Projet
Programmation Orientée Objet en C++

Daniel Wladdimiro
ESILV
daniel.wladdimiro@devinci.fr

Second semestre 2024-2025

1 Introduction

La programmation orientée objet (POO) est un paradigme fondamental en informatique, largement utilisé dans le développement de logiciels modernes. L'apprentissage de la POO est essentiel pour le développement de logiciels bien structurés et modulaires.

Ce projet doit être réalisé en **binôme** (équipes de deux). Il propose une approche ludique et pédagogique à travers la création d'un simulateur de combat Pokémon. L'univers des Pokémon est idéal pour illustrer les principes de la POO, car chaque créature possède des caractéristiques propres (nom, type, points de vie, attaque), qui peuvent être modélisées sous forme d'objets et organisées en hiérarchies de classes.

Ce projet vise à rendre l'apprentissage de la POO plus engageant en permettant aux équipes de mettre en pratique des concepts clés tels que l'héritage, le polymorphisme et l'encapsulation dans un cadre interactif et stimulant.

2 Description du Projet

Ce projet consiste en la création d'un simulateur de combat Pokémon en C++. Les équipes devront implémenter un système où des joueurs peuvent choisir leurs Pokémon et les faire combattre en respectant les règles classiques du jeu.

2.1 Objetif du projet

L'objectif est de permettre aux équipes de structurer leur code en définissant une hiérarchie de classes pour représenter les différents types de Pokémon et leurs attaques, ainsi que les personnages du jeu, tout en mettant en place un système de combat respectant les mécaniques de type et de dégâts.

3 Combat

Le combat repose sur un mécanisme au tour par tour où chaque Pokémon peut attaquer son adversaire. Les dégâts infligés dépendent du type de Pokémon et de l'attaque choisie. Le combat se termine lorsqu'un des Pokémon n'a plus de points de vie (HP).

Les combats sont structurés selon les règles suivantes :
— Chaque adversaire utilise le premier Pokémon de sa liste.
— Les combats sont en tour par tour.
— À chaque tour, le Pokémon actif attaque en appliquant son attaque et les multiplicateurs de dégâts (faiblesse/résistance et bonus éventuels).
— Le joueur attaque toujours en premier.
— Si un Pokémon atteint 0 HP, il est remplacé par le suivant dans la liste.
— Le combat continue de manière séquentielle selon l'ordre des Pokémon dans la liste.
— Si un des adversaires n'a plus de Pokémon disponibles, le combat se termine et le vainqueur est annoncé avant de retourner au menu principal.

4 Pokémon

Chaque Pokémon disposera des caractéristiques suivantes :
— Nom du Pokémon
— Un ou deux types de Pokémon (Feu, Eau, Plante, Électrik, Glace, Combat, Poison, Sol, Vol, Psy, Insecte, Roche, Spectre, Dragon, Ténèbres, Acier, Fée)
— Points de Vie (HP)
— Une attaque spécifique
— Dégâts de l'attaque
— Une faiblesse spécifique en fonction de son type

Les équipes devront implémenter au minimum trois types (par exemple : Feu, Eau, Plante) de Pokémon en utilisant l'héritage. Une classe de base `Pokemon` sera définie, et les types de Pokémon spécifiques hériteront de cette classe en redéfinissant les attaques et caractéristiques spécifiques.

Une considération importante est que certains Pokémon peuvent posséder deux types. De plus, chaque type a une faiblesse spécifique qui influencera les dégâts reçus lors des combats. Cette faiblesse devra être définie comme une facteur au sein du système de combat. Lorsqu'un Pokémon reçoit une attaque, il devra calculer les dégâts en fonction de sa faiblesse ou de sa résistance au type d'attaque adverse, ce qui impactera le multiplicateur de dégâts appliqué.

La règle suivante sera appliquée :
— Si un Pokémon est faible contre un type d'attaque, il subira le double des dégâts (multiplicateur de 2).
— Si un Pokémon est résistant contre un type d'attaque, il subira seulement la moitié des dégâts (multiplicateur de 0.5).

