;
flag11=0;
    }

    return false;
}
}
}

```

## [ 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.pachinko.Pachinko
            android:layout_height="match_parent"
            android:layout_width="match_parent"
            android:id="@+id/myview1" >
    </jp.seitoku.pachinko.Pachinko>
</LinearLayout>

```

### [ 3 ] PachinkoActivity.java

```

package jp.seitoku.pachinko;

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

public class PachinkoActivity 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;
    }
}

```