



Jose Luis Jimenez Urbano Enero de 2014 Programación
---

# Índice

1.- Introducción.....	3
2.- Obteniendo el código fuente.....	3
3.- Dependencias y requisitos previos.	4
4.- Compilando.....	7
5.- Estructuras de datos.....	10
5.1.- Variables Globales	
5.2.- Declaración de métodos y funciones	
6.- Cambios de la versión	
7.- Bugs	
8.- Bibliografía y Recursos	
8.1.- Bibliografía	
8.2.- Recursos	

# 1.-Introducción

Este es el manual técnico de la práctica de Stratego. En este manual vamos a tratar diferentes aspectos, desde la configuración de nuestro IDE para la compilación del código fuente, pasando por la explicación de las variables, procedimientos y funciones que utiliza nuestro código.

# 2.-Obteniendo el código fuente

Podemos conseguir el código fuente de este juego, “forkeando” el repositorio inferior directamente desde GitHub.

Este juego, así como su código está licenciado bajo licencia MIT.

<https://github.com/ESAT-jimenezur/Stratego.git>



## 3.- Dependencias y Requisitos Previos

Para poder compilar con éxito este juego, necesitamos varias cosas.

- Un IDE para C++ (Por ejemplo, CodeBlocks)
- Un compilador de C++ (Nosotros hemos usado GCC)
- Configurar nuestro compilador con las librerías incluidas en la carpeta “Libs”
- Configurar nuestro compilador con la librería “iJos.h”, descargable desde aquí: <https://github.com/ESAT-jimenezur/Programacion/tree/master/Libs>

## 4.- Compilando

Una vez configurado lo anterior, deberíamos poder compilar y ejecutar sin problemas.

## 5.- Estructuras de Datos

### 5.1.- Variables Globales

Estas son las variables globales, declaradas en la parte superior de "main.cpp" y usadas durante el resto del programa.

`bool debug = false;`

Esta variable se utiliza en el método "modoDebug" en la línea 1042. Si esta está activa, llamamos al modo Debug desde la línea 1263. Este método lo explicaremos más adelante.

`bool acercaDe = false;`

Esta variable sirve para mostrar en el menú principal la frase de "acerca De". Si está activa lo mostramos, sino no.

`bool instrucciones = false;`

Esta variable es igual que la anterior, pero con la frase de "instrucciones".

`bool seEstaJugando = true;`

Esta variable controla el flujo principal del método, si está a true, el juego sigue ejecutándose, sino no.

Esta variable pasa a false en el momento que un jugador se queda sin fichas o sin bandera.

`int posInicialCursorX = 0;`

Esta variable define la posición (En el eje X) del cursor, cuando empieza la partida. (Por comodidad, la partida empezará remarcando la casilla 1 de las fichas del jugador, para facilitar la partida)

`int posInicialCursorY = 42;`

Esta variable controla lo mismo que la anterior, pero en el eje (Y).

`int cursorX = posInicialCursorX;`

`int cursorY = posInicialCursorY;`

Iniciamos estas variables con el valor de las anteriores (Una copia, para no modificar las anteriores). Estas variables las usamos más adelante (En el método de control de cursor) para posicionar una ventana.

`int posActualX = 0;`

`int posActualY = 0;`

Estas dos variables controlan el eje X e Y de la posición actual. La vamos actualizando constantemente, para saber en que lugar estamos.

`int posFichaSeleccionadaX = 0;`

`int posFichaSeleccionadaY = 0;`

Aquí guardamos la posición de la ficha que "seleccionamos"



```
int arrayTemporalJugada[4][10] = {  
    0,0,0,0,0,0,0,0,0,0,  
    0,0,0,0,0,0,0,0,0,0,  
    0,0,0,0,0,0,0,0,0,0,  
    0,0,0,0,0,0,0,0,0,0  
};
```

Este array se utiliza para “traer” el tipo de jugada desde el fichero “jugadas.h”

## 5.2.- Declaración de métodos y funciones

En este juego, se ha decidido hacer declaraciones adelantadas de todos los métodos y funciones, para evitar problemas con las relaciones circulares existentes entre diferentes funciones y procedimientos.

De modo que a partir de la línea 80, tenemos definidas todas las funciones que utilizaremos más adelante.

Estas son las funciones utilizadas.

```
void setColors(int texto, int fondo);
```

Esta función toma 2 parámetros, el color del fondo, y el color del texto, lo único que hace es ejecutar la función “colorTexto” pasándole los parámetros anteriores. La función “colorTexto” se encuentra en la librería iJos.h

```
void menuNuevaPartida();
```

Esta función se encarga de mostrar el menú de nueva partida. Se llama desde el menú normal, o desde la pantalla de fin de partida.

