

# **Programació Orientada a Objecte**

## **Pràctica 1**

Miquel Rodríguez Juvany

Daniel Ruano Villalba

Grup 102

2021-22

## Índex

Participació.....	2
Sessió 1 JocVida.java.....	3
Sessió 2 Taulell.java.....	6
Sessió 2 JocVida.java.....	8

## Participació

A partir del programa ja existent de la última pràctica de Fonaments de Programació, hem fet l'anàlisi del nou repte i hem detectat les diferents parts que havíem de canviar del codi original.

Durant la primera sessió a classe vam treballar junts llegint l'enunciat amb els diferents exercicis i buscant com resoldre'ls adequadament. Al principi ens va resultar fàcil i coincidíem en gairebé tot amb els coneixements de Fonaments de Programació del Daniel i l'experiència prèvia amb C++ del Miquel.

Vam trobar problemes per canviar que el programa no treballés amb coordenades quadrades perquè el codi inicialment es va plantejar perquè funcionés d'aquesta manera i vam trobar problemes amb moltes de les parts en les que s'utilitzaven les coordenades però amb ajuda de la professora vam poder aprendre que fèiem línies de codi molt més complicades del que haurien de ser.

Un cop finalitzada la primera sessió vam acabar la pràctica cadascun pel seu compte i després vam posar en comú el que havíem fet per separat vam triar les parts de codi que ens agradaven més de les dues parts.

La segona sessió vam optar per fer-la separats des del principi perquè tractava sobre els nous conceptes introduïts a les classes de Programació Orientada a Objecte i ja teníem un codi sobre el Joc de la Vida en una sola classe. Quan un dels dos tenia algun dubte ens el resolíem i vam tenir alguns problemes amb algunes funcions que no feien el que havien de fer perquè ens faltaven línies de codi i vam estar aturats força temps però finalment ho hem pogut tirar endavant gràcies al treball en equip ja que les mancances d'un eren les forteses de l'altre.

# Sessió 1

## JocVida.java

```
package Practica1Sessio1;
import Keyboard.*;
import java.util.Random;

public class JocVida {

    private static int coincidencies(int buscar, int[] taula) {
        int trobats=0;
        for(int i=0; i<taula.length; i++) {
            if(buscar==taula[i]) trobats++;
        }

        return trobats;
    }

    private static boolean extincio(boolean[][] taulell) {
        for(int y=0;y<taulell.length; y++) {
            for(int x=0; x<taulell[y].length; x++) {
                if (taulell[y][x]) return false;
            }
        }
        return true;
    }

    private static void imprimirTaulell(boolean[][] taulell) {
        for(int y=0;y<taulell.length; y++) {
            for(int x=0; x<taulell[y].length; x++) {
                if (taulell[y][x]) System.out.print('■');
                else System.out.print('□');

                System.out.print(' ');
            }
            System.out.println("");
        }
    }

    private static int quantsVeins(boolean[][] taulell, int coordenades1, int coordenades2)
    {
        // coordenades: [0]=X , [1]=Y
        int veins=0;
        for(int y=-1; y<=1; y++) {
            // comprovar que Y estigui dintre de la array
            if (coordenades1+y>=0 && (coordenades1+y)<taulell.length) {
                for(int x=-1; x<=1; x++) {
                    if ( !(x==0 && y==0) ) {
                        // comprovar que X estigui dintre de la array
                        if (coordenades2+x>=0 &&
                            (coordenades2+x)<taulell[0].length) {

                            if
                                (taulell[coordenades1+y][coordenades2+x]) veins++;
                        }
                    }
                }
            }
        }

        return veins;
    }

    private static int updateAux(int veins,int coordenades1, int coordenades2, boolean
casella, boolean[][] aux) {
        if (veins==2) aux[coordenades1][coordenades2]=casella;
        else if (veins==3) aux[coordenades1][coordenades2]=true;
        // death = default

        if (aux[coordenades1][coordenades2])
    }
```

```

        return 1;

    return 0;
}

private static int demanarInt(String msg, int min, int max) {
    int triat=min-1;
    while(triat>max || triat<min) {
        System.out.print(msg+"["+min+", "+max+"]: ");
        triat=Keyboard.readInt();
    }

    return triat;
}

public static void main(String args[]) {

    boolean acabar = false;
    int partides=1;

    int dimensio1=demanarInt("Indica la dimensio X del taulell",5,15);
    int dimensio2=demanarInt("Indica la dimensio Y del taulell",5,15);
    System.out.println("");
    int organismes=demanarInt("Indica el num. d'organismes",1,dimensio1*dimensio2);

    while (!acabar) {

        int rand_x;
        int rand_y;

        Random r = new Random();

        rand_x = r.nextInt(dimensio1);
        rand_y = r.nextInt(dimensio2);

        boolean joc[][]=new boolean[dimensio2][dimensio1];
        for(int i=0; i<organismes; i++) {

            while(joc[rand_y][rand_x]) {
                rand_x = r.nextInt(dimensio1);
                rand_y = r.nextInt(dimensio2);
            }

            System.out.println("Organisme "+(i+1)+": "+rand_x+", "+rand_y);
            joc[rand_y][rand_x]=true;
        }

        System.out.println("\nTaulell inicial: ");
        imprimirTaulell(joc);
        System.out.println("");

        int generacions_totals=demanarInt("Indica el num. de generacions",1,10);

        int[] resums_generacions=new int[generacions_totals];
        int gen=0;
        while(gen<generacions_totals && !extincio(joc)) {
            int organismes_vius=0;
            boolean aux[][]=new boolean[dimensio2][dimensio1];
            for(int y=0;y<joc.length; y++) {
                for(int x=0; x<joc[y].length; x++) {

                    int veins=quantsVeins(joc, y, x);
                    boolean casella=joc[y][x];

                    organismes_vius+=updateAux(veins, y, x,casella,aux);
                }
            }
            joc=aux;

            System.out.println("-----");
            System.out.println("Gen: "+(gen+1)+"\n");
        }
    }
}

