

一、实验目的和要求

实验目的：掌握 Java Swing 组件的使用方法，理解委托事件处理模型，掌握多种布局方式，掌握窗口菜单和快捷菜单的设计方式，熟悉在组件上绘图的方法，设计出具有图形用户界面的、能够响应事件并处理异常的 Java 应用程序。

实验要求：

- ① 采用 Swing 组件设计图形用户界面，使用 3 个以上的相同组件要声明组件数组。
- ② 响应事件。
- ③ 处理异常，当输入数据错误时，弹出对话框，提示重新输入信息。

二、实验题目

线程管理弹弹球，控制每个球的大小、移动速度和方向均不同。

三、实验方法与步骤（需求分析、算法设计思路、流程图等）

首先设计一个球类，将球实体化，然后再设计一个画布，画布上面有多个球对象；接着再将画布放在一个 JFrame 用户界面上，使得球可以显示在用户面前。在画布上方添加一个工具面板，再在工具面板中增加不同的组合框与按钮，来控制画布中球的多少。

画布上增加一个线程，通过不停的 `repaint()`，使得球在画布上运动起来，然后通过控制线程的 `sleeptime` 来控制球在画布上重画的频率。

由于球是一个实体，可以通过给球增加不同的属性，来控制每个球的大小，方向以及速度。这样就可以达成本次实验的基本要求。

四、实验原始纪录（源程序、数据结构等）

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

private JTextField t[];
private Color color;
Thread thread;
public BallsJFrame()
public class BallsJFrame extends JFrame implements ActionListener, ChangeListener {
    //球的面板
    private BallsCanvas canvas;
    private JSpinner spinner;
    private JButton b1, b2, b3, b4;

    this.setBounds(300, 200, 600, 300);
    this.setDefaultCloseOperation(EXIT);
```

```

_ON_CLOSE);
        this.color=Color. red;
        this.canvas=new
BallsCanvas(this.color);

        this.getContentPane().add(this.can
vas);

        thread=new
Thread(this.canvas);

        JToolBar        panel=new
JToolBar();

        this.getContentPane().add(panel,"N
orth");

        panel.add(new JLabel("延时
"));

        SpinnerNumberModel
model=new SpinnerNumberModel();
        model.setStepSize(20);
        this.spinner=new
JSpinner(model);

        this.spinner.setValue(100);

        this.spinner.addChangeListener(thi
s);

        panel.add(this.spinner);

        b1=new JButton("选 择 颜 色
");

        b1.addActionListener(this);
        panel.add(b1);

        b2=new JButton("确定");

        b2.addActionListener(this);
        String str[]={"选择个数","
选择大小","选择速度"};
        this.t=new
JTextField[str.length];
        for(int
i=0;i<this.t.length;i++)
    {
        panel.add(new
JLabel(str[i]));
        panel.add(this.t[i]=new
JTextField(10));
    }
    panel.add(b2);

    b3=new JButton("开始");
    b4=new JButton("暂停");

    b3.addActionListener(this);

    b4.addActionListener(this);
    panel.add(b3);
    panel.add(b4);

    this.setVisible(true);
}

public          void
actionPerformed(ActionEvent ev)
{

    if(ev.getActionCommand().equals("
选择颜色"))

        this.color=JColorChooser. showDialo
g(this,"选择颜色",Color. white);

    if(ev.getActionCommand().equals("
确定"))
    {
        try
        {
            int
time=Integer. parseInt(""+this.spinner.
getValue());

            int
x=Integer. parseInt(t[0].getText());
            int
y=Integer. parseInt(t[1].getText());
            int
v=Integer. parseInt(t[2].getText());
            if(x>0&&y>0&&v>0)

```

