Explicación del Código del Juego de la Serpiente

Introducción

Este documento explica el funcionamiento del código del juego de la serpiente. El juego está escrito en C++ y utiliza las librerías estándar junto con algunas funciones específicas de Windows para manejar el posicionamiento del cursor en la consola.

Librerías Utilizadas

#include <windows.h>: Esta librería permite el uso de funciones específicas de Windows, como SetConsoleCursorPosition, que se usa para mover el cursor en la consola.

#include <iostream>: Esta librería permite el uso de operaciones de entrada y salida estándar, como printf y std::cout.

#include <conio.h>: Esta librería proporciona funciones para manejar la entrada de teclado, como _getch y _kbhit.

#include <stdlib.h>: Esta librería se usa para funciones de utilidad general, como rand para generar números aleatorios y system para ejecutar comandos del sistema.

Definiciones y Variables Globales

#define Up 72

#define Down 80

#define Left 75

#define Right 77

#define ESC 27

Estas definiciones establecen los códigos de las teclas para las flechas de dirección y la tecla ESC.

---

int body[200][2]	// Almacena las posiciones del cuerpo de la serpiente
int n = 1	// Contador para la longitud del cuerpo
int height = 3	// Longitud inicial de la serpiente
int x = 10, y = 12	// Posición inicial de la serpiente
int dir = 3	// Dirección inicial de la serpiente (3 = derecha)
int fx = 30, fy = 15	// Posición inicial de la comida
int vel = 100, g = 1	// Velocidad y nivel de dificultad
int score = 0	// Puntuación del juego
bool isPaused = false	// Estado de pausa del juego
char key	// Variable para capturar la entrada del teclado

Funciones del Juego

gotoxy(int x, int y)

```
void gotoxy(int x, int y)
{
   HANDLE hCon;
   COORD dwPos;

   dwPos.X = x;
   dwPos.Y = y;
   hCon = GetStdHandle(STD_OUTPUT_HANDLE);
```

```
    SetConsoleCursorPosition(hCon, dwPos);
}
```

Esta función mueve el cursor de la consola a la posición (x, y). Es crucial para dibujar el juego en la consola.

start_screen()

```
void start_screen()
{
    char cover[18][71] =
    {
        "  SSSSSSSS N     N AAAAAAAAA K    K  EEEEEEE ",
        " S      S N N    N A     A K   K   E       ",
        " S       N N   N A     A K  K    E       ",
        "  SSSSSSS  N  N  N AAAAAAAAA K K    EEEEE   ",
        "       S N   N N A     A K K     E       ",
        "       S N    N N A     A KK      E       ",
        " SSSSSSSS  N    N N A     A K K    EEEEEEE  ",
        "                                ",
        "                                ",
        "                                ",
        "                                ",
        "                                ",
        "                                ",
        "                                ",
        "                                ",
        "   SSSS                          ",
```

```
    "  S   S                          ",
    "   SSSS                           "
  };

  for (int i = 0; i < 18; ++i)
  {
    printf("%s\n", cover[i]);
  }

  printf("\n\nPresiona cualquier tecla para comenzar...\n");
  _getch(); // Esperamos a que el usuario presione una tecla
  system("cls");
}
```

Esta función muestra la pantalla de inicio con el título "SNAKE" y una pequeña serpiente. Espera a que el usuario presione una tecla para comenzar el juego.

---

draw()

```
void draw()
{
  //Horizontal Line
  for (int i = 2 ; i < 78 ; i++)
  {
    gotoxy(i, 3);    printf("%c", 205);
    gotoxy(i, 23);   printf("%c", 205);
  }
  //Vertical Line
```

```
  for(int i = 4 ; i < 23 ; i++)
  {
    gotoxy(2, i);    printf("%c", 186);
    gotoxy(77, i);    printf("%c", 186);
  }
  //Corner
  gotoxy(2, 3);    printf("%c", 201);
  gotoxy(2, 23);    printf("%c", 200);
  gotoxy(77, 3);    printf("%c", 187);
  gotoxy(77, 23);    printf("%c", 188);
}
```

Esta función dibuja el marco del área de juego usando caracteres ASCII.

---

save_position()

```
void save_position()
{
  body[n][0] = x;
  body[n][1] = y;
  n++;
  if(n == height) n = 1;
}
```

Guarda la posición actual de la cabeza de la serpiente en la matriz body.

---

draw_body()

```
void draw_body()
```

```
{
   for(int i = 1 ; i < height ; i++)
   {
      gotoxy(body[i][0], body[i][1]); printf("*");
   }
}
```

Dibuja el cuerpo de la serpiente en la consola.

---

remove_body()

```
void remove_body()
{
   gotoxy(body[n][0], body[n][1]); printf(" ");
}
```

Elimina la última posición de la serpiente para simular el movimiento.

---

move_character()

```
void move_character()
{
   if (_kbhit())
   {
      key = _getch();
      switch (key)
      {
         case Up:
            if(dir != 2) dir = 1;
```

```
            break;
        case Down:
            if(dir != 1) dir = 2;
            break;
        case Left:
            if(dir != 3) dir = 4;
            break;
        case Right:
            if(dir != 4) dir = 3;
            break;
        case ESC:
            isPaused = true;
            break;
    }
  }
}
```

Esta función captura la entrada del teclado y cambia la dirección de la serpiente o pausa el juego.

---

change_vel()

```
void change_vel()
{
  if(score == g * 20)
  {
    vel -= 10;
    g++;
```

```
    }
}
```

Aumenta la velocidad del juego cada vez que el jugador alcanza un múltiplo de 20 puntos.

