

El joc d'enfonsar els vaixells

ALUMNE: Laura Romero Huete

laura.romeroH@estudiants.urv.cat

PROFESSOR: Carlos Garcia-Barroso

ASSIGNATURA: Fonaments de programació

ENSENYAMENT: 1r Grau

GRUP: T3

DATA: 10/ Gener / 2016

Índex

ESPECIFICACIONS.....	3
ANÀLISI	6
DISSENY DE DADES	8
DISSENY DE L'ALGORISME.....	11
DISSENY DEL JOC DE PROVES	31
AVALUACIÓ.....	36

Especificacions

Programa principal

Es tracta d'un joc que consisteix a enfonsar tots els vaixells del jugador contrari. Es ostrarà un menú principal on 'usuari podrà escollir una opció, i a partir d'aquí s'aniran cridant els diversos procediments.

Els vaixells a enfonsar són:

- Submarí: 1 casella (x4)
- Dragamines: 2 caselles (x3)
- Destructor: 3 caselles (x2)
- Portaavions: 4 caselles (x1)

Quan es disparà es rep una puntuació depenent de la casella tocada:

- Casella repetida (0)
- Aigua(1)
- Tocat(2)
- Enfonsat(3)

Acció jugar

Aquesta acció s'executa quan la partida es de 0 jugadors, és a dir, quan juga la màquina juga contra ella mateixa.

Rep per paràmetre les coordenades del darrer llançament, en genera unes de noves aleatòriament que retornarà per paràmetre.

També controla que les coordenades siguin valides per les tres mides de taulell possibles.

Funció emmagatzema_records

Aquest procediment emmagatzema els rècords actuals(retornats en forma de taula per la funció afegir rècords) a un fitxer de text. Els records s'emmagatzemaran quan finalitzi una partida, ja que es llavors quan es calcula la puntuació de la partida. Si la puntuació de la partida és inferior als 25 ja no s'inicialitza la funció.

Aquesta funció retorna un booleà informant de si tot el procediment s'ha executat correctament.

Funció recupera_records

Aquest procediment recupera els rècords del fitxer de rècords. Es cridarà quan s'executa el joc per primera vegada per tal de poder comparar les puntuacions amb les de la nova partida.

També s'executa quan es vol veure el podi per pantalla. És per això que també es passa un enter per paràmetre per tal d'imprimir o no per pantalla els rècords del fitxer.

Acció inicia_taulers

L'acció inicia taulells et deixa triar entre inicialitzar un taulell fix (els vaixells sempre estaran disposats igual) o aleatòriament (els vaixells sempre estaran disposats de manera diferent).

Controla que no s'introdueixi un valor no vàlid i retorna la taula de vaixells per paràmetres.

Funció dispara

Primerament la funció dispara omple el taulell de llançaments amb '?'. Demanarà a l'usuari el nombre de jugadors (0 si juga la màquina contra ella mateixa, 1 si juga l'usuari contra la màquina). Aleshores es procedeix a realitzar la partida a partir d'un bucle que controla si s'han enfonsat tots els vaixells o si l'usuari vol continuar o emmagatzemar el joc, això es demanarà al final de la jugada.

Si es tracta d'una partida amb zero jugadors es crida a la funció jugar, i si es tracta d'un jugador es crida a la funció nova jugada.

Aleshores quan s'han retornat les noves coordenades es procedeix a disparar. El resultat s'imprimeix per pantalla juntament amb la coordenada realitzada. Depenent del resultat es modificarà el taulell de llançaments.

Un cop ja s'han realitzat els canvis al taulell, es preguntarà a l'usuari si es vol continuar amb la partida o es vol emmagatzemar el joc.

En el cas de voler seguir es torna a realitzar tot el procediment

En el cas de voler emmagatzemar el joc se surt del bucle i es procedeix a guardar la partida.

Quan es finalitza una partida (s'han enfonsat els deu vaixells) es procedeix a calcular i a retornar un enter amb la puntuació.

Funció afegir_records

Aquesta funció demana el nom del jugador i a partir de la puntuació de la partida (passada per paràmetre) i de la taula, on hi ha els rècords del fitxer, busca la posició del nou rècord i emmagatzema el nou ordre en una altra taula. Aquesta nova taula es guarda amb el nom de la primera per tal de poder-la retornar per referència.

Retorna un booleà informant de si tot el procediment s'ha executat correctament.

Funció emmagatzema_joc

Aquest procediment emmagatzema el joc actual en un fitxer binari per tal de poder seguir en un altre moment la partida. S'executarà quan l'usuari no vulgui començar una partida guardada o quan no vulgui seguir jugant.

Funció recupera_joc

El procediment recuperar_joc recupera la informació del joc emmagatzemat al fitxer binari per tal de poder seguir la partida exactament allà on l'havies deixat.

Aquest procediment s'executa sempre que hi hagi un joc emmagatzemat i l'usuari desitgi recuperar la partida. (només es pot tenir una partida guardada, i no es pot guardar dues vegades una partida)

Acció nova_jugada

L'acció nova jugada s'executa quan la partida es d'1 jugador, és a dir, l'usuari contra la màquina.

Demana a l'usuari que introdueixi unes coordenades noves per a realitzar el següent moviment. Es té en compte que les coordenades que introdueixi l'usuari siguin vàlides.

Anàlisi

Programa principal

El problema plantejat es pot solucionar amb una cerca per trobar una opció (introduïda per l'usuari) correcte.

Per inicialitzar i cridar al procediment corresponent hem implementat una estructura d'opció.

Acció jugar

El problema plantejat es pot solucionar amb una estructura d'opció per determinar els límits de la taula i amb una petita estructura de si per disparar al costat d'un vaixell enfonsat.