```

this.canvas.addballs(this.color, x, time
, y, v);
else
    JOptionPane.showMessageDialog(null
, "请输入大于 0 的数字! ");
}

catch (NumberFormatException nex)
{

    JOptionPane.showMessageDialog(null,
"请输入正确数字! ");
}

if (ev.getActionCommand().equals("
开始"))
{
    thread=new
Thread(this.canvas);
    thread.start();

    this.b4.setEnabled(true);

    this.b3.setEnabled(false);
}

if (ev.getActionCommand().equals("
暂停"))
{
    thread.interrupt();

    this.b3.setEnabled(true);

    this.b4.setEnabled(false);
}

public void
stateChanged(ChangeEvent e)
{

    this.canvas.setDelay(Integer.parse
Int(""+this.spinner.getValue()));
}

}

public static void main(String
ags[])
{
    new BallsJFrame();
}

class BallsCanvas extends Canvas
implements Runnable
{
    private Ball balls[];
    Thread thread;
    int sleeptime=100;
    private static class Ball
    {
        int x,y;
        int size;
        int v;
        Color color;
        boolean up, left;
        Ball(int x,int y,Color
color,int size,int v)
        {
            this.x=x;
            this.y=y;
            this.size=size;
            this.v=v;
            this.color=color;
        }
    }

    public BallsCanvas(Color
color)
    //默认画板上只有一个球
    {
        this.balls=new Ball[1];

        balls[0]=new
Ball(40, 40, color, 20, 10);

        thread=new
Thread(this);
    }

    public void addballs(Color
color,int num,int delay,int size,int v)
    {

```

```

        Ball
olds[]=this.balls;
        int
len=olds.length+num;
        this.balls=new
Ball[len];
        for(int
i=0;i<olds.length;i++)
            this.balls[i]=new
Ball(olds[i].x,olds[i].y,olds[i].color
,olds[i].size,olds[i].v);
        for(int
i=olds.length,x=40;i<len;i++,x+=40)
            this.balls[i]=new
Ball(x,x,color,size,v);
    }
    public void setDelay(int
delay)
    {
        this.sleepTime=delay;
    }
    public void paint(Graphics
g)
    {
        for(int
i=0;i<balls.length;i++)
        {
            g.setColor(balls[i].color);

            balls[i].x=balls[i].left?balls[i].
x-balls[i].v:balls[i].x+balls[i].v;
            //控制球的水平速度
            if(balls[i].x<=0)
            {
                balls[i].left=!balls[i].left;
                balls[i].x=0;
            }

            if(balls[i].x>=this.getWidth())
            {
                balls[i].left=!balls[i].left;
                balls[i].x=this.getWidth();
            }

            balls[i].y=balls[i].up?balls[i].y-
balls[i].v:balls[i].y+balls[i].v;
            //控制球的垂直速度
            if(balls[i].y<=0)
            {
                balls[i].up=!balls[i].up;
                balls[i].y=0;
            }

            if(balls[i].y>=this.getHeight())
            {
                balls[i].up=!balls[i].up;
                balls[i].y=this.getHeight();
            }

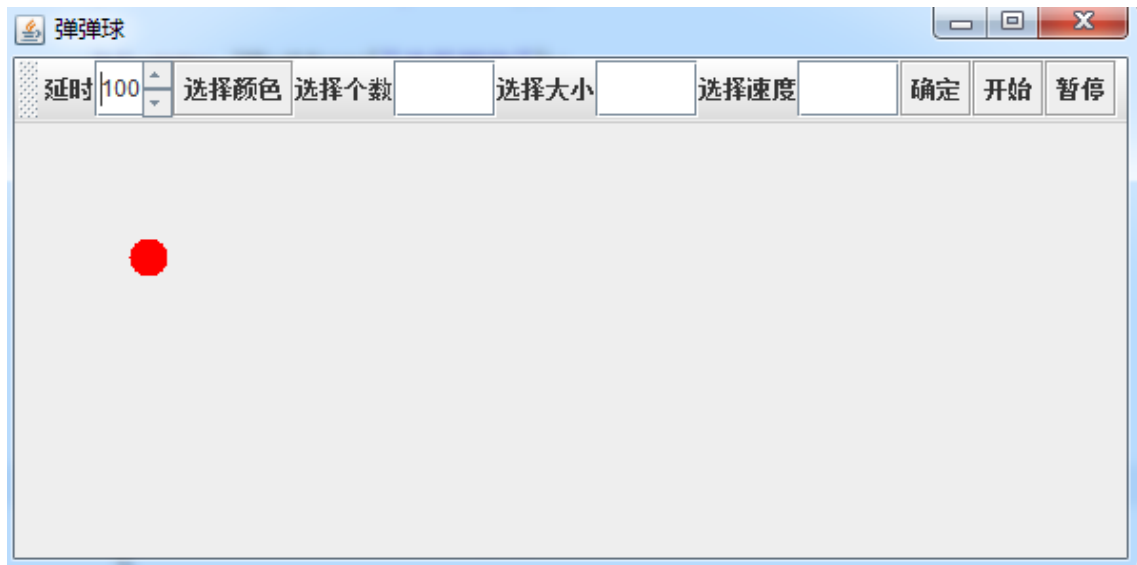
            g.fillOval(balls[i].x,balls[i].y,
balls[i].size,balls[i].size);
            //控制球的大小和方向
        }
    }
    public void run()
    //线程
    {
        while(true)
        try
        {
            repaint();

