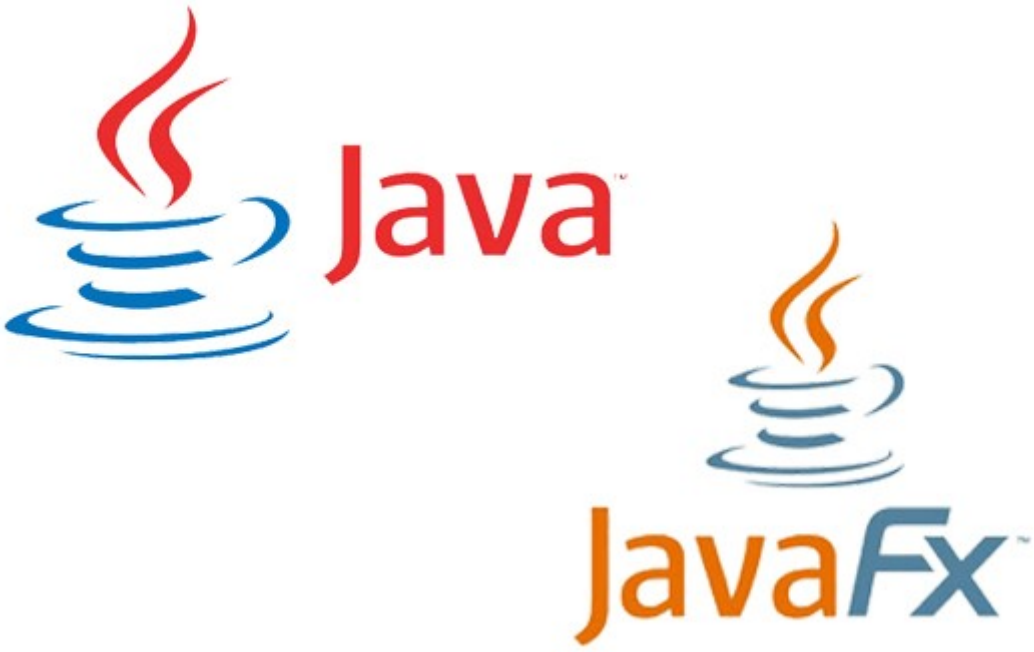
**Perret Louis**
**Wissocq Maxime**
**G3-4**

