

Projet LANGUAGE C

Groupe A-CHAMPION DES MATHS



Antoine ZAGO & Camilia TURKI

Sommaire :

Introduction.....	page 3
Fonctionnalités implémentées.....	page 4
Difficultés rencontrées.....	page 8
Commentaires et suggestions.....	page 9
Auto-évaluation.....	page 9
GitHub.....	page 10

Introduction:

Ce compte-rendu porte sur le projet du groupe A réalisé en Langage C. Ce projet a pour objectif de développer un programme destiné à la révision de notions mathématiques de niveau CM1. Il s'inspire d'une page web déjà existante (lien : <http://championmath.free.fr/>).

Le programme commence par l'affichage d'un menu interactif, puis invite l'utilisateur à saisir un choix. Ce choix détermine la fonction mathématique qui sera exécutée. Parmi les fonctionnalités proposées, on retrouve des opérations telles que l'addition, la multiplication, la division, mais aussi des exercices de conversion de durées et d'aires, permettant de tester les compétences de l'utilisateur.

Également, le programme comprend une gestion du nombre d'essais pour chaque question. Un système d'enregistrement des scores a également été mis en place, ce qui permet de suivre la progression de l'utilisateur au fil des exercices.

Fonctionnalités implémentées :

-Fonctionnalité des compétences:

Cette fonctionnalité regroupe l'ensemble des compétences mathématiques que l'utilisateur peut choisir de tester. Elle constitue la majeure partie du programme, en proposant différents types d'exercices adaptés au niveau CM1.

La fonctionnalité est rendue possible grâce à huit fonctions mathématiques distinctes, chacune correspondant à une compétence particulière. Ces fonctions sont les suivantes :

Addition, Soustraction, Soustraction, Multiplication, Tables de multiplication, Division, Conversions d'unités, Conversion en minutes-secondes, Ordre de grandeur

```
int addition() {  
    // Variables  
    int nb1 = rand() % 101, nb2 = rand() % 101, reponse, resultat = nb1 + nb2;  
    int score = 0;  
    // Programme  
    for (int essai = 1; essai < 4; essai++) {  
        printf("%d + %d = ?\n", nb1, nb2);  
        printf("Entrez le résultat : ");  
        scanf("%d", &reponse);  
        printf("\n");  
        if (essai == 1 && resultat == reponse) {  
            score = 10;  
            break;  
        } else if (essai == 2 && resultat == reponse) {  
            score = 5;  
            break;  
        } else if (essai == 3 && resultat == reponse) {  
            score = 1;  
            break;  
        } else {  
            printf("Mauvaise réponse, il vous reste %d essai(s)\n\n", 3 - essai);  
        }  
    }  
    printf("Le résultat est %d\n\n", resultat);  
    return score;  
}
```

-exemple de la fonction d'addition-

Toutes les fonctions portent également une partie destinée à faire la gestion des essais qui permet de déterminer le score de l'utilisateur.

- Fonctionnalité de sauvegarde :

Le programme permet de sauvegarder le nom d'utilisateur, le score et la date dans un fichier texte sous cette forme :

Nom	Score	Heure
antoine	54	27/03/2025 09:32:19

Pour cela on filtre les nouveaux joueurs des joueurs déjà enregistrer :

```
// On ouvre le fichier pour savoir si l'utilisateur est nouveau ou non
FILE *f = fopen("score.txt", "r+");
if (f == NULL) {
    printf("Impossible d'ouvrir le fichier.\n");
    exit(1);
}
// Tant que le fichier n'est pas fini
while (fscanf(f, "%s", nomf) != EOF) {
    // On compare les deux chaînes de caractère
    if (strcmp(nomf, nom) == 0) {
        // On actualise le score si l'utilisateur est déjà enregistré
        fprintf(f, "\t\t\t\t%d\t\t\t\t%s\n", score, tempsformate);
        ecrire = 1;
        break;
    }
}
fclose(f);
// Si l'utilisateur n'est pas déjà enregistré
if (ecrire == 0) {
    nouvel_utilisateur(score, nom, tempsformate);
}
```

On demande à l'utilisateur son nom au début du programme que l'on peut comparer à tous les noms du fichier.

Si il existe alors on écrase les informations par les nouvelles grâce à la fonction fprintf() à la suite de son nom.

Comme nous sommes en mode read/write les informations sont écrasées après la position du pointeur d'écriture seulement comme le fichier est divisé en ligne on ne supprime que les informations sur la ligne.

