

PROYECTO : IMPLEMENTACIÓN HUNDIR LA FLOTA

VERSIÓN BÁSICA – PRIMERA PARTE

Vamos a programar una versión básica del juego Hundir la Flota, en la que el jugador se enfrentará al ordenador.

Jugaremos en un tablero de 10 x 10 posiciones en el que indicaremos la fila mediante una letra (A-J) y la columna mediante un número(0 – 9), del siguiente modo.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[A]
[B]	.	5	5	5	5	5
[C]	1	.	3	.	1	.	.	.	3	.
[D]	.	.	3	1	3	.
[E]	.	.	3	.	3	.	.	.	3	.
[F]	3
[G]	.	.	1	.	3	.	1	.	.	.
[H]
[I]	.	5	5	5	5	5
[J]

¿EN QUÉ CONSISTE EL JUEGO?

El juego consiste en que ambos jugadores inicien la partida colocando varios barcos en posición horizontal y vertical, para a partir de ese momento ir indicando mediante coordenadas la casilla a la que deseamos disparar, a lo que el jugador contrario nos informará de si hemos dado a un barco o el disparo cae en el agua.

¿CÓMO SERÁ NUESTRA IMPLEMENTACIÓN?

- ✓ Barcos a colocar en el tablero: 2 barcos de 5 casillas, 3 de 3 casillas y 5 de 1 casilla (24 en total).
- ✓ En el juego original en cada disparo se indica Agua, Tocado y/o Hundido, esto último cuando se haya alcanzado a todas las posiciones del barco.
- ✓ En nuestra versión esta parte se simplificará y el ordenador sólo indicará si el disparo ha sido Agua o Tocado. Para tener información de la cantidad de posiciones que llevamos acertadas en cada momento se irán mostrando en nº de puntos del cada jugador.
- ✓ Cada jugador partirá teniendo 24 puntos ($5*2 + 3*3 + 5*1$) y se irán restando 1 punto cada vez que se acierte una posición.
- ✓ También a diferencia del juego original en el nuestro cada jugador realiza un solo disparo y pasa a disparar el otro, no repite hasta fallar.
- ✓ El juego acabará cuando alguno de los jugadores llegue a 0 puntos.
- ✓ Nosotros al jugar contra el ordenador no podremos ver como es lógico dónde se encuentran los barcos en el mapa del oponente, pero si iremos viendo cómo está nuestro mapa y dónde han sido nuestros disparos en su mapa en el mapa del nuestro oponente (el ordenador), del siguiente modo:

Esta sería la información que veríamos en los mapas.

MAPA DEL ORDENADOR

NUESTRO MAPA

REGISTRO DEL MAPA DEL ORDENADOR

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[A]
[B]	.	.	.	A	.	A	.	A	A	A
[C]	A	.	.	A	.	.
[D]	.	A
[E]	A
[F]	X
[G]	X
[H]	X
[I]	A
[J]

PUNTOS RESTANTES DEL ORDENADOR: 21

MAPA DEL USUARIO

	[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]
[A]	X
[B]	5	X
[C]	5	.	5	1	.
[D]	5	.	5	X	3	3	.	.	.	A
[E]	5	.	5	.	X
[F]	.	.	5	.	A	X	X	3	.	.
[G]	.	.	5	.	.	.	X	.	.	.
[H]	3	X	.	.
[I]	3	.	X	.
[J]	A	.	A

PUNTOS RESTANTES DEL JUGADOR: 15

PROGRAMACIÓN DEL JUEGO

- ✓ Declarar las siguientes variables final de tipo char:

- AGUA='A'
- AGUA_NO_TOCADO='.'
- TOCADO='X'
- TAMANNIO=10 (de tipo entero)

Las vamos a declarar fuera del método main() para poderlas usar en cualquier método del programa, al igual que el objeto de la clase Scanner.