Le tableau suivant illustre les faiblesses ($\times\mathbf{2}$) et les résistances ($\times\mathbf{0.5}$) de chaque type de Pokémon :

4.1 Exemples de Faiblesses et Résistances

Voici deux exemples concrets illustrant l'application des faiblesses et résistances dans un combat Pokémon :

4.1.1 Exemple 1 : Salamèche contre Carapuce

Salamèche (type Feu) affronte Carapuce (type Eau). Selon la table des faiblesses :

Type	Faiblesse (×2)	Résistance (×0.5)
Feu	Eau, Roche, Sol	Plante, Glace, Insecte, Acier, Fée
Eau	Plante, Électrik	Feu, Eau, Glace, Acier
Plante	Feu, Glace, Poison, Vol, Insecte	Eau, Sol, Roche
Électrik	Sol	Vol, Acier, Électrik
Glace	Feu, Combat, Roche, Acier	Glace
Combat	Vol, Psy, Fée	Roche, Insecte, Ténèbres
Poison	Sol, Psy	Plante, Fée, Combat, Poison, Insecte
Sol	Eau, Plante, Glace	Poison, Roche
Vol	Électrik, Glace, Roche	Plante, Combat, Insecte
Psy	Insecte, Spectre, Ténèbres	Combat, Psy
Insecte	Feu, Vol, Roche	Plante, Combat, Sol
Roche	Eau, Plante, Combat, Sol, Acier	Feu, Vol, Poison, Normal
Spectre	Spectre, Ténèbres	Poison, Insecte
Dragon	Glace, Dragon, Fée	Feu, Eau, Électrik, Plante
Ténèbres	Combat, Insecte, Fée	Spectre, Psy, Ténèbres
Acier	Feu, Combat, Sol	Normal, Plante, Glace, Vol, Psy, Insecte, Roche, Dragon, Acier, Fée
Fée	Poison, Acier	Combat, Insecte, Ténèbres, Dragon

Table 1 – Tableau des faiblesses et résistances par type de Pokémon.

— Carapuce a un avantage sur Salamèche, car l'eau est super efficace contre le feu. L'attaque *Pistolet à O* (65 dégâts) infligera **2**× les dégâts normaux à Salamèche.
— Inversement, les attaques Feu de Salamèche seront peu efficaces contre Carapuce, infligeant **0.5**× les dégâts normaux.

Pokémon	Type	HP	Attaque	Puissance	Multiplicateur	Dégâts finaux
Salamèche	Feu	39	Flammèche	70	×0.5	35
Carapuce	Eau	44	Pistolet à O	65	×2	130

Table 2 – Calcul des dégâts entre Salamèche et Carapuce

Dans ce combat, Carapuce a un net avantage et peut vaincre Salamèche en un seul coup.

4.1.2 Exemple 2 : Salamèche contre Bulbizarre

Salamèche (Feu) affronte Bulbizarre (Plante/Poison). D'après les règles de faiblesse et résistance :

— Salamèche a l'avantage, car le Feu est super efficace contre la Plante. L'attaque *Flammèche* (70 dégâts) infligera **2**× les dégâts normaux à Bulbizarre.
— Bulbizarre, en tant que Pokémon Plante, aura ses attaques réduites en efficacité contre Salamèche, causant **0.5**× les dégâts normaux.

Pokémon	Type	HP	Attaque	Puissance	Multiplicateur	Dégâts finaux
Salamèche	Feu	39	Flammèche	70	×2	140
Bulbizarre	Plante/Poison	45	Fouet Lianes	60	×0.5	30

Table 3 – Calcul des dégâts entre Salamèche et Bulbizarre

Dans ce combat, Salamèche peut vaincre Bulbizarre en un seul coup grâce à son attaque *Flammèche*.

5 Entraîneurs

De plus, une classe `Entraîneur` sera définie pour représenter les différents personnages du jeu. Cette classe inclura les caractéristiques suivantes :

- Nom de l'entraîneur
- Liste de Pokémon (maximum 6)

Elle servira de base pour :

- **Les joueurs** : Ils possèdent une équipe de Pokémon et peuvent affronter d'autres joueurs ou des leaders de gym. Chaque joueur disposera des caractéristiques suivantes :
  - Nombre de badges remportées
  - Nombre de combats gagnés
  - Nombre de combats perdus
- **Les leaders de gym** : Ils représentent un défi plus important et sont associés à un gymnase spécifique. Comme les joueurs, ils possèdent une équipe de Pokémon et sont caractérisés par les attributs suivants :
  - Badge spécifique du gymnase
  - Gymnase auquel ils appartiennent
- **Les Maîtres Pokémon** : Ils ne peuvent être affrontés qu'après avoir obtenu toutes les badges disponibles. Leur particularité est qu'ils possèdent une capacité spéciale qui augmente de 25% les dégâts des attaques de leurs Pokémon.

