Projet Jeu Pokémon

Version : 1.0

A rendre un fichier compressé (zip, 7z, rar) contenant :

- L'ensemble des fichiers du projet.

Ce projet peut-être réaliser seul ou par groupe de deux.

Nommer le fichier nom1.nom2.zip et mettre votre projet compressé dans le dossier de livrable prévu à cet effet, ou envoyer le fichier à l'adresse nicolas.chevalier@mail-formateur.net.

Création d'un mini jeu de Pokémon

Instructions

Notre jeu de Pokémon consistera à modéliser sous forme d'objet un combat entre deux dresseurs de Pokémon. Certaines règles sont volontairement simplifiées pour rendre le jeu plus simple à réaliser.

Vous trouverez sur internet des informations relatives au Pokémon comme leurs types, les différents Pokémons, les attaques, les objets, …

Exemple de site ou vous pouvez trouver des informations :

http://www.pokemontrash.com/pokedex/liste-pokemon.php
http://www.pokemontrash.com/pokedex/liste-attaques.php
http://www.pokepedia.fr/Capacit%C3%A9#Puissance.2C_Pr.C3.A9cision_et_PP

Travail à faire :

Créer un dossier nommé jeu_pokemon qui contiendra les fichiers Python de l'application.

Créer les fichiers :

- main.py : fonction principale du programme
- jeu.py : contient la classe Jeu
- joueur.py : contient la classe Joueur
- pokemon.py : contient la classe Pokemon
- attaque.py : contient la classe Attaque

Vous utiliserez le module rich pour améliorer l'affiche avec de la couleur et d'autre composant disponible avec cet API : https://rich.readthedocs.io/en/stable/introduction.html

Mettez en place la gestion des exceptions afin que le programme ne plante pas lors de mauvaise saisie faite par l'utilisateur.

Indication pour les classes :

Vous pouvez utiliser les classes listées si dessous dans votre programme.

La classe Jeu

Cette classe prendra en charge la gestion de tout le jeu. Le jeu se fera entre deux joueurs humains. Pour commencer un jeu, le premier joueur devra saisir son nom puis acheter avec l'argent qui lui sera attribué trois Pokémons. Le joueur deux fera ensuite la même chose. Chaque joueur sera représenté par une instance d'une classe Joueur qui sera à créer. Une fois cette phase de configuration terminée, le jeu peut commencer. Le combat sera composé de 3 rounds (le premier Pokémon du joueur 1 vs le premier Pokémon du joueur 2, et ainsi de suite). Chaque joueur choisit une attaque, et c'est le Pokémon ayant la plus grande vitesse qui attaquera en premier. La manche se termine si le Pokémon est K.O. (PV == 0) ou s'il ne peut plus utiliser d'attaques (les PP de toutes les attaques sont nuls). Le joueur qui gagnera au moins deux manches sera déclaré vainqueur.

La classe jeu possèdera :

- joueurs : liste contenant des joueurs (2 pour notre jeu).
- Méthode jouer : démarre le jeu, créé 2 joueurs et demande pour chaque joueur de saisir leur nom et d'acheter leurs Pokémons. Met en place l'algorithme du jeu avec les 3 rounds. Indique le joueur ayant gagné. Peu demander au joueur s'il veut rejouer.

Le programme principal servira pour créer un jeu et lancer le jeu en appelant la méthode Jouer.

Classe Joueur

- Nom : nom du joueur
- Manche gagnée : nombre de manche gagné par un joueur (sera utilisé pour déterminer le gagnant à la fin des 3 rounds).
- Argent : argent attribué au joueur pour acheter des Pokémons
- Pokemons : liste des Pokémons du joueur

Ajouter un initialiseur qui permettra de renseigner et d'initialiser les propriétés d'un joueur.

De plus la classe Joueur disposera de plusieurs méthodes :

- Une méthode **choisir_pokemon** qui va créer et affiche une liste de Pokémons dans laquelle le joueur pourra choisir ces 3 Pokémons. Chaque Pokémon aura ces attaques de créer.
- Une méthode **ajouter_pokemon** qui va ajouter un Pokémon à la liste. Elle prendra donc un Pokémon en paramètre (sous forme de pointeur).
- Une méthode **choisir_attaque** qui va afficher la liste des attaques du Pokémon engagé dans le combat et qui va demander au joueur de choisir quelle attaque utiliser pour attaquer le Pokémon adverse. Elle aura un Pokémon en paramètre et elle retournera l'attaque choisie pour ce Pokémon.
- Une méthode **recuperer_pokemon** qui va retourner le Pokémon du joueur qui sera engagé dans le combat. Prend en paramètre le numéro du Pokémon (1, 2, ou 3) à récupérer.
- Une méthode **afficher_pokemons** qui affichera les différents Pokémon (Nom, Type(s), PV, Niveau, …)
- Une méthode **afficher** qui affichera les différentes informations d'un Joueur (Nom, Manche gagnée, Argent, …)

