



UNIVERSITAT
ROVIRA I VIRGILI

Departament d'Enginyeria Informàtica i Matemàtiques

FP: Projecte Final Segona Fase

ALUMNE: Aleix Mariné
PROFESSOR: Carlos García
ASSIGNATURA: Fonaments de programació
ENSENYAMENT: Doble Grau de Biotec. Eng. Info.
GRUP: Grup G3
DATA: 10/1/2016

Índex

1	INTRODUCCIÓ I ESPECIFICACIONS GENERALS	4
2	ESPECIFICACIONS	5
2.1	NIVELL 1	5
2.2	NIVELL 2	5
2.3	NIVELL 3	6
3	ANÀLISI	7
3.1	NIVELL 1	7
3.1.1	Menu	7
3.1.2	Inicia taulell	7
3.1.3	Funció jugar + funció dispara	7
3.1.4	Emmagatzema/recupera rècords	7
3.1.5	Control del joc.....	8
3.2	NIVELL 2	8
3.2.1	Emmagatzema joc	8
3.2.2	Recupera joc.....	8
3.2.3	Nova jugada	8
3.3	NIVELL 3	8
3.3.1	Inicia el meu taulell.....	8
3.3.2	Per a efectuar una nova jugada	8
4	DISSENY.....	9
4.1	MENU.....	9
4.2	PREPARACIÓ DEL JOC.....	10
4.2.1	TIPUS_JOC	10
4.2.2	RANDOMIZE.....	12
4.2.3	INICIA_TAULER	13
4.2.4	INICIA_EL_MEU_TAULER.....	16
4.2.5	MONEDA	21
4.3	VISUALITZACIÓ DE MATRIUS	23
4.3.1	VISUALITZA_SIMPLE.....	23
4.3.2	VISUALITZA	24

4.4	TRACTAMENT DE RECORDS	26
4.4.1	INICIALITZA_TAUULA	26
4.4.2	IMPRIMIR_RECORDS	27
4.4.3	AFEGIR_RECORD	29
4.4.4	EMMATZEMA_RECORDS	32
4.4.5	RECUPERA_RECORDS	33
4.4.6	Estructura dels records	34
4.5	GUARDA/CARREGA JOC	35
4.5.1	EMMAGATZEMA_JOC	35
4.5.2	RECUPERA_JOC.....	36
4.6	TRANSCURS DEL JOC.....	37
4.6.1	DISPARA	37
4.6.2	COMPROVA.....	46
4.6.3	POSA_AIGUA	51
4.6.4	CALCULA_PUNTUACIÓ.....	56
4.6.5	Programa principal.....	57
4.7	INSTRUCCIONS	70
5	DISSENY GENERAL EN PSEUDOCODI.....	71
6	AVALUACIÓ	131
7	ANEXO	132

1 Introducció i especificacions generals

Des del primer moment que vaig començar amb el meu projecte final tenia molt clar que volia fer el nivell 3, de tal manera que en cap moment he utilitzat els fitxers que havien posat a la nostra disposició pel nivell 1 i 2.

He utilitzat tot el coneixement que tinc sobre el llenguatge de programació c, utilitzant en tot moment tant els meus recursos de coneixement o laboratoris previs com els que ens brinden les noves tecnologies per a resoldre de la manera més eficient i elegant que he trobat els problemes que se m'han anat plantejant i apareixent , encara que alguns no s'haguessin explicat prèviament a la assignatura.

Entrant més en detall, he utilitzat comandes i estructures que eren més eficients que les "clàssiques" ja que s'adequaven més al problema i donaven menys lloc a l'error que podia produir l'usuari en l'entrada de dades.

També cal dir que moltes de les expressions del c no es podien expressar en pseudocodi de manera que m'he inventat expressions en pseudocodi que emulen aquestes funcions. Aquestes són:

EXPRESSIÓ EN PSEUDOCODI	EXPRESSIÓ EN C
ALEATORI	rand()
NETEJAR PANTALLA	System("cls")
llegirASCII	getchar, getch o getche que guarden el seu resultat en una variable
escriureASCII	putchar
PAUSA	getch() sense guardar el resultat a cap lloc o bé system("pause")

2 Especificacions

2.1 Nivell 1

El nivell 1 o bàsic consisteix a implementar el joc per a la modalitat de cap jugador: la màquina contra ella mateixa.

El joc tindrà els següents elements.

- Menú inicial.
- Configurar del joc per a la màquina (cap jugador): crear un joc nou (*inicia_taulell()*).
- Crear la lògica de joc per descobrir els vaixells de la màquina: la funció *jugar()* indicarà les coordenades de llançament de la màquina. Aquesta funció representa el cervell de la màquina, per tant, es tracta de dissenyar l'algorisme que permet a la màquina endivinar la posició dels vaixells amb el número mínim de trets.
- Dissenyar l'algorisme general de joc per a què pugui jugar la màquina: obtenir les coordenades del següent tret amb la funció *jugar()*, efectuar el tret, amb la funció *dispara()*, marcar al taulell de llançament els resultats, controlar el final del joc...
- Emmagatzemar i recuperar els rècords.

Per dissenyar la lògica de joc de la màquina, heu de dissenyar el procediment *jugar()*. *acció jugar* (*var f:caràcter, var c:enter, dim:enter, taulell_llancament:taula de caràcters*);

El procediment *jugar()* rep, com paràmetres, les coordenades del darrer llençament (*f i c*), les dimensions del taulell (*dim*) i el taulell de llançaments (*taulell_llancaments*) i retornarà, per referència, les noves coordenades per fer el següent llançament (*f i c*). Les coordenades han de ser vàlides. Per emmagatzemar i recuperar els rècord, també heu de dissenyar dos procediments específics:

- *funció emmagatzema_records* (*fitxer_record:taula de caràcters, num_records:enter, records:taula de record_tipus*) retorna booleà.

Aquest procediment emmagatzema al fitxer de records els rècords actuals i informarà de si tot ha anat bé. Els rècords s'emmagatzemen quan finaliza una partida si el resultat de la partida és millor que el darrer rècord. *num_records* indica el nombre total de rècords emmagatzemats.

- *funció recupera_records* (*fitxer_record:taula de caràcters, num_records:enter, records:taula de record_tipus*) retorna booleà.

Aquest procediment recupera els rècords emmagatzemats i informa si tot ha anat bé.

Els rècords es recuperen quan s'executa el joc per primera vegada. S'ha de decidir què fer en cas que no existeixi cap fitxer amb rècords. A la implementació es pot utilitzar el fitxer objecte *vaixells.obj* amb els procediments necessaris per a inicialitzar el taulell de vaixells i per a disparar:

- *acció inicia_taulell* (*dim:enter, taulell_vaixells:taula de caràcters*);

Inicialitza la disposició de tots els tipus de vaixells dins d'un nou taulell de vaixells.

- *funció dispara* (*f:caràcter, c:enter, taulell_vaixells:taula de caràcters*) retorna *enter*;

Retorna el resultat del llançament (repetit-0, aigua-1...) i actualitza el taulell de vaixells amb les anotacions pertinents.

A més dels procediments indicats, podeu dissenyar tots els procediments addicionals que necesiteu.

2.2 Nivell 2

El nivell 2 o intermedi consisteix a completar el joc del nivell 1 per a permetre dues modalitats de joc diferents: cap jugador (juga la màquina) o un jugador (juga l'usuari contra la màquina).

A més a més, permetrà emmagatzemar el joc i continuar la partida a partir d'un joc emmagatzemat.

- Recuperar un joc emmagatzemat.
- Emmagatzemar el joc.

- Dissenyar l'algorisme general de joc per a què pugui jugar l'usuari: demanar a l'usuari les coordenades i confirmar que són vàlides, disparar amb la funció *dispara()*, marcar al taulell de llançament els resultats dels llançaments, controlar el final del joc...

Per a emmagatzemar i recuperar un joc s'han de dissenyar els següents procediments:

- *funció emmagatzema_joc (nom_fitxer:taula de caràcters, dim:enter, num_jugadors:enter, jugadors:taula de jugador_tipus) retorna booleà;*

Aquest procediment emmagatzema el joc actual, per això rep el nom del fitxer binari (*nom_fitxer*), les dimensions del taulell (*dim*), el nombre de jugadors (*num_jugadors*) i la informació dels jugadors (*jugadors*): nom, taulells, total de trets... I retorna si tot ha anat bé.

- *funció recupera_joc (nom_fitxer:taula de caràcters, var dim:enter, var num_jugadors:enter, jugadors:taula de jugador_tipus) retorna booleà;*

Aquest procediment recupera la informació del joc emmagatzemat al fitxer binari que s'indica per paràmetre (*nom_fitxer*) i informarà si tot ha anat bé.

A més de la informació dels jugadors ha d'indicar les dimensions del taulell i el nombre de jugadors.

Per demanar les coordenades al jugador heu de dissenyar el procediment *nova_jugada*.

- *nova_jugada (dim:enter, var coor:coor_tipus);*

El procediment rep les dimensions del joc (*dim*) i retornarà les coordenades per disparar. L'acció no retornarà fins que l'usuari proporcioni unes coordenades vàlides.

A la implementació es pot utilitzar el fitxer objecte *vaixells.obj* amb les funcions necessàries per a crear el taulell de vaixells i disparar: *inicia_taulell()* i *dispara()*.

2.3 Nivell 3

El nivell 3 o avançat consisteix a completar el joc del nivell 2 per a permetre jugar a totes les modalitats de joc:

- Cap jugador (nivell 1)
- Un jugador (nivell 2)
- Dos jugadors.

El joc tindrà els següents elements nous:

- Permetre al jugador triar entre iniciar el taulell de vaixells de forma automàtica (cridant a la funció *inicia_taulell()*) o crear el seu propi taulell de vaixells amb la funció *inicia_elmeu_taulell()*.
- Dissenyar l'algorisme general de joc per a què pugui jugar l'usuari contra la màquina: controlar els torns dels jugadors, demanar les coordenades de forma automàtica amb la funció *jugar()* o de forma manual amb la funció *nova_jugada()*, efectuar els trets, tant de l'usuari com de la màquina, amb la funció *dispara()*, marcar als taulells de llançament els resultats, controlar el final del joc...

S'ha de dissenyar una nova funció que permeti a l'usuari crear el seu propi taulell de vaixells.

- *acció inicia_elmeu_taulell (dim:enter, taulell_vaixells:taula de caràcters);*

Permet al jugador crear el seu propi taulell de vaixells. S'ha de garantir que el taulell sigui vàlid.

A més a més heu de dissenyar els procediments: *inicia_taulell()* i *dispara()*.

S'ha de crear una llibreria *vaixells.lib* amb tots o part dels procediments implements.

3 Anàlisi

3.1 Nivell 1

3.1.1 Menu

He concebut el menú com una funció que retorna un enter depenent de l'opció triada, aquestes opcions però, han de ser vàlides i també s'ha de tenir en compte que algunes opcions poden estar o no disponibles depenent de l'estat en que es trobi el joc (abans de jugar o emmagatzemar un joc aquest ha d'haver sigut creat).

Cal tenir en compte que el menú ja havia sigut dissenyat en la primera fase.

3.1.2 Inicia taulell

Com ja he dit abans, aquesta acció la vaig dissenyar en el primer nivell ja que des d'un principi volia fer el nivell 3 així que la inclouré en el primer nivell.

Aquest procediment consisteix en col·locar tots els vaixells de forma que no es superposin ni es toquin (deixant una casella amb aigua).

Vaig pensar que seria incorrecte (i també avorrit) el fet que el tauler fos sempre igual de manera que vaig estimar com a “motor” d'aquesta funció un generador de nombres aleatoris per a decidir les posicions de cada vaixell. Aquestes posicions aleatòries estarien sempre dins de les caselles de joc indicades per l'usuari i també tindrien una orientació aleatòria, horitzontal o vertical excepte per als vaixells d'una casella que no tenen orientació.

3.1.3 Funció jugar + funció dispara

Com des d'un primer moment vaig dissenyar totes les funcions inclús les que es donaven fetes, vaig decidir que seria més eficient unir les funcions jugar i les funcions dispara en una de sola, ja que són funcions molt relacionades que van una darrere de l'altre. A més la funció dispara realment és molt curta, requereix només d'una línia, de manera que vaig preferir unir-les. A partir d'ara, quan es parli de la funció dispara s'està parlant realment de la IA de la màquina.

A grans trets aquesta funció busca els vaixells del contrari i els intenta enfonsar tots de la manera més eficient. També depèn de nombres aleatoris per a tirar trets quan o ha trobat cap vaixell, en el moment en que en troba un, deixa de disparar aleatòriament i intenta enfonsar quant abans possible el vaixell. Per dir-ho d'una altra manera, té dos estats: l'estat inactivat on dispara aleatòriament i l'estat activat on rastreja el vaixell contrari fins enfonsar-lo. Els dos estats son dinàmics i interconvertibles l'un en l'altre.

3.1.4 Emmagatzema/recupera rècords

Són dues funcions que funcionen a la par, s'encarreguen de llegir un fitxer per emmagatzemar-lo a una taula i al contrari. Cal tenir en compte que el fitxer estigui buit o no existeixi. També tindrem dues funcions relacionades que seran INICIALITZA_TAULA i IMPRIMIR_RECORDS, les quals inicialitzaran la taula de records i els imprimiran per pantalla respectivament.

3.1.5 Control del joc

El joc serà un bucle del que es surti quan algun dels jugadors enfonsi tots els vaixells de l'altre. Això serà així per a tots els nivells.

3.2 Nivell 2

3.2.1 Emmagatzema joc

Aquesta part del programa guardarà totes les variables necessàries per a prosseguir amb el joc en un registre, i guardarà aquest registre en un fitxer binari. (fitxers buits, que no existeixin, etc.).

3.2.2 Recupera joc

Aquesta part del programa és la contrapart de la part anterior, s'encarrega de fer l'invers, llegeix el fitxer (i l'emmagatzema al registre) i després el mateix registre es carrega camp a camp a la seva variable corresponent. Cal tenir en compte el tractament de fitxers (fitxers buits, que no existeixin, etc.).

3.2.3 Nova jugada

El programa demanarà les coordenades a l'usuari fins que aquestes siguin correctes. Serà una estructura de tipus cerca, ja que buscarà que l'usuari introdueixi les dades correctes.

3.3 Nivell 3

3.3.1 Inicia el meu taulell

He pensat en aquesta funció com la mateixa funció INICIA_TAULELL, simplement he substituït el generador de nombres aleatoris per l'entrada de dades de l'usuari i també per codi de comprovació de que aquestes coordenades que indica l'usuari són correctes i vàlides. (Es troben dins del tauler).

3.3.2 Per a efectuar una nova jugada

No he dissenyat una funció específica per a fer una nova jugada, és a dir, per a que l'usuari introdueixi les coordenades tal i com es demanava, però ho he fet des del programa principal perquè així es pot sortir al menú en el moment desitjat.

4 Disseny

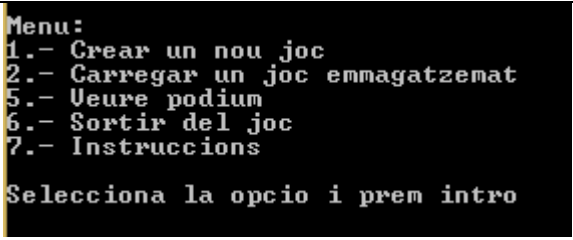
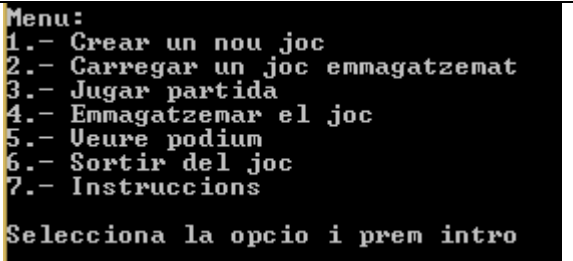
4.1 Menu

He col·locat la funció menú dins d'un bucle infinit de tal manera que l'única manera que té l'usuari de sortir del programa és escollint l'opció 6 (sortir). Les opcions 3 i 4 només es mostren si s'ha carregat o s'ha creat un joc prèviament, ja que aquestes opcions (fora del menú) activen una variable (load) que és la que fa que es mostrin les opcions. Si l'opció triada es incorrecta el bucle es reinicia de tal manera que l'usuari no ha escollit cap opció. Això evita la introducció d'opcions incorrectes. Aquesta funció retorna la variable menú. Aquesta variable és avaluada per un opció (switch) que dirigeix la lectura del codi a una determinada part del programa. El menú és la peça angular del programa.

PSEUDOCODI

```
funció MOSTRAR_MENU(load: enter) retorna enter és
    menu: enter;
    menu:=0;
    mentre ((menu>7) o (menu<1) o (menu=3 i load=0)
o (menu=4 i load=0) o (menu=0))
        escriure("\nMenu:\n");
        escriure("1.- Crear un nou joc\n");
        escriure("2.- Carregar un joc emmagatzemat\n");
        si (load=1)
            escriure("3.- Jugar partida\n");
            escriure("4.- Emmagatzemar el joc\n");
        fsi
        escriure("5.- Veure podium\n");
        escriure("6.- Sortir del joc\n");
        escriure("7.- Instruccions\n\n");
        escriure("Selecciona la opció i prem intro\n\n");
        menullegirASCII();
        menu:=menu-48;
        NETEJAR_PANTALLA
    fmentre
        retorna (menu);
ffunció
```

JOC DE PROVES

PROVA	DESCRIPCIÓ	SORTIDA TEORICA	SORTIDA REAL	OK?
1	Joc sense carregar (load=0)	Sortida amb totes es opcions excepte la 3 i la 4		ok
2	Joc carregat (load=1)	Sortida amb totes les opcion		ok

4.2 Preparació del joc

Aquí es troba el disseny de totes les funcions que s'han d'activar prèviament abans de començar el joc. Inicialitzen matrius, variables i defineixen el tipus de joc o el nombre de caselles que es necessiten per a fer l'anterior.

4.2.1 TIPUS_JOC

En el cas de que l'usuari vulgui crear un nou joc (opció 1) la màquina preguntarà el nombre de jugadors i el nombre de caselles. Aquestes dues opcions es troben dins de dos bucles de control dels quals no es pot sortir a no ser que s'introdueixi una opció vàlida. La variable caselles serà una variable que es passarà per referència per a poder ser modificada. El valor que retornarà la funció es guardarà a la variable opció. Aquesta variable serà avaluada per un opció "switch", que portarà la lectura del codi a la inicialització del taulell. Per l'opció d'un i cap jugador aquesta inicialització es farà automàticament, i per a dos es consultarà a l'usuari si vol crear el seu propi taulell de joc o inicialitzar-lo automàticament, però això ja forma part d'una altra funció.

