

Tarea de Programación en C++: Simulador del Robo del Tesoro del Dragón

Objetivo

El objetivo de esta tarea es practicar el uso de **cabeceras personalizadas** en C++, funciones, arreglos y el control de flujo de un programa. Además, se introduce el concepto de **generación de números aleatorios** con `srand()` y `rand()`.

Descripción del Problema

Se desea construir un **simulador interactivo** en el que un jugador intenta robar un tesoro protegido por un dragón dormido. El escenario se representa de manera simplificada:

- El jugador comienza en la entrada de la cueva (posición 0).
- El tesoro se encuentra en la posición 15.
- El dragón tiene una barra de sueño inicial de 30.
- Todos los turnos involucran el lanzamiento de un dado de 6 caras, donde el resultado se genera de manera aleatoria usando `rand()`.

El juego se divide en tres fases claramente diferenciadas:

Fase 1: Acercamiento al tesoro

- El jugador avanza hacia el tesoro lanzando un dado de avance de 6 caras.
- Se lanza un dado de ruido de 6 caras que disminuye la barra de sueño del dragón según el valor obtenido.
- Esta fase termina cuando el jugador alcanza la posición del tesoro (posición 15).

Fase 2: Abrir el cofre

- Mientras el jugador está en la posición del tesoro y el cofre no se ha abierto, se lanza un dado de ruido de 6 caras.
- Si el dado es **impar**, el cofre se abre exitosamente.
- Si el dado es **par**, el cofre permanece cerrado y el dragón pierde sueño según el valor del dado.

Fase 3: Escape con el tesoro

- Una vez abierto el cofre, el jugador debe regresar a la entrada de la cueva (posición 0).
- En esta fase, cada turno:
 - Se lanza un dado de avance de 6 caras para retroceder.
 - Se lanza un dado de ruido de 6 caras que disminuye la barra de sueño del dragón.
- La misión se completa si el jugador llega a la entrada con el cofre abierto antes de que el dragón se despierte.

Fin del juego

El juego termina si:

- La barra de sueño del dragón llega a 0 o menos → el jugador falla.
- El jugador abre el cofre y regresa a la entrada → misión completada.

Requerimientos Técnicos

- Crear una **cabecera personalizada dragon.h** donde se definan todas las funciones necesarias:
 - Funciones de avance, lanzamiento de ruido, abrir cofre, verificar sueño y comprobar escapada.
- Implementar la lógica en **dragon.cpp** y utilizarla desde **main.cpp**.
- El programa debe ser interactivo:
 - Pedir al usuario que presione Enter para cada turno.
 - Mostrar los resultados paso a paso: avance, retroceso, ruido, intento de abrir cofre y sueño restante del dragón.
- Inicializar la semilla para los números aleatorios al inicio del programa usando **rand(time(NULL))** para obtener resultados diferentes en cada ejecución.

Sugerencias

- Para usar números aleatorios:

```
#include <cstdlib>
#include <ctime>
srand(time(NULL)); // inicializa la semilla
int dado = rand() % 6 + 1; // devuelve 1..6
```