Si c'est un nouvel utilisateur, on change de mode d'écriture.

```
FILE *f = fopen("score.txt", "a");
if (f == NULL) {
    printf("Impossible d'ouvrir le fichier.\n");
    exit(1);
}
// On vérifie que le fichier n'est pas vide
fseek(f, 0, SEEK_END); // Se place à la fin du fichier
long taille = ftell(f); // Récupère la position (taille du fichier)
if (taille == 0) {
    // Si le fichier est vide, on ajoute la première ligne de description
    fprintf(f, "Nom\t\t\t\tScore\t\t\t\tHeure\n");
}
// On ajoute le nouvel utilisateur
fprintf(f, "%s\t\t\t\t%d\t\t\t\t%s\n", nom, score, tempsformate);
fclose(f);
```

On passe en mode append pour ajouter une ligne à la suite de la dernière existantes.

C'est la fonctionnalité qui a posé le plus de problème surtout avec la gestion des différents modes d'ouverture de fichier. On rentrera plus en détails dans la partie suivante.

- Fonctionnalité de récupération de score :

Pour cela on suit la même méthode qu'avec le filtre des utilisateur seulement on récupère en plus le score.

```
// On ouvre le fichier
FILE *f = fopen("score.txt", "r");
// On verifie que le fichier existe
if (f == NULL) {
    fclose(f);
}
// Tant que le fichier n'est pas fini
while (fscanf(f, "%s\t\t\t\t\t\t%d", nomf, &scoref) != EOF) {
    // On vérifie que les noms sont identiques
    if (strcmp(nomf, nom) == 0) {
        score = scoref;
        break;
    }
}
fclose(f);
printf("Votre score est de %d\n\n", score);
return score;
```

On vérifie à chaque ligne du fichier si le nom entré en début de programme correspond à celui d'une ligne. Si il y a correspondance, l'utilisateur récupère son score et le résultat des exercices y sera ajouté. Sinon l'utilisateur part de zéro.

Difficultés Rencontrées et Solutions Apportées

Le problème majeur qui a été rencontré est le système de sauvegarde. On cherche à sauvegarder le nom de l'utilisateur son score et la date a chaque fin d'exercice dans un fichier texte.

Premièrement, on a d'abord cherché à enregistrer chaque session d'exercice sans récupération de score. Il a fallu apprendre et comprendre les fonctions sur les fichiers.

A commencé par les paramètres de la fonction `fopen("nom_fichier.txt","mode")`. Nous cherchions à sauvegarder en mode write ("w") ce qui avait pour effet de supprimer les données antérieures.

Cependant grâce à la documentation en ligne nous avons appris à utiliser le mode append ("a") et read/write ("r+").

Cela nous permet d'ajouter une ligne d'enregistrement sans avoir à supprimer les données déjà existantes.

Puis nous avons voulu récupérer le score à partir d'un nom d'utilisateur déjà enregistré. Nous avons alors bataillé avec la fonction `fscanf()` car nous n'arrivions pas à récupérer ni le score ni le nom d'utilisateur.

C'est là que nous avons compris que comme les tabulations sont des caractères (`\t`), elles pouvaient servir à rendre le fichier lisible en plus de servir de séparateur entre les différentes informations d'une ligne.

Suite à cela nous n'avions plus qu'à bien utiliser les différents types d'ouvertures de fichier pour sauvegarder correctement les informations.

Il serait intéressant pour une prochaine fois d'étudier un autre système de sauvegarde pour pouvoir le comparer à celui-ci par exemple dans une base de données.

Commentaires et Suggestions

Remarques générales :

Les consignes du projet étaient bien structurées et permettait une découpe efficace entre les différents membres du binomes.

Suggestions d'amélioration :

Le cours n'englobant pas l'intégralité du projet, ce dernier demande de s'aider de la documentation internet pour pouvoir être réalisable.

Focus sur les Apprentissages :

Le projet nous a appris les bases de la programmation en C ainsi que la gestion de données dans un fichier texte. C'est une bonne première approche du langage.

Auto-évaluation

Les consignes de base ont été respectées et le projet répond aux attentes.

Chaque membre du binôme a réalisé sa partie du code.

Quelques exercices supplémentaires ont été ajoutés.

Un interface graphique pourrait être apprécié pour une prochaine fois.

Code et GitHub

Lien vers le dépôt GitHub : https://github.com/ZagoAnt/Projet_C_SujetA