import javax.swing.\*;

public class Main {

public static void main(String[] args) {

JFrame gioco = new JFrame("Palloni Cadenti");

gioco.setDefaultCloseOperation(3);

//Immagini

Disegna img = new Disegna(gioco);

gioco.add(img);

//dimensione finestra

gioco.pack(); //impone alla finestra le dimensioni del pannello

gioco.setLocationRelativeTo(null); //per posizionare al centro la schermata

gioco.setResizable(false); //impedimento ridimensionamento finestra

//visibilità

gioco.setVisible(true);

}

}

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.KeyEvent;

import java.awt.event.KeyListener;

import java.io.File;

public class Disegna extends JPanel {

protected Immagine sfondo, cestino;

protected ImmagineBot p1, p3, b1, b2;

private JFrame i;

protected int punteggio = 0;

protected String testo;

private int flag = 0, flag1 = 0;

private Sound suono;

public Disegna(JFrame i) {

this.setPreferredSize(new Dimension(900, 600));

this.i = i;

sfondo = new Immagine("sfondo.jpg", 0,0, 900, 600, true);

cestino = new Immagine("cestino.png", 410, 420, 85, 85, true);

p1 = new ImmagineBot("p1.png", 150, -100, 40, 40, true, this, 30);

b1 = new ImmagineBot("bomba.jpg", 350, -100, 40, 40, true, this, 30);

p3 = new ImmagineBot("p3.png", 550, -100, 40,40,true, this, 30);

b2 = new ImmagineBot("bomba.jpg", 750, -100, 40, 40, true, this, 30);

testo = "PUNTEGGIO: 0";

i.addKeyListener(new KeyListener() {

@Override

public void keyTyped(KeyEvent e) {}

//tasto premuto

@Override

public void keyPressed(KeyEvent e) {

int pressed = e.getKeyCode();

//spostamento a destra

if (pressed == KeyEvent.VK\_D || pressed == KeyEvent.VK\_RIGHT) {

cestino.spostaOrrizzontale(20);

}

//spostamento a sinistra

if (pressed == KeyEvent.VK\_A || pressed == KeyEvent.VK\_LEFT) {

cestino.spostaOrrizzontale(-20);

}

}

//tasto rilasciato

@Override

public void keyReleased(KeyEvent e) {}

});

}

//moficia testo

public void setTesto(String testo) {

this.testo = testo;

repaint();

}

public void paint(Graphics g) {

this.sfondo.disegna(g);

this.p1.disegna(g);

this.b1.disegna(g);

this.p3.disegna(g);

this.b2.disegna(g);

this.cestino.disegna(g);

//settaggio scritta

g.setColor(Color.BLACK);

g.setFont(new Font("Verdana", Font.BOLD, 25));

g.drawString(testo, 675, 580);

//canzone

if ((punteggio > -5 && punteggio < 10) && flag == 0) {

suono = new Sound(new File("Massive-Bass.wav"), 0);

flag = 1;

}

//settaggio situazione vinto/perso

if (punteggio >= 10) { //vittoria

sfondo = new Immagine("vinto.jpg",0,0,900,600,true);

this.sfondo.disegna(g);

if (flag1 == 0) {

suono.ferma();

suono = new Sound(new File("vinto.wav"), 0);

flag1 = 1;

}

}

else if (punteggio <= -5) { //perdita

sfondo = new Immagine("perso.jpg",0,0,900,600,true);

this.sfondo.disegna(g);

if (flag1 == 0) {

suono.ferma();

suono = new Sound(new File("perso.wav"), 0);

flag1 = 1;

}

}

}

}

import javax.swing.\*;

import java.awt.\*;

import java.util.Random;

public class Immagine {

private Image img;

private int x, y, lunghezza, altezza;

private boolean visible;

//--------------------------------------------------------------------------------------------

//METODO COMPLETO

public Immagine(String nomeImmagine, int x, int y, int lunghezza, int altezza, boolean visible) {

this.img = new ImageIcon(nomeImmagine).getImage();

this.x = x;

this.y = y;

this.lunghezza = lunghezza;

this.altezza = altezza;

this.visible = visible;

}

//--------------------------------------------------------------------------------------------

//GETTER

public Image getImg() {

return img;

}

public int getX() {

return x;

}

public int getY() {

return y;

}

public int getLunghezza() {

return lunghezza;

}

public int getAltezza() {

return altezza;

}

public boolean isVisible() {

return visible;

}

//--------------------------------------------------------------------------------------------

//SETTER

public void setImg(Image img) {

this.img = img;

}

public void setX(int x) {

this.x = x;

}

public void setY(int y) {

this.y = y;

}

public void setLunghezza(int lunghezza) {

this.lunghezza = lunghezza;

}

public void setAltezza(int altezza) {

this.altezza = altezza;

}

public void setVisible(boolean visible) {

this.visible = visible;

}

//--------------------------------------------------------------------------------------------

