

Rapport écrit projet informatique

Nous, Synthia, Wiame et Samia, avons collaboré sur un projet en informatique centré sur les Jeux Olympiques, en mettant l'accent sur la gestion des fichiers et les algorithmes de tri. Durant un mois et demi, nous avons travaillé sur ce projet en appliquant efficacement les différents concepts informatiques à notre projet. Ce rapport détaille les étapes de notre démarche, les défis rencontrés, et les solutions mises en œuvre pour aboutir à un résultat satisfaisant.

Premièrement, pendant les vacances, nous avons choisi notre sujet lors d'une réunion sur Teams. Nous avons comparé plusieurs sujets pour déterminer celui avec lequel nous étions le plus à l'aise (étant donné que c'était notre premier projet en informatique, nous avons évité les sujets trop complexes). Mais avant de choisir, nous voulions d'abord essayer de proposer un sujet : un jeu nommé Target Number, un peu plus complexe que le jeu original. Cela n'a pas été retenu car, bien qu'intéressant, le jeu ne complétait pas les attendus du projet sur lesquels nous sommes notés. Nous avons donc fini par choisir un des sujets proposés comme prévu. Ensuite, la semaine de la rentrée, nous avons pris connaissance du sujet et de son fonctionnement en détail. Nous avons identifié les chapitres importants pour ce sujet (que nous devons maîtriser pour notre prochaine réunion de groupe).

Le mardi 7 mai nous avons donc travaillé sur l'avancement du projet. Cependant, nous avons rencontré quelques difficultés et avons dû refaire plusieurs fois la base de notre projet. Finalement, en fin de journée, nous avons réussi à lister les grandes étapes de notre code ce qui nous permettait de mieux comprendre par où commencer à coder.

Le dimanche 12 mai, nous avons consacré toute la journée à travailler à la bibliothèque universitaire. Nous avons implémenté les fonctions de base et vérifié leur bon fonctionnement. Malgré plusieurs difficultés rencontrées, nous avons réussi à compléter l'ensemble du code principal, à l'exception des fonctions de statistiques. En effet, le rôle de Synthia consistait à créer des fichiers contenant des informations sur cinq athlètes : Dahmane, Rayan, Hicham, Amine et Hari. Pour chacun de ces athlètes, elle a dû concevoir des entraînements incluant des épreuves telles que le relais, le 100 mètres, le 400 mètres, le 5000 mètres et le marathon, en indiquant également les temps réalisés pour chacune de ces épreuves. De plus, Synthia a créé une structure appelée "training" comprenant une date, un type d'épreuve et un temps. Pour sa part, Samia a créé une structure nommée "constr_time". Cette structure permet de gérer et de manipuler efficacement les données temporelles des performances des athlètes. En parallèle, elle s'est également chargée de développer une fonction appelée "constr_date". Cette fonction est conçue pour interagir directement avec l'utilisateur : elle demande les informations relatives à une date d'entraînement et procède ensuite

à la vérification de la validité de ces informations. Samia a mis en place des mécanismes robustes pour s'assurer que les données saisies respectent les formats requis et correspondent à des dates réalistes. Cela inclut la vérification de la cohérence des jours, des mois et des années, ainsi que la gestion des années bissextiles et des formats spécifiques au type d'entraînement. Enfin, Wiame a développé deux fonctions cruciales. Tout d'abord, elle a conçu la fonction "get_training", chargée de récupérer et d'afficher les entraînements d'un athlète à partir du fichier texte correspondant. Ces deux fonctions sont fondamentales pour organiser et consulter les données d'entraînement de manière efficace et fiable. De plus, nous avons réparti les tâches à accomplir pour notre prochaine réunion.

Par la suite, le lundi 13 nous avons travaillé sur la fonction "add_training". Cette fonction ouvre un fichier en mode lecture pour vérifier les entraînements existants et en mode ajout pour écrire de nouveaux entraînements. Elle demande la date de l'entraînement et vérifie s'il y a déjà un relais programmé ce jour-là, car deux relais ne sont pas autorisés le même jour. En fonction du type d'entraînement sélectionné par l'utilisateur, tel que 100m, 400m, 5000m, marathon ou relais 4x400m, elle stocke ce type dans une structure Training. Elle demande ensuite le temps de l'athlète et, si c'est un relais, la position de l'athlète dans le relais. Enfin, elle écrit les informations collectées, comme la date, le type, le temps et éventuellement la position, dans le fichier et le referme.

Le mercredi 15 mai, nous avons validé notre code actuel avec le professeur et pris connaissance des fonctions à ajouter concernant les statistiques. Nous avons tenté de les implémenter et de les comprendre mais également réparti les tâches à accomplir pour le samedi 18 mai : Synthia se chargera de coder la fonction qui trouve le meilleur et le pire athlète, Samia travaillera la fonction qui permet d'avoir la différence entre deux temps pour évaluer la progression d'un athlète enfin Wiame se concentrera sur la partie du code qui trouve le top 3 des athlètes en fonction de leur temps moyen et de l'entraînement choisi, et éventuellement laisser la possibilité à l'utilisateur de choisir quel top il voudrait faire apparaître (1, 2, 3...).

