

```

1  /*
2  * Juego de la Serpiente v4
3  * Pablo_Villa 08/11/2023
4  */
5  #include <iostream>
6  #include "terminal.h"
7  #include <cstdlib>
8  #include <ctime>
9
10 using namespace std;
11
12 const char TECLA_SIGUIENTE = ' ';
13 const char TECLA_FIN = 'F';
14 const char SERPIENTE = '@';
15 const char CUERPO_SERPIENTE = 'o';
16 const char ARRIBA = 'W';
17 const char ABAJO = 'S';
18 const char IZQUIERDA = 'A';
19 const char DERECHA = 'D';
20 const char MANZANA = 'M';
21 const char SIMBOLO_VERTICAL = '|';
22 const char SIMBOLO_INTERMEDIO = ' ';
23 const char SIMBOLO_HORIZONTAL = '-';
24 const char SIMBOLO_EXTERIOR = '+';
25 const int PREMIO = 100;
26 const int LONG_SERPIENTE = 15;
27 const int BASE = 80;
28 const int ALTURA = 22;
29 const int RETARDO = 50;
30 const int SERPIENTE_X_INICIAL = 10;
31 const int LIMITE_SUPERIOR = 1;
32 const int LIMITE_INFERIOR = 20;
33 const int LIMITE_IZQUIERDA = 2;
34 const int LIMITE_DERECHA = 78;
35 const int SERPIENTE_Y_INICIAL = 15;
36 const int MOVIMIENTO_X_DERECHA = 1;
37 const int MOVIMIENTO_Y_DESCENDENTE = 1;
38 const int MOVIMIENTO_X_IZQUIERDA = -1;
39 const int MOVIMIENTO_Y_ASCENDENTE = -1;
40 const int MAX_MANZANAS = 10;
41 const string TITULO = "Juego de la serpiente ";
42 const string VERSION = "4.0";
43 const string TECLA_CONTINUAR = "ESPACIO";
44
45 struct posicion {
46     int x = 0;
47     int y = 0;
48 };
49
50 struct inc_unitario_posicion {
51     int x = 0;
52     int y = 0;
53 };
54
55 void iniciar_pantalla_inicial();
56 void inicializar_juego(char tecla, posicion serpiente[], inc_unitario_posicion& inc_unitario_posicion);
57 void pantalla_inicial();
58 void dibujar_linea(const char c_exterior, const char c_interior, const int largo);
59 void dibujar_rectangulo(const int base, const int altura);
60 void inicializar_serpiente(posicion serpiente[], inc_unitario_posicion& inc_unitario_posicion);
61 bool juego_terminado(char tecla, posicion serpiente[]);
62 void obtener_direccion(const char tecla, inc_unitario_posicion& inc_unitario_posicion);
63 void pintar_serpiente(const posicion serpiente[]);
64 void borrar_serpiente(const posicion serpiente[]);
65 bool serpiente_tocada(const posicion serpiente[]);
66 void mover_serpiente(posicion serpiente[], inc_unitario_posicion inc_unitario_posicion);

```

```

67
68 int main() {
69     char tecla = '\0';
70     posicion serpiente[LONG_SERPIENTE];
71     inc_unitario_posicion inc_unitario_posicion = {0, 0};
72
73     srand(time(0));
74     setlocale(LC_ALL, "");
75
76     inicializar_juego(tecla,serpiente, inc_unitario_posicion);
77     while ( ! juego_terminado(tecla, serpiente)) {
78         pintar_serpiente(serpiente);
79
80         retardar(RETARDO);
81
82         borrar_serpiente(serpiente);
83
84         obtener_direccion(tecla, inc_unitario_posicion);
85         mover_serpiente(serpiente, inc_unitario_posicion);
86
87         tecla = leer_tecla();
88     }
89     deshabilitar_modos_crudo_terminal();
90     borrar_terminal();
91 }
92
93 void iniciar_pantalla_inicial(){
94     retardar(RETARDO);
95     hacer_cursor_visible(false);
96     pantalla_inicial();
97 }
98
99 void inicializar_juego(char tecla ,posicion serpiente[], inc_unitario_posicion& inc_unitario_posicion){
100     iniciar_pantalla_inicial();
101     while(leer_tecla() != TECLA_SIGUIENTE){
102         retardar(RETARDO);
103     }
104     deshabilitar_modos_crudo_terminal();
105     borrar_terminal();
106
107     inicializar_serpiente(serpiente, inc_unitario_posicion);
108
109     dibujar_rectangulo(BASE, ALTURA);
110     retardar(RETARDO);
111
112     habilitar_modos_crudo_terminal();
113     hacer_cursor_visible(false);
114     tecla = leer_tecla();
115 }
116
117 void pantalla_inicial(){
118
119     poner_cursor(1,1);
120     cout << " ***** " << endl;
121     poner_cursor(1,2);
122     cout << " * "<< TITULO << VERSION << " * " << endl;
123     poner_cursor(1,3);
124     cout << " ***** " << endl;
125     poner_cursor(1,6);
126     cout << " _____ " << endl;
127     poner_cursor(1,7);
128     cout << " _/          \\" << endl;
129     poner_cursor(1,8);
130     cout << "  \\\_         \\" << endl;
131     poner_cursor(1,9);
132     cout << "           \\\  \\\_ " << endl;

