



PROGRAMACIÓN III

Trabajo práctico 1

Autores:

Elías Espinillo 40.663.777
Federico Farias 36.495.959

Profesores:

Javier Marengo
Patricia Bagnes

Contenidos

1	Implementación	3
----------	-----------------------	----------

1 Implementación

- Paquete comportamiento:

- Clase Flow:

```
public class Flow {

    public volatile static boolean juegoTerminado = false;
    public static boolean noPreparado = true;
    public static int filas, columnas, contador = 0;
    public Tablero juego;
    private Clip musica = JMusic.cargarSonido("externalMedia/musica.wav");

    public Flow() {

        columnasYFilas();

        //Si aparece la pantalla de ganador
        while (true) {
            if (juegoTerminado) {
                if (contador == 0) {

                    //Creamos la pantalla final
                    PantallaGanador fin = new PantallaGanador();
                    fin.setVisible(true);
                    juego.dispose();

                    //Listener para el boton de salir del juego
                    fin.salir.addActionListener(new ActionListener() {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                            musica.stop();
                            fin.dispose();
                        }
                    });

                    //Listener para el boton de jugar de nuevo
                    fin.seguirJugando.addActionListener(new ActionListener() {
                        @Override
                        public void actionPerformed(ActionEvent e) {
                            Tablero.cantMovimientos = 0;
                            columnasYFilas();
                            fin.dispose();
                        }
                    });
                    contador = 1;
                }
            }
        }

        private void columnasYFilas() {

            //Creamos la pantalla inicial
            PantallaInicial inicio = new PantallaInicial();
            musica.loop(Clip.LOOP_CONTINUOUSLY);
        }
    }
}
```

```
        inicio.setVisible(true);

        //Listener para el boton de personalizacion de columnas
        inicio.botonColumnas.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String inputColumna = null;
                try {
                    inputColumna = JOptionPane.showInputDialog("Ingresa las columnas que
desea");

                    columnas = Integer.parseInt(inputColumna);
                } catch (NumberFormatException i) {
                    if (inputColumna == null) {
                        noPreparado = false;
                    }
                }
                if (columnas == 0) {
                    JOptionPane.showMessageDialog(null, "El numero de columnas no puede
ser 0");
                }
            }
        });

        //Listener para el boton de personalizacion de filas
        inicio.botonFilas.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String inputFila = null;
                try {
                    inputFila = JOptionPane.showInputDialog("Ingresa las filas que
desea");

                    filas = Integer.parseInt(inputFila);
                } catch (NumberFormatException i) {
                    if (inputFila == null) {
                        noPreparado = false;
                    }
                }
                if (filas == 0) {
                    JOptionPane.showMessageDialog(null, "El numero de filas no puede ser
0");
                }
            }
        });

        //Listener para el boton de comienzo del juego
        inicio.botonInicioJuego.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if(filas == 0 || columnas == 0) {
                    JOptionPane.showMessageDialog(null, "El numero de columnas o filas
no puede ser 0");
                }
                juegoTerminado = false;
                juego = new Tablero();
                juego.setBounds(500, 200, 500, 500);
                inicio.dispose();
                Tablero.cantMovimientos = 0;
            }
        });
    }
}
```

- **Clase LucesFuera:**

```
public class LucesFuera {  
  
    public static Flow juego;  
  
    public static void main(String[] args) {  
        juego = new Flow();  
    }  
}
```

- **Paquete externalMedia:**

Contiene los archivos de fondo, fuente y sonidos del juego.