PSEUDOCODI

```
funció TIPUS_JOC(var caselles: enter) retorna enter és
    opcio: enter;
    opcio:=0;
    caselles:=0;
    mentre ((caselles>10) o (caselles<8))
```

```

        escriure("De quantes caselles vols que sigui
el tauler? min 8 max 10: ");
        llegirASCII(caselles);
        caselles:=caselles-48;
        si (caselles=1)
        mentre (caselles<>0)
        llegirASCII(caselles);
            caselles:=caselles-48;
        fmentre
            escriure("0");
        caselles:=10;
    fsi
NETEJAR_PANTALLA
fmentre
mentre ((opcio<1) o (opcio>3))
    NETEJAR_PANTALLA
        escriure("A quina modalitat de joc vols jugar?\n");
        escriure("1.-Cap jugador\n");
        escriure("2.-Un jugador\n");
        escriure("3.-Dos jugadors\n");
        llegirASCII(opcio);
        opcio:=opcio-48;
        NETEJAR_PANTALLA
    fmentre
retorna(opcio);
funció

```

JOC DE PROVES

Cas	Descripció	Sortida teòrica	Sortida real	Ok?
1	Selecció mida 8	Caselles=8	Caselles=8	Si
2	Selecció mida 9	Caselles=9	Caselles=9	Si
3	Selecció mode de joc 1	Opccio=1	Opció=1	si
4	Introduir qualsevol altre valor no acceptat	El programa no accepta el valor i torna a demanar un valor vàlid	El programa no accepta el valor i torna a demanar un valor vàlid	

4.2.2 RANDOMIZE

Aquesta funció genera aleatoris enters dins d'uns límits definits (les coordenades es troben entre 0 i caselles i la posició pot ser 0 o 1). Se'ls hi passen aquests 3 paràmetres per referència i s'emmagatzemen en les mateixes variables per estar a disposició de la resta del programa. Aquesta funció treballa conjuntament amb INICIA_TAULELL.

PSEUDOCODI

acció RANDOMIZE(var fila: enter, var columna: enter, var direccio: enter, caselles: enter) és

fila:=ALEATORI%(caselles);

columna:=ALEATORI%(caselles);

direccio:=ALEATORI%2; \$0 serà vertical i 1 serà horitzontal

facció

JOC DE PROVES

Per a fer un joc de proves d'aquest procediment s'hauria de jugar per a comprovar que els trets de la màquina en l'estat inactivat són completament aleatoris i no segueix cap patró ja

que depèn d'aquesta funció. Les posicions generades són sempre vàlides i es troben dins del tauler.

4.2.3 INICIA_TAULER

Consisteix en una funció que inicialitza el tauler de la forma adequada. Aquest procediment rep com a paràmetres el tauler de defensa i el taulell d'atac d'un jugador de tal manera que s'ha d'invocar un cop per a cada jugador. En primer lloc plena la matriu d'atac de interrogants i la matriu de defensa d'aigua. Després mitjançant un bucle i un doble comptador inicialitza la matriu amb els vaixells de la següent forma. Crida a la funció RANDOMIZE i li passa per referència les variables f, c i orientació. Això decidirà a partir de quin punt es generarà el vaixell i quina orientació tindrà. Aquestes coordenades sempre seran correctes i dins del tauler que hem escollit. Després d'això el vaixell guardarà les coordenades de les caselles circumdants al vaixell (sempre tenint en compte els límits) i comprovarà que totes aquestes caselles siguin aigua. En el cas de no ho siguin s'activarà un flag que farà que es descarti les coordenades i se'n generin de noves. En cas de que sí el vaixell es col·locarà i es modificaran els corresponents comptadors. Aquest procediment es repetirà fins haver col·locat tots els vaixells.

Una de les problemàtiques que vaig trobar va ser que es podia donar el cas (en taulells de 8 i 9 caselles) de que els vaixells, al col·locar-se aleatòriament no optimitzessin l'espai de manera que en ocasions la funció es quedava en bucle infinit. Per a solucionar això vaig incloure la variable control, que augmenta en u cada cop que hi ha una iteració. Normalment quan es col·loquen de forma correcta aquesta variable arriba a valors pròxims a 100. En el cas de que es quedi en infinit aquesta variable arriba a valors molt grans i no deixa de créixer. Per això he inclòs una part del codi que revisi que aquesta variable sigui mil. En cas de que ho sigui la funció torna a iniciar-se començant de nou, de manera que fins que no s'han col·locat tots els vaixells aquesta funció no acaba.

PSEUDOCODI

```
acció INICIA_TAULELL(var matriu[10][10]: matriu de  
caràcters, var matriuk[10][10]: matriu de caràcters, caselles:  
enter) és
```

```
control, vaixell, comptador, f, c, direccio, superior,  
dreta, inferior, esquerra, correcte, i, j: enter;
```

```
repeteix:
```

```
    i:=0;
```

```
    j:=0;
```

```
    control:=0;
```

```
    correcte:=0;
```

```
    vaixell:=3;
```

```
    comptador:=0;
```

```

mentre (i<10) $Inicialitza la matriu amb punts
    mentre (j<10)
        matriu[i][j]:='.';
        j++;
    fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (i<10) $Inicialitza la matriu amb ?
    mentre (j<10)
        matriuk[i][j]:='?';
        j++;
    fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (vaixell<>-1)
    i:=0;
    j:=0;
    mentre (correcte=0)
        correcte:=1;
        RANDOMIZE(f,c, direccio, caselles);
        si (((f+vaixell)>=caselles) i (direccio=1)) o
        (((c+vaixell)>=caselles) i (direccio=0)))
            correcte:=0;
        fsi
        si (direccio=0)
            superior:=f-1; si (superior=-1)superior++;}
            inferior:=f+1; si (inferior=caselles)inferior-
-; }

            esquerra:=c-1; si (esquerra=-1)esquerra++;}

```

```

        dreta:=c+vaixell+1; si (dreta=caselles)dreta--
; }

sino
    superior:=f-1; si (superior=-1)superior++;}
    inferior:=f+vaixell+1; si
(inferior=caselles)inferior--;}
    esquerra:=c-1; si (esquerra=-1)esquerra++;}
    dreta:=c+1; si (dreta=caselles)dreta--;}
fsi
mentre (esquerra<=dreta)
    mentre (superior<=inferior)
        si (matriu[superior][esquerra]<>'.' )
correcte:=0; fsi
        superior++;
        i++;
    fmentre
    superior:=superior-i;
    i:=0;
    esquerra++;
fmentre
control++;
si (control>=2000)
    goto repeteix;
fsi
fmentre
correcte:=0;
si (direccio=0)
    mentre (j<=vaixell)
        matriu[f][c+j]:='X';
        j++;
    fmentre
sino
    mentre(j<=vaixell)
        matriu[f+j][c]:='X';

```

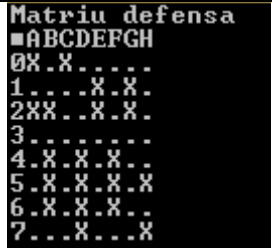


```

        j++;
    fmentre
    fsi
    comptador++;
    si (comptador=(4-vaixell))
        comptador:=0;
        vaixell--;
    fsi
fmentre
facció

```

JOC DE PROVES

prova	descripcio	Sortida esperada	Sortida real	Ok?
1	Prova de fum	Tots els vaixells col·locats de manera correcta sense que es toquin		ok

4.2.4 INICIA_EL_MEU_TAULER

Aquesta acció és molt similar a la anterior INICIA_TAULELL, la diferència és que no hi ha generador de nombres aleatoris, sinó que és l'usuari qui els col·loca. El programa té en compte els límits del tauler i els errors que pugui fer l'usuari.

PSEUDOCODI

```

acció INICIA_EL_MEU_TAULELL(var matriu[10][10]: taula de
caràcters,var matriuk[10][10]: taula de caràcters, caselles:
enter) és

```

```

    comprova,control, vaixell, comptador, f, c, direccio,
superior, dreta, inferior, esquerra, correcte, i, j: enter;

```

```

    i:=0;

```

```

    j:=0;

```

```

    control:=0;

```

```

    correcte:=0;

```

```

    vaixell:=3;

```

```

direccio:=-1;
c:=-1;
f:=-1;
comptador:=0;
comprova:=0;
mentre (i<10) $Inicialitza la matriu amb punts
    mentre (j<10)
        matriu[i][j]:='.';
        j++;
fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (i<10) $Inicialitza la matriu amb ?
    mentre (j<10)
        matriuk[i][j]:='?';
        j++;
    fmentre
        j:=0;
        i++;
fmentre
i:=0;
mentre (vaixell<>-1)
    i:=0;
    j:=0;
    mentre (correcte=0)
        si ((correcte=0) i (control<>0))
            escriure("\nEl vaixell a les coordenades que
has introduit no s'ha pogut colocar...");
            PAUSA();
        fsi
    NETEJAR_PANTALLA
    VISUALITZASIMPLE(matriu, caselles);

```

```

        escriure("\nEstem col·locant un vaixell de ' '
caselles", vaixell+1);
        mentre ((f<0) o (f>=caselles))
            si (comprova>0)
                escriure("\nFila incorrecta gamarus!!!");
            fsi
            escriure("\nEn quina fila vols col·locar el
vaixell?: ");
            llegirASCII(f);
            f:=f-48;
            comprova++;
        fmentre
        comprova:=0;
        mentre ((c<0) o (c>=caselles))
            si (comprova>0)
                escriure("\nColumna incorrecta cap de
pardals!!!");
            fsi
            escriure("\nEn quina columna vols col·locar el
vaixell?: ");
            llegirASCII(c);
            si (c>=97)
                c:=c-32;
            fsi
            c:=c-65;
            comprova++;
        fmentre
        comprova:=0;
        si (vaixell<>0)
            mentre ((direccio<>1) i (direccio<>0))
                si (comprova>0)
                    escriure("\nDireccio incorrecta
borinot!!!");
                fsi

```

```

        escriure("\nEn quina direccio vols colocar
el vaixell?: (H/V)");
        llegirASCII(direccio);
        si      (direccio>=97)      direccio:=direccio-
32;fsi

        si (direccio=72) direccio:=0;fsi
        si (direccio=86) direccio:=1;fsi
        comprova++;
        fmentre
        sino
            direccio:=1; //valor arbitrari ja que en
vaixells d'una sola casella no te importancia
            fsi
            correcte:=1;
            comprova:=0;
            si (((f+vaixell)>=caselles) i (direccio=1)) o
(((c+vaixell)>=caselles) i (direccio=0))
                correcte:=0;
            fsi
            si (direccio=0)
                superior:=f-1;si (superior=-1)superior++;fsi
                inferior:=f+1; si (inferior=caselles)inferior-
-;fsi

                esquerra:=c-1; si (esquerra=-1)esquerra++;fsi
                dreta:=c+vaixell+1; si (dreta=caselles)dreta--
;fsi

            sino
                superior:=f-1;si (superior=-1)superior++;fsi
                inferior:=f+vaixell+1;                                si
(inferior=caselles)inferior--;fsi
                esquerra:=c-1; si (esquerra=-1)esquerra++;fsi
                dreta:=c+1; si (dreta=caselles)dreta--;fsi
            fsi
        mentre (esquerra<=dreta)
            mentre (superior<=inferior)

```

```

                                si          (matriu[superior][esquerra]<>'.')
correcte:=0; fsi
                                superior++;
                                i++;
                                fmentre
                                superior:=superior-i;
                                i:=0;
                                esquerra++;
                                fmentre
                                control++;
                                si (correcte=0)
                                    direccio:=-1;
                                    c:=-1;
                                    f:=-1;
                                fsi
                                fmentre
                                si (direccio=0)
                                    mentre (j<=vaixell)
                                        matriu[f][c+j]:='X';
                                        j++;
                                    fmentre
                                sino
                                    mentre(j<=vaixell)
                                        matriu[f+j][c]:='X';
                                        j++;
                                    fmentre
                                fsi
                                comptador++;
                                si (comptador=(4-vaixell))
                                    comptador:=0;
                                    vaixell--;
                                fsi
                                control:=0;
                                correcte:=0;

```


```

    direccio:=-1;
    c:=-1;
    f:=-1;

    fmentre
        escriure("\nJa has colocat tots els vaixells!! si fa
patxoca... pero segur que perds :P");
    facció

```

JOC DE PROVES

PROVA	DESCRIPCIÓ	SORTIDA TEORICA	SORTIDA REAL	OK?
1	Valors incorrectes	Missatge d'error	Missate d'error	ok
2	Col·locar un vaixell en una posició correcta però una orientació que el fa estar fora del tauler	Missatge d'error	El teu vaixell no s'ha pogut col·locar.	ok
3	Valors correctes en posició correcta (casella 4c orientació horitzontal)	Col·locació del vaixell		ok

4.2.5 MONEDA

Em va sembla una mica absurd que sempre comencés el mateix jugador, de tal manera que vaig dissenyar un procediment que simula el llançament d'una moneda. L'usuari escull si vol

ser cara o creu i llavors un aleatori “genera” la moneda, en cas de que coincideixi comença el jugador 1, si s’equivoca amb la predicció comença el jugador 2. Aquesta funció retorna un valor que s’emmagatzema en la variable tira, que indica de qui és el torn.

PSEUDOCODI

```
funció MONEDA()
    retorna enter és
opcio: caràcter;
flag, opciok, guanya: enter;
flag:=0;
mentre (flag=0)
    escriure("Que vols ser? Cara o creu?  C/X ");
    llegirASCII(
opcio);
    si((opcio='C')  o (opcio='X')  o (opcio='x')  o (opcio='c'))
    si (opcio='c') opcio:='C';fsi
    si(opcio='x') opcio:='X';fsi
    flag:=true;
    sino
        NETEJARPANTALLA;
    fsi
fmentre
    escriure("\nApreta qualsevol tecla per a tirar la moneda: ");
    PAUSA;
    opciok:=ALEATORI%2;
    si (opciok=1)
        escriure("C \n");
        si (opcio='C')
            escriure("\nComença el jugador 1\n");
            guanya:=1;
        sino
            escriure("\nComença el jugador 2\n");
            guanya:=0;
        fsi
    sino
```

```

    escriure("X\n");
    si (opcio='X')
escriure("\nComenca el jugador 1\n");
        guanya:=1;
sino
escriure("\nComenca el jugador 2\n");
guanya:=0;
    fsi
fsi
si (guanya=0) guanya:=2;}
PAUSA;
NETEJAR_PANTALLA;
retorna(guanya);
ffunció

```

JOC DE PROVES

El programa només permet l'entrada dels valors acceptats per tant en cap moment es produeix un error degut a l'entrada. Crec que un joc de proves no és realment necessari.

4.3 Visualització de matrius

Per a veure les matrius de joc he dissenyat dues accions que serveixen per a això depenent de les necessitats del programa.

4.3.1 VISUALITZA_SIMPLE

Aquesta funció imprimeix per pantalla una sola matriu de joc (aquesta l'he utilitzat per al mode d'un jugador ja que no té sentit imprimir res més que la matriu d'atac de l'usuari).

PSEUDOCODI

```

acció VISUALITZASIMPLE(var matriujocaa[10][10]: taula de
caràcter,caselles: enter)es
    i, k: enter;
    k:=0;
    i:=0;

```



```

    escriure("Matriu atac\n");
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre
    i:=0;
    escriure("\n");
    mentre (i<caselles)
        escriure("",i);
        mentre(k<caselles)
            escriure("",matriujocaa[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("\n");
        i++;
    fmentre
facció

```

JOC DE PROVES

La matriu es visualitza perfectament:



4.3.2 VISUALITZA

Aquesta funció imprimeix per pantalla dos taulers de joc un al costat de l'altre.

PSEUDOCODI

```

acció    VISUALITZA(var    matriudefensa[10][10]:    taula    de
caràcters,var matriuatac[10][10]: taula de caràcters,caselles:
enter, jugador: enter) és

```

```

i, k: enter;
k:=0;
i:=0;
    escriure("Jugador          '\nMatriu          atac
",jugador);
    si (caselles=9)
        escriure(" ");
    fsi
    si (caselles=10)
        escriure(" ");
    fsi
    escriure("Matriu defensa\n");
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre
    escriure("          ");
    i:=0;
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre
    i:=0;
    escriure("\n");
    mentre (i<caselles)
        escriure("%i",i);
        mentre(k<caselles)
            escriure("%c",matriuatac[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("          ");

```

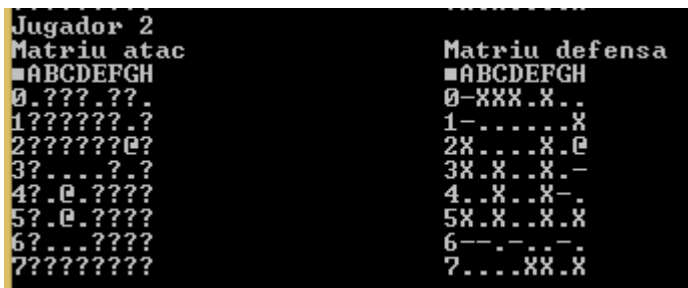
```

        escriure("%i",i);
        mentre (k<caselles)
            escriure("%c",matriudefensa[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("\n");
        i++;
    fmentre
facció

```

JOC DE PROVES

El procediment funciona perfectament. Exemple:



```

Jugador 2
Matriu atac
■ABCDEFGH
0.????.??
1???????.?
2???????.?
3?.....?
4?.@.????
5?.@.????
6?..????
7????????

Matriu defensa
■ABCDEFGH
0-XXX.X..
1-.....X
2X...X.@
3X.X..X-
4..X..X-
5X.X..X.X
6--.-.-.
7....XX.X

```

4.4 Tractament de records

Per a tractar els records he dissenyat diferents funcions per a tenir en compte tota la casuística dels records.

4.4.1 INICIALITZA_TAULA

Inicialitza la taula amb els valors buits de puntuació i la taula de nom amb espais. Això farà que després de carregar l'arxiu a la nostra taula, en cas de que no hi hagin deu records; a l'hora d'imprimir els rècords no s'imprimeixin les posicions buides de la taula.