            Thread.sleep(sleepTime);
        }
        catch(InterruptedException ex)
        {
            break;
        }
    }
}

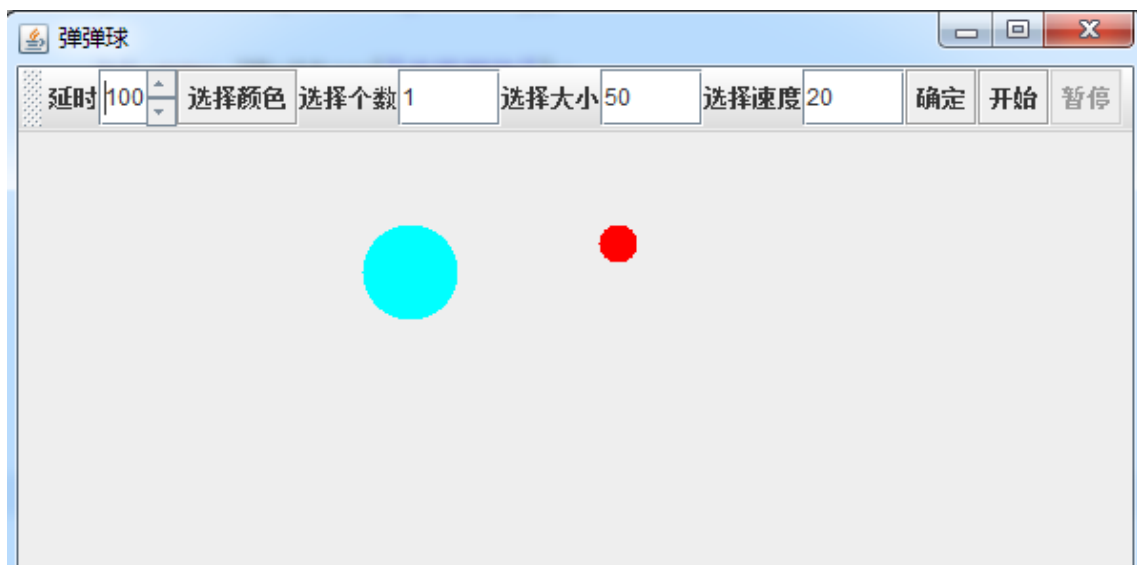
```

五、实验结果及分析（计算过程与结果、数据曲线、图表等）

默认一开始的界面只有一个球：

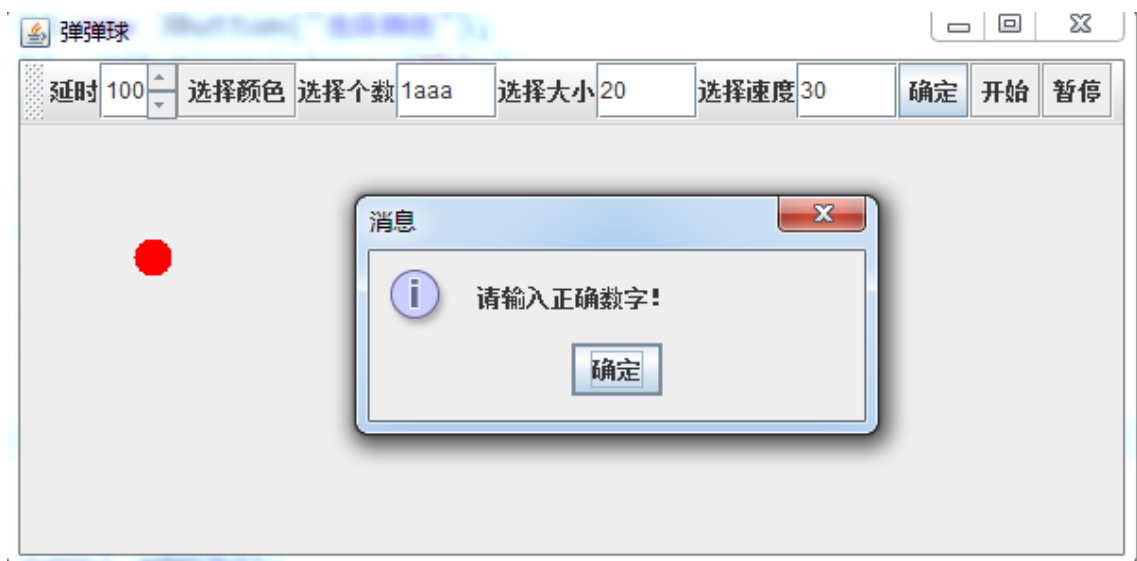


