

A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

27-3-2023

# Lliurament 4

Several thin, curved lines in dark blue and light grey originate from the bottom left corner and sweep upwards and to the right.

Eloi Egea i Miquel Rodríguez  
TECNOCAMPUS MATARO

## Index

Participació .....	2
JButtonMemory.java .....	3
Memory.java .....	5

## Participació

Hem treballat de manera conjunta, intentant trobar la millor solució als errors fins i tot demanant ajuda a altres companys, desgraciadament no ens hem en sortit amb la pràctica al 100%. Ens ha fallat la selecció i comprovació de les cartes. Al executar el programa ens indica un error: *“Exception in thread “AWT-EventQueue-0” java.lang.NullPointerException: Cannot invoke “java.net.URL.toExternalForm()” because “location” is null”*. Crèiem que no agafava bé la ruta de les imatges però debugant el programa hem pogut observar com n’agafa algunes bé i després el programa deixa de funcionar tot indicant aquesta exception. Hem intentat també que totes les imatges fossin tant del mateix format com de les mateixes dimensions però res ha funcionat.

## JButtonMemory.java

```
package Sessio1;

import javax.swing.JButton;
import javax.swing.ImageIcon;

public class JButtonMemory extends JButton {
    private ImageIcon img;
    private boolean estat;

    private static final ImageIcon imatgeDarrera = new
    ImageIcon(JButtonMemory.class.getResource("darrera.png"));

    public JButtonMemory(ImageIcon quina) {
        super(quina);
        this.img = quina;
        setEstat(false);
    }
    /*
    public void setEstat() {
        this.estat = !this.estat;
        if (this.estat) {
            setIcon(img);
        } else {
            setIcon(imatgeDarrera);
        }
    }

    public void setFinalitzar() {
        this.estat = true;
    }
    */
    public void setEstat(boolean estat) {
        if (this.estat) {
            setIcon(img);
        } else {
            setIcon(imatgeDarrera);
        }
    }
}
```

```
@Override
public boolean equals(Object c) {
    if (c instanceof JButtonMemory) {
        return this.img.equals(((JButtonMemory) c).img);
    }
    return false;
}
}
```

# Memory.java

```
package Sesi01;

import javax.swing.*;

import java.awt.event.*;
import java.awt.*;
import java.util.*;

public class Memory extends JFrame implements ActionListener {

    private JLabel Fila, Col, Datos;
    private JTextField fil, col;
    private JButton datos, reiniciar, mostrar;
    private JButtonMemory[][] tablero;
    private JPanel p1, p2;
    private int f, c, x1, y1, x2, y2;
    private boolean fet = false, click;

    private static final String[] motes = { "pato.png", "cisne.png", "elefante.png",
"gallo.png", "tortuga.png",
        "jaguar.png", "serpiente.png", "ganso.png", "iguana.png",
"mantarraya.png", "hormiga.png", "langosta.png",
        "armadillo.png", "lobo.png", "erizo.png", "polilla.png", "oso.png",
"cocodrilo.png" };

    public Memory(String s) {
        super(s);
        initComponents();
    }

    public void initComponents() {
        this.setLayout(new BorderLayout());
        this.p1 = new JPanel();
        this.p2 = new JPanel();
        this.Fila = new JLabel("Numero de filas");
        this.fil = new JTextField();
        this.Col = new JLabel("Numero de columnas");
        this.col = new JTextField();
        this.Datos = new JLabel("Datos correctos");
        this.datos = new JButton("Entra datos");
    }
}
```

```

        this.reiniciar = new JButton("Reiniciar");
        this.mostrar = new JButton("Mostrar");

        this.p1.setLayout(new GridLayout(20, 1));
        this.p1.add(Fila);
        this.p1.add(fil);
        this.p1.add(Col);
        this.p1.add(col);
        this.p1.add(Datos);
        this.p1.add(datos);
        this.p1.add(new JLabel());
        this.p1.add(new JLabel());
        this.p1.add(new JLabel());
        this.p1.add(new JLabel());
        this.p1.add(new JLabel());
        this.p1.add(new JLabel());
        this.p1.add(reiniciar);
        this.p1.add(mostrar);

        this.Datos.setVisible(false);
        this.mostrar.setEnabled(false);
        this.reiniciar.setEnabled(false);
        getContentPane().add(this.p1, BorderLayout.WEST);
        getContentPane().add(this.p2, BorderLayout.CENTER);
        this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        this.datos.addActionListener(this);
        this.reiniciar.addActionListener(this);
        this.mostrar.addActionListener(this);
    }

    /*
     * private boolean esCorrecte(int f, int c) { if (f < 0 || c < 0) return false;
     * if (f * c > 36 || f * c % 2 != 0) return false; return true; }
     */

    public void actionPerformed(ActionEvent x) {

        if (x.getSource() == this.datos) {
            try {
                this.f = Integer.parseInt(this.fil.getText());

