

Practica 02

Estructura de Datos y Algoritmos 1

Josué Alexis Campos Negrón
josue.campos@cimat.mx

Universidad de Guanajuato
10 de febrero del 2023

Fecha de entrega: Viernes 17 de febrero.

Problema

Minecraft es un juego de supervivencia caracterizado por su estructura de cubos y pixeles. En este podemos encontrar herramientas tales como las *espadas* de diferentes tipos y cada una de ellas con diferente durabilidad y daño.



Tu tarea será implementar la clase `minecraft_swords` considerando los siguientes puntos:

- Los tipos de espada que vamos a considerar estarán identificadas por una $ID \in \{1, 2, 3\}$ y sus propiedades se presenta en la siguiente tabla.

Propiedades/ID	1	2	3
Material	Madera	Hierro	Diamante
Durabilidad	59	250	1561
Daño	4	6	7

- La clase **debe tener** las siguientes variables:
 - name:** Almacena el nombre de la espada.
 - damage:** Almacena los puntos de daño de la espada. El usuario debe ser posible de acceder a este atributo.
 - durability:** Almacena la durabilidad de la espada.
 - mobs:** Es un arreglo dinámico que almacena el nombre de los enemigos con los que la espada ha acabado. Se te asegura que la espada no enfrentará a más de 100 enemigos.

Utiliza tu criterio para definir si las variables serán de tipo `private` o `public`. Considera utilizar variables extras que te ayuden a cumplir el propósito de las funciones de la espada.

- El **constructor** recibe dos parámetros los cuales son `type` y `sword_name` los cuales definen el tipo de espada según el ID y el nombre de la espada.
- El **destructor** debe liberar toda la memoria dinámica utilizada en la clase para evitar fugas de memoria.
- La clase **debe tener** las siguientes funciones:
 - `kills()`: Imprime el mensaje

`name` ha acabado con `num_kills` enemigos.

donde **name** es el nombre de la espada y **num_kills** el número de enemigos con los que ha acabado. También debe retornar el valor de **num_kills**.

- **life()**: Imprime el mensaje

name le queda **durability** puntos de durabilidad.

donde **name** es el nombre de la espada y **durability** la durabilidad actual de la espada. Recuerda que por cada golpe que realiza a un enemigo este decrementa su durabilidad en uno. También debe retornar el valor de **durability**.

- **hits(string enemy, int enemy_life)**: ¡Un enemigo se ha presentado y debes acabar con él!. Esta función recibe como parámetro las variables **enemy** y **enemy_life** los cuales definen el nombre del enemigo y los puntos de vida de este respectivamente. Debes imprimir el mensaje

name acabo con el **enemy** en **count_hits** golpes.

donde **name** es el nombre de la espada, **enemy** el nombre del enemigo y **count_hits** la cantidad de hits necesaria para acabar con el enemigo. Recuerda que solo puedes acabar con un enemigo si tienes durabilidad suficiente, en caso de que no puedas acabar con el enemigo imprime el siguiente mensaje

name no pudo acabar con el **enemy** porque se ha roto.

En este caso no debes retornar ningun valor.

- **history()**: Esta funcion presenta la lista de enemigos con los que ha acabado de la siguiente manera

name ha acabado con los siguientes enemigos: **enemy1 enemy2 ... enemyn**

donde **name** es el nombre de la espada y **enemyi** es el nombre del enemigo *i* con el que ha acabado considerando que ha acabado con *n* enemigos. Recuerda utilizar el arreglo **mobs**. En este no debes retornar ningun valor.

Entregable

Se entrega un archivo comprimido **.zip** con el siguiente formato **Apellido1Apellido2 Practica01.zip** el cual contiene un folder con los siguientes documentos.

- Un reporte tipo **pdf** con nombre **Apellido1Apellido2 Practica01.pdf**. En la sección **Reporte** se detalla los requisitos del **pdf**.
- Un archivo **.cpp** con nombre **sword.cpp** en donde vendrá la implementación de la clase **minecraft_swords**. En la sección **Código** se detalla los requisitos de las implementaciones.
- Imagenes de evidencia del **output** que demuestre el correcto funcionamiento de las implementaciones. Las imagenes deben tener nombres del estilo **evidencia.IMGn.png** donde **n** es el número de imagen.

Reporte

El reporte consta de cinco secciones las cuales son:

1. **Función hits()**: Explica la lógica de tu implementación de la función **hits**.
2. **Cuestionario**: Responde las siguientes preguntas brevemente que demuestre tu comprensión sobre las clases.
 - ¿Qué es la **programación orientada a objetos**?
 - ¿Cuáles son las diferencias entre clases y objetos?
 - ¿Cuál es el funcionamiento del **constructor** y **destructor** en las clases?

- ¿Cuál es la diferencia entre las variables de tipo `private` y `public`?
- 3. **Problemas encontrados:** Mención de los problemas encontrados (si es que hubo) al momento de implementar los algoritmos o comprenderlos.
- 4. **Conclusión:** Resumen breve de lo aprendido y menciona ejemplos que se puedan implementar con clases.
- 5. **Referencias:** Enunciar las referencias utilizadas para la realización de la practica.

Código

Input: En la primera linea el programa recibe un entero T y un string S que representan el ID de la espada a crear y el nombre de la espada respectivamente. En la siguiente linea tenemos un entero N que representa el número de preguntas a responder. Seguidamente tenemos N lineas con un caracter C que define el tipo de pregunta dado por

- K: Manda a llamar a la función `kills()`.
- D: Manda a llamar a la función `life()`.
- H: Recibe dos elementos más los cuales son `enemy` y `enemy_life`. Luego, manda a llamar la función `hits()` que envia como parámetro los elementos antes recibidos.
- Y: Manda a llamar a la función `history()`.

Output: La salida esperada son las impresiones al momento de llamar a las funciones.

Se adjunta archivos `.txt` con un ejemplo de **input** y otro con el **output** esperado para los algoritmos. Comentar las funciones de las clase explicando su funcionamiento.

Consejo

Para la realización de esta tarea requiere investigación acerca de las clases. Las siguientes ligas serán de utilidad.

- Programación ATS. Tiene una lista de reproducción sobre programación en C++. Apartir del tutorial 126 habla sobre las clases.
- W3school Explica de manera breve los conceptos sobre las clases.
- GeeksforGeeks es una página de gran utilidad sobre conceptos de programación y algoritmos. Recomendando buscar dudas del estilo como “Destructors in C++ [geeksforgeeks](#)”.

También aconsejo realizar primero el cuestionario para entender bien los conceptos y luego realizar la clase.