

## SnakeGame\src\my\snakegame\SnakeGame.java

```

package my.snakegame;

public class SnakeGame extends javax.swing.JFrame {

    public SnakeGame() {
        initComponents();
        setLocationRelativeTo(null);
    }

    private void jMenuItem1ActionPerformed(java.awt.event.ActionEvent evt) {
        setSize(snakePanel1.getWidth()*snakePanel1.getSCALE()+17,
snakePanel1.getHeight()*snakePanel1.getSCALE()+jPanel1.getHeight()+jMenuBar1.getHeight()+55);
        setLocationRelativeTo(null);
        snakePanel1.timer.start();
    }
}

```

## SnakeGame\src\my\snakegame\SnakePanel.java

```

package my.snakegame;

import com.sun.glass.events.KeyEvent;
import java.awt.Color;
import java.awt.Graphics;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.Timer;
import my.snakegame.objects.Snake;

public class SnakePanel extends javax.swing.JPanel {
    private static final int SCALE=32;
    private static final int WIDTH=20;
    private static final int HEIGHT=20;
    private static final int SPEED=5;

    Timer timer = new Timer(1000/SPEED,new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            snake.move();
            repaint();
        }
    });
    public static int getSCALE() {
        return SCALE;
    }

    public static int getWIDTH() {
        return WIDTH;
    }

    public static int getHEIGHT() {
        return HEIGHT;
    }

    Snake snake = new Snake(10,10,9,10);

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(new Color(5, 50, 10));
        g.fillRect(0, 0, WIDTH*SCALE, HEIGHT*SCALE);
    }
}

```

```

g.setColor(Color.YELLOW);
for (int i = 0; i <= WIDTH*SCALE; i+=SCALE) {
    g.drawLine(i, 0, i, HEIGHT*SCALE);
}
for (int i = 0; i <= HEIGHT*SCALE; i+=SCALE) {
    g.drawLine(0, i, WIDTH*SCALE,i);
}
g.setColor(new Color(155, 250, 100));
for (int i = 0; i < snake.getLength(); i++) {
    int x1=snake.getSnakeX()[i]*SCALE+1;
    int y1=snake.getSnakeY()[i]*SCALE+1;
    int x2=SCALE-1;
    int y2=SCALE-1;
    g.fillRect(x1, y1, x2, y2);
}
}

public SnakePanel() {
    initComponents();
    setFocusable(true); //Установить панель в фокусе
}

private void formKeyPressed(java.awt.event.KeyEvent evt) {
    int key= evt.getKeyCode(); //Получаем код клавиши
    if (key== KeyEvent.VK_**LEFT** && snake.getDirection()!=0) snake.setDirection(1);
    if (key== KeyEvent.VK_**RIGHT** && snake.getDirection()!=1) snake.setDirection(0);
    if (key== KeyEvent.VK_**UP** && snake.getDirection()!=2) snake.setDirection(3);
    if (key== KeyEvent.VK_**DOWN** && snake.getDirection()!=3) snake.setDirection(2);
}
}

```

\SnakeGame\src\my\snakegame\objects\Snake.java

```
package my.snakegame.objects;
```

```

public class Snake {
    private int direction=3;
    private int lenght=10;

    private int snakeX[]= new int[100];
    private int snakeY[]= new int[100];

    public Snake(int x0, int y0, int x1, int y1) {
        snakeX[0]=x0;
        snakeY[0]=y0;
        snakeX[1]=x1;
        snakeY[1]=y1;
    }

    public void move() {
        for (int i = lenght; i >0 ; i--) {
            snakeX[i]=snakeX[i-1];
            snakeY[i]=snakeY[i-1];
        }
        if (direction==0) snakeX[0]++;
        if (direction==1) snakeX[0]--;
        if (direction==2) snakeY[0]++;
        if (direction==3) snakeY[0]--;
        //проверка наезжания на хвост
        for (int i = lenght-1; i >2;i--)
            if (snakeX[0]==snakeX[i]&& snakeY[0]==snakeY[i]) lenght= i;
    }
}

```

```
public int getDirection() {  
    return direction;  
}  
  
public void setDirection(int direction) {  
    this.direction = direction;  
}  
  
public int getLenght() {  
    return lenght;  
}  
  
public void setLenght(int lenght) {  
    this.lenght = lenght;  
}  
  
public int[] getSnakeX() {  
    return snakeX;  
}  
  
public int[] getSnakeY() {  
    return snakeY;  
}  
}
```