

【出会いの実験】



1. はじめに

「山で遭難したら、一步も動かずに助けを待て！」とされています。毎年、夏休みに名古屋大学で開かれている「日本数学コンクール」で出題された次の問題を考えてみましょう。

【第4回日本数学コンクール問題(1993年8月22日)】より

ボロ博士も人の子、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てくれるまでに1時間もかかってしまって大喧嘩になりました。

実は、かなりの近視の上に乱視のボロ博士としては、うろうろ探し回るより、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ほう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、両方とも動かなければいつまでたって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。

もちろんボロ博士は、「山で遭難したら一步も動かずに助けを待て」と言うとおり、数学的には自分は絶対正しいのだと思っではいるのですが、彼女の剣幕には少々自信がぐらついてきました。

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。

この問題を考えるときの参考にしようと、シミュレーションソフト「出会いの実験」を作ってみました。ただし、ボロ博士は桃太郎、彼女は犬として作ってあります。

このシミュレーションソフト「出会いの実験」を使って、2人とも動き回る場合と、1人が動き回って1人はじっとしている場合を、それぞれ実験してみてください。

2. Java アプレット

(1) Javaプログラムリスト

```

////////////////////////////////////
//
//                               //
//               「出会いの実験」           //
//      Copyright (C) K.Niwa 2001.12.20      //
//               ( J a v a アプレット )      //
//
////////////////////////////////////

```

```

//クラスの読み込み
import java.applet.Applet;
import java.awt.*;
import java.awt.event.*;
import java.lang.Math;

/***** public class Deai2 extends Applet implements Runnable *****/
public class Deai2 extends Applet implements Runnable {

    //変数とオブジェクトの型宣言
    Thread myTh;                //myTh をスレッド型として宣言する
    Button[] myBtn;             //myBtn をボタン型として宣言する
    Panel myPanel;              //myPanel をパネル型として宣言する
    int imgNo;

    //桃太郎の x 座標 mx、桃太郎の y 座標 my、犬の x 座標 ix、犬の y 座標 iy
    int mx=95,my=130,ix=175,iy=130;

    Image[] img;                //img をイメージ型として宣言する
    MediaTracker myMt;           //myMt をメディアトラッカー型として宣言する
    int flag=0;                 //初期状態の判断識別子
    int k;                       //カウンター
    double r1,r2;               //乱数
    long t;                     //時間稼ぎ
    int ct=0;                   //出会い回数
    int count;                  //カウンター
    int N=70;                   //速度

/***** public void init() *****/
    public void init() {
        myTh=null;              //スレッドの初期化
        imgNo=0;
        img=new Image[2];

        img[0]=getImage(getDocumentBase(),"momota.jpg");
        img[1]=getImage(getDocumentBase(),"inu.jpg");

        myMt=new MediaTracker(this);

        for (count=0;count<=1;count++) { //メディアトラッカーに画像を追加する
            myMt.addImage(img[count],0);
        }
        setBackground(Color.lightGray); //背景色をライトグレーにする
        try {
            myMt.waitForID(0);
        }
        catch (InterruptedException e) {
        }

        myBtn=new Button[4];        //ボタンを実体化する
        myBtn[0]=new Button("初期化");
        myBtn[1]=new Button("2人動く");
        myBtn[2]=new Button("1人動く");
        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する

        for (count=0;count<=3;count++) {
            myPanel.add(myBtn[count]); //パネルにボタンを貼り付ける
        }
    }
}

```

```

setLayout(new BorderLayout());          //全体をボーダーレイアウトにする
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;                          //識別子
        mx=95;my=130;ix=175;iy=130;
        ct=0;                            //出会い回数の初期化
        k=0;
        repaint();                       //paint メソッドを呼び出す
    }
});

myBtn[1].addActionListener(new ActionListener() { //2 人動くボタン
    public void actionPerformed(ActionEvent e) {
        flag=1;                          //識別子
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        repaint();
    }
});

myBtn[2].addActionListener(new ActionListener() { //1 人動くボタン
    public void actionPerformed(ActionEvent e) {
        flag=2;                          //識別子
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        repaint();
    }
});

myBtn[3].addActionListener(new ActionListener() { //停止ボタン
    public void actionPerformed(ActionEvent e) {
        flag=3;                          //識別子
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        //k=0;
        repaint();
    }
});
} //public void init()

/***** public void start() メソッド *****/
public void start() {
    if (myTh==null) {
        myTh=new Thread(this); //スレッドを実体化する
        myTh.start();          //スレッドを開始する
    }
} //public void start()

/***** public void run() メソッド *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(N); //1000 分の 70 秒休む
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2)
        {
            repaint(); //1000 分の 70 秒休む
        }
    }
} //public void run()

