Année 2023-2024

RAPPORT DE PROJET

PROJET INFORMATIQUE
CER-J-O

Projet réalisé par :
Angel PASQUIN
Jean-Paul ALPHONSERAJ
Mohammed MOUBTAKIR

Projet encadré par :
Mohamed HADDACHE

SOMMAIRE

I. OBJECTIFS DU PROJET

Dans le cadre des Jeux Olympiques, l'objectif du projet CER-J-O est de développer un logiciel à destination de l'entraineur de l'équipe d'athlétisme de France lui permettant de :

- Suivre l'évolution des performances de ses athlètes sur 5 épreuves (le 100m, le 400m, le 5000 m, le marathon et le relais 4*400 m)
- Identifier les 3 meilleurs athlètes pour une discipline donnée afin de les sélectionner pour les Jeux Olympiques

II. DESCRIPTION DES BESOINS FONCTIONNELS

II.1 - Fonctionnalités du logiciel et règles fonctionnelles

Les principales fonctionnalités du logiciel à développer sont :

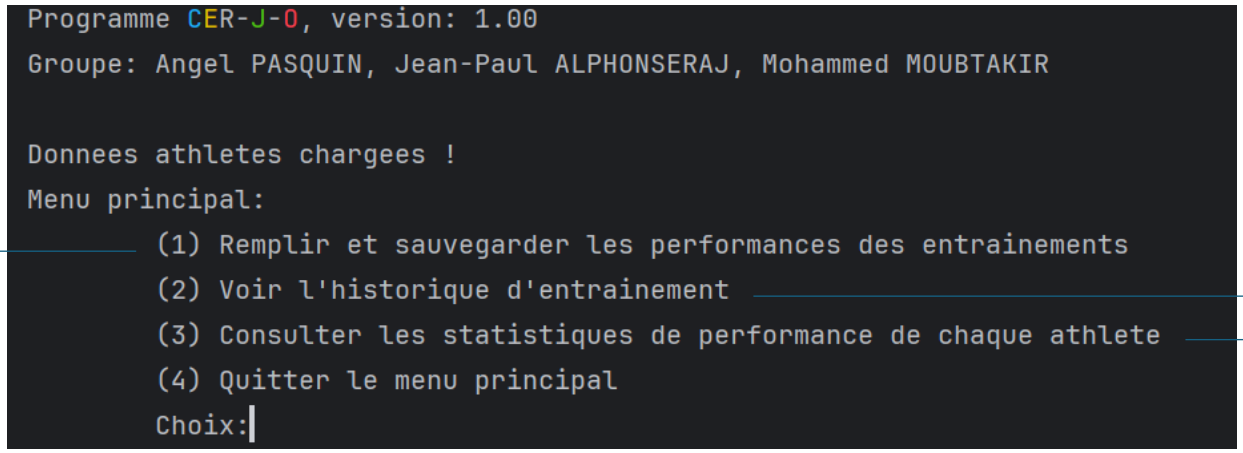
- Créer une fiche athlète qui contiendra les différents entrainements et performances associées
- Enregistrer les performances d'un nouvel entrainement par athlète et pour une épreuve et une date donnée
- Consulter l'historique des entrainements par athlète, par épreuve et/ou par date
- Consulter les statistiques de performances de chaque athlète (meilleur temps, pire temps et la moyenne pour une épreuve donnée)
- Consulter le déroulement complet de l'entrainement d'un relais 4*400 m
- Consulter la progression des athlètes et l'évolution de leurs performances pour une même épreuve donnée
- Lister les 3 meilleurs athlètes pour une épreuve donnée (meilleur temps moyen)

Les règles fonctionnelles sont :

