

TP 0 - Pokemon Battle Arena

Ferdinand Mom

August 26, 2019

Consignes de rendu

A la fin de ce TP, vous devez rendre une archive respectant l'architecture suivante:

```
|__ toolbox/  
|__ |__ toolbox.py  
|__ doc/  
|__ |__ database.csv  
|__ |__ typeTable.csv  
|__ |__ output  
|__ img/  
|__ |__ #Beaucoup d'images, ne pas les supprimer.  
|__ music/  
|__ |__ battleTheme.ogg  
|__ |__ faintNoHp.wav  
|__ |__ hitDamage.wav  
|__ |__ selection.wav  
|__ |__ victory.ogg  
|__ main.py  
|__ pokemon.py  
|__ trainer.py  
|__ botTrainer.py  
|__ tp_0_pokemon_arena_battle.pdf
```

Pour faire ce TP, veuillez créer votre environnement virtuel en utilisant **virtualenv** puis l'activer. (cf le [Survival Kit > CheatSheet > Virtualenv](#)).

Ensuite, lancez le script `toolbox > toolbox.py`

Vous pouvez désormais commencer le TP. Bonne chance!

1 Introduction

L'objectif de ce TP est de créer les mécanismes de combat du fameux jeu Pokemon. Pour ceux qui ne savent pas ce que c'est, voici à quoi cela ressemble (Oui, après ce TP vous aurez la même chose que sur l'image :P).



Figure 1: Combat Pokemon.

Petit rappel des règles:

- Vous pouvez choisir tous les pokémons de la génération 1 (soit 151 pokémon). Ces derniers sont stockés sous forme d'images dans le dossier **img/**.
- Le but étant de gagner le combat à l'aide de votre équipe de pokémons.
- Vous pouvez avoir autant de pokémons dans votre équipe que vous le souhaitez.
- Chaque pokémon possède au minimum 1 attaque et au maximum 4 attaques.
- La partie se termine lorsqu'un des deux dresseurs n'a plus de pokémons à envoyer au combat.
- Chaque formule que vous allez rencontrer durant ce TP doit être transformer en type **int**.

Désormais, vous avez toutes les règles en tête, le TP peut donc commencer!

2 trainer.py

Créez le constructeur de la classe **Trainer**. Ajoutez l'attribut **nbPokemon** qui correspond au nombre de pokémons dans **teamPokemon**.

```
__init__(self, teamPokemon)
```

Ecrire la méthode suivante.

```
displayTeamPokemon(self)
```

Elle doit afficher les pokémons du dresseur dans la console comme ceci.

```
Your team:  
Pikachu  
Florizarre
```

Ecrire la méthode suivante qui retourne le pokemon choisi.

```
choosePokemon(self)
```

Elle demande à l'utilisateur d'entrer le nom d'un pokémon depuis la console. Vous devez gérer le cas lorsque le nom du pokémon n'est pas correcte en reposant à nouveau la question.

```
What do you want to choose?  
Pikachu
```

Ecrire la méthode suivante qui supprime un pokémon de l'équipe du dresseur seulement si ce dernier a au moins 1 pokémon dans son équipe.

```
deletePokemon(self, pok)
```

3 pokemon.py

Créez le constructeur de **Pokemon**. Levez l'exception suivante : ”**__init__()** **Pokemon()** -> **listSpell: not enough spell or too much**” lorsque nécessaire.

```
__init__(self, name, pokeType, lvl, listSpell)
```

Toujours dans le constructeur, ajoutez la ligne suivante.

```
baseStats = df1.loc[name]
```

Ensuite, ajoutez dans le constructeur les variables suivants:

- **hpIV**, un nombre entier aléatoire entre 0 et 15 inclus.
- **attackIV**, un nombre entier aléatoire entre 0 et 15 inclus.
- **defenseIV**, un nombre entier aléatoire entre 0 et 15 inclus.
- **speedIV**, un nombre entier aléatoire entre 0 et 15 inclus.

