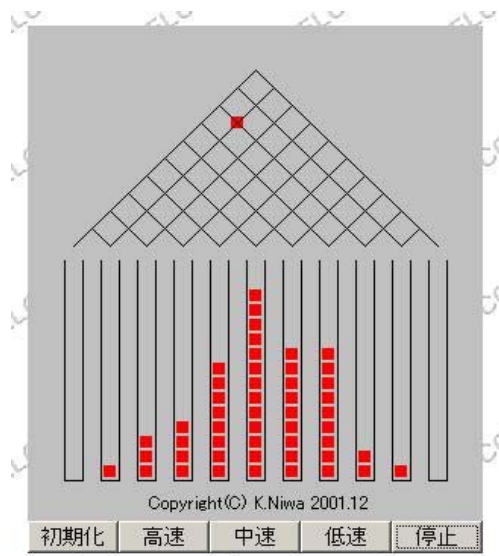
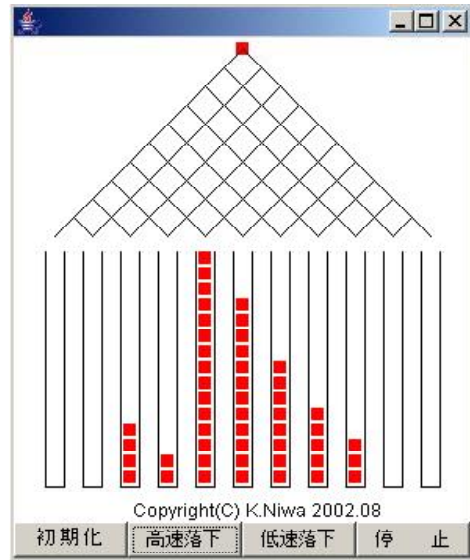


【パチンコ玉の落下の実験】



【J a v a アプレット】



【J a v a アプリケーション】

1. はじめに

1 個のパチンコ玉が釘に当たって左右に分かれながら落下するとき、パチンコ玉はどこに落下するのでしょうか。
ただし、パチンコ玉が釘に当たって左右に分かれるとき、その分かれ方は左右半々であるとした場合、パチンコ玉が落下し易い場所はあるのでしょうか。それとも、どこの場所も同じなのでしょうか。
シミュレーションソフト「パチンコ玉の落下の実験」を使って、パチンコ玉の落下の様子を観察してみてください。

