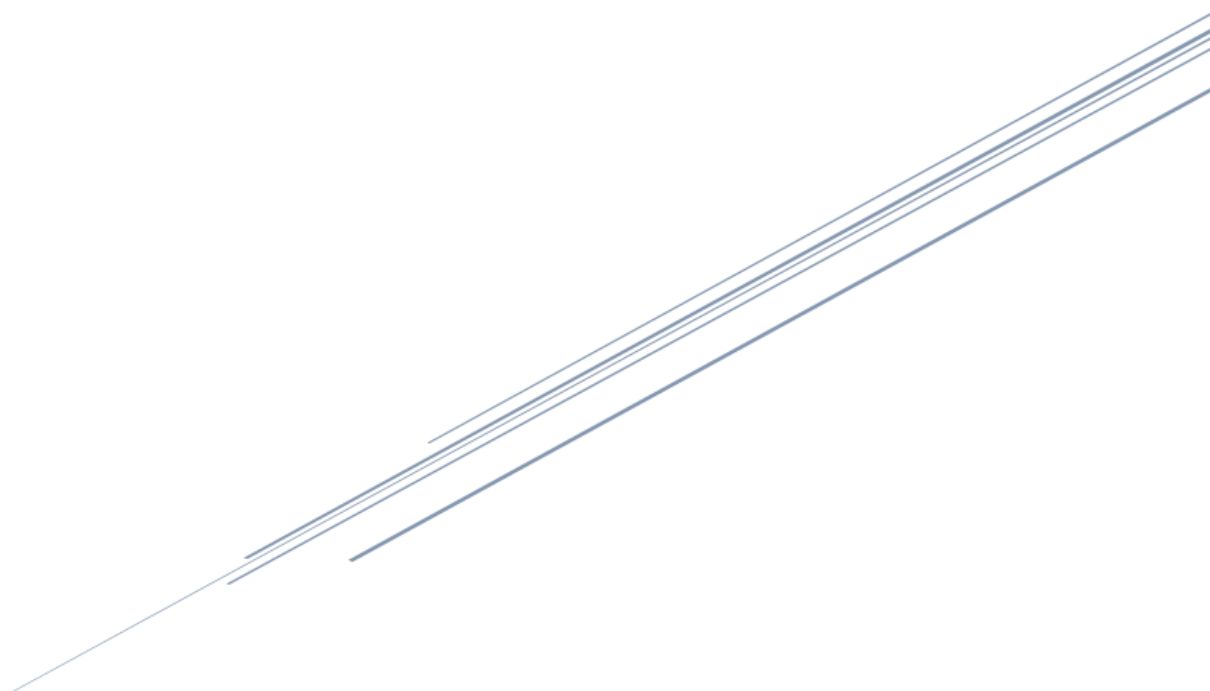


Pràctica 4



Rems Nalivaiko

Pol Rubio

23 de mar. de 2022

103112 - PROGRAMACIÓ ORIENTADA OBJECTE

Fundació TecnoCampus Mataró-Maresme

Índex:

Sessió 1:	3
Organització de la feina:	11
Valoració:	11

Sessió 1:

Es tracta de fer un joc de la vida amb una interfície gràfica:

Codi:

JocVida.java

```
package Sessio1;

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.util.Random;

public class JocVida extends JFrame implements ActionListener {

    private Casella[][] taulell;
    private int generacio=0;
    private int
generacionsTotals,files,columnes,organismes,organismesTotals;

    // 0=game on
    // 1=game over a causa de les generacions
    // 2=game over a causa de generacions duplicades
    // 3=game over a causa de 0 organismes vius
    private int game_over=0;

    JPanel panel_entrades,panel_taulell;
    JLabel
filesLabel,columnesLabel,generacionsLabel,organismesViusLabel,dataValidityL
abel;

    JTextField numFiles,numColumnes,numGeneracions,numEssersVius;
    JButton submitDades,nextGeneracio;

    public JocVida() {
        setLayout(new BorderLayout());

        panel_entrades=new JPanel();
```

```
panel_entrades.setLayout(new GridLayout(20,1));

panel_taulell=new JPanel();
panel_taulell.setLayout(new GridLayout(12,12));

setBounds(0,0,750,500);
setTitle("JOC de la VIDA!!!");

setDefaultCloseOperation(EXIT_ON_CLOSE);

filesLabel=new JLabel("# de files:");
numFiles=new JTextField(null, 15);

columnesLabel=new JLabel("# de columnes:");
numColumnes=new JTextField(null, 15);

generacionsLabel=new JLabel("# de generacions:");
numGeneracions=new JTextField(null, 15);

organismesViusLabel=new JLabel("# de vius:");
numEssersVius=new JTextField(null, 15);

dataValidityLabel=new JLabel("DADES INCORRECTES");
dataValidityLabel.setVisible(false);

submitDades=new JButton("Entrar dades");
submitDades.addActionListener(this);

nextGeneracio=new JButton("Generacio");
nextGeneracio.addActionListener(this);
nextGeneracio.setEnabled(false);

panel_entrades.add(filesLabel);
panel_entrades.add(numFiles);
panel_entrades.add(new JPanel());
```

```
        panel_entrades.add(columnesLabel);
        panel_entrades.add(numColumnes);
        panel_entrades.add(new JPanel());

        panel_entrades.add(generacionsLabel);
        panel_entrades.add(numGeneracions);
        panel_entrades.add(new JPanel());

        panel_entrades.add(organismesViusLabel);
        panel_entrades.add(numEssersVius);
        panel_entrades.add(new JPanel());

        panel_entrades.add(dataValidityLabel);

        panel_entrades.add(submitDades);
        panel_entrades.add(nextGeneracio);

        panel_entrades.add(new JPanel());
        panel_entrades.add(new JPanel());

        add(panel_entrades, BorderLayout.WEST);
        add(panel_taulell, BorderLayout.CENTER);
        setVisible(true);
    }

    public void actionPerformed(ActionEvent e) {
        if(e.getSource()==submitDades) {
            boolean dades=comprobacioDades();
            nextGeneracio.setEnabled(dades);

            if(dades) {
                dataValidityLabel.setText("Dades correctes");
                setEnabledDades(false);
                panel_taulell.setLayout(new
GridLayout(files,columnes));
                taulell=crearTaulell();

                posarVida(organismesTotals);
            }
        }
    }
}
```

```
        mostraTaulell();  
        textGenUpdate();  
  
        } else setEnabledDades(true);  
        dataValidityLabel.setVisible(true);  
    } else if(e.getSource()==nextGeneracio) {  
  
        mostraTaulell();  
  
        generacio++;  
        textGenUpdate();  
  
        if(generacio==generacionsTotals) game_over=1;  
        else if(!ferGeneracio()) game_over=2;  
        else if(organismes==0) game_over=3;  
  
        if(game_over!=0) {  
            gameOver(game_over);  
        }  
    }  
}  
  
private void setEnabledDades(boolean b) {  
    JTextField[] txt=  
{numFiles,numColumnnes,numGeneracions,numEssersVius};  
    for(JTextField elem:txt) elem.setEnabled(b);  
    submitDades.setEnabled(b);  
}  
  
private boolean comprobacioDades() {  
    try{  
        files=Integer.parseInt(numFiles.getText());  
        columnnes=Integer.parseInt(numColumnnes.getText());  
        generacionsTotals=Integer.parseInt(numGeneracions.getText());  
        organismesTotals=Integer.parseInt(numEssersVius.getText());  
  
        if(organismesTotals>files*columnnes || organismesTotals<1 ||  
generacionsTotals<1 || files<1 || columnnes<1)
```

```
        throw new Exception();
    } catch(Exception exc) {
        return false;
    }
    return true;
}

private Casella[][] crearTaulell(){
    Casella[][] t= new Casella[files][columnes];
    for(int y=0;y<files; y++){
        for(int x=0;x<columnes; x++){
            t[y][x]=new Casella();
        }
    }
    return t;
}

private void gameOver(int game_over) {
    nextGeneracio.setEnabled(false);
    String tmp="";
    switch(game_over) {
        case 1:
            tmp="Generacio";
            break;
        case 2:
            tmp="Generacio duplicada";
            break;
        case 3:
            tmp="Tots morts";
            break;
    }
    nextGeneracio.setText(tmp);
    setTitle("JOC de la VIDA. Fi.");
}

private void textGenUpdate() {
    nextGeneracio.setText("Generacio "+(generacio+1));
    setTitle("JOC de la VIDA!!!. Fes la generacio "+(generacio+1));
}

private void posarVida(int N) {
    for(int i=0; i<N; i++) {
        int x,y;
```

```
        do {
            Random rand=new Random();
            y=rand.nextInt(files); x=rand.nextInt(columnes);
        } while (taulell[y][x].getEstat());
        taulell[y][x].setEstat(true);
    }
}

private void mostraTaulell() {
    for (int y = 0; y<files; y++) {
        for (int x = 0; x<columnes; x++) {
            panel_taulell.add(taulell[y][x], y,x);
        }
    }
}

private boolean ferGeneracio() {
    Casella[][] aux=crearTaulell();
    int igualtats=0;

    organismes=0;

    for (int y=0; y<taulell.length; y++) {
        for (int x=0; x<taulell[0].length; x++) {
            int num_veines=quantasVeines(y,x);
            if (num_veines==3) aux[y][x].setEstat(true);
            else if (num_veines==2)
aux[y][x].setEstat(taulell[y][x].getEstat());
            // per defecte ja estan mortes.

            if(taulell[y][x].getEstat()==aux[y][x].getEstat())
igualtats++;

            if(aux[y][x].getEstat()) organismes++;

        }
    }
    copiar(aux);

    return( igualtats!=files*columnes );
}
```



```
private int quantesVeines(int fil, int col) {
    int veines=0;

    for(int y=fil-1; y<(fil+2); y++) {
        if(y>=0 && y<taulell.length) {
            for(int x=col-1; x<(col+2); x++) {
                if(x>=0 && x<taulell[0].length) {
                    if (taulell[y][x].getEstat() && !(x==col &&
y==fil)) veines++;
                }
            }
        }
    }

    return veines;
}

private void copiar(Casella[][] origen) {
    for (int y = 0; y<origen.length; y++) {
        for (int x = 0; x<origen[y].length; x++) {
            taulell[y][x].setEstat(origen[y][x].getEstat());
        }
    }
}

public static void main(String[] args) {
    new JocVida();
}
}
```