Classe Pokemon

Nous n'allons pas implémenter toutes les fonctionnalités que l'on trouve dans les jeux du commerce, mais nos Pokémons auront quand même une bonne partie d'entre elles. Ils possèderont :

- Nom : nom du Pokémon qui sera affiché.
- Prix : prix du Pokémon
- Types : un Pokémon possède un ou deux types (feu, dragon, psy, …) (pas besoin de liste ici, créé deux Type).
- Points de vie : chaque Pokémon dispose un nombre de point de vie qui sera différent suivant le Pokémon. Quand un Pokémon est K.O. quand sont PV est à 0.
- Niveau : chaque Pokémon a un niveau qui lui permet par la suite d'évoluer et qui est utilisé pour calculer le dégât occasionné par une attaque.
- Attaque qui est une valeur entière.
- Attaque spéciale qui est une valeur entière.
- Défense (diminue pas sauf avec certaine attaque) qui est une valeur entière
- Défense spéciale (si attaque spéciale) qui est une valeur entière.
- Vitesse : qui représente la rapidité d'attaquer d'un Pokémon.

- Attaque(s) : liste contenant les attaques du Pokémon. Chaque Pokémon dispose d'une à quatre attaques.

Ajouter un initialiseur qui permettra de renseigner une partie des propriétés d'un Pokémon.

De plus la classe Pokémon disposera de plusieurs méthodes :

- Une méthode **ajouter_attaque** qui va ajouter une attaque à la liste. Elle aura donc une attaque en paramètre (sous forme de pointeur).
- Une méthode **attaquer** qui prendra en paramètre le Pokémon à attaquer, ainsi que l'attaque utilisée.
- Une méthode **est_ko** indiquera en retournant un boolean si le Pokémon est KO ou pas.
- Une méthode **afficher_attaques** qui affichera les différentes attaques d'un Pokémon (Nom, Type, PV, Catagorie, …)
- Une méthode **afficher** qui affichera les différentes informations d'un Pokémon (Nom, Type(s), PV, Niveau, …)

Classe Attaque

Une attaque sera représentée également par une classe qui possèdera les informations suivantes :

- Nom : nom de l'attaque qui sera affiché.
- Type : type de l'attaque (normale, feu, eau, …). A noter que si un Pokémon de type Feu, utiliser une attaque de même type, alors la puissance de cette dernière sera multipliée par 1.5 (STAB).
- Catégorie attaque : catégorie de l'attaque (physique ou spéciale).
- Précision : précision de l'attaque qui sera utilisé pour le calcul des dégâts d'une attaque (il s'agit d'un pourcentage).
- Puissance : il s'agit de la puissance de base de l'attaque.
- PP : nombre de fois que l'on pourra utiliser une attaque.

Ajouter un initialiseur qui permettra de renseigner les propriétés d'une attaque.

Il existe plusieurs catégories d'attaques : les attaques physiques et les attaques spéciales. Chacune de ces deux catégories correspond à des propriétés spécifiques du Pokémon attaquant et du Pokémon cible :

- Attaque physique : les dégâts sont calculés en fonction de l'attaque du Pokémon attaquant et de la défense du Pokémon attaqué ;
- Attaque spéciale : les dégâts sont calculés en fonction de l'attaque spéciale et de la défense spéciale.

Le nombre de dégâts occasionné par une attaque sera effectué par une méthode **calculer_degats**. Cette méthode prendra en paramètre un Pokémon attaquant et un Pokémon attaqué. Elle retournera le résultat du calcul des dégâts.

La formule de calcul appliqué est la suivante :

$$PV_{perdus} = \lfloor \left(\frac{(Niv \times 0.4 + 2) \times Att \times Pui}{Def \times 50} + 2 \right) \rfloor \times CM$$

CM étant un coefficient multiplicateur résultant des paramètres suivants :

- Le STAB
- La précision de l'attaque

Exemple :

Un Zekrom de niveau 100 ayant 396 en Attaque utilise l'attaque Éclair Croix qui a une puissance de 100 et une catégorie Physique. La cible est un Reshiram ayant 299 en Défense. Comme Éclair Croix est de même type que l'un des deux types de Zekrom (type Électrik), alors elle bénéficie du STAB.

Le nombre maximal possible de PV perdus est donc :

(((100 * 0.4 + 2) * 396 * 100) / (299 * 50) + 2) * 1.5 x 1 = 169 PV

Dans ces conditions, Reshiram perd 169 PV.

On pourra également afficher les informations d'une attaque par la méthode **afficher**.

Précision et indications

Vous n'allez pas créer des centaines d'objets pour représenter tous les Pokémon et les attaques. Choisissez 5 ou 6 Pokémons et une dizaine d'attaques pour les implémenter.

Pour les types, vous devez implémenter au moins ceux correspondants aux Pokémon et attaques que vous avez choisi.

Vous avez la liberté du choix des prix des différents éléments.