Per trobar una nova coordenada aleatòria implementarem una estructura de cerca.

Funció emmagatzema_records

Per a resoldre el problema plantejat en la funció emmagatzema rècords, implementarem una estructura de recorregut per tal de llegir tota la taula de rècords i escriure-la al fitxer de text.

Funció recupera_records

Per a resoldre el problema plantejat en la funció emmagatzema rècords, implementarem una estructura de recorregut per tal de llegir tot el fitxer de text.

Acció inicia_taulells

El problema plantejat a l'hora d'iniciar un taulell es pot solucionar amb una petita estructura de cerca per saber el tipus de taulell que vol l'usuari, i una estructura de condicionals per inicialitzar el taulell corresponent.

Funció dispara

Per a resoldre el problema plantejat a la funció dispara, implementem un recorregut per omplir la taula de llançaments amb '?'. Per demanar a l'usuari el nombre de jugadors implementem una cerca.

Per jugar la partida utilitzem una cerca, que mirarà si s'han enfonsat tots els vaixells o si el jugador vol seguir jugant.

Aleshores s'implementa una petita estructura de condicionals per saber si la partida es d'un o de zero jugadors, i depenent de la resposta es crida una opció o una altre.

Seguidament es crida a la funció dispara i es procedeix a modificar el taulell de llançaments mitjançant una estructura d'opció.

3-vaixell enfonsat: imprimeix aigua al voltant del vaixell enfonsat.

2-vaixell tocat: s'imprimeix una '@' a les coordenades del llançament.

1-aigua: s'imprimeix un '.' a les coordenades del llançament.

0-repetit: no es realitza cap canvi ja que les coordenades ja havien estat modificades anteriorment.

A més a més, si es tracta d'un vaixell enfonsat s'incrementa un contador per tal de controlar si s'ha acabat la partida (quan s'enfonsen 10 vaixells).

Per saber si l'usuari vols seguir amb la partida o no implementem una cerca per buscar una resposta vàlida.

Quan es finalitza la partida (s'han enfonsat els 10 vaixells), es calcula la puntuació.

Funció afegir_record

Per afegir un rècord, primer procedim a comparar, amb un recorregut de la taula de rècords, els rècords ja emmagatzemats, amb l'actual per veure si son menors o no. Un cop es troba la posició del nou rècord s'acaba d'omplir la taula amb un altre recorregut.

Si és el primer rècord sempre s'escriu, no entra dins del recorregut , ja que sempre li tocarà la primera posició

Funció emmagatzema_joc

Per tal d'emmagatzemar el joc es van escrivint totes les dades a fitxer binari, cal tenir en compte que l'ordre ha de ser el mateix després alhora de llegir

Funció recupera_joc

Per tal d'emmagatzemar el joc es van llegint totes les dades del fitxer binari, cal tenir en compte que l'ordre ha de ser el mateix en el que s'han escrit al fitxer per tal de llegir correctament cada dada.

Acció nova_jugada

Per poder saber quins són els límits, hem implementat una opció per tal de donar una lletra d'acord amb la mida del taulell.

Per demanar les coordenades implementem dues cerques, una per la lletra i l'altre per als números. Cada una controla que el valor introduït sigui correcte.

Disseny de dades

Programa principal

Per saber si l'opció introduïda per teclat és correcte controlem que sigui un enter entre 1 i el número màxim d'opcions.

Només es pot sortir del programa triant la opció sortir, ja que sempre s'acaba tornant al menú.

Acció jugar

Per poder saber quins són els límits, hem implementat una opció per tal de donar una lletra d'acord amb la mida del taulell.

Per comprovar si la nova coordenada generada dispara a una casella '?', hem implementat una cerca, que transforma les coordenades reals a les d'una taula de caràcters i mira si correspon a un '?' en la taula, si no es així es genera una altre coordenada, fins a trobar-ne una de correcte, es a dir que encara no hagi sigut disparada.

Funció emmagatzema_records

Per tal de llegir la taula de rècords tenim com a sortida del bucle una variable del numero de rècords.

Funció recupera_records

Per tal d'omplir la taula de rècords, mirem si existeix el fitxer de rècords, si no es així es mostra un missatge per pantalla dient que encara no hi ha rècords emmagatzemats. Del contrari, es procedeix a llegir el fitxer fins al final.

També disposem d'una variable que ens indica si hem de mostrar els rècords per pantalla o no. Aquesta variable serà passada per paràmetre a la funció.

Acció inicia_taulells

Implementa una cerca que demana a l'usuari si el taulell que es vol crear es fix o aleatori i controla que el valor que introdueix l'usuari es correcte. De no ser així es torna a demanar a l'usuari el tipus de taulell que desitja.

Aleshores depenent del resultat es crida una funció o una altre.

Funció dispara

Per omplir la taula de llançaments fem una estructura de recorregut per les files i una altre per les columnes de tal manera que recorrin només la part el taulell igual a la dimensió que ha triat el jugador.

Per triar el numero de jugadors es realitza una cera per tal que l'usuari introdueixi valors vàlids.

La cerca implementada per realitzar la partida, controlarà per una banda el nombre de vaixells enfonsats, i per altre si el jugador vol seguir.

Amb una estructura condicional es cridarà a la funció jugar o nova jugada depenent del nombre de jugadors.

Per modificar el resultat es fa amb una estructura d'opció:

En el cas de vaixell tocat o de tocar aigua només es canvia la coordenada concreta. Si es tracta d'enfonsar un vaixell, també utilitzo estructures de cerca per buscar-ne els límits i així poder posar aigua a tot el voltat (reduint el nombre de tirades per acabar el joc). Si es tracte d'una casella repetida (només possible en 1 jugador) no es realitza cap canvi.