```

```

133     poner_cursor(1,10);
134     cout << "          \\\n          \\\n " << endl;
135     poner_cursor(1,11);
136     cout << "          \\\n          \\\n          _|_ " << endl;
137     poner_cursor(1,12);
138     cout << "          \\\n          0 \\\n / \\\n " << endl;
139     poner_cursor(1,13);
140     cout << "          \\\n          / \\\n \\\n / " << endl;
141     poner_cursor(1,17);
142     cout << "Pulsa la tecla de " << TECLA_CONTINUAR << " para continuar" << endl;
143 }
144
145 void inicializar_serpiente(posicion serpiente[], inc_unitario_posicion& inc_unitario_posicion) {
146     serpiente[0].x = SERPIENTE_X_INICIAL;
147     serpiente[0].y = SERPIENTE_Y_INICIAL;
148
149     inc_unitario_posicion.x = MOVIMIENTO_X_DERECHA;
150     inc_unitario_posicion.y = MOVIMIENTO_Y_ASCENDENTE;
151
152     for (int i = 1; i < LONG_SERPIENTE; i++) {
153         serpiente[i].x = serpiente[i - 1].x + 1;
154         serpiente[i].y = serpiente[i - 1].y;
155     }
156 }
157
158 void dibujar_linea(const char c_exterior, const char c_interior, const int largo){
159     cout << c_exterior;
160     for(int i = 0; i < largo - 2; i++){
161         cout << c_interior;
162     }
163     cout << c_exterior << endl;
164 }
165
166 void dibujar_rectangulo(const int base, const int altura){
167     poner_cursor(2,1);
168     cout << "+----- " << TITULO
169         << VERSION << " -----+ " << endl;
170     for (int i = 2; i < altura - 2; i++){
171         poner_cursor(2,i);
172         dibujar_linea(SIMBOLO_VERTICAL,SIMBOLO_INTERMEDIO,base);
173     }
174     poner_cursor(2,altura - 2);
175     dibujar_linea(SIMBOLO_EXTERIOR,SIMBOLO_HORIZONTAL,base);
176     poner_cursor(2, altura);
177     cout << ARRIBA << "-> Subir " << ABAJO << "-> Bajar " << IZQUIERDA
178         << "-> Izda " << DERECHA << "-> Dcha " << TECLA_FIN << "-> Fin" << endl;
179 }
180
181 bool juego_terminado(char tecla, posicion serpiente[]){
182     return(toupper(tecla) == TECLA_FIN ||
183         serpiente_tocada(serpiente) ||
184         serpiente[0].x == LIMITE_IZQUIERDA ||
185         serpiente[0].x == LIMITE_DERECHA ||
186         serpiente[0].y == LIMITE_SUPERIOR ||
187         serpiente[0].y == LIMITE_INFERIOR);
188 }
189
190 void obtener_direccion(const char tecla, inc_unitario_posicion& inc_unitario_posicion) {
191     switch (toupper(tecla)) {
192         case ARRIBA:
193             inc_unitario_posicion.x = 0;
194             inc_unitario_posicion.y = MOVIMIENTO_Y_ASCENDENTE;
195             break;
196
197         case ABAJO:
198             inc_unitario_posicion.x = 0;

```

```

199         inc_unitario_posicion.y = MOVIMIENTO_Y_DESCENDENTE;
200         break;
201
202     case IZQUIERDA:
203         inc_unitario_posicion.x = MOVIMIENTO_X_IZQUIERDA;
204         inc_unitario_posicion.y = 0;
205         break;
206
207     case DERECHA:
208         inc_unitario_posicion.x = MOVIMIENTO_X_DERECHA;
209         inc_unitario_posicion.y = 0;
210         break;
211     }
212 }
213
214 void mover_serpiente(posicion serpiente[], inc_unitario_posicion inc_unitario_posicion) {
215     posicion cabeza_anterior;
216     cabeza_anterior = serpiente[0];
217
218     serpiente[0].x = serpiente[0].x + inc_unitario_posicion.x;
219     serpiente[0].y = serpiente[0].y + inc_unitario_posicion.y;
220
221     for (int i = LONG_SERPIENTE - 1; i > 0; --i) {
222         serpiente[i] = serpiente[i - 1];
223     }
224     serpiente[1] = cabeza_anterior;
225 }
226
227
228 void pintar_serpiente(const posicion serpiente[]) {
229     poner_cursor(serpiente[0].x, serpiente[0].y);
230     cout << SERPIENTE;
231
232     for (int i = 1; i < LONG_SERPIENTE - 1; i++) {
233         poner_cursor(serpiente[i].x, serpiente[i].y);
234         cout << CUERPO_SERPIENTE;
235     }
236 }
237
238 void borrar_serpiente( const posicion serpiente[]) {
239     for (int i = 0; i < LONG_SERPIENTE - 1; i++) {
240         poner_cursor(serpiente[i].x, serpiente[i].y);
241         cout << " ";
242     }
243 }
244
245 bool serpiente_tocada(const posicion serpiente[]) {
246     for (int i = 1; i < LONG_SERPIENTE - 1; ++i) {
247         if (serpiente[0].x == serpiente[i].x &&
248             serpiente[0].y == serpiente[i].y) {
249             return true;
250         }
251     }
252     return false;
253 }

```