IUT CLERMONT - FD
UNIVERSITÉ
Clermont Auvergne

# Projet Java :
# Pokemon Ultimate

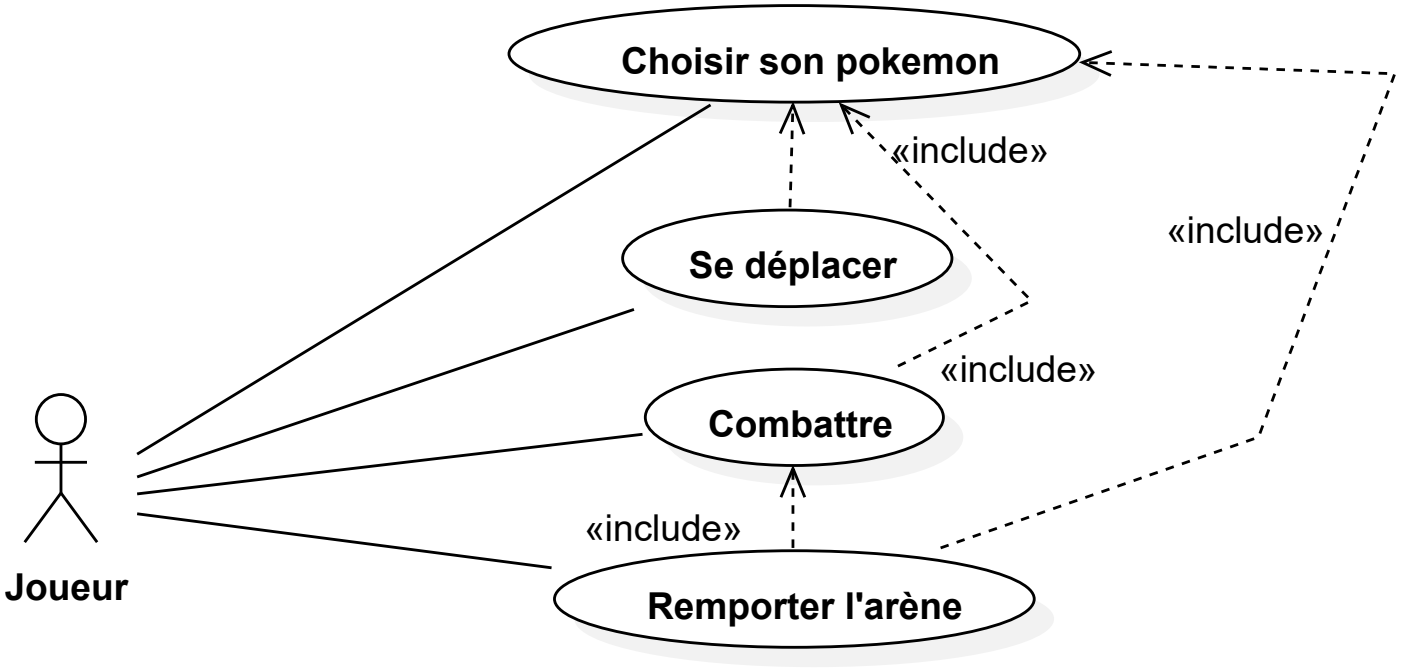


## Sommaire :

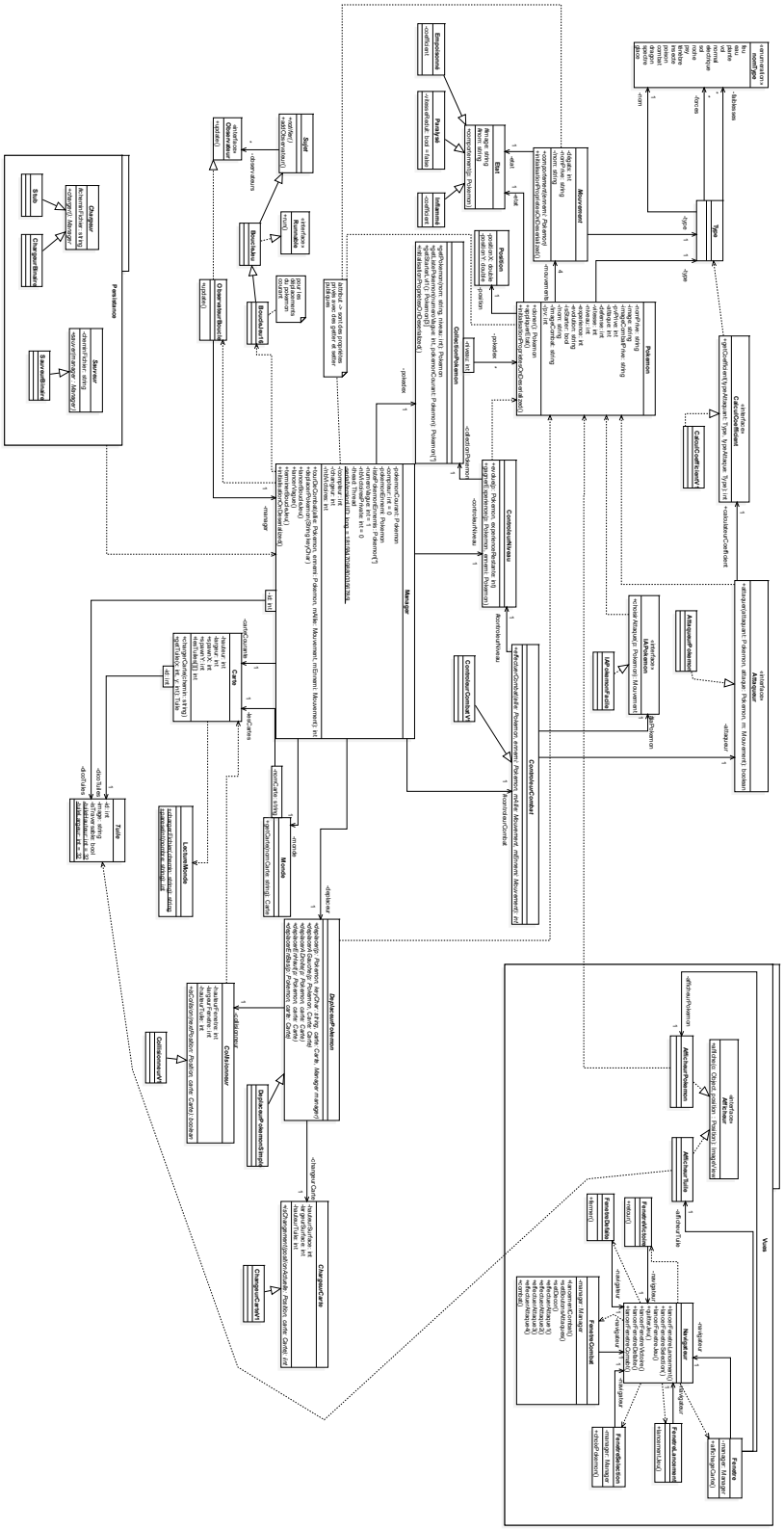
## Contexte

Le but du jeu est de survivre à plusieurs vagues de pokémons sauvages au sein d'une arène. Le joueur pourra choisir son pokémon parmi les 3 disponibles. Chacun de ces trois pokémons à ses propres forces et faiblesses. Le joueur devra ensuite se diriger vers l'arène puis une fois entré, il devra vaincre tous les pokémons de la vague. Les combats pokémons permettent à 2 pokémons de s'affronter à l'aide de leurs attaques.
Les pokémons s'attaquent chacun leur tour, le pokémon attaquant le premier dans un tour est déterminé par sa vitesse. Ces attaques font plus ou moins de dégâts en fonction de leur type et du type de l'ennemi. Une fois sorti vainqueur de la vague, le pokémon sera soigné et gagnera de l'expérience lui permettant d'évoluer. Il y a deux paliers d'évolution pour chaque pokémon jouable. Le joueur devra ensuite vaincre les vagues suivantes dans l'arène, vagues composées de pokémons de plus en plus puissant, souvent des évolutions de pokémons déjà vaincus dans des vagues précédentes. Une fois toutes les vagues vaincues, le pokémon sera sacré maître pokémon !

**Diagramme de cas d'utilisation**



## Diagramme de classes



## Description diagramme de classes

A. Partie Pokemon

Cette partie permet d'identifier ce qu'est un Pokémon :

- Classe Pokemon : représente un pokémon d'après beaucoup de données comme son nom, ses points de vies (pv), son attaque, etc
- Classe Mouvement : représente les attaques d'un pokémon