Per demanar a l'usuari si vol seguir jugant, implementem una cerca perquè l'usuari introdueixi una opció correcta.

En el cas de no voler seguir es procedeix a emmagatzemar el joc. Es passen tant la taula de vaixells i la taula de llançaments a una estructura de dades, i es crida a la funció emmagatzemar joc. També es posa la puntuació final a 0 ja que al sortir del bucle sense finalitzar la partida, no s'ha de calcular la puntuació ni emmagatzemar el rècord.

La funció dispara retorna un enter amb la puntuació final de la partida.

Funció afegir_record

Per tal de comparar els rècords tenim una taula amb el nom i la puntuació de les partides anteriors.

Amb un recorregut anem comparant les puntuacions velles amb la nova. Si la nova puntuació es menor, les anem escrivint en una altra taula d'iguals característiques que l'anterior.

Quan es troba la posició de la nova puntuació se surt del bucle, s'escriu a la nova taula la puntuació i s'entra en un nou recorregut per tal d'acabar d'omplir la taula en el cas que hi hagi rècords i espai lliure a la taula, sinó es surt de la funció

Funció emmagatzema_joc

Per procedir a emmagatzemar el joc, primerament es demanarà a l'usuari el seu nom. A continuació s'obre el fitxer per escriptura i es procedeix a escriure-hi tota la informació adient. Al acabar es torna a tancar el fitxer.

Funció recupera_joc

Per a recuperar el joc es procedeix a obrir el fitxer en mode lectura. Seguidament es van llegir totes les dades i guardant-les en una variable adient, tenint en compte l'ordre en que han estat escrites al fitxer.

Acció nova_jugada

Per poder saber quins són els límits, hem implementat una opció per tal de donar una lletra d'acord amb la mida del taulell.

Per demanar les coordenades implementem dues cerques, una per la lletra i l'altre per als números. Cada una controla que el valor introduït sigui correcte, de no ser així tornarà a demanar a l'usuari o la lletra o el numero introduït incorrectament, fins que aquest sigui vàlid.

Disseny de l'algorisme

Definició de constants

FITXER_RECORDS:="podium.txt"

FITXER_JOC:="partida.bin"

NUM:=25

NOM:=20

Definició de tipus

Tipus record_tipus és:

Nom[NOM]: caràcter;

Num: enter;

Ftipus

Tipus coor_tipus és:

Fila: caràcter;

Col: enter;

Ftipus

Tipus jugador_tipus és:

trets_fets, vaixells_enfonsats, resultat: enter;

taulell_vaixells[DIM_MAX][DIM_MAX],

taulell_llancaments[DIM_MAX][DIM_MAX]: caràcter;

Ftipus

Programa principal

Algorisme programa principal és:

```
var
    opcio_m, partida_guardada, dim, c=0, num_records=0, resultat, k,
    jugadors: enter;

    taulell[DIM_MAX][DIM_MAX], f='Z', nomf[NUM]: caràcter;
    record[NUM]: record_tipus;
    correcte: booleà;
    joc: booleà;
    taula_j: jugador_tipus;
    fvar
    inici
    correcte:=false;
    opcio_m:=0;
    partida_guardada:=0;

    nomf: FITXER_RECORDS;

    correcte:=recupera_records (nomf, &num_records, record, 0);

mentre
    ((opcio_m≠4) o (opcio_m<1) o (opcio_m>4) o (partida_guardada=0))
        escriure("\nTria una opcio introduint el numero
        corresponent:\t");

        escriure ("\n1-Crear un nou joc.");
        escriure ("\n2-carregar joc emmagatzemat.");
        escriure ("\n3-Veure podium.");
        escriure ("\n4-Sortir del joc.\n");

        llegir(opcio_m);
```

```

opcio(opcio_m)

valor 1: opcio_m:=0;

    escriure("\nIndica les dimensions del taulell(8, 9,
10): ");

    llegit(dim);

    mentre (dim<8 o dim>10)

        escriure("\nIndica les dimensions del taulell(8,
9, 10): ");

        llegir(dim);

    fmentre

    inicia_taulells(dim, taulell);

    mentre ((opcio_m<1)o(opcio_m>2))

        escriure("\nDespres de crear el joc que vols
fer?");

        escriure ("\n1-Jugar partida.");

        escriure ("\n2-Emagatzemar partida\n");

        llegir("%i", &opcio_m);

    fmentre

opcio(opcio_m)

valor 1: partida_guardada=1;

    resultat:=disparas(dim, f, c, taulell, 2, taula_j);

    si (resultat=0) break;

    fsi

    si ((num_records<25)o(resultat>record[num_records-
1].num))

        correcte:=afegir_record( nomf, record, &num_records,
resultat);

    fsi

    si (correcte=true) escriure("\nCorrecte afegir!");

```

```

        fsi

valor 2:

mentre ((jugadors≠0) i (jugadors≠1))

    escriure("\nIndica numero de jugadors(0-maquina, 1-
usuari): ");

    llegir(jugadors);

fmentre

    taula_j.trets_fets:=0;

    taula_j.vaixells_enfonsats:=0;

    taula_j.resultat:=0;


per(k=0; k<DIM_MAX; k=k+1)

    taula_j.taulell_vaixells[k]:=taulell[k];

fper

per(k=0; k<DIM_MAX; k=k+1)

    taula_j.taulell_llancaments[k]:=taulell[k];

fper

nomf:=FITXER_JOC);

correcte:=emmagatzema_joc(nomf, dim, jugadors, taula_j);

si (correcte=true) escriure("\nJoc guardat correctament!");

sino escriure("\nERROR AL GUARDAR JOC!");

fsi

partida_guardada:=1;

fopcio


valor 2:   joc:=obrir("partida.bin", "r");

si (joc=NULL) escriure ("\nNo tens cap partida
gurdada!\n");

sino  nomf:=FITXER_JOC;

