

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

INTRODUCCIÓN A LA PROGRAMACIÓN Y COMPUTACIÓN 1

CATEDRÁTICO: ING. WILLIAM ESTUARDO ESCOBAR ARGUETA

TUTOR ACADÉMICO: JOSUÉ RODOLFO MORALES CASTILLO



ELEAZAR NEFTALÍ COLOP COLOP

CARNÉ: 3198935960914

SECCIÓN: A

GUATEMALA, 20 DE JUNIO DEL 2,024

ÍNDICE

ÍNDICE	1
INTRODUCCIÓN	1
OBJETIVOS	1
1. GENERAL	1
2. ESPECÍFICOS	1
ALCANCES DEL SISTEMA	1
ESPECIFICACIÓN TÉCNICA	1
• REQUISITOS DE HARDWARE	1
• REQUISITOS DE SOFTWARE	1
DESCRIPCIÓN DE LA SOLUCIÓN	2
LÓGICA DEL PROGRAMA	2
❖ NOMBRE DE LA CLASE	
Captura de las librerías usadas	2
➤ Librerías	2
➤ Variables Globales de la clase _(El nombre de su clase actual)	3
➤ Función Main	3
➤ Métodos y Funciones utilizadas	3

INTRODUCCIÓN

El propósito de este manual técnico es proporcionar una guía detallada sobre el código fuente del proyecto de Space Invaders. Este manual está diseñado para ayudar a los desarrolladores a comprender la estructura, los componentes y la lógica detrás del juego. Se enfoca en la implementación técnica, incluyendo los requisitos de hardware y software, la descripción de la solución, y la lógica del programa.

OBJETIVOS

1. GENERAL

- 1.1. Este manual tiene como objetivo proporcionar una comprensión clara y detallada del código del juego Space Invaders, facilitando su mantenimiento y desarrollo futuro.

2. ESPECÍFICOS

- 2.1. Describir detalladamente las clases, métodos y variables utilizadas en el código, explicando su funcionalidad y cómo interactúan entre sí.
- 2.2. Proporcionar los requisitos de hardware y software necesarios para trabajar con la aplicación y desarrollar futuras mejoras.

ALCANCES DEL SISTEMA

El objetivo de este manual es ofrecer una descripción técnica exhaustiva del proyecto Space Invaders. Esto incluye la especificación de los requisitos de hardware y software, una descripción de la solución implementada, y una explicación detallada de la lógica del programa. El manual está dirigido a desarrolladores que deseen mantener o mejorar el código existente.

ESPECIFICACIÓN TÉCNICA

- **REQUISITOS DE HARDWARE**

- Procesador: Intel Core i5 o superior.
- Memoria RAM: 4 GB o más.
- Espacio en disco duro: 500 MB de espacio libre.
- Tarjeta gráfica: Cualquier tarjeta compatible con Java AWT/Swing.

- **REQUISITOS DE SOFTWARE**

- Sistema operativo: Windows 7/8/10, macOS, o Linux.
- JDK (Java Development Kit) 8 o superior.
- IDE (Integrated Development Environment) recomendado: IntelliJ IDEA, Eclipse, o NetBeans.
- Librerías adicionales: Ninguna, todas las librerías usadas son parte de Java Standard Edition.

DESCRIPCIÓN DE LA SOLUCIÓN

- La solución se basa en una interfaz gráfica de usuario desarrollada con Java Swing. El juego Space Invaders fue diseñado para incluir una pantalla de menú donde los usuarios pueden iniciar un nuevo juego, cargar un juego existente, ver los mejores puntajes o salir de la aplicación. Cada funcionalidad está implementada en una clase separada para mantener la modularidad y facilitar el mantenimiento.

LÓGICA DEL PROGRAMA

❖ MenuUsuario

Captura de las librerías usadas

➤ Librerías

- ❖ **java.awt.Graphics:** Utilizada para dibujar gráficos en componentes.
- ❖ **java.awt.Image:** Maneja imágenes.
- ❖ **java.awt.event.ActionEvent:** Maneja eventos de acción.
- ❖ **java.awt.event.ActionListener:** Define una interfaz para recibir eventos de acción.
- ❖ **javax.swing.ImageIcon:** Maneja iconos de imágenes.
- ❖ **javax.swing.JButton:** Crea botones.
- ❖ **javax.swing.JFrame:** Crea la ventana principal.
- ❖ **javax.swing.JLabel:** Muestra texto o imágenes.
- ❖ **javax.swing.JOptionPane:** Muestra cuadros de diálogo.
- ❖ **javax.swing.JPanel:** Panel genérico para contener otros componentes.