- Classe Type : Représente le type du pokémon et de ses attaques
- Enum NomType : Énumération composée des noms de chaque type existant
- Classe Position : Représente la position en (x,y) du pokémon
- Classe Etat (et ses filles) : Représente les différents statuts qu'un pokémon peut subir durant un combat. Nous avons utilisé un patron Etat puisque ce dernier nous permet de changer dynamiquement cet état lors d'un combat et modifier le comportement que cela peut avoir sur le pokémon. En effet, un pokémon peut n'avoir aucun état, puis une fois empoisonné, il commencera à perdre des points de vie au fil du combat.
- Classe CollectionPokemon : Permet de stocker tous les pokémon du jeu par niveau dans un dictionnaire.

B. Fonctionnalité combat:

Cette partie permet de gérer les combats entre les pokemons :

- Interface Attaqueur : Permet de gérer les dégâts que va causer une attaque à un pokémon suivant plusieurs paramètres : les dégâts de l'attaque ainsi que l'attaque de l'attaquant, la défense de l'attaqué et les affinités des types.
- Interface CalculCoefficient : Permet de calculer le coefficient de dégâts que l'on va ajouter/enlever aux dégâts de l'attaque suivant les affinités des types des deux pokémons.
- Interface IAPokemon : Permet de simuler le choix d'une attaque pour les pokémons adverses au joueur.
- Classe ControleurNiveau : Permet de contrôler le gain d'expérience, la montée d'un niveau et l'évolution d'un pokémon.
- Classe ControleurCombat : Permet de coordonner toutes ces classes entre elles afin d'effectuer correctement un combat entre deux pokémons.

