CIE 3 MINI PROJECT

1. Merwin pinto 202100102

28. Anurag Basu 202101301

45.Suraj Suratkar 202202063

48. Danish Patel 202202297

51. Neel Shah 202202421

//CONTROLLER

package game;

public class Controller implements java.awt.event.KeyListener, java.awt.event.MouseListener, java.awt.event.MouseMotionListener{

Window w;

Player p;

java.util.Vector<Enemy> e;

java.util.Vector<Shot> s;

javax.swing.JFrame f;

java.awt.Point mouse\_pos;

int health;

int shaking;

int level;

int xp;

int levelup;

int score;

int cooldown;

int waittime;

boolean pause;

Controller(javax.swing.JFrame f){

mouse\_pos = new java.awt.Point(0, 0);

w = new Window(this);

p = new Player(this);

e = new java.util.Vector<Enemy>();

s = new java.util.Vector<Shot>();

health = 450;

shaking = 0;

level = 1;

xp = 0;

levelup = 0;

score = 0;

cooldown = 0;

waittime = 0;

this.f = f;

this.f.getContentPane().add(w, java.awt.BorderLayout.CENTER);

this.f.pack();

this.f.addMouseListener(this);

this.f.addMouseMotionListener(this);

this.f.addKeyListener(this);

run();

}

void run(){

while(true){

if(pause){

w.repaint();

try{

Thread.sleep(100);

}catch(Exception e){}

continue;

}

if(cooldown > 0){

cooldown--;

}

if(Math.random() < ((0.002\*level\*level)/10)+0.0001\*waittime && !p.dead){

e.add(new Enemy(this));

waittime = 0;

if(Math.random() < 0.66){

e.get(e.size()-1).x = (Math.random() < 0.5? -50: 550);

e.get(e.size()-1).y = (int)(Math.random()\*400);

}else{

e.get(e.size()-1).x = (int)(Math.random()\*400);

e.get(e.size()-1).y = -50;

}

e.get(e.size()-1).health = 5\*level\*level;

}else{

op

waittime++;

}

p.run();

for(int i = 0; i < e.size(); i++){

e.get(i).run();

if(e.get(i).dead){

if(e.get(i).y > 500){

e.remove(i);

i--;

}

}

}

for(int i = 0; i < s.size(); i++){

s.get(i).run();

if(s.get(i).x > 500 || s.get(i).y > 500 || s.get(i).x < 0 || s.get(i).y < 0){

s.remove(i);

i--;

}else if(!s.get(i).friendly){

int xdiff = (p.x+p.cox)-(s.get(i).x);

int ydiff = (p.y+p.coy)-(s.get(i).y);

if(Math.abs(xdiff) < 50 && Math.abs(ydiff) < 50){

s.remove(i);

i--;

p.damage = 20;

health -= 25;

shaking = 20;

if(health <= 0){

p.dead = true;

p.vy = -10;

p.vx = 0;

}

}

}else{

for(int j = 0; j < e.size(); j++){

if(!e.get(j).dead){

int xdiff = (e.get(j).x+e.get(j).cox)-(s.get(i).x);

int ydiff = (e.get(j).y+e.get(j).coy)-(s.get(i).y);

if(Math.abs(xdiff) < 50 && Math.abs(ydiff) < 50){

s.remove(i);

i--;

e.get(j).damage = 20;

e.get(j).health -= 10;

if(e.get(j).health <= 0){

e.get(j).dead = true;

e.get(j).vy = -10;

score += 250\*level;

xp++;

if(xp > level\*6){

xp -= level\*6;

level++;

levelup = 150;

score += 10000\*level;

}

}

break;

}

}

}

}

}

w.repaint();

try{

Thread.sleep(20);

}catch(Exception e){}

}

}

public void keyReleased(java.awt.event.KeyEvent e){

if(!p.dead){

if(e.getKeyCode() == 65 || e.getKeyCode() == 68){

p.vx = 0;

}

}

}

public void keyPressed(java.awt.event.KeyEvent e){

if(!p.dead){

if(e.getKeyCode() == 65){

p.vx = -4;

}else if(e.getKeyCode() == 68){

p.vx = 4;

}else if(e.getKeyCode() == 32){

if(p.y == 400-p.h){

p.y -= 10;

p.vy = -10;

}

}else if(e.getKeyCode() == 80){

if(pause){

pause = false;

}else{

pause = true;

}

}

}

}

public void mouseReleased(java.awt.event.MouseEvent e){

if(p.y > 2000){

if(e.getButton() == e.BUTTON1){

if(mouse\_pos.x > 150 && mouse\_pos.x < 350 && mouse\_pos.y > 300 && mouse\_pos.y < 350){

System.exit(0);