```

```

/***** public void paint(Graphics g) *****/
public void paint(Graphics g) {
    if (myMt.checkID(0)) {

        //初期状態および初期化ボタンを押したときのイベント処理
        if (flag==0) {
            g.clearRect(0,0,300,350);
            g.drawImage(img[0],mx,my,this);
            g.drawImage(img[1],ix,iy,this);
            //出会い回数の表示
            g.drawString("出会い回数="+ct+"      ( "+k+" )",120,305);
            g.drawString("Copyright(C) K.Niwa 2001.12",80,325);
        }//if (flag==0)

        //2人動くボタンを押したときのイベント処理
        else if (flag==1) {
            g.clearRect(0,0,300,305);           //全体クリア
            g.drawImage(img[0],mx,my,this);      //桃太郎の表示
            g.drawImage(img[1],ix,iy,this);      //犬の表示
            k++;

            //桃太郎と犬の動き開始
            //桃太郎の動きの判断
            r1=Math.random();                    //乱数を発生する
            if (r1<0.25) {
                if (mx<230) {
                    mx=mx+10;
                    my=my;
                }
            }
            else if (r1<0.5) {
                if (my<230) {
                    mx=mx;
                    my=my+10;
                }
            }
            else if (r1<0.75) {
                if (mx>30) {
                    mx=mx-10;
                    my=my;
                }
            }
            else if (r1<1) {
                if (my>30) {
                    mx=mx;
                    my=my-10;
                }
            }
        }

        //犬の動きの判断
        r2=Math.random();                    //乱数を発生する
        if (r2<0.25) {
            if (ix<230) {
                ix=ix+10;
                iy=iy;
            }
        }
        else if (r2<0.5) {
            if (iy<230) {
                ix=ix;
                iy=iy+10;
            }
        }
    }
}

```

```

        else if (r2<0.75) {
            if (ix>30) {
                ix=ix-10;
                iy=iy;
            }
        }
        else if (r2<1) {
            if (iy>30) {
                ix=ix;
                iy=iy-10;
            }
        }
    }

    //出会いの判定と回数のカウント
    if (mx==ix && my==iy) {
        ct=ct+1;
        mx=95;my=130;ix=175;iy=130;
    }

    g.clearRect(0,0,300,305); //部分クリア
    g.drawImage(img[1],ix,iy,this); //犬の表示
    g.drawImage(img[0],mx,my,this); //桃太郎の表示
    //出会い回数の表示
    g.drawString("出会い回数="+ct+" ( "+k+" )",120,305);
    g.drawString("Copyright (C) K.Niwa 2001.12",80,325);

    //桃太郎と犬の動き終了
} //else if (flag==1)

// 1 人動くボタンを押したときのイベント処理
else if (flag==2) {
    g.clearRect(0,0,300,305); //全体クリア
    g.drawImage(img[0],mx,my,this); //桃太郎の表示
    g.drawImage(img[1],ix,iy,this); //犬の表示
    k++;

    //犬の動き開始
    //犬の動きの判断
    r2=Math.random(); //乱数を発生する
    if (r2<0.25) {
        if (ix<230) {
            ix=ix+10;
            iy=iy;
        }
    }
    else if (r2<0.5) {
        if (iy<230) {
            ix=ix;
            iy=iy+10;
        }
    }
    else if (r2<0.75) {
        if (ix>30) {
            ix=ix-10;
            iy=iy;
        }
    }
    else if (r2<1) {
        if (iy>30) {
            ix=ix;
            iy=iy-10;
        }
    }
}

```

```

        //出会いの判定と回数のカウント
        if (mx==ix && my==iy) {
            ct=ct+1;
            mx=95;my=130;ix=175;iy=130;
        }

        g.clearRect(0,0,300,305); //部分クリア
        g.drawImage(img[1],ix,iy,this); //犬の表示
        g.drawImage(img[0],mx,my,this); //桃太郎の表示

        //出会い回数の表示
        g.drawString("出会い回数="+ct+" ( "+k+" )",120,305);
        g.drawString("Copyright (C) K.Niwa 2001.12",80,325);
        //桃太郎と犬の動き終了
    } //else if (flag==2)
    // 1 人が動くとき終了

    //停止ボタンを押したときのイベント処理
    else if (flag==3) {
        g.clearRect(0,0,300,305); //部分クリア
        g.drawImage(img[1],ix,iy,this); //犬の表示
        g.drawImage(img[0],mx,my,this); //桃太郎の表示

        //出会い回数の表示
        g.drawString("出会い回数="+ct+" ( "+k+" )",120,305);
        g.drawString("Copyright (C) K.Niwa 2001.12",80,325);

        //桃太郎と犬の動き終了

    } //else if (flag==3)
    //停止の終了

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

} //public void paint(Graphics g)

} //public class Deai2 extends Applet implements Runnable