2. J a v a アプレット

(1) J a v a プログラムリスト

```
//////////////////////////////////////////////////////////////////////
//                                                                 //
//                                     「パチンコ玉の落下の実験」 //
//                                     Copyright (C) K.Niwa 2001.12.01 //
//                                     (J a v a アプレット)           //
//                                                                 //
//////////////////////////////////////////////////////////////////////

import java.applet.Applet;
import java.awt.*;
import java.lang.Math;

//implements Runnable でスレッドを使えるようにする
public class Patinko5 extends Applet implements Runnable {

/****** instance *****/

    Thread thread;           // thread をスレッド型として定義する
    boolean active = false; // active を boolean 型として定義し初期化する (スレッド停止フラグ)
    MediaTracker mt;        // mt をメディアトラッカー型として定義する
    Image scrimg;          // scrimg をイメージ型として定義する (グラフィック描画イメージ)
    Graphics scrbuf;       // scrbuf をグラフィック型として定義する
}
```

```

int SLEEP = 70;           // スリープ時間(定数)
int x=150,y=30;          //パチンコ玉の x 座標と y 座標
int oldx=150,oldy=30;    //パチンコ玉の前の x 座標と y 座標
int xx;                  //パチンコ玉の落下先の容器描写に使用
int k;                   //パチンコ玉の経路描写に使用
Image ball;              //ball をイメージ型として定義する (パチンコ玉画像)
double r;                //乱数
int kai=10;              //10階建て
int kago[];              //容器の中の玉の数
int flagStop=0;         //停止ボタンを押したかどうかの識別子

/***** init *****/
public void init() // init メソッド {

    kago = new int [11];           //容器の中の玉の数を実体化する
    scrimg = createImage (300,350); // scrimg を新規作成する
    scrbuf = scrimg.getGraphics (); // scrimg の描画オブジェクトを設定

    ball = getImage (getDocumentBase (), "tama.jpg"); //パチンコ玉画像の読み込み
    mt = new MediaTracker (this); //メディアトラッカーを実体化する
    mt.addImage (ball,0); //イメージ(パチンコ玉画像)をメディアトラッカーへ貼り付ける

    try {                          //例外処理(メディアトラッカー完了まで待機)
        mt.waitForID (0);
    }
    catch (InterruptedException e) {};

    Button init = new Button ("初期化"); //init をボタン型として定義し実体化する
    Button fast = new Button ("高速"); //fast をボタン型として定義し実体化する
    Button middle = new Button ("中速"); //middle をボタン型として定義し実体化する
    Button slow = new Button ("低速"); //slow をボタン型として定義し実体化する
    Button teisi = new Button ("停止"); //teisi をボタン型として定義し実体化する

    Panel panel = new Panel (); //panel をパネル型として定義し実体化する
    panel.setBackground (Color.lightGray); //panel の背景色を lightGray に設定する
    panel.setFont (new Font ("Dialog", Font.PLAIN, 14)); //panel のフォントを設定する
    panel.setLayout (new GridLayout (1,5)); //panel のレイアウトを設定する
    panel.add (init); //panel に初期化ボタンを追加する
    panel.add (fast); //panel に高速ボタンを追加する
    panel.add (middle); //panel に中速ボタンを追加する
    panel.add (slow); //panel に低速ボタンを追加する
    panel.add (teisi); //panel に停止ボタンを追加する
    setLayout (new BorderLayout ()); //全体をボーダーレイアウトに設定する
    add ("South", panel); //panel をアプレットの南に追加する
    setBackground (Color.lightGray); //アプレットの背景色を lightGray に設定する

} //public void init ()

/***** start *****/
public void start() //start メソッド (スレッドを用いるときに使うメソッド) {

    thread = new Thread (this); //スレッドの実体化
    thread.start (); //スレッドを開始する

} //public void start ()

/***** stop *****/
//public void stop () //stop メソッド
//{
//    //thread.stop (); //スレッドを停止する
//    //thread = null; //スレッドを初期化する
//} //public void stop ()

```

```

/***** run *****/
public void run () { //run メソッド
    while ( true ) { // 無限ループ
        try {
            Thread.currentThread () .sleep (SLEEP); //休む
        }
        catch (InterruptedException e) {};
        repaint (); //paint () メソッドを呼び出す
    }
} //public void run ()

/***** paint *****/
public void paint (Graphics scr) { //paint メソッド
    scrbuf.clearRect (0,0,299,349); //全体をクリアする
    scrbuf.drawString ("Copyright (C) K.Niwa 2001.12",80,330); //作者名
    keirodraw (); //ユーザー定義メソッド
    youkidraw (); //ユーザー定義メソッド
    scrbuf.drawImage (ball,x-4,y-4,this); //パチンコ玉の描写
} //public void paint (Graphics scr)

/***** update *****/
public void update (Graphics scr) { //update メソッド

    int i;

    if (active == true) {
        if (kai >= 1) {
            r = Math.random ();
            if (r < 0.5) {
                x = x + 12;
                y = y + 12;
            }
            else {
                x = x - 12;
                y = y + 12;
            }
            scrbuf.drawImage (ball,x-4,y-4,this); //パチンコ玉を描写する
            scrbuf.clearRect (oldx-4,oldy-4,8,8); //パチンコ玉を消す
            keirodraw (); //ユーザー定義メソッド
            oldx = x;
            oldy = y;
            scr.drawImage (scrimg,0,0,this); //scrimg をアプレットに描画する
            kai--;
        } //if (kai >= 1)
        else {
            if (kai <= 0) {
                kago [(x-30)/24]++;
                //パチンコ玉の描写
                scrbuf.drawImage (ball,x-4,310-kago [(x-30)/24] * 10,this);
                //パチンコ玉を消す
                scrbuf.clearRect (oldx-4,oldy-4,8,8);
                keirodraw ();
                x = 150;
                y = 30;
                oldx = 150;
                oldy = 30;
                kai = 10;
            }
        }
    }
}

