

Mini-projet de travaux pratiques

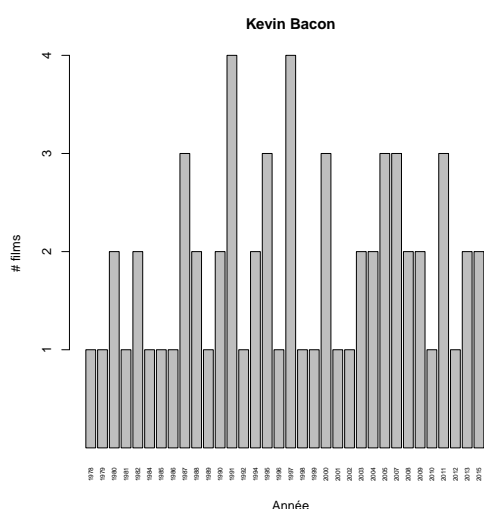
Le projet est à effectuer en binôme (ou, *exceptionnellement*, en monôme) en utilisant une combinaison de Bash et de Python.

L'ensemble du code parfaitement commenté ainsi que le dossier de programmation sont à déposer sur Madoc le **29 novembre 2019 à 23h55** au plus tard, sous la forme d'une archive dont le nom sera composé des noms de chacun des étudiants du binôme — ex. `tartempion-dupont.tar.gz`). La décompression de cette archive devra créer un répertoire `tartempion-dupont/` contenant les fichiers source. On re-précisera en commentaires dans chaque fichier du programme les noms et prénoms de chacun-e des étudiant-e-s.

On dispose d'une base de données au format **JSON** de 164 318 films et l'on souhaite développer un logiciel permettant de créer un diagramme indiquant, année par année, le nombre de films dans lequel un acteur a joué, ou un réalisateur a dirigé (voir exemple ci-contre pour l'acteur Kevin BACON).

On a choisi une solution faisant intervenir trois programmes :

- Un programme Python pour analyser la base de données initiale et créer deux bases de données annexes : une base recensant les films dans lesquels a joué un acteur et une base pour les réalisateurs. Le programme créera aussi le fichier au format **CSV** recensant le nombre de films par année pour un acteur ou un réalisateur choisi par l'utilisateur ;
- Un programme R prenant en entrée un fichier CSV et créant le diagramme récapitulatif au format Acrobat PDF. Ce programme vous est directement fourni ;
- Un programme BASH servant de point d'entrée à l'utilisateur et pilotant les deux programmes précédents.



Les programmes produits pourront être testés avec des bases de données de tailles variables disponibles sur madoc.

Le programme Python (`moviedb.py`)

Le programme Python doit accepter au moins trois options :

- **-c ou --create-from-db.** Le programme prend en entrée le fichier JSON passé en paramètre à cette option et crée à partir de la base qu'il contient une base de données d'acteurs `actors.json` et une base de données de réalisateurs `directors.json` répertoriant pour chaque personne une liste de couples (titre, année) de tous les films dans lesquels elle a joué (ou qu'elle a dirigé). **Attention** : pour certains films, la base de données ne recense pas d'année de sortie ou pas de réalisateur. Ces films doivent être ignorés ;
- **-a ou --actor.** À partir de la base `actors.json`, le programme crée une liste au format CSV recensant le nombre de films tournés pour chaque année où l'acteur dont le nom est passé en paramètre de cette option a été actif. Exemple de sortie pour Kevin BACON :

```
year,nmovies
1978,1
1979,1
[...]
```

La liste est affichée sur la sortie standard ;

-d ou --director. Même chose que l’option précédente à partir de la base `directors.json` pour les réalisateurs.

On utilisera le paquetage `argparse` pour gérer les options du programme et le paquetage `json` pour la lecture et l’écriture des fichiers au format JSON.

Points importants :

- On affichera un message d’erreur et l’on retournera le code d’erreur 3 si l’utilisateur utilise à la fois l’option `-a` et l’option `-d` (ou leurs versions longues) ;
- On affichera un message d’erreur et l’on retournera le code d’erreur 2 si l’option `-c` (ou sa version longue) est choisie et que le fichier JSON associé ne peut être chargé ;
- Les bases de données fournies sont constituées d’un enregistrement JSON par ligne. Leur lecture avec le paquetage JSON doit donc se faire ligne par ligne ;
- Si le nom d’acteur ou de réalisateur passé en paramètre des options `-a` et `-d` (ou leurs versions longues) ne se trouve pas dans la base de données, le programme Python retournera la chaîne « None » plutôt qu’une liste au format CSV.

Le programme BASH (`moviedb.sh`)

Le programme BASH accepte deux options :

-a ou --actor. Cette option est suivie d’un nom d’acteur ;

-d ou --director. Cette option est suivie d’un nom de réalisateur.

Le programme doit être appelé avec l’une de ces deux options. À défaut, on devra afficher un message d’aide rappelant les options possibles et l’on retournera le code d’erreur 1.

Le programme BASH doit s’assurer de la présence des fichiers `actors.json` et `directors.json` à chaque lancement. S’ils ne sont pas présents, c’est l’indication que le programme est appelé pour la première fois. Dans ce cas, on doit appeler le programme `moviedb.py` avec l’option `-c` et le nom de la base de données à utiliser (fixée dans le script BASH) avant toute autre action. Si cette initialisation ne se passe pas bien, on affichera un message et l’on retournera le code d’erreur 3.

De même, le script BASH doit vérifier la présence des répertoires `cache/actor` et `cache/director` en les créant si besoin.

Lorsque le script est appelé avec l’option `-a/--actor` (resp. `-d/--director`), il vérifie si le fichier CSV portant le nom de l’acteur (resp. du réalisateur) ne se trouve pas déjà dans le répertoire `cache/actor` (resp. `cache/director`). Si c’est le cas, il appelle directement le script R `stats_name_years.R` sur ce fichier. Sinon, il appelle au préalable le programme `moviedb.py` pour générer le fichier CSV et il le sauve dans le répertoire `cache/actor` (resp. `cache/director`), de façon à ce qu’un appel ultérieur pour la même personne puisse être traité plus rapidement.

Si le fichier CSV contient « None », on affiche un message indiquant que la personne recherchée n’est pas dans la base. L’appel du script R doit être suivi d’un message indiquant le nom du fichier PDF produit.