C. Partie déplacement:

Cette partie gère le déplacement du pokémon du joueur dans notre jeu :

- Classe DeplaceurPokemon: Cette classe contient des méthodes changeant les coordonnées du personnage dans 4 directions (haut, bas, gauche, droite) et une méthode appelant ces méthodes en fonction des touches pressées.
- Classe Collisionneur: Vérifie que la tuile se trouvant à la position où le personnage souhaite se déplacer est traversable.
- Classe ChangeurCarte: Vérifie si un événement est lié à la tuile se trouvant à la position où le personnage souhaite se déplacer.

D. Partie carte/monde:

Cette partie gère la création des cartes dans lesquelles le joueur joue :

- Classe Tuile: Classe métier de l'objet Tuile, un objet avec une image et un id. Une tuile peut être traversable et des évènements peuvent être attachés à ces tuiles. Les tuiles composent l'environnement graphique dans lequel évolue le joueur.
- Classe LectureMonde: Classe permettant de charger un fichier à partir de son nom et de le parser.
- Classe Carte: Permet de récupérer une hauteur, une largeur, des points d'apparition et un tableau de tuiles à partir du parsage d'un fichier texte.
- Classe Monde: Représente une Map contenant toutes les cartes et leurs noms.

E. Partie Boucle de Jeu

Cette partie permet de gérer notre boucle de jeu que nous avons implémentée afin d'actualiser la position et l'affichage du pokémon sur notre fenêtre. Pour ce faire nous avons implémenté un patron Observateur afin de pouvoir observer cette boucle et d'être notifié à chaque boucle de cette dernière. Cette notification aura pour but de venir actualiser la position du personnage sur la fenêtre :

- Classe Sujet : Représente le sujet observé et possède une collection d'observateurs à venir notifier.
- Interface Observateur : Représente les observateurs qui effectuent un traitement particulier lorsqu'ils sont notifiés.
- Classe BoucleJeu : Représente notre boucle de jeu.

F. Partie persistance:

Cette partie permet de gérer la persistance de notre jeu. Nous avons implémenté un patron Stratégie qui nous permet de changer facilement la façon dont on persiste à l'exécution. A l'heure actuelle, nous sauvegardons nos données sous format binaire, mais si nous souhaitons le faire à l'aide d'une base de données par exemple, cela ne demandera pas de changement conséquent dans notre code :

- Classe Chargeur : Gère le chargement de nos données
- Classe Sauveur : Gère la sauvegarde de nos données

G. Partie vue:

La partie vue correspond à l'interface graphique de notre jeu. Elle est donc composée des classes permettant l'affichage de notre jeu sur un assemblage de conteneurs mais également les contrôleurs.

- Classe AfficheurPokemon: Permet d'afficher des pokémons passés en paramètre à une position donnée en convertissant cet objet pokémon en ImageView
- AfficheurTuile: Même utilité que pour AfficheurPokemon, mais pour les objets de type Tuile.
- Contrôleurs:
  - FenetreLancement: Contrôleur appelé en premier lors du démarrage du jeu. Il permet de quitter le jeu où de se diriger vers le contrôleur gérant la sélection des pokémons.
  - FenetreSelection: Contrôleur appliquant le pokémon courant en fonction du choix de l'utilisateur.
  - Fenetre: Correspond à la fenêtre de jeu, la fenêtre sur laquelle notre personnage se déplace. Cette classe réalise l'affichage de la carte en ajoutant des ImageView au conteneur de la fenêtre.
  - FentreCombat: Correspond à la fenêtre de combat, là où sont affichés les pokémons participant à un combat ainsi que l'interface de combat(attaques, points de vie des pokémons, état). C'est également depuis cette classe que l'on appelle les méthodes permettant de simuler les attaques et ici, que l'action à réaliser en fonction du résultat du combat est réalisée.
  - FenetreDefaite: Contrôleur appelée en cas de défaite du joueur lors d'un combat. Elle permet de revenir à la fenêtre de lancement.
  - FenetreVictoire: Fenetre affichée en cas de victoire du joueur. Elle permet de revenir à la fenêtre de lancement.
  - Navigateur: Contrôleur contenant des méthodes permettant de charger et d'appliquer des scènes (de changer de fenêtre).