

Choudhury Dimittri
Achour Hajar
Sayed Sanem
MEF 1

RAPPORT DU PROJET D'INFORMATIQUE S2

Nous avons commencé par un sondage pour établir un classement personnel de nos trois sujets favoris, le thème des Jeux Olympiques a émergé en tête des résultats. Après avoir réalisé le choix du thème, nous avons immédiatement fixé des dates de rendez-vous, que ce soit en présentiel ou même en distanciel, afin de mieux gérer notre temps et de nous assurer du rendu du programme en temps voulu. Le 24 avril, nous avons choisi le sujet des Jeux Olympiques. Le 7 mai, nous avons fait une réunion à Condorcet pour discuter des premières idées et établir un plan de travail. Ce jour-là, nous avons mis par écrit toute la structure du programme en français et avons réparti les tâches entre les membres de l'équipe. Le 15 mai a marqué un avancement significatif du programme. Nous avons commencé à coder les premières fonctionnalités et à intégrer nos idées dans un fichier Replit que nous avons créé afin de coder tous ensemble sur un même espace de travail. Le 19 mai, nous avons tenu une session de travail en ligne via notre groupe WhatsApp. Nous avons discuté des progrès réalisés et résolu quelques problèmes techniques que nous décrirons plus tard. Le 20 mai, nous avons continué le travail sur le programme et commencé la rédaction du rapport. Nous avons également échangé des idées et des retours sur le groupe WhatsApp pour affiner notre approche. Le 21 mai, nous nous sommes réunis en présentiel pour poursuivre le développement du programme. Nous avons intégré de nouvelles fonctionnalités, ajusté le code existant et vérifié l'interopérabilité des différentes parties du programme. Le 22 mai a été dédié à la finalisation des détails du programme lors d'une autre rencontre en personne. Nous avons peaufiné les fonctionnalités, corrigé les derniers bugs et effectué des tests complets pour nous assurer de la stabilité et de la fiabilité du programme. Enfin, le 24 mai, nous avons achevé le programme et finalisé le rapport. Nous avons passé en revue l'ensemble du projet pour nous assurer que tout était en ordre avant de le soumettre, incluant une dernière vérification de la documentation et des commentaires dans le code pour garantir une compréhension claire de notre travail. Cette approche structurée et collaborative nous a permis de

mener à bien notre projet tout en optimisant notre temps et nos ressources. La communication continue et l'engagement de chaque membre ont été essentiels pour surmonter les défis et aboutir à un programme robuste et fonctionnel.

Pour ce qui est du fonctionnement du groupe, nous avons travaillé individuellement, chacun de notre côté, en faisant de notre mieux pour avancer. Pour faciliter la collaboration, nous avons donc ouvert un fichier Replit partagé où chacun pouvait contribuer à l'avancement du code. Le groupe WhatsApp nous servait à nous tenir informés des modifications apportées au programme et assurer une communication continue. Chacun a fait de son mieux pour coder les différentes parties du projet. Ensuite, nous avons combiné nos codes respectifs, en sélectionnant les fonctions les plus pertinentes de chaque personne. Par exemple, nous avons intégré une fonction avec une grande plus-value même si plus complexe, à savoir la régression linéaire et son affichage sur un graphique (avec gnuplot). Nous avons pensé que cette méthode nous permettrait de progresser plus rapidement et efficacement. En tirant parti des compétences individuelles de chacun, nous avons pu créer un programme plus complet et fonctionnel, surmontant ainsi les défis techniques rencontrés tout au long du projet. Grâce à cette approche collaborative, nous avons optimisé notre travail et accéléré notre avancement. Enfin, nous avons accordé une attention particulière à la documentation du code et à la rédaction de commentaires clairs et précis. Cela a permis à chaque membre de comprendre les contributions des autres et de s'assurer que le programme final était non seulement fonctionnel mais aussi maintenable et extensible. Lors de la réalisation de ce programme de gestion des performances des athlètes, nous avons rencontré plusieurs défis et apporté diverses améliorations pour résoudre ces problèmes. Nous avons remarqué qu'il était primordial de vérifier que les dates proposées par l'utilisateur étaient correctes. Pour cela, nous avons ajouté une fonction `is_valid_date` qui vérifie la validité des dates au format YYYY-MM-DD. Cette fonction assure que le format est respecté et que les valeurs des mois et des jours sont réalistes, considérant les années bissextiles pour le mois de février par exemple. Ensuite, initialement, la fonction `create_athlete_file` n'était pas suffisante pour garantir la création des fichiers des athlètes dans le bon répertoire. Pour remédier à cela, nous avons ajouté la fonction `create_data_dir`. Cette fonction vérifie si le répertoire `data_athletes` existe, et si ce n'est pas le cas, elle le crée. Cela garantit que tous les fichiers des athlètes sont stockés dans un répertoire centralisé. En finalisant le programme, nous avons compris que nous avions besoin des fonctions de tri pour sélectionner les trois meilleurs athlètes d'une épreuve et les envoyer aux JO. Pour cela, nous avons ajouté la fonction `find_top_athletes` qui lit les performances de tous les athlètes pour une épreuve donnée, les trie par temps, et affiche les trois meilleurs. Nous avons utilisé `qsort` pour trier les performances, ce qui garantit une sélection correcte et efficace des meilleurs athlètes. La fonction `qsort` implémente un algorithme de tri rapide

