

Enseignant(s)

PICHARD Félix

Email(s)

fpichard@esgi.fr

Création d'un outil de gestion des clients

1 Matières, formations et groupes

Matière liée au projet :

Formations : -

Nombre d'étudiant
par groupe :

2 à 3

Règles de constitution des groupes: **Libre**

Charge de travail
estimée par étudiant : **40,00 h**

2 Sujet(s) du projet

Type de sujet : **Imposé**

Projet de C

Vous allez devoir créer un Pokédex permettant de stocker les Pokémons que vous avez croisé et capturé durant votre aventure ! Il est possible pour un utilisateur d'enregistrer un Pokémon, s'il l'a croisé, même s'il ne l'a pas capturé. En revanche, il existe un compteur dans votre pokédex, permettant de savoir combien d'occurrence de ce pokémon vous possédez.

Evidemment, les opérations CRUD doivent être implémentées sur la liste chaînée. Le contenu de la liste chaînée doit être stocké en local dans le format de fichier que vous souhaitez (CSV ou SQLite).

Chaque Pokémon devra posséder a minima les données suivantes :

- Nom
- Type
- Nombres de Pokémons possédés
- Date de première découverte
- Date de première capture

Les contraintes techniques sont les suivantes :

- Vous devez posséder une liste chaînée « Pokémon », contenant l'intégralité de votre pokédex.
- Les opérations CRUD (Create, Read, Update, Delete) doivent être implémentées. Leur appel doit se réaliser via un tableau de pointeurs sur fonction :

- o Il doit être possible de créer une entrée de deux types : « J'ai vu un Pokémon » ou « J'ai capturé un Pokémon ». Lorsqu'on capture un Pokémon, un compteur s'incrémente au sein de la structure. Evidemment, si un Pokémon déjà capturé, l'a été à nouveau, on ne crée pas de nouveau nœud dans la liste chaînée, mais on incrémente de 1 le nombre de ces Pokémon dans la structure pré-existante.

- o Il doit être possible de lire les données a minima sous les formes suivantes :

- Affichage de tout le contenu de la liste chaînée ;

- Affichage des caractéristiques d'un Pokémon en particulier, en fonction de son nom ;

- Effectuer un filtre sur les Pokémon que nous possédons (excluant donc ceux qui n'ont été qu'observés).

- o Il doit être possible pour un utilisateur de modifier un nom de Pokémon (utile dans le cas d'une faute de frappe lors de l'enregistrement). De plus, comme évoqué ci-dessus, le fait d'ajouter un Pokémon étant déjà connu impliquera une mise à jour d'une structure existante.

- o Enfin, il est possible de supprimer des Pokémon de son pokédex. Attention, il existe plusieurs fonctionnalités de suppression :

- Si le Pokémon est connu, mais n'est pas possédé, on peut l'oublier ;

- Si nous avons un Pokémon en un seul exemplaire, nous pouvons le supprimer et garder les caractéristiques du Pokémon ou le supprimer et oublier les caractéristiques du Pokémon ;

- Si nous avons un Pokémon en plusieurs exemplaires, nous pouvons supprimer une, plusieurs ou toutes ces occurrences. Dans le cas où toutes ces occurrences sont supprimées, nous pouvons aussi l'oublier, si l'utilisateur le demande.

- Un parseur doit être implémenté pour lire un CSV ou SQLite (au choix) et le transformer en liste chaînée.

- Attention à la clarté et la propreté de votre code. Un malus pouvant monter jusqu'à 4 points peut être appliqué.

- Une fonctionnalité de sauvegarde de la liste chaînée doit être implémentée de 3 manières différentes. Cette sauvegarde se fait dans un fichier CSV ou une base de données SQLite :

- o Une commande « save » (ou équivalent) dans l'interface interactive du programme ;

- o Un signal SIGUSR1 envoyé au programme sauvegarde automatiquement ;

- o Un signal SIGINT (CTRL+C) doit demander si l'utilisateur veut :

- Quitter sans sauvegarder ;

- Quitter en sauvegardant ;

- Ne pas quitter.

