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 KeyListener{

int Count=0;

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

ArrayList snake\_list=new ArrayList();

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

int Direction = 1; //方向标志 1:上 2:下 3:左 4:右

//构造方法

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(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);

}

//判断拟增蛇头是否与自身有碰撞

public boolean is\_cross(Point p){

boolean Flag=false;

for(int i=0;i19|| 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);

}

}

}