

```

1 package juego;
2
3 import java.awt.Color;
4
5
6
7
8
9 public class Juego extends InterfaceJuego {
10
11     private Entorno entorno;
12     private Mono mono;
13     private Arbol[] arbol;
14     private Tigre[] tigre;
15     private Serpiente[] serpiente;
16     private Aguila[] aguila;
17     private Piedra[] piedra;
18     private Items items;
19     private Selva[] selva;
20     private Image gameOver;
21     private Image menuImg;
22     private int punto;
23     private int vida;
24     private boolean pause;
25     private boolean menu;
26     private int enemigosMuertos;
27     private int nivel;
28
29 //
30     public Juego() {
31
32         this.entorno = new Entorno(this, "Escape del mono - Grupo 1 - Correa A - Rolon G - Bentacor L - V0.01",
33             800,
34             600);
35         this.punto = 0;
36         this.vida = 3;
37         this.selva = new Selva[2];
38         this.mono = new Mono(0, 500);
39         this.piedra = new Piedra[3];
40         this.gameOver = Herramientas.cargarImagen("game_over.jpg");
41         this.menuImg = Herramientas.cargarImagen("menu.jpg");
42         this.items = new Items();
43         this.pause = false;
44         this.arbol = new Arbol[5];
45         this.tigre = new Tigre[2];
46         this.serpiente = new Serpiente[2];
47         this.aguila = new Aguila[2];
48         this.menu = true;
49         this.nivel = 1;
50         this.enemigosMuertos = 0;
51
52         // se crean el fondo de pantalla, los tigres, arboles y las serpientes
53         Selva.iniciaSelva(selva);
54         Arbol.crearArboles(this.arbol, entorno);
55         Tigre.agregaTigre(this.tigre, entorno);
56         Serpiente.agregaSerpiente(this.serpiente, this.arbol);
57         Aguila.agregaAguila(this.aguila, entorno);
58
59         // una vez q se cargan los datos se inicia el juego
60         this.entorno.iniciar();
61     } // juego
62
63     public void tick() {
64
65         Selva.dibujarFondo(selva, entorno);
66         // Procesamiento de un instante de tiempo.
67         if (menu) {
68
69             // se grafican las palabras en el menu
70             entorno.dibujarImagen(this.menuImg, entorno.ancho() / 2, entorno.alto() / 2, 0, 1);
71             entorno.cambiarFont(" ", 50, Color.red);
72             entorno.escribirTexto("MENÚ", 320, 50);
73             entorno.cambiarFont(" ", 26, Color.red);
74             entorno.escribirTexto("Presione [ENTER] comenzar", 235, 100);
75             // si se oprime enter entonces se sale del menu
76             if (entorno.sePresiono(entorno.TECLA_ENTER)) {
77                 menu = false;
78             }
79         } else if (vida > 0 && !pause) {
80

```

```

81 // si la piedra sale de la pantalla se crea una nueva
82 if (this.items.saleDePantallaP()) {
83     this.items.crearPiedra();
84 }
85 // si la banana sale de la pantalla se crea una nueva
86 if (this.items.saleDePantallaB()) {
87     this.items.crearBananas();
88 }
89 // si el mono agarra la banana entonces se crea una nueva, e incrementa su vida
90 // en una unidad
91 if (mono.chocaConBanana(items)) {
92     this.items.crearBananas();
93     vida += 1;
94 }
95 // si el mono agarra las piedras(item) entonces se crea una nueva, e incrementa
96 // sus municiones en 3 unidades
97 if (mono.chocaConPiedra(items)) {
98     // se crea el item piedra
99     this.items.crearPiedra();
100    // se reestablece la cantidad de municiones
101    mono.setDisparosDisp(3);
102 }
103
104 // condicionales del salto
105 if (entorno.estaPresionada(entorno.TECLA_ARRIBA)) {
106     // aumenta el contador para controlar el tiempo del salto del mono
107     mono.aumentarTimer();
108
109     // cuenta los saltos cada vez q se oprime la tecla
110     if (entorno.sePresiono(entorno.TECLA_ARRIBA)) {
111         mono.aumentarSalto();
