Como Funciona el Juego

El juego es una implementación del clásico *Snake* donde el jugador controla una serpiente que se mueve en un rectángulo cerrado, recolectando comida para crecer y ganar puntos mientras evita chocar contra las paredes o su propio cuerpo. Además, incluye características adicionales como comida especial, una pausa con mensaje, un temporizador, y ajustes de velocidad. A continuación, se explica cómo funciona:

**Elementos clave del juego**

1. **Serpiente**:
   * Compuesta por segmentos representados por \*, que se mueven en dirección fija hasta que el jugador cambia la dirección con las flechas del teclado.
   * La serpiente crece al consumir comida y almacena sus posiciones en un arreglo bidimensional cuerpo.
2. **Tablero de juego**:
   * Representado por un rectángulo cerrado delimitado con caracteres ASCII.
   * El área tiene un tamaño fijo de 76x20 dentro del rectángulo.
3. **Comida normal**:
   * Representada por \* en una posición aleatoria dentro del tablero.
   * Cada vez que la serpiente la consume:
     + La serpiente crece en tamaño.
     + El jugador gana puntos (+10 por comida).
     + Se genera una nueva comida en una posición aleatoria.
4. **Comida especial**:
   * Representada por @ y aparece cada 30 segundos en una posición aleatoria.
   * La comida especial permanece activa durante 10 segundos. Si es consumida:
     + Se altera la velocidad del juego: puede aumentar o disminuir aleatoriamente.
     + El efecto de velocidad dura 10 segundos.
5. **Pausa**:
   * El juego puede pausarse presionando la tecla P. Durante la pausa:
     + El movimiento de la serpiente se detiene.
     + El temporizador se congela.
     + Un mensaje aparece en el centro del tablero indicando que el juego está pausado.
6. **Game over**:
   * El juego termina si:
     + La serpiente choca contra una pared.
     + La serpiente choca contra su propio cuerpo.
7. **Temporizador**:
   * Cuenta el tiempo transcurrido desde el inicio del juego.
   * El tiempo se detiene al pausar el juego.
   * Se muestra al jugador en pantalla y se utiliza para medir la duración de la partida.
8. **Puntaje final**:
   * Al finalizar el juego, se muestra el puntaje acumulado en la parte inferior de la pantalla.

Código Del Juego

#include <windows.h>

#include <iostream>

#include <conio.h>

#include <stdlib.h>

#include <ctime>

#define ARRIBA 72

#define IZQUIERDA 75

#define DERECHA 77

#define ABAJO 80

#define ESC 27

#define PAUSA 'p'

int cuerpo[200][2];

int n = 1;

int tamano\_serpiente = 5;

int x = 10, y = 12;

int sent = DERECHA;

char tecla;

int xcom = 30, ycom = 15;

int xcom\_esp = 0, ycom\_esp = 0;

int vel = 100;

int punt = 0;

bool pausa = false;

bool comida\_especial\_activa = false;

time\_t tiempo\_comida\_especial = 0;

time\_t inicio\_juego;

time\_t tiempo\_pausa\_total = 0;

time\_t tiempo\_inicio\_pausa;

int velocidad\_original = 100;

bool velocidad\_alterada = false;

time\_t inicio\_velocidad\_alterada;

void posicion(int x, int y) {

HANDLE hCon;

COORD dwPos;

dwPos.X = x;

dwPos.Y = y;

hCon = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleCursorPosition(hCon, dwPos);

}

void pintar() {

for (int i = 2; i < 78; i++) {

posicion(i, 3); printf("%c", 205);

posicion(i, 23); printf("%c", 205);

}

for (int v = 4; v < 23; v++) {

posicion(2, v); printf("%c", 186);

posicion(77, v); printf("%c", 186);

}

posicion(2, 3); printf("%c", 201);

posicion(2, 23); printf("%c", 200);

posicion(77, 3); printf("%c", 187);

posicion(77, 23); printf("%c", 188);

}

void guardar\_posicion() {

cuerpo[n][0] = x;

cuerpo[n][1] = y;

cuerpo[0][0] = x;

cuerpo[0][1] = y;

n++;

if (n == tamano\_serpiente) n = 1;

}

void dibujar\_cuerpo() {

for (int i = 1; i < tamano\_serpiente; i++) {

posicion(cuerpo[i][0], cuerpo[i][1]); printf("\*");

}

}

void borrar\_cuerpo() {

for (int i = 1; i < tamano\_serpiente; i++) {

posicion(cuerpo[i][0], cuerpo[i][1]); printf(" ");

}

}

void tecleado() {

if (\_kbhit()) {

tecla = \_getch();

switch (tecla) {

case ARRIBA:

if (sent != ABAJO) sent = ARRIBA;

break;

case ABAJO:

if (sent != ARRIBA) sent = ABAJO;

break;

case DERECHA:

if (sent != IZQUIERDA) sent = DERECHA;

break;

case IZQUIERDA:

if (sent != DERECHA) sent = IZQUIERDA;

break;

case PAUSA:

pausa = !pausa;

if (pausa) {

tiempo\_inicio\_pausa = time(NULL);

posicion(31, 13);

printf("Juego en pausa. Presiona 'P' para continuar.");

} else {

tiempo\_pausa\_total += time(NULL) - tiempo\_inicio\_pausa;

posicion(31, 13);

printf(" ");

}

break;

