



Batalla Naval por comparación de elementos de arreglos

Semestre 2021-1

Profesor: M.I. Marco Antonio Martínez
Quintana

Fundamentos de Programación
Elaboración Nov 2020-Ene 2021



28 DE ENERO DE 2021
FACULTAD DE INGENIERIA UNAM
Universidad Nacional Autónoma de México

Resumen:

Mi proyecto se basa en la comparación de arreglos utilizando estructuras de control y de repetición anidadas unas dentro de otras. Así mismo se utilizan las funciones de la librería stdio para que el programa pueda recopilar los datos almacenarlos y mostrar al usuario qué es lo que está sucediendo de manera concisa y clara.

El programa podría resumirse en 3 grandes pasos, que serían menú, obtención de coordenadas y finalmente comparación.

El primer paso consta de un ciclo do while con un switch y una serie casos dentro de manera que el usuario puede visualizar las opciones que tiene para iniciar el juego las cuales son: Jugar, instructivo y salir, estas son demandadas a partir de un scanf el cual guarda un número entero que representa la opción.

La obtención de coordenadas también consta de un ciclo do while el cual a su vez entra a un switch para ir cambiando de barcos según las coordenadas ingresadas, cabe recalcar que el usuario solo ingresará 2 coordenadas de cada barco y la computadora calculará las demás en función del espacio que hay y en caso de que exceda los límites o ingrese 2 coordenadas que no generen un barco mandará un mensaje de error y recomenzará la recolección de coordenadas, y finalmente las guardará en distintos arreglos que representan las coordenadas enteras de un barco o su posición en el tablero como carácter.

Finalmente la comparación también se vale de un ciclo do while el cual se romperá hasta haber un ganador, dentro de este habrá otro ciclo do while que podría ser representado con los turnos de los jugadores y se irá rompiendo al no encontrar coincidencias entre las coordenadas que ingresemos con las guardadas, su ejecución es simple comparando los números enteros de las componentes de las coordenadas guardadas con las ingresadas y saltando casos en caso de que esté "hundido" el barco, finalmente compara cuantos barcos hay hundidos y si hay o no ganador en ese turno y quién es.

Por otro lado, los tableros son mostrados con un arreglo de caracteres en los cuales la "X" representa una posición del barco, una doble línea una posición encontrada y una "O" un lugar vacío.

También hay comandos repartidos a lo largo del programa los cuales se encargan de cambiar el color de la letra esto con el fin de que los usuarios sean capaces de distinguir entre turnos de una manera más empírica como lo es observar el color de las letras

Introducción:

Dentro del primer semestre de la carrera de ingeniería en computación es imprescindible que los alumnos salgan con los fundamentos mínimos de programación y no hay mejor herramienta para comprenderlos que el lenguaje C clásico y una serie de ejercicios.

El lenguaje C es un lenguaje de bajo nivel el cual logra que el programador y la computadora se entiendan a través de una serie de instrucciones predeterminadas.

A los arreglos hechos con dichas instrucciones se les llama programas, estos pueden hacer cualquier cosa que el programador sea capaz de expresar en las instrucciones predeterminadas y una serie de operaciones que se realicen dentro de este de forma que el alcance de este programa es prácticamente ilimitado o bien limitado por las capacidades del compilador, la computadora y la imaginación del programador.

Las instrucciones del lenguaje c pueden ser divididas fácilmente en 3 categorías las cuales serían funciones, son todas aquellas operaciones que se realizan matemática cómo gráficamente, por ejemplo, el pedir o imprimir datos y que dentro de la computadora sea capaz de operarlos o disponerlos en un arreglo específico. Estructuras de repetición, son todas aquellas cuya función es cumplir ciclos, esto en aras de que el programa sea de menor tamaño, repetir indefinidamente funciones, así cómo evitarle al programador escribir cada caso individualmente. Finalmente, las estructuras de control permiten generar una serie de caminos o condiciones para que ciertas estructuras o funciones sean ejecutadas o no de forma que en un solo programa somos capaces de distribuir una serie de casos específicos cuando lo pidamos específicamente logrando así que tanto el usuario como el programador junte 2 programas similares en un solo y gran programa sin la necesidad de ejecutar uno y luego otro.

Para poder programar hay que comprender cómo la computadora logra entender lo que le escribimos.

Primeramente es importante recalcar que los programas se leen de izquierda a derecha y de arriba abajo siempre, no es posible romper ese orden de alguna manera, luego hay que considerar que en caso de haber una estructura de control el programa elegirá el camino que se adapte a los datos que tiene y continuará sin regresar a ese punto al menos que esté anidada a una estructura de repetición, por otro lado las estructuras de repetición se seguirán repitiendo hasta que un parámetro sea o no cumplido, cabe recalcar que hay que tener extremo cuidado de que la computadora comprenda cuando se entra o no a una estructura de repetición, de lo contrario el programa podría quedar completamente atorado hasta que se forcé su salida lo cual lo hará completamente inutilizable.

La computadora podrá leer cualquier dato que sea correctamente declarado y almacenado de otra forma es posible que el programa no pueda ser compilado o de lo contrario que los valores no sean guardados de manera correcta generando un fallo en el programa y dejándolo inutilizable.

A estos datos se le llamarán variables, estas podrán ser de tipo entero, decimal, booleano o de tipo carácter representados por sus respectivas notaciones.

Las variables pueden almacenarse en arreglos los cuales son analógicamente cajas de variables o variables de variables, estos podrán tener una cantidad n de dimensiones las cuales determinará el programador y en ellas serán guardadas las variables según un arreglo matricial donde el primer número se refiere a la línea y el segundo a la columna, se deberá tener sumo cuidado a la hora de almacenar una variable en un arreglo considerando si la posición es la correcta y si el tipo de arreglo coincide con el tipo de variable la cual es almacenada, de lo contrario no se podrá guardar o en su defecto se guardará como su equivalente en el tipo de variable que sea pudiendo entorpecer el manejo de estos datos a la hora de realizar el programa.

A partir de estos conceptos básicos los cuales fueron enseñados a lo largo del semestre pude ser capaz de aplicar la mayoría en un programa que simula al juego de mesa batalla naval.

Desarrollo:

Descripción general del proyecto:

El objetivo de mi proyecto es sobre todo agrupar la mayoría de los conocimientos aprendidos a lo largo del semestre para poder realizar un programa funcional el cual sea capaz de interactuar con los usuarios de este, de manera que se vean reflejados los conocimientos obtenidos a lo largo del semestre y un uso práctico de estos.

Su valor más que comercial radica en el ámbito educativo, ya que puede ser utilizado para poder enseñar a otros cómo funcionan los conceptos básicos de la programación así mismo esto se hace de una manera didáctica y amena que además impulsa a los otros a tratar de hacer algo similar o comenzar a programar.

Por lo tanto, el alcance de este proyecto es desde un ámbito educativo cómo material didáctico hasta simplemente la recreación de sus usuarios al poder jugar en su computadora el clásico juego con un compañero.

Problema y algoritmo de solución:

El principal problema que se presenta en este programa es la comparación de datos almacenados de una forma cíclica relativamente sencilla y más corta de lo que sería compararlos uno a uno con su instrucción cada uno.

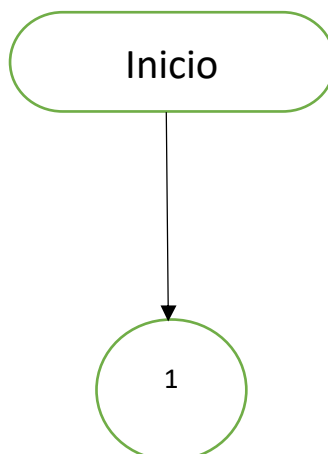
Algoritmo de comparación de coordenadas ingresadas con coordenadas guardadas:

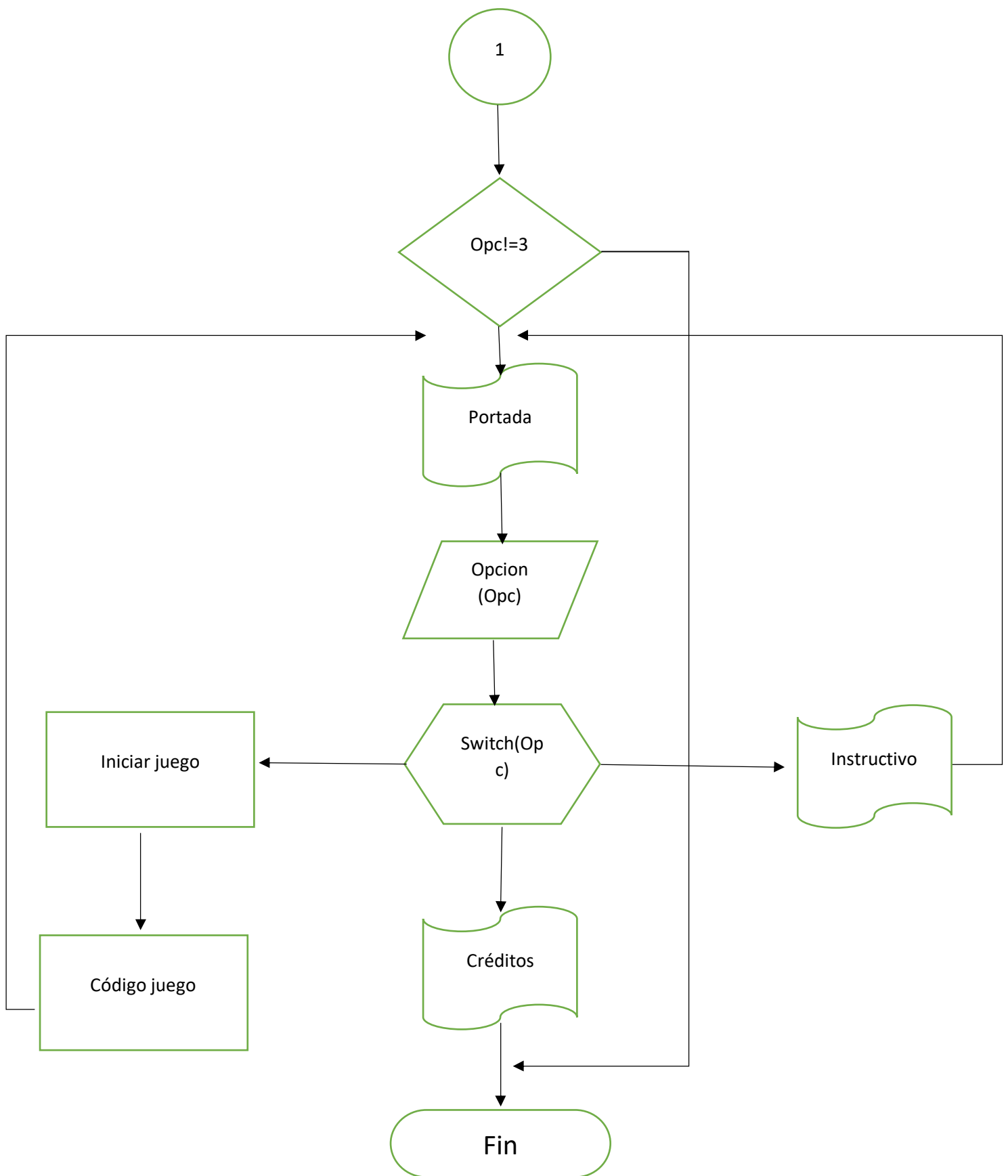
1. Se solicitan las coordenadas
2. Se selecciona el primer caso el cual se comparará
3. Se comprueba si la primera componente de la coordenada es igual a la primera componente que ha sido guardada anteriormente.
4. Si la componente fue igual se procederá a comparar si la siguiente componente de la coordenada guardada con la coordenada ingresada
5. Si la componente fue igual se procederá a marcar dicha posición en el tablero con la coordenada que fue ingresada
6. De otra manera se procederá a buscar concordancia con la siguiente posición del primer caso y se repetirá
7. De no haber encontrado ninguna concordancia con ninguna posición en dicho caso pasará al siguiente caso y se ejecutará el mismo procedimiento anterior hasta llegar al ultimo caso
8. Si el ultimo caso sigue sin resultar con concordancias se procederá a marcar esa coordenada cómo fallida y se mandará un mensaje de que el dato no estaba en dicha coordenada dándole el paso al siguiente jugador.

