



MANUAL TÉCNICO

PRÁCTICA 2

Ángel Andrés Godoy Valdéz
Laboratorio IPC 1 Sección "B"
Carnet: 202113539

INTRODUCCIÓN	3
REQUERIMIENTOS TÉCNICOS.....	3
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	3
Java:.....	3
Intellij IDEA:	3
EJECUCIÓN DE APLICACIÓN DESDE SÍMBOLO DE SISTEMA	4
APLICACIÓN (EJECUTABLE JAR)	4
Clases.....	4
MyThread	4
FrameJuego	5
JuegoPanel	5

INTRODUCCIÓN

Este manual describe los procesos detrás del manejo de datos en la interfaz gráfica de la aplicación denominada “Juego Serpiente” en formato java realizada así como el código fuente que fue necesario para la practica No.2 para el Laboratorio del curso Introducción a la Programación de Computadoras 1.

REQUERIMIENTOS TÉCNICOS

Sistema Operativo: Windows 7/Vista/8/8.1/10/11

Aplicaciones: Java Development Kit (JDK) 12.0-actual

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

Java:

Java es una plataforma informática de lenguaje de programación creada por Sun Microsystems en 1995. Ha evolucionado desde sus humildes comienzos hasta impulsar una gran parte del mundo digital actual, ya que es una plataforma fiable en la que se crean muchos servicios y aplicaciones. Los nuevos e innovadores productos y servicios digitales diseñados para el futuro también siguen basándose en Java.

Aunque la mayoría de aplicaciones Java modernas combinan el tiempo de ejecución y la aplicación de Java, todavía existen algunas aplicaciones e incluso sitios web que no funcionan sin instalar Java para escritorio. El sitio web Java.com está pensado para consumidores que todavía necesitan Java en sus aplicaciones de escritorio, sobre todo las aplicaciones que tienen como destino Java 8.

IntelliJ IDEA:

IntelliJ IDEA es un entorno de desarrollo integrado (IDE) para el desarrollo de programas informáticos. Es desarrollado por JetBrains (anteriormente conocido como IntelliJ), y está disponible en dos ediciones: edición para la comunidad y edición comercial.

Cada aspecto de IntelliJ IDEA ha sido diseñado para maximizar la productividad del desarrollador. Juntos, la asistencia de codificación inteligente y el diseño ergonómico hacen que el desarrollo no solo sea productivo sino también agradable.

Después de que IntelliJ IDEA haya indexado su código fuente, ofrece una experiencia increíblemente rápida e inteligente al brindar sugerencias relevantes en cada contexto: finalización de código instantánea e inteligente, análisis de código sobre la marcha y herramientas de refactorización confiables. Las herramientas de misión crítica, como los sistemas de control de versiones integrados y una amplia variedad de lenguajes y marcos compatibles, están todos a mano, sin complicaciones

de complementos incluidos. Mientras que la finalización básica sugiere nombres de clases, métodos, campos y palabras clave dentro del ámbito de visibilidad, la finalización inteligente sugiere solo los tipos que se esperan en el contexto actual.

EJECUCIÓN DE APLICACIÓN DESDE SÍMBOLO DE SISTEMA

Una vez descargado desde el repositorio de github, se debe encontrar la ruta de acceso para el jar, el cual se encuentra ubicado dentro de la carpeta .idea/artifacts.

Pasos a realizar:

1. Presionar Windows+R
2. Teclear "cmd" (esto abra el símbolo del sistema)
3. Dentro del cmd teclear "java -jar <ruta del jar>"
4. Presionar Enter
5. ¡Listo, ya está dentro de la aplicación!

APLICACIÓN (EJECUTABLE JAR)

Clases

Cada clase es instanciada entre sí, es importante mencionar que el único y obligatorio thread de este proyecto se encuentra en una clase llamada Mythread, el resto se utilizan para la creación del frame y el panel que contiene la interfaz del juego Snake.

MyThread

Bajo esta clase se crea y gestiona un thread que es posible por un "Extends" al inicio de la clase, un único método que corre el thread y lo detiene cuando así se instancia.

```
public class MyThread extends Thread {  
    2 usages  
    public static int minuto=0;  
    4 usages  
    public static int segundos=0;  
    Angel Andres Godoy Valdez  
    @Override  
    public void run(){  
        try {  
            while(Thread.currentThread().isAlive()) {  
                Thread.sleep(1000);  
                if (segundos==60){  
                    segundos=0;  
                    minuto++;  
                }else {  
                    segundos++;  
                }  
            }  
        }  
        catch (Exception e){  
        }  
    }  
}
```

FrameJuego

Dentro de esta clase se crea el frame, se definen sus dimensiones, operaciones de salida y otras propiedades. De igual forma se crea el panel en el que se almacena el juego.

```
FrameJuego(){
    JuegoPanel PJuego = new JuegoPanel();

    JComboBox difCBox = new JComboBox();
    difCBox.setBounds(x: 800, y: 100, width: 150, height: 20);
    difCBox.setBackground(Color.WHITE);
    String[] arrayAux = {"Fácil", "Medio", "Difícil"};
    JComboBox comboBoxAux = new JComboBox(arrayAux);
    difCBox.setModel(comboBoxAux.getModel());

    JButton nuevoJuegoBtn = new JButton();
    nuevoJuegoBtn.setFocusable(false);
    nuevoJuegoBtn.setBackground(Color.WHITE);
    nuevoJuegoBtn.setText("Nuevo Juego");
    nuevoJuegoBtn.setBounds(x: 1000, y: 90, width: 150, height: 50);
    nuevoJuegoBtn.setBorder(BorderFactory.createMatteBorder(top: 2, left: 2, bottom: 2, right: 2,
```

Angel Andres Godoy Valdez

JuegoPanel

Aquí es donde se encuentra la mayor cantidad de métodos pues controla todas las funciones del juego, como la posición de la serpiente y la comida, sus componentes graficos y puntaje entre otros controles.

```
JComboBox difCBox = new JComboBox();
difCBox.setBounds(x: 800, y: 100, width: 150, height: 20);
difCBox.setBackground(Color.WHITE);
String[] arrayAux = {"Fácil", "Medio", "Difícil"};
JComboBox comboBoxAux = new JComboBox(arrayAux);
difCBox.setModel(comboBoxAux.getModel());

JButton nuevoJuegoBtn = new JButton();
nuevoJuegoBtn.setFocusable(false);
nuevoJuegoBtn.setBackground(Color.WHITE);
nuevoJuegoBtn.setText("Nuevo Juego");
nuevoJuegoBtn.setBounds(x: 1000, y: 90, width: 150, height: 50);
nuevoJuegoBtn.setBorder(BorderFactory.createMatteBorder(top: 2, left: 2, bottom: 2, right: 2, Color.black));

Angel Andres Godoy Valdez
nuevoJuegoBtn.addActionListener(new ActionListener() {
    Angel Andres Godoy Valdez
    @Override
    public void actionPerformed(ActionEvent e) {
        switch (difCBox.getSelectedIndex()){
            case 0:
                JuegoPanel.dificultad="F";
                JuegoPanel.corriendo=true;
```