PSEUDOCODI

```

acció    INICIALITZA_TAULA (var    records[10]:    taula    de
records_tipus) és
i,k: enter;
k:=0;
mentre (k<=9)
    records[k].puntuacio:=-666;
    k++;

```

```

fmentre
k:=0;
i:=0;
mentre (i<=9)
    mentre (k<20)
        records[i].nom[k]:=' ';
        k++;
    fmentre
k:=0;
i++;
fmentre
facció

```

JOC DE PROVES

Després d'inicialitzar la taula i imprimir els records, no se'n mostra cap per tant la taula s'inicialitza correctament.

4.4.2 IMPRIMIR_RECORDS

Simplement imprimeix els records per pantalla. El valor -666 és el que indica a la funció que es tracta d'una posició buida i per tant no la imprimeix. Aquesta funció també imprimirà un missatge d'advertència si no hi han records, és a dir, si totes les posicions són -666.

PSEUDOCODI

```

acció IMPRIMIR_RECORDS (var records[10]: taula de
records_tipus) és
flag,i, k: enter;
i:=0;
k:=0;
flag:=0;
NETEJAR_PANTALLA
mentre (i<=9)
    si (records[i].puntuacio<>-666)
        flag:=1;
    fsi
    i++;
fmentre

```

```

si (flag=1)
    i:=0;
    escriure ("Nom                                Puntuacio\n");
mentre (i<=9)
    si (records[i].puntuacio<>-666)
        flag:=1;
        mentre (k<>20)
            escriure ("%c", records[i].nom[k]);
            k++;
        fmentre
        escriure ("%i", records[i].puntuacio);
        escriure ("\n");
        k:=0;
    fsi
    i++;
fmentre
sino
    escriure("El fitxer de records esta buit o no existeix,
prova a afegir un record jugant.");
fsi
facció

```

JOC DE PROVES

Cas	Descripc io	Sortida teorica	Sortida real	Ok?
1	Fitxer buit	Missatge d'error	El fitxer de records esta buit o no existeix, afegeix records jugant	Si
2	Un record	Records: maleitpseud ocodi: 600	Records: maleitpseud ocodi: 600	Si
3	Fitxer inexistent	Missatge d'error	El fitxer de records esta buit o no existeix,	si

			afegeix records jugant	
4	Uns quants records	Pubis 600, jo_69 500, monstreee 89	Pubis 600, jo_69 500, monstreee 89	si

4.4.3 AFEGIR_RECORD

Afegeix un record i ordena la taula. Aquesta funció rep com a paràmetre la puntuació i comprova que la puntuació rebuda és igual o més gran que les que hi ha actualment a la taula de records. Si és així pregunta a l'usuari si vol afegir un record i demana el nom a l'usuari si l'opció és que si per a procedir a afegir el seu rècord a la taula. Per a fer-ho, el programa compara la puntuació rebuda amb totes les posicions de la taula i la seva immediatament superior. En la posició 0 no compara amb la superior sinó que només comprova que aquesta sigui igual o inferior. En cas de trobar una posició amb una puntuació igual o inferior i que la seva immediatament superior sigui més petita (excepte en la posició 0 com ja he dit), totes les posicions de a taula fins a la posició candidata són mogudes una posició a baix i la posició candidata queda lliure. El records de la posició 9 es perd en cas de que n'hi hagi un. Després d'això el rècord es guarda a la posició candidata. El programa retorna un flag que indica si s'ha afegit un rècord o no per a que el programa principal mostri el missatge corresponent.

PSEUDOCODI

```
funció AFEGIR_RECORD(puntuacio: enter, var records[10]: taula
de records_tipus, jugador: enter) retorna booleà és
record: records_tipus;
opcio: record, lectura: caràcter;
i, j, k: enter;
flag: booleà;
flag:=0;
i:=0;
j:=0;
opcio: record:='k';
lectura:='k';
mentre (i<>10)
    si (records[i].puntuacio<=puntuacio) flag:=1; si
    i++;
fmentre
si (flag=1)
    flag:=0;
```

```

    mentre ((opciorecord<>'s') i (opciorecord<>'S') i
(opciorecord<>'N') i (opciorecord<>'n'))
        NETEJAR_PANTALLA
        escriure("Jugador '", vols actualitzar els record? S/N
", jugador);
        llegirASCII(opciorecord);
    fmentre
    si ((opciorecord='s') o (opciorecord='S'))
        i:=0;
        escriure("\nIntrodueix el nom del jugador ' ' (20
caracters com a maxím) i apreta intro: ", jugador);
        mentre ((lectura<>'r') i (i<20))
            llegirASCII(lectura);
            si (lectura<>'r') record.nom[i]:=lectura;fsi
            i++;
        fmentre
        i--;
        mentre (i<>20)
            record.nom[i]:=' ';
            i++;
        fmentre
        NETEJAR_PANTALLA
        record.puntuacio:=puntuacio;
        i:=9;
        k:=8;
        mentre (i>=0)
            si (((records[i-1].puntuacio>record.puntuacio) i
(records[i].puntuacio<=record.puntuacio)) o
((records[i].puntuacio<=record.puntuacio) i (i=0)))
                flag:=1;
                mentre (k<>(i-1))

records[k+1].puntuacio:=records[k].puntuacio;
                mentre (20<>j)

records[k+1].nom[j]:=records[k].nom[j];

```

```

        j++;
    fmentre
    k--;
    j:=0;
    fmentre
    records[i].puntuacio:=record.puntuacio;
    k:=0;
    mentre (k<>20)
        records[i].nom[k]:=record.nom[k];
        k++;
    fmentre
    fsi
    i--;
fmentre
    fsi
fsi
retorna (flag);
funció

```

JOC DE PROVES

Cas	Descripció	Sortida teòrica	Sortida real	Ok?
1	Inserció nou record Sonia: 745	Records: Sonia: 745 Tirant lo blanc: 600 Joanot: 585	Records: Sonia: 745 Tirant lo blanc: 600 Joanot: 585	Si
2	Inserció nou record	Records:	Records:	Si

	Cuco: 656	Sonia: 745	Sonia: 745	
		Cuco:656	Cuco:656	
		Tirant lo blanc: 600	Tirant lo blanc: 600	
		Joanot: 585	Joanot: 585	

La funció funciona per a tots els casos.

4.4.4 EMMATZEMA_RECORDS

Aquesta funció obre un fitxer en mode escriptura i guarda la taula de records en aquest.

PSEUDOCODI

```

funció EMMAGATZEMA_RECORDS(var records[10]: taula de
records_tipus) retorna enter
    n, i, j: enter
    f: fitxer;
    f:=obrir("records.txt","w");
    i:=0;
    j:=0;
    si (f<>NULL)
        n:=0;
        mentre (i<>10)
            mentre (j<>20)
                escriure(f,records[i].nom[j]);
                j++;
            fmentre
                fescriure(f, '\t');
                fescriure(f, records[i].puntuacio);
                fescriure(f, '\n');
                j:=0;
                i++;
            fmentre
        sino
            n:=-1;
    fsi

```

```

    fclose(f);
    retorna(n);
funció

```

JOC DE PROVES

PROVA	DESCRIPCIO	SORTIDA ESPERADA	SORTIDA REAL	Ok?
1	Fitxer inexistent	Es crea el fitxer	Es crea el fitxer	ok
2	Fitxer existent	S'actualitza amb els nous records	S'actualitza amb els nous records	ok
3	Fitxer existent però buit	S'actualitza amb l'estructura de la taula de records buida	S'actualitza amb l'estructura de la taula de records buida	ok

4.4.5 RECUPERA_RECORDS

Aquesta funció recupera els records des de l'arxiu de records. Si el fitxer no existeix o està buit aquest no es tracta i simplement no passa res, ja que aquesta situació es veurà reflexada a l'acció `imprimir_records`.

JOC DE PROVES

```

funció RECUPERA_RECORDS(var records[10]: taula de
records_tipus) retorna enter és
    i,k,n: enter;
    lectura: caràcter;
    i:=0;
    k:=0;
    f: fitxer;
    f:=obrir("records.txt","r");
    si (f=NULL)
        n:=-1;
    sino
        mentre (nofinaldefitxer(f))
            mentre (k<>21)
                llegirf(f, "%c", lectura);

```

```

                si                                     (lectura<>'\\t')
records[i].nom[k]:=lectura;fsi
                k++;
fmentre
    llegirf(f, "%i", records[i].puntuacio);
    llegirf(f, "%c", lectura);
    k:=0;
    i++;
fmentre
    tancar(f);
fsi
retorna(n);
ffunció

```

JOC DE PROVES

PROVA	DESCRIPCIO	SORTIDA ESPERADA	SORTIDA REAL	Ok?
1	Fitxer inexistent	No succeeix res	No seuceeix res	ok
2	Fitxer existent	S'actualitza la taula amb els records del fitxer	S'actualitza la taula amb els records del fitxer	ok
3	Fitxer existent però buit	No succeeix res	No succeeix res	ok

4.4.6 Estructura dels records

La taula de records serà una taula de tipus `tipus_records` que consistirà en dos camps, un de puntuació i un altre de nom, que alhora serà una taula de caràcters de vint espais.

PSEUDOCODI

```

tipus
registre records_tipus és
nom[20]: caràcter;
puntuacio: enter;

```

```
fregistre  
ftipus
```

4.5 Guarda/Carrega joc

Per a guardar/carregar el joc utilitzo dues funcions que van de la mà i escriuen/llegeixen en un fitxer binari. Per a utilitzar-les cal prémer el tabulador mentre es juga (en el moment en que s'anuncia el resultat de la jugada).

4.5.1 EMMAGATZEMA_JOC

Aquesta funció obre un arxiu en mode d'escriptura i hi escriu un registre que rep la funció com a paràmetre que conté totes les variables necessàries per a continuar el joc des d'un punt determinat. Prèviament a iniciar la funció aquest registre es carrega amb les variables adequades. La funció retorna n amb valor 1, que indica si tot ha anat bé per a que el programa principal mostri el missatge corresponent, si no, retorna n amb valor 0.

PSEUDOCODI

```
funció EMMAGATZEMA_JOC (var dades: tipus dadesjoc) retorna  
booleà és  
    f: fitxer;  
    n: enter;  
    n:=0;  
    f:=obrir("partida.bin", "w");  
    si (f<>NULL)  
        n:=escriuref(f, dades);  
    fsi  
    tancar(f);  
retorna(n);  
ffunció
```

JOC DE PROVES

Cas	Descripció	Sortida teòrica	Sortida real	Ok?
1	Guardar dades indispensables pel	Caselles: 9 opcio: 1	caselles: 9 opcio: 1	Si

	funcionament del joc	encerts: 3 llançaments: 42	encerts: 3 llançaments: 42	
2	No existeix el fitxer	Es crea	Es crea	Si
3	Existeix amb una altra partida i volem guardar la nostra	S'actualitza	S'actualitza	ok

4.5.2 RECUPERA_JOC

Aquesta funció fa l'invers de la funció anterior, obre un fitxer i si no existeix retorna n amb valor 0, en cas de que estigui buit també retorna 0. Cal fixar-se en que en aquesta funció el comandament tancar esta dins del condicional ja que si el fitxer no s'obre no s'ha de tancar. He tingut alguns errors amb això. A la funció anterior o passa ja que els fitxers que s'obren en mode w sempre es poden obrir ja que si no existeixen son creats.

PSEUDOCODI

funció RECUPERA_JOC(var dades: tipus dadesjoc) retorna booleà és

```
f: fitxer;
n: enter;
n:=0;
f:=obrir("partida.bin","r");
si (f<>NULL)
    nllegirASCII(f, dades);
    tancar(f);
fsi
retorna(n);
```

ffunció

JOC DE PROVES

Cas	Descripció	Sortida teòrica	Sortida real	Ok?
-----	------------	-----------------	--------------	-----

1	carregar dades indispensables pel funcionament del joc	Caselles: 9 opcio: 1 encerts: 3 llançaments: 42	caselles: 9 opcio: 1 encerts: 3 llançaments: 42	Si
2	No existeix el fitxer	Missatge error	Missatge error	Si
3	Fitxer buit	Missatge error	Missatge error	ok

4.6 Transcurs del joc

El control del joc succeeix principalment entre el main i la funció DISPARA, el qual utilitza funcions complementaries per a disparar, plenar caselles d'aigua, generar els aleatoris, calcular la puntuació, comprovar si el vaixell ha estat tocat o tocat i enfonsat, etc. Aquestes funcions reben una gran quantitat de paràmetres de tipus enters que indiquen diferents tipus d'informació de l'estat del joc, la jugada feta, el torn, la fila i columna que s'ha atacat. Gairebé totes són per referència per a que si són modificades es pugui accedir a elles en la jugada següent.

4.6.1 DISPARA

La funció dispara està fusionada amb la funció jugar ja que em vaig adonar que eren complementaries, encara que quan l'usuari fa una jugada, es marca el taulell al mateix main sense cridar a cap funció.

La funció dispara té dos estats, un estat inactiu on dispara caselles aleatòriament (sense repetir) i un altre on després d'haver fet tocat rastreja el vaixell per a enfonsar-lo. La variable dispara rep com a paràmetres per referència la posició de l'últim tret i la orientació teòrica del vaixell i també la continuació del vaixell per a saber si segueix cap a dalt, cap baix, a la dreta o a l'esquerra. Això es reflexa a les variables vertical i horitzontal depenent de si són 1 o 0, en el cas de que no es sàpiga seran -1 i es generarà aleatòriament una continuació i orientació hipotètica, en les següents jugades el programa disparar en aquetes caselles de continuació hipotètica que determinen la orientació del vaixell, en cas de que la orientació del vaixell hipotètica no sigui la correcta (es veurà reflexat en el tret següent) el valor de la variable canviarà per indicar a la IA per on continua el vaixell. Això es per a tots els casos de manera que la IA de la màquina en l'estat activat són un conjunt de estaments condicionals anàlegs els uns als altres per a les diferents direccions que pot prendre el vaixell. En tot moment es té en compte que no es poden comprovar caselles fora del tauler. Aquesta funció es troba dins d'un bucle que es repeteix fins que no es tira en una casella.

Per a que s'entengui millor faré una descripció completa d'una jugada: Suposem que el programa està disparant aleatòriament i toca un vaixell (suposarem que es un vaixell gran, de 3 o 4 caselles i el toca pel mig). En aquell moment entra en l'estat activat, com no sap

l'orientació hipotètica que pot tenir genera un aleatori que sigui 0 o 1, que volen dir horitzontal o vertical respectivament. Suposem que es genera 1, com no sabem cap a on disparar (dreta o esquerra) es genera un aleatori a la variable horitzontal que ho indica. Es genera 1 i llavors la màquina comprova que la posició a la dreta de la tocada estigui dins del taulell, si no ho fos o fos aigua canviaria a 0 per atacar l'Esquerra, on passaria el mateix. En el moment en que les dues caselles contigües (dreta i esquerra) son aigua o estan fora del taulell l'orientació canvia i es posa a buscar en vertical (orientació=1). Succeeix el mateix, es busca per dalt o per baix a partir d'un aleatori generat entre 0 i 1. Si la màquina dispara per dalt i fa tocat es guardarà la casella acabada de tocar i a la següent jugada es seguirà disparant per dalt fins fer tocat i enfonsat, fora de taulell o aigua i llavors els dispars continuaran per baix. En el moment en que sortim d'una direcció o orientació hipotètica la variable que ho indica torna a ser -1 per a no interferir en la jugada següent. Això és a grans trets com funciona aquest procediment. Bàsicament són molts condicionals un dins dels altres per a comprovar tots els casos possibles.

JOC DE PROVES

```
funció DISPARA(var defensa[10][10]: taula de caràcters, var
atac[10][10]: taula de caràcters, f: enter, c: enter,
jugada:enter, caselles:enter, orientacio:enter,
vertical:enter, horitzontal:enter, noacabat:enter) retorna
enter és

tirada, fila, columna, correcte:enter;
correcte:=0;
tirada:=0;
mentre (tirada=0)
si ((jugada=1) i (noacabat=0))
    mentre (correcte=0)
        f:=ALEATORI%(caselles);
        c:=ALEATORI%(caselles);
        si (atac[f][c]='?') correcte:=1;}
fmentre
si (defensa[f][c]='X')
    defensa[f][c]:='@';
    atac[f][c]:='@';
    fila:=f;
    columna:=c;
    jugada:=COMPROVA(defensa, fila, columna, caselles,
tirada);
fsi
si (defensa[f][c]='.')
    tirada:=1;
```

```

        defensa[f][c]:='-';
        atac[f][c]:='.';
        jugada:=1;
    fsi
    tirada:=1;
fsi
si (((jugada=2) o (noacabat=1)) i (tirada=0))
    si (orientacio=-1)
        orientacio:=ALEATORI%2;
    fsi
    si (orientacio=0)
        si (horitzontal=-1)
            horitzontal:=ALEATORI%2;
        fsi
        si (horitzontal=0)
            si (c-1>=0)
                si (atac[f][c-1]='?')
                    si ((defensa[f][c-1]='X') i (tirada=0))
                        defensa[f][c-1]:='@';
                        atac[f][c-1]:='@';
                        c:=c-1;
                        fila:=f;
                        columna:=c;
                        jugada:=COMPROVA(defensa,          fila,
columna, caselles, tirada);
                        orientacio:=0;
                    si (jugada=3)
                        orientacio:=-1;
                        horitzontal:=-1;
                    fsi
                    tirada:=1;
                fsi
            si (((defensa[f][c-1]='.') i (tirada=0))
                defensa[f][c-1]:='-';

```



```

        atac[f][c-1]:='.';
        horitzontal:=1;
        jugada:=1;
        tirada:=1;
        si (c+1<caselles)
            si ((atac[f][c-1]='.') i
(atac[f][c+1]='.'))
                orientacio:=1;
                horitzontal:=-1;
            fsi
        sino
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    si ((atac[f][c-1]='@') i (tirada=0))
        c:=c-1;
        orientacio:=0;
    fsi
    si ((atac[f][c-1]='.') i (tirada=0))
        horitzontal:=1;
        si ((atac[f][c-1]='.') i
(atac[f][c+1]='.'))
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino
        horitzontal:=1;
        si (atac[f][c+1]='.')
            orientacio:=1;
            horitzontal:=-1;
        fsi

```

```

        fsi
sino $horitzontal:=1
si (c+1<caselles)
    si (atac[f][c+1]='?')
        si ((defensa[f][c+1]='X') i (tirada=0))
            defensa[f][c+1]:='@';
            atac[f][c+1]:='@';
            c:=c+1;
            fila:=f;
            columna:=c;
            jugada:=COMPROVA(defensa, fila,
columna, caselles, tirada);
            orientacio:=0;
            tirada:=1;
            si (jugada=3)
                orientacio:=-1;
                horitzontal:=-1;
            fsi
        fsi
    si ((defensa[f][c+1]='.') i (tirada=0))
        defensa[f][c+1]:='-';
        atac[f][c+1]:='.';
        horitzontal:=0;
        jugada:=1;
        tirada:=1;
        si (c-1>=0)
            si ((atac[f][c-1]='.') i
(atac[f][c+1]='.'))
                orientacio:=1;
                horitzontal:=-1;
            fsi
        sino
            orientacio:=1;
            horitzontal:=-1;

