

Enunciado del Juego de Lucha 2D en Java

Descripción General

Se debe desarrollar un **juego de lucha en 2D** utilizando **Java** y la biblioteca **Swing** o **JavaFX** para la interfaz gráfica. En este juego, dos personajes se enfrentarán en un escenario con animaciones y una mecánica de combate basada en golpes y detección de colisiones.

Cada jugador tendrá una **barra de vida**, un **temporizador** y un sistema de ataque que detecta impactos al oponente. El objetivo es **reducir la vida del adversario a cero antes de que el tiempo se acabe**.

Requisitos del Juego

1. Interfaz Gráfica

- El juego debe ejecutarse en una **ventana de Java** (**JFrame** en Swing o **Stage** en JavaFX).
- Contará con un **panel de juego** (**JPanel** o **Canvas**) donde se renderizarán los personajes y el escenario.
- Se incluirá un **HUD** con:
 - **Barras de vida** de los dos jugadores.
 - **Temporizador** que cuenta regresivamente desde 120 segundos.
 - **Mensaje de victoria/derrota** al finalizar la partida.

2. Personajes y Animaciones

- Se implementarán dos personajes jugables con distintas animaciones para:
 - **Caminar** (izquierda/derecha).
 - **Saltar y caer** con efecto de gravedad.
 - **Atacar** con un área de impacto (hitbox).
 - **Recibir daño** y reducir la vida al ser golpeados.
 - **Muerte** con animación de derrota.
- Los sprites se cargarán desde archivos de imagen (**BufferedImage** en Swing o **ImageView** en JavaFX).

3. Controles del Jugador

Cada jugador tendrá su propio conjunto de teclas para moverse y atacar:

- **Jugador 1** (Samurai Mack)
 - **A** → Moverse a la izquierda.
 - **D** → Moverse a la derecha.
 - **W** → Saltar.
 - **S** → Atacar.
- **Jugador 2** (Kenji)
 - **←** → Moverse a la izquierda.
 - **→** → Moverse a la derecha.

- ↑ → Saltar.
- ↓ → Atacar.

Los eventos del teclado se manejarán con **KeyListener** en Swing o con **setOnKeyPressed()** en JavaFX.

Mecánica del Juego

4. Sistema de Batalla

- **Colisiones:** Se detectará si el hitbox del ataque de un personaje toca al oponente.
- **Reducción de Vida:** Un golpe exitoso **quita 20 puntos de vida** al adversario.
- **Temporizador:** Un contador regresivo **termina la partida** si llega a 0.
- **Diferentes estados de los personajes:** Se usarán animaciones para ataque, movimiento y recibir daño.

5. Condiciones de Victoria

El juego finaliza en los siguientes casos:

- **Si un jugador pierde toda su vida**, el oponente gana.
 - **Si el tiempo llega a 0**, gana el jugador con **más vida restante**.
 - **Si ambos tienen la misma vida cuando el tiempo se acaba, hay un empate.**
-

Estructura del Código

El programa estará organizado en diferentes clases:

1. **Juego.java** → Contiene el **main()** y controla el flujo general del juego.
 2. **VentanaJuego.java** → Administra la ventana principal (**JFrame** en Swing o **Stage** en JavaFX).
 3. **PanelJuego.java** → Maneja la lógica del juego y el renderizado de los personajes.
 4. **Personaje.java** → Clase base para los personajes, con atributos como posición, velocidad, animaciones y salud.
 5. **Jugador.java y Enemigo.java** → Extienden **Personaje** y manejan las acciones de cada luchador.
 6. **Colisionador.java** → Clase que detecta colisiones entre los hitboxes de los personajes.
 7. **Temporizador.java** → Controla la cuenta regresiva del tiempo.
-

Tecnologías y Librerías Utilizadas

- **Java SE 8+**
 - **Swing o JavaFX** para la interfaz gráfica.
 - **BufferedImage / ImageView** para la carga de sprites.
 - **KeyListener o setOnKeyPressed** para los controles del jugador.
 - **Timer / AnimationTimer** para el manejo del tiempo y la lógica del juego.
-

Objetivo del Proyecto

Este juego servirá como **práctica de programación en Java**, aplicando conceptos de **POO (Programación Orientada a Objetos)**, **manipulación de imágenes, detección de colisiones y eventos del teclado**. Se busca que los jugadores disfruten de un **versus dinámico y entretenido**, con mecánicas similares a los clásicos juegos de lucha en 2D.