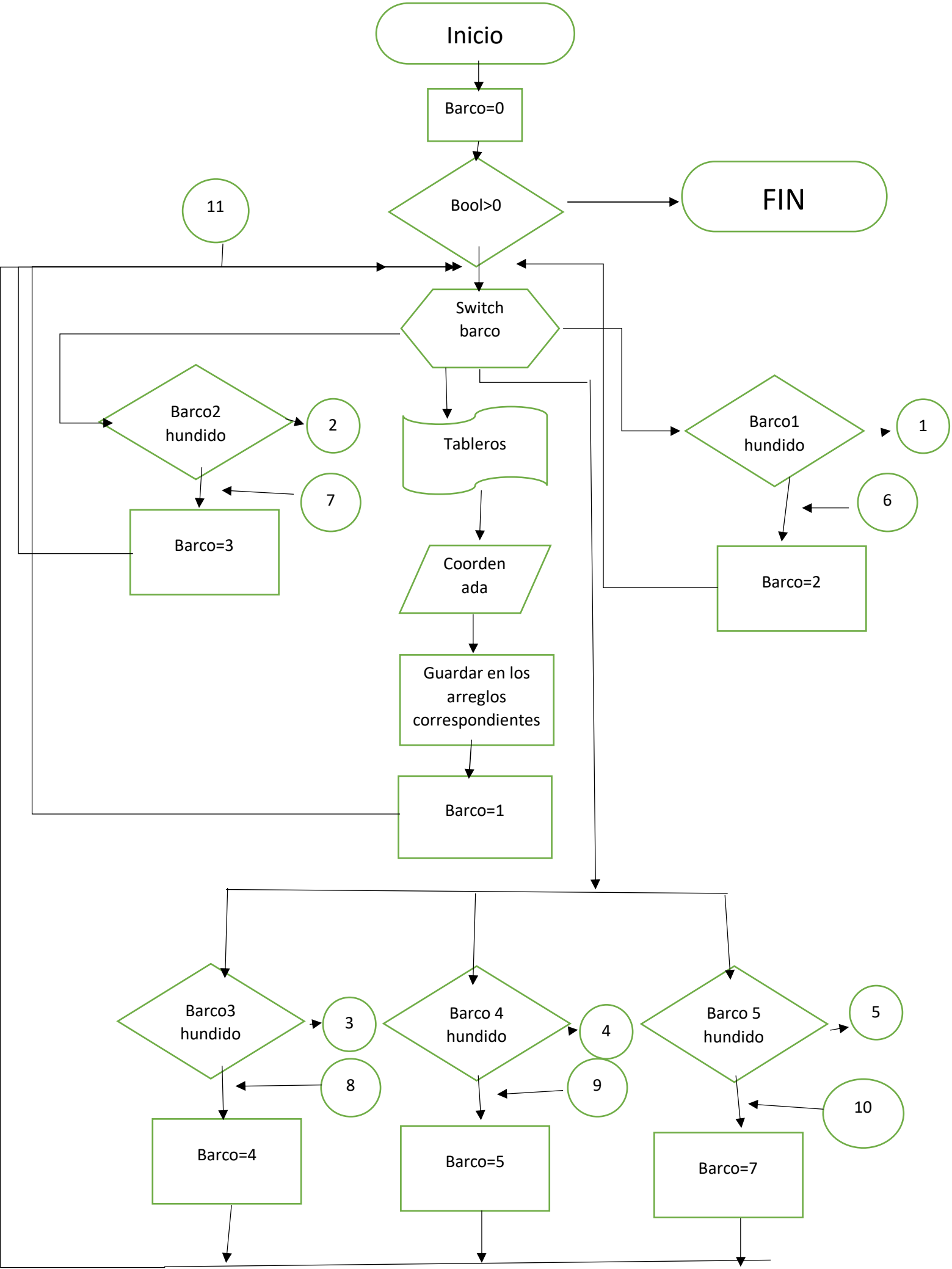
Diagramas de flujo:

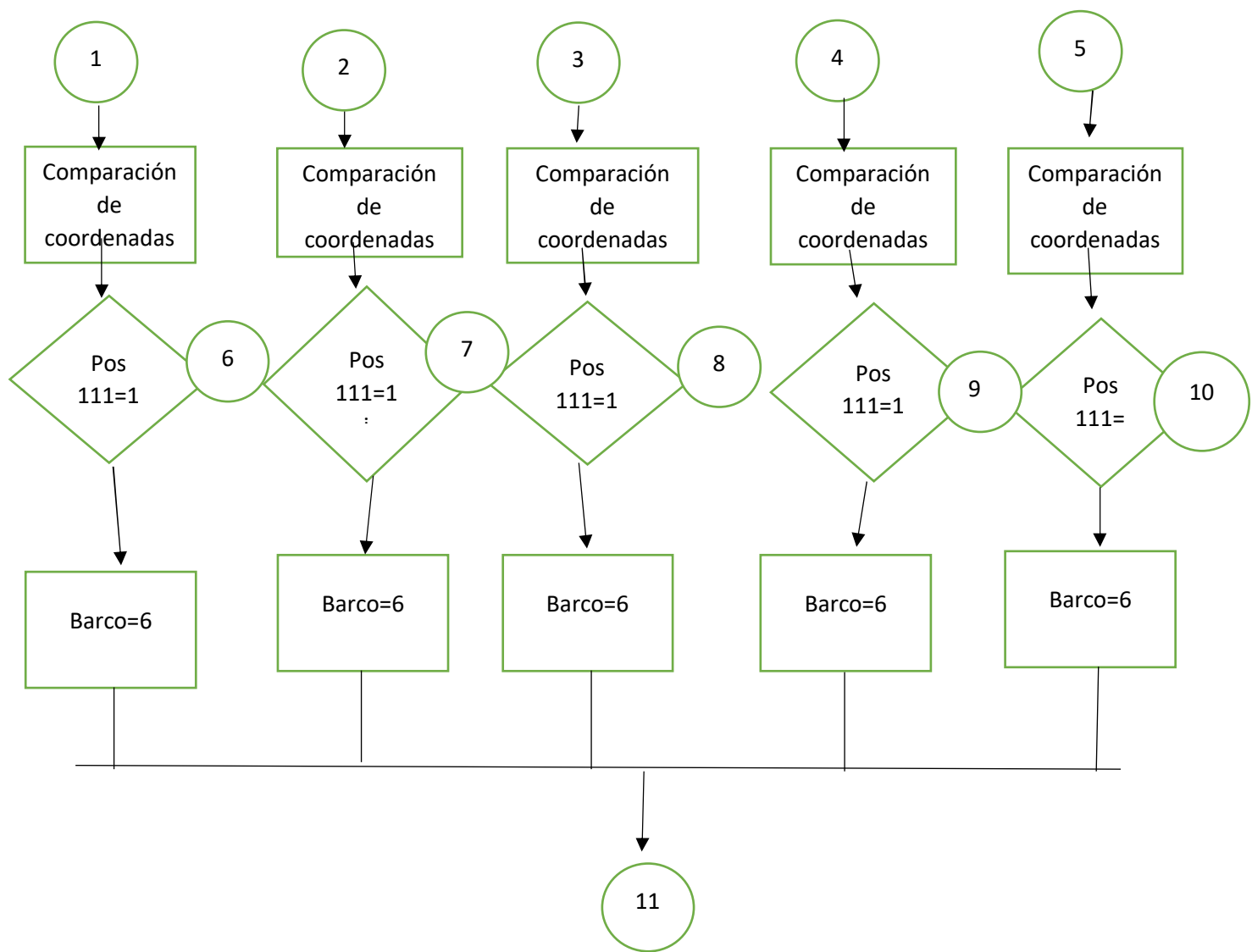
Parcial del menú de todo el juego:

Y parcial de comprobación de coordenadas de primer barco:









Pseudo código:

Para pedir coordenadas del primer barco:

Hacer mientras la variable $cor1 > 7$

[Seleccionar la variable cor

Caso 1 valor 1=>

[Color de la letra verde claro

Escribir coordenadas del primer jugador

Escribir inserta las coordenadas de tus barcos divididas por comas

Escribir primera

Leer y guardar a y b

Escribir segunda

Leer y guardar c y d

Si $d=b-1$ y $a=c$ y $b \geq 5$

[Escribir "la tercera coordenada es a, b-2 "

Guardar tercera coordenada

Escribir "la cuarta coordenada es a, b-3 "

Escribir "la quinta coordenada es a, b-4"

Guardar la cuarta coordenada

Cor=2

Parar hasta que se toque una tecla]

Fin si

Si $d==b+1$ && $a==c$ && $b \leq 6$

[Escribir "la tercera coordenada es a, b+2 "

Guardar tercera coordenada

Escribir "la cuarta coordenada es a, b+3 "

Guardar cuarta coordenada

Escribir "la quinta coordenada es a, b+4"

Guardar la quinta coordenada

Cor=

Parar hasta que se toque una tecla]

Fin si

Si $c==a-1 \ \&\& \ b==d \ \&\& \ a \geq 5$

[Escribir "la tercera coordenada es a-2, b "

Guardar tercera coordenada

Escribir "la cuarta coordenada es a-3, b "

Guardar tercera coordenada

Escribir "la quinta coordenada es a-4, b"

Guardar la quinta coordenada

Cor=2

Parar hasta que se toque una tecla]

Fin si

Si $c==a+1 \ \&\& \ b==d \ \&\& \ a \leq 6$

[Escribir "la tercera coordenada es a+2, b "

Guardar tercera coordenada

Escribir "la cuarta coordenada es a+3, b "

Guardar tercera coordenada

Escribir "la quinta coordenada es a+4, b"

Guardar la quinta coordenada

Cor=2

Parar hasta que se toque una tecla]

Fin si

Si cor $\neq 2$

[cor=6]

Fin si

Borrar la pantalla]

Caso 2=> cor2

[....]

Caso 3=> cor3

[....]

Caso 4=> cor4

[....]

