snakepanel.java

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Arrays;
import java.util.Random;
public class SnakePanel extends JPanel implements ActionListener{
    static final int S_Width=600;
    static final int S_Height=600;
    static final int Game_unit_size=50;
    Timer timer;
    Random random;
    int foodEaten;
    int foodX;
    int foodY;
    int bodLength = 2;
    boolean game_flag = false;
    char dir = 'R';
    static final int DELAY = 160;
    static final int G_Size=(S_Width*S_Height)/(Game_unit_size*Game_unit_size);
    final int[] x_snake=new int[G_Size];
    final int[] y_snake=new int[G_Size];
    SnakePanel(){
        this.setPreferredSize(new Dimension(S_Width,S_Height));
        this.setBackground(Color.black);
        this.setFocusable(true);
        this.addKeyListener(new MyKey());
        this.setVisible(true);
        random = new Random();
        Game_start();
    }
    public void Game_start() {
        newFoodPosition();
        game_flag=true;
        timer=new Timer(DELAY,this);
        timer.start();
    }
    public void paintComponent(Graphics graphic) {
        super.paintComponent(graphic);
        draw(graphic);
    }
    public void draw(Graphics graphic) {
        if(game_flag){
            graphic.setColor(Color.yellow);
            graphic.fillRect(foodX, foodY,Game_unit_size,Game_unit_size);
            for(int i=0;i<bodLength;i++){
                if(i==0){
                    graphic.setColor(Color.green);
//                      graphic.fillRect(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size);
//                      graphic.drawRoundRect(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size,50,30);
                    graphic.fillOval(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size);
                }
                else{
                    graphic.setColor(new Color(50,180,0));
//                      graphic.fillRect(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size);
//                      graphic.drawRoundRect(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size,50,30);
                    graphic.fillOval(x_snake[i],y_snake[i],Game_unit_size,Game_unit_size);
                }
            }
            graphic.setColor(Color.blue);
            graphic.setFont(new Font("Ink Free",Font.BOLD,40));
            FontMetrics font_me=getFontMetrics(graphic.getFont());
            graphic.drawString("Score:"+foodEaten,(S_Width-font_me.stringWidth("Score:"+foodEaten))/2,graphic.getFont().getSize());
        }
        else{
            gameOver(graphic);
        }
    }
    public void move() {
        for(int i=bodLength;i>0;i--){
            x_snake[i]=x_snake[i-1];
            y_snake[i]=y_snake[i-1];
        }
        switch (dir) {
            case 'U':
                y_snake[0]=y_snake[0]-Game_unit_size;
                break;
            case 'L':
                x_snake[0] = x_snake[0] - Game_unit_size;
                break;
            case 'D':
                y_snake[0] = y_snake[0] + Game_unit_size;
                break;
            case 'R':
                x_snake[0] = x_snake[0] + Game_unit_size;
                break;
        }
    }
    public void newFoodPosition() {
        foodX=random.nextInt((int)(S_Width/Game_unit_size))*Game_unit_size;
        foodY=random.nextInt((int)(S_Height/Game_unit_size))*Game_unit_size;
    }
    public void food_EatenOrNot() {
        if((x_snake[0]==foodX)&&(y_snake[0]==foodY)){
            bodLength++;
            foodEaten++;
            newFoodPosition();
        }
    }
    public void checkHit() {
        for (int i=bodLength;i>0;i--)
        {if((x_snake[0]==x_snake[i])&&(y_snake[0]==y_snake[i]))
        {game_flag=false;}}
        if(x_snake[0]<0)
        {game_flag=false;}
        if(x_snake[0]>S_Width)
        { game_flag=false;}
        if (y_snake[0] < 0)
        {    game_flag = false;}
        if (y_snake[0] > S_Height)
        {    game_flag = false;}
        if(!game_flag)
        {  timer.stop();}
    }
    public void gameOver(Graphics graphic) {// When ever game is over this
function will be called.
        graphic.setColor(Color.red);
        graphic.setFont(new Font("Ink Free", Font.BOLD, 40));
        FontMetrics font_me = getFontMetrics(graphic.getFont());
        graphic.drawString("Score:" + foodEaten, (S_Width -
font_me.stringWidth("Score:" + foodEaten)) / 2,
                graphic.getFont().getSize());
        graphic.setColor(Color.red);
        graphic.setFont(new Font("Ink Free", Font.BOLD, 75));
        FontMetrics font_me2 = getFontMetrics(graphic.getFont());
        graphic.drawString("Game Over", (S_Width - font_me2.stringWidth("Game
Over")) / 2,
                S_Height/2);
        graphic.setColor(Color.red);
        graphic.setFont(new Font("Ink Free", Font.BOLD, 40));
        FontMetrics font_me3 = getFontMetrics(graphic.getFont());
        graphic.drawString("Press R to Replay", (S_Width -
font_me3.stringWidth("Press R to Replay")) / 2, S_Height / 2-150);
    }
    public class MyKey extends KeyAdapter{
        @Override
        public void keyPressed(KeyEvent e) {
            switch (e.getKeyCode()) {
                case KeyEvent.VK_LEFT:
                    if(dir!='R'){
                        dir='L';
                    }
                    break;
                case KeyEvent.VK_UP:
                    if (dir != 'D') {
                        dir = 'U';
                    }
                    break;
                case KeyEvent.VK_RIGHT:
                    if (dir != 'L') {
                        dir = 'R';
                    }
                    break;
                case KeyEvent.VK_DOWN:
                    if (dir != 'U') {
                        dir = 'D';
                    }
                    break;
                case KeyEvent.VK_R:
                    if(!game_flag){
                        foodEaten=0;
                        bodLength=2;
                        dir='R';
                        Arrays.fill(x_snake,0);
                        Arrays.fill(y_snake,0);
                        Game_start();
                    }
                    break;
            }
        }
    }

    @Override
    public void actionPerformed(ActionEvent arg0) {
        if (game_flag) {
            move();
            food_EatenOrNot();
            checkHit();
        }
        repaint();
    }

}
```

snakeframe.java

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

public class  SnakeFrame extends JFrame {
    SnakeFrame(){
        this.add(new SnakePanel());
        this.setTitle("Snake");
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.setResizable(false);
        this.pack();
        this.setVisible(true);
        this.setLocationRelativeTo(null);
    }
}
```

snakegame.java

```
public class SnakeGame {
    public static void main(String[] args) throws Exception {
        new SnakeFrame();

    }
}
```