```

```

//パチンコ玉の描写
scrbuf.drawImage(ball,x-4,y-4,this);
} //if(kai<=0)

for(i=0;i<11;i++){
    if(kago[i]>=15){
        active = false;
        flagStop=0;
    }
};
} //else

} //if(active == true)

scr.drawImage(scrimg,0,0,this); //scrimg をアプレットに描画する

} //public void update(Graphics scr)

/***** action *****/

public boolean action(Event evt, Object obj) {
    if (evt.target instanceof Button) { //ボタン関係のイベントか?
        if ("高速".equals(obj)) { // "高速"ボタンを押したとき
            if (active == false) {
                active = !active;
                if (flagStop!=1) {
                    initdraw();
                }
                //repaint();
            }
            SLEEP=10; // (ウェイトを減少)
            return true;
        }

        if ("中速".equals(obj)) { // "中速"ボタンを押したとき
            if (active == false) {
                active = !active;
                if (flagStop!=1) {
                    initdraw();
                }
                //repaint();
            }

            SLEEP=70;
            return true;
        }

        if ("低速".equals(obj)) { // "低速"ボタンを押したとき
            if (active == false) {
                active = !active;
                if (flagStop!=1) {
                    initdraw();
                }
                //repaint();
            }
            SLEEP=200; // (ウェイトを増加)
            return true;
        }

        if ("初期化".equals(obj)) { // "初期化"ボタンを押したとき
            flagStop=0;
            if (active == true) {
                active = !active;
            }
        }
    }
}

```

```

        initdraw();
        return true;
    }

    if ("停止".equals(obj)) { // "停止"ボタンを押したとき
        flagStop=1;
        if (active == true) {
            active = !active;
        }
        //initdraw();
        //return true;
    }
} //if (evt.target instanceof Button)
return false;

} //public boolean action(Event evt, Object obj)

/*****      initdraw() ユーザー定義メソッド      *****/

public void initdraw() { //初期化を行うプロシージャ

    int i;
    kai=10;
    x=150;
    y=30;
    oldx=150;
    oldy=30;
    for (i=0;i<11;i++) {
        kago[i]=0;
    };
    scrbuf.clearRect(0,0,299,349); //全体クリア
    scrbuf.drawString("Copyright (C) K.Niwa 2001.12",80,330); //作者名
    keirodraw();
    youkidraw();
    scrbuf.drawImage(ball,x-4,y-4,this); //パチンコ玉の描写
    repaint();

} //public void initdraw()

/*****      keirodraw() ユーザー定義メソッド      *****/

public void keirodraw() { //落下経路を描写するプロシージャ
    for (k=12;k<=132;k=k+12) {
        scrbuf.drawLine(150+(k-12),30+(k-12),30+2*(k-12),150);
    }
    for (k=12;k<=132;k=k+12) {
        scrbuf.drawLine(150-(k-12),30+(k-12),270-2*(k-12),150);
    }
}

/*****      keirodraw() ユーザー定義メソッド      *****/

public void youkidraw() //容器を描写するプロシージャ
{
    for (xx=24;xx<=264;xx=xx+24)
    {
        scrbuf.drawLine(xx,310,xx,160);
        scrbuf.drawLine(xx+12,310,xx+12,160);
        scrbuf.drawLine(xx,310,xx+12,310);
    }
}

} //public class Patinko4 extends Applet implements Runnable

```

(2) HTMLリスト

```
<HTML>
<HEAD>
< !-----
                                   「パチンコ玉の落下の実験」
                                   Copyright (C) K.Niwa 2001.12.01
                                   ----->
</HEAD>
<BODY>
  <CENTER>
    <B>「パチンコ玉の落下の実験」</B>
    <BR><BR>
    <APPLET CODE="Patinko5.class" WIDTH="300" HEIGHT="360"></APPLET>
  </CENTER>
</BODY>
</HTML>
```

