

ATTENTION : Toute fraude ou plagiat de diagramme UML / de code (reprise partielle ou totale du diagramme / code d'un autre groupe ou d'Internet) sera sanctionnée par une **convocation devant le conseil de discipline**.

Changer le nom des variables ou ajouter des commentaires dans un code récupéré ailleurs reste du plagiat.

MODALITÉS

Ce projet a pour but de vous faire pratiquer l'ensemble des concepts objets vu dans cette UE et de les appliquer sur un sujet plus vaste qu'un simple TP. Il est à réaliser **par groupe de 3 en autonomie**. Aucun créneau de TP n'est prévu pour sa réalisation mais vos chargés de TP sont disponibles pour suivre votre travail (c'est à vous de les solliciter !).

Même si certains concepts peuvent vous manquer au début du projet (le sujet vous étant dévoilé avant la fin des cours magistraux), il reste néanmoins possible de travailler à une solution "classique" (*i.e.* non-objet mais avec des fonctions / librairies) qui sera enrichie par la suite avec ces nouveaux concepts (*e.g.* certaines librairies pourront devenir des objets, les fonctions deviendront des méthodes de classes, *etc.*). Ceci vous permettra de travailler régulièrement sur le projet et vous donnera du temps pour sa réalisation.

Les rendus attendus pour ce projet sont :

1. Diagramme de classes UML :

- (a) Votre diagramme UML devra modéliser le projet tout en utilisant l'ensemble des principes objet (liens entre classes, encapsulation, abstraction, polymorphisme).
- (b) Le projet permet l'utilisation de ces concepts et votre diagramme devra donc contenir
 - au moins une aggrégation ou une composition
 - au moins un lien d'héritage
 - au moins une classe abstraite
- (c) Les choix de lien entre classes (autre que l'héritage) et d'encapsulation devront être justifiés (en quelques lignes).
- (d) Le diagramme UML devra être lisible, et il est donc conseillé de le réaliser
 - soit à l'aide d'un logiciel (Dia, ArgoUML, ...)
 - soit via un site internet (online.visual-paradigm.com, creately.com/lp/uml-diagram-tool)

2. Implémentation Python :

- (a) Votre implémentation Python devra correspondre rigoureusement à votre diagramme de classe UML. Il devra donc contenir les mêmes classes, liens et champs que ceux indiqués dans votre diagramme.
- (b) Votre code devra
 - être commenté de sorte à être compréhensible
 - satisfaire le "Zen of Python" (www.python.org/dev/peps/pep-0020).
 - satisfaire les principes objets (utilisation de constructeur, méthodes de conversion d'objet en chaîne de caractères, comparaison d'objets, ...)
 - être composé de plusieurs fichiers / modules qui seront importés par le programme principal
 - se compiler et s'exécuter sans erreur
- (c) Votre programme devra fonctionner
 - à partir d'affichages clairs dans la console (mode textuel, cf images ci-dessous)
 - en mode Joueur contre Joueur (*i.e.* que des utilisateurs humains)
 - à l'aide de valeurs passées par l'utilisateur (pour choisir l'action à exécuter, la prochaine étape, ...)

3. Options : en plus de l'implémentation de base décrite en Q2, votre programme devra implémenter au moins une des options proposées dans le sujet.

Les groupes de projet sont à définir **au sein de votre groupe de TP** avant le **4 octobre**. Le projet (diagramme UML et ses justifications, code Python) est à rendre pour le **17 décembre**.

SUJET - POKEMON™

Pokemon™ est une franchise créée en 1996 (et dérivée depuis sous la forme de jeux vidéos, d'anime, de mangas, *etc.*) et se situant dans un monde fictionnel où des dresseurs capturent et entraînent de charmantes petites créatures (les Pokemon) dans le but de les faire combattre dans une arène.

L'objectif de ce projet est de créer un système (simplifié) de combat entre Pokemon, pour lequel on considèrera les règles / contraintes suivantes :

- Un dresseur sera représenté par son nom, l'ensemble des Pokemon qu'il a déjà capturé et son "deck", c'est à dire les 3 Pokemon qu'il utilise pour ses combats. Un dresseur pourra capturer des Pokemon, combattre d'autres dresseurs et changer son deck (s'il n'est pas en combat).
- 2 types de combat existent :

- Joueur contre Joueur (JcJ) : dans un combat JcJ, 2 dresseurs s'affrontent. Le combat se déroule au tour par tour. À son tour, un dresseur peut soit utiliser une compétence de son Pokemon actif, soit choisir de remplacer son Pokemon actif par un autre Pokemon de son deck, soit déclarer forfait (et perdre le combat).