```

```

fsi

correcte:=recupera_joc(nomf, &dim, &jugadors, &taula_j);

si (correcte=true) escriure("\nJoc carregat
correctament!");

sino escriure("\nERROR AL carregar JOC!");

fsi

escriure("\nDespres de carregar el joc emmagatzemat que
vols fer?");

escriure ("\n1-Jugar partida.");

escriure ("\n2-Emmagatzemar partida.\n");

llegir(opcio_m);

opcio(opcio_m)

valor 1: partida_guardada=1;

escriure("menu vaixells enfonsats:%i",
taula_j.vaixells_enfonsats);

resultat:=disparas(dim, f, c, taulell, jugadors, taula_j);

correcte:=emmagatzema_records(nomf, &num_records,
&record[NUM]);

si (correcte=true) escriure("\nCorrecte!");

fsi

valor2: escriure("\nOpcio 2");

partida_guardada:=1;

fopcio

valor 3: partida_guardada:=1;

correcte:=recupera_records (nomf, &num_records, record, 1);

si (correcte=true) escriure("\nCorrecte!");

else escriure("\nIncorrecte");

```

```

    fsi

    valor4: retorna 0;

    fopcio

fmentre

falgorisme

```

Acció jugar

Capçalera jugar és:

```

Acció jugar(int dim, char f, int *c, char
taulell_llancament[][DIM_MAX], int res)

```

Algorisme jugar és:

```

var

max: caràcter;

int j, p: enter;

fvar

inici

opció (dim)

    valor 8: max:='H';

    valor 9: max:='I';

    valor 10: max:='J';

fopció

si (res=2)

j:=(real)f-65;

P:=c-1;

    si((j>0) i (taulell_llancament [j-1][p]='?')) f:=(char)j-
1+65;

```



```

sino
    {si ((j<(dim-1)) i (taulell_llancament [j+1][p]='?')) °
    f:=(char)j+1+65;

sino
    {si ((p>0) i (taulell_llancament[j][p-1]='?'))
    c:=p-1;

    else c=p+1;

    fsi

    fsi

    fsi

sino
c:=1+rand()%(dim+1-1);
f:='A'+rand()%(max+1-'A');

j:=(real)f-65;
p:=c-1;

mentre (taulell_llancament [j][p]!='?')
    c:=1+rand()%(dim+1-1);
    f:='A'+rand()%(max+1-'A');
    j:=(real)f-65;
    p:=c-1;

femntre
fsino
facció

```

Funció emmagatzema_records

Capçalera emmagatzema_records és:

Funció emmagatzema_records(char nomf[], int *num_records, record_tipus taula[NUM]) retorna booleà.

Algorisme emmagatzema_records és:

```
Var
Fitxer: FILE;
I:enter;
Fvar
inici
i:=0;
fitxer:=obrir(nomf, "w");

per (i:=0; i<num_records; i=i+1)
    escriuref(fitxer, "%s\t%i\n", taula[i].nom, taula[i].num);
fper
tancar(fitxer);
retorna(true);
ffunció
```

Funció recupera_records

Capçalera recupera_records és:

funció recupera_records(char nomf[], int *num_records, record_tipus record[NUM], int imprimir) retrna booleà

Algorisme recupera_records és:

```
var
fitxer: FILE;
i: enter;
fvar
```

```

inici

i:=0;

fitxer:=obrir(nomf, "r");

si (fitxer=NULL) escriure("\nEncara no hi ha records
registrats.");

sino      si( no ftell(fitxer)) escriure("Encara no hi ha
records registrats.");

sino

llegirf(fitxer, "%s", record[i].nom);

llegirf (fitxer, "%d", &record[i].num);

mentre (no final(fitxer))

    si (imprimir=1)

        escriure(\nrecord[i].nom);

        escriure(\trecord[i].num);

    fsi

    i=i+1;

    llegirf(fitxer, "%s", record[i].nom);

    llegirf(fitxer, "%d", &record[i].num);

fmentre

num_records:=i;

tancar(fitxer);

fsi

fsi

retorna (true);

ffunció

```

Acció inicia_taulells

Capçalera inicia_taulells és:

```
acció inicia_taulells(int dim, char taulell_vaixells[][DIM_MAX])
```

Algorisme inicia_taulells és:

```
var
opcio: enter;
fvar
inici
opcio:=0;
mentre ((opcio<1)o(opcio>2))
    escriure("\nTipus de taulell(1-Fixe, 2-Aleatori): ");
    llegir(opcio);
fmentre
si (opcio=1) (inicia_taulell_fix (dim, taulell_vaixells));
fsi
si (opcio=2) (inicia_taulell(dim, taulell_vaixells));
fsi
facció
```

Funció dispara

Capçalera dispara és:

```
funció disparas(int dim, char f, int c, char
taulell_vaixells[][DIM_MAX]) retorna enter
```

