Trabajo Práctico: El camino de Gondolf

# Carátula

Integrantes del grupo 10:

Cabral Bruno

Cabral Walter

# Introducción

“Fear and hunger arcade” es un videojuego desarrollado en Java como parte del trabajo práctico de la materia Programación I. Utilizando la librería gráfica entorno.jar provista por la cátedra, diseñamos una propuesta en la que el jugador controla a un mago, quien debe sobrevivir y eliminar enemigos (murciélagos) usando distintos hechizos. El juego pone a prueba la lógica de programación orientada a objetos, el control de eventos y la detección de colisiones.

# Descripción de las clases

## Juego

Clase principal que inicia y gestiona la ejecución del juego. Controla los estados del menú, partida activa y pantalla de game over. Coordina el dibujo de objetos y actualiza su lógica de forma cíclica.

## Menu

Gestiona el menú de inicio. Presenta una seleccion de personajes, contando con 2 personajes a elegir.

## Personaje

Representa al mago controlado por el jugador.

## Murcielago

Representa a los enemigos voladores que atacan al jugador.

## Obstaculo

Bloques fijos en el escenario que impiden el paso del jugador y los enemigos.

## Poderes

Contiene la lógica para uno los hechizos lanzados por el personaje.

## Pocion

Objetos que curan y dan maná al personaje y al recogerlos.

## Barra

Representa visualmente la vida y el maná del personaje en pantalla.

## Boton

Representa botones interactivos del menú. Permite seleccionar los distintos poderes.

## Bomba

La clase Bomba representa uno de los poderes ofensivos que puede usar el personaje principal del juego. Consiste en un area con forma de bomba que se lanza hacia una dirección específica con el mouse y puede dañar a los enemigos que se encuentran en ese radio.

# Problemas y soluciones

Durante el desarrollo del juego surgieron varios desafíos:  
- Colisiones entre objetos: Fue necesario implementar métodos precisos de detección de colisiones entre personajes, enemigos, obstáculos y proyectiles.  
- Control de estados: Para manejar los distintos momentos del juego (inicio, juego activo, game over), se implementó una máquina de estados que facilitó la lógica condicional.  
- Limitación del movimiento: Se desarrolló una verificación para evitar que el personaje atraviese obstáculos o salga del escenario.  
- Control de enemigos: Se implementó una lista dinámica de enemigos con comportamiento independiente, lo que permitió una mayor dificultad progresiva.  
- Balanceo del juego: Se ajustó la frecuencia de aparición de enemigos y pociones, así como el daño recibido, para ofrecer una experiencia equilibrada.

# Implementación

El código fuente completo se encuentra en los archivos adjuntos, correctamente formateado y comentado para facilitar su comprensión. Cada clase está documentada en su encabezado y los métodos incluyen comentarios explicativos de su propósito.

# Conclusiones

Desarrollar “El camino de Gondolf” nos permitió aplicar todos los conocimientos adquiridos en Programación I, desde la estructura de clases y objetos hasta la lógica de interacción en tiempo real. Además de afianzar el uso de Java, aprendimos a trabajar con una librería gráfica, organizar el proyecto en múltiples clases y resolver problemas comunes en videojuegos 2D.  
  
Como grupo, logramos organizar el trabajo de manera eficiente y dividir tareas según las habilidades de cada integrante. Entre las principales lecciones aprendidas destacamos:  
- La importancia de planificar la arquitectura del juego antes de programar.  
- La necesidad de mantener el código limpio, modular y reutilizable.  
- El valor de la depuración sistemática para resolver errores complejos.  
  
El trabajo fue una experiencia muy enriquecedora que nos motivó a seguir explorando el desarrollo de videojuegos y la programación orientada a objetos.