



Projecte Final: Primera Entrega

ALUMNE: Aleix Mariné Tena

PROFESSOR: Carlos García

ASSIGNATURA: Fonaments de Programació I

ENSENYAMENT: Doble titulació Bioteχνologia + Enginyeria Informàtica

GRUP: G3

DATA: 1/11/2015

Índex

1	MENU	2
1.1	ESPECIFICACIONS	¡ERROR! MARCADOR NO DEFINIDO.
1.2	ANÀLISI DEL PROBLEMA PLANTEJAT	¡ERROR! MARCADOR NO DEFINIDO.
1.3	DISSENY I PSEUDOCODI	2
1.4	AVALUACIÓ I JOC DE PROVES	4
2	FITXER DE TEXT 1	4
2.1	ESPECIFICACIONS	4
2.2	ANÀLISI DEL PROBLEMA PLANTEJAT	4
2.3	DISSENY I PSEUDOCODI	5
2.4	AVALUACIÓ I JOC DE PROVES	7
3	FITXER DE TEXT 2	8
3.1	ESPECIFICACIONS	8
3.2	ANÀLISI DEL PROBLEMA PLANTEJAT	9
3.3	DISSENY I PSEUDOCODI	9
3.4	AVALUACIÓ I JOC DE PROVES	10
4	CALCULAR PUNTUACIÓ	10
4.1	ESPECIFICACIONS	10
4.2	ANÀLISI DEL PROBLEMA PLANTEJAT	11
4.3	DISSENY I PSEUDOCODI	11
4.4	AVALUACIÓ I JOC DE PROVES	12
5	DISSENY GENERAL	12

1.- Menú

1.1.- Especificacions

Crear un programa que mostri el menú amb les opcions de joc indicades (1.

Crear un nou joc, 2. Carregar un joc...) i demani a l'usuari que en triï una. Una vegada l'usuari ha triat una opció, es mostrarà un missatge amb l'opció triada i es tornarà a mostrar el menú. El programa ha de controlar el desenvolupament lògic del joc: no es mostrarà l'opció jugar o emmagatzemar un joc si prèviament no s'ha triat l'opció de crear o carregar un joc... El programa finalitza quan es tria l'opció de Sortir.

1.2.- Anàlisi del problema plantejat

Es tracta bàsicament d'imprimir un menú a la pantalla amb més o menys opcions segons les opcions elegides per l'usuari.

1.3.- Disseny i pseudocodi

S'ha decidit utilitzar un menú d'opcions, on el nombre d'opcions mostrades depenen d'un flag booleà que s'activa si s'escolleix l'opció adequada.

Pseudocodi:

```
algorisme menu es
var
menu=0:enter
load=fals: booleà;
fvar
inici
menu:
    escriure("\nMenu:\n");
    escriure("1.- Crear un nou joc\n");
    escriure("2.- Carregar un joc emmagatzemat\n");
    si (load=cert)
        escriure("3.- Jugar partida\n");
        escriure("4.- Emmagatzemar el joc\n");
    fsi
    escriure("5.- Veure podium\n");
    escriure("6.- Sortir del joc\n\n");
    escriure("Selecciona la opcio i prem intro\n\n");
```

```

        llegirf(menu);
        si ((menu>6) o (menu<1) o (menu=3 i load=0) o (menu=4 i
load=0)) llavors
            escriure("\nOpcio incorrecta, prem qualsevol tecla i
torna-ho a intentar\n");
            pausa();
            Netejar_pantalla();
            anar_a menu;
        fsi
        opcio (menu)
        valor1:
            load:=cert;
            Netejar_pantalla();
            anar_a menu;
        valor 2:
            load=true;
            Netejar_pantalla();
            escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");
            pausa();
            Netejar_pantalla();
            anar_a menu;
        valor 3:
            Netejar_pantalla();
            escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");
            pausa();
            Netejar_pantalla();
            anar_a menu;
        valor 4:
            Netejar_pantalla();
            escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");
            pausa();
            Netejar_pantalla();
            anar_a menu;

```

```

        valor 5:
            Netejar_pantalla();
            escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");
            pausa();
            Netejar_pantalla();
            anar_a_menu;
valor 6:
    falgorisme
fopció
falgorisme

```

1.4.- Avaluació i joc de proves

S'han fet proves per a cada opció i totes funcionin. En el cas de que s'escrigui un nombre del qual no hi ha opció salta un missatge d'error i aquest es reinicia per a començar de nou la rutina de l'algorisme. S'han anat comprovant casos i tots donaven el resultat esperat.

