**III. Xây dựng chương trình**

**1.Khởi tạo trò chơi**

Lớp FlappyBirds là lớp dùng để khởi tạo chò trơi. Lớp này có giao diện đồ họa dễ dàng sử dụng. Ở giao diện đồ họa này người chơi biết biết được cách thức điều khiển chú chim, số điểm đã đạt được sau khi đã chơi trước đó và hoàn thành trò chơi. Người chơi có thể khởi tạo trò chơi mới chỉ một nút nháy. Giao điện bằng tiếng Việt, chỉ cần dùng một nút đơn giản là người chơi có thể tự mình tham gia trò chơi.

Để có được giao diện và thực hiện những hành động đó lớp FlappyBirds đã kế thừa từ lớp GameScreen và lắng nghe sự kiện Animation.

public void GAME\_UPDATE(long deltaTime){

if(CurrentScreen == BEGIN\_SCREEN){

restGame();

}else if(CurrentScreen == GAMEPLAY\_SCREEN){

if(bird.getLive()) birds\_anim.Update\_Me(deltaTime);

Lớp GameScreen sẽ lấy các thiết bị đồ họa trên một môi trường tương ứng với từng máy tính khác nhau. Với mỗi một máy tính khác nhau nó sẽ có 15 môi trường đồ họa khác nhau. Trình biên dịch của Java sẽ ánh xạ sang từng máy để nó trả về môi trường cụ thể. Lớp AFrameOnImage được gọi sẽ khởi tao ra một cửa sổ giao diện mà hình. Nó sẽ lấy độ phân giải cụ thể của từng máy sau đó sẽ trả về một giao diện đầy màn hình trên từng máy

public FlappyBirds(){

super(800,600);

try {

birds = ImageIO.read(new File("Assets/bird\_sprite.png"));

} catch (IOException ex) {}

birds\_anim = new Animation(70);

AFrameOnImage f;

f = new AFrameOnImage(0, 0, 60, 60);

birds\_anim.AddFrame(f);

f = new AFrameOnImage(60, 0, 60, 60);

birds\_anim.AddFrame(f);

f = new AFrameOnImage(120, 0, 60, 60);

birds\_anim.AddFrame(f);

f = new AFrameOnImage(60, 0, 60, 60);

birds\_anim.AddFrame(f);

bird = new Bird(350, 250, 50, 50);

ground = new Ground();

chimneyGroup = new ChimneyGroup();

BeginGame();

}

Graphics2D sẽ trả về giá trị thông qua lớp GameScreen, nó sẽ lấy đồ họa và hiển thị toàn màn hình. Phương thức GAME\_PAINT sẽ thức hiện liên tục và được quản lí thông

public void GAME\_PAINT(Graphics2D g2) {

g2.setColor(Color.decode("#b8daef"));

g2.fillRect(0, 0, MASTER\_WIDTH, MASTER\_HEIGHT);

chimneyGroup.paint(g2);

ground.Paint(g2);

if(bird.getIsFlying())

birds\_anim.PaintAnims((int)bird.getPosX(), (int)bird.getPosY(), birds, g2, 0, -1);

else

birds\_anim.PaintAnims((int)bird.getPosX(), (int)bird.getPosY(), birds, g2, 0, 0);

if(CurrentScreen == BEGIN\_SCREEN){

g2.setColor(Color.red);

g2.drawString("ẤN ĐỂ CHƠI", 200, 300);

}

if(CurrentScreen == GAMEOVER\_SCREEN){

g2.setColor(Color.white);

g2.drawString("ẤN ĐỂ THỬ LẠI", 200, 300);

g2.drawString("Điểm cao nhất: "+Point, 200, 350);

}

g2.setColor(Color.red);

g2.drawString("Point: "+Point, 20, 50);