Ajoutez dans le constructeur la ligne suivante.

```
EV = int(65535 * self.lvl / 100)
```

Puis, implémentez les formules suivantes dans le constructeur.

$$HP = \left\lfloor \frac{[Base + IV + 50 + \frac{\sqrt{EV}}{8}] * LVL}{50} \right\rfloor + 10$$

- avec Base = baseStats[0]

$$ATTACK = \left\lfloor \frac{[(Base + IV) * 2 + \frac{\sqrt{EV}}{4}] * LVL}{100} \right\rfloor + 5$$

- avec Base = baseStats[1]

$$DEFENSE = \left\lfloor \frac{[(Base + IV) * 2 + \frac{\sqrt{EV}}{4}] * LVL}{100} \right\rfloor + 5$$

- avec Base = baseStats[2]

$$SPEED = \left\lfloor \frac{[(Base + IV) * 2 + \frac{\sqrt{EV}}{4}] * LVL}{100} \right\rfloor + 5$$

- avec Base = baseStats[3]

Pour en finir avec le constructeur, créez l'attribut **maxHp** qui correspond au *HP* du Pokemon.

Créez la méthode ci-dessous. Elle retourne **True** si le pokémon n'a plus de *HP* sinon **False**.

```
isKo(self)
```

Créez la méthode ci-dessous. Elle retournera le tuple suivant (**pouvoir_du_sort**, **type_du_pouvoir**).

```
attackPokemon(self, spell)
```

Pour information, **listSpell** est une liste de tuples. Chaque tuple est de la forme suivante: (**nom_pokemon**, **pouvoir_du_sort**, **type_du_pouvoir**).

Créez la méthode ci-dessous qui retire des *HP* au pokémon.

```
takeDamage(self, spellType, spellPower, attacker)
```

Ajoutez à cette méthode les lignes suivantes.

```
randomNb = randint(85, 100)/100  
typeEffectivness = df2.loc[self.pokeType, spellType]  
modifier = 8 * randomNb * typeEffectivness  
ratioAD = attacker.attack / self.defense
```

Puis, implémentez la formule suivante:

$$DAMAGE = \left[\frac{\left(\frac{2*LVL}{5} + 2\right) * spellPower * ratioAD}{150} + 2 \right] * modifier$$

4 botTrainer.py

Importez la classe **Trainer** du fichier **trainer.py** puis créez la sous-classe **BotTrainer** qui hérite de **Trainer**.

Créez le constructeur de **BotTrainer**.

```
__init__(self, teamPokemon)
```

Faites un **method overriding** sur **choosePokemon()**.

```
choosePokemon(self)
```

Cette méthode devra retourner un pokémon aléatoire depuis l'équipe du dresseur. Elle devra également afficher dans la console "*nom_du_pokemon_aléatoire, I choose you !*".

Ecrire la méthode suivante qui retourne un sort aléatoire parmi les sorts que le pokémon connaît.

```
chooseSpell(self, pok)
```

Ecrire la méthode suivante qui retourne un pokémon aléatoire depuis l'équipe du dresseur.

```
switchPokemon(self)
```

5 main.py

Importer les classes **Trainer**, **BotTrainer** et **Pokemon**.

Puis, complétez **initMain()**. Pour cela :

- Instanciez les pokémons que vous souhaitez.
- Instanciez 2 dresseurs nommés **trainer1** (instancié par la classe **Trainer**) et **trainer2** (instancié par la sous-classe **BotTrainer**).
- Affichez dans la console les pokémons de **trainer1**.
- Appelez **choosePokemon()** sur chaque trainer et attribuez le retour de la fonction aux variables **pok1** et **pok2** respectivement.

Si aucune erreur n'a été faite, vous pouvez désormais lancer le jeu en écrivant dans la console la ligne suivante: **python3 main.py**.

True knowledge exists in knowing that you know nothing.

- Socrates