

```

1  /*
2  * Pablo Villa 874773
3  * 08/11/23
4  */
5  #include <iostream>
6  #include <iomanip>
7
8  using namespace std;
9  const int RECTANGULO = 1;
10 const int CUADRADO = 2;
11 const int TRIANGULO = 3;
12 const int FIN = 4;
13 const char SIMBOLO_EXTERIOR = '+';
14 const char SIMBOLO_HORIZONTAL = '-';
15 const char SIMBOLO_INTERMEDIO = ' ';
16 const char SIMBOLO_VERTICAL = '|';
17 const char SIMBOLO_PIRAMIDE = '*';
18 const string MENU = "1->Rectángulo 2->Cuadrado 3->Triángulo 4->Acabar: ";
19 void dibujar_linea(const char c_exterior, const char c_interior, const int largo);
20 void dibujar_triangulo(const char altura);
21 void dibujar_rectangulo(const int base, const int altura);
22 bool medidas_validas(const int base, const int altura);
23
24 int main(){
25
26     int opcion = 0;
27     int base = 0;
28     int altura = 0;
29     cout << "**** Dibujo de figuras geométricas ****" << endl;
30     while (opcion != FIN){
31         cout << MENU << endl;
32         cin >> opcion;
33
34         switch(opcion){
35             case RECTANGULO:
36                 cout << "Introduce la altura y la base del rectangulo positivas: ";
37                 cin >> altura >> base;
38                 if(medidas_validas(altura, base)){
39                     dibujar_rectangulo(base, altura);
40                 }
41                 else{
42                     cout << "base o altura no positivas" << endl;
43                 }
44                 break;
45
46             case CUADRADO:
47                 cout << "Introduce la altura del cuadrado: ";
48                 cin >> altura;
49                 if (medidas_validas(altura, altura)){
50                     dibujar_rectangulo(altura, altura);
51                 }
52                 else{
53                     cout << "base o altura no positivas" << endl;
54                 }
55                 break;
56
57             case TRIANGULO:
58                 cout << "Introduce la altura del triángulo: ";
59                 cin >> altura;
60                 if (medidas_validas(altura, altura)){
61                     dibujar_triangulo(altura);
62                 }
63                 else{
64                     cout << "base o altura no positivas" << endl;
65                 }
66                 break;

```

```

67         default:
68             cout << "Opcion incorrecta" << endl;
69     }
70 }
71 }
72 void dibujar_linea(const char c_exterior, const char c_interior, const int largo){
73     cout << c_exterior;
74     for(int i = 0; i < largo - 2; i++){
75         cout << c_interior;
76     }
77     cout << c_exterior << endl;
78 }
79
80 void dibujar_rectangulo(const int base, const int altura){
81     dibujar_linea(SIMBOLO_EXTERIOR, SIMBOLO_HORIZONTAL, base);
82     for (int i = 0; i < altura - 2; i++){
83         dibujar_linea(SIMBOLO_VERTICAL, SIMBOLO_INTERMEDIO, base);
84     }
85     dibujar_linea(SIMBOLO_EXTERIOR, SIMBOLO_HORIZONTAL, base);
86 }
87
88 void dibujar_triangulo(const char altura){
89     cout << setw(altura) << SIMBOLO_EXTERIOR << endl;
90     for(int i = 1; i < altura - 1; i++){
91         cout << setw(altura - i) << SIMBOLO_EXTERIOR << setw( 2 * i )
92             << SIMBOLO_EXTERIOR << endl;
93     }
94     cout << setw(1) << SIMBOLO_EXTERIOR << setw(altura * 2 - 2 )
95         << setfill(SIMBOLO_HORIZONTAL) << SIMBOLO_EXTERIOR << endl;
96 }
97 bool medidas_validas(const int base, const int altura){
98     if (altura > 0 && base > 0){
99         return true;
100     }
101 }