}

}

}

void comida() {

if (x == xcom && y == ycom) {

xcom = (rand() % 73) + 4;

ycom = (rand() % 19) + 4;

tamano\_serpiente++;

punt += 10;

posicion(xcom, ycom); printf("%c", '\*');

}

}

void comida\_especial() {

time\_t tiempo\_actual = time(NULL);

if (!comida\_especial\_activa && tiempo\_actual % 30 == 0) {

xcom\_esp = (rand() % 73) + 4;

ycom\_esp = (rand() % 19) + 4;

comida\_especial\_activa = true;

tiempo\_comida\_especial = tiempo\_actual;

posicion(xcom\_esp, ycom\_esp); printf("%c", '@');

}

if (comida\_especial\_activa && (tiempo\_actual - tiempo\_comida\_especial >= 10)) {

posicion(xcom\_esp, ycom\_esp); printf(" ");

comida\_especial\_activa = false;

}

if (comida\_especial\_activa && x == xcom\_esp && y == ycom\_esp) {

comida\_especial\_activa = false;

posicion(xcom\_esp, ycom\_esp); printf(" ");

velocidad\_alterada = true;

inicio\_velocidad\_alterada = tiempo\_actual;

if (rand() % 2 == 0) {

vel = vel > 50 ? vel - 50 : vel;

} else {

vel = vel < 300 ? vel + 50 : vel;

}

}

if (velocidad\_alterada && (tiempo\_actual - inicio\_velocidad\_alterada >= 10)) {

vel = velocidad\_original;

velocidad\_alterada = false;

}

}

void mostrar\_temporizador() {

time\_t tiempo\_actual = time(NULL);

int tiempo\_transcurrido = tiempo\_actual - inicio\_juego - tiempo\_pausa\_total;

posicion(60, 2);

printf("Tiempo: %02d:%02d", tiempo\_transcurrido / 60, tiempo\_transcurrido % 60);

}

bool game\_over() {

if (y == 3 || y == 23 || x == 2 || x == 77) return true;

for (int j = n - 1; j > 0; j--) {

if (cuerpo[j][0] == x && cuerpo[j][1] == y)

return true;

}

return false;

}

void mostrar\_puntaje\_final() {

posicion(2, 25);

printf("Puntaje final: %d", punt);

}

int main() {

srand(time(NULL));

inicio\_juego = time(NULL);

pintar();

posicion(xcom, ycom); printf("%c", '\*');

while (tecla != ESC && !game\_over()) {

if (!pausa) {

borrar\_cuerpo();

guardar\_posicion();

dibujar\_cuerpo();

comida();

comida\_especial();

mostrar\_temporizador();

tecleado();

switch (sent) {

case ARRIBA: y--; break;

case ABAJO: y++; break;

case DERECHA: x++; break;

case IZQUIERDA: x--; break;

}

Sleep(vel);

posicion(4, 2);

printf("Puntaje: %d", punt);

} else {

tecleado();

Sleep(100);

}

}

mostrar\_puntaje\_final();

posicion(0, 26);

system("pause>null");

return 0;

}

Explicación Del Código

**Librerías Incluidas**

* **windows.h**: Usada para funciones relacionadas con la consola, como mover el cursor y pausar el programa.
* **iostream**: Maneja entrada/salida estándar, como printf y cin.
* **conio.h**: Permite capturar teclas sin bloquear (\_kbhit() y \_getch()).
* **stdlib.h**: Proporciona funciones como rand() para generar números aleatorios.
* **ctime**: Utilizada para trabajar con el tiempo (time()).

**Declaración de Variables**

* **cuerpo [200][2]**: Matriz que almacena las posiciones (x, y) de los segmentos del cuerpo de la serpiente.
* **tamano\_serpiente**: Longitud actual de la serpiente.
* **x, y**: Posición actual de la cabeza de la serpiente.
* **sent**: Dirección actual de movimiento.
* **xcom, ycom**: Coordenadas de la comida normal.
* **xcom\_esp, ycom\_esp**: Coordenadas de la comida especial.
* **vel**: Velocidad actual del juego en milisegundos (tiempo entre cada fotograma).
* **punt**: Puntuación actual del jugador.
* **Temporizadores**: Variables relacionadas con el tiempo total del juego y las pausas.

**Funciones Principales**

**a) posicion ()**

* Mueve el cursor de la consola a las coordenadas (x, y). Esto permite que el juego se dibuje en la posición correcta.

**b) pintar ()**

* Dibuja el borde del rectángulo donde se juega.

**c) Movimiento y Dibujado de la Serpiente**

* **guardar\_posicion()**: Guarda la posición de cada segmento del cuerpo de la serpiente.
* **dibujar\_cuerpo()**: Dibuja la serpiente en su nueva posición.
* **borrar\_cuerpo()**: Borra la serpiente en su posición actual antes de moverla.

**d) Control de Teclado**

* Captura la entrada del usuario para cambiar la dirección de la serpiente o pausar el juego.

**e) Comida Normal y Especial**

* La **comida normal** aumenta el tamaño de la serpiente y la puntuación.
* La **comida especial** altera la velocidad durante 10 segundos.

**f) Temporizador**

* Muestra el tiempo transcurrido desde el inicio del juego, descontando el tiempo en pausa.

**g) Fin del Juego**

* Muestra el puntaje final en la parte inferior de la consola.

**Ciclo Principal del Juego**

* **Control del Juego**: Mueve la serpiente, dibuja el entorno y gestiona la lógica de pausa, comida y colisión.
* **Pausa**: Detiene todos los cálculos, incluyendo el temporizador.