- ✓ Iniciamos el juego declarando dos variables contador de tipo entero que inicializaremos con 24 pts, una contabilizará los puntos d el ordenador y otra los del jugador(nosotros).
- ✓ Tendremos 2 variables boolean, juegoTerminado y tiroCorrecto, que tendrán valor inicial false.
- ✓ Crearemos 3 mapas: mapaUsuario, mapaOrdenador, mapaOrdenadorParaUsuario de tipo char[TAMANNIO][TAMANNIO] que declararemos como Arrays bidimensionales dentro del main().
- ✓ Situaremos los barcos en mapaUsuario y mapaOrdenador generando las posiciones de forma aleatoria mediante un método que describiremos más adelante.
- ✓ Nuestro proceso se terminará cuando la variable juegoTerminado sea true.
- ✓ Cada tiro lo registraremos en un array[2] de enteros.
- ✓ Inicializaremos los mapas del USUARIO Y DEL ORDENADOR de forma aleatoria en un método al que se le pasan los dos mapas.
- ✓ En cada iteración del bucle principal se realizará un disparo del jugador y otro del Ordenador
- ✓ Si tras la tirada del jugador, este hunde todos los barcos, el ordenador no juega, lo saltamos.

MÉTODO DE INICIALIZACIÓN DE MAPAS:

- ✓ Programaremos un método al que llamaremos inicializacionMapa que recibirá como parámetros un array bidimensional de tipo char y no devolverá nada.
- ✓ Rellenará todos las celdas del array con el carácter AGUA_NO_TOCADO.

MÉTODO DE REGISTRO DE LOS BARCOS EN LOS MAPAS DE JUGADOR Y ORDENADOR:

- ✓ Programaremos un método al que llamaremos registraBarcos que recibirá como parámetros un array bidimensional de tipo char y no devolverá nada.
- ✓ A continuación, colocaremos los distintos barcos en posiciones aleatorias del siguiente modo:
 - Declaramos dos vectores uno de enteros con los barcos a situar, cuyo contenido será {5,5,3,3,3,1,1,1,1,1}
 - Otro de char para la dos posibles direcciones de colocación de los barcos `V` o `H`, para indicar vertical o horizontal respectivamente.
- Hacemos un bucle que recorra el vector de barcos uno a uno.
- Por cada barco, programaremos un bucle que se repetirá mientras el barco no esté colocado en su totalidad.
 - Primero generamos aleatoriamente una fila(0-9), una columna(0-9) y una dirección 0 ó 1, asociado a horizontal y vertical respectivamente.
 - A continuación, veremos si el barco, según su tamaño, cabe en la dirección que ha salido, contando desde la posición hacia la derecha o hacia abajo según el caso. $\text{fila} + (\text{tamaño del barco}) \leq \text{TAMANNO} - 1$
 - Si el barco cabe en la dirección, tendremos que ver si se solapa con otro. Esto tendremos que comprobarlo posición a posición.
 - Si tras estas dos comprobaciones no hay inconveniente, poner la anotación correspondiente en el mapa, sino descartarlo y generar otro para ese mismo barco.

```
MAPA DEL USUARIO
  [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
[A]  1  .  .  .  .  1  .  .  .  .
[B]  .  3  3  3  .  .  .  .  .  .
[C]  .  .  5  .  .  .  5  .  .  .
[D]  .  .  5  1  .  .  5  .  .  .
[E]  .  .  5  .  3  .  5  .  .  .
[F]  .  .  5  .  3  .  5  .  .  .
[G]  3  .  5  .  3  .  5  .  .  .
[H]  3  .  .  .  .  .  .  1  .  .
[I]  3  .  .  .  .  .  .  .  1  .
[J]  .  .  .  .  .  .  .  .  .  .
PUNTOS RESTANTES DEL JUGADOR: 24
TURNO DEL JUGADOR
Introduzca la casilla (por ejemplo B4): |
```

○ La anotación se realiza en el mapa rellenando las celdas con el tamaño del barco en todas sus posiciones, por ejemplo 333 para un barco de 3 celdas en horizontal, como se puede ver en el mapa de ejemplo.

- ✓ Este método lo tenemos que ejecutar para situar los barcos en el mapaJugador y mapaOrdenador. Ambos arrays se le pasarán por separado como argumento al método desde el main().

MÉTODO PARA VISUALIZAR LOS MAPAS

- ✓ Programaremos un método al que llamaremos `imprimirMapa` para visualizar un mapa que recibirá como parámetro y el cual no devolverá nada.
- ✓ El método además del contenido del array mostrará las cabeceras de filas y columnas como indican los ejemplos.
- ✓ Podemos crear un vector de `char` con las letras que corresponden a cada fila, lo que nos facilitará el trabajo.