

EXTENSÃO: Semana Acadêmica de Eng. de Computação - 2025

ROTEIRO

Minicurso de Desenvolvimento de Jogos em Java



Aluno responsável: Gabriel Victor Kaizer de Carvalho
Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues
Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho

Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues

Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

1 - Entendendo o Game Loop e capturando as entradas do computador:



2 - criando e movendo o personagem do jogador:

Classe Personagem:

```
import java.awt.Color;
import java.awt.Graphics;

public class Personagem {
    protected Jogo jogo;
    protected Color color;
    protected float x, y;
    protected int width, height;
    protected int speed;
    protected boolean up, down, left, right;

    public Personagem(Jogo jogo, Color color, float x, float y, int width, int height, int
speed) {
        this.jogo = jogo;
        this.color = color;
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
        this.speed = speed;
    }

    public void update() {
        this.move();
    }

    public void desenhar(Graphics g) {
        g.setColor(color);
        g.fillRect(Math.round(x), Math.round(y), width, height);
    }

    getters e setters...
}
```

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho

Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues

Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Movendo o personagem:

Na classe personagem

```
public void move() {
    float saveX = this.x;
    float saveY = this.y;

    float realSpeed = this.speed;
    if((this.right || this.left) && (this.up || this.down)){
        realSpeed = this.speed/((float) Math.sqrt(2));
    }

    if(up){
        this.y -= realSpeed;
    }
    if(down){
        this.y += realSpeed;
    }

    if(!this.jogo.colisao(this)){
        this.y = saveY;
    }

    if(left){
        this.x -= realSpeed;
    }
    if(right){
        this.x += realSpeed;
    }

    if(!jogo.colisao(this)){
        this.x = saveX;
    }
}
```

Na classe do jogo

```
public void keyPressed(KeyEvent e) {
    switch(e.getKeyCode()) {
        case KeyEvent.VK_W -> this.jogador.setUp(true);
        case KeyEvent.VK_S -> this.jogador.setDown(true);
        case KeyEvent.VK_A -> this.jogador.setLeft(true);
        case KeyEvent.VK_D -> this.jogador.setRigth(true);
    }
}

public void keyReleased(KeyEvent e) {
    switch(e.getKeyCode()) {
        case KeyEvent.VK_W -> this.jogador.setUp(false);
        case KeyEvent.VK_S -> this.jogador.setDown(false);
        case KeyEvent.VK_A -> this.jogador.setLeft(false);
        case KeyEvent.VK_D -> this.jogador.setRigth(false);
    }
}
```

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho
Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues
Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Impedir o personagem de sair da tela:

```
public boolean colisao(Personagem personagem) {  
    if(personagem.getX() < 0 || personagem.getY() < 0 || (personagem.getX() +  
personagem.getWidth()) > Programa.screenWidth || (personagem.getY() + personagem.getHeigth())  
> Programa.screenHeigth){  
        return false;  
    }  
    return true;  
}
```

3 - criando os inimigos e fazendo eles perseguirem o jogador:

Classe Inimigo:

```
import java.awt.Color;  
  
public class Inimigo extends Personagem{  
  
    public Inimigo(Jogo jogo, Color color, int x, int y, int width, int height, int speed) {  
        super(jogo, color, x, y, width, height, speed);  
    }  
  
    @Override  
    public void update(){  
        Personagem jogador = this.jogo.getJogador();  
  
        this.reset();  
  
        if(jogador.getX() > this.x){  
            this.right = true;  
        }else if(jogador.getX() < this.x){  
            this.left = true;  
        }  
  
        if(jogador.getY() > this.y){  
            this.down = true;  
        }else if(jogador.getY() < this.y){  
            this.up = true;  
        }  
  
        this.move();  
    }  
}
```

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho
Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues
Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Cria o inimigo em algum canto da tela aleatório:

```
public void novoInimigo() {  
    Random random = new Random();  
    int rx, ry;  
    int width = 32;  
    int height = 32;  
    int canto = random.nextInt(4);  
    if(canto == 0 || canto == 3){  
        rx = 5;  
    }else{  
        rx = Programa.screenWidth - 5 - width;  
    }  
    if(canto == 0 || canto == 1){  
        ry = 5;  
    }else{  
        ry = Programa.screenHeight - 5 - height;  
    }  
    this.inimigos.add(new Inimigo(this, Color.red, rx, ry, width, height, 2));  
}
```