Caso 5=> cor5

[Escribir "Coordenadas del quinto barco "

Escribir "primera"

Leer q y r

Escribir "segunda"

Leer s y t

Guardar los valores

Cor=8

Parar hasta que se toque una tecla]

Caso 6=> cor 6

[

Borrar la pantalla

Cambiar color de letra a rojo

Escribir "Tu barco no puede tener esas coordenadas,
tendr s que volver al principio "

Cor=1

Parar hasta que se toque una tecla]

]

[C digo completo comentado tanto c mo se pudo \(Link al
archivo.c en git hub para que sea m s sencillo de leer en los
anexos\):](#)

// Batalla naval con matrices

#include <stdio.h>

#include <windows.h> // permite borrar la pantalla y cambiar el color

#include <conio.h> // permite el getch

#include <stdbool.h>

// proyecto final en c videojuego batalla naval

```

int main()
{

    char aa=160, ai= 161, ao=162, au=163, tl= 240, esi= 201, esd= 187, lv=
186, eii= 200, eid= 188, an= 164;

    int OP,cor, cor1, barco, barco2, Bool, a, b, c, d, e, f, g, h, i, j, k, l, m, n, o,
p, q, r, s, t, u, v, x, y;

    do // pantalla de bienvenida
    {

        system("Color 07"); // color blanco de letra

        system ("cls");

        printf("\n          *          *****          **** **
**          **** ");

        printf("\n          *          *** **          *****          ****
**      **          ***** ");

        printf("\n          *          *** **          * **          **          * ** **          **
* ** ");

        printf("\n          %c*%c          *****          *****          **          *****
**      **          ***** ", tl, tl);

        printf("\n          %c%c*%c%c          *** **          *****          **
***** **          **          ***** ", tl, tl, tl, tl);

        printf("\n          %c*%c          *** **          ***          **          ***          **
*****          *****          ** ", tl, tl);

        printf("\n          *          *****          ***          **          ***          **          *****
*****          **          ** ");

        printf("\n %c-----*-----%c
", esi, esd);

        printf("\n %c          *          %c
", lv, lv);

        printf("\n %c          *          %c          *****          **          *****          **
**** **          ", lv, lv);

        printf("\n %c          *          %c          ** **          **          *****          **
*****          **          ", lv, lv);

```

```

printf("\n %c * %c          ** ** ** * ** ** ** *
** ** ", lv, lv);

printf("\n %c * %c          ** ** ** ***** ** **
***** ** ", lv, lv);

printf("\n %c * %c          ** ** ** ***** ** **
***** ** ", lv, lv);

printf("\n %c * %c          ** ***** ** ***** ***
** ***** ", lv, lv);

printf("\n %c * %c          ** *** ** ** *** ***
** ***** ", lv, lv);

printf("\n %c * %c
", lv, lv);

printf("\n ---%c=====*====== %c---
",eii, eid );

printf("\n %c%c%c%c%c%c%c%c%c ==
Jugar.....(1) ", lv, lv, lv, lv, lv, lv, lv, lv);

printf("\n %c%c%c%c%c%c%c%c %c %c%c%c%c%c%c==
Instructivo.....(2) ", lv, lv, lv, lv, lv, lv, lv, lv, lv, lv, lv, lv);

printf("\n%c %c == %c          Salir.....(3)
", lv, lv, lv);

printf("\n%c %c == %c ==
", lv, lv, lv);

printf("\n%c %c ===
", lv, lv);

printf("\n%c %c %c
", lv, lv, lv);

printf("\n%c %c %c
", lv, lv, lv);

printf("\n%c %c %c
", lv, lv, lv);

printf("\n %c%c %c %c%c
", lv, lv, lv, lv, lv);

printf("\n %c%c%c%c%c %c %c%c%c%c
", lv, lv, lv, lv, lv, lv, lv, lv, lv);

printf("\n %c%c%c%c%c%c%c%c          Free software made
by Fr%cas on a busy afternoon 2021 ", lv, lv, lv, lv, lv, lv, lv, lv, ai);

scanf ("\n %i",&OP);

```

```

switch(OP)//menú de opciones
{
    case 2: //instructivo
        system("Color 07"); // color blanco de letra
        system ("cls"); // Borrar todo lo que había en pantalla
        printf ("\n \t Bienvenido al basto y ancho mar podr%c
no parecer muy grande pero tanto esconderse como buscar requiere cierta
habilidad y mucha suerte",aa);

        printf(" \n \n \t Las instrucciones son simples el
tablero de arriba representa un mapa donde est%c tu enemigo el de abajo es tu
mapa donde estar%cs t%c.", aa, aa, au);

        printf (" \n \n \t Escribe las coordenadas donde te
ubicas y reza porque no te encuentren despues preparete para de tratar evitar
que vuelva a intentarlo de nuevo escribiendo las coordenadas donde podria
estar");

        int ejemplo [11][11] ={{0,1,2,3,4,5,6,7,8,9,10},
{1,0,0,0,0,0,0,0,0,0,0}, {2,0,0,0,0,0,0,0,0,0,0}, {3,0,0,0,0,0,0,0,0,0,0},
{4,0,0,0,0,0,0,0,0,0,0}, {5,0,0,0,0,0,0,0,0,0,0}, {6,0,0,0,0,0,0,0,0,0,0},
{7,0,0,0,0,0,0,0,0,0,0}, {8,0,0,0,0,0,0,0,0,0,0}, {9,0,0,0,0,0,0,0,0,0,0},
{10,0,0,0,0,0,0,0,0,0,0}};

        int i,j,k,l, OP;
        printf("\n \n ");
        for(i=0;i<11;i++)
        {
            for(j=0;j<11;j++)
            {
                printf("\t %i ",ejemplo[i][j]);
            }
            printf ("\n");
        }

        printf ("\n \n"); // espacio entre los tableros
        system("Color 01"); // color azul de letra
        for(i=0;i<11;i++)

```

```

    {
        for(j=0;j<11;j++)
        {
            printf("\t %i ",ejemplo[i][j]);
        }
        printf("\n");
    }

    system("Color 07"); // color blanco de letra
    getch ();

    break;

    case 1: // código de todo el juego

    system ("cls");

```

```

    char matriz1 [11][11] ={{'-','A','B','C','D','E','F','G','H','I','J'},
    {'A','O','O','O','O','O','O','O','O','O','O'}, {'B','O','O','O','O','O','O','O','O','O','O'},
    {'C','O','O','O','O','O','O','O','O','O','O'}, {'D','O','O','O','O','O','O','O','O','O','O'},
    {'E','O','O','O','O','O','O','O','O','O','O'}, {'F','O','O','O','O','O','O','O','O','O','O'},
    {'G','O','O','O','O','O','O','O','O','O','O'}, {'H','O','O','O','O','O','O','O','O','O','O'},
    {'I','O','O','O','O','O','O','O','O','O','O'}, {'J','O','O','O','O','O','O','O','O','O','O'}};

```

```

    char matriz2 [11][11] ={{'-','A','B','C','D','E','F','G','H','I','J'},
    {'A','O','O','O','O','O','O','O','O','O','O'}, {'B','O','O','O','O','O','O','O','O','O','O'},
    {'C','O','O','O','O','O','O','O','O','O','O'}, {'D','O','O','O','O','O','O','O','O','O','O'},
    {'E','O','O','O','O','O','O','O','O','O','O'}, {'F','O','O','O','O','O','O','O','O','O','O'},
    {'G','O','O','O','O','O','O','O','O','O','O'}, {'H','O','O','O','O','O','O','O','O','O','O'},
    {'I','O','O','O','O','O','O','O','O','O','O'}, {'J','O','O','O','O','O','O','O','O','O','O'}};

```

```

    char matriz3 [11][11] ={{'-','A','B','C','D','E','F','G','H','I','J'},
    {'A','O','O','O','O','O','O','O','O','O','O'}, {'B','O','O','O','O','O','O','O','O','O','O'},
    {'C','O','O','O','O','O','O','O','O','O','O'}, {'D','O','O','O','O','O','O','O','O','O','O'},
    {'E','O','O','O','O','O','O','O','O','O','O'}, {'F','O','O','O','O','O','O','O','O','O','O'},
    {'G','O','O','O','O','O','O','O','O','O','O'}, {'H','O','O','O','O','O','O','O','O','O','O'},
    {'I','O','O','O','O','O','O','O','O','O','O'}, {'J','O','O','O','O','O','O','O','O','O','O'}};

```

```

    char matriz4 [11][11] ={{'-','A','B','C','D','E','F','G','H','I','J'},
    {'A','O','O','O','O','O','O','O','O','O','O'}, {'B','O','O','O','O','O','O','O','O','O','O'},
    {'C','O','O','O','O','O','O','O','O','O','O'}, {'D','O','O','O','O','O','O','O','O','O','O'},
    {'E','O','O','O','O','O','O','O','O','O','O'}, {'F','O','O','O','O','O','O','O','O','O','O'},
    {'G','O','O','O','O','O','O','O','O','O','O'}, {'H','O','O','O','O','O','O','O','O','O','O'},
    {'I','O','O','O','O','O','O','O','O','O','O'}, {'J','O','O','O','O','O','O','O','O','O','O'}};

```

```

int posicion1 [11][11] ={{0,1,2,3,4,5,6,7,8,9,10},
{1,0,0,0,0,0,0,0,0,0,0}, {2,0,0,0,0,0,0,0,0,0,0}, {3,0,0,0,0,0,0,0,0,0,0},
{4,0,0,0,0,0,0,0,0,0,0}, {5,0,0,0,0,0,0,0,0,0,0}, {6,0,0,0,0,0,0,0,0,0,0},
{7,0,0,0,0,0,0,0,0,0,0}, {8,0,0,0,0,0,0,0,0,0,0}, {9,0,0,0,0,0,0,0,0,0,0},
{10,0,0,0,0,0,0,0,0,0,0}};

```

```

int posicion2 [11][11] ={{0,1,2,3,4,5,6,7,8,9,10},
{1,0,0,0,0,0,0,0,0,0,0}, {2,0,0,0,0,0,0,0,0,0,0}, {3,0,0,0,0,0,0,0,0,0,0},
{4,0,0,0,0,0,0,0,0,0,0}, {5,0,0,0,0,0,0,0,0,0,0}, {6,0,0,0,0,0,0,0,0,0,0},
{7,0,0,0,0,0,0,0,0,0,0}, {8,0,0,0,0,0,0,0,0,0,0}, {9,0,0,0,0,0,0,0,0,0,0},
{10,0,0,0,0,0,0,0,0,0,0}};

```

```

int Barco11 [5][2]= {{0,0},{0,0},{0,0},{0,0},{0,0}};

```

```

int Barco12 [4][2]= {{0,0},{0,0},{0,0},{0,0}};

```

```

int Barco13 [3][3]= {{0,0},{0,0},{0,0}};

```

```

int Barco14 [3][3]= {{0,0},{0,0},{0,0}};

```

```

int Barco15 [3][10]= {{0,0},{0,0}};

```

```

int Barco21 [5][3]= {{0,0},{0,0},{0,0},{0,0},{0,0}};

```

```

int Barco22 [4][3]= {{0,0},{0,0},{0,0},{0,0}};

```

```

int Barco23 [3][3]= {{0,0},{0,0},{0,0}};

```

```

int Barco24 [3][3]= {{0,0},{0,0},{0,0}};

```

```

int Barco25 [2][3]= {{0,0},{0,0}};

```

```

cor=1;

```

```

do

```

```

{

```

```

    switch (cor)

```

```

    {

```

```

        case 1:

```

```

system ("Color 0A");

```

```

system("cls");

```

```

printf ("\n \t Coordinadas jugador 1, porfavor no te equivoques ");

