

```

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
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 << RECTANGULO << "->Rectángulo" << CUADRADO << "->Cuadrado "
32             << TRIANGULO << "->Triángulo " << FIN << "->Acabar: " << endl;
33         cin >> opcion;
34
35         switch(opcion){
36             case RECTANGULO:
37                 cout << "Introduce la altura y la base del rectangulo positivas: ";
38                 cin >> altura >> base;
39                 if(medidas_validas(altura, base)){
40                     dibujar_rectangulo(base, altura);
41                 }
42                 else{
43                     cout << "base o altura no positivas" << endl;
44                 }
45                 break;
46
47             case CUADRADO:
48                 cout << "Introduce la altura del cuadrado: ";
49                 cin >> altura;
50                 if (medidas_validas(altura, altura)){
51                     dibujar_rectangulo(altura, altura);
52                 }
53                 else{
54                     cout << "base o altura no positivas" << endl;
55                 }
56                 break;
57
58             case TRIANGULO:
59                 cout << "Introduce la altura del triángulo: ";
60                 cin >> altura;
61                 if (medidas_validas(altura, altura)){
62                     dibujar_triangulo(altura);
63                 }
64                 else{
65                     cout << "base o altura no positivas" << endl;
66                 }

```

```

67         break;
68     default:
69         cout << "Opcion incorrecta" << endl;
70     }
71 }
72 }
73 void dibujar_linea(const char c_exterior, const char c_interior, const int largo){
74     cout << c_exterior;
75     for(int i = 0; i < largo - 2; i++){
76         cout << c_interior;
77     }
78     cout << c_exterior << endl;
79 }
80
81 void dibujar_rectangulo(const int base, const int altura){
82     dibujar_linea(SIMBOLO_EXTERIOR, SIMBOLO_HORIZONTAL, base);
83     for (int i = 0; i < altura - 2; i++){
84         dibujar_linea(SIMBOLO_VERTICAL, SIMBOLO_INTERMEDIO, base);
85     }
86     dibujar_linea(SIMBOLO_EXTERIOR, SIMBOLO_HORIZONTAL, base);
87 }
88
89 void dibujar_triangulo(const char altura){
90     cout << setw(altura) << SIMBOLO_EXTERIOR << endl;
91     for(int i = 1; i < altura - 1; i++){
92         cout << setw(altura - i) << SIMBOLO_EXTERIOR << setw( 2 * i )
93             << SIMBOLO_EXTERIOR << endl;
94     }
95     cout << setw(1) << SIMBOLO_EXTERIOR << setw(altura * 2 - 2 )
96         << setfill(SIMBOLO_HORIZONTAL) << SIMBOLO_EXTERIOR << endl;
97 }
98 bool medidas_validas(const int base, const int altura){
99     return( base > 0 && altura > 0);
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	12	9	12	27	7
21	12	9	12	27	-22
22	12	9	24	2	-22
23	12	9	24	2	-22
15	2	9	0	0	0
16	2	9	9	2	0
19	2	9	9	2	7
21	2	9	9	2	0
22	2	9	-1	2	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_terminado(char tecla, int serpiente_x, int serpiente_y);
47 void obtener_direccion(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_teclea() != TECLA_SIGUIENTE){
62         retardar(RETARDO);
63     }
64     deshabilitar_modos_crudo_terminal();
65     borrar_terminal();
66