增加一个球

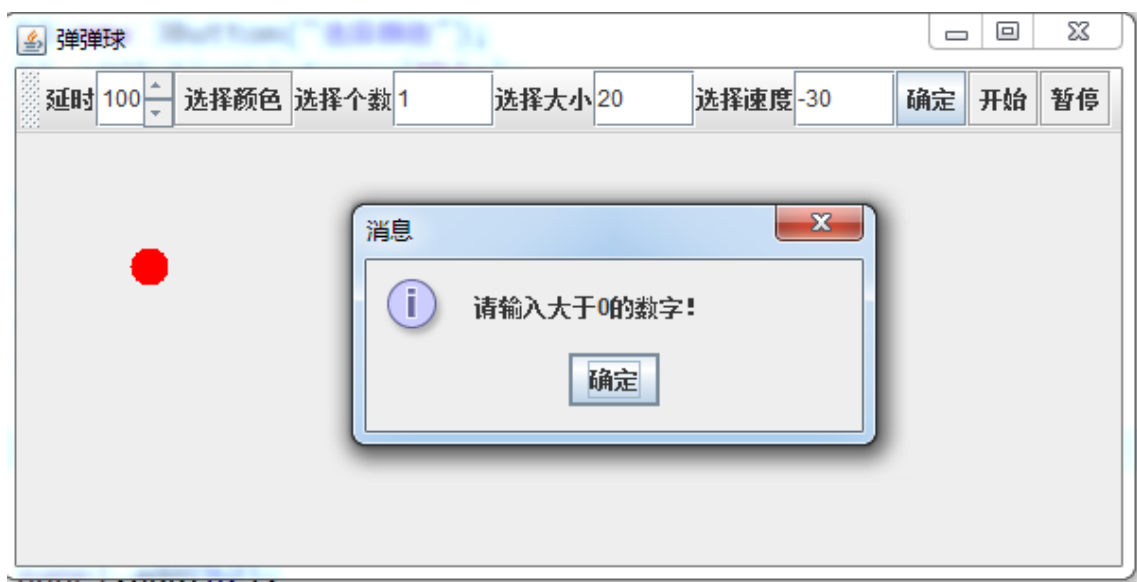


异常处理：

数值格式异常：



数字范围异常:



六、实验改进与思考

本次实验中，由于一开始的异常处理中，只是处理了数值格式异常，没有考虑到数字为负数的情况，所以，经过改进后，在文本框中数字进行转换的时候增加了一个选择条件，如果为零，则弹出一个对话框提醒用户。

因为一开始我没有完全明白，小球的属性，所以我不能控制每个球运动的方向，后来经过老师的指点之后，我知道了可以通过设置球的 `left` 和 `up` 来控制球的方向。