

Super Héros

COMPTE-RENDU PROJET

Auteurs : Mathis Grattepanche, Nathan Mariage

Date : 11 avril 2025

Module : Programmation en Langage C

Sommaire

1. Introduction	3
2. Fonctionnalités implémentées	4
○ Affichage des Super-Héros	4
○ Détails sur le Super-Héros	5
○ Recherche et Filtre	5
○ Comparaison de Super-Héros	5
○ Personnalisation de Profil	6
○ Quiz interactif	7
3. Difficultés rencontrées et solutions apportées	7
4. Commentaires et suggestions	8
5. Auto-évaluation	8
6. GitHub	9

Introduction

Ce projet a pour objectif la création d'une application en ligne de commande permettant aux utilisateurs de parcourir, rechercher et comparer des informations sur différents super-héros. Développé en langage C, ce programme utilise des fichiers JSON pour stocker et manipuler les données des super-héros, offrant une interface interactive en mode console.

L'application permet d'afficher les statistiques détaillées des super-héros, de les comparer entre eux, d'effectuer des recherches par nom et de filtrer selon divers critères. Des fonctionnalités avancées comme la gestion de favoris et un quiz interactif viennent enrichir l'expérience utilisateur.

Pour réaliser ce projet, nous avons utilisé la bibliothèque jansson pour la manipulation des données JSON et mis en œuvre des structures de données optimisées pour une gestion efficace des informations.

Fonctionnalités implémentées

Structure SuperHero

```
#define FILE_PATH "SuperHeros.json"

typedef struct {
    int id;
    char name[50];
    int intelligence;
    int strength;
    int speed;
    int durability;
    int power;
    int combat;
} SuperHero;
```

on crée une structure qui permet de stocker les informations pour chaque superhero dans un objet

Affichage des Super-Héros

```
printf("\nListe des super-héros :\n");
json_array_foreach(root, index, hero) {
    int id = json_integer_value(json_object_get(hero, "id"));
    const char *name = json_string_value(json_object_get(hero, "name"));
    printf("ID: %d | Nom: %s\n", id, name);
}
json_decref(root);
```

Description détaillée : Notre application permet de charger et d'afficher la liste complète des super-héros depuis un fichier JSON. Chaque super-héros est présenté avec un identifiant unique, son nom et quelques informations de base et ses principales caractéristiques.

L'affichage est organisé sous forme de successions de printf a retour de ligne pour une meilleure lisibilité,

Aspects techniques :

On utilise une boucle foreach effectuer sur le json pour explorer structure par structure on récupère le name de la structure à chaque itérations puis on l'affiche

Détails sur le Super-Héros

```
json_array_foreach(root, index, hero) {
    if (json_integer_value(json_object_get(hero, "id")) == id) {
        const char *name = json_string_value(json_object_get(hero, "name"));
        json_t *stats = json_object_get(hero, "powerstats");

        printf("\n\nDétails de %s :\n", name);
        printf("\nPowerstats:\n");
        printf("Intelligence: %lld\n", json_integer_value(json_object_get(stats, "intelligence")));
        printf("Force: %lld\n", json_integer_value(json_object_get(stats, "strength")));
        printf("Vitesse: %lld\n", json_integer_value(json_object_get(stats, "speed")));
        printf("Durabilité: %lld\n", json_integer_value(json_object_get(stats, "durability")));
        printf("Puissance: %lld\n", json_integer_value(json_object_get(stats, "power")));
        printf("Combat: %lld\n", json_integer_value(json_object_get(stats, "combat")));

        json_t *appearance = json_object_get(hero, "appearance");
```

Description détaillée : Cette fonctionnalité permet à l'utilisateur de consulter les informations détaillées d'un super-héros spécifique en entrant son identifiant. Les détails affichés incluent:

- Powerstats (force, intelligence, vitesse, durabilité, puissance, combat)
- Apparence (genre, race, taille, poids)

L'interface présente ces informations de manière structurée et lisible

Aspects techniques :

On utilise à nouveau une boucle ou on test que la valeurs de l'objet ID est la même que celle entrée si c'est le cas on utilise la fonction `json_object_get()` pour récupérer la relation clef valeur du json et on utilise `json_integer_value()` pour y récupère uniquement la valeur et ensuite l'afficher on fait ça pour chaque caractéristique les une après les autre

Recherche et Filtre

Description détaillée : Notre application offre des fonctionnalités avancées de recherche et de filtrage:

- Recherche par nom (insensible à la casse)
- Filtrage par niveau de puissance (basé sur différentes statistiques)

Ces options permettent aux utilisateurs de trouver rapidement les super-héros qui les intéressent, même dans une base de données volumineuse.

