

## Viga.java

```
1 package juego;
2
3 import java.awt.Color;
4
5
6
7
8
9 public class Viga {
10
11     private int pos;
12     private double x;
13     private double y;
14     private double largo;
15     private double alto;
16
17     /*
18      * Este constructor, ya tiene definida de forma estricta y estática las
19      * posiciones de las vigas
20      */
21
22     public Viga(int pos) {
23
24         switch (pos) {
25             case 1:
26                 this.x = 400;
27                 this.y = 575;
28                 this.largo = 820;
29                 this.alto = 25;
30                 break;
31             case 2:
32                 this.x = 325;
33                 this.y = 475;
34                 this.largo = 700;
35                 this.alto = 25;
36                 break;
37             case 3:
38                 this.x = 475;
39                 this.y = 375;
40                 this.largo = 700;
41                 this.alto = 25;
42                 break;
43             case 4:
44                 this.x = 325;
45                 this.y = 275;
46                 this.largo = 700;
47                 this.alto = 25;
48                 break;
49             case 5:
50                 this.x = 475;
51                 this.y = 175;
52                 this.largo = 700;
53                 this.alto = 25;
54                 break;
55             case 6:
56                 this.x = 325;
57                 this.y = 75;
58                 this.largo = 700;
59                 this.alto = 25;
60                 break;
61         }
62
63         this.pos = pos;
64
65     }
66 }
```

## Viga.java

```
67  /*
68  * Dibujar
69  *
70  * Esta función debe ser llamada en cada tick por cada viga que exista. Por cada
71  * tick dibuja la viga.
72  *
73  * La construcción gráfica de la viga tiene un condimento especial. Para dar la
74  * sensación de que es una estructura metálica, se dibuja un rectángulo de color
75  * rojo de fondo, y sobre el de forma estratégica, triángulos del mismo color
76  * que el fondo, en juegos de a dos. Cada uno invertido 90º con respecto al
77  * anterior.
78  *
79  */
80  public void dibujar(Entorno entorno) {
81
82      // Rectángulo básico de la viga, respetando los valores indicados por el
83      // constructor
84      entorno.dibujarRectangulo(this.x, this.y, this.largo, this.alto, 0.0, Color.RED);
85
86      // El extremo izquierdo de la viga corrido 10px
87      double paso = this.x - (this.largo / 2) + 10;
88
89      // Se decide que la suma de la base de un triangulo, la punta del triángulo
90      // adyacente y un espacio
91      // extra sea la 25ava parte del ancho de la viga - 4 pixeles
92      double triangulos = (this.largo / 25) - 4;
93
94      // Indica la cantidad de parejas de triangulos dibujados. Una pareja es un
95      // triangulo con la punta hacia arriba
96      // y el otro con la punta hacia abajo.
97      int dibujados = 0;
98
99      // Este bucle dibuja la pareja de triángulos a lo largo de la viga.
100     while (dibujados <= triangulos) {
101
102         entorno.dibujarTriangulo(paso, this.y, 21, 21, Herramientas.radians(90),
java.awt.Color.BLACK);
103         paso += 14;
104         entorno.dibujarTriangulo(paso, this.y, 21, 21, Herramientas.radians(270),
java.awt.Color.BLACK);
105         paso += 14;
106         dibujados += 1;
107     }
108 }
109
110 }
111
112 // Devuelve la posX
113 public int getPosx() {
114     return (int) this.x;
115 }
116
117 // Devuelve la posY
118 public int getPosy() {
119     return (int) this.y;
120 }
121
122 // Devuelve el Ancho
123 public int getAncho() {
124     return (int) this.largo;
125 }
126
```

## Viga.java

```
127 public int getPos() {
128     return pos;
129 }
130
131 /*
132  * DondeEmpiezaElSuelo
133  *
134  * Esta función devuelve el valor en el 'y' en el cual comienza la viga.
135  * Sabiendo que el 'y' se encuentra en el centro.
136  *
137  */
138 public double dondeEmpiezaElSuelo() {
139
140     return this.y - (this.alto / 2) - 1;
141 }
142
143 public double dondeTerminaElTecho() {
144
145     return this.y + (this.alto / 2) + 1;
146 }
147
148 /*
149  * Esta función indica donde comienza la viga en el eje X.
150  */
151 public int extremoIzquierdo() {
152     return this.getPosx() - this.getAncho() / 2;
153 }
154
155 /*
156  * Esta función indica donde termina la viga en el eje X.
157  */
158 public int extremoDerecho() {
159     return this.getPosx() + this.getAncho() / 2;
160 }
161
162 }
163
```