}

}

}

}

public void mousePressed(java.awt.event.MouseEvent e){

if(!p.dead && !pause){

if(e.getButton() == e.BUTTON1){

int xdiff = mouse\_pos.x-(p.x+p.cox);

int ydiff = mouse\_pos.y-(p.y+p.coy);

double length = Math.sqrt(xdiff\*xdiff+ydiff\*ydiff);

s.add(new Shot(p.x+p.cox, p.y+p.coy, (int)(xdiff\*(5/length)), (int)(ydiff\*(5/length)), true));

}else if(e.getButton() == e.BUTTON3){

if(cooldown == 0){

for(int i = 0; i < 360; i += 360/40){

s.add(new Shot(p.x+p.cox, p.y+p.coy, (int)(Math.cos(i)\*10), (int)(Math.sin(i)\*10), true));

}

cooldown = 450-level\*15;

}

}

}

}

public void mouseMoved(java.awt.event.MouseEvent e){

mouse\_pos.setLocation(e.getX(), e.getY());

}

public void mouseDragged(java.awt.event.MouseEvent e){

mouse\_pos.setLocation(e.getX(), e.getY());

}

}

//ENEMY

package game;

public class Enemy extends Player{

int health;

public Enemy(Controller parent){

super(parent);

gp1 = new java.awt.GradientPaint(x, y, new java.awt.Color(255, 50, 0), x, y+h, new java.awt.Color(150, 50, 0), true);

x = 0;

y = 0;

health = 1;

}

public int target(int i){

if(i == 0){

return parent.p.x+parent.p.cox;

}else{

return parent.p.y+parent.p.coy;

}

}

public void run(){

if(dead){

vy += 0.2;

y += vy;

}else{

if(parent.p.dead){

return;

}

int xdiff = (parent.p.x+parent.p.cox)-(x+cox);

int ydiff = (parent.p.y+parent.p.coy)-(y+coy);

if(xdiff > 0){

x += 2;

}else if(xdiff < 0){

x -= 2;

}

if(ydiff > 0){

y += 2;

}else if(ydiff < 0){

y -= 2;

}

if(Math.abs(xdiff) < 50 && Math.abs(ydiff) < 50){

parent.health--;

parent.shaking = 10;

parent.p.damage = 10;

if(parent.health <= 0){

parent.p.dead = true;

parent.p.vy = -10;

parent.p.vx = 0;

}

}

if(Math.random() < ((0.002\*parent.level\*parent.level)/10)+0.005 && !parent.p.dead){

xdiff = (parent.p.x+parent.p.cox)-(x+cox);

ydiff = (parent.p.y+parent.p.coy)-(y+coy);

double length = Math.sqrt(xdiff\*xdiff+ydiff\*ydiff);

parent.s.add(new Shot(x+cox, y+coy, (int)(xdiff\*(5/length)), (int)(ydiff\*(5/length)), false));

}

}

}

}

//PLAYER

package game;

public class Player{

public int x, y, w, h, cox, coy;

public double vx, vy;

java.awt.GradientPaint gp1;

java.awt.geom.Ellipse2D e1, e2, e3;

java.awt.geom.Rectangle2D r1;

java.awt.BasicStroke bs1;``

java.awt.geom.Line2D l1, l2, l3, l4, l5, l6;

Controller parent;

int damage;

boolean dead;

public Player(Controller parent){

x = 200;

y = 100;

w = 75;

h = 75;

cox = 30;

coy = 45;

damage = 0;

this.parent = parent;

gp1 = new java.awt.GradientPaint(x, y, new java.awt.Color(255, 200, 0), x, y+h, new java.awt.Color(255, 75, 0), true);

e1 = new java.awt.geom.Ellipse2D.Double(0, 0, w, h);

e2 = new java.awt.geom.Ellipse2D.Double(0, 0, 20, 30);

e3 = new java.awt.geom.Ellipse2D.Double(0, 0, 8, 8);

r1 = new java.awt.geom.Rectangle2D.Double(-25, -10, 75, 25);

bs1 = new java.awt.BasicStroke(2);

l1 = new java.awt.geom.Line2D.Double(0, 0, 40, 20);

l2 = new java.awt.geom.Line2D.Double(0, 20, 40, 0);

l3 = new java.awt.geom.Line2D.Double(0, 20, 20, 0);

l4 = new java.awt.geom.Line2D.Double(0, 0, 20, 20);

l5 = new java.awt.geom.Line2D.Double(20, 20, 40, 0);

l6 = new java.awt.geom.Line2D.Double(20, 0, 40, 20);

}

public int target(int i){

if(i == 0){

return parent.mouse\_pos.x;

}else{

return parent.mouse\_pos.y;