Si le Pokemon actif a été mis ko, il n'est évidemment pas possible d'utiliser une de ses compétences. Le combat s'arrête si les 3 Pokemon du deck d'un des dresseurs sont ko (c'est à dire que leur vie est égale à 0) ou si l'un des dresseurs déclare forfait.

- Joueur contre Environnement (JcE) : dans un combat JcE, 1 dresseur affronte 1 Pokemon soit pour le capturer soit pour entrainer ses Pokemon. Le combat se déroule au tour par tour et s'arrête si le Pokemon est ko ou capturé, si les 3 Pokemon du deck du dresseur sont ko ou si le dresseur s'enfuit. Le dresseur peut essayer de capturer le Pokemon dès que sa vie descend en-dessous de 20% avec une probabilité égale à $4 \left(0.2 - \frac{V}{V_{\max}} \right)$, où V est la vie actuelle du Pokemon et V_{\max} sa vie maximale.

- Lors d'un combat, tous les Pokemon commencent avec leur vie et énergie au maximum. Un Pokemon ne peut utiliser une compétence que s'il a assez d'énergie pour cela.

- Un Pokemon sera représenté par son nom, son élément (air, eau, feu ou terre), son niveau et son expérience, sa vie maximale, son énergie maximale et sa régénération d'énergie (gain d'énergie à chaque tour d'un combat). Un Pokemon dispose de 3 à 5 compétences qu'il peut utiliser en combat. Elles servent soit à attaquer le Pokemon adverse, soit à se défendre (gain de vie ou d'énergie).

Une liste des Pokemon avec leurs caractéristiques et les compétences est fournie (sous la forme de fichier texte) et devra être lue par votre programme.

- Toutes les compétences ont un nom, une description, un élément et un coût. Les compétences d'attaque auront en plus une puissance et une précision tandis que les compétences défensives auront une valeur de soin et d'énergie (l'une des 2 pouvant être nulle).

Lors de l'utilisation d'une compétence défensive par un Pokemon, on générera un nombre aléatoire compris entre les 2 valeurs données pour déterminer la quantité de vie ou d'énergie restaurée par la compétence.

Lorsqu'un Pokemon attaque, on détermine d'abord la réussite de l'attaque (échec de l'attaque si un tirage aléatoire entre 0 et 100 est supérieur à la précision de l'attaque). Si l'attaque réussie, on calcule alors les dégâts infligés par l'attaque de la manière suivante

$$\delta = \text{arrondi} \left(cm \left(\frac{\rho(4\eta_a + 2)}{\Omega_d} + 2 \right) \right) \text{ avec}$$

- ρ : puissance de l'attaque
- η_a : niveau du Pokemon assaillant
- Ω_d : résistance du Pokemon assailli
- $cm = b \times \text{random}(0.85, 1)$
- b : coefficient multiplicateur défini ci-contre

		Compétence			
		Air	Eau	Feu	Terre
Pokemon	Air	1	1	0.5	1.5
	Eau	1.5	1	1	0.5
	Feu	0.5	1.5	1	1
	Terre	1	0.5	1.5	1

Quelque soit le résultat de la compétence (succès ou échec), son coût est retranché à l'énergie du Pokemon.

- À la fin d'un combat JcJ, les 3 Pokemon du dresseur vainqueur gagnent de l'expérience de la manière suivante : $10 + \text{niveau moyen des Pokemon vaincus} - \text{niveau du Pokemon}$.

À la fin d'un combat JcE, les 3 Pokemon du dresseur gagnent de l'expérience si le Pokemon a été mis ko (mais pas s'il est capturé) de la manière suivante : $(10 + \text{niveau du Pokemon vaincu} - \text{niveau du Pokemon}) / 3$. Si l'expérience totale d'un Pokemon est égale à 100 (niveau + 1), il évolue alors au niveau supérieur et ses statistiques (vie, énergie, résistance) augmente d'une valeur aléatoire (comprise entre 1 et 5 par

exemple).