3. J a v a アプリケーション・プログラムリスト

```
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//                                                                                                                //
//////////////////////////////////////////////////////////////////////////////////////////////////////////////////

import java.awt.*;
import java.awt.event.*;
import java.lang.Math;

public class FPatinko3 extends Frame implements Runnable{ //スレッドを使えるようにする

  Thread myTh; //myTh をスレッド型で宣言する
  Button[] myBtn; //myBtn をボタン型配列で宣言する
  Panel myPanel; //myPanel をパネル型で宣言する
  int x=150,y=30,oldx,oldy; //パチンコ玉の x 座標と y 座標を整数型で宣言する
  int xx; //パチンコ玉の落下先の容器描写に使用
  int yy; //落下したパチンコ玉の描写に使用
  int k; //パチンコ玉の経路描写に使用
  int flag=0; //落下開始ボタンを押したか(1) 否か(0)の判断
  int y1=310,y2=310,y3=310,y4=310,y5=310,y6=310; //落下したパチンコ玉カントの y 座標
  int y7=310,y8=310,y9=310,y10=310,y11=310; //落下したパチンコ玉カントの y 座標
  double r; //乱数
  Image myImg; //イメージ型で宣言する
  MediaTracker myMt; //メディアトラッカー型で宣言する
  int count; //ループカウンター
  int Speed; //パチンコ玉の落下速度

//***** public FPatinko3() メソッド *****
public FPatinko3() {
  setSize (300,360); //フレームの大きさを設定する

  addWindowListener(new WindowAdapter() { //閉じるボタンのイベント処理
    public void windowClosing (WindowEvent e) {
      System.exit (0);
    }
  });

  myTh=null; //スレッドを初期化する
}
```

```

if (myTh==null) {
    myTh=new Thread (this);           //スレッドを実体化する
    myTh.start ();                     //スレッドをスタートする
}
//パチンコ玉画像を読み込む
myImg=Toolkit.getDefaultToolkit().getImage("tama.jpg");
myMt=new MediaTracker (this);         //メディアトラッカーを実体化する
myMt.addImage (myImg,0);              //イメージをメディアトラッカーに貼り付ける

try {
    myMt.waitForID (0);                //例外処理を実施する
}
catch (InterruptedException e) {
}

myBtn=new Button [4];                 //ボタンを実体化する
myBtn [0]=new Button ("初期化");      //ボタンを実体化する
myBtn [1]=new Button ("高速落下");
myBtn [2]=new Button ("低速落下");
myBtn [3]=new Button ("停止");

myPanel=new Panel ();                 //パネルを実体化する
myPanel.setLayout (new GridLayout (1,4)); //パネルをグリッドレイアウトに設定する
for (count=0;count<=3;count++) {
    myPanel.add (myBtn [count]);        //ボタンをパネルに貼り付ける
}
setLayout (new BorderLayout ());      //全体をボーダーレイアウトに設定する
add ("South",myPanel);                //パネルを南に貼り付ける

myBtn [0].addActionListener (new ActionListener () { //初期化ボタン
    public void actionPerformed (ActionEvent e) {
        flag=0;
        x=150;y=30;
        y1=310;y2=310;y3=310;y4=310;y5=310;y6=310;
        y7=310;y8=310;y9=310;y10=310;y11=310;
        repaint ();
    }
});

myBtn [1].addActionListener (new ActionListener () { //落下高速ボタン
    public void actionPerformed (ActionEvent e) {
        if (flag==0) {
            x=150;y=30;
            y1=310;y2=310;y3=310;y4=310;y5=310;y6=310;
            y7=310;y8=310;y9=310;y10=310;y11=310;
        }
        flag=1;
        Speed=30; //高速スピード
        repaint ();
    }
});

myBtn [2].addActionListener (new ActionListener () { //落下低速ボタン
    public void actionPerformed (ActionEvent e) {
        if (flag==0) {
            x=150;y=30;
            y1=310;y2=310;y3=310;y4=310;y5=310;y6=310;
            y7=310;y8=310;y9=310;y10=310;y11=310;
        }
        flag=2;
        Speed=300; //低速スピード
        repaint ();
    }
});