```

```

        imprimirTaulell(joc);

        System.out.println("");

        resums_generacions[gen]=organismes_vius;
        gen++;
    }

    if (extincio(joc)) {
        System.out.println("Tots els organismes han mort.");
        System.out.print("Vols tornar a jugar? (S/N): ");
        char decisio = Keyboard.readChar();
        System.out.println("");
        if (decisio == 'N' || decisio == 'n') {
            System.out.print("Has jugat " + partides + " partides");
            acabar = true;
        }
        else {
            partides++;
        }
    } else {
        for(int i=0; i<generacions_totals; i++) {
            System.out.println("Gen. " +(i+1)+": " +resums_generacions[i]+
organismes_vius.");
        }

        int cnt=0;
        boolean found=false;

        while(cnt==0 && !found) {
            int search=resums_generacions[cnt];

            int nums=0;
            for(int u=0; u<generacions_totals; u++) {
                if (search==resums_generacions[u])
                    nums++;

                if(nums>2)
                    found=true;
            }
            cnt++;
        }
        if (found) {
            System.out.println("Hi ha mes d'una repeticio. Com per exemple
la gen. " +cnt);
        }

        System.out.println("Vols tornar a jugar? (S/N): ");
        char decisio = Keyboard.readChar();
        System.out.println("");
        if (decisio == 'N' || decisio == 'n') {
            System.out.print("Has jugat " + partides + " partides");
            acabar = true;
        }
        else {
            partides++;
        }
    }
}

}

}