}

}

public void paint(java.awt.Graphics2D g){

int tx = x, ty = y, tcox = cox, tcoy = coy;

g.translate(tx, ty);

g.setPaint(gp1);

g.fill(e1);

g.setStroke(bs1);

g.setPaint(java.awt.Color.RED);

g.draw(e1);

g.translate(17, 10);

if(dead){

g.draw(l3);

g.draw(l4);

g.draw(l5);

g.draw(l6);

}else if(damage > 0){

damage--;

g.draw(l1);

g.draw(l2);

}else{

g.setPaint(java.awt.Color.WHITE);

g.fill(e2);

g.setPaint(java.awt.Color.BLACK);

g.draw(e2);

g.translate(6, 10);

g.fill(e3);

g.translate(-6, -10);

g.translate(22, 0);

g.setPaint(java.awt.Color.WHITE);

g.fill(e2);

g.setPaint(java.awt.Color.BLACK);

g.draw(e2);

g.translate(6, 10);

g.fill(e3);

g.translate(-6, -10);

g.translate(-22, 0);

}

g.translate(-17, -10);

g.translate(tcox, tcoy);

double xdiff = target(0)-(tx+tcox);

double ydiff = target(1)-(ty+tcoy);

if(ydiff != 0){

g.rotate(Math.atan(ydiff/xdiff));

if(xdiff < 0){

g.rotate(Math.PI);

}

}

g.setPaint(java.awt.Color.GRAY);

g.fill(r1);

g.setPaint(java.awt.Color.BLACK);

g.draw(r1);

if(ydiff != 0){

g.rotate(-Math.atan(ydiff/xdiff));

if(xdiff < 0){

g.rotate(-Math.PI);

}

}

g.translate(-tcox, -tcoy);

g.translate(-tx, -ty);

}

public void run(){

if(dead){

vy += 0.2;

}else{

if(y+h < 400){

vy += 0.2;

}else if(y+h > 400){

vy = 0;

y = 400-h;

}

}

x += Math.ceil(vx);

y += Math.ceil(vy);

if(x > 500-h){

x = 500-h;

}else if(x < 0){

x = 0;

}

}

}

//SHOT

package game;

public class Shot{

public int x, y;

double vx, vy;

boolean friendly;

java.awt.geom.Ellipse2D e;

public Shot(int x, int y, int vx, int vy, boolean friendly){

this.x = x;

this.y = y;

this.vx = vx;

this.vy = vy;

this.friendly = friendly;

e = new java.awt.geom.Ellipse2D.Double(-8, -8, 16, 16);

}

public void paint(java.awt.Graphics2D g){

g.translate(x, y);

if(friendly){

g.setPaint(java.awt.Color.YELLOW);

}else{

g.setPaint(java.awt.Color.ORANGE);

}

g.fill(e);

g.translate(-x, -y);

}

public void run(){

x += vx;

y += vy;

}

}

//WINDOW

package game;

public class Window extends javax.swing.JPanel{

Controller parent;

java.awt.GradientPaint gp1, gp2;

java.awt.geom.Rectangle2D r1, r2, r3, r4;

java.awt.BasicStroke bs1;

java.awt.geom.Line2D s1, s2;

java.awt.Font f1, f2;

java.awt.geom.RoundRectangle2D rr1, rr2;

public Window(Controller c){

parent = c;

setPreferredSize(new java.awt.Dimension(500, 500));

gp1 = new java.awt.GradientPaint(0, 0, new java.awt.Color(50, 200, 255), 0, 400, java.awt.Color.BLACK);

gp2 = new java.awt.GradientPaint(0, 100, new java.awt.Color(150, 255, 150), 0, 500, new java.awt.Color(50, 200, 50));

r1 = new java.awt.geom.Rectangle2D.Double(0, 0, 500, 400);

r2 = new java.awt.geom.Rectangle2D.Double(0, 400, 500, 100);

r3 = new java.awt.geom.Rectangle2D.Double(0, 0, 450, 50);

r4 = new java.awt.geom.Rectangle2D.Double(0, 0, 450, 5);

bs1 = new java.awt.BasicStroke(1);

s1 = new java.awt.geom.Line2D.Double(-16, 0, 16, 0);

s2 = new java.awt.geom.Line2D.Double(0, -16, 0, 16);

f1 = new java.awt.Font("Arial", 0, 40);

f2 = new java.awt.Font("Arial", 0, 20);

rr1 = new java.awt.geom.RoundRectangle2D.Double(100, 100, 300, 300, 10, 10);

rr2 = new java.awt.geom.RoundRectangle2D.Double(150, 300, 200, 50, 10, 10);

}

public void paint(java.awt.Graphics g){

update(g);