3 Détails du projet

Objectif du projet (à la fin du projet les étudiants sauront réaliser un...)

L'objectif du projet est de créer un outil de gestion de clients en ligne de commande, en C, en utilisant les techniques vues en cours.

Descriptif détaillé

Vous allez devoir créer un Pokédex permettant de stocker les Pokémon que vous avez croisé et capturé durant votre aventure ! Il est possible pour un utilisateur d'enregistrer un Pokémon, s'il l'a croisé, même s'il ne l'a pas capturé. En revanche, il existe un compteur dans votre poké dex, permettant de savoir combien d'occurrence de ce pokémon vous possédez.

Evidemment, les opérations CRUD doivent être implémentées sur la liste chaînée. Le contenu de la liste chaînée doit être stocké en local dans le format de fichier que vous souhaitez (CSV ou SQLite).

Chaque Pokémon devra posséder a minima les données suivantes :

- Nom
- Type
- Nombres de Pokémon possédés
- Date de première découverte
- Date de première capture

Les contraintes techniques sont les suivantes :

- Vous devez posséder une liste chaînée « Pokémon », contenant l'intégralité de votre pokédex.
- Les opérations CRUD (Create, Read, Update, Delete) doivent être implémentées. Leur appel doit se réaliser via un tableau de pointeurs sur fonction :

- o Il doit être possible de créer une entrée de deux types : « J'ai vu un Pokémon » ou « J'ai capturé un Pokémon ». Lorsqu'on capture un Pokémon, un compteur s'incrémente au sein de la structure. Evidemment, si un Pokémon déjà capturé, l'a été à nouveau, on ne crée pas de nouveau nœud dans la liste chaînée, mais on incrémente de 1 le nombre de ces Pokémon dans la structure pré-existante.

- o Il doit être possible de lire les données a minima sous les formes suivantes :

- Affichage de tout le contenu de la liste chaînée ;

- Affichage des caractéristiques d'un Pokémon en particulier, en fonction de son nom ;

- Effectuer un filtre sur les Pokémon que nous possédons (excluant donc ceux qui n'ont été qu'observés).

- o Il doit être possible pour un utilisateur de modifier un nom de Pokémon (utile dans le cas d'une faute de frappe lors de l'enregistrement). De plus, comme évoqué ci-dessus, le fait d'ajouter un Pokémon étant déjà connu impliquera une mise à jour d'une structure existante.

- o Enfin, il est possible de supprimer des Pokémon de son pokédex. Attention, il existe plusieurs fonctionnalités de suppression :

- Si le Pokémon est connu, mais n'est pas possédé, on peut l'oublier ;

- Si nous avons un Pokémon en un seul exemplaire, nous pouvons le supprimer et garder les caractéristiques du Pokémon ou le supprimer et oublier les caractéristiques du Pokémon ;

- Si nous avons un Pokémon en plusieurs exemplaires, nous pouvons supprimer une, plusieurs ou toutes ces occurrences. Dans le cas où toutes ces occurrences sont supprimées, nous pouvons aussi l'oublier, si l'utilisateur le demande.

- Un parseur doit être implémenté pour lire un CSV ou SQLite (au choix) et le transformer en liste chaînée.

- Attention à la clarté et la propreté de votre code. Un malus pouvant monter jusqu'à 4 points peut être appliqué.

- Une fonctionnalité de sauvegarde de la liste chaînée doit être implémentée de 3 manières différentes.

Cette sauvegarde se fait dans un fichier CSV ou une base de données SQLite :

- o Une commande « save » (ou équivalent) dans l'interface interactive du programme ;
- o Un signal SIGUSR1 envoyé au programme sauvegarde automatiquement ;
- o Un signal SIGINT (CTRL+C) doit demander si l'utilisateur veut :
 - Quitter sans sauvegarder ;
 - Quitter en sauvegardant ;
 - Ne pas quitter.

Ouvrages de référence (livres, articles, revues, sites web...)

Outils informatiques à installer

| gcc, vim

4

Livrables et étapes de suivi

5**Soutenance**Durée de présentation
par groupe :**30 min**Audience : **A huis clos**

Type de présentation :

Démonstration

Précisions :