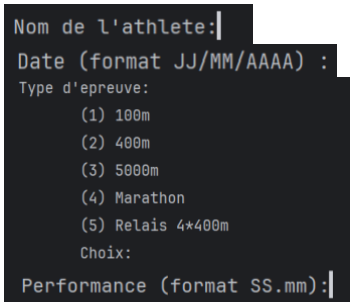
Catégorie	**Règles fonctionnelles**
Gestion des épreuves	Performances d'un athlète stocké dans un fichier unique
Gestion des entrainements	Date de l'entrainement, type d'épreuve, temps de l'athlète
Gestion des entrainements	0 à N entrainements par jour (sauf le relais 4*400m, 1 seul)
Progression des athlètes	Pour une même épreuve, différence de temps d'un athlète entre 2 dates
Evolution des performances	Pour une même épreuve, différence de temps d'un athlète entre 2 entrainements (2 dates)
Relais 4*400m	1 seul entrainement de relais par jour
Relais 4*400m	4 athlètes uniquement
Relais 4*400m	Position des athlètes pendant le relais
Sélection des athlètes	3 meilleurs athlètes pour une discipline donnée

II.2 – Cinématique des écrans

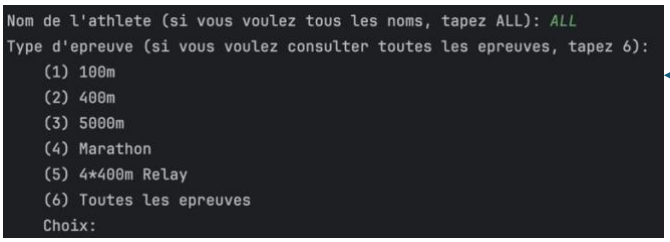
Menu principal



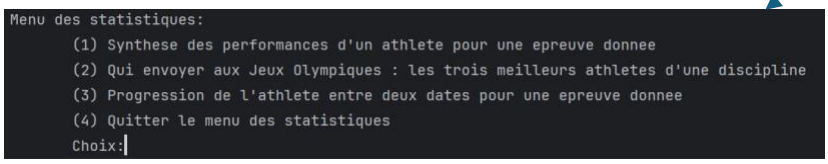
(1) Créer un enregistrement Athlète/performance



(2) Historique d'entrainement



(3) Menu des statistiques



III. SPECIFICATIONS TECHNIQUES

III.1 – Système d'exploitation et environnement utilisé

OS utilisé : Windows
Application utilisée pour développer et déboguer : Online GBD (https://www.onlinegdb.com) au départ, puis changement vers l'EDI (Environnement de Développement Intégré) CLion (https://www.jetbrains.com/fr-fr/clion/).

III.2 – Stockage des données

Pour stocker les données, nous écrivons les fichiers dans un répertoire nommé « data ». Il y a autant de fichiers que d'athlètes.

Le format des enregistrements du fichier est le suivant :

<AAAAMMDD>;<TYPE_EPREUVE>;<TEMPS_PERFORMANCE>;<POSITION>

- AAAAMMDD : format de la date
- TYPE_EPREUVE : 0 (100m), 1 (400m), 2 (5000m), 3 (Marathon), 4 (Relais 400m)
- TEMPS_PERFORMANCE : format float
- POSITION : la position (1-4) est valorisée pour le relais 4*400m, sinon la valeur est 0

Ecriture
La première chose à vérifier est l'existence de ce répertoire (instruction « opendir »). Si le répertoire n'existe pas, il faut le créer en utilisant l'instruction « mkdir ». Une fois dans le répertoire, on peut, soit utiliser l'instruction « file » en utilisant le paramètre write « w » pour créer le fichier, soit utiliser le paramètre append « a » pour ouvrir en mode ajout. Si le fichier n'existe pas, il sera alors créé. Sinon, son contenu sera conservé et la position courante sera à la fin du fichier.
On peut alors écrire le contenu souhaité en utilisant l'instruction « fprintf ». Pour fermer le fichier, l'instruction « fclose » est utilisée.

Lecture
On ouvre le fichier en lecture (mode « r ») en utilisant l'instruction « file », et on lit les enregistrements en utilisant l'instruction « fgets ». Pour fermer le fichier, il faut utiliser l'instruction « fclose ».

