

---

## **SAE 1.01 - Implémentation d'un besoin client**

**29 octobre 2021**

---

Réalisé par :

**FERNANDEZ Mickaël**

**HURÉ Tilian**

---

## Sommaire :

<b>Introduction :</b>	<b>3</b>
<b>Explication des mécanismes :</b>	<b>5</b>
<b>Fonctionnalités de l'application :</b>	<b>8</b>
Fonctionnalité n°1 :	8
Trace d'exécution :	9
Fonctionnalité n°2 :	9
Trace d'exécution :	10
Fonctionnalité n°3 :	11
Trace d'exécution :	11
Fonctionnalité n°4 :	11
Trace d'exécution :	12
<b>Conclusion :</b>	<b>12</b>

## Introduction :

L'objectif de cette SAE est, par groupe de deux, de concevoir une application répondant aux besoins du client. Le projet consiste en un programme Java qui permet d'analyser une liste d'étudiants (sous format .csv) et de tirer au sort un étudiant parmi cette liste en question. On appellera un "tour", lorsque tous les étudiants de la promotion ont été tirés au sort. Néanmoins, des contraintes nous sont imposées :

- Un étudiant ne peut pas être tiré au sort deux fois d'affilée.
- Un étudiant déjà tiré au sort ne peut être re-choisi que lorsque tous les autres élèves de la promotion l'ont également été (soit lorsque un tour a été effectué).

Le programme se divise en fait en deux classes :

- La classe **ListeEtudiants** qui transforme le fichier **csv** comportant les étudiants de la promotion, en un tableau de chaînes de caractère de la forme `{{"nom1", "prénom1"}, {"nom2", "prénom2"}, ...}`.
- Et la classe **Menu** qui affiche un menu et qui traite les choix des options de l'utilisateur.

Lors de son exécution, le programme affiche donc un menu de la forme suivante :

```
Nombre de lignes : 5
Il y a : 5 etudiants

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
```

L'utilisateur effectue donc ses choix en renseignant le chiffre correspondant à l'option du menu. Il est obligatoire de renseigner un chiffre entre 1 et 4 (inclus). Dans le cas où une valeur non-numérique est renseignée, le programme s'arrête.

```
Nombre de lignes : 199
Il y a : 199 etudiants

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
0

Erreur ! Donnez une valeur comprise entre 1 et 4 : 5

Erreur ! Donnez une valeur comprise entre 1 et 4 : aucun

Erreur: null
Ligne : 939
```

## Explication des mécanismes :

La fonction principale de notre projet consiste en tirer une suite d'étudiants au hasard, parmi toute une promotion. Plusieurs mécanismes ont donc été développés afin de respecter les contraintes imposées.

Tout d'abord, lorsque l'utilisateur choisi dans le menu affiché, l'option n°1 pour justement débiter la suite du tirage au sort des élèves, il devra confirmer s'il désire ou non tirer au sort un premier étudiant. Il devra répéter cette action à chaque fois, tant qu'il désirera tirer au sort d'autres étudiants.

```
Nombre de lignes : 199
Il y a : 199 etudiants

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
1
Tirer au hasard un(e) étudiant(e) ? [Oui/non]
oui
L'étudiant tiré au hasard est : Saintmichel Edmond
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
o
L'étudiant tiré au hasard est : Tar Guy
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
Non

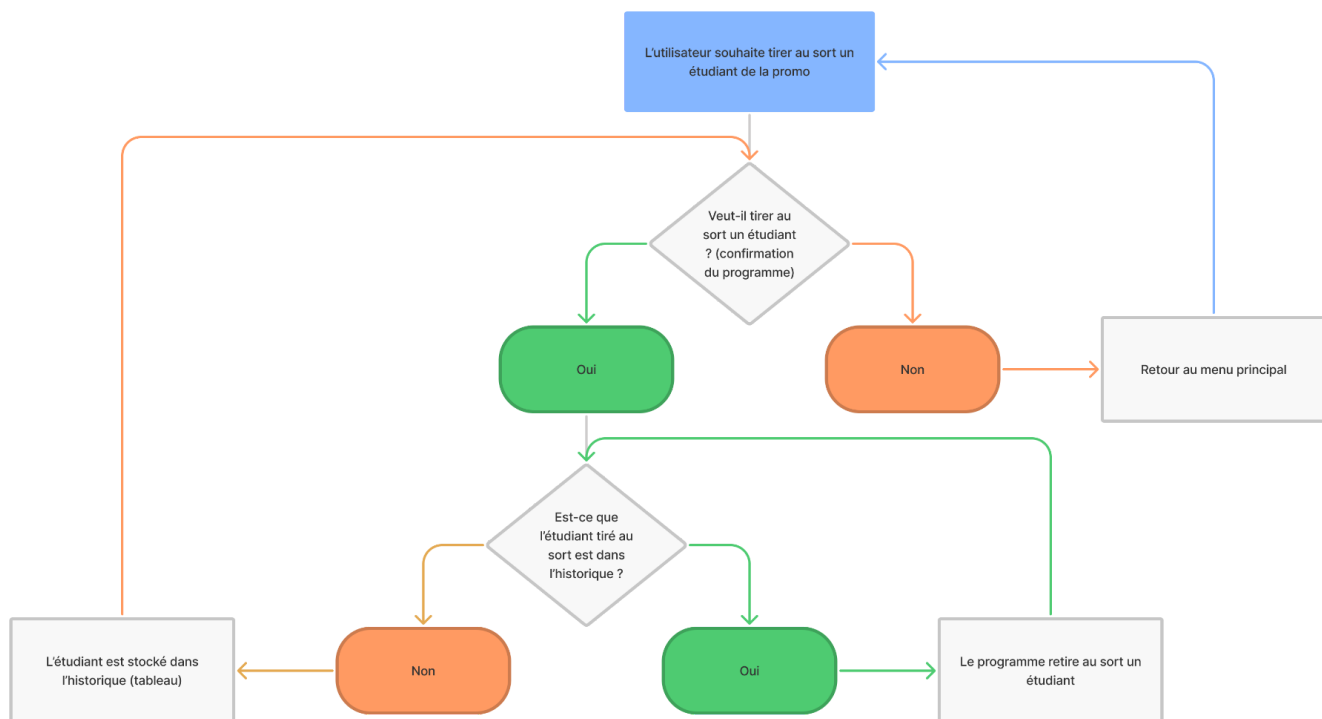
/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
```