```

```

myBtn[3].addActionListener(new ActionListener() { //停止ボタン
    public void actionPerformed(ActionEvent e) {
        flag=3;
        repaint();
    }
});

} //public FPatinko3()

/***** public void run()メソッド *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(Speed);
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2) { //[高速ボタン]または[低速ボタン]が押されたとき
            repaint();
        }
    }
} //public void run()

/***** public void paint(Graphics g)メソッド *****/
public void paint(Graphics g) {

    if (myMt.checkID(0)) { //メディアトラッカーにイメージがちゃんと読み込めたとき
        //[初期化ボタン]を押したとき
        if (flag==0) {
            g.clearRect(0,0,300,360); //全体をクリアする
            g.drawString("Copyright (C) K.Niwa 2002.08",80,330); //作者名表示
            for (xx=24;xx<=264;xx=xx+24) { //容器を描写する
                g.drawLine(xx,310,xx,160);
                g.drawLine(xx+12,310,xx+12,160);
                g.drawLine(xx,310,xx+12,310);
            }
            g.drawImage(myImg,x-4,y-4,this); //パチンコ玉を描写する
            for (k=12;k<=132;k=k+12) { //落下経路を描写する
                g.drawLine(150+(k-12),30+(k-12),30+2*(k-12),150);
            }
            for (k=12;k<=132;k=k+12) {
                g.drawLine(150-(k-12),30+(k-12),270-2*(k-12),150);
            }
        } //else if (flag==0)

        //[高速落下ボタン]または[低速落下ボタン]を押したとき
        else if (flag==1 || flag==2 || flag==3) {
            if(y<180 && flag!=0) {
                if(flag==1 || flag==2) {
                    oldx=x;
                    oldy=y;
                    r=Math.random();
                    if (r<0.5 && y<150) { //右へ分岐する
                        x=x+12;
                        y=y+12;
                    }
                    else if (r>=0.5 && y<150) { //左へ分岐する
                        x=x-12;
                        y=y+12;
                    }
                }
                g.clearRect(oldx-4,oldy-4,8,8); //パチンコ玉を消す
            }
        }
    }
}

```



```

g.drawImage(myImg,x-4,y-4,this); //パチンコ玉を描く
//作者名
g.drawString("Copyright (C) K.Niwa 2002.08",80,330);
for (xx=24;xx<=264;xx=xx+24) { //容器を描写する
    g.drawLine(xx,310,xx,160);
    g.drawLine(xx+12,310,xx+12,160);
    g.drawLine(xx,310,xx+12,310);
}
for (k=12;k<=132;k=k+12) { //落下経路を描写する
    g.drawLine(150+(k-12),30+(k-12),30+2*(k-12),150);
}
for (k=12;k<=132;k=k+12) {
    g.drawLine(150-(k-12),30+(k-12),270-2*(k-12),150);
}

//落下したパチンコ玉を全て描く
for (yy=300;yy>=y1;yy=yy-10) {
    g.drawImage(myImg,26,yy,this);
}
if (y1==160) {
    flag=0;
}
for (yy=300;yy>=y2;yy=yy-10) {
    g.drawImage(myImg,50,yy,this);
}
if (y2==160) {
    flag=0;
}
for (yy=300;yy>=y3;yy=yy-10) {
    g.drawImage(myImg,74,yy,this);
}
if (y3==160) {
    flag=0;
}
for (yy=300;yy>=y4;yy=yy-10) {
    g.drawImage(myImg,98,yy,this);
}
if (y4==160) {
    flag=0;
}
for (yy=300;yy>=y5;yy=yy-10) {
    g.drawImage(myImg,122,yy,this);
}
if (y5==160) {
    flag=0;
}
for (yy=300;yy>=y6;yy=yy-10) {
    g.drawImage(myImg,146,yy,this);
}
if (y6==160) {
    flag=0;
}
for (yy=300;yy>=y7;yy=yy-10) {
    g.drawImage(myImg,170,yy,this);
}
if (y7==160) {
    flag=0;
}
for (yy=300;yy>=y8;yy=yy-10) {
    g.drawImage(myImg,194,yy,this);
}
if (y8==160) {
    flag=0;
}
}