```

```

        fsi
    fsi
    fsi
    si ((atac[f][c+1]='@') i (tirada=0))
        c:=c+1;
        orientacio:=0;
    fsi
    si ((atac[f][c+1]='.') i (tirada=0))
        horitzontal:=0;
        si ((atac[f][c+1]='.') i (atac[f][c-
1]='.'))
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino
        horitzontal:=0;
        si (atac[f][c-1]='.')
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino $orientacio:=1 vertical per tant
        si (vertical=-1)
            vertical:=ALEATORI%2;
        fsi
    si (vertical=0) //apuntant capa dalt
        si (f-1>=0)
            si (atac[f-1][c]='?')
                si ((defensa[f-1][c]='X') i (tirada=0))
                    defensa[f-1][c]:='@';
                    atac[f-1][c]:='@';
                    f:=f-1;

```

```

        fila:=f;
        columna:=c;
        jugada:=COMPROVA(defensa,          fila,
columna, caselles, tirada);
        orientacio:=1;
        tirada:=1;
        si (jugada=3)
            orientacio:=-1;
            vertical:=-1;
        fsi
    fsi
    si ((defensa[f-1][c]='.') i (tirada=0))
        defensa[f-1][c]:='-';
        atac[f-1][c]:='.';
        vertical:=1;
        jugada:=1;
        tirada:=1;
        si          ((atac[f-1][c]='.')          i
(atac[f+1][c]='.'))
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
    si ((atac[f-1][c]='@') i (tirada=0))
        f:=f-1;
        orientacio:=1;
    fsi
    si ((atac[f-1][c]='.') i (tirada=0))
        vertical:=1;
        si (f+1<caselles)
            si          ((atac[f-1][c]='.')          i
(atac[f+1][c]='.'))
                orientacio:=0;

```

```

        vertical:=-1;
    fsi
    sino
        vertical:=-1;
        orientacio:=0;
    fsi
    fsi
    sino
        vertical:=1;
        si (atac[f+1][c]='.')
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
    sino $vertical:=1
    si (f+1<caselles)
        si (atac[f+1][c]='?')
            si ((defensa[f+1][c]='X') i (tirada=0))
                defensa[f+1][c]:='@';
                atac[f+1][c]:='@';
                f:=f+1;
                fila:=f;
                columna:=c;
                jugada:=COMPROVA(defensa,          fila,
columna, caselles, tirada);
                orientacio:=1;
                tirada:=1;
                si (jugada=3)
                    orientacio:=-1;
                    vertical:=-1;
            fsi
        fsi
        si ((defensa[f+1][c]='.') i (tirada=0))
            defensa[f+1][c]:='-';

```

```

        atac[f+1][c]:='.';
        vertical:=0;
        jugada:=1;
        tirada:=1;
        si (f-1>=0)
            si      ((atac[f-1][c]='.')      i
(atac[f+1][c]='.'))
                orientacio:=0;
                vertical:=-1;
            fsi
        sino
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
    fsi
    si ((atac[f+1][c]='@') i (tirada=0))
        f:=f+1;
        orientacio:=1;
    fsi
    si ((atac[f+1][c]='.') i (tirada=0))
        vertical:=0;
        si (atac[f-1][c]='.')
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
    sino
        vertical:=0;
        si (atac[f-1][c]='.')
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi

```

```

        fsi
    fsi
fsi
si ((jugada=3) i (tirada=0))
    jugada:=1;
    noacabat:=0;
    tirada:=0;
fsi
fmentre
retorna (jugada);
ffunció

```

JOC DE PROVES

La lògica del programa funciona perfectament i el joc de proves que es pot fer és simplement fer una partida al joc i veure que funciona amb lògica sense errors. Hi han massa casos a comprovar però el procediment funciona tal i com es descriu a l'apartat de disseny.

4.6.2 COMPROVA

Aquesta funció s'activa després de tocar un vaixell i comprova si aquest últim tret era només tocat o tocat i enfonsat. Aquesta funció rep les coordenades de l'últim tret i el tauler de defensa i rastreja que totes les posicions contigües al tret estiguin tocadetes "@", en cas de que ho estiguin retorna 3 i sinó retorna 2. Aquests nombres es corresponen amb l'estat del joc.

JOC DE PROVES

```

funció COMPROVA(var defensa[10][10]: taula de caràcters,
f:enter, c:enter,caselles:enter, var tirada:enter)retorna
enter és
jugada, up, down, left, right: enter;
up:=0;
down:=0;
left:=0;
right:=0;
    si (f-1>=0)
        si (defensa[f-1][c]='X')
            jugada:=2;
        fsi
        si ((defensa[f-1][c]='.') o (defensa[f-1][c]='-'))
            up:=1;

```

```

    fsi
    si (defensa[f-1][c]='@')
        si (f-2>=0)
            si (defensa[f-2][c]='X')
                jugada:=2;
            fsi
            si ((defensa[f-2][c]='.') o (defensa[f-
2][c]='-'))
                up:=1;
            fsi
            si (defensa[f-2][c]='@')
                si (f-3>=0)
                    si (defensa[f-3][c]='X')
                        jugada:=2;
                    fsi
                    si ((defensa[f-3][c]='.') o
(defensa[f-3][c]='-'))
                        up:=1;
                    fsi
                    si (defensa[f-3][c]='@') up:=1;}
                sino
                    up:=1;
                fsi
            fsi
        sino
            up:=1;
        fsi
    sino
        up:=1;
    fsi
    si (f+1<caselles)
        si (defensa[f+1][c]='X')
            jugada:=2;
        fsi

```



```

    si ((defensa[f+1][c]='.') o (defensa[f+1][c]='-'))
        down:=1;
    fsi
    si (defensa[f+1][c]='@')
        si (f+2<caselles)
            si (defensa[f+2][c]='X')
                jugada:=2;
            fsi
            si ((defensa[f+2][c]='.') o
(defensa[f+2][c]='-'))
                down:=1;
            fsi
            si (defensa[f+2][c]='@')
                si (f+3<caselles)
                    si (defensa[f+3][c]='X')
                        jugada:=2;
                    fsi
                    si ((defensa[f+3][c]='.') o
(defensa[f+3][c]='-'))
                        down:=1;
                    fsi
                    si (defensa[f+3][c]='@') down:=1;}
            sino
                down:=1;
            fsi
        fsi
    sino
        down:=1;
    fsi
    si (c-1>=0)

```

```

    si (defensa[f][c-1]='X')
        jugada:=2;
    fsi
    si ((defensa[f][c-1]='.') o (defensa[f][c-1]='-'))
        left:=1;
    fsi
    si (defensa[f][c-1]='@')
        si (c-2>=0)
            si (defensa[f][c-2]='X')
                jugada:=2;
            fsi
            si ((defensa[f][c-2]='.') o (defensa[f][c-
2]='-'))
                left:=1;
            fsi
            si (defensa[f][c-2]='@')
                si (c-3>=0)
                    si (defensa[f][c-3]='X')
                        jugada:=2;
                    fsi
                    si ((defensa[f][c-3]='.') o
(defensa[f][c-3]='-'))
                        left:=1;
                    fsi
                    si (defensa[f][c-3]='@') left:=1;}
                sino
                    left:=1;
                fsi
            fsi
        sino
            left:=1;
        fsi
    fsi
sino

```

```

        left:=1;
    fsi
si (c+1<caselles)
    si (defensa[f][c+1]='X')
        jugada:=2;
    fsi
    si ((defensa[f][c+1]='.') o (defensa[f][c+1]='-'))
        right:=1;
    fsi
    si (defensa[f][c+1]='@')
        si (c+2<caselles)
            si (defensa[f][c+2]='X')
                jugada:=2;
            fsi
            si ((defensa[f][c+2]='.') o
(defensa[f][c+2]='-'))
                right:=1;
            fsi
            si (defensa[f][c+2]='@')
                si (c+3<caselles)
                    si (defensa[f][c+3]='X')
                        jugada:=2;
                    fsi
                    si ((defensa[f][c+3]='.') o
(defensa[f][c+3]='-'))
                        right:=1;
                    fsi
                    si (defensa[f][c+3]='@')
right:=1;}
                sino
                    right:=1;
                fsi
            fsi
        sino

```

```

        right:=1;
    fsi
    fsi
    sino
        right:=1;
    fsi
    si ((up=1) i (down=1) i (left=1) i (right=1))
jugada:=3;fsi
    tirada:=1;
    retorna(jugada);
ffunció

```

JOC DE PROVES

PROVA	DESCRIPCIO	Sortida esperada?	Sortida real?	OK?
1	Casella repetida	Casella repetida	Casella repetida	Ok
2	Aigua	Aigua	Aigua	Ok
3	Tocat	Tocat	Tocat	Ok
4	Tocat i enfonsat	Tocat i enfonsat	Tocat i enfonsat	Ok

4.6.3 POSA_AIGUA

Aquesta funció posa aigua al voltant dels vaixells tocats i enfonsats. Rastreja els voltants del vaixell i plena totes les posicions properes d'aigua excepte les que estan marcades amb @.

PSEUDOCODI

```

acció POSA_AIGUA(var atac[10][10]: taula de caràcters,f:
enter,c: enter, caselles: enter) és
    up, down, right, left, i: enter;
    i:=0;
    si (f-1>=0)
        si (atac[f-1][c]='@')
            si (f-2>=0)
                si (atac[f-2][c]='@')
                    si (f-3>=0)
                        si (atac[f-3][c]='@')
                            si (f-4>=0)

```

```

                                up:=f-4;
                                sino
                                up:=f-3;
                                fsi
                                sino
                                up:=f-3;
fsi
                                sino
                                up:=f-2;
                                fsi
                                sino
                                up:=f-2;
                                fsi
                                sino
                                up:=f-1;
                                fsi
sino
                                up:=f-1;
fsi
sino
                                up:=f;
fsi
si (c-1>=0)
    si (atac[f][c-1]='@')
        si (c-2>=0)
            si (atac[f][c-2]='@')
                si (c-3>=0)
                    si (atac[f][c-3]='@')
                        si (c-4>=0)
                            left:=c-4;
                        sino
                            left:=c-3;
                    fsi
                sino
            sino
        sino
    sino

```

```

                left:=c-3;
            fsi
        sino
            left:=c-2;
        fsi
    sino
        left:=c-2;
    fsi
sino
    left:=c-1;
fsi
sino
    left:=c-1;
fsi
sino
    left:=c;
fsi
si (c+1<caselles)
    si (atac[f][c+1]='@')
        si (c+2<caselles)
            si (atac[f][c+2]='@')
                si (c+3<caselles)
                    si (atac[f][c+3]='@')
                        si (c+4<caselles)
                            right:=c+4;
                        sino
                            right:=c+3;
                    fsi
                sino
                    right:=c+3;
                fsi
            sino
                right:=c+2;
            fsi

```

```

        sino
            right:=c+2;
        fsi
    sino
        right:=c+1;
    fsi
sino
    right:=c+1;
fsi
sino
    right:=c;
fsi
si (f+1<caselles)
    si (atac[f+1][c]='@')
        si (f+2<caselles)
            si (atac[f+2][c]='@')
                si (f+3<caselles)
                    si (atac[f+3][c]='@')
                        si (f+4<caselles)
                            down:=f+4;
                        fsi
                    sino
                        down:=f+3;
                    fsi
                sino
                    down:=f+3;
                fsi
            sino
                down:=f+2;
            fsi
        sino
            down:=f+2;
        fsi
    sino

```

```

        down:=f+1;
    fsi
sino
    down:=f+1;
    fsi
sino
    down:=f;
    fsi
    mentre (left<=right)
        mentre (up<=down)
            si
            atac[up][left]:='.';fsi
            up++;
            i++;
            fmentre
            up:=up-i;
            i:=0;
            left++;
        fmentre
    acció

```

JOC DE PROVES

Cas	Descripció	Sortida teorica	Sortida real	Ok?
1	Col·locacio vaixells	<pre> Matriu atac ■ABCDEFGH 0...????? 10.???...? 2..???..0.? 3????..0.? 4????..0.. 5????..0.0 6?.....0 7?.0.???.0 </pre>	<pre> Matriu atac ■ABCDEFGH 0...????? 10.???...? 2..???..0.? 3????..0.? 4????..0.. 5????..0.0 6?.....0 7?.0.???.0 </pre>	Si
2	Col·locacio vaixells	<pre> Matriu atac ■ABCDEFGH 0...????? 1.0.???... 2.0.???..0 3.0...?.0 4...0.?.0 5???...?.0 6???????.. 7???.????? </pre>	<pre> Matriu atac ■ABCDEFGH 0...????? 1.0.???... 2.0.???..0 3.0...?.0 4...0.?.0 5???...?.0 6???????.. 7???.????? </pre>	Si

Els vaixells es rodegen d'aigua correctament, encara que estiguin als límits del taulell.

4.6.4 CALCULA_PUNTUACIÓ

Aquesta funció calcula la puntuació total dels jugadors. Al programa principal hi ha un acumulador que es correspon a la part del sumatori de l'equació següent. Aquesta funció rep com a paràmetre el valor d'aquest sumatori i la resta de valors de l'equació per a calcular-la. En un principi calculava en una sola línia la puntuació i no utilitzava la funció però em vaig

$$puntuació = 100 * \frac{\text{dimensió_taulell}}{\text{total_llançaments}} \sum (\text{resultat_llançament} - 1)$$

adonar que per culpa de la divisió que hi ha no es pot calcular correctament la puntuació (tractem enters i la divisió resulta en un real). De manera que tracto tots els valors dins de la funció com a flotant i després forço que aquest flotant sigui un enter, que és la puntuació, la qual és el valor de retorn de la funció.

PSEUDOCODI

funció CALCULAPUNTUACIO(caselles: enter, puntuacio: enter, llancaments: enter) retorna enter és

```
casellesf, puntuaciof, llancamentsf: real
casellesf:=caselles;
llancamentsf:=llancaments;
puntuaciof:=puntuacio100(casellesf/llancamentsf);
puntuacio:=puntuaciof;
retorna(puntuacio);
```

ffunció

JOC DE PROVES

Cas	Descripcio	Sortida teorica	Sortida real	Ok?
1	Puntuacio	653	653	Si
2	Puntuacio	522	522	Si

Les puntuacions calculades a mà coincideixen amb les que calcula el programa.

4.6.5 Programa principal

El programa principal és la part del codi que interacciona amb l'usuari a l'hora de jugar, però no només això sinó que conté la base de totes les parts del programa, de manera que només posaré aquí la part de pseudocodi que correspon al joc. Per a veure tot el programa principal cal consultar l'apartat 5, disseny general en pseudocodi, on es troba tot el programa.

La part del joc està dins d'un "opció" que dirigeix el codi cap al mode de joc que es vol jugar, (n'hi ha tres casos possibles). Cada apartat difereix en el nombre de jugadors de manera que bàsicament canvia l'entrada (que pot ser per l'usuari o a través d'una funció d'aleatoris). Tots els modes de joc estan dins d'un bucle del que només es surt quan un dels dos jugadors enfonsa tots els vaixells de l'altre. En acabar es fa tot el relacionat amb el fi de joc: Afegiment de records, càlcul de puntuació, reinici de variables per a la pròxima partida, etc.

PSEUDOCODI

```
valor 3:
    opció(opcio)
    valor 1:
        si (monflag=1)
            tira:=MONEDA();
            monflag:=0;
        fsi
    mentre (fi=0)
        mentre (((jugadaa=2 o jugadaa=3) o
tira=1) i (fi=0))
            jugadaa:=DISPARA(matriujocbb,
matriujocab, filaa, columnaa, jugadaa, caselles, orientacioa,
verticala, horitzontala, noacabata);
            si (jugadaa=3)
                POSA_AIGUA(matriujocab, filaa,
columnaa, caselles);
            fsi
            puntuacioa:=puntuacioa+jugadaa-1;
            llancamentsa++;
            NETEJAR_PANTALLA
            VISUALITZA(matriujocaa,
matriujocab, caselles, 1);
            VISUALITZA(matriujocbb,
matriujocba, caselles, 2);
            escriure("\nTorn del jugador 1:");
```

```

                                escriure("\n\nResultat: Jugador 1
casella ''':", filaa, columnaa+65);
                                opció(jugadaa)
                                valor 1:
                                    escriure(" Aigua.");
                                    tira:=2;

                                valor 2:
                                    escriure(" Tocat.");
                                    encertsa++;
                                    si (encertsa<>20)
                                        escriure(" Torna a
tirar.");

                                    fsi
                                    noacabata:=1;
                                    tira:=1;
                                valor 3:
                                    escriure(" Tocat i enfonsat.
");

                                    encertsa++;
                                    si (encertsa<>20)
                                        escriure("Torna a
tirar.");

                                    fsi
                                    noacabata:=0;
                                    tira:=1;

                                fopcio

                                llegeixASCII(escapadetime);
                                si (escapadetime='\t')
                                    NETEJAR_PANTALLA
                                    goto menu;
                                fsi
                                si (encertsa=20)
                                    fi:=1;
                                fsi

```

```

                                fmentre
                                mentre (((jugadab=2) o (jugadab=3) o
(tira=2)) i (fi=0))
                                jugadab:=DISPARA(matriujocaa,
matriujocba, filab, columnab, jugadab, caselles, orientaciob,
verticalb, horitzontalb, noacabatb);
                                si (jugadab=3)
                                POSA_AIGUA(matriujocba,
filab, columnab, caselles);
                                fsi
                                llancamentsb++;
                                NETEJAR_PANTALLA
                                VISUALITZA(matriujocaa,
matriujocab, caselles, 1);
                                VISUALITZA(matriujocbb,
matriujocba, caselles, 2);
                                puntuaciob:=puntuaciob+jugadab-1;
                                escriure("\nTorn del jugador 2:");
                                escriure("\n\nResultat: Jugador 2
casella '':", filab, columnab+65);
                                opció(jugadab)
                                valor 1:
                                    escriure(" Aigua.");
                                    tira:=1;
                                valor 2:
                                    escriure(" Tocat. ");
                                    encertsb++;
                                    si (encertsb<>20)
                                        escriure("Torna a
tirar.");
                                    fsi
                                    noacabatb:=1;
                                    tira:=2;
                                valor 3:
                                    escriure(" Tocat i enfonsat.
");