---

food()

```
void food()
{
    if (x == fx && y == fy)
    {
        fx = (rand() % 73) + 4;
        fy = (rand() % 19) + 4;

        height++;
        score += 10;
        gotoxy(fx, fy); printf("%c", 250);

        change_vel();
    }
}
```

Genera nueva comida en una posición aleatoria cuando la serpiente la come y aumenta la longitud de la serpiente.

---

game_over()

```
bool game_over()
{
    if (y == 3 || y == 23 || x == 2 || x == 77) return false;
```

```
    for(int j = height - 1 ; j > 0 ; j--)
    {
        if(body[j][0] == x && body[j][1] == y)
        {
            return false;
        }
    }
    return true;
}
```

Verifica si el juego ha terminado, ya sea porque la serpiente choca con las paredes o consigo misma.

---

point()

```
void point()
{
    gotoxy(3, 1); printf("score %d", score);
}
```

Muestra la puntuación del jugador en la parte superior izquierda de la pantalla.

---

pause_menu()

```
void pause_menu()
{
    int option = 0;
    bool inMenu = true;
    char options[2][10] = { "Continuar", "Salir" };
```

```
while (inMenu)
{
    // Dibuja el marco del menú de pausa
    for (int i = 30; i <= 50; ++i)
    {
        gotoxy(i, 10); printf("%c", 205);
        gotoxy(i, 14); printf("%c", 205);
    }
    for (int i = 11; i <= 13; ++i)
    {
        gotoxy(30, i); printf("%c", 186);
        gotoxy(50, i); printf("%c", 186);
    }
    gotoxy(30, 10); printf("%c", 201);
    gotoxy(50, 10); printf("%c", 187);
    gotoxy(30, 14); printf("%c", 200);
    gotoxy(50, 14); printf("%c", 188);

    // Dibuja las opciones del menú
    for (int i = 0; i < 2; ++i)
    {
        gotoxy(35, 11 + i);
        if (i == option)
            printf("-> %s", options[i]);
        else
```

```
        printf("   %s", options[i]);
}


char ch = _getch();
if (ch == Up && option > 0)
{
   option--;
}
else if (ch == Down && option < 1)
{
   option++;
}
else if (ch == '\r') // tecla Enter
{
   if (option == 0)
   {
      inMenu = false;
      isPaused = false;
      // Limpia el menú de pausa
      for (int i = 10; i <= 14; ++i)
      {
         gotoxy(30, i);
         printf("                    ");
      }
      draw();
```

```
        }
        else if (option == 1)
        {
            exit(0);
        }
    }
  }
}
```

Muestra un menú de pausa donde el jugador puede continuar el juego o salir. Usa las teclas de flecha para navegar y Enter para seleccionar.

---

game_over_screen()

```
void game_over_screen()
{
  char gameOver[14][50] =
  {
    "  GGGGGG    AAAAA   M      M  EEEEEEE   ",
    " G       A     A  M M   M M  E         ",
    "G        A      A M  M M  M  EEEEE     ",
    "G   GGGG  AAAAAAAAA M   M   M  E        ",
    "G     G  A      A M      M  E         ",
    " G    G  A      A M      M  E         ",
    "  GGGGG   A      A M      M  EEEEEEE   ",
    "                          ",
    "  OOOOO   V     V  EEEEEEE  RRRRRR       ",
    " O    O  V     V  E       R    R      ",
```

```
    "O     O V    V  EEEEE   RRRRRR     ",
    "O     O V  V   E     R R      ",
    " O    O   V V   E     R  R     ",
    "  OOOOO    V    EEEEEEE R   R     "
  };

  system("cls");
  for (int i = 0; i < 14; ++i)
  {
    gotoxy(15, 7 + i); // Centro aproximado de la pantalla
    printf("%s\n", gameOver[i]);
  }

  printf("\n\nPresiona cualquier tecla para salir...\n");
  _getch(); // Esperamos a que el usuario presione una tecla
  exit(0);
}
```

Muestra una pantalla de "GAME OVER" cuando el juego termina. El jugador debe presionar una tecla para salir del juego.

Lógica Principal del Juego

```
int main()
{
  start_screen();
  draw();
```

```
gotoxy(fx, fy); printf("%c", 250);


while (true)
{
   if (isPaused)
   {
      pause_menu();
   }
   else
   {
      if (!game_over())
      {
         game_over_screen(); // Llama a la pantalla de fin de juego
      }
      remove_body();
      save_position();
      draw_body();
      food();
      point();
      move_character();
      move_character(); // Se agrega 2do para realizar movimientos bruscos


      if (dir == 1) y--;
      if (dir == 2) y++;
      if (dir == 3) x++;
```

```
        if (dir == 4) x--;


        Sleep(vel);
      }
    }
    return 0;
}
```

La función principal main() maneja el flujo del juego:

1. Muestra la pantalla de inicio.
2. Dibuja el marco del juego.
3. Coloca la comida en una posición inicial.
4. Ejecuta un bucle infinito donde:
    - Si el juego está pausado, muestra el menú de pausa.
    - Si el juego no ha terminado, actualiza la posición de la serpiente, verifica si ha comido la comida, actualiza la puntuación y maneja la entrada del jugador.
    - Si el juego ha terminado, muestra la pantalla de "GAME OVER".

Este código crea una versión sencilla del clásico juego de la serpiente, donde el jugador controla una serpiente que se mueve por la pantalla, come comida para crecer y debe evitar chocar con las paredes o consigo misma.