112     }
113
114     // aumenta el salto en 1 para no realizar mas saltos en el aire
115     if (mono.contadorSalto()) {
116         mono.saltar(8);
117     } else {
118         // si se mantiene apretado se activa gravedad
119         if (!mono.chocaConSuelo() && !mono.chocaConArboles(arbol)) {
120             mono.gravedad();
121         }
122     }
123 } else {
124     mono.setTimer(0);
125     // cuando el mono esta sobre el suelo o esta en arbol entonces puede saltar
126     if (mono.chocaConSuelo() || mono.chocaConArboles(arbol)) {
127         mono.setSalto(0);
128     } else {
129         mono.gravedad();
130     }
131 }
132
133 // condiciones de los arboles
134 for (int i = 0; i < arbol.length; i++) {
135     if (arbol[i] != null) {
136
137         arbol[i].dibujarArbol(entorno);
138
139         // dependiendo del nivel cambia la velocidad de desplazamiento
140         if (nivel == 1) {
141             arbol[i].desplazar(2);
142         } else if (nivel == 2) {
143             arbol[i].desplazar(2.5);
144         } else if (nivel == 3) {
145             arbol[i].desplazar(3);
146         } else {
147             arbol[i].desplazar(3.5);
148         }
149
150         // si el mono choca con el arbol y el arbol no dio punto entonces incrementa su
151         // puntaje
152         if (mono.chocaConArbol(arbol[i]) && arbol[i].isDioPuntos() == false) {
153             punto += 5;
154             arbol[i].setDioPuntos(true);
155         }
156
157         // si el arbol sale de la pantalla entonces se anula el elemento y se crea uno

```

```

158         // nuevo
159         if (arbol[i].saleDePantalla()) {
160             arbol[i] = null;
161             Arbol.crearArboles(this.arbol, entorno);
162         }
163     }
164 } // for arboles
165
166 // condiciones de los tigres
167 for (int i = 0; i < tigre.length; i++) {
168     if (tigre[i] != null) {
169         tigre[i].dibujarTigre(entorno);
170
171         // dependiendo del nivel cambia la velocidad de desplazamiento
172         if (nivel == 1) {
173             tigre[i].desplazar(4);
174         } else if (nivel == 2) {
175             tigre[i].desplazar(5);
176         } else if (nivel == 3) {
177             tigre[i].desplazar(6);
178         } else {
179             tigre[i].desplazar(7);
180         }
181         // si el tigre sale de la pantalla entonces se anula el elemento y se crea uno
182         // nuevo
183         if (tigre[i].saleDePantalla()) {
184             tigre[i] = null;
185             Tigre.agregaTigre(tigre, entorno);
186         } else {
187
188             // si el mono choca con el tigre y el mono aun no perdio vida con este tigre
189             // entonces se resta una vida
190             if (mono.chocaConTigre(tigre[i]) && tigre[i].isPerdioVida() == false) {
191                 vida -= 1;
192                 tigre[i].setPerdioVida(true);
193             }
194
195             // si el tigre choca con la piedra se aumenta el puntaje y aumenta la cantidad
196             // de enemigos aniquilados,
197             // el tigre se pone null y se instancia uno nuevo
198             if (tigre[i].chocaConPiedra(piedra)) {
199                 punto += 5;
200                 this.enemigosMuertos++;
201                 tigre[i] = null;
202                 Tigre.agregaTigre(tigre, entorno);
203             }
204         }
205     }
206 }
207
208 // condiciones de las serpientes
209 for (int i = 0; i < serpiente.length; i++) {
210     if (serpiente[i] != null) {
211         serpiente[i].dibujarSerpiente(entorno);
212         // si el tigre sale de la pantalla entonces se anula el elemento y se crea uno
213         // nuevo
214         if (nivel == 1) {
215             serpiente[i].desplazar(2);
216         } else if (nivel == 2) {
217             serpiente[i].desplazar(2.5);
