

```

1 package juego;
2
3 import java.awt.Color;
4
5
6
7
8
9
10 public class Tigre {
11
12     private double x;
13     private double y;
14     private double ancho;
15     private double alto;
16     private double angulo;
17     private double escala;
18     private Color color;
19     private Image imagen;
20     private boolean perdioVida;
21
22     public Tigre(double x, double y) {
23         this.x = x;
24         this.y = y;
25         this.ancho = 80;
26         this.alto = 60;
27         this.angulo = 0;
28         this.escala = 0.4;
29         this.color = Color.magenta;
30         this.imagen = Herramientas.cargarImagen("tigre1.gif");
31         this.perdioVida = false;
32     } // Arbol
33
34     public boolean chocaConPiedra(Piedra[] piedra) {
35         for (int i = 0; i < piedra.length; i++) {
36             // si piedra no es null y tigre choque con piedra, esta ultima se vuelve null y retorna true
37             if (piedra[i] != null && piedra[i].getX() - piedra[i].getDiametro() / 2 < x + ancho / 2
38                 && x - ancho / 2 < piedra[i].getX() + piedra[i].getDiametro() / 2 &&
39                 piedra[i].getY() < y + alto / 2 && y - alto / 2 < piedra[i].getY())
40             {
41                 piedra[i] = null;
42                 return true;
43             }
44         }
45         return false;
46     }
47
48
49     public static void agregaTigre(Tigre[] t, Entorno e) {
50         Random random = new Random();
51         double max = e.ancho();
52
53         for (int i = 0; i < t.length; i++) {
54             // si el tigre no es null y el eje x de tigre se pasa del entorno, max se vuelve el eje de x del tigre
55             if (t[i] != null && t[i].x > max) {
56                 max = t[i].x;
57             }
58         }
59
60         for (int i = 0; i < t.length; i++) {
61             // si el tigre es null se crea un nuevo tigre a partir de un random limitado sumado a max
62             if (t[i] == null) {
63                 int rand = random.nextInt(100, 200);
64                 t[i] = new Tigre(rand + max, 500);
65                 return;
66             }
67         }
68
69     } // agregarTigre
70
71     public void dibujarTigre(Entorno e) {
72         e.dibujarRectangulo(this.x, this.y, this.ancho, this.alto, this.angulo, this.color);
73         e.dibujarImagen(imagen, x + 30, y - 20, angulo, escala);
74     } // dibujarTigre
75
76     public boolean saleDePantalla() {
77         // cuando el eje x pasa a ser mayor a -100 devuelve true
78         if (this.x < -100) {
79             return true;
80         } else {
81             return false;
82         }
83     }

```

```
83     }
84
85     public void desplazar(double v) {
86         this.x -=v;
87     }// desplazar
88
89     public double getX() {
90         return x;
91     }
92
93     public double getY() {
94         return y;
95     }
96
97     public double getAncho() {
98         return ancho;
99     }
100
101     public double getAlto() {
102         return alto;
103     }
104
105     public boolean isPerdioVida() {
106         return perdioVida;
107     }
108
109     public void setPerdioVida(boolean perdioVida) {
110         this.perdioVida = perdioVida;
111     }
112
113 }// class Tigre
```