Pour respecter les conditions et faire en sorte que les élèves ayant déjà été tirés au sort, ne repassent pas tant que tous les élèves n'ont pas été choisis, nous avons mis en place un **historique des étudiants**. De la même forme que le tableau contenant les étudiants (`{{ "nom1", "prénom1" }, ... }`), ce tableau permet de stocker tous les étudiants ayant été tirés au sort.

Ainsi, à chaque fois que l'on tire au sort un nouvel étudiant, on regarde s'il n'est pas déjà présent dans cette historique.

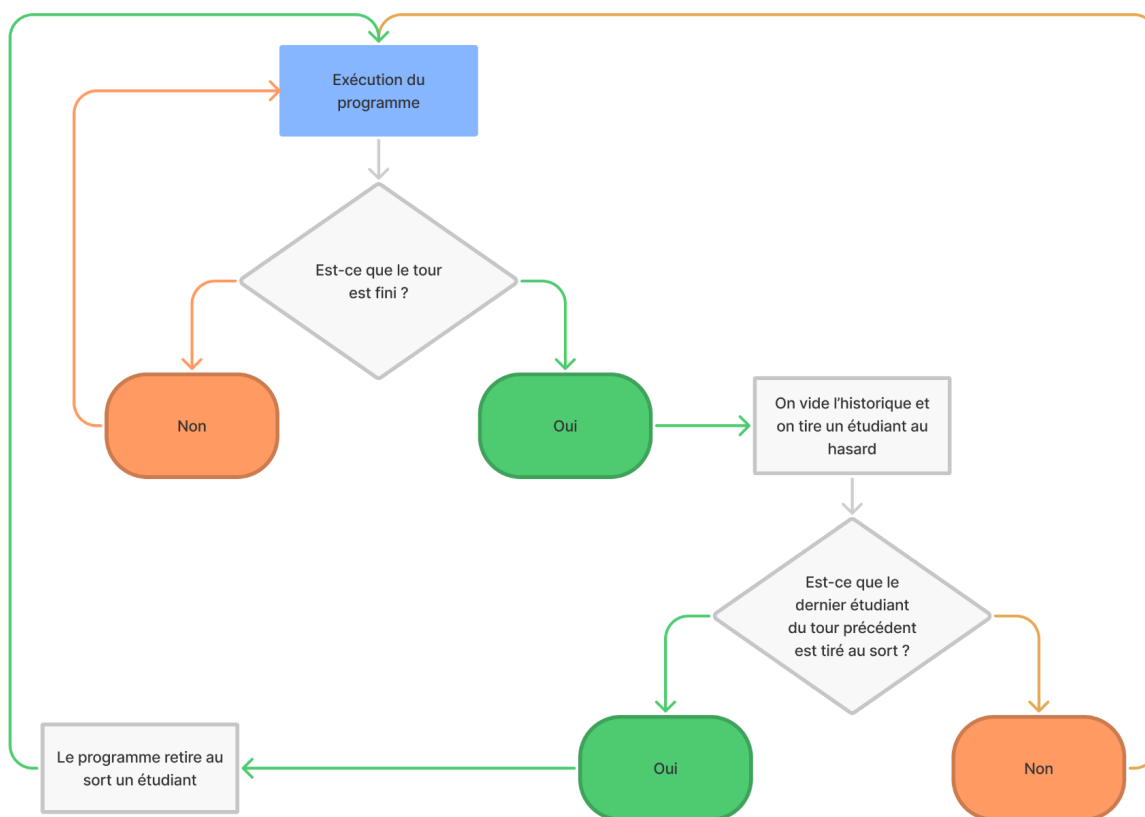
- Si c'est le cas, on tire un autre étudiant au hasard puis on répète la démarche.
- Si ce n'est pas le cas, on ajoute cet étudiant à l'historique.



