import java.awt.Button;

import java.awt.Color;

import java.awt.GridLayout;

import java.awt.Point;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.util.\*;

import javax.swing.JFrame;

import javax.swing.JOptionPane;

public class Snake extends JFrame [implements](https://wenwen.sogou.com/s/?w=implements&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank) KeyListener{

int Count=0;

Button[][] grid = new Button[20][20];

[ArrayList](https://wenwen.sogou.com/s/?w=ArrayList&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)<Point> snake\_list=new ArrayList<Point>();

Point bean=new Point(-1,-1);//保存随机豆子【坐标】

int [Direction](https://wenwen.sogou.com/s/?w=Direction&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank) = 1; //方向标志 1:上 2:下 3:左 4:右

//[构造方法](https://wenwen.sogou.com/s/?w=%E6%9E%84%E9%80%A0%E6%96%B9%E6%B3%95&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)

public Snake()

{

//窗体初始化

this.setBounds(400,300,390,395);

this.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

GridLayout f=new GridLayout(20,20);

this.getContentPane().[setBackground](https://wenwen.sogou.com/s/?w=setBackground&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)(Color.gray);

this.setLayout(f);

//初始化20\*20个按钮

for(int i=0;i<20;i++)

for(int j=0;j<20;j++)

{

grid[i][j]=new Button();

this.add(grid[i][j]);

grid[i][j].setVisible(false);

grid[i][j].addKeyListener(this);

grid[i][j].setBackground(Color.blue);

}

//蛇体初始化

grid[10][10].setVisible(true);

grid[11][10].setVisible(true);

grid[12][10].setVisible(true);

grid[13][10].setVisible(true);

grid[14][10].setVisible(true);

//在[动态数组](https://wenwen.sogou.com/s/?w=%E5%8A%A8%E6%80%81%E6%95%B0%E7%BB%84&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)中保存蛇体按钮坐标【行列】信息

snake\_list.add(new Point(10,10));

snake\_list.add(new Point(11,10));

snake\_list.add(new Point(12,10));

snake\_list.add(new Point(13,10));

snake\_list.add(new Point(14,10));

this.rand\_bean();

this.setTitle("总分：0");

this.setVisible(true);

}

//该方法随机一个豆子，且不在蛇体上，并使豆子可见

public void rand\_bean(){

Random rd=new Random();

do{

bean.x=rd.nextInt(20);//行

bean.y=rd.nextInt(20);//列

}while(snake\_list.contains(bean));

grid[bean.x][bean.y].setVisible(true);

grid[bean.x][bean.y].setBackground(Color.red);

}

//判断拟增[蛇头](https://wenwen.sogou.com/s/?w=%E8%9B%87%E5%A4%B4&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)是否与自身有碰撞

public boolean is\_cross(Point p){

boolean Flag=false;

for(int i=0;i<snake\_list.size();i++){

if(p.equals(snake\_list.get(i) )){

Flag=true;break;

}

}

return Flag;

}

//判断蛇即将前进位置是否有豆子，有返回true，无返回false

public boolean isHaveBean(){

boolean Flag=false;

int x=snake\_list.get(0).x;

int y=snake\_list.get(0).y;

Point p=null;

if([Direction](https://wenwen.sogou.com/s/?w=Direction&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)==1)p=new Point(x-1,y);

if(Direction==2)p=new Point(x+1,y);

if(Direction==3)p=new Point(x,y-1);

if(Direction==4)p=new Point(x,y+1);

if(bean.equals(p))Flag=true;

return Flag;

}

//前进一格

public void snake\_move(){

if(isHaveBean()==true){//////////////有豆子吃

Point p=new Point(bean.x,bean.y);//【很重要，保证吃掉的是豆子的复制对象】

snake\_list.add(0,p); //吃豆子

grid[p.x][p.y].[setBackground](https://wenwen.sogou.com/s/?w=setBackground&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)(Color.blue);

this.Count++;

this.setTitle("总分："+Count);

this.rand\_bean(); //再产生一个豆子

}else{///////////////////无豆子吃

//取原蛇头坐标

int x=snake\_list.get(0).x;

int y=snake\_list.get(0).y;

//根据蛇头坐标推算出拟新增蛇头坐标

Point p=null;

if([Direction](https://wenwen.sogou.com/s/?w=Direction&ch=ww.xqy.chain" \t "https://wenwen.sogou.com/z/_blank)==1)p=new Point(x-1,y);//计算出向上的新坐标

if(Direction==2)p=new Point(x+1,y);//计算出向下的新坐标

if(Direction==3)p=new Point(x,y-1);//计算出向左的新坐标

if(Direction==4)p=new Point(x,y+1);//计算出向右的新坐标

//若拟新增蛇头碰壁，或缠绕则游戏结束

if(p.x<0||p.x>19|| p.y<0||p.y>19||is\_cross(p)==true){

JOptionPane.showMessageDialog(null, "游戏结束！");

System.exit(0);

}

//向蛇体增加新的蛇头坐标,并使新蛇头可见

snake\_list.add(0,p);

grid[p.x][p.y].setVisible(true);

//删除原蛇尾坐标，使蛇尾不可见

int x1=snake\_list.get(snake\_list.size()-1).x;

int y1=snake\_list.get(snake\_list.size()-1).y;

grid[x1][y1].setVisible(false);

snake\_list.remove(snake\_list.size()-1);

}

}

@Override

public void keyPressed(KeyEvent e) {

if(e.getKeyCode()==KeyEvent.VK\_UP && Direction!=2) Direction=1;

if(e.getKeyCode()==KeyEvent.VK\_DOWN && Direction!=1) Direction=2;

if(e.getKeyCode()==KeyEvent.VK\_LEFT && Direction!=4) Direction=3;

if(e.getKeyCode()==KeyEvent.VK\_RIGHT && Direction!=3) Direction=4;

}

@Override

public void keyReleased(KeyEvent e) { }

@Override

public void keyTyped(KeyEvent e) { }

public static void main(String[] args) throws InterruptedException {

Snake win=new Snake();

while(true){

win.snake\_move();

Thread.sleep(300);

}

}

}