

¿En dónde se desarrolló el juego?

Se desarrolló en la aplicación de **CodeBlocks** en ella en el código se desarrollaron 6 librerías y se definieron 4 palabras primordiales para poder desarrollarlo, en el código se tuvieron que poner clases ya sean públicas o privadas, también se utilizaron bucles para poder hacer el disparo para poner los meteoritos para poder moverte, se utilizó el **main** para poder juntar todo acerca de las clases y los bucles.

Sentencias Utilizadas

- **Función principal main**

Biblioteca de entrada y salida, la cual será para imprimir en pantalla, es la biblioteca estándar **input, output dc**.

Para mandar a imprimir en pantalla, tomaremos el siguiente ejemplo:

Mandamos a imprimir un asterisco.

- **Para esto utilizamos la función printf**

```
1  #include<stdio.h>
2
3
4  int main(){
5      printf("*");
6
7  }
```

(Figura 1. Ejemplo de la función printf)

Ahora bien, empecemos con la biblioteca **Windows.h** y al final del código colocamos un **return 0**.

```

1  #include<stdio.h>
2  #include<windows.h>
3
4
5  int main(){
6      printf("*");
7
8
9      return 0;
10 ]
11 )
12

```

(Figura 2. Ejemplo de la biblioteca Windows.h y línea final del código)

En este momento la biblioteca **Windows.h** nos está permitiendo dar una posición al cursor y ya teniendo esa posición pedimos mandar a imprimir lo que queramos en dichas coordenadas. En este caso el asterisco.

La función siguiente es **SetConsoleCursorPosition ()**; esta función tiene dos parámetros.

1. Identificador llamado **Handle**
2. Estructura de datos

Se conoce como **Handle** a un tipo particular de punteros "inteligentes". Los **Handles** son utilizados cuando un programa hace referencia a bloques de memoria u objetos controlados por otros sistemas, tales como una base de datos o un sistema operativo.

Entonces para poder hacer uso del **SetConsoleCursorPosition ()**; primero debemos saber el identificador de una ventana al que se quiere manipular.

Al escribir la palabra **Handle**, estamos creando un identificador al cual le podemos dar un nombre. Con este **Handle** podemos recuperar el identificador de la consola que aparece al correr nuestro programa. Para hacer esto colocamos la variable que creamos (el nombre del identificador) y la hacemos igual (=) a un **GetStdHandle ()**;

```

1  #include<stdio.h>
2  #include<windows.h>
3
4
5  int main(){
6      printf("***");
7
8      HANDLE hCon;
9      hCon = GetStdHandle();
10
11     SetConsoleCursorPosition();
12
13     return 0;
14
15 }
16

```

(Figura 3. Ejemplo de la función **SetConsoleCursorPosition ()** ;)

Con este **GetStdHandle** es recuperar el identificador de la consola; esto tiene un parámetro donde se identifica un tipo de control que deseamos tener de esta consola.

❖ *Parámetros utilizados*

OutPut_Handle: Toma el control de la salida de la consola.

A su vez, con este parámetro podemos situar el cursor en cualquier posición para mandar a imprimir lo que queramos

```

1  #include<stdio.h>
2  #include<windows.h>
3
4
5  int main(){
6      printf("***");
7
8      HANDLE hCon;
9      hCon = GetStdHandle(STD_OUTPUT_HANDLE);
10
11     SetConsoleCursorPosition();
12
13     return 0;
14
15 }
16

```

(Figura 4. Ejemplo del parámetro **OutPut Handle**)

- ✓ Primer parámetro: Identificador de la consola a la función **SetConsoleCursorPosition ();**

```
1 #include<stdio.h>
2 #include<windows.h>
3
4
5 int main(){
6     printf("***");
7
8     HANDLE hCon;
9     hCon = GetStdHandle(STD_OUTPUT_HANDLE);
10
11     SetConsoleCursorPosition(hCon, );
12
13     return 0;
14
15 }
16
```

(Figura 5. Ejemplo de el identificador de la consola en la función de **SetConsoleCursorPosition () ;**)

- ✓ Segundo parámetro: Estructura de datos que tiene dos variables, estas dos variables son una **X** y una **Y**. Cuyos valores deben ser las coordenadas donde queremos situar el cursor.

Este parámetro pide un objeto de la estructura **Coord** la cual ya está definida en la biblioteca **Windows.h**.

Es importante que las variables sean mayúsculas. Por consiguiente es pasaremos la estructura como parámetro al **SetConsoleCursorPosition ();** de esta forma sus variables serán interpretados como coordenadas. Para poder mandar a imprimir solo colocaremos **printf** y el asterisco que colocamos al principio.

```
1 #include<stdio.h>
2 #include<windows.h>
3
4
5 int main(){
6     printf("*");
7
8     HANDLE hCon;
9     hCon = GetStdHandle(STD_OUTPUT_HANDLE);
10    COORD dwPos;
11    dwPos.X = 4;
12    dwPos.Y = 5;
13
14    SetConsoleCursorPosition(hCon, dwPos);
15    printf("+");
16
17    return 0;
18
19 }
```

(Figura 6. **dwPos**: Es la posición/coordenadas)

Ahora solo basta con compilar y ejecutar el programa. Y listo.