```

```

                                encertsb++;
                                si (encertsb<>20)
                                    escriure("Torna a
tirar.");

                                fsi
                                noacabatb:=0;
                                tira:=2;

                                fopcio
                                llegeixASCII(escapadatetime);
                                si (escapadatetime='\t')
                                    NETEJAR_PANTALLA
                                    goto menu;

                                fsi
                                si (encertsb=20)
                                    fi:=1;

                                fsi
                                fmentre
                                fmentre
                                    puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);
                                    puntuaciob:=CALCULAPUNTUACIO(caselles,
puntuaciob, llancamentsb);
                                si (encertsa=20)
                                    escriure("\nHa guanyat el jugador 1,
amb %i punts, felicitats!\n", puntuacioa);
                                fsi
                                si (encertsb=20)
                                    escriure("\nHa guanyat el jugador 2,
amb %i punts, felicitats!\n", puntuaciob);
                                fsi
                                PAUSA;

                                si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");fsi
                                si (0=AFEGIR_RECORD(puntuaciob, taula, 2))
escriure("El teu record no s'ha afegit...\n");fsi
                                EMMAGATZEMA_RECORDS(taula);

```

```

IMPRIMIR_RECORDS(taula);
fi:=0;
noacabata:=0;
noacabatb:=0;
encertsa:=0;
encertsb:=0;
jugadaa:=1;
jugadab:=1;
orientacioa:=-1;
orientaciob:=-1;
horitzontala:=-1;
horitzontalb:=-1;
verticala:=-1;
verticalb:=-1;
puntuacioa:=0;
puntuaciob:=0;
llancamentsa:=0;
llancamentsb:=0;
load:=0;
monflag:=1;
PAUSA;
valor 2:
    mentre (fi=0)
        filaa:=-1;
        columnaa:=-1;
        VISUALITZASIMPLE(matriujocab,
caselles);

        escriure("\nTorn del jugador 1:");
        mentre ((filaa<0) o (filaa>=caselles))
            escriure("\nIntrodueix la fila per
a disparar: ");

            filaallegirASCII();
            filaa:=filaa+48;
            si ((filaa<0) o (filaa>=caselles))

```

```

                                escriure("\nFila incorrecta!
tu creus que en aqueta fila pots llancar una bomba?");
                                fsi
                                fmentre
                                mentre ((columnaa<0) o
(columnnaa>=caselles))
                                escriure("\nIntrodueix la columna
per a disparar: ");
                                columnaallegirASCII();
                                si (columnnaa>=97)
                                    columna:=columnnaa-32;
                                fsi
                                columna:=columnnaa-65;
                                si ((columnnaa<0) o
(columnnaa>=caselles))
                                    escriure("\nColumna incorrecta
tros d'ase!");
                                fsi
                                fmentre
                                llancamentsa++;
                                si (matriujocab[filaa][columnna]='?')
                                    si
(matriujocbb[filaa][columnna]='.')
matriujocbb[filaa][columnna]:='-';
matriujocab[filaa][columnna]:='.';
                                jugadaa:=1;
                                fsi
                                si
(matriujocbb[filaa][columnna]='X')
matriujocbb[filaa][columnna]:='@';
matriujocab[filaa][columnna]:='@';
                                encertsa++;

```

```

                                jugadaa:=COMPROVA(matriujocbb,
filaa, columnaa,caselles, tira);
                                si (jugadaa=3)
                                    POSA_AIGUA(matriujocab,
filaa, columnaa, caselles);
                                fsi
                                fsi
                                sino
                                    jugadaa:=0;
                                fsi
                                puntuacioa:=puntuacioa+jugadaa-1;
                                escriure("\nResultat:      Jugador      1
casella '':", filaa, columnaa+65);
                                opció(jugadaa)
                                valor 0:
                                    escriure(" Casella repetida. ");
                                valor 1:
                                    escriure(" Aigua. ");
                                valor 2:
                                    escriure(" Tocat. ");
                                valor 3:
                                    escriure(" Tocat i enfonsat. ");
                                fopcio
                                si (encertsa=20)
                                    fi:=1;
                                sino
                                    escriure("Torna a tirar.");
                                fsi
                                llegirASCII(escapadetime);
                                si (escapadetime='\t')
                                    NETEJAR_PANTALLA;
                                    goto menu;
                                fsi
                                NETEJAR_PANTALLA;

```



```

        fmentre
            puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);
            si (encertsa=20)
                escriure("\nHa guanyat el jugador 1,
amb '' punts, felicitats!\n", puntuacioa);
            fsi
            PAUSA;
            si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");}
            EMMAGATZEMA_RECORDS(taula);
            IMPRIMIR_RECORDS(taula);
            fi:=0;
            noacabata:=0;
            encertsa:=0;
            jugadaa:=1;
            orientacioa:=-1;
            horitzontala:=-1;
            verticala:=-1;
            puntuacioa:=0;
            llancamentsa:=0;
            load:=0;
            PAUSA;
valor 3:
            si (monflag=1)
                tira:=MONEDA();
                monflag:=0;
            fsi
mentre (fi=0)
            mentre (((jugadaa=2 o jugadaa=3) o
tira=1) i (fi=0))
                NETEJAR_PANTALLA
                filaa:=-1;
                columnaa:=-1;

```

```

caselles, 1);
VISUALITZA(matriujocaa, matriujocab,
mentre ((filaa<0) o (filaa>=caselles))
    escriure("\nTorn del jugador 1");
    escriure("\nIntrodueix la fila per
a disparar: ");
    llegirASCII(filaa);
    filaa:=filaa-48;
    si ((filaa<0) o (filaa>=caselles))
        escriure("\nHas introduit una
fila incorrecta empanat!!");
    fsi
fmentre
mentre ((columnaa<0) o
(columnaa>=caselles))
    escriure("\nIntrodueix la columna
per a disparar: ");
    llegirASCII(columnaa);
    si (columnaa>=97)
        columnaa:=columnaa-32;
    fsi
    columnaa:=columnaa-65;
    si ((columnaa<0) o
(columnaa>=caselles))
        escriure("\nPero que no veus
que el que acabes de posar esta malament? torna-hi...");
    fsi
fmentre
llancamentsa++;
si (matriujocab[filaa][columnaa]='?')
    si
(matriujocbb[filaa][columnaa]='.')
matriujocbb[filaa][columnaa]:='-';
matriujocab[filaa][columnaa]:='.';
jugadaa:=1;

```

```

            tira:=2;
        fsi
        si
(matriujocbb[filaa][columnaa]='X')

matriujocbb[filaa][columnaa]:='@';

matriujocab[filaa][columnaa]:='@';

        encertsa++;
        jugadaa:=COMPROVA(matriujocbb,
filaa, columnaa,caselles, tira);
        si (jugadaa=3)
            POSA_AIGUA(matriujocab,
filaa, columnaa, caselles);

        fsi
        tira:=1;
    fsi
sino
        jugadaa:=0;
    fsi
    puntuacioa:=puntuacioa+jugadaa-1;
    escriure("\n\nResultat:      Jugador      1
casella '':", filaa, columnaa+65);
    opció(jugadaa)
valor 0:
        escriure(" Casella repetida. ");
        tira:=2;
valor 1:
        escriure(" Aigua. ");
valor 2:
        escriure(" Tocat. ");
valor 3:
        escriure(" Tocat i enfonsat. ");
fopcio
si (encertsa=20)

```

```

        fi:=1;
sino
    si ((jugadaa=2) o (jugadaa=3))
        escriure("Torna a tirar.");
    fsi
fsi
escapadatetime llegirASCII
    si (escapadatetime='\t')
        NETEJAR_PANTALLA
        goto menu;
    fsi
fmentre
    mentre (((jugadab=2) o (jugadab=3) o
(tira=2)) i (fi=0))
        jugadab:=DISPARA(matriujocaa,
matriujocba,      filab,      columnab,      jugadab,      caselles,
orientaciob,      verticalb,      horitzontalb,      noacabatb);
        si (jugadab=3)
            POSA_AIGUA(matriujocba,      filab,
columnab,      caselles);
        fsi
        llancamentsb++;
        NETEJAR_PANTALLA;
        VISUALITZA(matriujocaa,      matriujocab,
caselles, 1);

        escriure("\nTorn del jugador 2:");
        puntuaciob:=puntuaciob+jugadab-1;
        escriure("\n\nResultat:      Jugador      2
casella '':", filab, columnab+65);
        opció(jugadab)
        valor 1:
            escriure(" Aigua.");
            tira:=1;
        valor 2:
            escriure(" Tocat. ");

```

```

        encertsb++;
        si (encertsb<>20)
            escriure(" Torna a tirar");
        fsi
        noacabatb:=1;
        tira:=2;
valor 3:
        escriure(" Tocat i enfonsat. ");
        encertsb++;
        si (encertsb<>20)
            escriure(" Torna a tirar.");
        fsi
        noacabatb:=0;
        tira:=2;

fopcio

        llegirASCII(escapadetime);
        si (escapadetime='\t')
            NETEJAR_PANTALLA
            goto menu;
        fsi
        si (encertsb=20)
            fi:=1;
        fsi
        fmentre
            fmentre
                puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);
                puntuaciob:=CALCULAPUNTUACIO(caselles,
puntuaciob, llancamentsb);
                si (encertsa=20)
                    escriure("\nHa guanyat el jugador 1,
amb '' punts, felicitats!\n", puntuacioa);
                fsi
                si (encertsb=20)

```

```

                                escriure("\nHa guanyat el jugador 2,
amb '' punts, felicitats!\n", puntuaciob);
                                fsi
                                getch();
                                si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");fsi
                                si (0=AFEGIR_RECORD(puntuaciob, taula, 2))
escriure("El teu record no s'ha afegit...\n");fsi
                                EMMAGATZEMA_RECORDS(taula);
                                IMPRIMIR_RECORDS(taula);
                                fi:=0;
                                noacabata:=0;
                                noacabatb:=0;
                                encertsa:=0;
                                encertsb:=0;
                                jugadaa:=1;
                                jugadab:=1;
                                orientacioa:=-1;
                                orientaciob:=-1;
                                horitzontala:=-1;
                                horitzontalb:=-1;
                                verticala:=-1;
                                verticalb:=-1;
                                puntuacioa:=0;
                                puntuaciob:=0;
                                llancamentsa:=0;
                                llancamentsb:=0;
                                load:=0;
                                monflag:=1;
                                opciotauler:=-1;
                                PAUSA;

```

Fopcio

JOC DE PROVES

Funciona perfectament. El programa no permet que s'introdueixin valors erronis i per tant l'entrada de l'usuari es indiferent per als errors.

4.7 Instruccions

He inclòs unes petites instruccions a les quals es pot accedir des del menú (opció 7). No cal joc de proves ja que és una part del menú que només mostra text.

5 Disseny general en pseudocodi

tipus

registre dadesjoc és

```
    opcio: enter;
    caselles: enter;
    char matriujocaa[10][10];
    char matriujocab[10][10];
    char matriujocba[10][10];
    char matriujocbb[10][10];
    noacabata: enter;
    noacabatb: enter;
    puntuacioa: enter;
    puntuaciob: enter;
    llancamentsa: enter;
    llancamentsb: enter;
    tira: enter;
    encertsa: enter;
    encertsb: enter;
    filaa: enter;
    filab: enter;
    columnaa: enter;
    columnab: enter;
    jugadaa: enter;
    jugadab: enter;
    orientacioa: enter;
    orientaciob: enter;
    verticala: enter;
    verticalb: enter;
    horitzontala: enter;
    horitzontalb: enter;
    monflag: enter;
```

fregistre

ftipus


```

tipus
registre records_tipus és
nom[20]: caràcter;
puntuacio: enter;
fregistre
ftipus
acció  VISUALITZASIMPLE(var  matriujocaa[10][10]:  taula  de
caràcter,caselles: enter)es
    i, k: enter;
    k:=0;
    i:=0;
    escriure("Matriu atac\n");
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre
    i:=0;
    escriure("\n");
    mentre (i<caselles)
        escriure("",i);
        mentre(k<caselles)
            escriure("",matriujocaa[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("\n");
        i++;
    fmentre
facció
acció PAUSA és
    escriure("\nPrem qualsevol tecla per continuar...\n");
    Pausa
facció

```

acció INICIA_EL_MEU_TAULELL(var matriu[10][10]: taula de caràcters,var matriuk[10][10]: taula de caràcters, caselles: enter) és

comprova,control, vaixell, comptador, f, c, direccio, superior, dreta, inferior, esquerra, correcte, i, j: enter;

i:=0;

j:=0;

control:=0;

correcte:=0;

vaixell:=3;

direccio:=-1;

c:=-1;

f:=-1;

comptador:=0;

comprova:=0;

mentre (i<10) \$Inicialitza la matriu amb punts

mentre (j<10)

matriu[i][j]:='.';

j++;

fmentre

j:=0;

i++;

fmentre

i:=0;

mentre (i<10) \$Inicialitza la matriu amb ?

mentre (j<10)

matriuk[i][j]:='?';

j++;

fmentre

j:=0;

i++;

fmentre

i:=0;

mentre (vaixell<>-1)

```

i:=0;
j:=0;
mentre (correcte=0)
    si ((correcte=0) i (control<>0))
        escriure("\nEl vaixell a les coordenades que
has introduit no s'ha pogut colocar...");
        PAUSA();
    fsi
    NETEJAR_PANTALLA
    VISUALITZASIMPLE(matriu, caselles);
    escriure("\nEstem col·locant un vaixell de ' '
caselles", vaixell+1);
    mentre ((f<0) o (f>=caselles))
        si (comprova>0)
            escriure("\nFila incorrecta gamarus!!!");
        fsi
        escriure("\nEn quina fila vols col·locar el
vaixell?: ");
        llegirASCII(f);
        f:=f-48;
        comprova++;
    fmentre
    comprova:=0;
    mentre ((c<0) o (c>=caselles))
        si (comprova>0)
            escriure("\nColumna incorrecta cap de
pardals!!!");
        fsi
        escriure("\nEn quina columna vols col·locar el
vaixell?: ");
        llegirASCII(c);
        si (c>=97)
            c:=c-32;
        fsi
        c:=c-65;

```

```

        comprova++;
fmentre
comprova:=0;
si (vaixell<>0)
    mentre ((direccio<>1) i (direccio<>0))
        si (comprova>0)
            escriure("\nDireccio          incorrecta
borinot!!!");
        fsi
        escriure("\nEn quina direccio vols colocar
el vaixell?: (H/V)");
        llegirASCII(direccio);
        si (direccio>=97) direccio:=direccio-
32;fsi
        si (direccio=72) direccio:=0;fsi
        si (direccio=86) direccio:=1;fsi
        comprova++;
fmentre
sino
    direccio:=1; //valor arbitrari ja que en
vaixells d'una sola casella no te importancia
    fsi
    correcte:=1;
    comprova:=0;
    si (((f+vaixell)>=caselles) i (direccio=1)) o
(((c+vaixell)>=caselles) i (direccio=0))
        correcte:=0;
    fsi
    si (direccio=0)
        superior:=f-1;si (superior=-1)superior++;fsi
        inferior:=f+1; si (inferior=caselles)inferior-
-;fsi
        esquerra:=c-1; si (esquerra=-1)esquerra++;fsi
        dreta:=c+vaixell+1; si (dreta=caselles)dreta--
;fsi
sino

```

```

        superior:=f-1; si (superior=-1) superior++; fsi
        inferior:=f+vaixell+1; si
(inferior=caselles) inferior--; fsi
        esquerra:=c-1; si (esquerra=-1) esquerra++; fsi
        dreta:=c+1; si (dreta=caselles) dreta--; fsi
    fsi
    mentre (esquerra<=dreta)
        mentre (superior<=inferior)
            si (matriu[superior][esquerra]<>'.')
correcte:=0; fsi
                superior++;
                i++;
            fmentre
                superior:=superior-i;
                i:=0;
                esquerra++;
            fmentre
                control++;
            si (correcte=0)
                direccio:=-1;
                c:=-1;
                f:=-1;
            fsi
        fmentre
            si (direccio=0)
                mentre (j<=vaixell)
                    matriu[f][c+j]:='X';
                    j++;
                fmentre
            sino
                mentre (j<=vaixell)
                    matriu[f+j][c]:='X';
                    j++;
                fmentre

```

```

        fsi
        comptador++;
        si (comptador=(4-vaixell))
            comptador:=0;
            vaixell--;
        fsi
        control:=0;
        correcte:=0;
        direccio:=-1;
        c:=-1;
        f:=-1;
    fmentre
        escriure("\nJa has colocat tots els vaixells!! si fa
patxoca... pero segur que perds :P");
    facció

funció EMMAGATZEMA_JOC(var dades: tipus dadesjoc) retorna
booleà és
    f: fitxer;
    n: enter;
    n:=0;
    f:=obrir("partida.bin","w");
    si (f<>NULL)
        n:=escriuref(f, dades);
    fsi
    tancar(f);
retorna(n);
ffunció
funció RECUPERA_JOC(var dades: tipus dadesjoc) retorna booleà
és
    f: fitxer;
    n: enter;
    n:=0;
    f:=obrir("partida.bin","r");

```

```

    si (f<>NULL)
        nllegirASCIIIf(f, dades);
        tancar(f);
    fsi
    retorna(n);
ffunció
funció CALCULAPUNTUACIO(caselles: enter, puntuacio: enter,
llancaments: enter) retorna enter és
    casellesf, puntuaciof, llancamentsf: real
    casellesf:=caselles;
    llancamentsf:=llancaments;
    puntuaciof:=puntuacio100(casellesf/llancamentsf);
    puntuacio:=puntuaciof;
    retorna(puntuacio);
ffunció
funció EMMAGATZEMA_RECORDS(var records[10]: taula de
records_tipus) retorna enter
    n, i, j: enter
    f: fitxer;
    f:=obrir("records.txt","w");
    i:=0;
    j:=0;
    si (f<>NULL)
        n:=0;
        mentre (i<>10)
            mentre (j<>20)
                escriuref(f,records[i].nom[j]);
                j++;
            fmentre
                fescriure(f, '\t');
                fescriure(f, records[i].puntuacio);
                fescriure(f, '\n');
            j:=0;
            i++;

