

CB100 - Algoritmos y Estructuras de Datos

Cátedra Ing. Gustavo Schmidt - 2er cuatrimestre 2025

Trabajo Práctico 1: Rolgar

Objetivo

Implementar un juego sencillo por consola en Java donde un personaje se mueva en un tablero de tamaño $N \times M$.

Enunciado

Definir el **tablero** de tamaño $N \times M$ (ingresado por teclado o fijo al inicio), mostrarlo en consola como una grilla, donde se vean los casilleros vacíos y un Personaje en el centro.

Ejemplo para 5×5 (P = Personaje, . = vacío):



Diagrama de un tablero 5x5 representado como una grilla. El personaje 'P' está en el centro (fila 3, columna 3). Los casilleros vacíos están representados por puntos '.'.

```
css
. . . . .
. . . . .
. . P . .
. . . . .
. . . . .
```

Se debe una **Personaje** con atributos:

- 1) nombre
- 2) vida (nivel de energía)

3) posX, posY (posición en el tablero)

El personaje comienza en el centro del tablero.

Los movimientos

Una vez empezado el juego, el usuario puede mover al personaje con las teclas:

W = arriba

S = abajo

A = izquierda

D = derecha

Enemigos

Colocar aleatoriamente 3 enemigos en el tablero (marcados con E). Al crear un enemigo también se cuenta con nombre, energía y la posición.

El jugador debe buscar los enemigos hasta estar en la misma celda, al encontrarlo el alumno debe definir una forma de pelea para encontrar un ganador.

Al pasar los turnos, el Personaje recupera un % de vida. Al imprimir el tablero, el Personaje solo ve las 8 casillas alrededor. El jugador gana cuando su personaje eliminó a los 3 enemigos, o pierde cuando se queda sin energía.

En el menú se deben ver las instrucciones del juego.

Interfaz de usuario

Toda la interfaz de usuario debe estar basada en texto. El menú debe ser muy sencillo y solo se imprime el tiempo demorado por pantalla.

No es necesario que se limpie la pantalla, simplemente escribir el menú cada vez que se necesite.

Cuestionario

Responder el siguiente Cuestionario:

- 1) ¿Qué es un Debug?
- 2) ¿Qué es un "Breakpoint"?
- 3) ¿Qué es "Step Into", "Step Over" y "Step Out"?

Normas de entrega

Trabajo práctico individual: 1 persona.

Reglas generales: respetar el Apéndice A.

Se deberá subir un único archivo comprimido al campus, en un link que se habilitará para esta entrega. Este archivo deberá tener un nombre formado de la siguiente manera:

Padron-TP1.zip

Deberá contener los archivos fuentes (no los binarios), los datos personales (padrón, nombre y mail), el informe del trabajo realizado, las respuestas al cuestionario, el manual del usuario y el manual del programador (Todo en el mismo PDF).

La fecha de entrega vence el día viernes 05/09/25 a las 23.59hs por el campus.

Se evaluará: funcionalidad, eficiencia, algoritmos utilizados, buenas prácticas de programación, modularización, documentación, gestión de memoria y estructuras de datos.

Apéndice A

- 1) Usar las siguientes convenciones para nombrar identificadores.
 - a) Clases: Los nombres de clases y structs siempre deben comenzar con la primera letra en mayúscula en cada palabra, deben ser simples y descriptivos. Se concatenan todas las palabras. Ejemplo: Coche, Vehiculo, CentralTelefonica.

- b) Métodos: Deben comenzar con letra minúscula, y si está compuesta por 2 o más palabras, la primera letra de la segunda palabra debe comenzar con mayúscula. De preferencia que sean verbos. Ejemplo: arrancarCoche(), sumar().
 - c) Variables y objetos: las variables siguen la misma convención que los métodos. Por Ejemplo: alumno, padronElectoral.
 - d) Constantes: Las variables constantes o finales, las cuales no cambian su valor durante todo el programa se deben escribir en mayúsculas, concatenadas por "_". Ejemplo: ANCHO, VACIO, COLOR_BASE.
- 2) Si el trabajo práctico requiere archivos para procesar, entregar los archivos de prueba en la entrega del TP. Utilizar siempre rutas relativas y no absolutas.
 - 3) Entregar el informe explicando el TP realizado, manual de usuario y manual del programador.
 - 4) Comentar el código. Todos los tipos, métodos y funciones deberían tener sus comentarios.
 - 5) Modularizar el código. No entregar 1 o 2 archivos, separar cada clase.
 - 6) No utilizar variables globales.
 - 7) Si cualquier estructura de control tiene 1 línea, utilizar {} siempre, por ejemplo:

```
for(int i = 0; i < 10; i++) {  
    System.out.println( i );  
}
```