IV. ORGANISATION DE L'EQUIPE

Démarrage du projet :

Au démarrage du projet, nous avons eu du mal à nous organiser ; d'une part c'est la première fois que nous travaillions ensemble sur un projet, et d'autre part, nous avons eu des problèmes logistiques pour nous retrouver, car nous ne vivions pas tous dans la même région. Nous avons donc effectué la majorité de nos points de synchronisation à distance et créé un groupe WhatsApp pour améliorer notre communication.

Dans un premier temps, nous avons pris le temps de bien comprendre le sujet. Parfois, nous avions des interprétations différentes des consignes fournis et des interrogations sur la terminologie utilisée. Par exemple,

- Pour la consigne suivante : « On aimerait pouvoir indiquer la différence de temps pour une même épreuve entre deux entrainements (entre deux dates) », nous avons eu des difficultés à se mettre d'accord sur ce que nous devions faire. En effet, nous nous sommes demandé s'il fallait soit faire la différence entre 2 temps soit faire la différence entre 2 périodes. Nous avons finalement opté pour la première proposition.

- Nous nous sommes aussi interrogés sur la définition d'un entraînement. Un entraînement pouvait-il regrouper plusieurs épreuves dans une même journée (sauf pour le 4*400m) ?

Organisation du projet :

Une fois que nous avions une compréhension commune du sujet, nous avons identifié 5 blocs principales de fonctionnalités :

Bloc de Fonctions	Description
Fonctions communes	Fonctions communes, utilisées par les fonctions lecture/écriture, statistiques et historiques, permettant la gestion des saisies, de l'affichage, de la suppression des caractères espaces, des tris des entrainements par date et tri du classement des performances JO
Fonctions Menu principal	Fonctions de gestion du menu permettant d'accéder aux fonctionnalités de l'application
Fonctions statistiques	Fonctions de gestion des statistiques des performances des entraînements des athlètes
Fonctions historiques	Fonctions de gestion de l'historique des performances des entraînements des athlètes
Fonctions lecture/écriture	Fonctions de gestion des écritures et des lectures des données de performances des entraînements des athlètes

Au départ, chacun devait prendre 1 ou 2 blocs de fonctions et était responsable de sa conception technique, du codage et des tests unitaires. Dans la journée, chacun avançait de son côté et nous faisions des points 2 fois par semaine pour échanger sur l'avancement des sujets, intégrer les blocs de fonctions dans un programme global et faire de potentielles retouches/correction sur le code. Très rapidement, nous nous sommes rendu compte que cette organisation ne fonctionnait pas pour les raisons suivantes :

1. **Manque de communication quotidienne** : en travaillant de manière isolée et en ne faisant des points que deux fois par semaine, nous manquions de communication continue ce qui a entrainé des malentendus et des déconnexions sur l'état d'avancement et les décisions techniques
2. **Problèmes d'intégration** : l'intégration des blocs de fonctions dans un programme global deux fois par semaine a révélé des incompatibilités et des conflits de code. Ces problèmes prenaient du temps à résoudre et retardaient le projet
3. **Dépendances non gérées** : Chaque bloc de fonctions n'était pas toujours développé de manière totalement indépendante. Il y avait des interdépendances que nous n'avions pas anticipées, ce qui rendait l'intégration difficile et augmentait le nombre de retouches nécessaires.
4. **Problèmes de qualité et de cohérence** : Chaque personne étant responsable de sa propre conception, codage et tests unitaires, il y avait des variations dans la qualité du code et les méthodes utilisées. Cela rendait le code global incohérent et difficile à maintenir.
5. **Réactivité Insuffisante aux problèmes** : Les retouches et corrections étaient souvent découvertes et traitées tardivement, lors des points bihebdomadaires. Cela créait des retards et des cycles de correction prolongés.
6. **Surcharge de Travail pour les retouches** : En ayant des retouches à faire après chaque point, nous passions plus de temps à corriger les erreurs et à intégrer les modifications qu'à avancer sur de nouvelles fonctionnalités.
7. **Manque de Vision globale** : Chacun étant focalisé sur ses propres blocs de fonctions, nous avions une vision fragmentée du projet. Cela empêchait une compréhension claire des objectifs globaux et de la direction du projet.
8. **Difficulté à Partager les connaissances** : Les connaissances restaient souvent isolées avec la personne responsable de chaque bloc, rendant difficile le partage d'informations critiques et la montée en compétence de l'équipe sur l'ensemble du projet.