```

(2) HTML リスト

```

<HTML>
<HEAD>
< !-----
                        「出会いの実験」
                        Copyright (C) K.Niwa 2001.12.20
----->
</HEAD>
    <BODY>
        <CENTER>
            <B>「出会いの実験」</B>
            <BR><BR>
            <APPLET CODE="Deai2.class" WIDTH="300" HEIGHT="350"></APPLET>
            <BR><BR>
        </CENTER>
    </BODY>
</HTML>

```

```

////////////////////////////////////
//
//                                     「F 出会いの実験」
//                                     Copyright (C) K.Niwa 2002.08.18
//                                     (J a v a アプリケーション)
//
//
////////////////////////////////////

```

- 42 -

```

try {
    myMt.waitForID(0);
}
catch (InterruptedException e) {
}

myBtn=new Button[4];
myBtn[0]=new Button("初期化");
myBtn[1]=new Button(" 2 人 動 く ");
myBtn[2]=new Button(" 1 人 動 く ");
myBtn[3]=new Button("停 止");

myPanel=new Panel();
myPanel.setLayout(new GridLayout(1,4));
for (count=0;count<=3;count++) {
    myPanel.add(myBtn[count]);
}
setLayout(new BorderLayout());
add("South",myPanel);

myBtn[0].addActionListener(new ActionListener() { //初期化ボタン
    public void actionPerformed(ActionEvent e) {
        flag=0;
        mx=95;my=130;ix=175;iy=130;
        ct=0; //出会い回数の初期化
        k=0;
        repaint();
    }
});

myBtn[1].addActionListener(new ActionListener() { // 2 人 動 く ボタン
    public void actionPerformed(ActionEvent e) {
        flag=1;
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        repaint();
    }
});

myBtn[2].addActionListener(new ActionListener() { // 1 人 動 く ボタン
    public void actionPerformed(ActionEvent e) {
        flag=2;
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        repaint();
    }
});

myBtn[3].addActionListener(new ActionListener() { //停止ボタン
    public void actionPerformed(ActionEvent e) {
        flag=3;
        //mx=95;my=130;ix=175;iy=130;
        //ct=0; //出会い回数の初期化
        //k=0;
        //repaint();
    }
});

} //public FDeai2()

```

```

/***** public void run() メソッド *****/
public void run() {
    while (true) {
        try {
            myTh.sleep(N);
        }
        catch (InterruptedException e) {}
        if (flag==1 || flag==2) {
            repaint();
        }
    }
} //public void run()

/***** public void paint(Graphics g) メソッド *****/
public void paint(Graphics g) {
    if (myMt.checkID(0)) {

        //初期状態および初期化ボタンを押したときのイベント処理
        if (flag==0) {
            //g.clearRect(0,0,300,350)
            g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
            g.clearRect(oldix,oldiy,30+5,40); //犬の消去
            g.drawImage(img[0],mx,my,this);
            g.drawImage(img[1],ix,iy,this);
            //出会い回数の表示
            g.drawString("出会い回数="+ct+"      ( "+k+" )",120,305);
            g.drawString("Copyright (C) K.Niwa 2002.08",80,325);
        } //if (flag==0)

        // 2人動くボタンを押したときのイベント処理
        else if (flag==1) {
            //g.clearRect(0,0,300,305); //全体クリア
            g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
            g.clearRect(oldix,oldiy,30+5,40); //犬の消去
            g.drawImage(img[0],mx,my,this); //桃太郎の表示
            g.drawImage(img[1],ix,iy,this); //犬の表示
            k++;

            //桃太郎と犬の動き開始
            //桃太郎の動きの判断
            r1=Math.random();
            if (r1<0.25) {
                if (mx<230) {
                    oldmx=mx;
                    oldmy=my;
                    mx=mx+10;
                    my=my;
                }
            }
            else if (r1<0.5) {
                if (my<230) {
                    oldmx=mx;
                    oldmy=my;
                    mx=mx;
                    my=my+10;
                }
            }
            else if (r1<0.75) {
                if (mx>30) {
                    oldmx=mx;
                    oldmy=my;
                    mx=mx-10;
                    my=my;
                }
            }
        }
    }
}