```

Líneas	a	b	a	b	x
11	12	0	0	0	0
12	12	9	0	0	0
14	12	9	12	27	0
19	2	9	12	27	7
21	2	9	12	27	-15/7
22	2	9	48 / 25	27	-15/7
23	2	9	48/25	23	-15/7
15	2	9	0	0	0
16	2	9	9	2	7
19	2	9	9	2	0
21	2	9	2	2	0
22	2	9	2	-1	0
17	2	-1	0	0	0

```

1  /*
2  * Juego de la Serpiente v1
3  * Pablo_Villa 08/11/2023
4  */
5  #include <iostream>
6  #include "terminal.h"
7  #include <cstdlib>
8
9
10 using namespace std;
11
12 const char TECLA_SIGUIENTE = ' ';
13 const char TECLA_FIN = 'F';
14 const char SERPIENTE = '@';
15 const char ARRIBA = 'W';
16 const char ABAJO = 'S';
17 const char IZQUIERDA = 'A';
18 const char DERECHA = 'D';
19 const char SIMBOLO_VERTICAL = '|';
20 const char SIMBOLO_INTERMEDIO = ' ';
21 const char SIMBOLO_HORIZONTAL = '-';
22 const char SIMBOLO_EXTERIOR = '+';
23 const int BASE = 80;
24 const int ALTURA = 22;
25 const int RETARDO = 60;
26 const int SERPIENTE_X_INICIAL = 10;
27 const int LIMITE_SUPERIOR = 3;
28 const int LIMITE_INFERIOR = 19;
29 const int LIMITE_IZQUIERDA = 2;
30 const int LIMITE_DERECHA = 78;
31 const int SERPIENTE_Y_INICIAL = 15;
32 const int MOVIMIENTO_X_DERECHA = 1;
33 const int MOVIMIENTO_Y_DESCENDENTE = 1;
34 const int MOVIMIENTO_X_IZQUIERDA = -1;
35 const int MOVIMIENTO_Y_ASCENDENTE = -1;
36 const string TITULO = "Juego de la serpiente ";
37 const string VERSION = "3.0";
38 const string TECLA_CONTINUAR = "ESPACIO";
39
40 void pantalla_inicial();
41 void inicializar_serpiente(int &serpiente_x, int &serpiente_y, int &movimiento_x, int &movimiento_y);
42 void dibujar_linea(const char c_exterior, const char c_interior, const int largo);
43 void dibujar_rectangulo(const int base, const int altura);
44 void pintar_cabeza_serpiente(const int serpiente_x, const int serpiente_y);
45 void borrar_cabeza_serpiente(const int serpiente_x, const int serpiente_y);
46 bool juego_noterminado(char tecla, int serpiente_x, int serpiente_y);
47 void elegir_movimiento(const char tecla, int &movimiento_x, int &movimiento_y);
48 void mover_serpiente(int &serpiente_x, int &serpiente_y, int movimiento_x, int movimiento_y);
49
50 int main() {
51     int serpiente_x = 0;
52     int serpiente_y = 0;
53     int movimiento_x = 0;
54     int movimiento_y = 0;
55     char tecla = '\0';
56
57     inicializar_serpiente(serpiente_x, serpiente_y, movimiento_x, movimiento_y);
58     retardar(RETARDO);
59     hacer_cursor_visible(false);
60     pantalla_inicial();
61     while(leer_tecla() != TECLA_SIGUIENTE){
62         retardar(RETARDO);
63     }
64     deshabilitar_modos_crudo_terminal();
65     borrar_terminal();
66

