**Guía para crear videojuego Space Invaders en JavaFX**

**Preparacion de Entorno y un Menu Inicial en JavaFx**

* Vamos a ela

1. Verificamos JDK y Maven (hasta este punto estos serian comandos que se pueden probar de los mas utilizados).

* Ejecutar en terminal:
* java -version
* mvn -v
* Asegurarse de que coinciden.
* Comandos Básicos de Maven:
* Compilar proyectos: mvn compile
* Ejecutar pruebas: mvn test
* Empaquetar proyecto: mvn package(jar/war)
* Limpiar la carpeta target: mvn clean
* Limpiar y recompilar: mvn clean compile
* Limpiar y empaquetar: mvn clean package
* Instalar en el repositorio local:mvn install
* Ejecutar el proyecto con JavaFx plugin: mvn javafx:run
* Comandos Auditar y comprobar POM:
  + Ver que lo que se usa de Maven: mvn help:effective-pom -Dverbose
  + Arbol de dependencias: mvn dependency:tree
  + Comprobar versiones de dependencias: mvn versions:display-dependency-updates
  + Comprobar versiones plugins: mvn versions:display-plugin-updates
  + Comprobar valores del POM:
    - mvn -q help:evaluate -Dexpression=project.groupId
    - mvn -q help:evaluate -Dexpression=project.version
    - mvn -q help:evaluate -Dexpression=javafx.version

1. Estructura del proyecto

src/

|----------- main/

|---------- java/tuempresa/spaceinvaders/

| | -- App.java

| | -- gui/MenuInicio.java

|----------resources/css/

| -- styles.css

1. Configuracion del pom.xml

Project Object Model de Maven.

Aquí se define toda la configuración del proyecto, nombre, versión, dependencias, plugins y compilador….

<?xml version="1.0" encoding="UTF-8"?>

<project xmlns="http://maven.apache.org/POM/4.0.0"

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"

xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">

Versión del modelo de Maven

<modelVersion>4.0.0</modelVersion>

Identificadores del proyecto

<groupId>kabalera82</groupId>

<artifactId>spaceinvaders</artifactId>

<version>1.0</version>

<name>spaceinvaders</name>

Propiedades generales del proyecto

<properties>

Codificación de ficheros fuente

<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>

Versión de Java utilizada para compilar

<maven.compiler.source>17</maven.compiler.source>

<maven.compiler.target>17</maven.compiler.target>

Versión de JavaFX que vamos a usar

<javafx.version>17</javafx.version>

</properties>

Dependencias necesarias para compilar y ejecutar el proyecto

<dependencies>

Librerías de controles de JavaFX (Botones, Labels, Paneles, etc.)

<dependency>

<groupId>org.openjfx</groupId>

<artifactId>javafx-controls</artifactId>

<version>${javafx.version}</version>

</dependency>

Librerías de JavaFX para usar FXML (si usamos Scene Builder/FXML)

<dependency>

<groupId>org.openjfx</groupId>

<artifactId>javafx-fxml</artifactId>

<version>${javafx.version}</version>

</dependency>

</dependencies>

Configuración de la construcción (build)

<build>

<plugins>

Plugin oficial de JavaFX para ejecutar con Maven

<plugin>

<groupId>org.openjfx</groupId>

<artifactId>javafx-maven-plugin</artifactId>

<version>0.0.8</version>

Configuración del plugin

<configuration>

Clase principal que arranca el programa (extends Application)

<mainClass>kabalera82.spaceinvaders.App</mainClass>

Aquí se añadirán <modules> para indicar los módulos de JavaFX

(javafx.controls, javafx.fxml, javafx.media, etc.)

si se necesitan más en el futuro.

</configuration>

</plugin>

</plugins>

</build>

</project>

1. Queremos un **menú de inicio** con:

* Un título que diga *"Menú Inicio"*.
* Un botón que diga *"Start"*.
* Que al pulsarlo aparezca el mensaje *"Cargando..."*.

**4. El corazón del juego: App.java**

Es el **motor de arranque**. Dice: "abre la ventana principal y enseña el menú".

public class App extends Application {

@Override

public void start(Stage escenario) {

Scene menu = new MenuInicio().construir();

escenario.setTitle("Juego FX");

escenario.setResizable(false);

escenario.setScene(menu);

escenario.show();

}

public static void main(String[] args) { launch(); }

}

**5. El menú: MenuInicio.java**

Aquí dibujamos lo que se ve en pantalla: título y botón.

public class MenuInicio {

public Scene construir() {

Text titulo = new Text("Menú Inicio");

titulo.getStyleClass().add("titulo");

Button botonStart = new Button("Start");

botonStart.getStyleClass().add("boton-principal");

botonStart.setOnAction(e -> {

botonStart.setDisable(true);

botonStart.setText("Cargando...");

});

VBox contenedor = new VBox(20, titulo, botonStart);

contenedor.setAlignment(Pos.CENTER);

contenedor.setPrefSize(720, 480);

contenedor.getStyleClass().add("fondo-inicio");

Scene escena = new Scene(contenedor);

escena.getStylesheets().add(getClass().getResource("/css/styles.css").toExternalForm());

return escena;

}

}

**6. Los colores y estilos: styles.css**

Aquí ponemos la **ropa bonita** de nuestra aplicación.

.fondo-inicio {

-fx-background-color: linear-gradient(to bottom, #0b1020, #141a33);

}

.titulo {

-fx-fill: white;

-fx-font-size: 36px;

-fx-font-weight: bold;

}

.boton-principal {

-fx-font-size: 20px;

-fx-padding: 10 20;

-fx-background-radius: 10;

-fx-background-color: #3366cc;

-fx-text-fill: white;

}

.boton-principal:hover {

-fx-background-color: #5588ff;

}

**7. Cómo ejecutarlo**

1. Abre la terminal en la carpeta del proyecto.
2. Escribe:
3. mvn javafx:run
4. ¡Se abrirá la ventana con tu **menú de inicio**!

**8. ¿Qué hemos aprendido?**

* **JavaFX** = herramienta para crear pantallas con botones y colores.
* **App.java** = motor que arranca el juego.
* **MenuInicio.java** = lo que se dibuja en pantalla.
* **CSS** = pinta y decora los botones y textos.

**9. Próximo paso**

Cuando pulsemos *Start*, podemos cambiar a otra pantalla donde empiece el juego de verdad: la nave, los disparos, los enemigos…