**Programació Orientada a Objecte**

**Pràctica 4**

Miquel Rodríguez Juvany

Eloi Egea

Grup 101

2021-22

**Índex**

[**Participació** 2](#_Toc98880779)

[**Casella.java** 3](#_Toc98880780)

[**JocVida.java** 4](#_Toc98880781)

# **Participació**

Hem tingut bastants problemes. Al principi no sabíem com implementar el joc de la vida original perquè funcionés amb la interfície gràfica de Java. Ens ha costat força la distribució dels dos panells (un a l’esquerra i l’altre a la dreta) i fer el GridLayout. La implementació dels mètodes ha estat difícil i n’hem fet molts que finalment no han funcionat i hem hagut d’eliminar. Ens ha costat treballar amb una taula de boolean i una altre de Casella.

El programa funcionava perfectament però ens faltaven algunes funcionalitats estètiques del tipus que al botó de Generació es mostrava la generació però la última generació no s’arribava a mostrar i al intentar arreglar-ho, hem modificat algunes línies que han fet que el Java es pengés al apretar el botó Generació. Hem provat a reiniciar l’Eclipse, el qual no ens ha arreglat res. Després no hem pogut tornar enrere i intentant arreglar el problema, ens ha deixat de funcionar tot.

Quan hem aconseguit arreglar el problema de que la finestra del Java es quedava completament penjada, el programa no deixa de fer generacions (més de les que li demanem), no mostra el nombre de generacions transcorregudes correctament (sempre mostra Generació: 1) i el nom de la finestra de Windows no s’actualitza.

Constantment l’Eclipse ens mostra problemes de NullPointer i moltíssims altres errors (uns 40 errors per execució més o menys).

# **Casella.java**

**package** Sessio1;

**import** javax.swing.\*;

**import** javax.swing.plaf.BorderUIResource.LineBorderUIResource;

**import** java.awt.\*;

**public** **class** casella **extends** JPanel {

**public** casella() {

**super**();

**this**.setMort();

}

**public** **void** setViu() {

**this**.setBackground(Color.***green***);

**this**.setBorder(**new** LineBorderUIResource(Color.***gray***));

}

**public** **void** setMort() {

**this**.setBackground(Color.***black***);

**this**.setBorder(**new** LineBorderUIResource(Color.***gray***));

}

**public** **boolean** getEstat() {

**if** (**this**.getBackground() == Color.***green***) {

**return** **true**; // si el color es verd dons esta viva

} **else**

**return** **false**; // qualsevol altre color esta morta

}

}

# **JocVida.java**

**package** Sessio1;

**import** javax.swing.\*;

**import** java.awt.event.\*;

**import** java.awt.\*;

**import** java.util.\*;

**public** **class** JocVida **extends** JFrame **implements** ActionListener {

**private** JLabel fila, columna, generacio, vius, text;// text

**private** JTextField fil, col, gen, life; // casella per introduir valor

**private** casella[][] Panell, aux; // matriu panells

**private** JButton dades, fgen; // buttons per fer generacio i comencar

**private** **int** nfila, ncol, ngen, nvius, cont=1, contVius; // valors

**private** JPanel ec, centre;// dividir interfaz

**public** JocVida(String nom) {

**super**(nom);

initComponents();

}

**public** **void** initComponents() {

// esquerra = new JPanel();

ec = **new** JPanel();

**this**.setLayout(**new** BorderLayout());

ec.setLayout(**new** GridLayout(20, 1)); // fer una taula amb una columna i 13 files per posar la info

centre = **new** JPanel();

**this**.setLayout(**new** BorderLayout());

// nuemero de files

fila = **new** JLabel("Numero de files");

fil = **new** JTextField();

// nuemro de columnes

columna = **new** JLabel("Numero de columnes");

col = **new** JTextField();

// numero de generacions

generacio = **new** JLabel("Numero de generacions");

gen = **new** JTextField();

// numero de vius

vius = **new** JLabel("Numero d'essers vius inicials");

life = **new** JTextField();

// butons dels final

text = **new** JLabel();

// if (nfila, ncol, ngen, vius==true) llavors visible true i enable true

dades = **new** JButton("Entra dades");

fgen = **new** JButton("Generacio");

fgen.setEnabled(**false**);

ec.add(fila);

ec.add(fil);

ec.add(**new** JLabel());

ec.add(columna);

ec.add(col);

ec.add(**new** JLabel());

ec.add(generacio);

ec.add(gen);

ec.add(**new** JLabel());

ec.add(**new** JLabel());

ec.add(vius);

ec.add(life);

ec.add(**new** JLabel());

ec.add(text);

ec.add(dades);

ec.add(fgen);

dades.addActionListener(**this**);

fgen.addActionListener(**this**);