Se encarga de recoger un valor numérico por entrada de teclado, y ejecutar una acción.

```
void menuOpciones();
```

Este método, es igual que los anteriores, pero muestra el menú de opciones.

```
void menu();
```

Este es el primer menú en mostrarse al ejecutar el juego. Se llama en el método main();

```
void pantallaResultados(int ganador);
```

Este método se encarga de mostrar una pantalla final (de resultados) según lo que pase durante la partida.

Recibe como entero el ganador de la partida. Y según esto, cambia el color de toda la ventana a verde/rojo y muestra una frase u otra.

```
void setFichas(int tipoJugada);
```

Este método, recibe por parámetro el número de la jugada a establecer.

Luego utiliza el array de la jugada precargada (en jugadas.h) y lo inserta en el tablero.

```
void pintaCasillasInaccesibles();
```

Este método solo se encarga de pintar las casillas inaccesibles (agua).

```
void pintaParrilla();
```

Este método pinta la parrilla de juego, se le llama al inicio, y cuando necesitamos refrescar los datos del tablero.

```
void movimientoCursor();
```

Este método se encarga de diferentes cosas.

En primer lugar captura (Durante el turno del jugador) las teclas del teclado, permitiendo mover el cursor, y seleccionar fichas con la tecla ENTER.

Además de eso, también controla el movimiento de las fichas del jugador, cogiéndolas, depositándolas y verificando que la posición donde se dejan es correcta. También realiza combates, si es necesario.

```
void movimientoPC();
```

Este método, se encarga de controlar el movimiento de las fichas del PC.

Realiza varias acciones:

Primero, se centra en sacar una ficha aleatoria, de la primera fila (La más inferior de todas las del PC). (Aumenta en 1 el contador de Fichas de la primera fila)

Si ha movido menos de 5 fichas de la primera fila, no se plantea pasar a otra fila. Así sucesivamente.

Cuando tiene una ficha, comprueba si esta se puede mover hacia adelante, sino puede moverse, comprueba ambos lados. Si en algún momento puede moverse, se moverá.

Cuando acaba de todo, cambia el turno.

```
void compruebaCombate(int turno, int valorAtacante, int posX_Inicio, int posY_Inicio, int posX_Destino, int posY_Destino);
```

Este método se encarga de comprobar el combate.

Recibe por parámetro el turno.

Hace las comprobaciones necesarias para saber quien gana el combate. También resta fichas al jugador que pierda la partida.

```
void modoDebug();
```

Este método, únicamente se activa si en las opciones hemos marcado "ON" al modo Debug.

Se encarga de mostrarnos más datos además de los habituales. (Útil para programadores).

```
void logs(int log);
```

Este método se encarga de mostrar mensajes de Log (Eventos) durante la partida. Hay varios tipos, por ejemplo si intentamos coger la ficha del enemigo, irnos fuera del tablero...

Recibe por parámetro el tipo de log que tiene que mostrar

```
void logFichas();
```

Este método es como los anteriores, pero muestra un log de las fichas de cada jugador, así como las banderas etc...

```
void juego(int tipoInicio);
```

Este método controla el flujo del juego, ejecuta por orden los diferentes métodos, hasta que se cumplan las condiciones para finalizar partida. En ese caso, nos mostrara la pantalla de resultados, indicando el ganador.

```
int main();
```

Este es nuestro método principal, se encarga de dar tamaño y nombre a la ventana de juego, y de mostrar el menú



## 6.-Cambios de la versión

### 1.01 prealpha (01/2014)

- En esta versión, se han integrado las principales funcionalidades del juego

## 7.-Bugs

Aquí se indicarán los diferentes tipos de bugs encontrados en el juego, para reparar en futuras actualizaciones:

Si se encuentra algún tipo de error, se recomienda abrir un nuevo “ticket” en el repositorio. (<https://github.com/ESAT-jimenezur/Stratego/issues>)

- Las fichas del PC no atacan, solo esquivan
- Cuando acaba la partida, si se selecciona la opción de rejugar, aparecen errores (Hay que reiniciar las variables globales)
- Hay que llevar un recuento de las fichas de ambos jugadores, para actualizar el log del HUD

## 8.- Bibliografía y Recursos

Para la realización de este proyecto, se han consultado diversas fuentes, y se han utilizado numerosos recursos, los más destacables son:

### 8.1.-Bibliografía

- <http://es.wikipedia.org/wiki/Stratego>
- <http://www.ultrastratego.com/setups.php> (Setups)
- <http://stackoverflow.com/>

### 8.2.-Recursos

- <https://github.com/> (Control de versiones)
- CodeBlocks (IDE)
- GCC (Compilador de C)
- OpenOffice (Editor de textos libre y gratuito)
- Y mucho café :)