```



```
printf ("\n \t \t Inserta las coordenadas de tus barcos divididas por comas  
\n ");
```

```
printf ("\n Coordenadas del primer barco (5): ");
```

```
printf ("\n \t 1ra: _"); // pedir coordenadas del primer barco
```

```
scanf ("%i,%i", &a, &b);
```

```
matriz1 [a][b]='X';
```

```
Barco11 [1][1] = a;
```

```
Barco11 [1][2] = b;
```

```
printf ("\n \t 2da: _");
```

```
scanf ("%i,%i", &c, &d);
```

```
matriz1 [c][d]='X';
```

```
Barco11 [2][1] = c;
```

```
Barco11 [2][2] = d;
```

```
if (d==b-1 && a==c && b>=5)
```

```
{
```

```
printf("\n \t 3ra: _%i,%i \n", a, b-2);
```

```
matriz1 [a][b-2]='X';
```

```
Barco11 [3][1] = a;
```

```
Barco11 [3][2] = b-2;
```

```
printf("\n \t 4ta:_%i,%i \n", a, b-3);
```

```
matriz1 [a][b-3]='X';
```

```
Barco11 [4][1] = a;
```

```
Barco11 [4][2] = b-3;
```

```
printf("\n \t 5ta:_%i,%i \n", a, b-4);
```

```
matriz1 [a][b-4]='X';
```

```
Barco11 [5][1] = a;
```

```
Barco11 [5][2] = b-4;
```

```
cor=2;
```

```
getch ();
```

```
}
```

```
if(d==b+1 && a==c && b<=6)
```

```
{
```

```
    printf("\n \t 3ra:_%i,%i \n", a, b+2);
```

```
    matriz1 [a][b+2]='X';
```

```
    Barco11 [3][1] = a;
```

```
    Barco11 [3][2] = b+2;
```

```
printf("\n \t 4ta:_%i,%i \n", a, b+3);

matriz1 [a][b+3]='X';

Barco11 [4][1] = a;

Barco11 [4][2] = b+3;

printf("\n \t 5ta:_%i,%i \n", a, b+4);

matriz1 [a][b+4]='X';

Barco11 [5][1] = a;

Barco11 [5][2] = b+4;

cor=2;

getch ();

}
```

```
if (c==a-1 && b==d && a>=5)

{

printf("\n \t 3ra:_%i,%i \n", a-2, b);

matriz1 [a-2][b]='X';

Barco11 [3][1] = a-2;

Barco11 [3][2] = b;
```

```
printf("\n \t 4ta:_%i,%i \n", a-3, b);
```

```
matriz1 [a-3][b]='X';
```

```
Barco11 [4][1] = a-3;
```

```
Barco11 [4][2] = b;
```

```
printf("\n \t 5ta:_%i,%i \n", a-4, b);
```

```
matriz1 [a-4][b]='X';
```

```
Barco11 [5][1] = a-4;
```

```
Barco11 [5][2] = b;
```

```
cor=2;
```

```
getch ();
```

```
}
```

```
if (c==a+1 && b==d && a<=6)
```

```
{
```

```
    printf("\n \t 3ra:_%i,%i \n", a+2, b);
```

```
    matriz1 [a+2][b]='X';
```

```
    Barco11 [3][1] = a+2;
```

```
    Barco11 [3][2] = b;
```

```
    printf("\n \t 4ta:_%i,%i \n", a+3, b);
```

```
matriz1 [a+3][b]='X';
```

```
Barco11 [3][1] = a+3;
```

```
Barco11 [3][2] = b;
```

```
printf("\n \t 5ta: _%i,%i \n", a+4, b);
```

```
matriz1 [a+4][b]='X';
```

```
Barco11 [3][1] = a+4;
```

```
Barco11 [3][2] = b;
```

```
cor=2;
```

```
getch ();
```

```
}
```

```
if(cor!=2)
```

```
{
```

```
cor=6;
```

```
}
```

```
printf ("\n \t %i ", cor);
```

```
break;
```

```
case 2:
```

```

// fin
de pedir coordenadas de primer barco

    system("cls");

                                                                    printf
("
\t Coordenadas del segundo barco (4):");// pedir coordenadas del segundo
barco

                                                                    printf
("
\t 1ra:");

                                                                    scanf
("%i,%i", &e, &f);

Barco12 [1][1] = e;

Barco12 [1][2] = f;

matriz1 [e][f]='X';

printf ("
\t 2da:");

scanf ("%i,%i", &g, &h);

Barco12 [2][1] = g;

Barco12 [2][2] = h;

matriz1 [g][h]='X';

if (h==f-1 && e==g && f>=4)

{

printf("
\t 3ra:_%i,%i\n", e, f-2);

```

```
matriz1 [e][f-2]='X';
```

```
Barco12 [3][1] = e;
```

```
Barco12 [3][2] = f-2;
```

```
printf("\n \t 4ta:_%i,%i \n", e, f-3);
```

```
matriz1 [e][f-3]='X';
```

```
Barco12 [4][1] = e;
```

```
Barco12 [4][2] = f-3;
```

```
cor=3;
```

```
getch ();
```

```
}
```

```
if(h==f+1 && e==g && f<=7)
```

```
{
```

```
    printf("\n \t 3ra:_%i,%i \n", e, f+2);
```

```
    matriz1 [e][f+2]='X';
```

```
    Barco12 [3][1] = e;
```

```
    Barco12 [3][2] = f+2;
```

```
    printf("\n \t 4ta:_%i,%i \n", e, f+3);
```

```
    matriz1 [e][f+3]='X';
```

```

Barco12 [4][1] = e;

Barco12 [4][2] = f+3;

cor=3;

getch ();

}

if (g==e-1 && f==h && e>=4)

{

printf("\n \t 3ra:_%i,%i \n", e-2, f);

matriz1 [e-2][f]='X';

Barco12 [3][1] = e-2;

Barco12 [3][2] = f;

printf("\n \t 4ta:_%i,%i \n", e-3, f);

matriz1 [e-3][f]='X';

Barco12 [4][1] = e-3;

Barco12 [4][2] = f;

cor=3;

getch ();

```



```
}
```

```
if (g=e+1 && f==h && e<=7)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", e+2, f);
```

```
matriz1 [e+2][f]='X';
```

```
Barco12 [3][1] = e+2;
```

```
Barco12 [3][2] = f;
```

```
printf("\n \t 4ta:_%i,%i \n", e+3, f);
```

```
matriz1 [e+3][f]='X';
```

```
Barco12 [4][1] = e+3;
```

```
Barco12 [4][2] = f;
```

```
cor=3;
```

```
getch ();
```

```
}
```

```
if(cor!=3)
```

```
{
```

```
cor=6;
```

```
}
```

```
printf ("\n \t %i ", cor);
```

```
break;
```

```
// fin de las coordenadas segundo barco
```

```
case 3:
```

```
system("cls");
```

```
printf
```

```
(" \t Coordenadas del tercer barco (3): ");
```

```
printf
```

```
(" \n \t 1ra: _");
```

```
scanf
```

```
(" %i, %i", &i, &j);
```

```
matriz1 [i][j]='X';
```

```
Barco13 [1][1] = i;
```

```
Barco13 [1][2] = j;
```

```
printf
```

```
(" \n \t 2da: _");
```

```
scanf
```

```
(" %i, %i", &k, &l);
```

```
matriz1 [k][l]='X';
```

```
Barco13 [2][1] = k;
```

```
Barco13 [2][2] = l;
```

```
if
```

```
(l==j-1 && i==k && j>=3)
```

```
{
```

```
printf("\n \t 3ra: _%i,%i \n", i, j-2);
```

```
matriz1 [i][j-2]='X';
```

```
Barco13 [3][1] = i;
```

```
Barco13 [3][2] = j-2;
```

```
cor=4;
```

```
getch ();
```

```
}
```

```
if(l==j+1 && i==k && j<=8)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", i, j+2);
```

```
matriz1 [i][j+2]='X';
```

```
Barco13 [3][1] = i;
```

```
Barco13 [3][2] = j+2;
```

```
cor=4;
```

```
getch ();
```

```
}
```

```
if (k==i-1 && j==l && i>=3)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", i-2, j);
```

```
matriz1 [i-2][j]='X';
```

```
Barco13 [3][1] = i-2;
```

```
Barco13 [3][2] = j;
```

```
cor=4;
```

```
getch ();
```

```
}
```

```
if (k=i+1 && j==l && i<=8)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", i+2, j);
```

```
matriz1 [i+2][j]='X';
```

```
Barco13 [3][1] = i+2;
```

```
Barco13 [3][2] = j;
```

```
cor=4;
```

```
getch ();
```

```
}
```

```
if(cor!=4)
```

```
{
```

```
cor=6;
```

```
}
```

```
printf ("\n \t %i ", cor);
```

```
break;
```

```
// fin
```

```
de las coordenadas 4to barco
```

```
case 4:
```

```
system("cls");
```

```
printf("\t Coordenadas del cuarto barco (3):");
```

```
printf
```

```
(" \n \t 1ra: _");
```

```
scanf
```

```
(" %i, %i", &m, &n);
```

```
matriz1 [m][n]='X';
```

```
Barco14 [1][1] = m;
```

```
Barco14 [1][2] = n;
```

```
printf
```

```
(" \n \t 2da: _");
```

```
scanf
```

```
(" %i, %i", &o, &p);
```

```
matriz1 [o][p]='X';
```

```
Barco14 [2][1] = o;
```

```
Barco14 [2][2] = p;
```

```

                                                                    if
(p==n-1 && m==o && n>=3)

                                                                    {

printf("\n \t 3ra:_%i,%i \n", m, n-2);

matriz1 [m][n-2]='X';

Barco14 [3][1] = m;

Barco14 [3][2] = n-2;

cor=5;

getch ();

                                                                    }

if(p==n+1 && m==o && n<=8)

{

printf("\n \t 3ra:_%i,%i \n", m, n+2);

matriz1 [m][n+2]='X';

Barco14 [3][1] = m;

Barco14 [3][2] = n+2;

cor=5;

getch ();

}

if (o==m-1 && n==p && m>=3)

```

```
{  
  
printf("\n \t 3ra:_%i,%i \n", m-2, n);  
  
matriz1 [m-2][n]='X';  
  
Barco14 [3][1] = m-2;  
  
Barco14 [3][2] = n;  
  
cor=5;  
  
getch ();  
  
}
```

```
if (o=m+1 && n==p && m<=8)  
  
{  
  
    printf("\n \t 3ra:_%i,%i \n", m+2, n);  
  
    matriz1 [m+2][n]='X';  
  
    Barco14 [3][1] = m+2;  
  
    Barco14 [3][2] = n;  
  
    cor=5;  
  
    getch ();  
  
}
```

```

        if(cor!=5)

        {

                cor=6;

        }

        printf ("\n \t %i ", cor);


                                                break;

                                                // fin
de las coordenadas del cuarto barco

                                                case 5:

system("cls");

printf("\t Coordenadas del quinto barco (2): ");

                                                printf
("\n \t 1ra: _");

                                                scanf
("%i,%i", &q, &r);

                                                printf
("\n \t 2da: _");

                                                scanf
("%i,%i", &s, &t);

        matriz1 [q][r]='X';

        matriz1 [s][t]='X';

        Barco15 [1][1] = q;

        Barco15 [1][2] = r;

