progettino INFO 3A: C++

Michele Dentella (1063244)

nov 2021

1. Introduzione

Piccolo gioco scritto in C++, utile al solo scopo di implementare alcuni concetti visti a lezione.

Il progetto è volutamente sintetico (specialmente nelle "dinamiche di gioco").

1.1 regole del gioco

Il gioco è composto da un *Protagonista* e da dei *Nemici*.
Il protagonista ha delle capacità che servono per sconfiggere i nemici; queste aumentano quando si elimina un nemico. A seconda di quale nemico viene sconfitto cambiano le capacità incrementate.
I nemici sono di diverso tipo, ogni tipo richiede abilità diverse per essere sconfitto.
Il Protagonista ha una vita che viene ridotta ogni volta che non riesce a sconfiggere un nemico.

Il gioco finisce quando i nemici sono tutti sconfitti o quando il giocatore esaurisce la vita.

2. Struttura delle Classi

ci sono due classi base: Nemico e Protagonista.

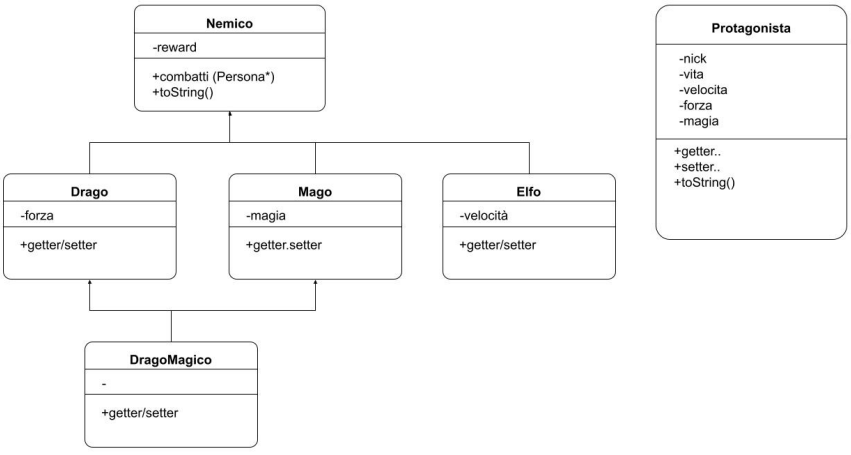

Figure 1: struttura classi

2. Funzionalità

le funzionalità sono prese dal sito del corso

2.1 Costruttore/distruttore

Esiste un solo costruttore per ogni classe, questo imposta i valori passati come argomenti. Ogni costruttore di classi ereditate chiama il costruttore della classe padre per impostare i campi ereditati. I distruttori sono virtual e stampano una stringa identificativa in modo che si capisca quando vengono chiamati.

2.2 campi pubblici e privati

Tutti i metodi sono pubblici perchè principalmente utili al di fuori degli oggetti; non avendo oggetti particolarmente articolati non è stato necessario usare metodi privati (quindi accessibili solo all'interno della classe). Gli attributi sono protected, in modo che siano visibili anche alle classi figlie (e non accessibili dall'esterno). Questi sono comunque acceduti solo tramite metodi.

2.3 membri virtual e non

Sia i distruttori che il metodo "combatti" sono virtual, in modo che, grazie all'overriding, sia eseguito il metodo giusto in base al tipo di oggetto (determinato a runtime) che esegue il metodo.

2.4 ereditarietà multipla

Vedi classe DragoMagico. Usata ereditarietà virtual public nelle classi padre, in modo da evitare "confusioni" con i nomi dei metodi.

2.5 STL (struttura dati e algoritmo)

I nemici sono memorizzati in un Vector, vengono usati gli iteratori necessari per la stampa di tutti i nemici e per la selezione del nemico da affrontare.

2.6 smart pointers

Sia il protagonista che i singoli nemici sono creati con smart pointer di tipo unique, così che sia gestita in automatico l'eliminazione. Sono state aggiunte delle

stampe ai distruttori per evidenziare i momenti in cui vengono automaticamente eliminati gli oggetti.