```

```

        this.c = Integer.parseInt(this.col.getText());
    } catch (NumberFormatException e) {
        this.Datos.setText("Datos incorrectos");
        this.Datos.setVisible(true);
    }

    if (this.f > 0 && this.c > 0 && this.f * this.c <= 36 && (this.f *
this.c) % 2 == 0) {

        this.p2.setVisible(true);
        this.Datos.setText("Dades correctes");
        this.Datos.setVisible(true);
        this.mostrar.setEnabled(true);
        this.reiniciar.setEnabled(true);
        this.datos.setEnabled(false);
        this.fil.setEnabled(false);
        this.col.setEnabled(false);

        reset();

        this.tablero = new JButtonMemory[f][c];
        omplir();
        this.fet = true;
        this.click = false;
        this.p2.setLayout(new GridLayout(f, c));
    } else {
        this.Datos.setText("Dades incorrectes");
        this.Datos.setVisible(true);
    }

    /*
    * if(esCorrecte(f,c)) { Datos.setText("Datos correctos");
    * Datos.setVisible(true); mostrar.setEnabled(true);
reiniciar.setEnabled(true);
    * datos.setEnabled(false); fil.setEnabled(false);
col.setEnabled(false); } else
    * { Datos.setText("Datos incorrectos"); Datos.setVisible(true); }
    */
}

if (x.getSource() == this.mostrar) {
    for (int i = 0; i < this.tablero.length; i++) {
        for (int j = 0; j < this.tablero[i].length; i++) {

```



```

        this.tablero[i][j].setEstat(true);
        this.tablero[i][j].setEnabled(false);
    }
}

}

if (x.getSource() == this.reiniciar) {
    // initComponents(); // no se si esta be
    this.Datos.setVisible(false);
    this.mostrar.setEnabled(false);
    this.reiniciar.setEnabled(false);
    this.datos.setEnabled(true);
    this.fil.setEnabled(true);
    this.fil.setText(null);
    this.col.setEnabled(true);
    this.col.setText(null);
    this.p2.removeAll();
    this.p2.setVisible(false);
    this.fet = false;
    this.tablero = null;
}

if (fet == true) {
    for (int i = 0; i < this.tablero.length; i++) {
        for (int j = 0; j < this.tablero[i].length; j++) {
            if (x.getSource() == this.tablero[i][j]) {
                if (x1 != -1 && y1 != -1) {
                    if (click) {

this.tablero[x1][y1].setEstat(false);

this.tablero[x2][y2].setEstat(false);

reset();
this.click = false;
} else {
this.x2 = i;
this.y2 = j;

this.tablero[i][j].setEstat(true);

this.click = true;
}

```

```

        if (fanParella(this.x1, this.y1,
this.x2, this.y2) && click) {

            reset();

            this.click = false;

        }

    } else {

        this.x1 = i;

        this.x2 = j;

        this.tablero[i][j].setEstat(true);

    }

}

}

}

}

}

public void omplir() {

    int n, x, y;

    ImageIcon imatgeActual;

    Random rnd = new Random();

    Boolean[] marcats = new Boolean[motes.length];

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

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

            this.tablero[i][j] = null;

        }

    }

    for (int i = 0; i < motes.length; i++) {

        marcats[i] = false;

    }

    for (int i = 0; i < (this.tablero.length * this.tablero[0].length) / 2; i++) {

        do {

            n = rnd.nextInt(motes.length);

        } while (marcats[n] != false);

        imatgeActual = new ImageIcon(JButtonMemory.class.getResource(motes[n]));

        marcats[n] = true;

        JButtonMemory r1 = new JButtonMemory(imatgeActual);

        JButtonMemory r2 = new JButtonMemory(imatgeActual);

```

```

        // while (taulell[x][y] != null) {
        do {
            x = rnd.nextInt(this.tablero.length);
            y = rnd.nextInt(this.tablero[0].length);
        } while (this.tablero[x][y] != null);

        this.tablero[x][y] = r1;
        this.tablero[x][y].addActionListener(this);

        do {
            x = rnd.nextInt(this.tablero.length);
            y = rnd.nextInt(this.tablero[0].length);
        } while (tablero[x][y] != null);

        this.tablero[x][y] = r2;
        this.tablero[x][y].addActionListener(this);
    }

    for (int i = 0; i < this.tablero.length; i++) {
        for (int j = 0; j < this.tablero[0].length; j++) {
            p2.add(this.tablero[i][j]);
        }
    }
}

public boolean fanParella(int fil1, int col1, int fil2, int col2) {
    if (fil1 == -1 || col1 == -1 || fil2 == -1 || col2 == -1)
        return false;
    if (this.tablero[fil1][col1].equals(this.tablero[fil2][col2]) && fil1 - col1 !=
fil2 - col2) {
        this.tablero[fil1][col1].setEnabled(false);
        this.tablero[fil2][col2].setEnabled(false);
        return true;
    }
    return false;
}

private void reset() {
    this.x1 = -1;
    this.y1 = -1;
}

```

```

        this.x2 = -1;
        this.y2 = -1;
    }

    public static void main(String[] args) {
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                JFrame xp = new Memory("Memory");
                xp.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                xp.setSize(800, 600);
                xp.setVisible(true);
            }
        });
    }
}

```