```



```
Barco15 [2][1] = s;
```

```
Barco15 [2][2] = t;
```

```
cor=8;
```

```
getch ();
```

```
break;
```

```
case 6:
```

```
system ("cls");
```

```
system ("Color 04");
```

```
printf ("\a \n \t Tu barco no  
puede estar en pedazos dejale eso al rival y por equivocarte vas de nuevo al  
principio \n \t presiona enter");
```

```
cor=1;
```

```
getch ();
```

```
break;
```

```
}
```

```
}
```

```
while (cor <7);
```

```
getch ();// fin de las coordenadas del jugador 1
```

```
system ("cls");
```

```
printf ("\n \t Tu tablero ser%c: ", aa);// Tablero con barcos
```

```
printf("\n \n ");
```

```
for(i=0;i<11;i++)
```

```
{
```

```
for(j=0;j<11;j++)
```

```
{
```

```
printf("\t %c ",matriz1[i][j]);
```

```
}
```

```

        printf ("\n");
    }
    getch ();

    cor1=1;
    do
    {
        switch (cor1)
        {
            case 1:

                system
                ("cls");

                system
                ("Color 0B"); // Color azul claro

                printf ("\n \t
Coordenadas jugador 2, porfavor no te equivoques o ambos volver%c\n al
comienzo", aa);

                printf ("\n \t \t
Inserta las coordenadas de tus barcos divididas por comas \n ");

                printf ("\n
Coordenadas del primer barco (5): ");

                printf ("\n \t
1ra:_");// pedir coordenadas del primer barco

                scanf

                matriz2

                Barco21

                Barco21

                printf ("\n \t
2da:_");

```

```

("%i,%i", &c, &d);

[c][d]='X';

[2][1] = c;

[2][2] = d;

a==c && b>=5)

scanf

matriz2

Barco21

Barco21

if (d==b-1 &&

{

printf("\n \t 3ra:_%i,%i \n", a, b-2);

matriz2 [a][b-2]='X';

Barco21 [3][1] = a;

Barco21 [3][2] = b-2;

printf("\n \t 4ta:_%i,%i \n", a, b-3);

matriz2 [a][b-3]='X';

Barco21 [4][1] = a;

Barco21 [4][2] = b-3;

printf("\n \t 5ta:_%i,%i \n", a, b-4);

matriz2 [a][b-4]='X';

Barco21 [5][1] = a;

Barco21 [5][2] = b-4;

cor1=2;

```

```

                                getch
();
                                }
                                if(d==b+1 &&
a==c && b<=6)
                                {

printf("\n \t 3ra:_%i,%i \n", a, b+2);

matriz2 [a][b+2]='X';

Barco21 [3][1] = a;

Barco21 [3][2] = b+2;

printf("\n \t 4ta:_%i,%i \n", a, b+3);

matriz2 [a][b+3]='X';

Barco21 [4][1] = a;

Barco21 [4][2] = b+3;

printf("\n \t 5ta:_%i,%i \n", a, b+4);

matriz2 [a][b+4]='X';

Barco21 [5][1] = a;

Barco21 [5][2] = b+4;

cor1=2;

getch ();

                                }

```

```
b==d && a>=5)
```

```
if (c==a-1 &&
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", a-2, b);
```

```
matriz2 [a-2][b]='X';
```

```
Barco21 [3][1] = a-2;
```

```
Barco21 [3][2] = b;
```

```
printf("\n \t 4ta:_%i,%i \n", a-3, b);
```

```
matriz2 [a-3][b]='X';
```

```
Barco21 [4][1] = a-3;
```

```
Barco21 [4][2] = b;
```

```
printf("\n \t 5ta:_%i,%i \n", a-4, b);
```

```
matriz2 [a-4][b]='X';
```

```
Barco21 [5][1] = a-4;
```

```
Barco21 [5][2] = b;
```

```
cor1=2;
```

```
getch
```

```
();
```

```
}
```

```
if (c=a+1 && b==d && a<=6)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", a+2, b);
```

```
matriz2 [a+2][b]='X';
```

```
Barco21 [3][1] = a+2;
```

```
Barco21 [3][2] = b;
```

```
printf("\n \t 4ta:_%i,%i \n", a+3, b);
```

```
matriz2 [a+3][b]='X';
```

```
Barco21 [3][1] = a+3;
```

```
Barco21 [3][2] = b;
```

```
printf("\n \t 5ta:_%i,%i \n", a+4, b);
```

```
matriz2 [a+4][b]='X';
```

```
Barco21 [3][1] = a+4;
```

```
Barco21 [3][2] = b;
```

```
cor1=2;
```

```
getch ();
```

```
}
```

```
if(cor1!=2)
```

```
{
```

```
    cor1=6;
```

```
}
```

```
printf ("\n \t %i ", cor);
```

```
break;
```

```
case 2:
```

```
// fin
```

```
de pedir coordenadas de primer barco
```

```
system ("cls");
```

```
printf  
("\n \t Coordenadas del segundo barco (4):");// pedir coordenadas del segundo  
barco
```

```
printf  
("\n \t 1ra: _");
```

```
scanf  
("%i,%i", &e, &f);
```

```
Barco22 [1][1] = e;
```

```
Barco22 [1][2] = f;
```

```
matriz2 [e][f]='X';
```

```
printf  
("\n \t 2da: _");
```

```
scanf  
("%i,%i", &g, &h);
```

```
Barco22 [2][1] = g;
```

```
Barco22 [2][2] = h;
```

```

matriz2 [g][h]='X';

(h==f-1 && e==g && f>=4)

printf("\n \t 3ra:_%i,%i \n", e, f-2);

matriz2 [e][f-2]='X';

Barco22 [3][1] = e;

Barco22 [3][2] = f-2;

printf("\n \t 4ta:_%i,%i \n", e, f-3);

matriz2 [e][f-3]='X';

Barco22 [4][1] = e;

Barco22 [4][2] = f-3;

cor1=3;

getch ();

if(h==f+1 && e==g && f<=7)

{

printf("\n \t 3ra:_%i,%i \n", e, f+2);

matriz2 [e][f+2]='X';

Barco22 [3][1] = e;

```



```
Barco22 [3][2] = f+2;
```

```
printf("\n \t 4ta:_%i,%i \n", e, f+3);
```

```
matriz2 [e][f+3]='X';
```

```
Barco22 [4][1] = e;
```

```
Barco22 [4][2] = f+3;
```

```
cor1=3;
```

```
getch ();
```

```
}
```

```
if (g==e-1 && f==h && e>=4)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", e-2, f);
```

```
matriz2 [e-2][f]='X';
```

```
Barco22 [3][1] = e-2;
```

```
Barco22 [3][2] = f;
```

```
printf("\n \t 4ta:_%i,%i \n", e-3, f);
```

```
matriz2 [e-3][f]='X';
```

```
Barco22 [4][1] = e-3;
```

```
Barco22 [4][2] = f;
```

```
cor1=3;
```

```
getch ();
```

```
}
```

```
if (g=e+1 && f==h && e<=7)
```

```
{
```

```
printf("\n \t 3ra:_%i,%i \n", e+2, f);
```

```
matriz2 [e+2][f]='X';
```

```
Barco22 [3][1] = e+2;
```

```
Barco22 [3][2] = f;
```

```
printf("\n \t 4ta:_%i,%i \n", e+3, f);
```

```
matriz2 [e+3][f]='X';
```

```
Barco22 [4][1] = e+3;
```

```
Barco22 [4][2] = f;
```

```
cor1=3;
```

```
getch ();
```

```
}
```

```
if(cor1!=3)
```

```
{
```

cor1=6;	
}	
printf ("\n \t %i ", cor);	
	break;
	case 3:// coordenadas del
barco 3	
	system
("cls");	
	printf (" \t
Coordenadas del tercer barco (3): ");	
	printf ("\n \t
1ra:_");	
	scanf
("%i,%i", &i, &j);	
	matriz2
[i][j]='X';	
	Barco23
[1][1] = i;	
	Barco23
[1][2] = j;	
	printf ("\n \t
2da:_");	
	scanf
("%i,%i", &k, &l);	
	matriz2
[k][l]='X';	
	Barco23
[2][1] = k;	
	Barco23
[2][2] = l;	

i==k && j>=3)

printf("\n \t 3ra:_%i,%i \n", i, j-2);

matriz2 [i][j-2]='X';

Barco23 [3][1] = i;

Barco23 [3][2] = j-2;

cor1=4;

();

i==k && j<=8)

printf("\n \t 3ra:_%i,%i \n", i, j+2);

matriz2 [i][j+2]='X';

Barco23 [3][1] = i;

Barco23 [3][2] = j+2;

cor1=4;

getch ();

(k==i-1 && j==l && i>=3)

if (l==j-1 &&

{

getch

}

if(l==j+1 &&

{

}

if

{

```
printf("\n \t 3ra:_%i,%i \n", i-2, j);
```

```
matriz2 [i-2][j]='X';
```

```
Barco23 [3][1] = i-2;
```

```
Barco23 [3][2] = j;
```

```
cor1=4;
```

```
getch ();
```

```
}
```

```
if (k=i+1 && j==l && i<=8)
```

```
{
```

```
    printf("\n \t 3ra:_%i,%i \n", i+2, j);
```

```
    matriz2 [i+2][j]='X';
```

```
    Barco23 [3][1] = i+2;
```

```
    Barco23 [3][2] = j;
```

```
    cor1=4;
```

```
    getch ();
```

```
}
```

```
if(cor1!=4)
```

```
{
```

cor1=6;	
}	
printf ("\n \t %i ", cor);	
	break;
	case 4:
	// fin de las
coordenadas 4to barco	
	system
("cls");	
	printf("\t
Coordenadasdel cuarto barco (3):");	
	printf ("\n \t
1ra: _");	
	scanf
("%i,%i", &m, &n);	
	matriz2
[m][n]='X';	
	Barco24
[1][1] = m;	
	Barco24
[1][2] = n;	
	printf ("\n \t
2da: _");	
	scanf
("%i,%i", &o, &p);	
	matriz2
[o][p]='X';	
	Barco24
[2][1] = o;	
	Barco24
[2][2] = p;	

```
m==o && n>=3)
```

```
printf("\n \t 3ra:_%i,%i \n", m, n-2);
```

```
matriz2 [m][n-2]='X';
```

```
Barco24 [3][1] = m;
```

```
Barco24 [3][2] = n-2;
```

```
cor1=5;
```

```
();
```

```
m==o && n<=8)
```

```
printf("\n \t 3ra:_%i,%i \n", m, n+2);
```

```
matriz2 [m][n+2]='X';
```

```
Barco24 [3][1] = m;
```

```
Barco24 [3][2] = n+2;
```

```
cor1=5;
```

```
getch ();
```

```
(o==m-1 && n==p && m>=3)
```

```
if (p==n-1 &&
```

```
{
```

```
getch
```

```
}
```

```
if(p==n+1 &&
```

```
{
```

```
}
```

```
if
```

{

```
printf("\n \t 3ra: _%i,%i \n", m-2, n);
```

```
matriz2 [m-2][n]='X';
```

```
Barco24 [3][1] = m-2;
```

```
Barco24 [3][2] = n;
```

```
cor1=5;
```

```
getch ();
```

}

```
if (o=m+1 && n==p && m<=8)
```

{

```
printf("\n \t 3ra: _%i,%i \n", m+2, n);
```

```
matriz2 [m+2][n]='X';
```

```
Barco24 [3][1] = m+2;
```

```
Barco24 [3][2] = n;
```

```
cor1=5;
```

```
getch ();
```

}

```
if(cor1!=5)
```



```

{

    cor1=6;

}

printf ("\n \t %i ", cor);

break;

case 5:

    // fin de las
    coordenadas del cuarto barco

    system

    ("cls");

    printf(" \t

Coordenadas del quinto barco (2): ");

    printf ("\n \t

1ra:_");

    scanf

    ("%i,%i", &q, &r);

    printf ("\n \t

2da:_");

    scanf

    ("%i,%i", &s, &t);

    matriz2 [q][r]='X';

    matriz2 [s][t]='X';

    Barco25 [1][1] = q;

    Barco25 [1][2] = r;

    Barco25 [2][1] = s;

