

Simulazione di Combattimenti tra Animali in Java

Di ogni **Animale** è necessario sapere la **velocità** (VEL), la **forza** (FOR), la **vita** (PV o HP), l'**energia** (ENER), l'**attacco** (ATK), la **difesa** (DEF), i **danni** (DANN), l'**altezza**, la **lunghezza**, la **larghezza** e il **peso**.

Tutti gli **Uccelli** hanno la *beccata* (ATK +), inoltre possono essere suddivisi in:

Rapace: *picchiata* (DANN + , ENER -) e *volare* (VEL +)

Struzzo: *testa_sotto_la_sabbia* (DEF + , al 50% recupera anche 1 ENER).

Tutti gli **Anfibi** hanno *vantaggio_in_acqua* (ATK + , DANN +), inoltre possono essere suddivisi in:

Coccodrillo: *presa_in_morso* (se l'attacco ha successo VEL e ATK avversario diventano 0, dal prossimo turno DANN + , ENER -)

Serpente: *veleno* (DANN + , e con 5% probabilità è KO istantaneo) e *morso_rapido* (ATK +).

Tutti i **Predatori** hanno *graffio* (DANN +), inoltre possono essere suddivisi in:

Leone: *agilità* (VEL +) e *agguato* (usa 2 volte *graffio* con DANN +1 (ogni graffio consuma 1 ENER)).

Orso: *pelliccia_spessa* (DEF +) e *possanza* (FOR +).

Il combattimento si svolge in un **Terreno** che può essere suddiviso in:

- **Pianura** (nessun bonus)
- **Fiume** (*vantaggio_in_acqua* agli anfibi, vedi sopra)
- **Foresta** (VEL + per animali piccoli, VEL - - per animali **grandi**: un animale è considerato grande se il suo volume è superiore a 8m³ o se pesa più di 250kg, altrimenti è considerato piccolo).

Ogni attacco richiede 1 energia. Al suo turno un animale fa un attacco normale o un'abilità.

Se un animale all'inizio del suo turno **non ha più energia lo scontro termina.**

L'attacco ha successo se **(ATK + FOR + VEL dell'attaccante + 1d20) > (DEF + VEL dell'avversario)**, se l'attacco va a segno i danni inflitti sono pari a **(DANN + FOR)**, altrimenti il colpo manca (0 danni).

Vince l'animale che resta in vita. In caso di fine combattimento per energia esaurita: **se** ha combattuto **un predatore allora vince l'altro** animale, altrimenti **vince** l'animale che è rimasto con **più punti vita** (PV).

Si richiede:

- **una rappresentazione UML delle classi** di una possibile soluzione al problema descritto.
- **l'implementazione in linguaggio Java del diagramma UML** progettato.
- la realizzazione delle seguenti funzionalità:
 - 1) Calcolare **quanti** animali di una specie a scelta ci stanno in una gabbia **4m x 4m x 4m** e che può reggere fino a **9000 kg**.

- 2) Implementare il **combattimento tra 2 animali qualsiasi**.