```

```

        fmentre
sino
    n:=-1;
fsi
fclose(f);
retorna(n);
ffunció
funció RECUPERA_RECORDS(var records[10]: taula de
records_tipus) retorna enter és
    i,k,n: enter;
    lectura: caràcter;
    i:=0;
    k:=0;
    f: fitxer;
    f:=obrir("records.txt","r");
    si (f=NULL)
        n:=-1;
    sino
        mentre (nofinaldefitxer(f))
            mentre (k<>21)
                llegirf(f, "%c", lectura);
                si (lectura<>'\\t')
records[i].nom[k]:=lectura;fsi
                k++;
            fmentre
                llegirf(f, "%i", records[i].puntuacio);
                llegirf(f, "%c", lectura);
                k:=0;
                i++;
            fmentre
                tancar(f);
        fsi
    retorna(n);
ffunció

```



```

acció      INICIALITZA_TAULA(var      records[10]:      taula      de
records_tipus) és
i,k: enter;
k:=0;
mentre (k<=9)
    records[k].puntuacio:=-666;
    k++;
fmentre
k:=0;
i:=0;
mentre (i<=9)
    mentre (k<20)
        records[i].nom[k]:=' ';
        k++;
    fmentre
k:=0;
i++;
fmentre
facció
acció      IMPRIMIR_RECORDS(var      records[10]:      taula      de
records_tipus) és
flag,i, k: enter;
i:=0;
k:=0;
flag:=0;
NETEJAR_PANTALLA
mentre (i<=9)
    si (records[i].puntuacio<>-666)
        flag:=1;
    fsi
    i++;
fmentre
si (flag=1)
    i:=0;

```

```

        escriure ("Nom                Puntuacio\n");
mentre (i<=9)
    si (records[i].puntuacio<>-666)
        flag:=1;
        mentre (k<>20)
            escriure ("%c",records[i].nom[k]);
            k++;
        fmentre
        escriure ("%i", records[i].puntuacio);
        escriure ("\n");
        k:=0;
    fsi
    i++;
fmentre
sino
    escriure("El fitxer de records esta buit o no existeix,
prova a afegir un record jugant.");
fsi
facció
acció POSA_AIGUA(var atac[10][10]: taula de caràcters,f:
enter,c: enter, caselles: enter) és
    up, down, right, left, i: enter;
    i:=0;
    si (f-1>=0)
        si (atac[f-1][c]='@')
            si (f-2>=0)
                si (atac[f-2][c]='@')
                    si (f-3>=0)
                        si (atac[f-3][c]='@')
                            si (f-4>=0)
                                up:=f-4;
                            sino
                                up:=f-3;
                        fsi
                    
```

```

                                sino
                                up:=f-3;
fsi
                                sino
                                up:=f-2;
                                fsi
                                sino
                                up:=f-2;
                                fsi
                                sino
                                up:=f-1;
                                fsi
sino
                                up:=f-1;
fsi
sino
                                up:=f;
fsi
si (c-1>=0)
    si (atac[f][c-1]='@')
        si (c-2>=0)
            si (atac[f][c-2]='@')
                si (c-3>=0)
                    si (atac[f][c-3]='@')
                        si (c-4>=0)
                            left:=c-4;
                        sino
                            left:=c-3;
                    fsi
                sino
                    left:=c-3;
                fsi
            sino
                left:=c-2;
        82

```

```

        fsi
    sino
        left:=c-2;
    fsi
    sino
        left:=c-1;
    fsi
    sino
        left:=c-1;
    fsi
sino
    left:=c;
fsi
si (c+1<caselles)
    si (atac[f][c+1]='@')
        si (c+2<caselles)
            si (atac[f][c+2]='@')
                si (c+3<caselles)
                    si (atac[f][c+3]='@')
                        si (c+4<caselles)
                            right:=c+4;
                        sino
                            right:=c+3;
                    fsi
                sino
                    right:=c+3;
                fsi
            sino
                right:=c+2;
            fsi
        sino
            right:=c+2;
        fsi
    sino

```

```

        right:=c+1;
    fsi
sino
    right:=c+1;
fsi
sino
    right:=c;
fsi
si (f+1<caselles)
    si (atac[f+1][c]='@')
        si (f+2<caselles)
            si (atac[f+2][c]='@')
                si (f+3<caselles)
                    si (atac[f+3][c]='@')
                        si (f+4<caselles)
                            down:=f+4;
                        fsi
                    sino
                        down:=f+3;
                    fsi
                sino
                    down:=f+3;
                fsi
            sino
                down:=f+2;
            fsi
        sino
            down:=f+2;
        fsi
    sino
        down:=f+1;
    fsi
sino
    down:=f+1;

```

```

        fsi
    sino
        down:=f;
    fsi
    mentre (left<=right)
        mentre (up<=down)
            si (atac[up][left]<>'@')
atac[up][left]:='.';fsi
            up++;
            i++;
        fmentre
        up:=up-i;
        i:=0;
        left++;
    fmentre
acció
funció AFEGIR_RECORD(puntuacio: enter,var records[10]: taula
de records_tipus, jugador: enter)retorna booleà és
record:records_tipus;
opciorecord, lectura: caràcter;
i,j,k: enter;
flag: booleà;
flag:=0;
i:=0;
j:=0;
opciorecord:='k';
lectura:='k';
mentre (i<>10)
    si (records[i].puntuacio<=puntuacio) flag:=1;fsi
    i++;
fmentre
si (flag=1)
    flag:=0;

```

```

    mentre ((opciorecord<>'s') i (opciorecord<>'S') i
(opciorecord<>'N') i (opciorecord<>'n'))
        NETEJAR_PANTALLA
        escriure("Jugador '", vols actualitzar els record? S/N
", jugador);
        llegirASCII(opciorecord);
    fmentre
    si ((opciorecord='s') o (opciorecord='S'))
        i:=0;
        escriure("\nIntrodueix el nom del jugador ' ' (20
caracters com a maxím) i apreta intro: ", jugador);
        mentre ((lectura<>'r') i (i<20))
            llegirASCII(lectura);
            si (lectura<>'r') record.nom[i]:=lectura;fsi
            i++;
        fmentre
        i--;
        mentre (i<>20)
            record.nom[i]:=' ';
            i++;
        fmentre
        NETEJAR_PANTALLA
        record.puntuacio:=puntuacio;
        i:=9;
        k:=8;
        mentre (i>=0)
            si (((records[i-1].puntuacio>record.puntuacio) i
(records[i].puntuacio<=record.puntuacio)) o
((records[i].puntuacio<=record.puntuacio) i (i=0)))
                flag:=1;
                mentre (k<>(i-1))

records[k+1].puntuacio:=records[k].puntuacio;
                mentre (20<>j)

records[k+1].nom[j]:=records[k].nom[j];

```

```

        j++;
    fmentre
    k--;
    j:=0;
    fmentre
    records[i].puntuacio:=record.puntuacio;
    k:=0;
    mentre (k<>20)
        records[i].nom[k]:=record.nom[k];
        k++;
    fmentre
    fsi
    i--;
fmentre
    fsi
fsi
retorna (flag);
ffunció

funció MONEDA()
    retorna enter és
opcio: caràcter;
flag, opciok, guanya: enter;
flag:=0;

mentre (flag=0)

    escriure("Que vols ser? Cara o creu?  C/X ");
    llegirASCII(
opcio);

    si((opcio='C')  o (opcio='X')  o (opcio='x')  o (opcio='c'))

```



```

si (opcio='c') opcio:='C';fsi

si(opcio='x') opcio:='X';fsi

flag:=true;

    sino

        NETEJARPANTALLA;
    fsi
fmentre

escriure("\nApreta qualsevol tecla per a tirar la moneda: ");

PAUSA;

opciok:=ALEATORI%2;

si (opciok=1)

    escriure("C \n");

    si (opcio='C')

        escriure("\nComenca el jugador 1\n");

        guanya:=1;

    sino

```

```
escriure("\nComenca el jugador 2\n");
```

```
    guanya:=0;
```

```
    fsi
```

```
sino
```

```
    escriure("X\n");
```

```
    si (opcio='X')
```

```
escriure("\nComenca el jugador 1\n");
```

```
    guanya:=1;
```

```
sino
```

```
escriure("\nComenca el jugador 2\n");
```

```
guanya:=0;
```

```
    fsi
```

```
fsi
```

```
si (guanya=0) guanya:=2;}
```

```
PAUSA;
```

```
NETEJAR_PANTALLA;
```

```
retorna(guanya);
```

ffunció

funció COMPROVA(var defensa[10][10]: taula de caràcters,
f:enter, c:enter,caselles:enter, var tirada:enter)retorna
enter és

jugada, up, down, left, right: enter;

up:=0;

down:=0;

left:=0;

right:=0;

si (f-1>=0)

si (defensa[f-1][c]='X')

jugada:=2;

fsi

si ((defensa[f-1][c]='.') o (defensa[f-1][c]='-'))

up:=1;

fsi

si (defensa[f-1][c]='@')

si (f-2>=0)

si (defensa[f-2][c]='X')

jugada:=2;

fsi

si ((defensa[f-2][c]='.') o (defensa[f-
2][c]='-'))

up:=1;

fsi

si (defensa[f-2][c]='@')

si (f-3>=0)

si (defensa[f-3][c]='X')

jugada:=2;

fsi

si ((defensa[f-3][c]='.') o
(defensa[f-3][c]='-'))

up:=1;

fsi

```

                si (defensa[f-3][c]='@') up:=1;}
            sino
                up:=1;
            fsi
        fsi
    sino
        up:=1;
    fsi
si (f+1<caselles)
    si (defensa[f+1][c]='X')
        jugada:=2;
    fsi
    si ((defensa[f+1][c]='.') o (defensa[f+1][c]='-'))
        down:=1;
    fsi
    si (defensa[f+1][c]='@')
        si (f+2<caselles)
            si (defensa[f+2][c]='X')
                jugada:=2;
            fsi
            si ((defensa[f+2][c]='.') o
(defensa[f+2][c]='-'))
                down:=1;
            fsi
            si (defensa[f+2][c]='@')
                si (f+3<caselles)
                    si (defensa[f+3][c]='X')
                        jugada:=2;
                    fsi
                    si ((defensa[f+3][c]='.') o
(defensa[f+3][c]='-'))

```

```

                                down:=1;
                                fsi
                                si (defensa[f+3][c]='@') down:=1;}
                                sino
                                down:=1;
                                fsi
                                fsi
                                sino
                                down:=1;
                                fsi
                                fsi
                                sino
                                down:=1;
                                fsi
                                si (c-1>=0)
                                    si (defensa[f][c-1]='X')
                                        jugada:=2;
                                    fsi
                                    si ((defensa[f][c-1]='.') o (defensa[f][c-1]='-'))
                                        left:=1;
                                    fsi
                                    si (defensa[f][c-1]='@')
                                        si (c-2>=0)
                                            si (defensa[f][c-2]='X')
                                                jugada:=2;
                                            fsi
                                            si ((defensa[f][c-2]='.') o (defensa[f][c-
2]='-'))
                                                left:=1;
                                            fsi
                                            si (defensa[f][c-2]='@')
                                                si (c-3>=0)
                                                    si (defensa[f][c-3]='X')
                                                        jugada:=2;

```

```

                                fsi
                                si      ((defensa[f][c-3]='.')      o
(defensa[f][c-3]='-'))
                                left:=1;
                                fsi
                                si (defensa[f][c-3]='@') left:=1;}
                                sino
                                left:=1;
                                fsi
                                fsi
                                sino
                                left:=1;
                                fsi
                                fsi
                                sino
                                left:=1;
                                fsi
                                si (c+1<caselles)
                                    si (defensa[f][c+1]='X')
                                        jugada:=2;
                                    fsi
                                    si ((defensa[f][c+1]='.') o (defensa[f][c+1]='-'))
                                        right:=1;
                                    fsi
                                    si (defensa[f][c+1]='@')
                                        si (c+2<caselles)
                                            si (defensa[f][c+2]='X')
                                                jugada:=2;
                                            fsi
                                            si      ((defensa[f][c+2]='.')      o
(defensa[f][c+2]='-'))
                                                right:=1;
                                            fsi
                                            si (defensa[f][c+2]='@')

```

```

        si (c+3<caselles)
            si (defensa[f][c+3]='X')
                jugada:=2;
            fsi
            si      ((defensa[f][c+3]='.')      o
(defensa[f][c+3]='-'))
                right:=1;
            fsi
            si      (defensa[f][c+3]='@')
right:=1;}

        sino
            right:=1;
        fsi
    fsi
    sino
        right:=1;
    fsi
    fsi
    sino
        right:=1;
    fsi
    si ((up=1) i (down=1) i (left=1) i (right=1))
jugada:=3;fsi
    tirada:=1;
    retorna(jugada);
ffunció

```

```

acció RANDOMIZE(var fila: enter, var columna: enter, var
direccio: enter, caselles: enter) és
    fila:=ALEATORI%(caselles);
    columna:=ALEATORI%(caselles);
    direccio:=ALEATORI%2; $0 serà vertical i 1 serà
horitzontal
facció

```

```

funció TIPUS_JOC(var caselles: enter) retorna enter és
    opcio: enter;
    opcio:=0;
    caselles:=0;
    mentre ((caselles>10) o (caselles<8))
        escriure("De quantes caselles vols que sigui
el tauler? min 8 max 10: ");
        llegirASCII(caselles);
        caselles:=caselles-48;
        si (caselles=1)

    mentre (caselles<>0)

        llegirASCII(caselles);
        caselles:=caselles-48;

    fmentre
        escriure("0");
        caselles:=10;

    fsi
NETEJAR_PANTALLA
fmentre
mentre ((opcio<1) o (opcio>3))
    NETEJAR_PANTALLA
        escriure("A quina modalitat de joc vols jugar?\n");
        escriure("1.-Cap jugador\n");
        escriure("2.-Un jugador\n");
        escriure("3.-Dos jugadors\n");
        llegirASCII(opcio);
        opcio:=opcio-48;
    NETEJAR_PANTALLA

```



```

    fmentre
retorna(opcio);
ffunció

acció    INICIA_TAULELL(var    matriu[10][10]:    matriu    de
caràcters,var matriuk[10][10]: matriu de caràcters, caselles:
enter) és

    control, vaixell, comptador, f, c, direccio, superior,
dreta, inferior, esquerra, correcte, i, j: enter;
repeteix:
    i:=0;
    j:=0;
    control:=0;
    correcte:=0;
    vaixell:=3;
    comptador:=0;
    mentre (i<10) $Inicialitza la matriu amb punts
        mentre (j<10)
            matriu[i][j]:='.';
            j++;
        fmentre
        j:=0;
        i++;
    fmentre
    i:=0;
    mentre (i<10) $Inicialitza la matriu amb ?
        mentre (j<10)
            matriuk[i][j]:='?';
            j++;
        fmentre
        j:=0;
        i++;
    fmentre
    i:=0;

```

```

mentre (vaixell<>-1)
    i:=0;
    j:=0;
    mentre (correcte=0)
        correcte:=1;
        RANDOMIZE(f,c, direccio, caselles);
        si (((f+vaixell)>=caselles) i (direccio=1)) o
        (((c+vaixell)>=caselles) i (direccio=0))
            correcte:=0;
        fsi
        si (direccio=0)
            superior:=f-1; si (superior=-1)superior++;}
            inferior:=f+1; si (inferior=caselles)inferior-
-; }

            esquerra:=c-1; si (esquerra=-1)esquerra++;}
            dreta:=c+vaixell+1; si (dreta=caselles)dreta--
; }

        sino
            superior:=f-1; si (superior=-1)superior++;}
            inferior:=f+vaixell+1; si
(inferior=caselles)inferior--; }
            esquerra:=c-1; si (esquerra=-1)esquerra++;}
            dreta:=c+1; si (dreta=caselles)dreta--; }
        fsi
        mentre (esquerra<=dreta)
            mentre (superior<=inferior)
                si (matriu[superior][esquerra]<>'.' )
correcte:=0; fsi
                    superior++;
                    i++;
                fmentre
                    superior:=superior-i;
                    i:=0;
                    esquerra++;
                fmentre

```

```

        control++;
        si (control>=2000)
            goto repeteix;
        fsi
fmentre
correcte:=0;
si (direccio=0)
    mentre (j<=vaixell)
        matriu[f][c+j]:='X';
        j++;
    fmentre
sino
    mentre(j<=vaixell)
        matriu[f+j][c]:='X';
        j++;
    fmentre
fsi
comptador++;
si (comptador=(4-vaixell))
    comptador:=0;
    vaixell--;
fsi
fmentre
facció

```

```

funció MOSTRAR_MENU(load: enter) retorna enter és
    menu: enter;
    menu:=0;
    o mentre ((menu>7) o (menu<1) o (menu=3 i load=0)
    o (menu=4 i load=0) o (menu=0))
        escriure("\nMenu:\n");
        escriure("1.- Crear un nou joc\n");
        escriure("2.- Carregar un joc emmagatzemat\n");
        si (load=1)

```

```

        escriure("3.- Jugar partida\n");
        escriure("4.- Emmagatzemar el joc\n");
    fsi
    escriure("5.- Veure podium\n");
    escriure("6.- Sortir del joc\n");
    escriure("7.- Instruccions\n\n");
    escriure("Selecciona la opció i prem intro\n\n");
    menullegirASCII();
    menu:=menu-48;
    NETEJAR_PANTALLA
fmentre
    retorna(menu);
ffunció

acció    VISUALITZA(var    matriudefensa[10][10]:    taula    de
caràcters,var matriuatac[10][10]: taula de caràcters,caselles:
enter, jugador: enter) és
    i, k: enter;
    k:=0;
    i:=0;
    escriure("Jugador          '\nMatriu          atac
",jugador);
    si (caselles=9)
        escriure(" ");
    fsi
    si (caselles=10)
        escriure(" ");
    fsi
    escriure("Matriu defensa\n");
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre

```

```

    escriure("                ");
    i:=0;
    escriureASCII(254);
    mentre (i<caselles)
        escriureASCII(i+65);
        i++;
    fmentre
    i:=0;
    escriure("\n");
    mentre (i<caselles)
        escriure("%i",i);
        mentre(k<caselles)
            escriure("%c",matriuatac[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("                ");
        escriure("%i",i);
        mentre(k<caselles)
            escriure("%c",matriudefensa[i][k]);
            k++;
        fmentre
        k:=0;
        escriure("\n");
        i++;
    fmentre
facció

```

```

funció DISPARA(var defensa[10][10]: taula de caràcters, var
atac[10][10]: taula de caràcters, f: enter, c: enter,
jugada:enter, caselles:enter, orientacio:enter,
vertical:enter, horitzontal:enter, noacabat:enter) retorna
enter és
tirada, fila, columna, correcte:enter;
correcte:=0;

```

```

tirada:=0;
mentre (tirada=0)
si ((jugada=1) i (noacabat=0))
    mentre (correcte=0)
        f:=ALEATORI%(caselles);
        c:=ALEATORI%(caselles);
        si (atac[f][c]='?') correcte:=1;}
fmentre
si (defensa[f][c]='X')
    defensa[f][c]:='@';
    atac[f][c]:='@';
    fila:=f;
    columna:=c;
    jugada:=COMPROVA(defensa, fila, columna, caselles,
tirada);
    fsi
    si (defensa[f][c]='.')
        defensa[f][c]:='-';
        atac[f][c]:='.';
        jugada:=1;
    fsi
    tirada:=1;
fsi
si (((jugada=2) o (noacabat=1)) i (tirada=0))
    si (orientacio=-1)
        orientacio:=ALEATORI%2;
    fsi
    si (orientacio=0)
        si (horitzontal=-1)
            horitzontal:=ALEATORI%2;
        fsi
        si (horitzontal=0)
            si (c-1>=0)
                si (atac[f][c-1]='?')