Casella.java

```
package Sessio1;

import javax.swing.*;
import java.awt.*;

public class Casella extends JPanel{
```

```
// Aquests dos atributs no calen, es per simplement per comoditat.  
// Es podrien canviar per els seus valors sense cap problema.  
Color colorMort=Color.white;  
Color colorViu=Color.black;  
  
public Casella(){  
    setBackground(colorMort);  
    setBorder(BorderFactory.createLineBorder(Color.black));  
}  
  
public boolean getEstat(){ return !(getBackground().equals(colorMort));  
} // 1=viu;0=mort;  
public void setEstat(boolean val) { setBackground( val ? colorViu :  
colorMort); }  
public void alternarEstat(){  
    if(getEstat()) setBackground(colorMort);  
    else setBackground(colorViu);  
}  
}
```

Organització de la feina:

En aquesta pràctica n'hem dividit les diferents classes en cada sessió per tal d'anar fent-ho paral·lelament amb l'objectiu d'acabar més de pressa, una vegada acabàvem les nostres classes, les posàvem en comú per tal d'entendre que havia fet l'altre i detectar possibles errors o afegir possibles optimitzacions.

Valoració:

Pel que fa a la valoració del funcionament dels exercicis desenvolupats en aquesta primera pràctica, val a dir que no hem tingut cap problema greu per a resoldre'ls.