Il pourra être nécessaire d'implémenter une "IA" pour permettre un jeu Humain(s) vs Ordinateur. Plusieurs niveaux de difficulté pourront être proposés. Une IA consistera simplement en une suite d'instructions permettant de simuler un joueur humain (inutile de recoder AlphaGo ou Watson ...). Par exemple, l'ordinateur utilisera prioritairement les compétences offensives les plus puissantes, se soignera si besoin ou changera de Pokemon pour bénéficier du bonus élémentaire b .

Les options proposées pour ce projet sont :

1. avoir une interface graphique (mais une version console textuelle devra toujours exister).
2. avoir un système de tournoi ou de championnat : un grand nombre de dresseurs s'affrontent dans des combats JcJ soit à élimination directe (tournoi, le perdant est éliminé), soit avec un système de points pour chaque victoire (championnat, tous les dresseurs se rencontrent 1 fois).
Une fonctionnalité de sauvegarde (dans un fichier texte) pour permettre d'interrompre une partie à la fin d'un combat pourra être utile. On prévoira évidemment une fonction de chargement de partie pour la reprendre lorsqu'on le souhaite.
3. gérer des compétences agissant sur plusieurs tours ou plusieurs Pokemon. Un Pokemon pourra ainsi être brûlé, gelé, ... , ce qui lui infligera des dégâts pendant plusieurs tours de suite ou inversement, se soigner, récupérer plus d'énergie ou augmenter sa résistance pendant plusieurs tours. Les compétences pourront aussi cibler tous les Pokemon (du lanceur si défensif, de l'adversaire si offensif), avoir des effets lors de changement de Pokemon, *etc.* On pourra aussi ajouter des temps de rechargement pour les compétences (*e.g.* une même compétence ne peut pas être utiliser 2 fois de suite).
4. gérer l'évolution des Pokemon : lorsqu'un Pokemon atteint un certain niveau, il peut évoluer vers une nouvelle forme et avoir ainsi accès à de nouvelles compétences (*cf.* fichier pokemon.txt)

Des affichages possibles de votre programme sont donnés ci-dessous et des exemples de combats de Pokemon peuvent être vu à ce lien : <https://play.pokemonshowdown.com/>.

```
#-----#
#      Bienvenue dans POKEMON!      #
# Le jeu de Pokemon Oriente Objet :D #
#-----#
```

```
Quel est votre nom? MT
Voilà votre dresseur: MT: 4 Pokemon. Le deck est:
0/ Galopa(Lvl 9, 900/1000, Feu): Vie 127/127, Energie 94/94 (+11), Resistance 75
Canicule, Pied Brûleur, Boutefeu, Roue de Feu, Sang chaud
1/ Tortank(Lvl 14, 1400/1500, Eau): Vie 291/291, Energie 127/127 (+16), Resistance 195
Bulles d'O, Ecume, Surf, Cascade, Hydrocanon
2/ Roucarnage(Lvl 11, 1100/1200, Air): Vie 229/229, Energie 127/127 (+18), Resistance 159
Bec Vrille, Vol, Picpic, Cru-Ailes, Tornade
0/ Voir vos pokemon
1/ Changer le deck
2/ Combattre / Capturer un pokemon
3/ Combattre un autre dresseur
4/ Creer un nouveau dresseur
5/ Quitter

Que voulez vous faire? (0-5)
```

(a) Écran d'accueil

```
Changement de deck
Voici votre deck:
0/ Roucarnage(Lvl 15, 1500/1600, Air): Vie 219/219, Energie 117/117 (+17), Resistance 124
Bec Vrille, Vol, Picpic, Cru-Ailes, Tornade
1/ Raichu(Lvl 7, 700/800, Air): Vie 155/155, Energie 80/80 (+15), Resistance 82
Eclair, Tonnerre, Cage Eclair, Fatal-Foudre, Pile Duracell
2/ Sabelette(Lvl 3, 300/400, Terre): Vie 95/95, Energie 62/62 (+5), Resistance 39
Jet-Pierres, Lame de Roc, Roc-Boulet
```

```
Quel pokemon voulez-vous changer? (0-2) 2
Voici vos pokemon:
3/ Nidoran(Lvl 1, 100/200, Terre): Vie 122/122, Energie 50/50 (+7), Resistance 47
Roulade, Martoboïs, Eboulement
4/ Goupix(Lvl 5, 500/600, Feu): Vie 94/94, Energie 54/54 (+7), Resistance 47
Poing Feu, Feu Sacre, Crocs Feu
5/ Ponyta(Lvl 1, 100/200, Feu): Vie 81/81, Energie 74/74 (+8), Resistance 46
Canicule, Pied Brûleur, Boutefeu
```

```
Quel pokemon voulez-vous ajouter? (0-5) 4
```