```

```

        si ((defensa[f][c-1]='X') i (tirada=0))
            defensa[f][c-1]:='@';
            atac[f][c-1]:='@';
            c:=c-1;
            fila:=f;
            columna:=c;
            jugada:=COMPROVA(defensa,          fila,
columna, caselles, tirada);
            orientacio:=0;
            si (jugada=3)
                orientacio:=-1;
                horitzontal:=-1;
            fsi
            tirada:=1;
        fsi
        si ((defensa[f][c-1]='.') i (tirada=0))
            defensa[f][c-1]:='-';
            atac[f][c-1]:='.';
            horitzontal:=1;
            jugada:=1;
            tirada:=1;
            si (c+1<caselles)
                si          ((atac[f][c-1]='.')          i
(atac[f][c+1]='.'))
                    orientacio:=1;
                    horitzontal:=-1;
                fsi
            sino
                orientacio:=1;
                horitzontal:=-1;
            fsi
        fsi
        si ((atac[f][c-1]='@') i (tirada=0))

```

```

        c:=c-1;
        orientacio:=0;
    fsi
    si ((atac[f][c-1]='.') i (tirada=0))
        horitzontal:=1;
        si ((atac[f][c-1]='.') i
(atac[f][c+1]='.'))
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino
        horitzontal:=1;
        si (atac[f][c+1]='.')
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino $horitzontal:=1
    si (c+1<caselles)
        si (atac[f][c+1]='?')
            si ((defensa[f][c+1]='X') i (tirada=0))
                defensa[f][c+1]:='@';
                atac[f][c+1]:='@';
                c:=c+1;
                fila:=f;
                columna:=c;
                jugada:=COMPROVA(defensa, fila,
columna, caselles, tirada);
                orientacio:=0;
                tirada:=1;
                si (jugada=3)
                    orientacio:=-1;
                    horitzontal:=-1;

```



```

        fsi
    fsi
    si ((defensa[f][c+1]='.') i (tirada=0))
        defensa[f][c+1]:='-';
        atac[f][c+1]:='.';
        horitzontal:=0;
        jugada:=1;
        tirada:=1;
        si (c-1>=0)
            si ((atac[f][c-1]='.') i
(atac[f][c+1]='.'))
                orientacio:=1;
                horitzontal:=-1;
            fsi
        sino
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    fsi
    si ((atac[f][c+1]='@') i (tirada=0))
        c:=c+1;
        orientacio:=0;
    fsi
    si ((atac[f][c+1]='.') i (tirada=0))
        horitzontal:=0;
        si ((atac[f][c+1]='.') i (atac[f][c-
1]='.'))
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
    sino
        horitzontal:=0;

```

```

        si (atac[f][c-1]='.')
            orientacio:=1;
            horitzontal:=-1;
        fsi
    fsi
sino $orientacio:=1 vertical per tant
    si (vertical=-1)
        vertical:=ALEATORI%2;
    fsi
    si (vertical=0) //apuntant cap a dalt
        si (f-1>=0)
            si (atac[f-1][c]='?')
                si ((defensa[f-1][c]='X') i (tirada=0))
                    defensa[f-1][c]:='@';
                    atac[f-1][c]:='@';
                    f:=f-1;
                    fila:=f;
                    columna:=c;
                    jugada:=COMPROVA(defensa, fila,
columna, caselles, tirada);
                    orientacio:=1;
                    tirada:=1;
                si (jugada=3)
                    orientacio:=-1;
                    vertical:=-1;
            fsi
        fsi
    si ((defensa[f-1][c]='.') i (tirada=0))
        defensa[f-1][c]:='-';
        atac[f-1][c]:='.';
        vertical:=1;
        jugada:=1;
        tirada:=1;

```

```

                                si                ((atac[f-1][c]='.') i
(atac[f+1][c]='. '))
                                orientacio:=0;
                                vertical:=-1;
                                fsi
                                fsi
                                fsi
                                si ((atac[f-1][c]='@') i (tirada=0))
                                    f:=f-1;
                                    orientacio:=1;
                                fsi
                                si ((atac[f-1][c]='.') i (tirada=0))
                                    vertical:=1;
                                    si (f+1<caselles)
                                        si                ((atac[f-1][c]='.') i
(atac[f+1][c]='. '))
                                        orientacio:=0;
                                        vertical:=-1;
                                        fsi
                                        sino
                                            vertical:=-1;
                                            orientacio:=0;
                                        fsi
                                        fsi
                                        sino
                                            vertical:=1;
                                            si (atac[f+1][c]='.')
                                                orientacio:=0;
                                                vertical:=-1;
                                            fsi
                                        fsi
                                        sino $vertical:=1
                                        si (f+1<caselles)
                                            si (atac[f+1][c]='?')

```

```

        si ((defensa[f+1][c]='X') i (tirada=0))
            defensa[f+1][c]:='@';
            atac[f+1][c]:='@';
            f:=f+1;
            fila:=f;
            columna:=c;
            jugada:=COMPROVA(defensa, fila,
columna, caselles, tirada);
            orientacio:=1;
            tirada:=1;
            si (jugada=3)
                orientacio:=-1;
                vertical:=-1;
            fsi
        fsi
    si ((defensa[f+1][c]='.') i (tirada=0))
        defensa[f+1][c]:='-';
        atac[f+1][c]:='.';
        vertical:=0;
        jugada:=1;
        tirada:=1;
        si (f-1>=0)
            si ((atac[f-1][c]='.') i
(atac[f+1][c]='.'))
                orientacio:=0;
                vertical:=-1;
            fsi
        sino
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
    si ((atac[f+1][c]='@') i (tirada=0))

```

```

        f:=f+1;
        orientacio:=1;
    fsi
    si ((atac[f+1][c]='.') i (tirada=0))
        vertical:=0;
        si (atac[f-1][c]='.')
            orientacio:=0;
            vertical:=-1;
        fsi
    fsi
sino
    vertical:=0;
    si (atac[f-1][c]='.')
        orientacio:=0;
        vertical:=-1;
    fsi
fsi
fsi
fsi
si ((jugada=3) i (tirada=0))
    jugada:=1;
    noacabat:=0;
    tirada:=0;
fsi
fmentre
retorna (jugada);
ffunció

```

```

int main()
    matriujocaa[10][10]: taula de caràcter;
    matriujocab[10][10]: taula de caràcter;
    matriujocba[10][10]: taula de caràcter;
    matriujocbb[10][10]: taula de caràcter;

```

```

    escapadetime: caràcter;
    taula[10]: records_tipus;
    i,j,monflag,      opciotauler,      noacabata,      noacabatb,
puntuacioa,      puntuaciob,llancamentsa,      llancamentsb,      tira,
encertsa, encertsb, filaa, filab, columnaa, columnab, jugadaa,
fi, jugadab, opcio, caselles, load, menu, orientacioa,
orientaciob, verticala, verticalb, horitzontala, horitzontalb:
enter;

    fi:=0;
    load:=0;
    opciotauler:=-1;
    dadesjoc dades;
    noacabata:=0;
    noacabatb:=0;
    encertsa:=0;
    encertsb:=0;
    jugadaa:=1;
    jugadab:=1;
    orientacioa:=-1;
    orientaciob:=-1;
    horitzontala:=-1;
    horitzontalb:=-1;
    verticala:=-1;
    verticalb:=-1;
    puntuacioa:=0;
    puntuaciob:=0;
    llancamentsa:=0;
    llancamentsb:=0;
    monflag:=1;
    INICIALITZA_Taula(taula);
    RECUPERA_RECORDS(taula);
    mentre (1<>0)
    menu:
    menu:=MOSTRAR_MENU(load);
    opció (menu)

```

```

valor 1:
    opcio:=TIPUS_JOC(caselles);
    opció(opcio)
        valor 1:
            INICIA_TAULELL(matriujocaa, matriujocab,
caselles);
            INICIA_TAULELL(matriujocbb, matriujocba,
caselles);
            load:=1;
        valor 2:
            INICIA_TAULELL(matriujocaa, matriujocab,
caselles);
            INICIA_TAULELL(matriujocbb, matriujocba,
caselles);
            load:=1;

        valor 3:
            mentre ((opciotauler<>78) i
(opciotauler<>83))
                escriure("\nJugador 1: Vols colocar
els teus propis vaixells? (S/N)");
                opciotaulerlllegirASCII();
                si (opciotauler>=97)
                    opciotauler:=opciotauler-32;
                fsi
            fmentre
            si (opciotauler=78)
                INICIA_TAULELL(matriujocaa,
matriujocab, caselles);
                INICIA_TAULELL(matriujocbb,
matriujocba, caselles);
            sino
                INICIA_EL_MEU_TAULELL(matriujocaa,
matriujocab, caselles);
                INICIA_TAULELL(matriujocbb,
matriujocba, caselles);
            fsi

```

```

        load:=1;
        fopcio
valor 2:
        si (0=RECUPERA_JOC( dades))
            escriure("No s'ha carregat cap joc...");
        sino
            escriure("El teu joc s'ha carregat
correctament");
        monflag:=0;
        opcio:=dades.opcio;
        caselles:=dades.caselles;
        i:=0;
        j:=0;
        mentre (i<10)
            mentre (j<10)
                matriujocaa[i][j]:=dades.matriujocaa[i][j];
                j++;
            fmentre
            j:=0;
            i++;
        fmentre
        i:=0;
        mentre (i<10)
            mentre (j<10)
                matriujocab[i][j]:=dades.matriujocab[i][j];
                j++;
            fmentre
            j:=0;
            i++;
        fmentre
        i:=0;
        mentre (i<10)
            mentre (j<10)
                matriujocbb[i][j]:=dades.matriujocbb[i][j];

```



```

        j++;
    fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (i<10)
    mentre (j<10)
        matriujocba[i][j]:=dades.matriujocba[i][j];
        j++;
    fmentre
    j:=0;
    i++;
fmentre
noacabata:=dades.noacabata;
noacabatb:=dades.noacabatb;
puntuacioa:=dades.puntuacioa;
puntuaciob:=dades.puntuaciob;
llancamentsa:=dades.llancamentsa;
llancamentsb:=dades.llancamentsb;
tira:=dades.tira;
encertsa:=dades.encertsa;
encertsb:=dades.encertsb;
filaa:=dades.filaa;
filab:=dades.filab;
columnaa:=dades.columnaa;
columnab:=dades.columnab;
jugadaa:=dades.jugadaa;
jugadab:=dades.jugadab;
orientacioa:=dades.orientacioa;
orientaciob:=dades.orientaciob;
verticala:=dades.verticala;
verticalb:=dades.verticalb;
horitzontala:=dades.horitzontala;

```

```

horitzontalb:=dades.horitzontalb;
monflag:=dades.monflag;
    load:=true;
    menu:=0;
escriure("\nLa partida es de %i jugadors", opcio-1);
escriure("\nI el tauler es de %i caselles", caselles);
fsi
valor 3:
    opció(opcio)
    valor 1:
        si (monflag=1)
            tira:=MONEDA();
            monflag:=0;
        fsi
    mentre (fi=0)
        mentre (((jugadaa=2 o jugadaa=3) o
tira=1) i (fi=0))
            jugadaa:=DISPARA(matriujocbb,
matriujocab, filaa, columnaa, jugadaa, caselles, orientacioa,
verticala, horitzontala, noacabata);
            si (jugadaa=3)
                POSA_AIGUA(matriujocab, filaa,
columnaa, caselles);
            fsi
            puntuacioa:=puntuacioa+jugadaa-1;
            llancamentsa++;
            NETEJAR_PANTALLA
            VISUALITZA(matriujocaa,
matriujocab, caselles, 1);
            VISUALITZA(matriujocbb,
matriujocba, caselles, 2);
            escriure("\nTorn del jugador 1:");
            escriure("\n\nResultat: Jugador 1
casella '':", filaa, columnaa+65);
            opció(jugadaa)
            valor 1:

```

```

        escriure(" Aigua.");
        tira:=2;

valor 2:
    escriure(" Tocat.");
    encertsa++;
    si (encertsa<>20)
        escriure("      Torna      a
tirar.");

    fsi
    noacabata:=1;
    tira:=1;
valor 3:
    escriure(" Tocat i enfonsat.
");
    encertsa++;
    si (encertsa<>20)
        escriure("Torna      a
tirar.");

    fsi
    noacabata:=0;
    tira:=1;

fopcio

llegeixASCII(escapadetime);
si (escapadetime='\t')
    NETEJAR_PANTALLA
    goto menu;
fsi
si (encertsa=20)
    fi:=1;
fsi
fmentre
mentre (((jugadab=2) o (jugadab=3) o
(tira=2)) i (fi=0))

```

```

                                jugadab:=DISPARA(matriujocaa,
matriujocba, filab, columnab, jugadab, caselles, orientaciob,
verticalb, horitzontalb, noacabatb);
                                si (jugadab=3)
                                    POSA_AIGUA(matriujocba,
filab, columnab, caselles);
                                fsi
                                llancamentsb++;
                                NETEJAR_PANTALLA
                                VISUALITZA(matriujocaa,
matriujocab, caselles, 1);
                                VISUALITZA(matriujocbb,
matriujocba, caselles, 2);
                                puntuaciob:=puntuaciob+jugadab-1;
                                escriure("\nTorn del jugador 2:");
                                escriure("\n\nResultat: Jugador 2
casella '':", filab, columnab+65);
                                opció(jugadab)
                                valor 1:
                                    escriure(" Aigua.");
                                    tira:=1;
                                valor 2:
                                    escriure(" Tocat. ");
                                    encertsb++;
                                    si (encertsb<>20)
                                        escriure("Torna a
tirar.");
                                fsi
                                noacabatb:=1;
                                tira:=2;
                                valor 3:
                                    escriure(" Tocat i enfonsat.
");
                                    encertsb++;
                                    si (encertsb<>20)

```

```

tirar.");
                                escriure("Torna a
                                fsi
                                noacabatb:=0;
                                tira:=2;
                                fopcio
                                llegeixASCII(escapadetime);
                                si (escapadetime='\t')
                                    NETEJAR_PANTALLA
                                    goto menu;
                                fsi
                                si (encertsb=20)
                                    fi:=1;
                                fsi
                                fmentre
                                fmentre
                                    puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);
                                    puntuaciob:=CALCULAPUNTUACIO(caselles,
puntuaciob, llancamentsb);
                                    si (encertsa=20)
                                        escriure("\nHa guanyat el jugador 1,
amb %i punts, felicitats!\n", puntuacioa);
                                        fsi
                                        si (encertsb=20)
                                            escriure("\nHa guanyat el jugador 2,
amb %i punts, felicitats!\n", puntuaciob);
                                            fsi
                                            PAUSA;
                                            si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");fsi
                                            si (0=AFEGIR_RECORD(puntuaciob, taula, 2))
escriure("El teu record no s'ha afegit...\n");fsi
                                            EMMAGATZEMA_RECORDS(taula);
                                            IMPRIMIR_RECORDS(taula);
                                            fi:=0;

```

```

noacabata:=0;
noacabatb:=0;
encertsa:=0;
encertsb:=0;
jugadaa:=1;
jugadab:=1;
orientacioa:=-1;
orientaciob:=-1;
horitzontala:=-1;
horitzontalb:=-1;
verticala:=-1;
verticalb:=-1;
puntuacioa:=0;
puntuaciob:=0;
llancamentsa:=0;
llancamentsb:=0;
load:=0;
monflag:=1;
PAUSA;
valor 2:
    mentre (fi=0)
        filaa:=-1;
        columnaa:=-1;
        VISUALITZASIMPLE(matriujocab,
caselles);

        escriure("\nTorn del jugador 1:");
        mentre ((filaa<0) o (filaa>=caselles))
            escriure("\nIntrodueix la fila per
a disparar: ");

            filaallegirASCII();
            filaa:=filaa-48;
            si ((filaa<0) o (filaa>=caselles))
                escriure("\nFila incorrecta!
tu creus que en aqueta fila pots llancar una bomba?");

```

```

                fsi
            fmentre
            mentre ((columnaa<0) o
(columnnaa>=caselles))
                escriure("\nIntrodueix la columna
per a disparar: ");
                columnaallegirASCII();
                si (columnaa>=97)
                    columna:=columnaa-32;
                fsi
                columna:=columnaa-65;
                si ((columnaa<0) o
(columnnaa>=caselles))
                    escriure("\nColumna incorrecta
tros d'ase!");
                fsi
            fmentre
            llancamentsa++;
            si (matriujocab[filaa][columnaa]='?')
                si
(matriujocbb[filaa][columnaa]='.')
matriujocbb[filaa][columnaa]:='-';
matriujocab[filaa][columnaa]:='.';
                jugadaa:=1;
                fsi
                si
(matriujocbb[filaa][columnaa]='X')
matriujocbb[filaa][columnaa]:='@';
matriujocab[filaa][columnaa]:='@';
                encertsa++;
                jugadaa:=COMPROVA(matriujocbb,
filaa, columna,caselles, tira);
                si (jugadaa=3)