```

```

67     dibujar_rectangulo(BASE, ALTURA);
68     habilitar_modocruerto_terminal();
69     hacer_cursor_visible(false);
70     tecla = leer_tecla();
71
72     while (juego_noterminado(tecla ,serpiente_x, serpiente_y)) {
73         pintar_cabeza_serpiente(serpiente_x, serpiente_y);
74
75         retardar(RETARDO);
76
77         borrar_cabeza_serpiente(serpiente_x, serpiente_y);
78
79         elegir_movimiento(tecla,movimiento_x, movimiento_y);
80
81         mover_serpiente(serpiente_x, serpiente_y, movimiento_x, movimiento_y);
82
83         tecla = leer_tecla();
84     }
85     deshabilitar_modocruerto_terminal();
86     borrar_terminal();
87 }
88 void pantalla_inicial(){
89
90     poner_cursor(1,1);
91     cout << " ***** " << endl;
92     poner_cursor(1,2);
93     cout << " * "<< TITULO << VERSION << " * " << endl;
94     poner_cursor(1,3);
95     cout << " ***** " << endl;
96     poner_cursor(1,6);
97     cout << "   " << endl;
98     poner_cursor(1,7);
99     cout << " _/      \\" << endl;
100    poner_cursor(1,8);
101    cout << "  \\\_      \\" << endl;
102    poner_cursor(1,9);
103    cout << "      \\\_      \\" << endl;
104    poner_cursor(1,10);
105    cout << "          \\\_      \\" << endl;
106    poner_cursor(1,11);
107    cout << "          \\\_      \\\_      _|_ " << endl;
108    poner_cursor(1,12);
109    cout << "          \\\_      0 \\\_/ /      \\" << endl;
110    poner_cursor(1,13);
111    cout << "          \\\_      \\\_      \\\_/ " << endl;
112    poner_cursor(1,17);
113    cout << "Pulsa la tecla de " << TECLA_CONTINUAR << " para continuar" << endl;
114 }
115
116 void inicializar_serpiente(int &serpiente_x, int &serpiente_y, int &movimiento_x, int &movimiento_y) {
117     serpiente_x = SERPIENTE_X_INICIAL;
118     serpiente_y = SERPIENTE_Y_INICIAL;
119     movimiento_x = 0;
120     movimiento_y = 0;
121 }
122 void dibujar_linea(const char c_exterior, const char c_interior, const int largo){
123     cout << c_exterior;
124     for(int i = 0; i < largo - 2; i++){
125         cout << c_interior;
126     }
127     cout << c_exterior << endl;
128 }
129
130 void dibujar_rectangulo(const int base, const int altura){
131     poner_cursor(2,1);
132     cout << "+-----" << TITULO

```

```

133         << VERSION << " -----+ " << endl;
134     for (int i = 2; i < altura - 2; i++){
135         poner_cursor(2,i);
136         dibujar_linea(SIMBOLO_VERTICAL,SIMBOLO_INTERMEDIO,base);
137     }
138     poner_cursor(2,altura - 2);
139     dibujar_linea(SIMBOLO_EXTERIOR,SIMBOLO_HORIZONTAL,base);
140     poner_cursor(2, altura);
141     cout << ARRIBA << "-> Subir " << ABAJO << "-> Bajar " << IZQUIERDA
142         << "-> Izda " << DERECHA << "-> Dcha " << TECLA_FIN << "-> Fin" << endl;
143 }
144
145 void pintar_cabeza_serpiente(const int serpiente_x , const int serpiente_y){
146     poner_cursor(serpiente_x,serpiente_y);
147     cout << SERPIENTE;
148 }
149
150 void borrar_cabeza_serpiente(const int serpiente_x ,const int serpiente_y){
151     poner_cursor(serpiente_x,serpiente_y);
152     cout << " ";
153 }
154
155 bool juego_noterminado(char tecla, int serpiente_x, int serpiente_y){
156     return(toupper(tecla) != TECLA_FIN
157         && serpiente_x != LIMITE_IZQUIERDA && serpiente_x != LIMITE_DERECHA
158         && serpiente_y != LIMITE_SUPERIOR && serpiente_y != LIMITE_INFERIOR);
159 }
160
161 void elegir_movimiento(const char tecla, int &movimiento_x, int &movimiento_y){
162     switch (toupper(tecla)) {
163         case ARRIBA:
164             movimiento_x = 0;
165             movimiento_y = MOVIMIENTO_Y_ASCENDENTE;
166             break;
167
168         case ABAJO:
169             movimiento_x = 0;
170             movimiento_y = MOVIMIENTO_Y_DESCENDENTE;
171             break;
172
173         case IZQUIERDA:
174             movimiento_x = MOVIMIENTO_X_IZQUIERDA;
175             movimiento_y = 0;
176             break;
177
178         case DERECHA:
179             movimiento_x = MOVIMIENTO_X_DERECHA;
180             movimiento_y = 0;
181             break;
182     }
183 }
184 void mover_serpiente(int &serpiente_x, int &serpiente_y, int movimiento_x , int movimiento_y){
185     serpiente_x = serpiente_x + movimiento_x;
186     serpiente_y = serpiente_y + movimiento_y;
187 }