2.- Fitxer de text1

2.1.- Especificacions

El programa ha de permetre llegir, escriure i actualitzar un fitxer de text. El fitxer de text conté la informació dels deu millors rècords, segons el següent format: nom\t rècord\n L'usuari podrà triar entre les següents opcions:

- *Llegir fitxer.*

Llegir el contingut del fitxer i escriure la informació obtinguda per pantalla.

- *Afegir rècords al fitxer.*

Es demanarà a l'usuari el nom del jugador (taula de caràcters) i el rècord (enter). El nou rècord s'afegirà al fitxer.

- *Sortir.*

Sortir del programa.

2.2.- Anàlisi del problema plantejat

Es tracta de donar-li l'opció a l'usuari de llegir o escriure en un fitxer d'emmagatzematge de records.

2.3.- Disseny i pseudocodi

S'ha decidit utilitzar un menú d'opcions on en una opció llegeix el fitxer emmagatzemant caràcter per caràcter el text del fitxer en una taula per a després imprimir-lo en pantalla.

Pseudocodi

```
algorisme records es
var
i:=0,e, record: enter;
llegir, joc[100], cosa: character;
records:arxiu;
fvar
inici:
e:=0
netejar_pantalla();
escriure("Escull una opcio i prem intro \n\n");
escriure("1.- Llegir fitxer de puntuacions\n");
escriure("2.- Escriure sobre el fitxer de puntuacions\n");
escriure("3.- Sortir\n");
llegir(i);
si ((i<1) o (i>3)) llavors
    netejar:pantalla();
    escriure("Opció incorrecta, prem qualsevol tecla i torna-ho a
intentar");
    pausa();
    reiniciar();
fsi
opcio(i)
opciol
    netejar:pantalla();
    records= obrir("records.txt","r");
    si (records==NULL) llavors
        tancar(records);
        records =obrir("records.txt","w");
    fsi
    escriure("||||||||||||||||\n");
escriure("|HALL DE LA FAMA|\n");
```

```

escriure("||||||||||||||||\n");
    mentre (nofinalarxiu(records))
        llegir = llegirf(records);
        escriure(" ",llegir);
        si (llegir=32) llavors

            e:=e+1;

        fsi
    si ((llegir=32) i (e%2==0))

        escriure('\n');

    fsi
fmentre
    escriure("\n\n Prem qualsevol tecla per tornar al menu...");
    pausa();
    tancar(records);
    reiniciar();
opcio 2:
    i:=0;

    cosa:=0;
    records= obrir("records.txt", "a");
    escriure("Introdueix el nom :\n");
    llegir(cosa);
    llegir(cosa);
    mentre(cosa!='\n')
        joc[i]=cosa;
        i++;
        llegir(cosa);
    fmentre
    joc[i]='\0';
    i=0;
    mentre(joc[i]!='\0')
        escriuref(records,joc[i]);

```

```

        i++;
    fmentre
    escriure("Introdueix el record: ");
    llegir(record);
    fescriture(records,"    Record: ' ' \n", record);
    tancar(records);

opcio3:
    falgorisme
fopcio
falgorisme

```

2.4.- Avaluació i joc de proves

S'ha anat comprovant els diferents casos possibles. El programa està dissenyat per a grabar un nom de rècord de fins a cent caràcters, per tant no n'admetrà més. En cas de que el fitxer no existeixi es crearà automàticament. Si el fitxer està buit, no mostrarà cap dada simplement. En escollir una opció que no està a la llista el programa mostra un missatge d'error i reinicia el programa. S'han anat comprovant diversos valors i amb tots s'ha obtingut el resultat esperat.

Pseudocodi:

```

algorisme fitxertext2 es
var
    taula[12], i, temp, opcio: enter;
    f: fitxer;
fvar
    escriure("Tenim un fitxer amb unes dimensions, un numero de
jugadors i una taula:\n");
    escriure(" 1.- Llegir el fitxer \n 2.- Escriure en el
fitxer\n 3.- Sortir del programa\n");
    escriure(" Escull una opcio i apreta intro: \t");
    llegir(opcio);
    opcio(opcio)
    valor 1:
        f = fopen("Fitxer.txt","r");
        i=0;
        mentre(i<129) fer