```

```

cor1=8;

getch

());

break;

case 6:

    system ("cls");

    system ("Color 04");

    printf ("\a \n \t Tu barco no
puede estar en pedazos dejale eso al rival y por equivocarte vas de nuevo al
principio \n \t presiona enter");

    getch ();

    cor1=1;

    break;

}

}

while(cor1<7);

getch ();// fin de las coordenadas del jugador 2
system ("cls");
printf ("\n \t Tu tablero ser%c: ", aa);// Tablero con barcos
printf("\n \n ");
for(i=0;i<11;i++)
{
    for(j=0;j<11;j++)
    {
        printf("\t %c ",matriz2 [i][j]);

    }

    printf ("\n");
}

```

```

    }

    getch ();

    int win=-1, pos, Pos111=0, Pos112=0, Pos113=0,
    Pos114=0, Pos115=0, Pos121=0, Pos122=0, Pos123=0, Pos124=0, Pos131=0,
    Pos132=0, Pos133=0, Pos141=0, Pos142=0, Pos143=0, Pos151=0, Pos152=0,
    Pos211=0, Pos212=0, Pos213=0, Pos214=0, Pos215=0, Pos221=0, Pos222=0,
    Pos223=0, Pos224=0, Pos231=0, Pos232=0, Pos233=0, Pos241=0, Pos242=0,
    Pos243=0, Pos251=0, Pos252=0;

    int hundido11=0, hundido12=0, hundido13=0,
    hundido14=0, hundido15=0, hundido21=0, hundido22=0, hundido23=0,
    hundido24=0, hundido25=0;

    do // inicio de algoritmo de turnos
    {

        system ("cls");

        system ("color 0A"); // color verde claro

        if (Pos211==1 && Pos212==1 &&
        Pos213==1 && Pos214 ==1 && Pos215==1 && Pos221==1 && Pos222==1 &&
        Pos223==1 && Pos224 ==1 && Pos231==1 && Pos232==1 && Pos233==1 &&
        Pos241==1 && Pos242==1 && Pos243==1 && Pos251==1 && Pos252==1 )

        {

            win=2;

            Bool=0;

        }

        else

        {

            printf("\n \t \n \t \n \t Turno
del jugador 1 \n Presione enter para continuar \n");

            getch ();

            for(i=0;i<11;i++)

            {

                for(j=0;j<11;j++)

                {

                    printf("\t %c
",matriz3 [i][j]); //tablero con marcas

```

",matriz1 [i][j]);//tablero propio

concordancia de cordenadas con barcos

0://inicio

printf ("\n \t Introduzca las coordenadas en donde desea disparar_");

scanf ("%i,%i", &u, &v);

matriz3 [u][v]='X';

barco=1;

```
    }
    printf ("\n");
}
printf ("\n \n ");
for(i=0;i<11;i++)
{
    for(j=0;j<11;j++)
    {
        printf("\t %c",matriz1[i][j]);
    }
    printf ("\n");
}
barco=0;
Bool=1;
```

do // Revisión de

```
{

    switch (barco)
    {
        case
```

```
break;  
case 1:
```

```
//Barco 1
```

```
    if(hundido21==1)  
  
    {  
  
        printf ("\n \t Barco1---Hundido---");  
  
        barco=2;  
  
    }  
  
    else  
  
    {  
  
        for (i=1;i<=5;)  
  
        {  
  
            j=1;  
  
            if (u==Barco21 [i][j])  
  
            {  
  
                j++;  
  
                if (v==Barco21 [i][j])  
  
                {  
  
                    pos=i;
```

```

switch (pos)
{
    case 1:
        Pos211 =1;

        printf ("\n \t Has
acertado");

        matriz3 [u][v]=lv;

        barco=6;

        i=7;

        break;

    case 2:
        Pos212 =1;

        printf ("\n \t Has
acertado");

        matriz3 [u][v]=lv;

        barco=6;

        i=7;

        break;

```

```

acertado");

case 3:

    Pos213 =1;

    printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 4:

    Pos214 =1;

    printf ("\n \t Has

acertado");

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 5:

    Pos215 =1;

```

acertado");

printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

}

}

else

{

i++;

}

}

else

{

i++;

}

}


```
        if (i==7)

        {

                                barco =6;

        }

        else

        {

                                barco=2;

        }

}
```

break;

case 2:

//Barco 2

```
        if(hundido22==1)

        {

                printf ("\n Barco2 ---Hundido---");

                barco=3;

        }

        else
```

```

{

    for (i=1;i<=4;)

    {

        j=1;

        if (u==Barco22 [i][j])

        {

            j++;

            if (v==Barco22 [i][j])

            {

                pos=i;

                switch (pos)

                {

                    case 1:

                        Pos221 =1;

                        printf ("\n \t Has

acertado");

                        matriz3 [u][v]=lv;

                        barco=6;

                        i=7;

```

break;

case 2:

Pos222 =1;

acertado");

printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 3:

Pos223 =1;

acertado");

printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

```

                                case 4:

                                Pos224 =1;

                                printf ("\n \t Has
acertado");

                                matriz3 [u][v]=lv;

                                barco=6;

                                i=7;

                                break;

                                }

                                }

                                else

                                {

                                i++;

                                }

                                }

                                else

                                {

                                i++;

```

```
}
```

```
}
```

```
if (i==7)
```

```
{
```

```
    barco =6;
```

```
}
```

```
else
```

```
{
```

```
    barco=3;
```

```
}
```

```
}
```

```
break;
```

```
case 3:
```

```
//Barco 3
```

```
if (hundido23==1)
```

```
{
```

```
printf ("\n Barco3 ---Hundido---");
```



```
matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 2:

Pos232 =1;

printf ("\n \t Has
acertado");

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 3:

Pos233 =1;

printf ("\n \t Has
acertado");

matriz3 [u][v]=lv;
```

```
barco=6;
```

```
i=7;
```

```
break;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```
}
```



```
        if (i==7)

        {

                                barco =6;

        }

        else

        {

                                barco=4;

        }

}
```

break;

case 4: //

Barco 4

```
        if (hundido24==1)

        {

                printf ("\n Barco4 ---Hundido---");

                barco =5;

        }

        else
```

```

{

    for (i=1;i<=3;)

    {

        j=1;

        if (u==Barco24 [i][j])

        {

            j++;

            if (v==Barco24 [i][j])

            {

                pos=i;

                switch (pos)

                {

                    case 1:

                        Pos241 =1;

                        printf ("\n \t Has

acertado");

                        matriz3 [u][v]=lv;

                        barco=6;

                        i=7;

```

break;

case 2:

Pos242 =1;

acertado");

printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

case 3:

Pos243 =1;

acertado");

printf ("\n \t Has

matriz3 [u][v]=lv;

barco=6;

i=7;

break;

```
}
```

```
}
```

```
else
```

```
{
```

```
    i++;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    i++;
```

```
}
```

```
}
```

```
if (i==7)
```

```
{
```

```
    barco =6;
```

```
}
```

```
else
```

```
    {  
  
        barco=5;  
  
    }  
  
}
```

break;

case 5:

```
if (hundido25==1)  
  
    {  
  
        printf ("\n Barco5 ---Hundido---");  
  
        barco=7;  
  
    }  
  
else  
  
    {  
  
        for (i=1;i<=2;)   
  
            {  
  
                j=1;  
  
                if (u==Barco25 [i][j])  
  
                    {
```

```

j++;

if (v==Barco25 [i][j])

{

    pos=i;

    switch (pos)

    {

        case 1:

            Pos251 =1;

            printf ("\n \t Has

acertado");

            matriz3 [u][v]=lv;

            barco=6;

            i=7;

            break;

        case 2:

            Pos252 =1;

            printf ("\n \t Has

acertado");

```

```
matriz3 [u][v]=lv;
```

```
barco=6;
```

```
i=7;
```

```
break;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```

    }

    if (i==7)

    {

        barco =6;

    }

    else

    {

        barco=7;

    }

}

```

```

break;
case 6: // En

```

caso de acertar

```

    if (Pos211==1 && Pos212==1 && Pos213==1 && Pos214 ==1 &&
Pos215==1 )

    {

        hundido21=1;

    }

```



```
if (Pos221==1 && Pos222==1 && Pos223==1 && Pos224 ==1)
```

```
{
```

```
    hundido22=1;
```

```
}
```

```
if (Pos231==1 && Pos232==1 && Pos233==1)
```

```
{
```

```
    hundido23 =1;
```

```
}
```

```
if (Pos241==1 && Pos242==1 && Pos243==1)
```

```
{
```

```
    hundido24=1;
```

```
}
```

```
if (Pos251==1 && Pos252==1)
```

```
{
```

```
    hundido25=1;
```

```
}
```

```
        if (Pos211==1 && Pos212==1 && Pos213==1 && Pos214 ==1 &&
Pos215==1 && Pos221==1 && Pos222==1 && Pos223==1 && Pos224 ==1 &&
Pos231==1 && Pos232==1 && Pos233==1 && Pos241==1 && Pos242==1 &&
Pos243==1 && Pos251==1 && Pos252==1 )
```

```
{
```

```
    win=1;
```

```
    Bool=0;
```

```
}
```

```
else
```

```
{
```

```
    printf ("\n \t Te toca de nuevo");
```

```
    getch();
```

```
    system ("cls");
```

```
    printf("\n \t \n \t \n \t Turno del jugador 1 \n Presione enter para
continuar \n");
```

```
    for(i=0;i<11;i++)
```

```
{
```

```
        for(j=0;j<11;j++)
```

```
{
```

```
            printf("\t %c ",matriz3 [i][j]);
```

```
        }

        printf ("\n");

    }

    printf ("\n \n ");

    for(i=0;i<11;i++)

    {

        for(j=0;j<11;j++)

        {

            printf("\t %c ",matriz1 [i][j]);

        }

        printf ("\n");

    }

    Bool =1;

    barco=0;

}

break;

case 7:
```

```

printf ("\n \t Has fallado, mejor suerte a la proxima ");

barco=8;

break;

case 8:

Bool=0;

break;
}

}

while (Bool>0);

}

printf ("\n \t Presione enter para
continuar ");

getch ();
system ("cls");
system("Color 0B");
printf("\n \t \n \t \n \t Turno del jugador 2
\n Presione enter para continuar \n");

getch ();

for(i=0;i<11;i++)

{

for(j=0;j<11;j++)

{

```

```

        printf("\t %c ",matriz4 [i][j]);

    }

    printf ("\n");

}

printf ("\n \n ");

for(i=0;i<11;i++)

{

    for(j=0;j<11;j++)

    {

        printf("\t %c ",matriz2 [i][j]);

    }

    printf ("\n");

}

```

```

        if (Pos211==1 && Pos212==1 &&
Pos213==1 && Pos214 ==1 && Pos215==1 && Pos221==1 && Pos222==1 &&
Pos223==1 && Pos224 ==1 && Pos231==1 && Pos232==1 && Pos233==1 &&
Pos241==1 && Pos242==1 && Pos243==1 && Pos251==1 && Pos252==1 )

```

```

{

```

```

    win=1;

```

```

    Bool=0;