(b) Changement d'un Pokemon du deck

```
Combat JcJ entre MT et Chuck Norris!
0/ Nidoran(Lvl 5, 500/600, Terre): Vie 127/127, Energie 58/58 (+9), Resistance 39
Roulade, Martoboïs, Eboulement
1/ Reptincel(Lvl 7, 700/800, Feu): Vie 130/130, Energie 87/87 (+15), Resistance 54
Crocs Feu, Poing Feu, Flammeche, Deflagration
2/ Roucool(Lvl 4, 400/500, Air): Vie 113/113, Energie 56/56 (+9), Resistance 38
Bec Vrille, Vol, Picpic

Quel pokemon voulez vous utiliser? (0-2) 1
0/ Herbizarre(Lvl 6, 600/700, Terre): Vie 223/223, Energie 76/76 (+13), Resistance 98
Fulgurine, Canon Graine, Balle Graine, Fouet Lianes
1/ Feunard(Lvl 6, 600/700, Feu): Vie 165/165, Energie 84/84 (+12), Resistance 50
Poing Feu, Feu Sacre, Crocs Feu, Roue de Feu, Aurore
2/ Roucarnage(Lvl 14, 1400/1500, Air): Vie 177/177, Energie 150/150 (+19), Resistance 178
Bec Vrille, Vol, Picpic, Cru-Ailes, Tornade

Quel pokemon voulez vous utiliser? (0-2) 2
```

(c) Début d'un combat JcJ

```
-----
Tour 1
C'est a MT de jouer!
Reptincel(Lvl 7, 700/800, Feu): Vie 130/130, Energie 87/87 (+15), Resistance 54
Crocs Feu, Poing Feu, Flammeche, Deflagration
0/ Crocs Feu (Attaque, Feu, Cout: 15): Le lanceur utilise une morsure enflammée.
1/ Poing Feu (Attaque, Feu, Cout: 15): Un coup de poing enflamme vient frapper l'ennemi.
2/ Flammeche (Attaque, Feu, Cout: 25): L'ennemi est attaque par une faible flamme.
3/ Deflagration (Attaque, Feu, Cout: 5): Un deluge de flammes ardentes submerge l'ennemi.
4/ Changer de pokemon
5/ Passer votre tour
6/ Fuir le combat

Que voulez vous faire? (0-6) 1
Attaque reussie (Poing Feu): 5 degats
C'est a Chuck Norris de jouer!
Roucarnage(Lvl 14, 1400/1500, Air): Vie 172/177, Energie 150/150 (+19), Resistance 178
Bec Vrille, Vol, Picpic, Cru-Ailes, Tornade
0/ Bec Vrille (Attaque, Air, Cout: 15): Une attaque utilisant le bec comme une perceuse.
1/ Vol (Attaque, Air, Cout: 10): Le lanceur s'envole et frappe l'ennemi.
2/ Picpic (Attaque, Air, Cout: 25): Frappe l'ennemi d'un bec pointu ou d'une corne pour infliger des degats.
3/ Cru-Ailes (Attaque, Air, Cout: 25): L'ennemi est frappe par de larges ailes deployees pour infliger des degats.
4/ Tornade (Attaque, Air, Cout: 25): Le lanceur bat des ailes pour generer une bourrasque qui blesse l'ennemi.
5/ Changer de pokemon
6/ Passer votre tour
7/ Fuir le combat

Que voulez vous faire? (0-7) 4
Attaque reussie (Tornade): 22 degats
-----
Tour 2
```

(d) Tour d'un combat JcJ