package snakegame;

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.util.Random;

public class GamePanel extends JPanel implements ActionListener , KeyListener{

private int [] snakexlength = new int [750];

private int [] snakeylength = new int [750];

private int lengthOfsnake = 3 ;

int xPos [] = {25,50,75,100,125,150,175,200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600,625,650,675,700,725,750,775,800};

int yPos [] = {75,100,125,150,175,200,225,250,275,300,325,350,375,400,425,450,475,500,525,550,575,600,625};

private Random random = new Random();

private int enemyX,enemyY;

private boolean left = false ;

private boolean right = true;

private boolean up = false ;

private boolean down = false ;

private int moves = 0 ;

private int score = 0 ;

private boolean gameOver = false ;

private ImageIcon snaketitle = new ImageIcon(getClass().getResource("snaketitle.jpg"));

private ImageIcon leftmouth = new ImageIcon(getClass().getResource("leftmouth.png"));

private ImageIcon rightmouth = new ImageIcon(getClass().getResource("rightmouth.png"));

private ImageIcon upmouth = new ImageIcon(getClass().getResource("upmouth.png"));

private ImageIcon downmouth = new ImageIcon(getClass().getResource("downmouth.png"));

private ImageIcon snakeimage = new ImageIcon(getClass().getResource("snakeimage.png"));

private ImageIcon enemy = new ImageIcon(getClass().getResource("enemy.png"));

private Timer timer ;

private int delay = 110 ;

GamePanel() {

addKeyListener(this);

setFocusable(true);

setFocusTraversalKeysEnabled(true);

timer=new Timer(delay,this);

timer.start();

newEnemy();

}

@Override

public void paint(Graphics g) {

super.paint(g);

g.setColor(Color.WHITE);

g.drawRect(24, 10, 851, 55);

g.drawRect(24, 74, 851, 576);

snaketitle.paintIcon(this , g , 25 , 11 );

g.setColor(Color.DARK\_GRAY);

g.fillRect(25,75,850,575);

if (moves == 0 ) {

snakexlength[0]=100;

snakexlength[1]=75;

snakexlength[2]=50;

snakeylength[0]=100;

snakeylength[1]=100;

snakeylength[2]=100;

}

if ( left ){

leftmouth.paintIcon(this,g,snakexlength[0],snakeylength[0]);

}

if ( right ){

rightmouth.paintIcon(this,g,snakexlength[0],snakeylength[0]);

}

if ( up ){

upmouth.paintIcon(this,g,snakexlength[0],snakeylength[0]);

}

if ( down ){

downmouth.paintIcon(this,g,snakexlength[0],snakeylength[0]);

}

for(int i = 1 ; i < lengthOfsnake ; i++){

snakeimage.paintIcon(this ,g ,snakexlength[i],snakeylength[i]);

}

enemy.paintIcon(this,g,enemyX,enemyY);

if(gameOver){

g.setColor(Color.WHITE);

g.setFont(new Font("Arial",Font.BOLD,50));

g.drawString("Game Over ",300,300);

g.setFont(new Font("Arial",Font.PLAIN,20));

g.drawString("Press ENTER To Restart ",300,350);

}

g.setColor(Color.WHITE);

g.setFont(new Font("Arial",Font.PLAIN,14));

g.drawString("Score "+score,750,30);

g.drawString("Length "+lengthOfsnake,750,50);

g.dispose();

}

@Override

public void actionPerformed(ActionEvent e) {

for(int i = lengthOfsnake - 1 ; i > 0 ; i--){

snakexlength[i]=snakexlength[i-1];

snakeylength[i]=snakeylength[i-1];

}

if(left){

snakexlength[0]=snakexlength[0]-25;

}

if(right){

snakexlength[0]=snakexlength[0]+25;

}

if(up){

snakeylength[0]=snakeylength[0]-25;

}

if(down){

snakeylength[0]=snakeylength[0]+25;

}

if(snakexlength[0]>850)snakexlength[0]=25;

if(snakexlength[0]<25)snakexlength[0]=850;

if(snakeylength[0]>625)snakeylength[0]=75;

if(snakeylength[0]<75)snakeylength[0]=625;

collidesWithEnemy ();

collidesWithBody();

repaint();

}

@Override

public void keyTyped(KeyEvent e) {

}

@Override

public void keyPressed(KeyEvent e) {

if(e.getKeyCode()==KeyEvent.VK\_ENTER){

restart();

}

if(e.getKeyCode()==KeyEvent.VK\_LEFT && (!right)){

left = true ;

right = false ;

up = false ;

down = false ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_RIGHT && (!left)){

left = false ;

right = true ;

up = false ;

down = false ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_UP && ( !down )){

left = false;

right = false ;

up = true ;

down = false ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_DOWN && ( ! up)){

left = false ;

right = false ;

up = false ;

down = true ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_W && (! down)){

left = false;

right = false ;

up = true ;

down = false ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_A && (! right)){

left = true;

right = false ;

up = false ;

down = false ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_S && (! up)){

left = false;

right = false ;

up = false ;

down = true ;

moves++;

}

if(e.getKeyCode()==KeyEvent.VK\_D && (! left)){

left = false;

right = true ;

up = false;

down = false ;

moves++;

}

}

@Override

public void keyReleased(KeyEvent e) {

}

private void newEnemy() {

enemyX = xPos[random.nextInt(34)];

enemyY= yPos[random.nextInt(23)];

for(int i = lengthOfsnake-1 ; i>=0 ; i--){

if(snakexlength[i]==enemyX && snakeylength[i]==enemyY){

newEnemy();

}

}

}

private void collidesWithEnemy() {

if(snakexlength[0]==enemyX&& snakeylength[0]==enemyY){

newEnemy();

lengthOfsnake++;

score++;

}

}

private void collidesWithBody () {

for ( int i = lengthOfsnake -1 ; i > 0 ; i--){

if(snakexlength[i]==snakexlength[0] && snakeylength[i]==snakeylength[0]){

timer.stop();

gameOver=true;

}

}

}

private void restart (){

gameOver = false ;

moves= 0 ;

score=0;

lengthOfsnake=3;

left=false;

right=true;

up=false;

down=false;

timer.start();

repaint();

}

}