```

```

    }
    else
    {

        barco2=0;
        Bool =1;

        do
        {
            switch (barco2)
            {
                case
                    system

0://inicio
("cls");

                system("Color 0B");
                printf("\n \t \n \t \n \t Turno del jugador 2
\n Presione enter para continuar \n");

                getch ();

                for(i=0;i<11;i++)

                {

                    for(j=0;j<11;j++)

                    {

                        printf("\t %c ",matriz4 [i][j]);

                    }

                }

                printf ("\n");

```

```

}

printf ("\n \n ");

for(i=0;i<11;i++)

{

    for(j=0;j<11;j++)

    {

        printf("\t %c ",matriz2 [i][j]);

    }

    printf ("\n");

}

printf ("\n \t Introduzca las coordenadas en donde desea disparar_");

scanf ("%i,%i", &x, &y);

matriz4 [x][y]='X';

barco2=1;

break;

case 1:

//Barco 1

```

```
        if (Pos111==1 && Pos112==1 && Pos113==1 && Pos114 ==1 &&
Pos115==1)
```

```
{
```

```
    printf ("\n Barco1---Hundido---");
```

```
    barco2=2;
```

```
}
```

```
else
```

```
{
```

```
    for (i=1;i<=5;)
```

```
{
```

```
    j=1;
```

```
    if (x==Barco11 [i][j])
```

```
{
```

```
        j++;
```

```
        if (y==Barco11 [i][j])
```

```
{
```

```
            pos=i;
```

```
            switch (pos)
```



```

{

    case 1:

        Pos111 =1;

        printf ("\n \t Has
acertado");

        matriz4 [x][y]=lv;

        barco2=6;

        i=7;

        break;

    case 2:

        Pos112 =1;

        printf ("\n \t Has
acertado");

        matriz4 [x][y]=lv;

        barco2=6;

        i=7;

        break;

```

```
acertado");

case 3:

    Pos113 =1;

    printf ("\n \t Has

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

case 4:

    Pos114 =1;

    printf ("\n \t Has

acertado");

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

case 5:

    Pos115 =1;
```

acertado");

printf ("\n \t Has

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

}

}

else

{

i++;

}

}

else

{

i++;

}

}

```

        if (i==7)

        {

                barco2 =6;

        }

        else

        {

                barco2=2;

        }

}

break;

case 2:

//Barco 2

if(Pos121==1 && Pos122==1 && Pos123==1 && Pos124 ==1 )

{

        printf ("\nBarco2 ---Hundido---");

        barco2=3;

}

else

{

```

```

for (i=1;i<=4;)

{

    j=1;

    if (x==Barco12 [i][j])

    {

        j++;

        if (y==Barco12 [i][j])

        {

            pos=i;

            switch (pos)

            {

                case 1:

                    Pos121 =1;

                    printf ("\n \t Has

acertado");

                    matriz4 [x][y]=lv;

                    barco2=6;

                    i=7;

                    break;

```

```
acertado");

case 2:

    Pos122 =1;

    printf ("\n \t Has

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

case 3:

    Pos123 =1;

    printf ("\n \t Has

acertado");

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

case 4:
```

acertado");

Pos124 =1;

printf ("\n \t Has

matriz4 [x][y]=lv;

barco2=6;

i=7;

break;

}

}

else

{

i++;

}

}

else

{

i++;

}

```
    }  
  
    if (i==7)  
  
    {  
  
        barco2 =6;  
  
    }  
  
    else  
  
    {  
  
        barco2=3;  
  
    }  
  
}
```

break;

case 3:

//Barco 3

```
if(Pos131==1 && Pos132==1 && Pos133==1)  
  
{  
  
printf ("\nBarco3 ---Hundido---");  
  
barco2=4;  
  
}
```



```

else

{

    for (i=1;i<=3;)

    {

        j=1;

        if (x==Barco13 [i][j])

        {

            j++;

            if (y==Barco13 [i][j])

            {

                pos=i;

                switch (pos)

                {

                    case 1:

                        Pos131 =1;

                        printf ("\n \t

Has acertado");

                        matriz4

[x][y]=lv;

```

	barco2=6;
	i=7;
	break;
	case 2:
	Pos132 =1;
Has acertado");	printf ("\n \t
	matriz4
[x][y]=lv;	barco2=6;
	i=7;
	break;
	case 3:
	Pos133 =1;
Has acertado");	printf ("\n \t
	matriz4
[x][y]=lv;	barco2=6;

i=7;

break;

}

}

else

{

i++;

}

}

else

{

i++;

}

}

if (i==7)

```

        {

            barco2 =6;

        }

        else

        {

            barco2=4;

        }

    }

```

break;

case 4: //

Barco 4

```

    if (Pos141==1 && Pos142==1 && Pos143==1)

    {

        printf ("\nBarco4---Hundido---");

        barco2=5;

    }

    else

    {

        for (i=1;i<=3;)

```

```

{

    j=1;

    if (x==Barco14 [i][j])

    {

        j++;

        if (y==Barco14 [i][j])

        {

            pos=i;

            switch (pos)

            {

                case 1:

                    Pos141 =1;

                    printf ("\n \t Has

acertado");

                    matriz4 [x][y]=lv;

                    barco2=6;

                    i=7;

                    break;

```

```

                                case 2:

                                    Pos142 =1;

                                    printf ("\n \t Has
acertado");

                                    matriz4 [x][y]=lv;

                                    barco2=6;

                                    i=7;

                                break;

                                case 3:

                                    Pos143 =1;

                                    printf ("\n \t Has
acertado");

                                    matriz4 [x][y]=lv;

                                    barco2=6;

                                    i=7;

                                break;

                                }

```

```
}
```

```
else
```

```
{
```

```
    i++;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    i++;
```

```
}
```

```
}
```

```
if (i==7)
```

```
{
```

```
    barco2 =6;
```

```
}
```

```
else
```

```
{
```

```
barco2=5;
```

```
}
```

```
}
```

```
break;
```

```
case 5:
```

```
if (Pos151==1 && Pos152==1)
```

```
{
```

```
printf ("\nBarco5---Hundido---");
```

```
barco2=7;
```

```
}
```

```
else
```

```
{
```

```
for (i=1;i<=2;)
```

```
{
```

```
j=1;
```

```
if (x==Barco15 [i][j])
```

```
{
```

```
j++;
```



```

if (y==Barco15 [i][j])

{

    pos=i;

    switch (pos)

    {

        case 1:

            Pos151 =1;

            printf ("\n \t Has

acertado");

            matriz4 [x][y]=lv;

            barco2=6;

            i=7;

            break;

        case 2:

            Pos152 =1;

            printf ("\n \t Has

acertado");

            matriz4 [x][y]=lv;

```

```
barco2=6;
```

```
i=7;
```

```
break;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```
}
```

```
else
```

```
{
```

```
i++;
```

```
}
```

```
}
```

```

        if (i==7)

        {

                                barco2 =6;

        }

        else

        {

                                barco2=7;

        }

}

```

```

break;
case 6: // En

```

caso de acertar

```

        if (Pos111==1 && Pos112==1 && Pos113==1 && Pos114 ==1 &&
Pos115==1 )

```

```

        {

                                hundido11=1;

        }

```

```

        if (Pos121==1 && Pos122==1 && Pos123==1 && Pos124 ==1)

```

```

        {

                                hundido12=1;

```

```
}
```

```
if (Pos131==1 && Pos132==1 && Pos133==1)
```

```
{
```

```
    hundido13 =1;
```

```
}
```

```
if (Pos141==1 && Pos142==1 && Pos143==1)
```

```
{
```

```
    hundido14=1;
```

```
}
```

```
if (Pos151==1 && Pos152==1)
```

```
{
```

```
    hundido15=1;
```

```
}
```

```
    if (Pos111==1 && Pos112==1 && Pos113==1 && Pos114 ==1 &&  
Pos115==1 && Pos121==1 && Pos122==1 && Pos123==1 && Pos124 ==1 &&  
Pos131==1 && Pos132==1 && Pos133==1  && Pos141==1 && Pos142==1 &&  
Pos143==1  && Pos151==1 && Pos152==1 )
```

```
{
```

```
    win=2;
```

```
    Bool=0;
```

```
}
```

```
else
```

```
{
```

```
printf ("\n \t Te toca de nuevo");
```

```
getch ();
```

```
system ("cls");
```

```
Bool =1;
```

```
barco2=0;
```

```
}
```

```
break;
```

```
case 7:// en
```

```
caso de fallar
```

```
printf ("\n \t Has fallado o ya has hundido dicho barco, mejor suerte a la proxima ");
```

```
getch ();
```

```
barco2 =8;
```

```
break;
```

```
case 8:
```

```

                                Bool =
0;
                                printf
                                ("%i", Bool);
                                break;
                                }
                                }
                                while (Bool>0);
                                }

                                }
                                while (win<1);
                                system("color 0F");
                                system("cls");
                                printf ("\n \n \n \n \n \t \t \t Felicitades jugador %i has
ganado", win);
                                getch ();
                                case 3:// mensaje de salida
                                system ("cls");
                                printf ("\n \t Gracias por jugar vuelva pronto :3");
                                printf ("\n \t .....creditos.....");
                                printf ("\n \n \t Dise%co.....Fr%cas", an, ai);
                                printf ("\n \n \t Programaci%cn y c%cdigo.....Fr%cas",
ao, ao, ai);
                                printf ("\n \n \t Todo lo dem%cs.....yo mismo otra vez", aa);
                                getch();
                                system ("cls");
                                break;
                                }
                                }
                                while (OP!=3);

```

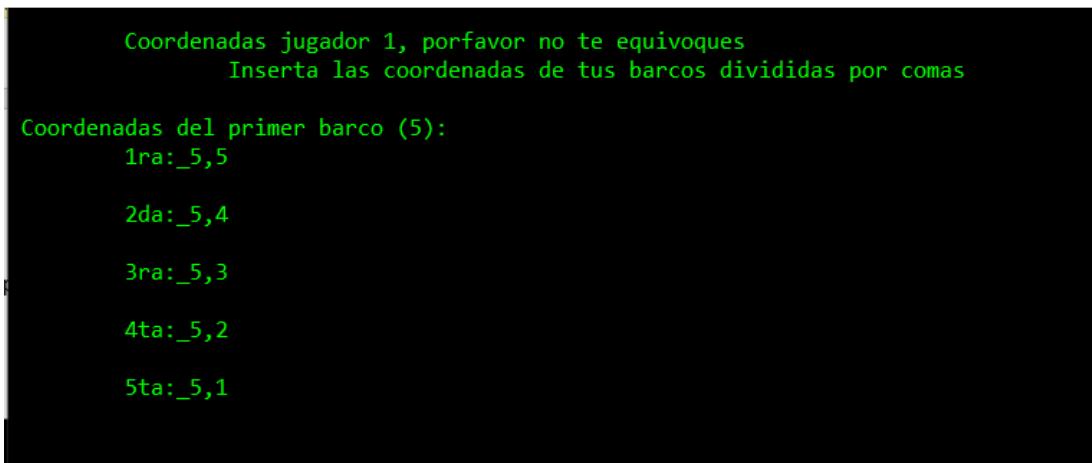
```

    return 0;
}

```

Resultados:

Capturas de pantalla:



Tu barco no puede estar en pedazos dejale eso al rival y por equivocarte vas de nuevo al principio
presiona enter_

Tu tablero será:

