

Esame di Programmazione 2

Geraldo da Ravina (TN) è un combattente che subisce mutazioni. L'obiettivo di questo progetto è scrivere il codice che permette di visualizzare le stats Geraldo, di visualizzare le mutazioni che sono attive su Geraldo, e di applicargli diverse mutazioni.

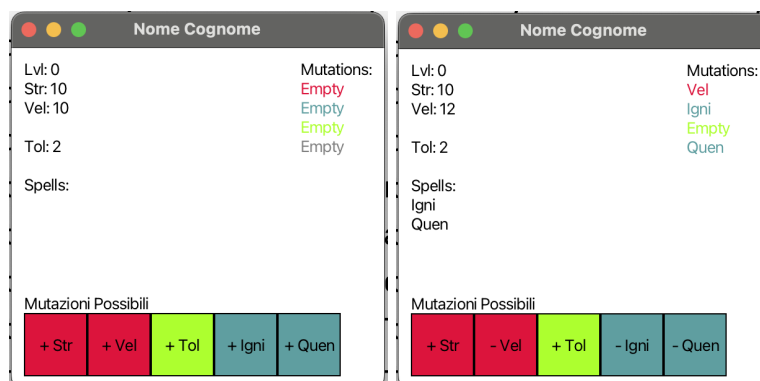
La lista seguente presenta le mutazioni esistenti, divise per tipologia. Ogni tipologia di mutazione ha delle mutazioni specifiche che modificano le stats di Geraldo.

Mutazioni di combattimento	Mutazioni di magia	Mutazioni di pozioni
<ul style="list-style-type: none">• +2 forza• +2 velocità	<ul style="list-style-type: none">• Magia di fuoco• Magia difensiva	<ul style="list-style-type: none">• +2 Tolleranza

Le stats di Geraldo sono: Livello (inizializzato a 0), Forza (inizializzato a 10), Velocità (inizializzato a 10), Tolleranza alle pozioni (inizializzato a 2), Lista di magie conosciute (inizialmente vuota).

Si programmi una GUI come in figura (a sinistra si vede l'applicazione appena parte, a destra l'applicazione dopo aver aggiunto delle mutazioni).

La GUI visualizza le stats di Geraldo (a sinistra, incolonnate), le mutazioni applicate (a destra, incolonnate) e le mutazioni disponibili (in basso, su una riga). Le mutazioni si possono aggiungere o rimuovere da Geraldo cliccando sulle mutazioni disponibili in basso: ciò ha un effetto diretto



sulle stats visualizzate (appena si aggiunge la mutazione +2 forza, la GUI mostra 12 in forza, se si rimuove quella mutazione, la GUI mostra 10). Se Geraldo non ha una mutazione, la mutazione corrispondente nella lista delle disponibili appare con un '+', altrimenti appare con un '-'. Se si clicca su una mutazione con un '+', viene aggiunta a quelle di Geraldo, se si clicca su una con un '-', viene rimossa da quelle di Geraldo. Una volta aggiunte le mutazioni a Geraldo, queste vengono messe in degli slot mutazione appropriati, Geraldo ha uno slot per ogni tipologia di mutazione, più uno slot generale (inizialmente vuoto, ma colorati come il tipo di mutazione che possono contenere, grigio nel caso generale). Quando viene aggiunta una mutazione, essa va nello slot della sua tipologia, se questo slot è già preso, allora la mutazione va nello slot generale, se anche quest'ultimo è preso, non viene aggiunta la mutazione.

Si scriva codice generale, che funzioni anche se si vanno ad aggiungere mutazioni, o se si aggiungono slot.

- Si usi, quando evidente/utile, una gerarchia di ereditarietà.
- Si usino costanti ove ragionevole, e si badi alla pulizia del codice, evitando duplicazioni e codice inutilmente complesso.
- La disposizione degli elementi grafici deve rispecchiare quanto mostrato; tuttavia, ciò è considerato meno importante rispetto alla corretta funzionalità dell'interfaccia e del resto dell'applicazione.
- Si richiede il class diagram UML. È sufficiente la vista di alto livello (no attributi/metodi) ma devono essere mostrate le associazioni più importanti, oltre a quella di ereditarietà. Il diagramma UML deve essere consegnato su un foglio di protocollo su cui sono indicati nome, cognome, numero di matricola.