Algorisme dispara és:

```
var
i, j, p, k, mida, res, res_total=0, num_disp, puntuacio,
jugadors, fila, columna, fila1, fila2, col1, col2, max2: enter;
taulell_llancaments[DIM_MAX][DIM_MAX], lletres[DIM_MAX], LLETRA,
seguir, nomf[NUM]: caràcter;
```

```

correcte: booleà;
taula_j: jugador_tipus;
coordenades[1]: coor_tipus;

fvar
inici
correcte:=true;
mida:= dim;
num_disp:=0;
i:=0;
puntuacio:=0
jugadors:=2;
LLETRA:='A';

srand (time(NULL));
per(i:=0; i<=DIM_MAX; i=i+1)
    lletres[i]:= LLETRA;
    LLETRA=lletra+1;

Fper
per (j:=0; j<mida; j=j+1)
    per (c:=0; c<mida; c=c+1)
        taulell_llancaments[j][c]:='?';
    fper
fper
mentre((jugadors≠0) i (jugadors≠1))
    escriure("\nIndica numero de jugadors(0-maquina, 1-usuari): ");
    llegir (jugadors);
femntre
res:=0;

```

```

i:=0;
f:='\0';
c:=0;
seguir:='1';

mentre ((i<10)i(seguir='1'))
    escriure("\n ");
        per(j:=0; j<mida; j=j+1)
            escriure("%i ", j+1 )
            fper
        escrriure("\n");
        per (j:=0; j<mida; j=j+1)
            escriure("\n%c ", lletres[j] );
                per (k:=0; k<mida; k=k+1)
                    escriure("%c ",taulell_llancaments[j][k]);
                fper
        fper
        escriure("\n");

si (jugadors=0) jugar(dim, &f, &c, taulell_llancaments, res);
sino
j:=(real)f-65;
p:=c-1;
mentre (taulell_llancaments [j][p]!='?')
    nova_jugada(dim, coordenades);
    c:=coordenades[0].col;
    f:=coordenades[0].fila;
    j:=(real)f-65;

```

```

        p:=c-1;

fmentre

fsi

res:= dispara (f, c, taulell_vaixells);
escriure("\nResultat (%c,%d): %d", f, c, res);

j:=(real)f-65;

p:=c-1;

opcio(res)

    valor 3: taulell_llancaments[j][p]:='@';

    fila:=(real)f-65;

    columna:=c-1;

    max2:=dim-1;

    col:=columna-1;

    si (col1<0) col1:=0;

    mentre((col1≠0) i (taulell_llancaments[fila][col1]='@'))
    col1:=col1-1;

    fmentre

    col2:=columna+1; si (col2>max2) col2=max2;

    mentre((col2≠max2) i (taulell_llancaments[fila][col2]='@'))
    col2=col2+1;

    fmentre

    fila1:=fila-1; si (fila1<0) fila1:=0;

    mentre((fila1≠0) i (taulell_llancaments[fila1][columna]='@'))
    fila1:=fila1-1;

    fmentre

    fila2:=fila+1; si (fila2>max2) fila2:=max2;

    mentre((fila2≠max2) i (taulell_llancaments[fila2][columna]='@
'))

```

```

    fila2:=fila2+1;

    fmentre

    per (fila:=fila1; fila<=fila2; fila=fila+1)

        per (columna:=col1; columna<=col2; columna=columna+1)

            si (taulell_llancaments[fila][columna]='?')
                taulell_llancaments[fila][columna]:='.';

            fper

        fper

    valor 2: taulell_llancaments[j][p]:='@';
    valor 1: taulell_llancaments[j][p]:='.';
    valor 0:

fopcio

si (res=3) i=i+1;
res_total:=res_total+res;
num_disp=num_disp+1;

seguir:='2';
mentre((seguir≠'1')i(seguir≠'0'))

    escriure("\n(1-Continuar, 0-Guargar): ");

    llegir (seguir);

fmentre

fmentre

si (seguir='0')

taula_j.trets_fets:=num_disp;
taula_j.vaixells_enfonsats:=i;

```



```

per(k:=0; k<DIM_MAX; k=k+1

    taula_j.taulell_vaixells[k]:=taulell_vaixells[k];

    taula_j.taulell_llancaments[k]:=taulell_llancaments[k];

fper

nomf:=FITXER_JOC;

correcte:=emmagatzema_joc(nomf, dim, jugadors, taula_j);

si (correcte=true) escriure("\nJoc guardat correctament!");

sino escriure("\nERROR AL GUARDAR JOC!");

puntuacio:=0;

fsi

sino

puntuacio:=(100*((float)mida/(float)num_disp)*(float)res_total);


$$\text{puntuació} := 100 * \frac{(\text{real})\text{mida}}{(\text{real})\text{num}_{\text{disp}}} * (\text{real})\text{res}_{\text{total}} \quad (1)$$


escriure("\nPuntuacio de la partida: %i", puntuacio);

fsino

retorna puntuacio;

ffuncio

```

Funció afegir_records

Capçalera afegir_records és:

funció afegir_record(char nomf[], record_tipus record[NUM], int *num_records, int puntuacio) retorna booleà

Algorisme afegir_records és:

```

var

taula[NUM]: record_tipus;

i, j, fet: enter;

```

(1) Formula per calcular la puntuació final

```

correcte: booleà;

nom[NOM]: caràcter;

fvar

inici

j:=0;

fet:=0;

escriure("\nIntrodueix el nom: ");

llegir ( nom);


per(i:=0; i<num_records; i=i+1)
    si (puntuacio>record[i].num i fet=0)
        taula[j].num:=puntuacio;
        taula[j].nom:=nom;
        j:=j+1;
        fet:=1;
    fsi
    si (j=25) break;
    fsi
    taula[j].num:=record[i].num;
    taula[j].nom:= record[i].nom;
    j:=j+1;
fper

si(num_records=0)
    taula[j].num:=puntuacio;
    taula[j].nom:= record[i].nom;
    j=1;
fsi

```

```

num_records:=j;
per(i:=0; i<num_records; i=i+1)
    record[i].num:=taula[i].num;
    taula[j].nom:= record[i].nom;
fper

correcte:=emmagatzema_records(nomf, &*num_records, record);
si (correcte=true) escriure("\nCorrecte emagatzema!");

retorna(true);
ffunció