```



```

        fllegir(f,temp);
        escriure(" ",temp);
        i++;
    fmentre
        tancar(f);
valor 2:
        escriure("Introdueix cada valor confirmant amb intro
fins a 12 valors");
        f = fopen("Fitxer.txt","w");
        i=0;
        mentre(i<12) fer
            llegir(temp);
            taula[i]=temp;
            fescriure(f, taula[i]);
            i++;
        fmentre
            escriure("Finalitzat");
            pausa();
            tancar(f);
        falgorisme
valor 3:
        falgorisme
fopcio
        Altrevalor: {escriure( "opcio no valida, sortint del joc...");

        falgorisme

```

3.- Fitxer de text 2

3.1.- Especificacions

El programa ha de permetre llegir i escriure un fitxer de text. El fitxer de text conté la següent informació:

dimensions(enter) num_jugadors(enter) número número (fins a 10 enters) L'usuari podrà triar entre les següents opcions:

- Llegir fitxer.

Llegir el contingut del fitxer i escriure a pantalla: una dada per fila. Els 10 números s'emmagatzemaran a una taula d'enters.

- Escriure fitxer.

Escriure el fitxer amb la informació. La informació es trobarà a una taula d'enters.

- Sortir.

Sortir del programa.

3.2.- Anàlisi del problema plantejat

Es tracta de donar-li l'opció a l'usuari de llegir o escriure en un fitxer d'emmagatzematge de configuració.

3.3.- Disseny i pseudocodi

S'ha decidit utilitzar un menú d'opcions on en una opció llegeix el fitxer emmagatzemant caràcter per caràcter el text del fitxer en una taula per a després imprimir-lo en pantalla.

algorisme fitxertext2 es

var

taula[12], i, temp, opcio: enter;

f: fitxer;

fvar

escriure("Tenim un fitxer amb unes dimensions, un numero de jugadors i una taula:\n");

escriure(" 1.- Llegir el fitxer \n 2.- Escriure en el fitxer\n 3.- Sortir del programa\n");

escriure(" Escull una opcio i apreta intro: \t");

llegir("%i",&opcio);

opcio(opcio)

valor 1:

f = fopen("Fitxer.txt","r");

i=0;

mentre(i<129 fer

fllegir(f,temp);

escriure(" ",temp);

i++;

fmentre

tancar(f);

valor 2:

escriure("Introdueix cada valor confirmant amb intro fins a 12 valors");

```

f = fopen("Fitxer.txt","w");
i=0;
mentre(i<12) fer
    llegir("%i",&temp);
    taula[i]=temp;
    fescrivre(f,"%d ",taula[i]);
    i++;
fmentre
    escriure("Finalitzat");
    pausa();
    tancar(f);
    falgorisme
valor 3:
falgorisme
fopcio
    Altrevalor: {escriure( "opcio no valida, sortint del joc...");
    Falgorisme

```

3.4.- Avaluació i joc de proves

S'ha anat comprovant els diferents casos possibles. El programa està dissenyat per a treballar amb 12 valors numèrics, per tant no n'admetrà més. En cas de que el fitxer no existeixi es crearà automàticament. Si el fitxer està buit, no mostrarà cap dada simplement. En escollir una opció que no està a la llista el programa mostra un missatge d'error i reinicia el programa.

4.- Calcular puntuació

4.1.- Especificacions

El programa ha d'implementar l'equació 1, que serveix per a calcular la puntuació d'un jugador. Les dades s'introdueixen per teclat. En primer lloc es demanarà la dimensió del tauler. A continuació, l'usuari introduirà els resultats dels llançaments: 0 (repetit), 1 (aigua), 2 (tocat), 3 (enfonsat). Amb cada nou resultat s'incrementarà el nombre total de llançaments. El càlcul finalitza quan l'usuari introdueix el sentinella. El resultat final de la puntuació ha de ser un valor enter.

4.2.- Anàlisi del problema plantejat

Es tracta de calcular una puntuació mitjançant l'equació matemàtica (1). Els valors de dimensió del tauler i de resultat es demanaran a l'usuari per teclat.

(1)

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

4.3.- Disseny i pseudocodi

Hi ha una fase d'entrada de dades on el programa obté les dades necessàries per a poder calcular la puntuació, després hi ha una fase de càlcul en el que s'implementa l'equació 1 i després una fase de sortida en que s'imprimeixen els resultats per pantalla.