(QuickSort). QuickSort est un algorithme de tri très efficace et rapide qui fonctionne en suivant le principe de "diviser pour régner". Il partitionne les éléments du tableau en deux sous-tableaux, puis trie chaque sous-tableau de manière récursive. Manipuler dynamiquement la mémoire pour les performances des athlètes a été un défi, en particulier pour garantir qu'il n'y ait pas de fuites de mémoire. Nous avons utilisé malloc, realloc, et free de manière judicieuse pour allouer et libérer la mémoire nécessaire pour stocker les performances des athlètes. Cela assure une gestion efficace de la mémoire et prévient les fuites. Ces défis ont été surmontés grâce à des améliorations successives et à l'ajout de nouvelles fonctionnalités, garantissant ainsi un programme robuste et efficace pour la gestion des performances des athlètes. Nous avons continuellement revu et optimisé le code pour nous assurer qu'il répondait aux exigences fonctionnelles tout en restant maintenable et évolutif. Cette approche méthodique nous a permis de transformer les obstacles en opportunités d'amélioration, aboutissant à un produit final de haute qualité.

L'objectif de ce programme est de permettre à l'entraîneur de l'équipe d'athlétisme de France de suivre l'évolution des performances de ses athlètes. Le programme permet de créer des fichiers pour chaque athlète, d'ajouter des performances, de lire et d'afficher les performances, de générer des statistiques, de visualiser les tendances des performances et de sélectionner les meilleurs athlètes pour les compétitions. Nous avons inclus les bibliothèques standard nécessaires pour les opérations d'entrée/sortie, la gestion de la mémoire et la manipulation des chaînes de caractères. Les structures définies permettent de stocker les informations sur une performance (Performance) et sur un athlète et ses performances (Athlete). Nous avons également choisi de faire quelques améliorations possibles comme avec la fonction `create_data_dir` qui vérifie si le répertoire `data_athletes` existe et le créer si nécessaire pour centraliser les fichiers des athlètes. La fonction `create_athlete_file` crée un fichier pour un athlète spécifique dans ce répertoire. Nous avons décidé aussi d'introduire une régression linéaire avec la fonction `athlete_progress_linear`. Pour ajouter une performance à un athlète, nous utilisons la fonction `add_performance`, qui gère également les relais en enregistrant la position. La fonction `create_initial_athletes` crée des athlètes initiaux avec des performances pour des tests et des démonstrations. Pour lire les performances d'un fichier et les stocker dans la structure `Athlete`, nous avons la fonction `read_performances`. Pour afficher l'historique des performances d'un athlète, nous utilisons `display_history`. En termes de statistiques et tendances, la fonction `athlete_statistics` calcule et affiche les meilleures, pires et moyennes performances pour un événement spécifique, tandis que `athlete_progress_linear` calcule et affiche la tendance linéaire des performances en utilisant la régression linéaire et génère un graphique avec Gnuplot. La fonction `athlete_progress_dates` compare les performances entre deux dates spécifiques pour un événement donné. Pour la validation et la saisie

utilisateur, nous avons des fonctions telles que `is_valid_date` pour vérifier si une date est au format valide YYYY-MM-DD et réaliste, `prompt_date` pour demander à l'utilisateur d'entrer une date valide, `is_valid_sport` pour vérifier si un sport est valide, `prompt_sport` pour demander à l'utilisateur d'entrer un sport valide, `is_valid_position` pour vérifier si une position de relais est valide, et `prompt_position` pour demander à l'utilisateur d'entrer une position valide. Pour sélectionner les meilleurs athlètes pour un événement, la fonction `find_top_athletes` trouve et affiche les trois meilleurs athlètes en les triant par leurs performances. La fonction principale du programme (`main`) gère l'interface utilisateur, permettant les interactions suivantes : ajouter une nouvelle performance, afficher l'historique des performances, afficher les statistiques, afficher les tendances, trouver les meilleurs athlètes pour un événement, et quitter le programme. En résumé, ce programme offre une gestion complète des performances des athlètes, facilitant ainsi le suivi et l'analyse des données par l'entraîneur.

En conclusion, ce projet de gestion des performances des athlètes de l'équipe d'athlétisme de France a abouti à la création d'un outil complet et fonctionnel permettant de suivre et d'analyser les performances des sportifs. Grâce à une organisation méthodique, des réunions régulières, et une collaboration continue via des outils comme Replit et WhatsApp, nous avons intégré des fonctionnalités essentielles telles que la création de fichiers pour chaque athlète, l'ajout de performances, la génération de statistiques, et la visualisation des tendances. En surmontant les défis techniques, notamment la validation des dates, la gestion de la mémoire et le tri des performances, nous avons développé un programme robuste et fiable. Ce projet nous a permis de renforcer nos compétences techniques tout en mettant en évidence l'importance du travail d'équipe et de la communication pour mener à bien un projet informatique complexe.