```

Funció emmagatzema_joc

Capçalera emmagatzema_joc és:

```

funció emmagatzema_joc (char nomf[], int dim, int num_jugadors,
jugador_tipus taula_j) retorna booleà

```

Algorisme emmagatzema_joc és:

```

Joc: fitxer;
nom_jugador[NOM]: caràcter;

escriure("\nIntrodueix el teu nom: ");
llegir ( nom_jugador);

joc:=obrir(nomf,"wb");

escriuref(nom_jugador);
escriuref (dim);
escriuref (num_jugadors);

```

```

escriuref (taula_j.trets_fets);
escriuref (taula_j.vaixells_enfonsats);
escriuref (taula_j.resultat);
escriuref (taula_j.taulell_llancaments);
escriuref (taula_j.taulell_vaixells);

```

```

tancar(joc);

retorna true;

```

Funció recupera_joc

Capçalera recupera_joc és:

funció recupera_joc (char nomf[], int *dim, int *num_jugadors,
jugador_tipus *taula_j) retorna booleà

Algorisme recupera_joc és:

```

Joc: fitxer;

nom_jugador[NOM]: caràcter:

joc:=fopen(nomf,"rb");

llegirf(nom_jugador);
llegirf (dim);
llegirf (num_jugadors);
llegirf (taula_j.trets_fets);
llegirf (taula_j.vaixells_enfonsatsjoc);
llegirf (taula_j.resultat);
llegirf (taula_j.taulell_llancaments);
llegirf (taula_j.taulell_vaixells);

```

```
tancar(joc);  
retorna true;
```

Acció nova_jugada

Capçalera nova_jugada és:

```
Acció nova_jugada(int dim, coor_tipus coor[])
```

Algorisme nova_jugada és:

```
Var  
F, lletra: caràcter;  
C; enter;  
Fvar inici  
f:='Z';  
lletra:='Z';  
c:=20;  
opcio (dim)  
    valor 8: lletra='H';  
    valor 9: lletra='I';  
    valor 10: lletra='J';  
fopcio  
mentre ((f>lletra)o(f<'A'))  
    escriure ("\nIntrodueix el caracter per la cordenada: ");  
    llegir (f);  
fmentre  
mentre ((c>dim)o(c<1))  
    escriure ("\nIntrodueix el numero per la cordenada: ");  
    llegir (c);  
fmentre  
coor[0].fila:=f;
```

```
coor[0].col:=c;
```

```
facció
```

Disseny del joc de proves

En aquest apartat el que es pretén és determinar quins són els diferents tipus d'entrades que ens podem trobar i comprovar que l'aplicació els tracta tots de forma correcta.

Programa principal

Cas	Entrada	Sortida teòrica
Sortir directament	4	sortir
Crear un joc i jugar partida	1 8/9/10 1	Es crea la partida i es juga
Crear un joc i emmagatzemar	1 8/9/10 2	Es crea la partida i s'emmagatzema sense cap tirada
Carregar joc i jugar	2 i 1	Es carrega la partida i es juga
Carregar joc i emmagatzemar	2 i 2	Es carrega la partida i es torna a emmagatzemar
Veure pòdium	3	Es mostren tots els rècords

Acció jugar

Cas	Entrada	Sortida teòrica
Resultat anterior \neq 2	1 8/9/10 1/2 0 1...	Es dispara aleatòriament
Resultat anterior = 2 i posició superior diferent a aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a dalt de la casella tocada
Resultat anterior = 2 i posició superior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la dreta de la casella tocada
Resultat anterior = 2 i posició superior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la casella inferior a la casella tocada
Resultat anterior = 2 i posició superior, dreta i inferior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la casella a l'esquerra de la casella tocada

Resultat anterior = 2 i posició superior, dreta, inferior i esquerra aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara aleatòriament
---	-----------------------	--------------------------

Funció emmagatzema_records

Cas	Entrada	Sortida teòrica
Fitxer inexistent	1 8/9/10 1/2 1 0 1... nom	Es crea el fitxer
taula amb 25 rècords	1 8/9/10 1/2 1 0 1... nom	S'escriuen els 25 rècords un sota l'altre
taula amb 1 rècord	1 8/9/10 1/2 1 0 1... nom	S'escriu el rècord al fitxer
Taula amb 13 rècords	1 8/9/10 1/2 1 0 1... nom	S'escriuen els 13 rècords un sota l'altre
Taula buida	-	No s'arriba a cridar la opció emmagatzema_records

Funció recupera_records

Cas	Entrada	Sortida teòrica
Fitxer inexistent	3	Missatge d'error
Fitxer buit	3	Missatge d'error
Fitxer amb 25 rècords	3	S'omplen 25 posicions de la taula
Fitxer amb 1 rècord	3	S'omple una posició de la taula
Fitxer amb 13 rècords	3	S'omplen 13 posicions de la taula

Acció inicia_taulells

Cas	Entrada	Sortida teòrica
Entrada no vàlida	5	Tipus de taulell (1-Fixe, 2-Aleatori):
Crear taulell fix	1	Es crea un taulell sempre igual
Crear taulell aleatori	2	Es crea un taulell diferent cada vegada

Funció dispara

Cas	Entrada	Sortida teòrica
Després de crear una partida	1 8/9/10 1	Es demana si es vol seguir o guardar la partida
Després de carregar una partida	2 1	Es demana si es vol seguir o guardar la partida
Cap partida creada	-	No s'arriba a disparar