Aspects techniques :

Tous les cas de figure sont ressemblant à celle pour l'affichage du héros par ID en changeant l'élément tester

Comparaison de Super-Héros

```

json_array_foreach(root, index, hero1) {
    if (json_integer_value(json_object_get(hero1, "id")) == id1) {
        json_array_foreach(root, index, hero2) {
            if (json_integer_value(json_object_get(hero2, "id")) == id2) {
                const char *name1 = json_string_value(json_object_get(hero1, "name"));
                const char *name2 = json_string_value(json_object_get(hero2, "name"));
                json_t *stats1 = json_object_get(hero1, "powerstats");
                json_t *stats2 = json_object_get(hero2, "powerstats");

                printf("\nComparaison entre %s %s : \n", name1, name2);
                printf("Intelligence: %d %d\n", json_integer_value(json_object_get(stats1, "intelligence")),
                    json_integer_value(json_object_get(stats2, "intelligence")));
                printf("Force: %d %d\n", json_integer_value(json_object_get(stats1, "strength")),
                    json_integer_value(json_object_get(stats2, "strength")));
                printf("Vitesse: %d %d\n", json_integer_value(json_object_get(stats1, "speed")),
                    json_integer_value(json_object_get(stats2, "speed")));
                printf("Durabilité: %d %d\n", json_integer_value(json_object_get(stats1, "durability")),
                    json_integer_value(json_object_get(stats2, "durability")));
                printf("Puissance: %d %d\n", json_integer_value(json_object_get(stats1, "power")),
                    json_integer_value(json_object_get(stats2, "power")));
                printf("Combat: %d %d\n", json_integer_value(json_object_get(stats1, "combat")),
                    json_integer_value(json_object_get(stats2, "combat")));
                json_decref(root);
                return;
            }
        }
    }
}

```

Description détaillée : Cette fonctionnalité permet aux utilisateurs de comparer deux super-héros côte à côte. La comparaison présente:

- Un tableau comparatif des powerstats
- Une visualisation graphique en mode texte (barres horizontales) pour faciliter la comparaison visuelle

Aspects techniques :

Personnalisation de Profil

```
void sauvegardeSuperHeros(int n, int Tab[], const char *filepath) {
    json_error_t error;
    json_t *root = json_load_file(filepath, 0, &error);

    if (!root) {
        fprintf(stderr, "Erreur de chargement JSON : %s\n", error.text);
        return;
    }

    json_t *new_root = json_array();
    size_t index;
    json_t *hero;

    for (int i = 0; i < n; i++) {
        json_array_foreach(root, index, hero) {
            if (json_integer_value(json_object_get(hero, "id")) == Tab[i]) {
                json_array_append(new_root, hero);
                break;
            }
        }
    }

    if (json_dump_file(new_root, "bin/SuperHeros_Saved.json", JSON_INDENT(4)) != 0) {
        fprintf(stderr, "Erreur de sauvegarde du fichier JSON.\n");
    } else {
        printf("Super-héros sauvegardés avec succès dans SuperHeros_Saved.json\n");
    }
}
```

Description détaillée : Notre application permet aux utilisateurs de gérer une liste de super-héros favoris:

- Ajout d'un super-héros aux favoris
- Suppression
- Modification

Les favoris sont enregistrés dans un fichier JSON séparé, permettant à l'utilisateur de retrouver sa liste personnalisée à chaque utilisation de l'application.

Aspects techniques :

on demande à l'utilisateur de choisir le nombre de superhéros qu'il souhaite garder dans son fichier favoris pour ensuite l'envoyer dans un fichier json qu'on peut manipuler simplement avec la bibliothèque jansson