- **Paquete vistas:**

- **Clase PantallaInicial:**

```
public class PantallaInicial extends JFrame{  
  
    public JButton botonInicioJuego, botonColumnas, botonFilas;  
    private JLabel titulo;  
  
    public PantallaInicial() {  
  
        setTitle("Luces Fuera!");  
        setSize(500, 500);  
        setBounds(500, 200, 500, 500);  
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
        getContentPane().setLayout(null);  
  
        //Le agregamos una imagen de fondo  
        try {  
            BufferedImage image =  
ImageIO.read(getClass().getResource("/externalMedia/fondo.png"));  
            setContentPane(new ImagenFondo(image));  
        } catch (IOException e) {  
            e.printStackTrace();  
        }  
  
        //Cargamos una fuente personalizada y creamos lo necesario para la  
interfaz  
        try {  
            Font font = Font.createFont(Font.TRUETYPE_FONT,  
getClass().getResourceAsStream("/externalMedia/ARCADEPI.ttf"));  
            titulo = new JLabel("L U C E S   F U E R A   !");  
            titulo.setHorizontalAlignment(SwingConstants.CENTER);  
            titulo.setForeground(Color.CYAN);  
            titulo.setBounds(21, 32, 438, 21);  
        }  
    }  
}
```

```

        titulo.setFont(font.deriveFont(Font.PLAIN, 18f));
        getContentPane().add(titulo);

        botonColumnas = new JButton("COLUMNAS");
        botonColumnas.setBackground(Color.CYAN);
        botonColumnas.setBounds(21, 265, 138, 35);
        botonColumnas.setFont(font.deriveFont(Font.PLAIN, 16f));
        getContentPane().add(botonColumnas);

        botonFilas = new JButton("FILAS");
        botonFilas.setBackground(Color.CYAN);
        botonFilas.setBounds(324, 265, 135, 35);
        botonFilas.setFont(font.deriveFont(Font.PLAIN, 16f));
        getContentPane().add(botonFilas);

        botonInicioJuego = new JButton("JUGAR !");
        botonInicioJuego.setBackground(Color.CYAN);
        botonInicioJuego.setBounds(175, 337, 135, 35);
        botonInicioJuego.setFont(font.deriveFont(Font.PLAIN, 16f));
        getContentPane().add(botonInicioJuego);

        JTextArea txtBienvenida = new JTextArea();
        txtBienvenida.setForeground(Color.WHITE);
        txtBienvenida.setBounds(0, 105, 486, 79);
        txtBienvenida.setText("          INGRESA LAS COLUMNAS Y LAS FILAS \n          CON LAS QUE QUIERAS ARMAR EL TABLERO \n          CUANDO TERMINES PULSA JUGAR !");
        txtBienvenida.setFont(font.deriveFont(Font.PLAIN, 14f));
        txtBienvenida.setEditable(false);
        txtBienvenida.setOpaque(false);
        getContentPane().add(txtBienvenida);
    } catch (IOException | FontFormatException ex) {
        ex.printStackTrace();
    }
}

//Extendemos para poder personalizar el fondo con una imagen
public class ImagenFondo extends JComponent {

    private Image image;

    public ImagenFondo(Image image) {
        this.image = image;
    }
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawImage(image, 0, 0, this);
    }
}

```

- **Clase PantallaGanador:**

```

public class PantallaGanador extends JFrame{

    public JButton seguirJugando, salir;
    private JLabel cartelGanador, cartelMovFinales;

    public PantallaGanador() {

        setTitle("Fin del juego!");
    }
}

```

```

setSize(500, 500);
setBounds(500, 200, 500, 500);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
getContentPane().setLayout(null);

//Le agregamos una imagen de fondo
try {
    BufferedImage image =
ImageIO.read(getClass().getResource("/externalMedia/fondo.png"));
    setContentPane(new ImagenFondo(image));
} catch (IOException e) {
    e.printStackTrace();
}

//Cargamos una fuente personalizada y creamos lo necesario para la
interfaz
try {
    Font font = Font.createFont(Font.TRUETYPE_FONT,
getClass().getResourceAsStream("/externalMedia/ARCADEPI.ttf"));
    cartelGanador = new JLabel();
    cartelGanador.setForeground(Color.CYAN);
    cartelGanador.setBounds(121, 32, 249, 21);
    cartelGanador.setText("G A N A S T E ! ! !");
    cartelGanador.setFont(font.deriveFont(Font.PLAIN, 18f));
    getContentPane().add(cartelGanador);

    cartelMovFinales = new JLabel();
    cartelMovFinales.setHorizontalAlignment(SwingConstants.CENTER);
    cartelMovFinales.setForeground(Color.WHITE);
    cartelMovFinales.setBounds(0, 110, 486, 31);
    cartelMovFinales.setText("LO COMPLETASTE EN :  "+ Tablero.cantMovimientos
+ " MOVIMIENTO(S) !");
    cartelMovFinales.setFont(font.deriveFont(Font.PLAIN, 16f));
    getContentPane().add(cartelMovFinales);

    seguirJugando = new JButton("JUGAR");
    seguirJugando.setBackground(Color.CYAN);
    seguirJugando.setForeground(Color.BLACK);
    seguirJugando.setBounds(33, 344, 130, 35);
    seguirJugando.setFont(font.deriveFont(Font.PLAIN, 16f));
    getContentPane().add(seguirJugando);

    salir = new JButton("SALIR");
    salir.setBackground(Color.CYAN);
    salir.setBounds(331, 344, 130, 35);
    salir.setFont(font.deriveFont(Font.PLAIN, 16f));
    getContentPane().add(salir);
} catch (IOException | FontFormatException ex) {
    ex.printStackTrace();
}
}

//Extendemos para poder personalizar el fondo con una imagen
public class ImagenFondo extends JComponent {

    private Image image;

    public ImagenFondo(Image image) {
        this.image = image;
    }
    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.drawImage(image, 0, 0, this);
    }
}