6 Menu de Simulation

Pour permettre aux utilisateurs d'interagir avec la simulation, un menu sera mis en place avec les options suivantes :

- Afficher les Pokémon et leurs attributs
- Récupérer les points de vie (HP) des Pokémon
- Changer l'ordre des Pokémon dans l'équipe
- Afficher les statistiques du joueur (nombre de badges, combats gagnés et perdus)
- Affronter un gymnase, avec un sous-menu affichant les leaders disponibles (minimum 4)
- Affronter un Maître Pokémon (disponible uniquement après avoir obtenu toutes les badges). Un maître sera sélectionné au hasard et un combat commencera.
- Interagir avec les Pokémon ou les entraîneurs vaincus.

6.1 Interaction avec les Pokémon et les Entraîneurs

L'option d'interaction permet aux joueurs d'interagir avec leurs Pokémon ainsi qu'avec les entraîneurs qu'ils ont vaincus en combat. Pour cela, une interface nommée `Interagir` sera mise en place et définira un comportement commun pour tous les entités du jeu pouvant être interagis.

- Seuls les Pokémon appartenant au joueur peuvent être interagis.
- Seuls les entraîneurs vaincus peuvent être interagis.
- Lorsqu'un joueur interagit avec un Pokémon, un message unique s'affiche pour chaque espèce de Pokémon.
- Lorsqu'un joueur interagit avec un entraîneur vaincu, ce dernier peut donner un message spécifique ou une information utile.

7 Chargement des Données

Les données du jeu seront chargées à partir de fichiers `.csv`, qui contiendront les valeurs des différentes classes du programme. Cela inclura notamment :

- **pokemon.csv** : Contient les statistiques des Pokémon.
- **joueur.csv** : Liste des joueurs avec leurs informations.
- **leaders.csv** : Informations sur les leaders de gymnase et leurs équipes.
- **maitres.csv** : Données sur les Maîtres Pokémon pouvant être affrontés après avoir obtenu toutes les médailles.

Ces fichiers seront fournis en exemple et pourront être utilisés pour initialiser le jeu. L'utilisation de fichiers CSV permet une gestion simple et efficace des données, facilitant ainsi leur modification et leur réutilisation dans le cadre du projet.

8 Evaluation

Ce projet devra être présenté lors du **dernier TD** sous forme d'une démonstration en présentiel. La réalisation en **binôme** est **obligatoire**. Chaque équipe devra expliquer le fonctionnement de son programme, démontrer les différentes fonctionnalités mises en place et répondre aux questions concernant leur implémentation.

L'évaluation sera notée sur *20 points*, répartis comme suit :

- **Diagramme UML** (2 points) : Présentation d'un diagramme UML clair et structuré représentant les relations entre les classes du projet.
- **Héritage** (2 points) : Utilisation correcte du principe d'héritage pour structurer les classes du jeu.
  - Classes abstraites (1 point).
  - Classes dérivées (1 point).
- **Encapsulation** (2 points) : Bonne gestion des attributs et méthodes en respectant les niveaux d'accès (`private`, `protected`, `public`).
  - Afficher les statistiques du joueur (nombre de badges, combats gagnés et perdus) (0.5 point).
  - Afficher les Pokémon et leurs attributs (0.5 point)
  - Accesseurs et modificateurs de chaque classe (1 point).
- **Polymorphisme** (2 points) : Implémentation correcte de méthodes polymorphiques permettant une flexibilité du code.
- **Interface** (2 points) : Création et utilisation d'interfaces permettant une interaction avec les entités du jeu (ex. `Interagir`).
- **Implémentation** (4 points) : Fonctionnalité complète et correcte du programme, incluant le combat, la gestion des Pokémon et des entraîneurs.
  - Système de combat (1 point).
  - Système de faiblesses et résistances (1 point).
  - Affronter un gymnase (1 point).
  - Affronter un Maître Pokémon (1 point).
- **Présentation** (4 points) : Clarté et qualité de la présentation du projet lors de la soutenance.
  - **Structure de la présentation** (1 point) : La présentation suit un ordre logique (introduction, explication du code, démonstration, conclusion).
  - **Démonstration fonctionnelle** (1 point) : Le projet fonctionne correctement et toutes les fonctionnalités principales sont démontrées.

  - **Explication technique** (1 point) : Les étudiants sont capables d'expliquer les choix techniques du projet (POO, héritage, polymorphisme, etc.).
  - **Clarté et communication** (1 point) : La présentation est fluide, bien articulée et compréhensible.
- **Questions** (2 points) : Capacité des étudiants à répondre aux questions techniques et à expliquer leurs choix d'implémentation.