}

Sử dụng phương thức KEY\_ACTION để nhấn mạnh các hành động

public void KEY\_ACTION(KeyEvent e, int Event) {

if(Event == KEY\_PRESSED){

if(CurrentScreen == BEGIN\_SCREEN){

CurrentScreen = GAMEPLAY\_SCREEN;

}else if(CurrentScreen == GAMEPLAY\_SCREEN){

if(bird.getLive()) bird.fly();

}else if(CurrentScreen == GAMEOVER\_SCREEN){

CurrentScreen = BEGIN\_SCREEN;

}

}

}

}

**2. Bắt đầu trò chơi.**

Lớp GameScreen được khởi tạo sẽ truyền vào một biến có kiểu dữ liệu là image. Ảnh được truyền vào chính là đại diện cho chú chim có mà người chơi sẽ thấy xuất hiện trên màn hình. Lớp Transport có các phương thức như: update(long elapsedTime), getX(), getY(), setX(float x), setY(float y), getWidth(), getHeight(), getImage(),... Sẽ dùng để lấy các tọa độ, tộc độ và gán tọa độ, tộc độ cho con chim đó. Riêng phương thức update(long elapsedTime)sẽ cập nhật lại tọa độ của phương tiện giao thông đó theo một khoảng thời gian.

public void GAME\_UPDATE(long deltaTime){

if(CurrentScreen == BEGIN\_SCREEN){

restGame();

}else if(CurrentScreen == GAMEPLAY\_SCREEN){

if(bird.getLive()) birds\_anim.Update\_Me(deltaTime);

bird.update(deltaTime);

ground.Update();

chimneyGroup.update();

for(int i = 0;i<chimneyGroup.SIZE;i++){

if(bird.getRect().intersects(chimneyGroup.getChimney(i).getRect())){

if(bird.getLive()) bird.bupSound.play();

bird.setLive(false);

System.out.println("Set live = false");

}

}

for(int i = 0;i<chimneyGroup.SIZE;i++){

if(bird.getPosX() > chimneyGroup.getChimney(i).getPosX()&& !chimneyGroup.getChimney(i).getIsBehindBird()&& i%2 == 0){

Point ++;

bird.PointSound.play();

chimneyGroup.getChimney(i).setIsBehindBird(true);

}

}

if(bird.getPosY() + bird.getH() > ground.getYground()) CurrentScreen = GAMEOVER\_SCREEN;

}else{

}

}

Lớp Bird sẽ cung cấp các hình, âm thanh, vận tốc cũng như các hành động của chú chim,

được kế thừa từ lớp Objects.

public class Bird extends Objects{

private float vt = 0;

private boolean isFlying = false;

private Rectangle rect;

private boolean isLive = true;

SoundPlayer flapSound;

SoundPlayer bupSound;

SoundPlayer PointSound;

public Bird(int x, int y, int w, int h){

super(x, y, w, h);

rect = new Rectangle(x, y, w, h);

flapSound = new SoundPlayer(new File("Assets/fap.wav"));

bupSound = new SoundPlayer(new File("Assets/fall.wav"));

PointSound = new SoundPlayer(new File("Assets/getpoint.wav"));

}

public void setLive(boolean b){

isLive = b;

}

public boolean getLive(){

return isLive;

}

public Rectangle getRect(){

return rect;

}

public void setVt(float vt){

this.vt = vt;

}

public void update(long deltaTime){

vt+=FlappyBirds.g;

this.setPosY(this.getPosY()+vt);

this.rect.setLocation((int)this.getPosX(),(int) this.getPosY());

if(vt<0) isFlying = true;

else isFlying = false;

}

public void fly(){

vt = -3;

flapSound.play();

}

public boolean getIsFlying(){

return isFlying;

}

}

Lớp Chimney sẽ cung cấp hình ảnh, kích thước của các vật cản và được kế thừa từ lớp Objects.

public class Chimney extends Objects{

private Rectangle rect;

private boolean isBehindBird = false;

public Chimney(int x, int y, int w, int h){

super(x, y, w, h);

rect = new Rectangle(x, y, w, h);