-	A	B	C	D	E	F	G	H	I	J
A	X	X	O	O	O	O	O	O	O	O
B	O	O	O	O	O	O	O	O	O	O
C	O	O	O	O	O	O	O	O	O	O
D	O	O	O	O	O	O	O	O	O	O
E	X	X	X	X	X	O	O	O	O	O
F	O	O	O	O	O	O	O	O	O	O
G	O	O	O	O	X	X	X	O	O	O
H	O	O	O	O	O	X	X	X	O	O
I	O	O	O	O	O	X	X	X	X	O
J	O	O	O	O	O	O	O	O	O	O

Coordenadas jugador 2, porfavor no te equivoques o ambos volverán al comienzo
Inserta las coordenadas de tus barcos divididas por comas

Coordenadas del primer barco (5):

1ra:_

Tu tablero será:

-	A	B	C	D	E	F	G	H	I	J
A	X	X	O	O	O	O	O	O	O	O
B	O	O	O	O	O	O	O	O	O	O
C	O	O	O	O	O	O	O	O	O	O
D	O	O	O	O	O	O	O	O	O	O
E	X	X	X	X	X	O	O	O	O	O
F	O	O	O	O	O	O	O	O	O	O
G	O	O	O	O	X	X	X	O	O	O
H	O	O	O	O	O	X	X	X	O	O
I	O	O	O	O	O	X	X	X	X	O
J	O	O	O	O	O	O	O	O	O	O

Turno del jugador 1
Presione enter para continuar

-	A	B	C	D	E	F	G	H	I	J
A	O	O	O	O	O	O	O	O	O	O
B	O	O	O	O	O	O	O	O	O	O
C	O	O	O	O	O	O	O	O	O	O
D	O	O	O	O	O	O	O	O	O	O
E	O	O	O	O	O	O	O	O	O	O
F	O	O	O	O	O	O	O	O	O	O
G	O	O	O	O	O	O	O	O	O	O
H	O	O	O	O	O	O	O	O	O	O
I	O	O	O	O	O	O	O	O	O	O
J	O	O	O	O	O	O	O	O	O	O

-	A	B	C	D	E	F	G	H	I	J
A	X	X	O	O	O	O	O	O	O	O
B	O	O	O	O	O	O	O	O	O	O
C	O	O	O	O	O	O	O	O	O	O
D	O	O	O	O	O	O	O	O	O	O
E	X	X	X	X	X	O	O	O	O	O
F	O	O	O	O	O	O	O	O	O	O
G	O	O	O	O	X	X	X	O	O	O
H	O	O	O	O	O	X	X	X	O	O
I	O	O	O	O	O	X	X	X	X	O
J	O	O	O	O	O	O	O	O	O	O

Introduzca las coordenadas en donde desea disparar_

Presione enter para continuar

-	A	B	C	D	E	F	G	H	I	J
A	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

-	A	B	C	D	E	F	G	H	I	J
A	X	X	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	X	X	X	X	X	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	X	X	X	0	0	0
H	0	0	0	0	0	X	X	X	0	0
I	0	0	0	0	0	X	X	X	X	0
J	0	0	0	0	0	0	0	0	0	0

Introduzca las coordenadas en donde desea disparar_5,5

Has acertado

Te toca de nuevo

-	A	B	C	D	E	F	G	H	I	J
A	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E						0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

-	A	B	C	D	E	F	G	H	I	J
A	X	X	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	X	X	X	X	X	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	X	X	X	0	0	0
H	0	0	0	0	0	X	X	X	0	0
I	0	0	0	0	0	X	X	X	X	0
J	0	0	0	0	0	0	0	0	0	0

Introduzca las coordenadas en donde desea disparar_5,5

Barco1---Hundido---

Has fallado, mejor suerte a la proxima

Presione enter para continuar

Turno del jugador 2
Presione enter para continuar

-	A	B	C	D	E	F	G	H	I	J
A	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

-	A	B	C	D	E	F	G	H	I	J
A	X	X	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	X	X	X	X	X	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	X	X	X	0	0	0
H	0	0	0	0	0	X	X	X	0	0
I	0	0	0	0	0	X	X	X	X	0
J	0	0	0	0	0	0	0	0	0	0

Introduzca las coordenadas en donde desea disparar

Presione enter para continuar

-	A	B	C	D	E	F	G	H	I	J
A	0	0	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	0	0	0	0	0	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	0	0	0	0	0	0
H	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	0	0	0	0
J	0	0	0	0	0	0	0	0	0	0

-	A	B	C	D	E	F	G	H	I	J
A	X	X	0	0	0	0	0	0	0	0
B	0	0	0	0	0	0	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0
D	0	0	0	0	0	0	0	0	0	0
E	X	X	X	X	X	0	0	0	0	0
F	0	0	0	0	0	0	0	0	0	0
G	0	0	0	0	X	X	X	0	0	0
H	0	0	0	0	0	X	X	X	0	0
I	0	0	0	0	0	X	X	X	X	0
J	0	0	0	0	0	0	0	0	0	0

Introduzca las coordenadas en donde desea disparar_5,5

Has acertado

Te toca de nuevo_

```

C      0      0      0      0      0      0      0      0      0      0      0
D      0      0      0      0      0      0      0      0      0      0      0
E      0      0      0      0      0      0      0      0      0      0      0
F      0      0      0      0      0      0      0      0      0      0      0
G      0      0      0      0      0      0      0      0      0      0      0
H      0      0      0      0      0      0      0      0      0      0      0
I      0      0      0      0      0      0      0      0      0      0      0
J      0      0      0      0      0      0      0      0      0      0      0

-      A      B      C      D      E      F      G      H      I      J
A      X      X      0      0      0      0      0      0      0      0
B      0      0      0      0      0      0      0      0      0      0      0
C      0      0      0      0      0      0      0      0      0      0      0
D      0      0      0      0      0      0      0      0      0      0      0
E      X      X      X      X      X      0      0      0      0      0      0
F      0      0      0      0      0      0      0      0      0      0      0
G      0      0      0      0      X      X      X      0      0      0      0
H      0      0      0      0      0      X      X      X      0      0      0
I      0      0      0      0      0      X      X      X      X      0      0
J      0      0      0      0      0      0      0      0      0      0      0

Introduzca las coordenadas en donde desea disparar_1,1
Barco1--Hundido--
Barco2 ---Hundido---
Barco3 ---Hundido---
Barco4---Hundido---
      Has acertado
      Te toca de nuevo_

```

Felicidades jugador 2 has ganado

Gracias por jugar vuelva pronto :3
.....creditos.....

Diseño.....Frías

Programación y código.....Frías

Todo lo demás.....yo mismo otra vez

Las instrucciones son simples el tablero de arriba representa un mapa donde está tu enemigo el de abajo es tu mapa donde estarás tú.

Escribe las coordenadas donde te ubicas y reza porque no te encuentren despues preparete para de tratar evitar que vuelva a intentarlo de nuevo escribiendo las coordenadas donde podria estar

0	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

0	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Tabla comparativa de recursos:

Antes	Despues
Computadora	Computadora
curso	Curso
Un buen compilador	Un compilador
Licencia	Licencia
Saber exportar	
Un repositorio	Un repositorio
18 meses	22 meses

Tabla comparativa de costos:

Pasos	Primer costo	Segundo costo
Planeación	5000	2400
Codificación	30000	20000
depuración	3000	90000

Comparación de diagramas:



Canal:

Camille frías

https://www.youtube.com/channel/UCQngWvCoH5-leJq-o_zlfHg

Video:

<https://youtu.be/zF5lrTHvzYs>

Repositorio:

<https://github.com/camfriher/Archivos-de-fundamentos-de-programaci-n->

<https://github.com/camfriher/Archivos-de-fundamentos-de-programaci-n-/tree/main/Proyecto%20final>

Conclusiones:

Mi proyecto cumplió con el propósito de demostrar los aprendizajes debido a que este fue exitoso, su relevancia es sobre todo en un ámbito escolar didáctico y de enseñanza más allá de tener una aplicación en la industria.

Fue compilado por

GCC el cual es un compilador de GNU el cual promueve la creación de programas y software libre, por lo que en honor de sus principios este software es libre igual.

En cuanto a la IEE no tiene una gran relevancia debido a su poca aplicación en la industria y poca innovación dentro de su código.

En un ámbito personal ayudo a comprobar mis conocimientos adquiridos y demostrar a los demás como a mí mismo que soy capaz de aplicarlos a un programa funcional que se decida realizar, así mismo he de aceptar que algo tan visualmente simple cómo esto requirió de bastante esfuerzo y que depurar un programa y entender una falla es un trabajo aún mayor que hacer el programa en sí, sin duda espero que me paguen bien en un futuro porque sí cuesta hacer este tipo de cosas.

Referencias:

Ingeniero Lara. (2013, 24 diciembre). Como cambiar los colores de texto y de fondo en c++ [Video]. YouTube.

<https://www.youtube.com/watch?v=NgbHKt5pssE&t=91s>

Metodología de la programación. Osvaldo Cairó, tercera edición, México D.F., Alfaomega 2005.

El lenguaje de programación C.Brian W. Kernighan, Dennis M. Ritchie,segunda edición, USA, Pearson Educación 1991.

Apuntes y prácticas de fundamentos de programación.

Vinicio Román Edmundo Frías Rojas (Padre), Teoría de estructuras de control anidadadas en, repetición y depuración ,27/01/2021, Personal

DevDocs — C documentation. (s. f.). <https://devdocs.io/c/>. Recuperado 29 de enero de 2021, de <https://devdocs.io/c/>

Glosario:

Acronimos:

Posnnn: significará posición donde la primera n se referirá a cual jugador pertenece el objeto, la segunda n será el número del barco y la tercera será el número de posición

Cor: coordenada donde cor serán las del jugador 1 y cor1 las del jugador 2

Int: una variable entera

Char: variable carácter

Matriz: arreglo de un tipo de variable en especifico

Anexos:

Manual:

Primeramente, el programa iniciará en símbolos de sistema con la instrucción VJ.exe una vez que ya ha sido compilado de forma correcta.

Esto desplegará el menú de opciones y la portada

Pudiendo elegir entre 1, 2 y 3, el uno siendo el inicio del juego, el 2 un pequeño instructivo de cómo funciona el juego y 3 que despliega los créditos y le da salida al programa.

Al iniciar el juego comenzará el jugador 1 a poner las coordenadas de sus barcos, iniciando en el barco 1 que ocupará 5 espacios, ingresará la componente en x de la coordenada una coma, se pondrá la componente en y y se dará enter lo cual hará que el programa pida la segunda, solo se necesitarán 2 coordenadas para que un barco sea completado gracias al increíble algoritmo del programa, si es que se da una coordenada imposible para un barco marcará un mensaje de error y comenzará a poner las coordenadas desde el barco 1.

Al terminar de poner las coordenadas se imprimirá un tablero donde marcará las posiciones donde están sus barcos.

Seguido de un mensaje para que el jugador 2 ponga sus coordenadas, posteriormente se iniciará el juego en el turno del jugador 1 el cual ingresará las coordenadas numéricas y de ser correcto marcará un mensaje de correcto y volverá a solicitar las nuevas coordenadas esto se repetirá hasta que falle o gane de fallar le irá al jugador 2 el cual seguirá el mismo procedimiento hasta que haya un ganador.

Al momento de haber un ganador se mandará un mensaje de salida y se regresará al menú donde podrá elegir de nuevo otra de las 3 opciones.

Archivo.c del programa (VJ.c):

<https://github.com/camfriher/Archivos-de-fundamentos-de-programaci-n-/blob/main/Proyecto%20final/VJ.c>

Video:

<https://youtu.be/zF5lrTHvzYs>