**PAINT BALL GAME**

# Team Members :

1. En.no. :- 180303105104 Name :- NAIYA JANI V.

**Abstract :**In these project I have made a paint ball game .In which these are many variables like gravitation, velocity, speed, distance, points .here at initial state the ball is at initial position and when we press the down arrow key of the keyboard the the ball will move down and then generates the velocity and then it will go upwards and then it will makes the points by colliding with the 4 balls which are known as point balls .if the ball collides with the bigger ball then it will have less points and if it collides with the smaller ball then it will have less points and the points will be displayed on the score lable which is located on the top most of the page and in case if the ball looses means the ball fall off from the panel then it will reset again and comes to its original position and again it could start from starting point. The ball will gain velocity with the velocity constant given in the code.i have used sin and cos function for defining the size of the balls. I have used the right and left key for moving the paddles for that I have used a function called key adaptor.

**Code :**

import javax.swing.JFrame;

import javax.swing.JPanel;

public class MainForGraphics1

{

public static void main(String[] args)

{

JFrame frame = new JFrame();

frame.setSize(1920,1080);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

Panel00 panel = new Panel00();

frame.add(panel);

frame.setVisible(true);

}

}

[2 slide]

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

public class Panel00 extends JPanel {

Timer t = new Timer(1, new Listener());

int ctr = 0;

double G = 0.1; //Gravitational Constant

final int xpos = 280;

double[] p2d = {280, 200};

double[] v2d = {0, 0};

int points = 0;

int lives = 0;

int sides = 13;

double snorm = 400;

double sd = 450;

double sv = 0;

boolean setlock = false;

boolean rdown, ldown;

double paddle = 130;

double rtheta = 0;

double ltheta = 0;

int preset[][] = {

{0, 400, 135, 450,1}, //right paddle

{135, 450, 270, 400,1}, //left paddle

{270, 0, 300, 20, 1}, //first bouncey thingy

{291, 0, 291, 500, 1}, //right wall

{-1, 0, 270, 0, 1}, //top wall

{0, -1, 0, 500, 1} //left wall

};

int[][] balls = {

{80, 80, 30, 50},

{230, 280, 20, 200},

{50, 200, 25, 100},

{200, 100, 10, 500}

};

int lines[][] = new int[100][5];

public Panel00(){

super();

t.start();

addKeyListener(new Key());

setFocusable(true);

for(int i = 0; i < preset.length; i++){

lines[i] = preset[i];

}

int plen = preset.length;

int ct = 0;

for(int k = 0; k < balls.length; k++){

int px = balls[k][0], py = balls[k][1], radius = balls[k][2];

for(double i = 0; i < 2 \* Math.PI; i+= 2 \* Math.PI/ sides){

ct++;

lines[plen + ct][0] = px + (int) (radius \* Math.cos(i));

lines[plen + ct][1] = py + (int) (radius \* Math.sin(i));

lines[plen + ct][2] = px + (int) (radius \* Math.cos(i - 2 \*Math.PI / sides));

lines[plen + ct][3] = py + (int) (radius \* Math.sin(i - 2 \* Math.PI / sides));

}

}

}

private class Listener implements ActionListener {

public void actionPerformed(ActionEvent e){

repaint();

}

}

public void paintComponent(Graphics g){

super.paintComponent(g);

v2d[1] += G;

p2d[1] += v2d[1];

p2d[0] += v2d[0];

if(p2d[1] > 1000){

p2d[0] = 280;

p2d[1] = 200;

v2d[0] = 0;

v2d[1] = 0;

lives++;

}

if(p2d[0] == 280 && p2d[1] > sd){

p2d[1] = sd;

v2d[1] = Math.min(v2d[1], sv);

}

if(setlock == false){

sv \*= 0.95; //the dampening coefficient for the springiness

sv -= (sd - snorm)/30;

sd += sv;

}

double rc = 0.1;

if(rdown){

rtheta = Math.max(-0.5, rtheta - rc);

}else{

rtheta = Math.min(0.5, rtheta + rc);

}

if(ldown){

ltheta = Math.max(-0.5, ltheta - rc);