```

## Sessió 2

### Taulell.java

```
package Practica1Sessio2;
import java.util.Random;

public class Taulell {

    private boolean [][] taulell;
    private int files;
    private int columnes;
    private int vius;

    public Taulell (int files, int columnes) {
        this.files = files;
        this.columnes = columnes;
        this.taulell = new boolean [files][columnes];
        treureVida();
    }

    public void treureVida() {
        for(int i=0; i<this.files; i++) {
            for(int j=0; j<this.columnes; j++) {
                this.taulell[i][j] = false;
            }
        }
        this.vius=0;
    }

    public void posarVida(int N) {
        Random r = new Random();
        for (int i=0; i<N; i++) {
            int rand_fil, rand_col;
            do {
                rand_fil = r.nextInt(taulell.length);
                rand_col = r.nextInt(taulell[0].length);
            } while (taulell[rand_fil][rand_col] == true);

            System.out.println("Organisme " + (i+1) + " a la posició. : " + rand_fil
+ ", " + rand_col);
            this.taulell[rand_fil][rand_col] = true;
        }
    }

    public void mostraTaulell() {
        for(int x=0; x<this.files; x++) {
            for(int y=0; y<this.columnes; y++) {
                if (taulell[x][y]) {
                    System.out.print('■');
                }
                else {
                    System.out.print('□');
                }
                System.out.print(' ');
            }
            System.out.println("");
        }
    }

    public boolean ferGeneracio() {
        this.vius=0;
        boolean taulellAux[][];
        boolean diferent = false;
        taulellAux = copiar(this.taulell);
        for(int i=0; i<taulellAux.length; i++) {
            for(int j=0; j<taulellAux[0].length; j++) {
                if(taulellAux[i][j] == false &&
quantesVeines(i,j) == 3) {
                    taulellAux[i][j] = true;
                } else if(taulellAux[i][j] == true) {
                    if(quantesVeines(i,j) >= 4) {

```

```

        taulellAux[i][j] = false;
        this.vius--;
    } else if(quantasVeines(i,j) <= 1) {
        taulellAux[i][j] = false;
        this.vius--;
    }
    if(taulellAux[i][j] != this.taulell[i][j]) {
        diferent = true;
    }
    if(taulellAux[i][j] == true) {
        this.vius++;
    }
}
}
this.taulell = copiar(taulellAux);
if(diferent == true) {
    return true;
} else {
    return false;
}
}

private int quantasVeines(int fil, int col) {
    int veins=0;
    for(int x=-1; x<=1; x++) {
        if (fil+x>=0 && (fil+x)<this.taulell.length) {
            for(int y=-1; y<=1; y++) {
                if ( !(y==0 && x==0) ) {
                    if (col+y>=0 && (col+y)<this.taulell[0].length)
                        if (taulell[fil+x][col+y]) veins++;
                }
            }
        }
    }
    return veins;
}

private boolean[][] copiar(boolean [][] origen) {
    boolean [][] taulellAux;
    taulellAux = origen;
    for(int i=0; i<this.files; i++) {
        for(int j=0; j<this.columnes; j++) {
            taulellAux[i][j] = origen[i][j];
        }
    }
    return taulellAux;
}

public boolean[][] getTaulell() {return this.taulell;}

public int getFiles() {return this.files;}

public int getColumnes() {return this.columnes;}

public int getVius() {return this.vius;}
}

```



## JocVida.java

```
package Practica1Sessio2;
import Keyboard.*;

public class JocVida {

    public static void main(String args[]) {
        Taulell t;

        int files;
        int columnes;
        int orgVius;

        System.out.println("                JOC DE LA VIDA");
        System.out.println("Miquel Rodríguez Juvany i Daniel Ruano Villalba");
        System.out.println();

        do {
            System.out.print("Quantes files tindra el taulell? [5, 15]: ");
            files = Keyboard.readInt();
        } while (files<5 || files>15);

        do {
            System.out.print("Quantes columnes tindra el taulell? [5, 15]: ");
            columnes = Keyboard.readInt();
        } while (columnes<5 || columnes>15);

        t = new Taulell(files, columnes);

        do {
            System.out.print("Quants organismes vius tindra el taulell? [1, " +
(t.getFiles()*t.getColumnes()) + "]: ");
            orgVius = Keyboard.readInt();
        } while (orgVius<1 || orgVius>t.getFiles()*t.getColumnes());
        System.out.println();

        t.posarVida(orgVius);

        System.out.println();
        System.out.println("Taulell inicial:");
        t.mostraTaulell();
        System.out.println();

        System.out.print("Quantes generacions vols fer? [1, 10]: ");
        int gen = Keyboard.readInt();
        System.out.println();

        for(int i=0; i<gen; i++)
        {
            t.ferGeneracio();
            System.out.println("Generació: " + (i+1));
            t.mostraTaulell();
            System.out.println("");
        }

        System.out.println("Numero d'organismes vius: " + t.getVius());

    }
}
```