

```

1 package juego;
2
3 import java.awt.Color;
4
5
6
7
8
9
10 public class Serpiente {
11
12     private double x;
13     private double y;
14     private double ancho;
15     private double alto;
16     private double angulo;
17     private Color color;
18     private double escala;
19     private Image imagen;
20     private boolean perdioVida;
21
22     public Serpiente(double x, double y) {
23         this.x = x;
24         this.y = y;
25         this.ancho = 40;
26         this.alto = 50;
27         this.angulo = 0;
28         this.color = Color.CYAN;
29         this.escala = .3;
30         this.imagen = Herramientas.cargarImagen("serpiente.gif");
31         this.perdioVida = false;
32     } // Arbol
33
34
35     public static Serpiente[] agregaSerpiente(Serpiente[] s, Arbol[] arbol) {
36         Random random = new Random();
37
38         for (int i = 0; i < s.length; i++) {
39             // elijo un arbol aleatorio disponible
40             int rand = random.nextInt(arbol.length);
41
42             // se debe elegir una arbol q no este en pantalla para lograr colocar serpiente
43             while (arbol[rand].getX() < 800) {
44                 rand = random.nextInt(arbol.length);
45             }
46             if (s[i] == null && !arbol[rand].isConSerpiente()) {
47                 // se suma 30 para q la serpiente quede al borde de la rama, y se resta 25 para
48                 // que quede centrada en el arbol
49                 s[i] = new Serpiente(arbol[rand].getX() + 30, arbol[rand].getY() - 25);
50                 arbol[rand].setConSerpiente(true);
51             }
52         }
53         return s;
54     }
55
56     public boolean chocaConPiedra(Piedra[] piedra) {
57         for (int i = 0; i < piedra.length; i++) {
58
59             if (piedra[i] != null && piedra[i].getX() - piedra[i].getDiametro() / 2 < x + ancho /
60                 // cuando las dimensiones de nuestro objeto choquen con el objeto pasado como
61                 // parametro y ese mismo objeto no sea nulo, retornamos true
62                 && x - ancho / 2 < piedra[i].getX() + piedra[i].getDiametro() / 2 && piedra[i].getY() < y +
63                 alto / 2
64                 && y - alto / 2 < piedra[i].getY()) {
65                 piedra[i] =
66                     null; // volvemos nula
67                     la piedra iterada
68                 return true;
69             }
70         }
71         return false;
72     }
73
74     public void dibujarSerpiente(Entorno e) {
75         e.dibujarImagen(imagen, x, y, angulo, escala); // graficamos en el entorno el objeto
76     } // dibujarSerpiente
77
78     public boolean saleDePantalla() {
79         // si el eje X es menos a -100 retorna true sino retorna false
80         if (this.x < -100) {
81             return true;
82         } else {
83
84         }
85     }
86 }

```

```
78         return false;
79     }
80 }
81
82 public void desplazar(double v) {    // restamos al eje X el valor pasado como parametro
83     this.x -= v;
84 }// desplazar
85
86
87 public double getX() {
88     return x;
89 }
90
91 public double getY() {
92     return y;
93 }
94
95 public double getAncho() {
96     return ancho;
97 }
98
99 public double getAlto() {
100     return alto;
101 }
102
103 public boolean isPerdioVida() {
104     return perdioVida;
105 }
106
107 public void setPerdioVida(boolean perdioVida) {
108     this.perdioVida = perdioVida;
109 }
110
111 }// class Serpiente
```