}

public void update(java.awt.Graphics g2){

java.awt.Graphics2D g = (java.awt.Graphics2D) g2;

g.setRenderingHint(java.awt.RenderingHints.KEY\_ANTIALIASING, java.awt.RenderingHints.VALUE\_ANTIALIAS\_ON);

if(parent.shaking > 0){

parent.shaking--;

g.translate(Math.random()\*5, Math.random()\*5);

}

g.setPaint(gp1);

g.fill(r1);

g.setPaint(gp2);

g.fill(r2);

for(int i = 0; i < parent.e.size(); i++){

parent.e.get(i).paint(g);

}

parent.p.paint(g);

for(int i = 0; i < parent.s.size(); i++){

parent.s.get(i).paint(g);

}

if(!parent.p.dead){

g.translate(25, 25);

r3.setRect(0, 0, parent.health, 25);

if(parent.health > 225){

g.setPaint(java.awt.Color.RED);

}else if(parent.health > 100){

g.setPaint(java.awt.Color.RED);

}else{

g.setPaint(java.awt.Color.RED);

}

g.fill(r3);

g.setPaint(java.awt.Color.RED);

g.draw(r3);

if(parent.cooldown > 0){

g.translate(0, 25);

g.setPaint(java.awt.Color.RED);

r4.setRect(0, 0, parent.cooldown, 5);

g.fill(r4);

g.setPaint(java.awt.Color.RED);

g.draw(r4);

g.translate(0, -25);

}

g.translate(225, 33);

g.setFont(f2);

java.awt.FontMetrics fm = g.getFontMetrics();

g.drawString("Score: "+parent.score+" Level: "+parent.level, -fm.stringWidth("Score: "+parent.score+" : Level: "+parent.level)/2, -fm.getHeight()/2);

g.translate(-225, -37);

g.translate(-25, -25);

}

if(parent.levelup > 0){

double rot = (Math.random()-0.5)/3;

if(parent.levelup < 50){

g.translate(250, 250-(-parent.levelup+50)\*5);

}else{

g.translate(250, 250);

}

g.setFont(f1);

java.awt.FontMetrics fm = g.getFontMetrics();

g.rotate(rot);

g.drawString("LEVEL UP", -fm.stringWidth("LEVEL UP")/2, -fm.getHeight());

g.rotate(-rot);

if(parent.levelup < 50){

g.translate(-250, -(250-(-parent.levelup+50)\*5));

}else{

g.translate(-250, -250);

}

parent.levelup--;

}

if(parent.p.y > 2000){

parent.p.vy = 0;

parent.p.y = 2001;

g.setPaint(java.awt.Color.RED);

g.fill(rr1);

g.setPaint(java.awt.Color.RED);

g.draw(rr1);

g.setFont(f1);

java.awt.FontMetrics fm = g.getFontMetrics();

g.drawString("GAME OVER!", 250-fm.stringWidth("GAME OVER!")/2, 175-fm.getHeight()/2);

g.drawString(""+parent.score, 250-fm.stringWidth(""+parent.score)/2, 275-fm.getHeight()/2);

g.setFont(f2);

fm = g.getFontMetrics();

g.drawString("Final Score:", 250-fm.stringWidth("final Score:")/2, 220-fm.getHeight()/2);

g.drawString("Quit", 250-fm.stringWidth("Quit")/2, 345-fm.getHeight()/2);

g.draw(rr2);

}

if(parent.pause){

g.setPaint(java.awt.Color.WHITE);

g.fill(rr2);

g.setPaint(java.awt.Color.BLACK);

g.draw(rr2);

g.setFont(f2);

java.awt.FontMetrics fm = g.getFontMetrics();

g.drawString("Pause", 250-fm.stringWidth("Pause")/2, 345-fm.getHeight()/2);

}

g.translate(parent.mouse\_pos.x, parent.mouse\_pos.y);

g.setStroke(bs1);

g.setPaint(java.awt.Color.RED);

g.draw(s1);

g.draw(s2);

}

}

//MAIN

package game;

public class Main{

public static void main(String[] args){

javax.swing.JFrame f = new javax.swing.JFrame("Java spil");

f.setVisible(true);

f.setDefaultCloseOperation(f.EXIT\_ON\_CLOSE);

f.setResizable(false);

int[] pixels = new int[16 \* 16];

java.awt.Image image = java.awt.Toolkit.getDefaultToolkit().createImage(new java.awt.image.MemoryImageSource(16, 16, pixels, 0, 16));

java.awt.Cursor transparentCursor = java.awt.Toolkit.getDefaultToolkit().createCustomCursor(image, new java.awt.Point(0, 0), "invisiblecursor");

f.setCursor(transparentCursor);

f.getContentPane().setLayout(new java.awt.BorderLayout());

f.setAlwaysOnTop(true);

new Controller(f);

}

}

OUTPUT :  