Au milieu du projet, nous avons décidé de changer l'organisation et de nommer un développeur principal en charge de reprendre le codage de l'ensemble des blocs de fonctions. Quant aux autres membres de l'équipe, ils avaient la responsabilité de tester toutes les fonctions du programme. Pour améliorer la réactivité, dès qu'une erreur était trouvée, les membres de l'équipe le notifiaient sous WhatsApp. Les deux points hebdos ont été maintenus pour échanger sur la correction des bugs et faire des retouches concertées sur le programme.

Lors d'un blocage technique, nous organisions un point adhoc. Cette organisation s'est avérée beaucoup plus efficace mais a mis beaucoup de pression sur le membre de l'équipe en charge du codage. De plus, il s'est avéré que la détection des bugs était elle aussi plus efficace du fait que les testeurs n'étaient pas partie prenante du codage.

Répartition des activités :

ACTIVITES	ANGEL	JEAN PAUL	MOHAMMED
Cadrage du projet			
1.Compréhension du sujet	X	X	X
2.Définition des objectifs du projet	X	X	X
Organisation de l'équipe			
1.Répartition du travail	X	X	X
2.Planification des activités	X	X	X
Conception			
1.Cinématique des écrans	X	X	X
2.Conception technique	X	X	X
Développement (Codage & tests unitaires)			
1.Codage Fonctions communes	X		
2. Codage Fonctions statistiques	X		
3. Codage fonctions historiques	X		
4. Codage Fonctions Lecture/écriture	X		
Tests Intégration & Validation			
1.Tests Fonctions Communes	X	X	X
2.Tests Fonctions statistiques	X	X	X
3.Tests Fonctions historiques	X	X	X
4.Tests Fonctions Lecture/écriture	X	X	X
Rapport de projet			
1.Plan du rapport	X	X	X
2.Rédaction	X	X	X
3.Assemblage	X		

V. DEVELOPPEMENT TECHNIQUE

Le programme est développé en langage C et s'exécute sur l'OS Windows. Il a aussi été testé avec succès sur l'OS MacOS.

L'EDI (Environnement de Développement Intégré) CLion (https://www.jetbrains.com/fr-fr/clion/) a été utilisé pour développer et déboguer le programme.

Le programme comprend 5 « modules » :

- Fonctions communes
- Fonctions menu principal
- Fonctions historiques
- Fonctions statistiques
- Fonctions lecture/écriture

V.1 – Fonctions communes

Description : Fonctions communes utilisées par les autres « modules »

Fichiers C et header : common.c, common.h

Fonctions	Description
supprimerEspacesChaineCaract	Suppression des caractères espaces multiples dans une chaîne de caractères
lireDate	Permet la saisie de la date (format: DD/MM/YYYY) , vérifie si la date est valide et la retourne en format unsigned long (format : YYYYMMDD)
afficherDate	Retourne la date au format "JJ/MM/AAAA" pour affichage
existanceFichier	Vérifie si le fichier existe
lireChaine	Permet la saisie d'une chaîne de caractères
lireLong	Permet la saisie d'une valeur de type "long"
lirePerformanceS	Lire performance format "SS.mm"
lirePerformanceM	Lire performance format "MM:SS.mm"
lirePerformanceL	Lire performance format "HH:MM:SS.mm"
triRapideDateEntrainements	Fonction de tri rapide du tableau des entraînements
triRapidePerfJO	Fonction de tri rapide du tableau de classement des performances JO
afficherPerformance	Fonction d'affichage des performances

V.2 – Fonctions menu principal

Description : Programme principal de CER-J-O

Fichier C : athletesJO.c

Fonctions	Description
main	Programme principal de CER-J-O, gestion du menu principal