```

```

1  /*
2  * Pablo Villa 874773
3  * 09/11/23
4  */
5
6  #include <iostream>
7  #include <time.h>
8  #include <ctime>
9
10 using namespace std;
11 const int OPCION_PIEDRA = 1;
12 const int OPCION_PAPEL = 2;
13 const int OPCION_TIJERA = 3;
14 const int OPCIONES = 3;
15 const char PIEDRA = 'P';
16 const char PAPEL = 'A';
17 const char TIJERAS = 'T';
18
19 char eleccion_jugador();
20 char eleccion_ordenador();
21 void mostrar_eleccion(char jugador);
22 void mostrar_ganador(char jugador, char ordenador, int& aciertos);
23 void declarar_ganador(int aciertos, int rondas);
24
25 int main(){
26     string nombre = "";
27     int rondas = 0;
28     int aciertos = 0;
29     char opcion_Jugador = ' ';
30     char opcion_Ordenador = ' ';
31     srand(time(0));
32     setlocale(LC_ALL, "");
33
34     cout << "**** Juego de Piedra, papel o tijeras ****" << endl;
35     cout << "PIEDRA => " << PIEDRA << " PAPEL => " << PAPEL << " TIJERAS => " << TIJERAS << endl;
36     cout << "Nombre jugador: ";
37     cin >> nombre;
38     cout << "Número de jugadas a realizar: ";
39     cin >> rondas;
40
41     for(int ronda = 1; ronda <= rondas; ronda++){
42         cout << " " << endl;
43         cout << "Introduce tu Jugada " << ronda << ": ";
44         opcion_Jugador = eleccion_jugador();
45         opcion_Ordenador = eleccion_ordenador();
46         mostrar_ganador(opcion_Jugador, opcion_Ordenador, aciertos);
47     }
48     cout << " " << endl;
49     cout << nombre;
50     declarar_ganador(aciertos, rondas);
51 }
52
53 char eleccion_jugador() {
54     char opcion_Jugador;
55     bool validar = false;
56
57     while (!validar) {
58         cin >> opcion_Jugador;
59
60         if (opcion_Jugador == PIEDRA || opcion_Jugador == PAPEL || opcion_Jugador == TIJERAS) {
61             cout << "La opción: " << opcion_Jugador << " es válida" << endl;
62             switch (opcion_Jugador) {
63                 case PIEDRA:
64                     cout << "Tu elección es PIEDRA" << endl;
65                     break;
66                 case PAPEL:

```

```

67         cout << "Tu elección es PAPEL " << endl;
68         break;
69     case TIJERAS:
70         cout << "Tu elección es TIJERA" << endl;
71         break;
72     }
73     validar = true;
74 } else {
75     cout << "Opción no válida. Repítela." << endl;
76 }
77 }
78
79 return opcion_Jugador;
80 }
81
82 char eleccion_ordenador(){
83     int num_aleatorio = 0;
84     char eleccion = ' '; //no compila sino
85     num_aleatorio = rand()% OPCIONES + 1;
86
87     switch(num_aleatorio){
88     case OPCION_PIEDRA :
89         eleccion = PIEDRA;
90         cout << "el ordenador ha elegido PIEDRA" << endl;
91         break;
92
93     case OPCION_PAPEL :
94         eleccion = PAPEL;
95         cout << "el ordenador ha elegido PAPEL " << endl;
96         break;
97
98     case OPCION_TIJERA :
99         eleccion = TIJERAS;
100        cout << "el ordenador ha elegido TIJERA" << endl;
101    }
102    return eleccion;
103 }
104
105 void mostrar_ganador(char jugador, char ordenador, int& aciertos) {
106     cout << "Resultado: ";
107
108     if (jugador == ordenador) {
109         cout << "empate" << endl;
110     } else if ((jugador == PIEDRA && ordenador == TIJERAS) ||
111        (jugador == PAPEL && ordenador == PIEDRA) ||
112        (jugador == TIJERAS && ordenador == PAPEL)) {
113         cout << "¡Ganas tú!" << endl;
114         aciertos++;
115     } else {
116         cout << "Gana ordenador" << endl;
117     }
118
119     cout << "Rondas ganadas: " << aciertos << endl;
120 }
121
122 void declarar_ganador(int aciertos, int rondas){
123
124     if (aciertos > rondas / 2){
125         cout << " después de " << rondas
126             << " jugadas ganas tú " << endl;
127     }
128     else {
129         cout << " después de " << rondas
130             << " jugadas ganas el ordenador " << endl;
131     }
132 }

```