}else{

ltheta = Math.min(0.5, ltheta + rc);

}

lines[0][2] = lines[0][0] + (int) (Math.cos(ltheta) \* paddle);

lines[0][3] = lines[0][1] + (int) (Math.sin(ltheta) \* paddle);

lines[1][0] = lines[1][2] + (int) (-Math.cos(rtheta) \* paddle);

lines[1][1] = lines[1][3] + (int) (Math.sin(rtheta) \* paddle);

int rX = (int) p2d[0];

int rY = (int) p2d[1];

int r = 10;

g.setColor(Color.blue);

g.drawArc(rX - r, rY - r, 2 \* r, 2 \* r, 0, 360);

g.setColor(Color.black);

for(int i = 0; i < lines.length; i++){

int x1 = lines[i][0],

y1 = lines[i][1],

x2 = lines[i][2];

double y2 = lines[i][3] + 0.0001;

if(i > preset.length){

g.setColor(Color.red);

}

g.drawLine(x1, y1, x2, (int) Math.round(y2));

double bmag = Math.sqrt(v2d[0] \* v2d[0] + v2d[1] \* v2d[1]);

double lineslope = ((double)(x2 - x1))/((double)(y2 - y1));

double ballslope = v2d[0] / v2d[1];

//System.out.println(ballslope + " " + lineslope);

//xpos \* ballslope + p2d[1] = xpos \* lineslope + y1;

double binter = p2d[0] - ballslope \* p2d[1];

double linter = x1 - lineslope \* y1;

double y = (binter - linter)/(lineslope - ballslope);

double sx = y \* ballslope + binter;

//double qx = y \* lineslope + linter;

double la = Math.atan2(y2 - y1, x2 - x1);

double ba = Math.atan2(v2d[1], v2d[0]);

double da = 2 \* la - ba;

//System.out.println(sx + " " + y);

/\*

g.drawLine((int)sx, (int)y, (int)p2d[0], (int)p2d[1]);

g.fillRect((int)sx - 2, (int)y - 2, 4, 4);

g.drawLine((int)p2d[0], (int)p2d[1], (int) (p2d[0] + Math.cos(da) \* 100), (int)(p2d[1] + Math.sin(da) \* 100));

//\*/

if(sx >= Math.min(x2, x1) && sx <= Math.max(x1, x2) &&

Math.min(y1, y2) <= y && Math.max(y1, y2) >= y){

double interdist = Math.sqrt(Math.pow(sx - p2d[0],2) + Math.pow(y - p2d[1],2));

double tiny = 0.0001;

double futuredist = Math.sqrt(Math.pow(sx - (p2d[0] + Math.cos(ba) \* tiny),2) + Math.pow(y - (p2d[1] + Math.sin(ba) \* tiny),2));

if(interdist <= bmag + r && futuredist < interdist){

//System.out.println("Carl Sagan" + i); //this is a pun because he wrote a book called Contact

if(i > preset.length){

int ball = (int) Math.floor((i - preset.length)/sides);

//System.out.println(balls[ball][2]);

points += balls[ball][3] \* bmag;

}

v2d[0] = Math.cos(da) \* bmag;

v2d[1] = Math.sin(da) \* bmag;

}

}

}

g.setColor(Color.black);

//System.out.println(sx + " " + qx);

//System.out.println(ballslope + " " + lineslope);

//double slope = Math.atan2(v2d[1], v2d[0]);

//g.drawLine((int) p2d[0], (int) p2d[1], (int) (p2d[0]+10\*v2d[0]), (int) (p2d[1]+10\*v2d[1]));

g.fillRect(xpos - 5, (int)sd + 10, 10, 20);

g.drawString("Score: " + points + " Resets: " + lives, 10, 15);

}

private class Key extends KeyAdapter {

public void keyPressed(KeyEvent e){

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

setlock = true;

sd += 2;

}

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

ldown = true;

}

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

rdown = true;

}

}

public void keyReleased(KeyEvent e){

setlock = false;

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

ldown = false;

}

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

rdown = false;

}

}

}

}

# Output :

# 