Chamar a função no Game Loop:

```
now = System.currentTimeMillis();  
if(now - timer >= 1000){  
    timer = now;0  
    sec++;  
    if(sec >= 60){  
        sec = 0;  
        min++;  
    }  
    this.novoInimigo();  
}
```

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho

Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues

Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Função para identificar a colisão entre o jogador e os inimigos (será chamada no fim de cada update):

```
public void check(){
    for(int i = 0; i < this.inimigos.size(); i++){
        if((this.inimigos.get(i).getX() >= this.jogador.getX() && this.inimigos.get(i).getX()
<= (this.jogador.getX() + this.jogador.getWidth())) || ((this.inimigos.get(i).getX() +
this.inimigos.get(i).getWidth()) >= this.jogador.getX() && (this.inimigos.get(i).getX() +
this.inimigos.get(i).getWidth()) <= (this.jogador.getX() + this.jogador.getWidth()))){

            if((this.inimigos.get(i).getY() >= this.jogador.getY() &&
this.inimigos.get(i).getY() <= (this.jogador.getY() + this.jogador.getHeight())) ||
((this.inimigos.get(i).getY() + this.inimigos.get(i).getHeight()) >= this.jogador.getY() &&
(this.inimigos.get(i).getY() + this.inimigos.get(i).getHeight()) <= (this.jogador.getY() +
this.jogador.getHeight()))){

                this.running = false;

            }

        }

    }
}
```

4 - Criando e disparando projéteis:

Classe Projétil:

```
import java.awt.Color;

public class Projétil extends Personagem{
    private float vx, vy;

    public Projétil(Jogo jogo, Color color, float x, float y, int width, int height, float vx,
float vy) {
        super(jogo, color, x, y, width, height, 0);

        this.vx = vx;
        this.vy = vy;
    }

    @Override
    public void move(){
        this.x += vx;
        this.y += vy;

        if(!this.jogo.colisao(this)){
            this.jogo.excluirProjétil(this);
        }
    }
}
```

Minicurso de Desenvolvimento de Jogos em Java SAEC-2025

Aluno responsável: Gabriel Victor Kaizer de Carvalho

Professor Colaborador SAEC-2025: Elder de Oliveira Rodrigues

Professor Coordenador SAEC-2025: Rodrigo Gaiba de Oliveira

Criando e definindo a velocidade de um Projétil:

```
public void mousePressed(MouseEvent e){
    float projetilSpeed = 7;
    float dx = e.getX() - this.jogador.getX();
    float dy = e.getY() - this.jogador.getY();
    float distancia = (float) Math.sqrt(dx*dx + dy*dy);
    float tempo = distancia/projetilSpeed;
    float vx = dx/tempo;
    float vy = dy/tempo;
    this.projeteis.add(new Projetil(this, Color.green, this.jogador.getX() +
    this.jogador.getWidth()/2, this.jogador.getY() + this.jogador.getHeigth()/2, 4, 4, vx, vy));
}
```

Excluindo projéteis para não sobrecarregar o jogo:

```
public void excluirProjetil(Personagem personagem){
    for(int i = 0; i < this.projeteis.size(); i++){
        if(this.projeteis.get(i) == personagem){
            this.projeteis.remove(i);
        }
    }
}
```

Atualizando o check para que os projéteis possam destruir os inimigos:

```
        this.running = false;
    }
}
for(int j = 0; j < this.projeteis.size(); j++){
    if((this.projeteis.get(j).getX() + this.projeteis.get(j).getWidth()/2) >=
this.inimigos.get(i).getX() && (this.projeteis.get(j).getX() +
this.projeteis.get(j).getWidth()/2) <= (this.inimigos.get(i).getX() +
this.inimigos.get(i).getWidth()) && (this.projeteis.get(j).getY() +
this.projeteis.get(j).getHeigth()/2) >= this.inimigos.get(i).getY() &&
(this.projeteis.get(j).getY() + this.projeteis.get(j).getHeigth()/2) <=
(this.inimigos.get(i).getY() + this.inimigos.get(i).getHeigth())){

        this.inimigos.remove(i);
        this.projeteis.remove(j);
        j = this.projeteis.size();
    }
}
}
```