

Rapport de projet : Résultats de football en bash

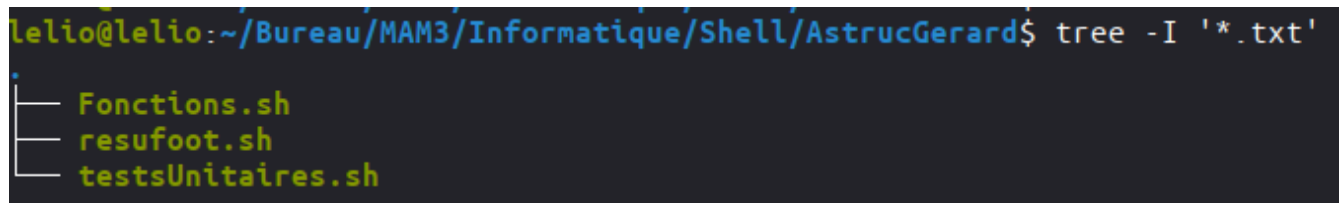
Présentation du projet

Dans le cadre de cette UE EIMA622, nous avons eu la possibilité de réaliser un programme permettant de récupérer des informations sur des matchs de football. Le but de ce rapport est de présenter la réalisation de fonctions ainsi que d'un script bash permettant de calculer les classements des équipes au premier tour du mondial de football, les points qui leur ont été attribués ainsi que d'autres informations.

Architecture du projet

L'architecture de notre projet est la suivante :

```
lelio@lelio:~/Bureau/MAM3/Informatique/Shell/AstrucGerard$ tree -I '*.txt'
```



```
├── Fonctions.sh
├── resufont.sh
└── testsUnitaires.sh
```

Les fichiers `.txt` ne sont pas affichés car ils sont bien trop nombreux. Le fichier `Fonctions.sh` est le fichier contenant les 5 premières fonctions du projet. Elles sont toutes regroupées dans un seul et même fichier car, lors de la réalisation du projet, nous avons rencontré des difficultés lorsque nous voulions appeler chaque fonction pour effectuer les tests. Le script `resufont.sh` correspond à la question 6 du projet. Enfin le fichier `testsUnitaires.sh` est un fichier dans lequel nous avons réalisé les tests.

Fonctionnement général du projet

Comme indiqué dans l'énoncé, l'objectif de ce projet est de réaliser un script bash permettant de calculer les classements des équipes au premier tour de la coupe du monde de football. Pour ce faire, nous disposons de 8 fichiers `groupe{A-G}.txt` que nous avons écrit, correspondant aux groupes de la coupe du monde de 2022. Nous avons également écrit les fichiers des matchs qui se sont joués contenant ; la date et l'heure, l'équipe 1, l'équipe 2, le score de l'équipe 1, et le score de l'équipe 2. Ces fichiers sont au nombre de 64 et sont nommés `match{1..64}.txt`. Nous nous sommes rendus compte à la fin du projet qu'en réalité, seuls les matchs 1 à 48 étaient utiles. Cependant, nous avons préféré les garder au cas où ils nous auraient servis.

La fonction AffiScore

La fonction `AffiScore` prend deux arguments en paramètres, le nombre de buts de l'équipe 1 et le nombre de buts de l'équipe 2. De là, on vérifie que les paramètres entrés par l'utilisateur sont bien des entiers, puis on calcule la différence de buts entre les deux équipes et on affiche, le nombre de points attribués à l'équipe 1, la différence de score entre les deux équipes ainsi que le score de l'équipe 1.

La fonction `ResultatMatch`

Comme la fonction précédente ainsi que pour toutes les suivantes, la fonction `ResultatMatch` vérifie que le nombre d'arguments est correct ainsi que le type. La fonction vérifie que le fichier de match existe et extrait les deux équipes à partir de ce fichier. Enfin, suivant l'équipe passée en paramètre, on utilise la fonction `AffiScore` pour calculer le résultat d'un match.

La fonction `ResultatsEquipe`

Le principe de la fonction est simple : on parcourt l'entière des fichiers de matchs et pour chaque fichier, si l'équipe passée en paramètre a joué le match, on appelle la fonction `AffiScore` avec son score et le score de l'équipe adverse, sinon, on passe au match suivant. Nous avons eu quelques soucis lors de la rédaction de la fonction notamment pour différencier une équipe considérée comme équipe 1 ou comme équipe 2.

La fonction `AssemblerResultats`

Pour l'équipe passée en argument, on appelle la fonction `ResultatsEquipe` qui affiche les résultats de chaque match de l'équipe. Ensuite, on peut calculer le nombre magique de l'équipe pour le tournoi en faisant la somme de chaque "colonne". Par exemple pour l'équipe de France :

```
$ ResultatEquipe France
3 1 2
3 1 1
3 3 3
```

ce qui donne 9 5 6 d'où le nombre magique 509505506. Dans notre code, les résultats sont différents puisque les matchs choisis sont les résultats de la coupe du monde de 2022.

La fonction `ClasserGroupe`

C'est sûrement la fonction qui nous aura fait le plus réfléchir. Nous avons commencé par imbriquer plusieurs conditions pour vérifier si le nombre de points était égal, puis les différences et ainsi de suite. Cependant, il est intéressant d'utiliser ici le nombre magique, puisqu'il permet de classer les différentes équipes du groupe en utilisant seulement une commande de `sort`.

Le script `resufot.sh`

Ce dernier script quant à lui, était plutôt simple au vu de la fonction `ClasserGroupe`. L'utilisation de `case` permet d'implémenter les différentes options que l'utilisateur peut choisir sans grande difficulté. Pour le cas `-h`, nous avons choisi de donner le même résultat que le cas `-g`, en supprimant le nombre magique. Cela est possible grâce à une expression `sed` qui supprime les séquences de 9 chiffres consécutifs.