➤ Variables Globales de la clase `_`(MenuUsuario)

- **nuevoJuegoButton:** Botón para iniciar un nuevo juego.
- **cargarJuegoButton:** Botón para cargar un juego existente.
- **puntuacionMaximaButton:** Botón para mostrar la puntuación máxima.
- **salirButton:** Botón para salir del juego.

➤ Función Main

La clase `MenuUsuario` no contiene una función main directa, ya que está diseñada para ser instanciada y mostrada desde otra clase `MAIN` que inicia la aplicación. La inicialización de la interfaz se realiza al llamar el constructor de `MenuUsuario`.

➤ Procedimientos, métodos y Funciones utilizadas

❖ `initComponents():`

Inicializa todos los componentes de la interfaz gráfica del menú de usuario.

```

33 private void initComponents() {
34     // Crear panel de fondo personalizado
35     ImagenFondo backgroundPanel = new ImagenFondo();
36     backgroundPanel.setLayout(null);
37     setContentPane(contentPane.backgroundPanel);
38
39     // Cargar la imagen del título y redimensionarla
40     ImageIcon titleIcon = new ImageIcon(location.getClass().getResource(name: "title.png"));
41     Image titleImage = titleIcon.getImage();
42     Image scaledTitleImage = titleImage.getScaledInstance(width: 300, height: 100, hints: Image.SCALE_SMOOTH);
43     ImageIcon scaledTitleIcon = new ImageIcon(image: scaledTitleImage);
44
45     JLabel titleLabel = new JLabel(image: scaledTitleIcon);
46     titleLabel.setBounds(x: 100, y: 20, width: scaledTitleIcon.getIconWidth(), height: scaledTitleIcon.getIconHeight());
47     backgroundPanel.add(comp: titleLabel);
48
49     // Botones
50     nuevoJuegoButton = new JButton(text: "Nuevo Juego");
51     nuevoJuegoButton.setBounds(x: 150, y: 150, width: 200, height: 30);
52     nuevoJuegoButton.addActionListener(l: this);
53     backgroundPanel.add(comp: nuevoJuegoButton);
54
55     cargarJuegoButton = new JButton(text: "Cargar Juego");
56     cargarJuegoButton.setBounds(x: 150, y: 200, width: 200, height: 30);
57     cargarJuegoButton.addActionListener(l: this);
58     backgroundPanel.add(comp: cargarJuegoButton);

```

❖ actionPerformed(ActionEvent ae):

Maneja las acciones de los botones del menú.

```

79 @Override
80 public void actionPerformed(ActionEvent ae) {
81     if (ae.getSource() == nuevoJuegoButton) {
82         // Iniciar un nuevo juego
83         String playerName = JOptionPane.showInputDialog(parentComponent: null, message: "Introduce tu nombre:");
84         if (playerName == null || playerName.trim().isEmpty()) {
85             playerName = "Jugador";
86         }
87
88         JFrame frame = new JFrame(title: "Space Invaders");
89         SpaceInvaders game = new SpaceInvaders(playerName);
90         frame.add(comp: game);
91         frame.setSize(width: 1200, height: 650);
92         frame.setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
93         frame.setResizable(resizable: false);
94         frame.setVisible(b: true);
95         this.dispose();
96     } else if (ae.getSource() == cargarJuegoButton) {
97         // Cargar un juego existente
98         new CargarJuego();
99         // Cerrar el menú de usuario
100         this.dispose();

```

❖ CargarJuego

Captura de las librerías usadas

➤ Librerías

- **java.awt.Image:** Maneja imágenes.
- **java.awt.Graphics:** Utilizada para dibujar gráficos en componentes.
- **java.awt.event.ActionEvent:** Maneja eventos de acción.
- **java.awt.event.ActionListener:** Define una interfaz para recibir eventos de acción.
- **java.io.File:** Maneja archivos.
- **javax.swing.ImageIcon:** Maneja iconos de imágenes.
- **javax.swing.JButton:** Crea botones.

- **javax.swing.JFileChooser:** Crea un cuadro de diálogo para seleccionar archivos.
- **javax.swing.JFrame:** Crea la ventana principal.
- **javax.swing.JLabel:** Muestra texto o imágenes.
- **javax.swing.JPanel:** Panel genérico para contener otros componentes.
- **javax.swing.filechooser.FileNameExtensionFilter:** Filtra archivos por extensión en el cuadro de diálogo.sd

➤ Variables Globales de la clase **_**(CargarJuego)

- **seleccionarArchivoButton:** Botón para seleccionar el archivo de juego.
- **cargarPartidaButton:** Botón para cargar la partida.
- **regresarMenuButton:** Botón para regresar al menú principal.
- **estadoLabel:** Etiqueta para mostrar el estado de la carga.
- **archivoSeleccionado:** Archivo seleccionado para cargar la partida.