```

```

                                POSA_AIGUA(matriujocab,
filaa, columnaa, caselles);

                                fsi
                                fsi
                                sino
                                    jugadaa:=0;
                                fsi
                                    puntuacioa:=puntuacioa+jugadaa-1;
                                    escriure("\nResultat:      Jugador      1
casella '':", filaa, columnaa+65);
                                    opció(jugadaa)
                                    valor 0:
                                        escriure(" Casella repetida. ");
                                    valor 1:
                                        escriure(" Aigua. ");
                                    valor 2:
                                        escriure(" Tocat. ");
                                    valor 3:
                                        escriure(" Tocat i enfonsat. ");

                                fopcio

                                si (encertsa=20)
                                    fi:=1;
                                sino
                                    escriure("Torna a tirar.");
                                fsi
                                llegirASCII(escapadetime);
                                    si (escapadetime='\t')
                                        NETEJAR_PANTALLA;
                                        goto menu;
                                    fsi
                                        NETEJAR_PANTALLA;
                                fmentre
                                    puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);

```



```

        si (encertsa=20)
            escriure("\nHa guanyat el jugador 1,
amb '' punts, felicitats!\n", puntuacioa);
        fsi
        PAUSA;
        si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");}
        EMMAGATZEMA_RECORDS(taula);
        IMPRIMIR_RECORDS(taula);
        fi:=0;
        noacabata:=0;
        encertsa:=0;
        jugadaa:=1;
        orientacioa:=-1;
        horitzontala:=-1;
        verticala:=-1;
        puntuacioa:=0;
        llanamentsa:=0;
        load:=0;
        PAUSA;
valor 3:
        si (monflag=1)
            tira:=MONEDA();
            monflag:=0;
        fsi
mentre (fi=0)
            mentre (((jugadaa=2 o jugadaa=3) o
tira=1) i (fi=0))
                NETEJAR_PANTALLA
                filaa:=-1;
                columnaa:=-1;
                VISUALITZA(matriujocaa, matriujocab,
caselles, 1);
                mentre ((filaa<0) o (filaa>=caselles))
                    escriure("\nTorn del jugador 1");

```

```

a disparar: ");
        escriure("\nIntrodueix la fila per
        llegirASCII(filaa);
        filaa:=filaa-48;
        si ((filaa<0) o (filaa>=caselles))
            escriure("\nHas introduit una
fila incorrecta empanat!!");
        fsi
    fmentre
mentre ((columnaa<0) o
(columnaa>=caselles))
        escriure("\nIntrodueix la columna
per a disparar: ");
        llegirASCII(columnaa);
        si (columnaa>=97)
            columnaa:=columnaa-32;
        fsi
        columnaa:=columnaa-65;
        si ((columnaa<0) o
(columnaa>=caselles))
            escriure("\nPero que no veus
que el que acabes de posar esta malament? torna-hi...");
        fsi
    fmentre
        llancamentsa++;
        si (matriujocab[filaa][columnaa]='?')
            si
(matriujocbb[filaa][columnaa]='.')
matriujocbb[filaa][columnaa]:='-';
matriujocab[filaa][columnaa]:='.';
            jugadaa:=1;
            tira:=2;
        fsi
        si
(matriujocbb[filaa][columnaa]='X')

```

```

matriujocbb[filaa][columnaa]:='@';

matriujocab[filaa][columnaa]:='@';

                                encertsa++;
                                jugadaa:=COMPROVA(matriujocbb,
filaa, columnaa,caselles, tira);
                                si (jugadaa=3)
                                    POSA_AIGUA(matriujocab,
filaa, columnaa, caselles);
                                fsi
                                tira:=1;
                                fsi
                                sino
                                    jugadaa:=0;
                                fsi
                                puntuacioa:=puntuacioa+jugadaa-1;
                                escriure("\n\nResultat:      Jugador      1
casella '':", filaa, columnaa+65);
                                opció(jugadaa)
                                valor 0:
                                    escriure(" Casella repetida. ");
                                    tira:=2;
                                valor 1:
                                    escriure(" Aigua. ");
                                valor 2:
                                    escriure(" Tocat. ");
                                valor 3:
                                    escriure(" Tocat i enfonsat. ");
                                fopcio
                                si (encertsa=20)
                                    fi:=1;
                                sino
                                    si ((jugadaa=2) o (jugadaa=3))
                                        escriure("Torna a tirar.");

```

```

        fsi
    fsi
    escapadetime llegirASCII
        si (escapadetime='\t')
            NETEJAR_PANTALLA
            goto menu;
        fsi
    fmentre
        mentre (((jugadab=2) o (jugadab=3) o
(tira=2)) i (fi=0))
            jugadab:=DISPARA(matriujocaa,
matriujocba, filab, columnab, jugadab, caselles,
orientaciob, verticalb, horitzontalb, noacabatb);
            si (jugadab=3)
                POSA_AIGUA(matriujocba, filab,
columnab, caselles);
            fsi
            llancamentsb++;
            NETEJAR_PANTALLA;
            VISUALITZA(matriujocaa, matriujocab,
caselles, 1);

            escriure("\nTorn del jugador 2:");
            puntuaciob:=puntuaciob+jugadab-1;
            escriure("\n\nResultat: Jugador 2
casella '':", filab, columnab+65);
            opció(jugadab)
            valor 1:
                escriure(" Aigua.");
                tira:=1;
            valor 2:
                escriure(" Tocat. ");
                encertsb++;
                si (encertsb<>20)
                    escriure(" Torna a tirar");
            fsi

```

```

        noacabatb:=1;
        tira:=2;
valor 3:
        escriure(" Tocat i enfonsat. ");
        encertsb++;
        si (encertsb<>20)
                escriure(" Torna a tirar.");
        fsi
        noacabatb:=0;
        tira:=2;

fopcio

        llegirASCII(escapadetime);
        si (escapadetime='\t')
                NETEJAR_PANTALLA
                goto menu;
        fsi
        si (encertsb=20)
                fi:=1;
        fsi
        fmentre
        fmentre
                puntuacioa:=CALCULAPUNTUACIO(caselles,
puntuacioa, llancamentsa);
                puntuaciob:=CALCULAPUNTUACIO(caselles,
puntuaciob, llancamentsb);
        si (encertsa=20)
                escriure("\nHa guanyat el jugador 1,
amb '' punts, felicitats!\n", puntuacioa);
        fsi
        si (encertsb=20)
                escriure("\nHa guanyat el jugador 2,
amb '' punts, felicitats!\n", puntuaciob);
        fsi
        getch();

```

```

        si (0=AFEGIR_RECORD(puntuacioa, taula, 1))
escriure("El teu record no s'ha afegit...\n");fsi
        si (0=AFEGIR_RECORD(puntuaciob, taula, 2))
escriure("El teu record no s'ha afegit...\n");fsi
        EMMAGATZEMA_RECORDS(taula);
        IMPRIMIR_RECORDS(taula);
        fi:=0;
        noacabata:=0;
        noacabatb:=0;
        encertsa:=0;
        encertsb:=0;
        jugadaa:=1;
        jugadab:=1;
        orientacioa:=-1;
        orientaciob:=-1;
        horitzontala:=-1;
        horitzontalb:=-1;
        verticala:=-1;
        verticalb:=-1;
        puntuacioa:=0;
        puntuaciob:=0;
        llancamentsa:=0;
        llancamentsb:=0;
        load:=0;
        monflag:=1;
        opciotauler:=-1;
        PAUSA;

        fopcio
valor 4:
        dades.opcio:=opcio;
        dades.caselles:=caselles;
        i:=0;
        j:=0;
        mentre (i<10)

```

```

        mentre (j<10)
            dades.matriujocaa[i][j]:=matriujocaa[i][j];
            j++;
        fmentre
        j:=0;
        i++;
    fmentre
    i:=0;
mentre (i<10)
    mentre (j<10)
        dades.matriujocab[i][j]:=matriujocab[i][j];
        j++;
    fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (i<10)
    mentre (j<10)
        dades.matriujocbb[i][j]:=matriujocbb[i][j];
        j++;
    fmentre
    j:=0;
    i++;
fmentre
i:=0;
mentre (i<10)
    mentre (j<10)
        dades.matriujocba[i][j]:=matriujocba[i][j];
        j++;
    fmentre
    j:=0;
    i++;
fmentre

```

```

dades.noacabata:=noacabata;
dades.noacabatb:=noacabatb;
dades.puntuacioa:=puntuacioa;
dades.puntuaciob:=puntuaciob;
dades.llancamentsa:=llancamentsa;
dades.llancamentsb:=llancamentsb;
dades.tira:=tira;
dades.encertsa:=encertsa;
dades.encertsb:=encertsb;
dades.filaa:=filaa;
dades.filab:=filab;
dades.columnaa:=columnaa;
dades.columnab:=columnab;
dades.jugadaa:=jugadaa;
dades.jugadab:=jugadab;
dades.orientacioa:=orientacioa;
dades.orientaciob:=orientaciob;
dades.verticala:=verticala;
dades.verticalb:=verticalb;
dades.horitzontala:=horitzontala;
dades.horitzontalb:=horitzontalb;
dades.monflag:=monflag;
si (0=EMMAGATZEMA_JOC(dades))
    escriure("No s'ha guardat cap joc...");
sino
    escriure("El teu joc s'ha guardat correctament");
fsi
valor 5:
    IMPRIMIR_RECORDS(taula);
valor 6:
    falgorisme;
valor 7:
    NETEJAR_PANTALLA

```



```

    escriure("\n");
    escriure("\nHi ha tres tipus de joc: cap jugador, on
nomes juga la maquina. Un jugador,");
    escriure("\non nomes jugues tu per a enfonsar tots els
vaixells del contrincant i la maquina");
    escriure("es completament passiva. I per ultim esta la
opcio de jugar tu contra la maquina");
    escriure("que jo he dissenyat per a que perdis un cop
rere un altre muahahahaha. Aquesta ");
    escriure("es l'opcio mes divertida.");
    escriure("A l' opcio de dos jugadors es troba de forma
exclusiva\nl'opcio de dissenyar el");
    escriure(" teu propi tauler de joc.");
    escriure("\n");
    escriure("\nEl joc esta dissenyat per obtenir
informacio de lusuari capturant les pulsacions");
    escriure("del teclat, de manera que un cop hakis
apretat una tecla no hi haura opcio per a");
    escriure("tornar enrere; procura no equivocarte ;).
Les opcions possibles a triar solen ");
    escriure("\nestar escrites al mateix enunciat de la
pregunta indicades entre parentesi,");
    escriure("\no be davant de l'opcio amb un numero, per
exemple: 3.-carregar partida, ");
    escriure("\nes a dir, que per a seleccionar aquella
opcio has de premer 3. De vegades pero,");
    escriure("\nsi la pantalla es queda quieta; el joc
espera que premis qualsevol tecla per a");
    escriure("\npoder continuar.");
    escriure("\n");
    escriure("\nEl joc esta fet a prova de poca-soltes que
intenten rebentar el codi o be a ");
    escriure("\nprova d'australopitecs que no son capacos
d'introduir l'informacio que el joc ");
    escriure("els hi esta demanant. Pots equivocarte sense
problema,pero procura no tocar molt");
    escriure("els pebrots o potser sere jo qui et faci una
visita per a aixafarte la guitarra ");

```

```

        escriure("poca-solta.");
        escriure("\n");
        escriure("\nEs pot sortir d'una partida per a tornar
al menu en el moment que s'anuncia el");
        escriure("\nresultat de la jugada (jugador 1: aigua)
prement la tecla tabulador; en la ");
        escriure("\nmajoria de teclats es troba sobre el bloc
de majuscules (bloq mayus). Això pot ");
        escriure("\nfer-se per a guardar la partida, per
exemple.");
        escriure("\n\nAu va!! prou d'instruccions i a enfonsar
vaixells, que jugar a aquesta andromina");
        escriure("es bufar i fer ampolles!!");
        escriure("\n");
        escriure("\nmolta sort!!! :D\n");
fopcio
        escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");
        getch();
        NETEJAR_PANTALLA
fmentre
falgorisme

```

6 Avaluació

Tots els jocs de proves donen el resultat esperat i demostra que el codi està molt ben treballat.

Com a avaluació general del programa cal dir que he dissenyat el programa de tal manera que l'usuari no pot introduir en el codi o en les variables valors estranys que puguin fer entrar el programa en col·lapse, i que en cas d'intentar-ho no deixa continuar amb el programa fins introduir un valor vàlid. Totes les entrades es troben dins d'un bucle que es podria dir que es un bucle de control de l'entrada, per això penso que els jocs de proves en aquesta documentació no cal que siguin massa llargs ja que només estan permeses le casuístiques que el programa accepta, i per tant reduïm gairebé a 0 tots els errors d'execució. Especialment els que provenen de l'entrada de l'usuari.

7 ANEXO

Caracteres de control ASCII				Caracteres ASCII imprimibles												ASCII extendido											
DEC	HEX	Símbolo ASCII		DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo	DEC	HEX	Símbolo
00	00h	NULL	(carácter nulo)	32	20h	espacio	64	40h	@	96	60h	`	128	80h	Ç	160	A0h	á	192	C0h	Ł	224	E0h	Ó			
01	01h	SOH	(inicio encabezado)	33	21h	!	65	41h	A	97	61h	a	129	81h	ü	161	A1h	í	193	C1h	ł	225	E1h	ô			
02	02h	STX	(inicio texto)	34	22h	"	66	42h	B	98	62h	b	130	82h	é	162	A2h	ô	194	C2h	Ł	226	E2h	Ô			
03	03h	ETX	(fin de texto)	35	23h	#	67	43h	C	99	63h	c	131	83h	â	163	A3h	û	195	C3h	ł	227	E3h	Õ			
04	04h	EOT	(fin transmisión)	36	24h	\$	68	44h	D	100	64h	d	132	84h	ä	164	A4h	ñ	196	C4h	Ł	228	E4h	ö			
05	05h	ENQ	(enquiry)	37	25h	%	69	45h	E	101	65h	e	133	85h	å	165	A5h	Ñ	197	C5h	ł	229	E5h	Ö			
06	06h	ACK	(acknowledgement)	38	26h	&	70	46h	F	102	66h	f	134	86h	ä	166	A6h	ª	198	C6h	Ł	230	E6h	µ			
07	07h	BEL	(timbre)	39	27h	'	71	47h	G	103	67h	g	135	87h	ç	167	A7h	º	199	C7h	ł	231	E7h	þ			
08	08h	BS	(retroceso)	40	28h	(72	48h	H	104	68h	h	136	88h	ê	168	A8h	¿	200	C8h	Ł	232	E8h	ÿ			
09	09h	HT	(tab horizontal)	41	29h)	73	49h	I	105	69h	i	137	89h	ë	169	A9h	®	201	C9h	ł	233	E9h	Ű			
10	0Ah	LF	(salto de línea)	42	2Ah	*	74	4Ah	J	106	6Ah	j	138	8Ah	è	170	AAh	¬	202	CAh	Ł	234	EAh	Ű			
11	0Bh	VT	(tab vertical)	43	2Bh	+	75	4Bh	K	107	6Bh	k	139	8Bh	ï	171	ABh	½	203	CBh	ł	235	EBh	Ű			
12	0Ch	FF	(form feed)	44	2Ch	,	76	4Ch	L	108	6Ch	l	140	8Ch	ì	172	ACH	¼	204	CCh	ł	236	ECh	Ű			
13	0Dh	CR	(retorno de carro)	45	2Dh	-	77	4Dh	M	109	6Dh	m	141	8Dh	í	173	ADh	½	205	CDh	Ł	237	EDh	Ű			
14	0Eh	SO	(shift Out)	46	2Eh	.	78	4Eh	N	110	6Eh	n	142	8Eh	Ā	174	AEh	¾	206	CEh	ł	238	EEh	Ű			
15	0Fh	SI	(shift In)	47	2Fh	/	79	4Fh	O	111	6Fh	o	143	8Fh	Ā	175	AFh	»	207	CFh	Ł	239	EFh	Ű			
16	10h	DLE	(data link escape)	48	30h	0	80	50h	P	112	70h	p	144	90h	Ē	176	B0h	»	208	D0h	ł	240	F0h	Ű			
17	11h	DC1	(device control 1)	49	31h	1	81	51h	Q	113	71h	q	145	91h	æ	177	B1h	»	209	D1h	Ł	241	F1h	Ű			
18	12h	DC2	(device control 2)	50	32h	2	82	52h	R	114	72h	r	146	92h	Æ	178	B2h	»	210	D2h	ł	242	F2h	Ű			
19	13h	DC3	(device control 3)	51	33h	3	83	53h	S	115	73h	s	147	93h	ø	179	B3h	»	211	D3h	Ł	243	F3h	Ű			
20	14h	DC4	(device control 4)	52	34h	4	84	54h	T	116	74h	t	148	94h	ö	180	B4h	»	212	D4h	ł	244	F4h	Ű			
21	15h	NAK	(negative acknowle.)	53	35h	5	85	55h	U	117	75h	u	149	95h	ò	181	B5h	»	213	D5h	Ł	245	F5h	Ű			
22	16h	SYN	(synchronous idle)	54	36h	6	86	56h	V	118	76h	v	150	96h	û	182	B6h	»	214	D6h	ł	246	F6h	Ű			
23	17h	ETB	(end of trans. block)	55	37h	7	87	57h	W	119	77h	w	151	97h	ü	183	B7h	»	215	D7h	Ł	247	F7h	Ű			
24	18h	CAN	(cancel)	56	38h	8	88	58h	X	120	78h	x	152	98h	ÿ	184	B8h	»	216	D8h	ł	248	F8h	Ű			
25	19h	EM	(end of medium)	57	39h	9	89	59h	Y	121	79h	y	153	99h	Ö	185	B9h	»	217	D9h	Ł	249	F9h	Ű			
26	1Ah	SUB	(substitute)	58	3Ah	:	90	5Ah	Z	122	7Ah	z	154	9Ah	Ü	186	BAh	»	218	DAh	ł	250	FAh	Ű			
27	1Bh	ESC	(escape)	59	3Bh	;	91	5Bh	[123	7Bh	{	155	9Bh	ø	187	BBh	»	219	DBh	Ł	251	FBh	Ű			
28	1Ch	FS	(file separator)	60	3Ch	<	92	5Ch	\	124	7Ch		156	9Ch	£	188	BCh	»	220	DCh	ł	252	FCh	Ű			
29	1Dh	GS	(group separator)	61	3Dh	=	93	5Dh]	125	7Dh	}	157	9Dh	Ø	189	BDh	»	221	DDh	Ł	253	FDh	Ű			
30	1Eh	RS	(record separator)	62	3Eh	>	94	5Eh	^	126	7Eh	~	158	9Eh	x	190	BEh	»	222	DEh	ł	254	FEh	Ű			
31	1Fh	US	(unit separator)	63	3Fh	?	95	5Fh	-				159	9Fh	f	191	BFh	»	223	DFh	Ł	255	FFh	Ű			
127	20h	DEL	(delete)																								

elCodigoASCII.com.ar