Funció afegir_rècords

Cas	Entrada	Sortida teòrica
Taula de rècords buida	-	No s'arriba a cridar la funció afegir_rècords
Taula de rècords mig plena	1 8/9/10 1/2 1 0 1(fins al final) nom...	Taula de rècords mig plena
Taula de rècords plena i el records és més petit que l'últim	-	No s'arriba a cridar la funció afegir_rècords
Taula de rècords plena i el records és més gran que l'últim	1 8/9/10 1/2 1 0 1(fins al final) nom...	S'escriu el rècord en la posició que li pertoca de la taula i s'elimina l'últim.
Rècord major al primer	1 8/9/10 1/2 1 0 1(fins al final) nom...	S'escriu el rècord en la primera posició i s'elimina l'últim

Funció emmagatzema_joc

Cas	Entrada	Sortida teòrica
Emmagatzemar joc sense cap tirada	1 8/9/10 1/2 0	S'emmagatzema el joc al fitxer
Emmagatzemar joc amb tirades fetes	1 8/9/10 1/2 1 1/0 0	S'emmagatzema el joc al fitxer
Emmagatzemar joc després de carregar joc	2 2	S'emmagatzema el joc al fitxer

Funció recupera_joc

Cas	Entrada	Sortida teòrica
Fitxer inexistent	2	No hi ha cap joc guardat
Joc guardat i finalitzar partida	2 1(fins al final)	Es recupera el joc i pots seguir jugant o tornar a emmagatzemar-lo
Recuperar joc i tornar a guardar	2 0	Es recupera el joc i pots seguir jugant i es torna a emmagatzemar correctament

Funció nova_jugada

Cas	Entrada	Sortida teòrica
Límit numèric superior dim=10	1 10 1/2 1 1 10	Disparar
Límit numèric superior dim=9	1 9 1/2 1 1 9	Disparar
Límit numèric superior dim=8	1 8 1/2 1 1 8	Disparar
Límit numèric inferior dim=10	1 8/9/10 1/2 1 1 1	Disparar
Límit lletra superior dim=10	1 10 1/2 1 1 1 J	Passa a demanar el numero
Límit lletra superior dim=9	1 9 1/2 1 1 1 I	Passa a demanar el numero
Límit lletra superior dim=8	1 8 1/2 1 1 1 H	Passa a demanar el numero
Límit lletra	1 8/9/10 1 1 1 A	Passa a demanar el

inferior		numero
Límit numero i lletra inferior	1 8/9/10 1/2 1 1 1 A	Disparar
Límit lletra inferior numero superior dim=10	1 10 1/2 1 1 J 10	Disparar
Límit lletra inferior numero superior dim=9	1 9 1/2 1 1 I 9	Disparar
Límit lletra inferior numero superior dim=8	1 8 1/2 1 1 H 8	Disparar
Lletra fora dels límits	1 8 1/2 1 1 K	Es torna a demanar la lletra
Numero fora dels límits	1 8 1/2 1 1 H 15	Es torna a demanar el número

Avaluació

A l'avaluació hem de detallar els resultats de fer les proves especificades al disseny i justificar els resultats reals obtinguts.

Programa principal

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Sortir directament	4	sortir	sortir	Si
Crear un joc i jugar partida	1 8/9/10 1	Es crea la partida i es juga	Es crea la partida i es juga	Si
Crear un joc i emmagatzemar	1 8/9/10 2	Es crea la partida i s'emmagatzema sense cap tirada	Es crea la partida i s'emmagatzema sense cap tirada	Si
Carregar joc i jugar	2 i 1	Es carrega la partida i es juga	Es carrega la partida i es juga	Si
Carregar joc i emmagatzemar	2 i 2	Es carrega la partida i es torna a emmagatzemar	Es carrega la partida i es torna a emmagatzemar	No
Veure pòdium	3	Es mostren tots els rècords	Es mostren tots els rècords	Si

Acció jugar

Cas	Entrada	Sortida teòrica	Sortida teòrica	Correcte?
Resultat anterior ≠ 2	1 8/9/10 1/2 0 1...	Es dispara aleatòriament	Es dispara aleatòriament	Si
Resultat anterior = 2 i posició superior diferent a aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a dalt de la casella tocada	Es dispara a dalt de la casella tocada	Si
Resultat anterior = 2 i posició superior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la dreta de la casella tocada	Es dispara a la dreta de la casella tocada	Si
Resultat anterior = 2 i posició superior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la casella inferior a la casella tocada	Es dispara a la casella inferior a la casella tocada	Si
Resultat anterior = 2 i posició superior, dreta i inferior aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara a la casella a l'esquerra de la casella tocada	Es dispara a la casella a l'esquerra de la casella tocada	Si

Resultat anterior = 2 i posició superior, dreta, inferior i esquerra aigua o final del taulell	1 8/9/10 1/2 1 0 1...	Es dispara aleatòriament	Es dispara aleatòriament	Si
---	-----------------------	--------------------------	--------------------------	----

Funció emmagatzema_records

Cas	Entrada	Sortida teòrica	Sortida teòrica	Correcte?
Fitxer inexistent	1 8/9/10 1/2 1 0 1... nom	Es crea el fitxer	Es crea el fitxer	Si
taula amb 25 rècords	1 8/9/10 1/2 1 0 1... nom	S'escriuen els 25 rècords un sota l'altre	S'escriuen els 25 rècords un sota l'altre	Si
taula amb 1 rècord	1 8/9/10 1/2 1 0 1... nom	S'escriu el rècord al fitxer	S'escriu el rècord al fitxer	Si
Taula amb 13 rècords	1 8/9/10 1/2 1 0 1... nom	S'escriuen els 13 rècords un sota l'altre	S'escriuen els 13 rècords un sota l'altre	Si
Taula buida	-	No s'arriba a cridar la opció emmagatzema_records	No s'arriba a cridar la opció emmagatzema_records	Si

