Mar 11, 2019

1. 1 - Rappels du cours - Ne sautez pas cette étape

Etudiez l'exemple présent dans le fichier exemple_chat.rb que vous pouvez exécuter pour valider ce qui est avancé dans les commentaires. (exécutez le, modifiez le mais surtout, comprenez le !) Avis aux filous : Si vous sautez cette étape, on vous fera flageller avec des orties.

2. 2 - Entrée en matière - Générateur de getter et de setter

Améliorez la classe *Object* avec deux nouvelles méthodes de classe attr_getter et attr_setter qui fonctionnent comme respectivement comme attr_reader et attr_writer. Indice : regardez les méthodes Object#instance_variable_get et Object#instance_variable_set dans la documentation. Avis aux filous : Ne pas utiliser d'alias.

Pour rappel un getter et un setter d'âge (#age) s'écriront :

```
def age
  @age
end

def age=(value)
  @age = value
end
```

3. 3 - YAC - Yet Another Creature

Testez vos méthodes attr_getter et attr_setter dans une classe Creature qui aura un nom (#nom) et des points de vie (#pv). Nommez bien les méthodes comme indiqué.

Ensuite ajoutez les méthodes suivantes sur la classe Créature :

- #tanker(degats) : Cette méthode simule une armure ou une esquive et permet de retourner les dégats qu'encaissera réellement une créature lorsqu'elle aura se fera attaquer Par défaut sur la créature, cette méthode renvoie les dégats tels quels.
- #encaisser(degats) : Cette méthode de faire en sorte que la créature tank les dégats grâce à la méthode #tanker et soustrait des pv de la créature la valeur de retour de ce qu'elle aura tanké.
- #force : Cette méthode renvoie la force d'une attaque de la créature. Par défaut sur la créature, cette méthode renvoie 1.
- #frapper(other_creature) : Cette méthode fait en sorte qu'une créature en frappe une autre et lui fait encaisser sa force physique.
- #mort? : Cette méthode renvoie vrai si la créature a des pv <= 0

4. 4 - YAML - Génération de classes

Lisez le fichier creatures.yml qui définit des sous classes de créatures et leur manière de calculer leur force et leur façon de tanker. Indice : Pour parser ce fichier, utilisez YAML.load(string) (consulter la documentation).

Pour chacune des sous classes de créature décrite :

- Créer une sous classe de créature
- Associer la constante qui correspond à son nom à son object
- Redéfinir ses méthodes #tanker(degats) et #force suivant la description du fichier (attention à la prise d'arguments pour tanker)

Testez en bricolant des sous classes de créatures chargées du fichier.

5. 5 - Round one, fight!

Ecrivez un module Arene qui dispose d'une méthode de classe #duel_a_mort(nom, dueliste_1, dueliste_2) qui affiche le nom du combat, fait

combattre les deux créatures duelistes et affiche le vainqueur et ses points de vie restant.

6. 6 - Yet Another YAML - Instanciation de créatures

Lisez le fichier duels.yml, pour chacun des combats décrits, instanciez les bonnes créatures et faites les combattre dans l'arène !

Supports de cours pour l'IUT
francois.delobel@uca.fr

TPs mis à disposition le plus simplement possible pour mes étudiants.