❖ CargarJuego

Captura de las librerías usadas

➤ Librerías

- **java.awt.Image:** Maneja imágenes.
- **java.awt.Graphics:** Utilizada para dibujar gráficos en componentes.
- **java.awt.event.ActionEvent:** Maneja eventos de acción.
- **java.awt.event.ActionListener:** Define una interfaz para recibir eventos de acción.
- **java.io.File:** Maneja archivos.
- **javax.swing.ImageIcon:** Maneja iconos de imágenes.
- **javax.swing.JButton:** Crea botones.
- **javax.swing.JFileChooser:** Crea un cuadro de diálogo para seleccionar archivos.
- **javax.swing.JFrame:** Crea la ventana principal.
- **javax.swing.JLabel:** Muestra texto o imágenes.
- **javax.swing.JPanel:** Panel genérico para contener otros componentes.
- **javax.swing.filechooser.FileNameExtensionFilter:** Filtra archivos por extensión en el cuadro de diálogo.sd

➤ Variables Globales de la clase **_**(MenuUsuario)

- **seleccionarArchivoButton:** Botón para seleccionar el archivo de juego.
- **cargarPartidaButton:** Botón para cargar la partida.
- **regresarMenuButton:** Botón para regresar al menú principal.
- **estadoLabel:** Etiqueta para mostrar el estado de la carga.
- **archivoSeleccionado:** Archivo seleccionado para cargar la partida.

➤ Función Main

La clase `MenuUsuario` no contiene una función main directa, ya que está diseñada para ser instanciada y mostrada desde otra clase MAIN que inicia la aplicación. La inicialización de la interfaz se realiza al llamar el constructor de `MenuUsuario`.

➤ Procedimientos, métodos y Funciones utilizadas

❖ cargarMejoresJugadoresDesdeCSV():

Carga los mejores jugadores desde un archivo CSV y los muestra en la interfaz.

```
67 private void cargarMejoresJugadoresDesdeCSV(String filePath) {
68     playerLabels = new ArrayList<>();
69     java.util.List<Jugador> jugadores = new ArrayList<>();
70     try (BufferedReader br = new BufferedReader(new FileReader(filePath))) {
71         String line;
72         while ((line = br.readLine()) != null) {
73             String[] parts = line.split(",");
74             if (parts.length >= 2) {
75                 Jugador jugador = new Jugador(parts[0], puntuacion: Integer.parseInt(parts[1]));
76                 jugadores.add(e: jugador);
77             }
78         }
79     } catch (IOException e) {
80         e.printStackTrace();
81         JOptionPane.showMessageDialog(parentComponent: this, message: "Error al leer el archivo CSV", title: "Error",
82                                     JOptionPane.ERROR_MESSAGE);
83         return;
84     }
85     // Ordenar jugadores por puntuación (de mayor a menor)
86     Collections.sort(list: jugadores, c: Collections.reverseOrder());
87
88     // Mostrar los mejores jugadores
89     int contador = 0;
90     for (int i = 0; i < jugadores.size() && contador < 5; i++) {
91         Jugador jugador = jugadores.get(index: i);
92         int j = i;
93         while (j < jugadores.size() && jugadores.get(index: j).getPuntuacion() == jugador.getPuntuacion()) {
94             JLabel playerLabel = new JLabel((contador + 1) + "% lugar: " + jugadores.get(index: j).getNombre());
```

❖ actionPerformed(ActionEvent ae):

Maneja las acciones de los botones del menú.

```
79 @Override
80 public void actionPerformed(ActionEvent ae) {
81     if (ae.getSource() == nuevoJuegoButton) {
82         // Iniciar un nuevo juego
83         String playerName = JOptionPane.showInputDialog(parentComponent: null, message: "Introduce tu nombre:");
84         if (playerName == null || playerName.trim().isEmpty()) {
85             playerName = "Jugador";
86         }
87
88         JFrame frame = new JFrame(title: "Space Invaders");
89         SpaceInvaders game = new SpaceInvaders(playerName);
90         frame.add(comp: game);
91         frame.setSize(width: 1200, height: 650);
92         frame.setDefaultCloseOperation(operation: JFrame.EXIT_ON_CLOSE);
93         frame.setResizable(resizable: false);
94         frame.setVisible(b: true);
95         this.dispose();
96     } else if (ae.getSource() == cargarJuegoButton) {
97         // Cargar un juego existente
98         new CargarJuego();
99         // Cerrar el menú de usuario
100        this.dispose();
101    }
```