```

```

for (yy=300;yy>=y9;yy=yy-10) {
    g.drawImage (myImg,218,yy,this) ;
}
if (y9==160) {
    flag=0;
}
for (yy=300;yy>=y10;yy=yy-10) {
    g.drawImage (myImg,242,yy,this) ;
}
    if (y10==160) {
        flag=0;
    }
for (yy=300;yy>=y11;yy=yy-10) {
    g.drawImage (myImg,266,yy,this) ;
}
if (y11==160) {
    flag=0;
}
}
//落下したパチンコ玉を全て描く終了
//落下したパチンコ玉の描写開始
if (y>=150) {
    //g.clearRect(0,0,300,310); //全体クリア

    oldx=x;
    oldy=y;
    g.clearRect(oldx-4,oldy-4,8,8);

    switch (x) { //落下したパチンコ玉の y 座標から
        case 30:
            y1=y1-10;
            break;
        case 54:
            y2=y2-10;
            break;
        case 78:
            y3=y3-10;
            break;
        case 102:
            y4=y4-10;
            break;
        case 126:
            y5=y5-10;
            break;
        case 150:
            y6=y6-10;
            break;
        case 174:
            y7=y7-10;
            break;
        case 198:
            y8=y8-10;
            break;
        case 222:
            y9=y9-10;
            break;
        case 246:
            y10=y10-10;
            break;
        case 270:
            y11=y11-10;
            break;
    } //switch (x)
}

```

```

//落下したパチンコ玉を全て描く
for (yy=300;yy>=y1;yy=yy-10) {
    g.drawImage(myImg,26,yy,this);
}
if (y1==160) {
    flag=0;
}
for (yy=300;yy>=y2;yy=yy-10) {
    g.drawImage(myImg,50,yy,this);
}
if (y2==160) {
    flag=0;
}
for (yy=300;yy>=y3;yy=yy-10) {
    g.drawImage(myImg,74,yy,this);
}
if (y3==160) {
    flag=0;
}
for (yy=300;yy>=y4;yy=yy-10) {
    g.drawImage(myImg,98,yy,this);
}
if (y4==160) {
    flag=0;
}
for (yy=300;yy>=y5;yy=yy-10) {
    g.drawImage(myImg,122,yy,this);
}
if (y5==160) {
    flag=0;
}
for (yy=300;yy>=y6;yy=yy-10) {
    g.drawImage(myImg,146,yy,this);
}
if (y6==160) {
    flag=0;
}
for (yy=300;yy>=y7;yy=yy-10) {
    g.drawImage(myImg,170,yy,this);
}
if (y7==160) {
    flag=0;
}
for (yy=300;yy>=y8;yy=yy-10) {
    g.drawImage(myImg,194,yy,this);
}
if (y8==160) {
    flag=0;
}
for (yy=300;yy>=y9;yy=yy-10) {
    g.drawImage(myImg,218,yy,this);
}
if (y9==160) {
    flag=0;
}
for (yy=300;yy>=y10;yy=yy-10) {
    g.drawImage(myImg,242,yy,this);
}
if (y10==160) {
    flag=0;
}
}

```

```

        for (yy=300;yy>=y11;yy=yy-10) {
            g.drawImage (myImg,266,yy,this) ;
        }
        if (y11==160) {
            flag=0;
        }
        //落下したパチンコ玉を全て描く終了
//----- 落下したパチンコ玉の描写終了

        x=150;y=30;
        g.drawImage (myImg,x-4,y-4,this) ;
        for (xx=24;xx<=264;xx=xx+24) { //容器を描写する
            g.drawLine (xx,310,xx,160) ;
            g.drawLine (xx+12,310,xx+12,160) ;
            g.drawLine (xx,310,xx+12,310) ;
        }
        for (k=12;k<=132;k=k+12) { //落下経路の描写
            g.drawLine (150+(k-12),30+(k-12),30+2*(k-12),150) ;
        }
        for (k=12;k<=132;k=k+12) {
            g.drawLine (150-(k-12),30+(k-12),270-2*(k-12),150) ;
        }

        } //if (y>=150)

    } //if (y<180 && flag!=0)

    } //else if (flag==1 || flag==2)

    } //if (myMt.checkID(0))

} //public void paint (Graphics g)

/***** public static void mainメソッド *****/
public static void main (String[] args) {
    Frame w=new FPatinko3 ();
    w.show ();
} //public static void main (String[] args)

} //public class FPatinko3 extends Frame implements Runnable

```