```

```

    }
    else if (r1<1) {
        if (my>30) {
            oldmx=mx;
            oldmy=my;
            mx=mx;
            my=my-10;
        }
    }
}

//犬の動きの判断
r2=Math.random();
if (r2<0.25) {
    if (ix<230) {
        oldix=ix;
        oldiy=iy;
        ix=ix+10;
        iy=iy;
    }
}
else if (r2<0.5) {
    if (iy<230) {
        oldix=ix;
        oldiy=iy;
        ix=ix;
        iy=iy+10;
    }
}
else if (r2<0.75) {
    if (ix>30) {
        oldix=ix;
        oldiy=iy;
        ix=ix-10;
        iy=iy;
    }
}
else if (r2<1) {
    if (iy>30) {
        oldix=ix;
        oldiy=iy;
        ix=ix;
        iy=iy-10;
    }
}
}

//出会いの判定と回数のカウント
if (mx==ix && my==iy) {
    ct=ct+1;
    mx=95;my=130;ix=175;iy=130;
}

//g.clearRect(0,0,300,305); //部分クリア
g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
g.clearRect(oldix,oldiy,30+5,40); //犬の消去
g.drawImage(img[1],ix,iy,this); //犬の表示
g.drawImage(img[0],mx,my,this); //桃太郎の表示
//出会い回数の表示
g.drawString("出会い回数="+ct+" ( "+k+" )",120,305);
g.drawString("Copyright (C) K.Niwa 2002.08",80,325);

//桃太郎と犬の動き終了
} //else if (flag==1)

```

```

// 1 人動くボタンを押したときのイベント処理
else if (flag==2) {
    //g.clearRect(0,0,300,305); //全体クリア
    g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
    g.clearRect(oldix,oldiy,30+5,40); //犬の消去
    g.drawImage(img[0],mx,my,this); //桃太郎の表示
    g.drawImage(img[1],ix,iy,this); //犬の表示
    k++;

    //犬の動き開始
    //犬の動きの判断
    r2=Math.random();
    if (r2<0.25) {
        if (ix<230) {
            oldix=ix;
            oldiy=iy;
            ix=ix+10;
            iy=iy;
        }
    }
    else if (r2<0.5) {
        if (iy<230) {
            oldix=ix;
            oldiy=iy;
            ix=ix;
            iy=iy+10;
        }
    }
    else if (r2<0.75) {
        if (ix>30) {
            oldix=ix;
            oldiy=iy;
            ix=ix-10;
            iy=iy;
        }
    }
    else if (r2<1) {
        if (iy>30) {
            oldix=ix;
            oldiy=iy;
            ix=ix;
            iy=iy-10;
        }
    }
    }

    //出会いの判定と回数のカウント
    if (mx==ix && my==iy) {
        ct=ct+1;
        mx=95;my=130;ix=175;iy=130;
    }

    //g.clearRect(0,0,300,305); //部分クリア
    g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
    g.clearRect(oldix,oldiy,30+5,40); //犬の消去
    g.drawImage(img[1],ix,iy,this); //犬の表示
    g.drawImage(img[0],mx,my,this); //桃太郎の表示
    //出会い回数の表示
    g.drawString("出会い回数="+ct+" ( "+k+" )",120,305);
    g.drawString("Copyright (C) K.Niwa 2002.08",80,325);

    //桃太郎と犬の動き終了
} //else if (flag==2)
// 1 人が動くとき終了

```

```

//停止ボタンを押したときのイベント処理
else if (flag==3) {
    //g.clearRect(0,0,300,305); //部分クリア
    g.clearRect(oldmx,oldmy,30+5,40); //桃太郎の消去
    g.clearRect(oldix,oldiy,30+5,40); //犬の消去
    g.drawImage(img[1],ix,iy,this); //犬の表示
    g.drawImage(img[0],mx,my,this); //桃太郎の表示
//出会い回数の表示
    g.drawString("出会い回数="+ct+"      ( "+k+" )",120,305);
    g.drawString("Copyright(C) K.Niwa 2002.08",80,325);
//桃太郎と犬の動き終了
} //else if (flag==3)
//停止の終了

} //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 FDeai2();
    w.show();
} //public static void main(String[] args)

} //public class FDeai2 extends Frame implements Runnable

```