218         } else if (nivel == 3) {
219             serpiente[i].desplazar(3);
220         } else {
221             serpiente[i].desplazar(3.5);
222         }
223         // si la serpiente sale de la pantalla entonces se anula el elemento y se crea
224         // uno nuevo
225         if (serpiente[i].saleDePantalla()) {
226             serpiente[i] = null;
227             Serpiente.agregaSerpiente(this.serpiente, this.arbol);
228         } else {
229             // si el mono choca con la serpiente y ademas no perdio vida con esta serpiente
230             // entonces pierde una vida
231             if (mono.chocaConSerpiente(serpiente[i]) && serpiente[i].isPerdioVida() == false) {
232                 vida -= 1;
233                 serpiente[i].setPerdioVida(true);
234             }
235         }
236     }
237 }

```

```

235         // si la serpiente choca con la piedra entonces el mono obtiene 5 puntos e
236         // incrementa los enemigos aniquilados
237         // la serpiente se pone en null y se crea una nueva
238         if (serpiente[i].chocaConPiedra(piedra)) {
239             punto += 5;
240             this.enemigosMuertos++;
241             serpiente[i] = null;
242             Serpiente.agregaSerpiente(this.serpiente, this.arbol);
243         }
244     }
245
246     }
247 }
248 // condiciones de las aguilas
249 for (int i = 0; i < aguila.length; i++) {
250     if (aguila[i] != null && nivel > 2) {
251
252         aguila[i].dibujarAguila(entorno);
253         // si el aguila sale de la pantalla entonces se anula el elemento y se crea uno
254         // nuevo
255         if (nivel == 1) {
256             aguila[i].desplazar(6);
257         } else if (nivel == 2) {
258             aguila[i].desplazar(6.2);
259         } else if (nivel == 3) {
260             aguila[i].desplazar(6.4);
261         } else {
262             aguila[i].desplazar(6.6);
263         }
264         aguila[i].descender();
265         // si el aguila sale de la pantalla entonces se anula el elemento y se crea uno
266         // nuevo
267         if (aguila[i].saleDePantalla()) {
268             aguila[i] = null;
269             Aguila.agregaAguila(aguila, entorno);
270         } else {
271             // si el aguila choca con la piedra entonces el mono obtiene 5 puntos e
272             // incrementa los enemigos aniquilados
273             // el aguila se pone en null y se crea una nueva
274             if (mono.chocaConAguila(aguila[i]) && aguila[i].isPerdioVida() == false) {
275                 vida -= 1;
276                 aguila[i].setPerdioVida(true);
277             }
278             // si el aguila choca con la piedra entonces el mono gana puntos, incrementa los
279             // enemigos aniquilados y el aguila se coloca null y se crea uno nuevo
280             if (aguila[i].chocaConPiedra(piedra)) {
281                 punto += 10;
282                 this.enemigosMuertos++;
283                 aguila[i] = null;
284                 Aguila.agregaAguila(aguila, entorno);
285             }
286         }
287     }
288 }
289 }
290
291 // condicional de la piedra
292 for (int i = 0; i < piedra.length; i++) {
293     //si la piedra no es null entonces la dibuja y la hace desplazar
294     if (piedra[i] != null) {
295         piedra[i].dibujarPiedra(entorno);
296         piedra[i].avanzar();
297         // si la piedra sale de la pantalla entonces se coloca nula
298         if (piedra[i].saleDePantalla(entorno)) {
299             piedra[i] = null;
300         }
301     }
302 }
303 //si se presiona la tecla espacio y tiene disparos disponibles entonces realiza disparo
304 if (entorno.sePresiono(entorno.TECLA_ESPACIO)) {
305     if (mono.getDisparosDisp() > 0) {
306         Piedra.agregarPiedra(piedra, mono);
307     }
308 }
309 // si se oprimi la tecla supr entonces el juego se coloca en pause