Cependant, ce jour-là aucune de nos trois fonctions fonctionnaient. Nous avons donc passé la journée entière à modifier notre code pour que deux fonctions soient fonctionnelles en fin de journée. En effet, ces fonctions étaient assez complexes, la fonction de Wiame contenait deux fonctions principales : top_athletes et partition (en plus de quicksort qui fonctionnait avec partition). La fonction top_athletes classe les athlètes du meilleur au moins bon pour une pratique donnée. Elle récupère les performances de chaque athlète à partir de fichiers portant leur nom, ouvrant chaque fichier en mode lecture. Elle calcule ensuite la moyenne des temps pour cette pratique, stocke cette moyenne et le nom de l'athlète correspondant. Après avoir récupéré les performances de tous les athlètes, elle trie ces derniers par temps moyen à l'aide de l'algorithme de tri rapide. Enfin, elle affiche les meilleurs athlètes pour cette pratique, en fonction du nombre spécifié (num_top). Si num_top est égal à 3 et qu'au moins 3 athlètes sont disponibles, elle recommande de les envoyer aux

Jeux Olympiques. De plus, la fonction partition est une partie de l'algorithme de tri rapide qui réorganise un tableau autour d'un pivot. Elle sélectionne un pivot, déplace les éléments plus petits ou égaux vers la gauche et les éléments plus grands vers la droite, puis retourne la position finale du pivot. De son côté, Samia faisait la fonction `athlete_progression` qui calcule la différence entre deux temps à deux dates données pour un type d'épreuve spécifié. Elle ouvre le fichier contenant les données d'entraînement et recherche les temps de l'athlète pour l'épreuve à chaque date. Si les deux temps sont trouvés aux bonnes dates, elle calcule la différence entre eux et affiche un message indiquant si l'athlète a progressé, régressé ou si son temps est resté le même entre les deux dates. Si aucun temps n'est trouvé pour l'une des dates, un message d'erreur est affiché. Enfin, Synthia faisait la fonction `athlete_summary` qui résume les performances d'un athlète pour un type d'entraînement spécifique, tel qu'enregistré dans un fichier. Elle ouvre un fichier en mode lecture pour vérifier les entraînements, parcourt chaque ligne pour extraire le type d'entraînement et le temps associé, et met à jour le nombre d'entraînements, le meilleur temps, le pire temps et le temps total si le type correspond. Ensuite, elle affiche un résumé des performances comprenant le meilleur temps, le pire temps et le temps moyen.

Le 22 mai, nous avons travaillé sur les fonctions en cours. Nous avons rencontré des difficultés avec les tris, initialement nous avons essayé de contourner cette étape mais nous avons finalement réalisé que c'était obligatoire. Effectivement, pour nous, la recherche du top 3 par exemple était faisable autrement, sans faire de tri. Nous avons donc repris progressivement après avoir reçu les conseils de notre professeur et réussi à enfin implémenter les tris. De plus, nous avons également rencontré des problèmes avec les relais. Nous n'étions pas tous d'accord sur la manière de rédiger le relais, ce qui a conduit à des incohérences dans notre code. Pour résoudre ces problèmes, nous avons dû revoir ensemble la rédaction du relais depuis le début. En parallèle, nous avons également commencé la rédaction de notre rapport de soutenance de projet en parallèle. De plus, nous avons revisité notre code à plusieurs reprises et nous avons remarqué un problème : lorsque nous utilisions des chiffres autres que 1 ou 2 (1 pour oui ou 2 pour non), notre code indiquait une erreur, cependant si nous introduisions des lettres ou des caractères spéciaux, notre code plantait. Pour résoudre ce problème, nous avons ajouté du code pour gérer ces cas spécifiques, notamment la fonction `get_int_input` qui vérifie que l'entrée est bien un chiffre (ou nombre) entier. Cela nous a permis de supprimer ce genre d'erreurs et simplifier la vérification : il nous restait simplement à vérifier que c'était 1 ou 2, ou bien un chiffre entre 1 et 5 par exemple.

En conclusion, malgré les nombreuses difficultés rencontrées tout au long du projet, notamment avec les tris et la gestion des relais, nous avons réussi à mener à bien notre projet en informatique. De plus, nous avons pris le temps de revoir notre code à plusieurs reprises, ce qui nous a permis de détecter et de résoudre des problèmes, comme la gestion des entrées utilisateur. En fin de compte, cette

expérience nous a enrichies sur le plan technique et nous a permis de mieux comprendre la rigueur nécessaire pour mener à bien un projet informatique.