V.3 – Fonctions historiques

Description : Gestion de l'historique des performances des entrainements des athletes

Fichiers C et header : history.c, history.h

Fonctions	Description
visualiserHistoriquetsAthletes	Visualisation de l'historique des performances des entraînements des athlètes

V.4 – Fonctions statistiques

Description : Gestion des statistiques des performances des entraînements des athlètes

Fichiers C et header : statistics.c, statistics.h

Fonctions	Description
performancesAthlete	Affiche les statistiques de performances (meilleure, pire, moyenne) de l'athlète pour un type d'épreuve
performancesJO	Affiche le classement des athletes pour les JO pour un type d'épreuve donné
progressionAthlete	Affiche la progression de l'athlète pour entre 2 dates d'entrainements
visualiserStatistiques	Gestion du menu des statistiques

V.5 – Fonctions lecture/écriture

Description : Gestion des écritures et des lectures des données de performances des entraînements des athlètes

Fichiers C et header : manageAthletesData.c, manageAthletesData.h

Fonctions	Description
enregistrerPerformancesEntrainementsAthletes	Enregistre dans le répertoire "./data" les performances des entraînements d'un athlète
chargerPerformancesEntrainementsAthletes	Lecture des données des fichiers de performances des entraînements des athlètes et chargement en mémoire

VI. TESTS DU PROGRAMME

VI.1 – Jeux de données

Les jeux de données ont été constitués manuellement de 2 manières :

- Soit par le programme, via la fonction « Remplir et sauvegarder les performances des entrainements »

- Soit par l'ajout de données directement dans les fichiers

NB : le choix stockage au format texte lisible permet de pouvoir éditer facilement les fichiers.

VI.2 - Tests techniques

Pour s'assurer que le programme fonctionne correctement, chaque fonctionnalité a été testée dans le but de vérifier que le programme ne comporte pas de bug.
Premièrement, il fallait s'assurer que le répertoire « data » existait afin de stocker les fichiers des athlètes. Ensuite, il fallait tester toutes les fonctionnalités du programme.

VI.3 – Tests fonctionnels

1. Vérification de la validité des dates

Test 1.1: Date valide

```
Date (format JJ/MM/AAAA) :01/01/2024
```

Test 1.2: Date invalide (jour)

```
Date (format JJ/MM/AAAA) :32/01/2024
Le jour n'est pas valide
```

Test Case 1.3: Date invalide (mois)

```
Date (format JJ/MM/AAAA) :01/13/2024
Le mois n'est pas valide
```

2. L'enregistrement des performances et des entraînements

Test 2.1: Nouveau fichier d'athlète

```
 Athelete trouve: Angel, nb entrainement 0
Nombre d'entrainement: 1
Creer un fichier ./data/Angel.txt
Performance et entrainement de Angel ont ete enregistres dans le fichier ./data/Angel.txt.
```

Test 2.2: Ajout d'entraînement à un athlète existant

```
 Athelete trouve: Angel, nb entrainement 2
Nombre d'entrainement: 3
Creer un fichier ./data/Angel.txt
Performance et entrainement de Angel ont ete enregistres dans le fichier ./data/Angel.txt.
```

3. Consultation de l'historique des entraînements

Test 3.1: Historique pour un athlète spécifique

```
Nom de l'athlete Angel, date de l'entrainement: 01/01/2024, epreuve: 100m, perf.: 10.23
Nom de l'athlete Angel, date de l'entrainement: 20/05/2024, epreuve: 100m, perf.: 10.44
Nom de l'athlete Angel, date de l'entrainement: 20/05/2024, epreuve: 100m, perf.: 10.34
Nom de l'athlete Angel, date de l'entrainement: 21/05/2024, epreuve: 100m, perf.: 10.45
Nom de l'athlete Angel, date de l'entrainement: 25/05/2024, epreuve: 100m, perf.: 10.12
Nom de l'athlete Angel, date de l'entrainement: 25/05/2024, epreuve: 100m, perf.: 10.14
Nom de l'athlete Angel, date de l'entrainement: 26/05/2024, epreuve: 100m, perf.: 10.78
```