```

```

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

```

```

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

```

```

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 GANA_ORDENADOR = 1;
12 const int EMPATE = 2;
13 const int GANA_USUARIO = 3;
14 const int OPCION_PIEDRA = 1;
15 const int OPCION_PAPEL = 2;
16 const int OPCION_TIJERA = 3;
17 const int OPCIONES = 3;
18 const char PIEDRA = 'P';
19 const char PAPEL = 'A';
20 const char TIJERAS = 'T';
21
22 char eleccion_jugador();
23 char eleccion_ordenador();
24 void mostrar_eleccion(char jugador);
25 int obtener_ganador_ronda(char jugador, char ordenador);
26 int mostrar_ganador_ronda(char jugador, char ordenador, int &aciertos);
27 void declarar_ganador(int aciertos, int rondas);
28
29 int main(){
30     string nombre = "";
31     int rondas = 0;
32     int aciertos = 0;
33     char jugada_usuario = '/0';
34     char jugada_ordenador = '/0';
35
36     srand(time(0));
37     setlocale(LC_ALL, "");
38
39     cout << "**** Juego de Piedra, papel o tijeras ****" << endl;
40     cout << "PIEDRA =>  " << PIEDRA << " PAPEL =>  " << PAPEL << " TIJERAS =>  " << TIJERAS << endl;
41     cout << "Nombre jugador: ";
42     cin >> nombre;
43     cout << "Número de jugadas a realizar: ";
44     cin >> rondas;
45
46     for(int ronda = 1; ronda <= rondas; ronda++){
47         cout << endl;
48         cout << "Introduce tu Jugada " << ronda << ": ";
49         jugada_usuario = eleccion_jugador();
50         jugada_ordenador = eleccion_ordenador();
51         obtener_ganador_ronda(jugada_usuario, jugada_ordenador);
52         mostrar_ganador_ronda(jugada_usuario, jugada_ordenador, aciertos);
53     }
54     cout << endl;
55     cout << nombre;
56     declarar_ganador(aciertos, rondas);
57 }
58
59 char eleccion_jugador() {
60     char jugada_usuario;
61     bool valida = false;
62
63     while (!valida) {
64         cin >> jugada_usuario;
65
66         if (jugada_usuario == PIEDRA || jugada_usuario == PAPEL || jugada_usuario == TIJERAS) {

```

```

67         cout << "La opción: " << jugada_usuario << " es válida" << endl;
68         cout << " Tu elección es";
69         switch (toupper(jugada_usuario)) {
70             case PIEDRA:
71                 cout << " PIEDRA" << endl;
72                 break;
73             case PAPEL:
74                 cout << " PAPEL " << endl;
75                 break;
76             case TIJERAS:
77                 cout << " TIJERAS" << endl;
78                 break;
79         }
80         valida = true;
81     } else {
82         cout << "Opción no válida. Repítela." << endl;
83     }
84 }
85
86 return jugada_usuario;
87 }
88
89 char eleccion_ordenador(){
90     int num_aleatorio = 0;
91     char eleccion = '/0';
92     num_aleatorio = rand()% OPCIONES + 1;
93
94     cout << "el ordenador ha elegido";
95     switch(num_aleatorio){
96         case OPCION_PIEDRA :
97             eleccion = PIEDRA;
98             cout << " PIEDRA" << endl;
99             break;
100
101         case OPCION_PAPEL :
102             eleccion = PAPEL;
103             cout << " PAPEL " << endl;
104             break;
105
106         case OPCION_TIJERA :
107             eleccion = TIJERAS;
108             cout << " TIJERA" << endl;
109     }
110     return eleccion;
111 }
112
113 int obtener_ganador_ronda(char jugador, char ordenador) {
114
115     if (jugador == ordenador) {
116         return EMPATE;
117     } else if ((jugador == PIEDRA && ordenador == TIJERAS) ||
118                (jugador == PAPEL && ordenador == PIEDRA) ||
119                (jugador == TIJERAS && ordenador == PAPEL)) {
120         return GANA_USUARIO;
121     } else {
122         return GANA_ORDENADOR;
123     }
124 }
125
126 int mostrar_ganador_ronda(char jugador, char ordenador, int &aciertos){
127     cout << "Resultado: ";
128     int resultado = obtener_ganador_ronda(jugador, ordenador);
129     switch (resultado) {
130         case EMPATE:
131             cout << "Es un empate" << endl;
132             break;
133         case GANA_USUARIO:

```



```
133         cout << "|Ganaste esta ronda!" << endl;
134         aciertos++;
135         break;
136     case GANA_ORDENADOR:
137         cout << "El ordenador gana esta ronda." << endl;
138         break;
139     }
140
141 }
142
143 void declarar_ganador(int aciertos, int rondas){
144
145     if (aciertos > rondas / 2){
146         cout << " después de " << rondas
147             << " jugadas ganas tú " << endl;
148     }
149     else {
150         cout << " después de " << rondas
151             << " jugadas ganas el ordenador " << endl;
152     }
153 }
```