}

public void update(){

setPosX(getPosX()-2);

rect.setLocation((int)this.getPosX(),(int)this.getPosY());

}

public Rectangle getRect(){

return rect;

}

public void setIsBehindBird(boolean b){

isBehindBird = b;

}

public boolean getIsBehindBird(){

return isBehindBird;

}

}

Lớp chimneyGroup được kế thừa thông qua lớp QueueList và cung cấp hình ảnh của một tập hợp vật cản một cách ngẫu nhiên và lặp lại liên tục

public class ChimneyGroup {

private QueueList<Chimney> chimneys;

private BufferedImage chimneyImage, chimneyImage2;

public static int SIZE = 6;

private int topChimneyY = -300;

private int bottomChimneyY = 300;

public Chimney getChimney(int i){

return chimneys.get(i);

}

public int getRandomY(){

Random random = new Random();

int a;

a = random.nextInt(10);

return a\*35;

}

public ChimneyGroup(){

try {

chimneyImage = ImageIO.read(new File("Assets/chimney.png"));

chimneyImage2 = ImageIO.read(new File("Assets/chimney\_.png"));

} catch (IOException ex) {}

chimneys = new QueueList<Chimney>();

Chimney cn;

for(int i =0;i<SIZE/2;i++){

int deltaY = getRandomY();

cn = new Chimney(830+i\*300, bottomChimneyY + deltaY, 74, 400 );

chimneys.push(cn);

cn = new Chimney(830+i\*300, topChimneyY + deltaY, 74, 400);

chimneys.push(cn);

}

}

public void restChimney(){

chimneys = new QueueList<Chimney>();

Chimney cn;

for(int i =0;i<SIZE/2;i++){

int deltaY = getRandomY();

cn = new Chimney(830+i\*300, bottomChimneyY + deltaY, 74, 400 );

chimneys.push(cn);

cn = new Chimney(830+i\*300, topChimneyY + deltaY, 74, 400 );

chimneys.push(cn);

}

}

public void update(){

for(int i=0;i<SIZE;i++){

chimneys.get(i).update();

}

if(chimneys.get(0).getPosX()<-74){

int deltaY = getRandomY();

Chimney cn;

cn = chimneys.pop();

cn.setPosX(chimneys.get(4).getPosX()+300);

cn.setPosY(bottomChimneyY + deltaY);

cn.setIsBehindBird(false);

chimneys.push(cn);

cn = chimneys.pop();

cn.setPosX(chimneys.get(4).getPosX());

cn.setPosY(topChimneyY + deltaY);

cn.setIsBehindBird(false);

chimneys.push(cn);

}

}

public void paint(Graphics2D g2){

for(int i=0;i< SIZE;i++)

if(i%2==0)

g2.drawImage(chimneyImage,(int) chimneys.get(i).getPosX(),(int) chimneys.get(i).getPosY(), null);

else

g2.drawImage(chimneyImage2,(int) chimneys.get(i).getPosX(),(int) chimneys.get(i).getPosY(), null);

}

}

Lớp Ground để tạo viền đất và di chuyển ngược chiều chú chim

public class Ground {

private BufferedImage groundImage;

private int x1, y1, x2, y2;

public Ground(){

try {

groundImage = ImageIO.read(new File("Assets/ground.png"));

} catch (IOException ex) {}

x1 = 0;

y1 = 500;

x2 = x1 + 830;

y2 = 500;

}

public void Update(){

x1-= 2;

x2-= 2;

if(x2 < 0) x1 = x2 + 830;

if(x1 < 0) x2 = x1 + 830;

}

public void Paint(Graphics2D g2){

g2.drawImage(groundImage, x1, y1, null);

g2.drawImage(groundImage, x2, y2, null);

}

public int getYground(){

return y1;

}

}