Funció recupera_records

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Fitxer inexistent	3	Missatge d'error	Missatge d'error	Si
Fitxer buit	3	Missatge d'error	Missatge d'error	Si
Fitxer amb 25 rècords	3	S'omplen 25 posicions de la taula	S'omplen 25 posicions de la taula i s'imprimeixen per pantalla	Si
Fitxer amb 1 rècord	3	S'omple una posició de la taula	S'omple una posició de la taula i s'imprimeixen per pantalla	Si
Fitxer amb 13 rècords	3	S'omplen 13 posicions de la taula	S'omplen 13 posicions de la taula i s'imprimeixen per pantalla	Si

Acció inicia_taulells

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Entrada no vàlida	5	Tipus de taulell (1-Fixe, 2-Aleatori):	Tipus de taulell (1-Fixe, 2-Aleatori):	Si
Crear taulell fix	1	Es crea un taulell sempre igual	Es crea un taulell sempre igual	Si
Crear taulell aleatori	2	Es crea un taulell diferent cada vegada	Es crea un taulell diferent cada vegada	Si

Funció dispara

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Després de crear una partida	1 8/9/10 1	Es demana si es vol seguir o guardar la partida	Es demana si es vol seguir o guardar la partida	Si
Després de carregar una partida	2 1	Es demana si es vol seguir o guardar la partida	Es demana si es vol seguir o guardar la partida	Si
Cap partida creada	-	No s'arriba a disparar	No s'arriba a disparar	si

Funció afegir_rècords

Cas	Entrada	Sortida teòrica	Sortida teòrica	Correcte?
Taula de rècords buida	-	No s'arriba a cridar la funció afegir_rècords	No s'arriba a cridar la funció afegir_rècords	Si
Taula de rècords mig plena	1 8/9/10 1/2 1 0 1(fins al final) nom...	Taula de rècords mig plena	Taula de rècords mig plena	Si
Taula de rècords plena i el records és més petit que l'últim	-	No s'arriba a cridar la funció afegir_rècords	No s'arriba a cridar la funció afegir_rècords	Si
Taula de rècords plena i el records és més gran que l'últim	1 8/9/10 1/2 1 0 1(fins al final) nom...	S'escriu el rècord en la posició que li pertoca de la taula i s'elimina l'últim.	S'escriu el rècord en la posició que li pertoca de la taula i s'elimina l'últim.	Si
Rècord major al primer	1 8/9/10 1/2 1 0 1(fins al final) nom...	S'escriu el rècord en la primera posició i s'elimina l'últim	S'escriu el rècord en la primera posició i s'elimina l'últim	Si

Funció emmagatzema_joc

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Emmagatzemar joc sense cap tirada	1 8/9/10 1/2 0	S'emmagatzema el joc al fitxer	S'emmagatzema el joc al fitxer	Si
Emmagatzemar joc amb tirades fetes	1 8/9/10 1/2 1 1/0 0	S'emmagatzema el joc al fitxer	S'emmagatzema el joc al fitxer	Si
Emmagatzemar joc després de carregar joc	2 2	S'emmagatzema el joc al fitxer	S'emmagatzema el joc al fitxer	No

Funció recupera_joc

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Fitxer inexistent	2	No hi ha cap joc guardat	No hi ha cap joc guardat	Si
Joc guardat i finalitzar partida	2 1(fins al final)	Es recupera el joc i pots seguir jugant o tornar a emmagatzemar-lo	Es recupera el joc i pots seguir jugant o tornar a emmagatzemar-lo	Si
Recuperar joc i tornar a guardar	2 0	Es recupera el joc i pots seguir jugant i es torna a emmagatzemar correctament	Es recupera el joc i pots seguir jugant però no es torna a emmagatzemar correctament	no

Funció nova_jugada

Cas	Entrada	Sortida teòrica	Sortida real	Correcte?
Límit numèric superior dim=10	1 10 1/2 1 1 10	Disparar	Disparar	Si
Límit numèric superior dim=9	1 9 1/2 1 1 9	Disparar	Disparar	Si
Límit numèric superior dim=8	1 8 1/2 1 1 8	Disparar	Disparar	Si
Límit numèric inferior dim=10	1 8/9/10 1/2 1 1 1	Disparar	Disparar	Si
Límit lletra superior dim=10	1 10 1/2 1 1 1 J	Passa a demanar el numero	Passa a demanar el numero	Si
Límit lletra superior dim=9	1 9 1/2 1 1 1 I	Passa a demanar el numero	Passa a demanar el numero	Si
Límit lletra superior dim=8	1 8 1/2 1 1 1 H	Passa a demanar el numero	Passa a demanar el numero	Si
Límit lletra	1 8/9/10 1 1 1 A	Passa a demanar el	Passa a demanar el	Si

inferior		numero	numero	
Límit numero i lletra inferior	1 8/9/10 1/2 1 1 1 A	Disparar	Disparar	Si
Límit lletra inferior numero superior dim=10	1 10 1/2 1 1 J 10	Disparar	Disparar	Si
Límit lletra inferior numero superior dim=9	1 9 1/2 1 1 I 9	Disparar	Disparar	Si
Límit lletra inferior numero superior dim=8	1 8 1/2 1 1 H 8	Disparar	Disparar	Si
Lletra fora dels límits	1 8 1/2 1 1 K	Es torna a demanar la lletra	Es torna a demanar la lletra	Si
Numero fora dels límits	1 8 1/2 1 1 H 15	Es torna a demanar el número	Es torna a demanar el número	Si

Passa el 94.33% de les proves realitzades d'acord amb les especificacions inicials.