```

```

    }
}

```

- **Clase Tablero:**

```

public class Tablero extends JFrame {

    private JButton[][] botonesDelTablero;
    private JPanel panelDelTablero, panelDePuntaje;
    private JLabel cartelMovimientos;
    public static int cantMovimientos;
    private Clip sonidoBoton = JMusic.cargarSonido("externalMedia/boton.wav");

    /**
     * Constructor del tablero y botones
     */
    public Tablero() {

        setTitle("Luces Fuera!");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //Creamos un panel para poder colocar el Label de movimientos y
        organizarlo junto a la matriz de botones
        try {
            Font font = Font.createFont(Font.TRUETYPE_FONT,
getClass().getResourceAsStream("/externalMedia/ARCADEPI.ttf"));
            panelDePuntaje = new JPanel();
            panelDePuntaje.setLayout(new GridLayout(1, 1));
            panelDePuntaje.add(cartelMovimientos = new JLabel());
            cartelMovimientos.setText("MOVIMIENTOS: " + "-");
            cartelMovimientos.setFont(font.deriveFont(Font.PLAIN, 14f));
            getContentPane().add(panelDePuntaje, BorderLayout.NORTH);
        } catch (IOException | FontFormatException ex) {
            ex.printStackTrace();
        }

        //Creamos una matriz para botones
        botonesDelTablero = new JButton[Flow.columnas][Flow.filas];
        panelDelTablero = new JPanel();
        panelDelTablero.setLayout(new GridLayout(Flow.columnas, Flow.filas));

        //Seteamos los valores del alto y ancho del tablero
        panelDelTablero.setPreferredSize(new Dimension(400, 400));

        //Ciclo anidado para crear los botones para la matriz
        for(int i = 0; i < botonesDelTablero.length; i++) {
            for(int j = 0; j < botonesDelTablero[i].length; j++) {

                //Colocamos los botones
                botonesDelTablero[i][j] = new JButton("");
                botonesDelTablero[i][j].setBackground(Color.magenta);

                //Le pasamos el Listener a cada boton
                botonesDelTablero[i][j].addActionListener(new Input());

                panelDelTablero.add(botonesDelTablero[i][j]);
            }
        }
    }
}

```



```

//Random para los botones y su posicion en el tablero
Random aleatorioParaColor = new Random();
for (JButton[] col : botonesDelTablero) {
    for (JButton boton : col) {
        boolean temp = aleatorioParaColor.nextBoolean();
        if (temp) {
            boton.doClick();
            cantMovimientos = -1;
        }
    }
}

//Finalmente agregamos el panel con botones
getContentPane().add(panelDelTablero);
getContentPane().repaint();
setVisible(true);
}

/**
 * Metodo que se utiliza para el cambio de colores de los botones
 * @param boton
 */
private void cambiarColor(JButton boton) {
    if (boton.getBackground().equals(Color.black)) {
        boton.setBackground(Color.magenta);
    }
    else {
        boton.setBackground(Color.black);
    }
}

/**
 * Metodo que verifica si todos los botones de la matriz son del color ganador
 */
private boolean ganar() {
    for (int i = 0; i < botonesDelTablero.length; i++) {
        for (int j = 0; j < botonesDelTablero[i].length; j++) {
            if(botonesDelTablero[i][j].getBackground() == Color.magenta) {
                return false;
            }
        }
    }
    return true;
}

//Implementamos los listener para las acciones de los botones del tablero y que
responda a lo que pide el juego
public class Input implements ActionListener {
    @Override
    public void actionPerformed(ActionEvent e) {
        cantMovimientos++;
        cartelMovimientos.setText("Movimientos: " + cantMovimientos);
        if(cantMovimientos > 0) {
            sonidoBoton.loop(1);
        }
        boolean flag = false;
        int columna = 0;
        int filas = 0;
        for (int i = 0; i < botonesDelTablero.length; i++) {
            for (int j = 0; j < botonesDelTablero[i].length; j++) {
                if (((JButton)e.getSource()).equals(botonesDelTablero[i][j])) {
                    flag = true;
                }
            }
        }
    }
}

```

```
                columna = i;
                filas = j;
                break;
            }
        }
        if (flag) {
            break;
        }
    }
    cambiarColor(botonesDelTablero[columna][filas]);
    if (columna > 0) {
        cambiarColor(botonesDelTablero[columna - 1][filas]);
    }
    if (columna < botonesDelTablero.length - 1) {
        cambiarColor(botonesDelTablero[columna + 1][filas]);
    }
    if (filas > 0) {
        cambiarColor(botonesDelTablero[columna][filas - 1]);
    }
    if (filas < botonesDelTablero[columna].length - 1) {
        cambiarColor(botonesDelTablero[columna][filas + 1]);
    }
    if (ganar()) {
        Flow.juegoTerminado = true;
        Flow.contador = 0;
    }
}
}
```