Algorisme:

```
var
    ida, num, acum, k, p, puntuacio: enters;
fvar
inici
    k:=1;
    p:=0;
    num:=1;
    escriure("Aquest programa calcula la puntuació\n\n");
    escriure("Introdueix la mida del tauler: ");
    llegir(mida);
    netejar_pantalla();
    escriure("Ara introdueix els valors del resultat dels
    llençaments\n");
    escriure("Aquests resultats poden ser 3, 2, 1 o 0\n");
    escriure("Sent tocat i enfonsat, tocat, aigua o posicio
    repetida respectivament\n");
    escriure("Introdueix cada valor amb intro afegint -1 per
    acabar: ");
    mentre (num!==-1) fer
        k:=num-1;
        acum:=acum+k;
        llegir(num);
        p:=p+1
```

```

fmentre
p:=p-1;
si (p=0) llavors
    escriure("No has introduït cap data, torna-ho a provar");
    reiniciar();
fsi
puntuacio:=(100*(mida/p)*acum);
escriure("La puntuacio es ",puntuacio);
pausa();
falgorisme

```

4.4.- Avaluació i joc de proves

He considerat que els valors introduïts per l'usuari seran correctes i vàlids i per això no hi ha cap part del programa que verifiqui que les dades introduïdes són correctes. Tot i així un dels possibles casos problemàtics és que dividim entre 0, però ja he fet que el programa tingui en compte que quan això passa ha de reiniciar l'algorisme ja que es tracta d'un error. He comprovat diversos casos i tots donen el valor esperat.

5.- Disseny general

La idea general es crear un programa que vagi cridant a diferents funcions i accions per tal de posar en marxa tot allò que el joc necessiti per a funcionar com a tal. Algunes de les parts que s'entreguen en aquest informe ja podrien afegir-se al programa com una acció o funció independent i funcionar. Inclueixo el disseny en pseudocodi de tot el joc per a que es

```

algorisme menu es
var
menu=0:enter
load=fals: booleà;
fvar
inici
menu:
    escriure("\nMenu:\n");
    escriure("1.- Crear un nou joc\n");
    escriure("2.- Carregar un joc emmagatzemat\n");

```

```

si (load=cert)
    escriure("3.- Jugar partida\n");
    escriure("4.- Emmagatzemar el joc\n");
fsi
escriure("5.- Veure podium\n");
escriure("6.- Sortir del joc\n\n");
escriure("Selecciona la opcio i prem intro\n\n");
llegirf(menu);
si ((menu>6)o (menu<1) o (menu=3 i load=0) o (menu=4 i
load=0)) llavors
    escriure("\nOpcio incorrecta, prem qualsevol tecla i
torna-ho a intentar\n");
    pausa();
    Netejar_pantalla();
    anar_a menu;
fsi
opcio (menu)
valor1:
    $aquí aniran totes les funcions per a crear la matriu
del joc i tot el que faci falta per a començar a jugar
    Inicia_taulell(); $accions per a crear totes les dades
per iniciar el joc
    ...
    load:=cert;
    Netejar_pantalla();
    anar_a menu;
valor 2:
    $aquí carregarem una partida
    $Consistirà en llegir variables des d'una font
externa per a recuperar l'estat d'un joc en un moment determinat
    Carregar_jocs(); $acció amb la que carregui l'estat
d'un joc anterior
    ...
    load=true;
    Netejar_pantalla();

```

```

        escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");

        pausa();

        Netejar_pantalla();

        anar_a menu;

valor3:

        dispara();$aquí iniciariem el vertader cos del
programa, on estarien la majoria de funcions que donen la
intel·ligència artificial al joc

        Netejar_pantalla();

        escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");

        pausa();

        Netejar_pantalla();

        anar_a menu;

valor4:

        $aquí guardariem les variables d'estat del joc en una
font externa que podriem carregar en l'apartat 2

        Netejar_pantalla();

        escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");

        pausa();

        Netejar_pantalla();

        anar_a menu;

valor 5:

        $aquí llegiriem el fitxer de podium

        $En la fase dos integrarem aquesta part del menú amb
l'exercici del fitxer de text de la fase 1

        $Com un programa independent

        Enmagatzema_records();

        Recupera_records(); $Principals funcions d'entrada sortida
d'aquest apartat

        Netejar_pantalla();

        escriure("\nPrem qualsevol tecla per a tornar al menu
principal \n");

        pausa();

        Netejar_pantalla();

```

```
        anar_a menu;  
valor6:  
        $sortir  
        falgorisme  
fopció  
falgorisme
```