




LANGAGE C

RAPPORT DE PROJET

GROUPE A

Linda Hmandi
Yasmine Gmiza

28/04/2025



SOMMAIRE:

1. Introduction
2. Fonctionnalités implémentées
3. Difficultés et Solutions
4. Commentaires et Suggestions
5. Auto-évaluation
6. Code et GitHub

1. Introduction

- Le projet choisi est un jeu éducatif qui regroupe plusieurs notions mathématiques à travers des mini-jeux (addition, soustraction, multiplication, tables de multiplication, division, parité).
- À la fin du jeu, le joueur reçoit un score basé sur ses compétences. Pour chaque question, il dispose de trois tentatives (10 points s'il répond correctement au premier essai, 5 points au deuxième essai, 1 point au troisième essai).
- Pour réaliser ce code on a utilisé Code Blocks.

2. Fonctionnalités Implémentées

- Dès que le joueur entre dans le jeu, il lui est demandé d'entrer son nom d'utilisateur:

Si l'utilisateur existe déjà, son score précédent ainsi que la date correspondante sont affichés, sinon, un nouveau profil est créé.

- À l'aide d'une structure switch, l'utilisateur choisit le mini-jeu auquel il souhaite jouer (addition, soustraction, multiplication, tables de multiplication, division, parité).

```
printf("+-----+\n");
printf("|1 : Addition           |\n");
printf("|2 : Soustraction        |\n");
printf("|3 : Multiplication       |\n");
printf("|4 : Tables des multiplications |\n");
printf("|5 : Divisions            |\n");
printf("|6 : Pariete              |\n");
printf("|0 : Sortir du jeu        |\n");
printf("+-----+\n");
printf("Quel est votre choix ?\n");
```

- Le joueur dispose de trois tentatives grâce à une boucle while.

```
while (essais < 3) {  
    printf("%d + %d = ?\n", x, y);  
    scanf("%d", &res);  
    essais++;  
    if (res == s) {  
        return (essais == 1) ? 10 : (essais == 2) ? 5 : 1;  
    } else {  
        printf("Incorrect. Essai %d/3\n", essais);  
    }  
}
```

- Pour les fonctions d'addition, de soustraction et de multiplication, nous utilisons srand pour générer des valeurs aléatoires comprises entre 0 et 100.

```
srand(time(NULL));  
x=rand()%111;  
y=rand()%111;
```

- Pour la table de multiplication, nous utilisons une boucle for afin d'afficher des équations de multiplication de 1 à 10.

- Pour enregistrer les scores, nous avons créé deux fichiers :

Le premier fichier (scores.txt) stocke le nom de l'utilisateur, son score et la date/heure.

Le deuxième fichier temporaire (scores_tmp.txt) assure l'unicité des utilisateurs et remplace l'ancien score par le nouveau si nécessaire.

```
if (fichierLecture != NULL) {
    while (fgets(ligne, sizeof(ligne), fichierLecture)) {
        if (sscanf(ligne, "%[^;];%[^;];%d", nomLu, dateHeureLu, &scoreLu) == 3) {
            if (strcmp(nomLu, user.nom) == 0) {
                // L'utilisateur existe, on écrit la nouvelle version
                fprintf(fichierEcriture, "%s;%s;%d\n", user.nom, user.temps, user.score);
                trouve = 1;
            } else {
                // Sinon, on copie la ligne telle quelle
                fprintf(fichierEcriture, "%s", ligne);
            }
        }
    }
    fclose(fichierLecture);
}

// Si l'utilisateur n'existait pas, on ajoute à la fin
if (!trouve) {
    fprintf(fichierEcriture, "%s;%s;%d\n", user.nom, user.temps, user.score);
}
```

```
remove("scores.txt");  
rename("scores_tmp.txt", "scores.txt");
```

```
if (fichier != NULL) {  
    while (fgets(ligne, sizeof(ligne), fichier)) {  
        // Lecture sous forme Nom;Date;Score  
        if (sscanf(ligne, " %[^\n]; %[^\n]; %d", nomLu, dateHeureLu, &scoreLu) == 3) {  
            if (strcmp(nomLu, user->nom) == 0) {  
                printf("Bienvenue %s ! Dernier score enregistre le %s : %d points\n", nomLu, dateHeureLu, scoreLu);  
                trouve = 1;  
            }  
        }  
    }  
    fclose(fichier);  
}  
  
if (!trouve) {  
    printf("Bienvenue %s ! Aucun score trouve, vous commencez a 0.\n", user->nom);  
}
```

3. Difficultés et Solutions

- La logique de créer deux fichiers n'était pas familière pour nous, ce qui nous a causé des difficultés au début pour résoudre le problème de gestion des scores.
- Ensuite, il a fallu bien manipuler les deux fichiers afin de conserver dans le fichier principal chaque utilisateur une seule fois, sans doublon.
- Après plusieurs essais et beaucoup de persévérance, nous avons réussi à détecter les erreurs présentes et à les corriger au fur et à mesure.

4. Commentaires et Suggestions

- Ce projet a été une excellente occasion d'apprendre davantage sur la manipulation de fichiers et de mieux s'habituer à l'écriture en langage C, notamment en travaillant avec des notions comme la gestion d'adresses mémoire.
- Pour améliorer le projet, nous pourrions Éviter la redondance de la partie "tentatives" en créant une fonction spécifique qui gérerait toutes les tentatives, afin de rendre le code plus propre et mieux organisé.

5.Auto-évaluation

- Le projet s'est bien déroulé et nous n'avons pas rencontré de difficultés majeures. Cette expérience a été très enrichissante.
- En ce qui concerne les objectifs futurs, nous visons à réaliser des projets plus complexes, impliquant une utilisation plus poussée des pointeurs et une gestion avancée des fichiers.
Nous espérons que les compétences acquises au cours de ce projet nous permettront de travailler avec plus de fluidité et d'aborder des projets plus ambitieux à l'avenir.

6. Code et GitHub

- Lien du code: https://github.com/Indhmi/math_game.git
- Pour réaliser ce code j'ai utilisé CodeBlocks
- l'utilisateur commence à entrer son nom
- un message de bienvenue et de son score précédent (si il existe) s'affiche
- l'utilisateur fait un choix à l'aide de la structure switch pour jouer à des mini-jeux
- le joueur à trois tentatives pour chaque question si il a la bonne réponse dès la première tentative il gagne 10points sinon 5 sinon 0
- le score sera modifié si le joueur existe sinon une nouvelle ligne sera ajoutée au fichier avec la date