310 if (entorno.sePresiono(entorno.TECLA_DELETE)) {
311     pause = true;

```

```

312     }
313
314     // condicionales de puntajes y niveles
315     if (punto > 120) {
316         nivel = 4;
317     } else if (punto > 80) {
318         nivel = 3;
319     } else if (punto > 30) {
320         nivel = 2;
321     }
322
323     // se grafican los objetos y las estadísticas del juego
324     mono.dibujarMono(entorno);
325     items.dibujarPiedras(entorno);
326     items.desplazarp();
327     items.dibujarBananas(entorno);
328     items.desplazarb();
329     entorno.cambiarFont(" ", 26, Color.RED);
330     entorno.escribirTexto("VIDAS: " + vida, 50, 50);
331     entorno.cambiarFont(" ", 26, Color.RED);
332     entorno.escribirTexto("NIVEL: " + nivel, 50, 575);
333     entorno.cambiarFont(" ", 26, Color.RED);
334     entorno.escribirTexto("PIEDRAS: " + mono.getDisparosDisp(), 300, 50);
335     entorno.cambiarFont(" ", 26, Color.red);
336     entorno.escribirTexto("PUNTAJE: " + punto, 600, 50);
337     entorno.cambiarFont(" ", 20, Color.red);
338     entorno.escribirTexto("Presione [DELETE] para PAUSE", 250, 590);
339 } else if (pause) {
340
341     //se grafica el mensaje "pause"
342     entorno.cambiarFont(" ", 50, Color.red);
343     entorno.escribirTexto("PAUSE", 320, 300);
344     if (entorno.sePresiono(entorno.TECLA_DELETE)) {
345         pause = false;
346     }
347
348 } else {
349     // se grafican las opciones luego de la partida junto con las estadísticas del juego
350     entorno.dibujarImagen(gameOver, entorno.ancho() / 2, entorno.alto() / 2, 0, 1.35);
351     entorno.cambiarFont(" ", 26, Color.red);
352     entorno.escribirTexto("Presione [INICIO] para reiniciar |", 15, 550);
353     entorno.cambiarFont(" ", 26, Color.red);
354     entorno.escribirTexto("PUNTAJE OBTENIDO: " + punto, 250, 50);
355     entorno.cambiarFont(" ", 26, Color.red);
356     entorno.escribirTexto("Presione [FIN] para volver al menú", 390, 550);
357     entorno.cambiarFont(" ", 26, Color.red);
358     entorno.escribirTexto("ENEMIGOS ABATIDOS: " + this.enemigosMuertos, 250, 80);
359
360     if (entorno.estaPresionada(entorno.TECLA_INICIO)) {
361
362         // reinicio de parametros iniciales
363         this.nivel = 1;
364         this.punto = 0;
365         this.vida = 3;
366         this.enemigosMuertos = 0;
367         this.mono = new Mono(0, 500);
368         this.piedra = new Piedra[3];
369         this.items = new Items();
370         this.arbol = new Arbol[5];
371         this.tigre = new Tigre[2];
372         this.aguila = new Aguila[2];
373         this.serpiente = new Serpiente[2];
374
375         Arbol.crearArboles(this.arbol, entorno);
376         Tigre.agregaTigre(this.tigre, entorno);
377         Serpiente.agregaSerpiente(this.serpiente, this.arbol);
378         Aguila.agregaAguila(aguila, entorno);
379
380     }
381
382     if (entorno.sePresiono(entorno.TECLA_FIN)) {
383
384         // reinicio de parametros iniciales
385         menu = true;
386         this.nivel = 1;
387         this.punto = 0;
388         this.vida = 3;

```

```
389         this.enemigosMuertos = 0;
390         this.mono = new Mono(0, 500);
391         this.piedra = new Piedra[3];
392         this.items = new Items();
393         this.arbol = new Arbol[5];
394         this.tigre = new Tigre[2];
395         this.aguila = new Aguila[2];
396         this.serpiente = new Serpiente[2];
397         Arbol.crearArboles(this.arbol, entorno);
398         Tigre.agregaTigre(this.tigre, entorno);
399         Serpiente.agregaSerpiente(this.serpiente, this.arbol);
400         Aguila.agregaAguila(agUILa, entorno);
401     }
402 }
403 }// fin tick()
404
405 @SuppressWarnings("unused")
406 public static void main(String[] args) {
407     Juego juego = new Juego();
408 }
409
410 }
411
```