



SCIENCES
SORBONNE
UNIVERSITÉ

MU4RBI01 Programmation objet Python : Projet Pokémon

Etudiant :

Sonia Slimi

Haonan Yuan

Valentin Cointrel

(Groupe n°34)

Enseignant : Thomas Dietenbeck

I – Lien entre les différentes classes

Mise en contexte : La classe JcJ correspond au combat joueur contre IA ou joueur contre joueur. Si on sélectionne la difficulté 1, 2 ou 3 on joue contre l'IA, si on sélectionne la difficulté -1 on joue contre un autre joueur humain.

La classe JcE correspond au combat joueur contre environnement autrement dit contre un Pokémon sauvage.

On peut jouer au jeu en exécutant le « main ».

La classe **JcJ** contient des références à plusieurs objets de la classe **dresseur**. La classe **JcJ** fonctionne donc avec des objets de la classe **dresseur**. Nous avons donc mis une relation d'association entre la classe **JcJ** et la classe **dresseur**.

La classe **JcE** contient des références un objet de la classe **dresseur** et à un objet de la classe **Pokémon**. La classe **JcE** fonctionne donc avec des objets de la classe **dresseur** et de la classe **Pokémon**. Nous avons donc mis une relation d'association entre la classe **JcE** et la classe **dresseur** ainsi qu'entre la classe **JcE** et la classe **Pokémon**.

La classe **Dresseur** contient des références à plusieurs objets de la classe **Pokémon**. La classe **Dresseur** fonctionne donc avec des objets de la classe **Pokémon**. Nous avons donc mis une relation d'association entre la classe **Dresseur** et la classe **Pokémon**.

Toutes les compétences, qu'elles soient des compétences d'attaque ou de défense, ont des attributs en commun. Elles ont toutes un **cout**, une **description**, un **élément**, un **nom**. La différence au niveau des attributs est que les compétences attaque ont des attributs **puissance**, **précision** en plus et les compétences défense ont des attributs **valeurSoin** et **valeurEnergie** en plus. Etant donné le grand nombre d'élément en commun entre les compétences attaque et les compétences défense, nous avons décidé de mettre un lien d'héritage entre la classe **Compétence** (la classe mère) et chacune des classes filles (**ComAttaque** et **ComDefense**).

La 2nd différence entre les compétences d'attaque et de défense est dans la définition de la méthode **utilCompétence**. Les compétences d'attaque et de défense ont toutes les deux cette méthode mais elle est définie différemment. Pour cette raison nous avons défini la méthode **utilCompétence** comme abstraite dans la classe **Compétence** et par conséquent la classe **Compétence** est une classe abstraite. La méthode **utilCompétence** est donc implémentée par les classes filles **ComAttaque** et **ComDefense**.

La classe **Pokémon** contient des références à plusieurs objets de la classe **Compétence**. La classe **Pokémon** fonctionne donc avec des objets de la classe **Compétence**. Mais les objets de classe **Compétence** n'existent pas sans un objet de classe **Pokémon**. Si on détruit un objet **Pokémon** alors ses compétences disparaissent, elles ne peuvent pas « exister » sans le Pokémon. La classe **Pokémon** détient donc des objets de classe **Compétence** et ne partage pas ces objets avec d'autres classes. Nous avons donc utilisé une relation de Composition entre la classe **Pokémon** et la classe **Compétence**.

II – Choix d'encapsulation

Tous les attributs des classes **compétences** (classe mère et filles) ainsi que de la classe **pokemons** sont en privés car on ne veut pas qu'un utilisateur/joueur puisse modifier un élément ou statistique (peu importe lesquels) de ses Pokémons car cet utilisateur pourrait alors tricher en augmentant les dégâts ou la vie de ses Pokémons par exemple.

Toutes les méthodes des classes **compétences** (classe mère et filles) ainsi que de la classe **pokemons** sont en publiques car ce sont des méthodes appelées en dehors de la classe telles que des utilisations de compétences ou des affichages (toString).

Tous les attributs de la classe **dresseur** sont en privés car on ne veut pas qu'un utilisateur/joueur puisse modifier son nom ou sa liste de Pokémon ou son deck de Pokémon. Son deck de Pokémon peut être modifié mais seulement lorsque l'utilisateur se voit proposer par le jeu la possibilité de le faire. Donc en dehors de ces instants particuliers le deck reste privé pour en empêcher son accès comme c'est le cas avec son nom ou sa liste de Pokémon.

On ne veut pas que l'utilisateur ait accès à ces attributs car il pourrait alors tricher en se donnant des Pokémon qu'il n'a pas par exemple ou en modifiant son deck en plein combat ce qui lui permettrait de jouer bien plus que 3 Pokémon au sein d'un même combat.

Toutes les méthodes de la classe **dresseur** sont en publiques car ce sont des méthodes utilisées par l'utilisateur telles que des changements de Deck, la possibilité de sauvegarder ou des affichages. Ces méthodes doivent donc pouvoir être appelées depuis l'extérieur de la classe, elles sont donc publiques.

Les attributs **dresseur**, **difficulté**, **pokemon**, **pokemonActive** de la classe **JcJ** sont en privés car un joueur ne doit pas pouvoir tricher en modifiant la difficulté de l'IA, ni changer les dresseurs ou Pokémon impliqué dans le combat. Les attributs gagnant et perdant étant utilisés en dehors de la classe ont les conserves en public.

Les méthodes **AI**, **changePokemon**, **évolution**, **forfait**, **règlement**, **réinitialisation** et **utilCompetence** de la classe **JcJ** sont en privées car ce sont des méthodes qui ne doivent pas être accessibles en dehors de la classe. Un utilisateur pourrait tricher en modifiant le comportement de l'AI ou pourrait réinitialiser les statistiques de certains Pokémon en plein combat par exemple. La seule méthode publique de cette classe est la méthode **combat** car elle est appelée en dehors de la classe par le joueur, c'est au joueur de lancer le combat.

Les attributs **dresseur**, **pokemon**, **pokemonActive** de la classe **JcE** sont en privés car un joueur ne doit pas pouvoir tricher en changeant les dresseurs ou Pokémon impliqués dans le combat. Les attributs gagnant et perdant étant utilisés en dehors de la classe ont les conserves en public.

Les méthodes **capturer**, **changePokemon**, **évolution**, **forfait**, **règlement**, **réinitialisation** et **utilCompetence** de la classe **JcE** sont en privées car ce sont des méthodes qui ne doivent pas être accessibles en dehors de la classe. Un utilisateur pourrait tricher en réinitialisant les statistiques de certains Pokémon en plein combat par exemple. La seule méthode publique de cette classe est la méthode **combat** car elle est appelée en dehors de la classe par le joueur, c'est au joueur de lancer le combat.