Quiz Interactif

```
while (vies > 0) {
    id = rand() % nbSuperHeros;
    json_t *hero = json_array_get(root, id);
    name = json_string_value(json_object_get(hero, "name"));
    json_t *stats = json_object_get(hero, "powerstats");
    json_t *appearance = json_object_get(hero, "appearance");

    if ((rand() % 2) == 0) {
        printf("\nApparence:\n");
        printf("Genre: %s\n", json_string_value(json_object_get(appearance, "gender")));
        printf("Race: %s\n", json_string_value(json_object_get(appearance, "race")));

        json_t *height = json_object_get(appearance, "height");
        printf("Taille: %s (%s)\n", json_string_value(json_array_get(height, 0)), json_string_value(json_array_get(height, 1)));

        json_t *weight = json_object_get(appearance, "weight");
        printf("Poids: %s (%s)\n", json_string_value(json_array_get(weight, 0)), json_string_value(json_array_get(weight, 1)));

        printf("Couleur des yeux: %s\n", json_string_value(json_object_get(appearance, "eyeColor")));
        printf("Couleur des cheveux: %s\n\n", json_string_value(json_object_get(appearance, "hairColor")));
    } else {
        printf("\nPowerstats:\n");
        printf("Intelligence: %lld\n", json_integer_value(json_object_get(stats, "intelligence")));
        printf("Force: %lld\n", json_integer_value(json_object_get(stats, "strength")));
        printf("Vitesse: %lld\n", json_integer_value(json_object_get(stats, "speed")));
        printf("Durabilité: %lld\n", json_integer_value(json_object_get(stats, "durability")));
        printf("Puissance: %lld\n", json_integer_value(json_object_get(stats, "power")));
        printf("Combat: %lld\n", json_integer_value(json_object_get(stats, "combat")));
    }

    reponse = readline("Qui est ce super-héros ? ");
    if (strcasecmp(reponse, name) == 0) {
        score++;
        printf("Bonne réponse !\n");
    } else {
        vies--;
    }
}
```

Description détaillée : Le quiz interactif est une fonctionnalité ludique qui teste les connaissances des utilisateurs sur les super-héros:

- Questions générées aléatoirement basées sur la base de données
- Différents types de questions (identification par description, devinette par statistiques)
- Système de score avec suivi des meilleurs résultats

Cette fonctionnalité rend l'application non seulement informative mais aussi divertissante, permettant aux utilisateurs d'approfondir leurs connaissances sur les super-héros.

Aspects techniques : on utilise une formule pour récupérer un Id au hasard avec `rand()` puis on génère un nombre entre 0 et 1

0 : la question sera identification par descriptions/ 1 : devinette par statistiques

Difficultés rencontrées et solutions apportées

Difficulté 1 : Parsing du fichier JSON

Description : La manipulation des données JSON en C s'est révélée complexe, notamment pour extraire correctement les informations imbriquées et gérer les différents types de données (chaînes, nombres, tableaux, objets).

Solution : Nous avons utilisé la bibliothèque jansson pour faciliter le parsing JSON. Un petit tour du côté de la Documentation de la bibliothèque et les erreurs sont résolues

Difficulté 2 : Gestion des Bibliothèques

Description : L'installation des bibliothèques a été une grosse difficulté du côté de Mathis étant sous windows il y avait trop souvent des erreurs de compilation de fichier non trouvé etc...

Solution : L'utilisation de L'interface MSYS qui a permis de faciliter la gestion des bibliothèques de leurs installations et leurs mises à niveau

Commentaires et Suggestions

Commentaires

Le projet couvre la majorité des fonctionnalités attendues, conformément aux objectifs initiaux. Cependant, par manque de temps, nous n'avons pas pu approfondir l'aspect visuel de l'application. Si un délai supplémentaire nous avait été accordé, nous l'aurions consacré à la conception et à l'amélioration d'une interface graphique, afin de rendre l'outil plus ergonomique et agréable à utiliser.

Suggestions

En complément de l'amélioration de l'interface graphique, nous aurions souhaité concentrer nos efforts sur l'optimisation des performances de l'application.

L'objectif était d'améliorer la fluidité, de réduire les temps de réponse et de garantir une meilleure expérience utilisateur.

Auto-Évaluations

Si nous devions nous noter, nous nous donnerions la note de 4 sur 5. L'application fonctionne correctement malgré les difficultés, les contraintes que nous nous étions fixées ont été respectées, et l'essentiel est là. Cependant, nous savons que ce n'est pas un projet parfait. Il manque encore des améliorations sur le plan visuel et certaines fonctionnalités pourraient être plus raffinées ou bien ajoutées. Nous estimons qu'il remplit son rôle même s'il peut être amélioré avec plus de temps et de travail.

GitHub

Notre Répertoire GitHub est composé comme ci-dessous le fichier README contient tous les éléments à connaître pour utiliser et explorer le Projet

- README.md : Ce fichier même qui contient les informations du Projet
- main.c : Contient le code principal du programme.
- bin/ : Répertoire pour les fichiers exécutables.

Liens repository Github (Public):

<https://github.com/ChocSathan/SuperHeros>