**this**.getContentPane().add(ec, BorderLayout.***WEST***);

**this**.getContentPane().add(centre, BorderLayout.***CENTER***);

}

**public** **void** actionPerformed(ActionEvent e) {

**if** (e.getSource() == **this**.dades) { // si acciona comprovar si les dades son correctes

nfila = Integer.*parseInt*(fil.getText());

ncol = Integer.*parseInt*(col.getText());

ngen = Integer.*parseInt*(gen.getText());

nvius = Integer.*parseInt*(life.getText());

**if** (nfila > 0 && ncol > 0) {

**if** (ngen <= 10 && ngen > 0) {

**if** (nvius <= (nfila \* ncol) && nvius > 0) {

text.setText("Dades Correctes");

fgen.setEnabled(**true**); // podem fer genreacio

fgen.setText("Generacio: " + **this**.cont);

dades.setEnabled(**false**);

fil.setEnabled(**false**);

col.setEnabled(**false**);

gen.setEnabled(**false**);

life.setEnabled(**false**);

**this**.Panell = **new** casella[nfila][ncol];

**this**.aux = **new** casella[nfila][ncol];

centre.setLayout(**new** GridLayout(nfila, ncol));

**for** (**int** i = 0; i < nfila; i++) {

**for** (**int** j = 0; j < ncol; j++) {

Panell[i][j] = **new** casella();

centre.add(Panell[i][j]);

}

}

posarVida(nvius);

contVius = nvius;

}

}

} **else** {

text.setText("Dades Incorrectes");

}

}

**if** (e.getSource() == **this**.fgen) {

**if** (**this**.cont == **this**.ngen || ferGeneracio() == **true** || contVius == 0) {

fgen.setEnabled(**false**);

cont++;

} **else** {

ferGeneracio();

}

fgen.setText("Generacio: " + cont);

cont++;

}

}

**public** **boolean** ferGeneracio() {

**boolean** diferent = **false**;

**for** (**int** i = 0; i < **this**.Panell.length; i++) {

**for** (**int** j = 0; j < **this**.Panell[i].length; j++) {

**if** (Panell[i][j].getEstat() == **false** && quantesVeines(i, j) == 3) {

contVius++;

**this**.Panell[i][j].setViu();

} **else** **if** ((Panell[i][j].getEstat() == **true** && quantesVeines(i, j) >= 4)

|| (Panell[i][j].getEstat() == **true** && quantesVeines(i, j) <= 1)) {

contVius--;

**this**.Panell[i][j].setMort();

}

}

}

**for** (**int** i = 0; i < **this**.nfila; i++) {

**for** (**int** j = 0; j < **this**.ncol; j++) {

**while** (diferent != **true**) {

**if** (**this**.Panell[i][j].getEstat() == aux[i][j].getEstat()) {

diferent = **false**;

} **else** {

diferent = **true**;

}

}

}

}

cont++;

copiar();

**return** diferent;

}

**public** **int** quantesVeines(**int** fil, **int** col) {

**int** veins = 0;

**for** (**int** i = fil - 1; i <= fil + 1; i++) {

**for** (**int** j = col - 1; j <= col + 1; j++) {

**if** (i >= 0 && i < **this**.nfila && j >= 0 && j < **this**.ncol) {

**if** (Panell[i][j].getEstat()) {

**if** (!(fil == i && col == j))

veins++;

}

}

}

}

**return** veins;

}

**public** **void** posarVida(**int** N) { // n= nombre de vius de la 1r gen

Random random = **new** Random();

**for** (**int** i = 0; i < N; i++) {

**int** fila = random.nextInt(**this**.nfila);

**int** col = random.nextInt(**this**.ncol);

**while** (**this**.Panell[fila][col].getEstat() == **true**) {

fila = random.nextInt(**this**.nfila);

col = random.nextInt(**this**.ncol);

}

**this**.Panell[fila][col].setViu();

}

}

**public** **void** copiar() { // copiar panell i fer tractament amb aux

**for** (**int** i = 0; i < **this**.nfila; i++) {

**for** (**int** j = 0; j < **this**.ncol; j++) {

aux[i][j] = Panell[i][j];

}

}

}

**public** **static** **void** main(String[] args) {

EventQueue.*invokeLater*(**new** Runnable() {

**public** **void** run() {

JFrame jf;

jf = **new** JocVida("Joc Vida");

jf.setBounds(300, 300, 500, 500);

jf.setVisible(**true**);

}

}

);

}

}