//METODO PER DISEGNARE

public void disegna(Graphics g) {

g.drawImage(img, x, y, lunghezza, altezza, null);

}

//--------------------------------------------------------------------------------------------

//SPOSTA GENERALE

public void sposta(int dx, int dy) {

this.x = dx;

this.y = dy;

}

//SPOSTA GENERALE

public void spostaOrrizzontale(int dx) {

this.x += dx;

}

//SPOSTA GENERALE

public void spostaVerticale(int dy) {

this.y += dy;

}

//--------------------------------------------------------------------------------------------

//GENERAZIONE CASUALE X

public int genX() {

Random random = new Random();

return(random.nextInt(600)+1);

}

}

import javax.swing.\*;

import java.awt.\*;

public class ImmagineBot extends Immagine implements Runnable{

private Disegna palestra;

private int velocità;

private Thread t;

public ImmagineBot(String nomeImg, int x, int y, int lunghezza, int altezza, boolean visible, Disegna palestra, int velocità) {

super(nomeImg, x, y, lunghezza, altezza, visible);

this.palestra = palestra;

this.velocità = velocità;

t = new Thread(this);

t.start();

}

@Override

public void run() {

try {

Thread.sleep(4000);

while((this.getY() < 500) && (palestra.punteggio > -5) && (palestra.punteggio < 10)) {

this.spostaVerticale(this.velocità);

try {

Thread.sleep(100);

}

catch (InterruptedException e) {

e.printStackTrace();

}

SwingUtilities.invokeLater(() -> palestra.repaint());

//ricaduta pallone

if (this.getY() >= 500) {

this.setY(-100);

this.setX(genX());

}

//collisione

if (collision()) {

this.setY(-100);

this.setX(genX());

}

}

}

catch (InterruptedException e) {

throw new RuntimeException(e);

}

}

public boolean collision() {

Rectangle RectCestino = new Rectangle(palestra.cestino.getX(), palestra.cestino.getY(), palestra.cestino.getLunghezza(), palestra.cestino.getAltezza());

Rectangle RectObj = new Rectangle(this.getX(), this.getY(), this.getLunghezza(), this.getAltezza());

if (RectCestino.intersects(RectObj)) {

if (this == palestra.p1) {

palestra.punteggio += 1;

System.out.println("-- COLLISISONE CON PALLONE DA BASKET --\n" +

"PUNTEGGIO: " + palestra.punteggio + "\n");

palestra.setTesto("PUNTEGGIO: " + palestra.punteggio);

}

else if (this == palestra.b1) {

palestra.punteggio -= 1;

System.out.println("-- COLLISISONE CON BOMBA --\n" +

"PUNTEGGIO: " + palestra.punteggio + "\n");

palestra.setTesto("PUNTEGGIO: " + palestra.punteggio);

}

else if (this == palestra.p3) {

palestra.punteggio += 3;

System.out.println("-- COLLISISONE CON PALLONE DA PALLAVOLO --\n" +

"PUNTEGGIO: " + palestra.punteggio + "\n");

palestra.setTesto("PUNTEGGIO: " + palestra.punteggio);

}

else if (this == palestra.b2) {

palestra.punteggio -= 2;

System.out.println("-- COLLISISONE CON PALLONE DA BASKET --\n" +

"PUNTEGGIO: " + palestra.punteggio + "\n");

palestra.setTesto("PUNTEGGIO: " + palestra.punteggio);

}

return true;

}

else {

return false;

}

}

}

import java.io.File;

import java.io.IOException;

import javax.sound.sampled.AudioFileFormat;

import javax.sound.sampled.AudioFormat;

import javax.sound.sampled.AudioInputStream;

import javax.sound.sampled.AudioSystem;

import javax.sound.sampled.Clip;

import javax.sound.sampled.DataLine;

import javax.sound.sampled.LineUnavailableException;

import javax.sound.sampled.UnsupportedAudioFileException;

public class Sound extends Thread{

AudioFileFormat aff;

AudioInputStream ais;

int n;

static boolean attivo = false;

private Clip clip;

public Sound (File sf, int n) {

aff=null;

ais=null;

try {

aff=AudioSystem.getAudioFileFormat(sf);

ais=AudioSystem.getAudioInputStream(sf);

}

catch (IOException | UnsupportedAudioFileException ex) {

ex.printStackTrace();

}

this.n = n;

this.start();

}

public void ferma() {

if (clip != null && clip.isRunning()) {

clip.stop();

}

interrupt();

}

public void run(){

try {

AudioFormat af=aff.getFormat();

DataLine.Info info = new DataLine.Info(Clip.class,ais.getFormat(),((int) ais.getFrameLength() \*af.getFrameSize()));

clip = (Clip) AudioSystem.getLine(info);

clip.open(ais);

clip.loop(n) ;

clip.start();

} catch (LineUnavailableException | IOException ex) {

ex.printStackTrace();

}

}

}