Test 3.2: Historique pour tous les athlètes

```
 de l'entrainement: 01/01/2024, epreuve: 100m, perf.: 10.00
Nom de l'athlete Angel, date de l'entrainement: 02/01/2024, epreuve: 100m, perf.: 10.27
Nom de l'athlete Angel, date de l'entrainement: 03/01/2024, epreuve: 100m, perf.: 10.36
Nom de l'athlete JeanPaul, date de l'entrainement: 01/01/2024, epreuve: 100m, perf.: 10.00
Nom de l'athlete JeanPaul, date de l'entrainement: 02/01/2024, epreuve: 100m, perf.: 10.53
Nom de l'athlete JeanPaul, date de l'entrainement: 03/01/2024, epreuve: 100m, perf.: 10.68
Nom de l'athlete Mohammed, date de l'entrainement: 01/01/2024, epreuve: 100m, perf.: 10.00
Nom de l'athlete Mohammed, date de l'entrainement: 02/01/2024, epreuve: 100m, perf.: 10.48
Nom de l'athlete Mohammed, date de l'entrainement: 03/01/2024, epreuve: 100m, perf.: 10.98
```

4. Consultation des performances d'un athlète

Test 4 : Performances d'un athlète

```
Nom de l'athlete: Angel, Epreuve: 100m, Perf.[meilleure: 10.00, pire: 10.36, moyenne: 10.21]
```

5. Performance des trois meilleurs athlètes

Test 5 : Top 3 des athlètes

```
Classement des athletes pour l'epreuve: 100m
        #1, nom athlete: Mohammed, moyenne perf.: 3.50
        #2, nom athlete: JeanPaul, moyenne perf.: 5.20
        #3, nom athlete: Angel, moyenne perf.: 10.21
```

6. Progression d'un athlète

Test 6 : Progression d'un athlète

```
Nom de l'athlete: Angel
        Perf. de debut: 10.00s, Perf. de fin: 10.36s, Delta perf.: +0.36s
```

VII. DIFFICULTES TECHNIQUES, SOLUTIONS APPORTEES ET RESULTATS

Difficultés techniques	Solutions apportées	Résultats
Gestion des accents de la langue Française : apparition de caractères spéciaux à la place des accents	Sol1 : Passer le texte du Français à l'Anglais pour supprimer les problèmes d'accents Sol2 : Supprimer les accents	Décision de tout garder en Français mais de supprimer les accents (Sol2), l'entraineur étant Français
Certaines instructions en langage C dépendent de l'OS utilisé (Linux ou Windows) : exemple affichage des couleurs	Recherches de commandes équivalentes entre Linux et Windows	Problème résolu
Gestion de la saisie des temps des épreuves est différente suivant que l'on saisit un temps pour un 100m (format SS.mm) ou un temps pour un marathon (HH:MM:SS.mm)	Suivant le type d'épreuve le format de saisie est différent	Problème résolu, les fonctions de saisie (lirePerformanceS, lirePerformanceM, lirePerformanceL) sont différentes suivant le type d'épreuve
Affichage de la saisie des temps des épreuves est différente suivant, par exemple, que l'on affiche un temps pour un 100m (format SS.mm) ou un temps pour un marathon (HH:MM:SS.mm)	Suivant le type d'épreuve le format d'affichage est différent	Problème résolu, la fonction d'affichage (afficherPerformance) permet de gérer les différents types d'épreuves
Gestion de l'écriture des performances des entraînements dans le fichier athlète diffère si l'on saisit une date antérieure à la date du dernier enregistrement.	Dans ce cas, on trie (tri rapide) les performances de l'athlète de manière à garder la chronologie des entrainements et des épreuves et l'on ré-écrit dans le fichier l'ensemble des données de l'athlète (mode « w »). NB : lorsque la chronologie est respectée le fichier est ouvert en mode append (« a ») et la performance de l'entrainement est stockée à la fin du fichier.	Solution proposée implémentée