Le programme affiche ensuite l'étudiant choisi, puis redemande à l'utilisateur s'il souhaite en tirer un autre. Si l'utilisateur accepte, la procédure est répétée.

Lorsqu'un tour s'achève et que tous les étudiants de la promotion sont passés, l'historique est réinitialisé soit vidé de son contenu, puis les étudiants sont à nouveau tirés au sort un par un.

Il faut cependant faire attention à ne pas tirer pour un nouveau tour, le dernier étudiant du tour précédent, puisqu'il ne doit pas être possible de choisir deux fois d'affilée la même personne. Ainsi, lorsqu'un tour se termine, on enregistre le **nom** et le **prénom** du dernier étudiant tiré, de manière à ne pas re-choisir ce dernier lors du premier tirage du prochain tour.



Tout ce protocole se répète jusqu'à ce que l'utilisateur décide d'arrêter de tirer au sort des étudiants, dans quel cas, il sera redirigé vers le menu principal. Lors de la prochaine série de tirage, l'historique des utilisateurs sera à nouveau réinitialisé.

---

## Fonctionnalités de l'application :

L'application possède donc quatre fonctionnalités en rapport au besoin du client :

- Tirer un élève au hasard dans la promo (et demander s'il souhaite ou non continuer de tirer au sort des étudiants)
- Vérifier si un étudiant fait parti de la promo (en renseignant le nom et le prénom de l'étudiant)
- Afficher l'ensemble des élèves de la promo
- Quitter le programme

### Fonctionnalité n°1 :

La première et principale fonctionnalité s'exécute lorsque l'utilisateur renseigne la valeur "1" du menu. Elle permet d'effectuer une série de tirages au sort des étudiants faisant partie de la promotion. Le programme demande alors une confirmation à l'utilisateur sur le tirage effectué dans la liste d'étudiants.

Plusieurs cas sont traités :

- Les choix "oui" et "non" (également "o" et "n")
- Les caractères majuscules et minuscules
- La touche "entrée", considérée comme un "oui" de manière plus optimisée



## Trace d'exécution :

```
Choix :
1
Tirer au hasard un(e) étudiant(e) ? [Oui/non]
o
L'étudiant tiré au hasard est : Aboisdormant Abel
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
0
L'étudiant tiré au hasard est : Afritt Barack
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
OUI
L'étudiant tiré au hasard est : Bu Amede
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
oui
L'étudiant tiré au hasard est : Balmaske Alonzo
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]

L'étudiant tiré au hasard est : Cape Andy

On recommence un tour !
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]

L'étudiant tiré au hasard est : Bu Amede
Tirer au hasard un(e) autre étudiant(e) ? [Oui/non]
non

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
```

On remarque sur les tirages, qu'une fois que tous les étudiants sont tirés au sort, aucun tirage ne se répète deux fois d'affilée, un étudiant ne peut pas passer plusieurs fois dans le premier tour, mais également que le programme annonce lorsqu'un tour est terminé.

## Fonctionnalité n°2 :

Une fonctionnalité certes facultative, mais tout aussi très intéressante s'exécute lorsque l'utilisateur renseigne la valeur "2". Elle permet de vérifier si un étudiant fait partie de la promotion. Le programme demande d'abord à l'utilisateur de renseigner le nom puis le prénom de l'étudiant recherché.

## Trace d'exécution :

```
Choix :
2
Veuillez renseigner le nom de l'étudiant :
Afritt
Veuillez renseigner le prénom de l'étudiant :
Barack

Afritt Barack est bien dans la promo.

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
2
Veuillez renseigner le nom de l'étudiant :
afritt
Veuillez renseigner le prénom de l'étudiant :
barack

afritt barack ne fait pas parti de la promo.
(Ou peut être avez-vous mal orthographié son nom ou son prénom ?)
Syntaxe : <Nom> et <Prenom>

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/
```

Deux cas d'exécution sont représentés ci-dessus.

Le premier cas d'exécution est correct puisque les valeurs saisies par l'utilisateur (**nom**, **prénom**) et celles recherchées de la promo sont strictement identiques.

On remarque toutefois qu'une certaine casse est à respecter puisqu'en effet, en observant le deuxième cas, l'utilisateur a oublié de renseigner les caractères majuscules ce qui fait qu'à la comparaison, comme les valeurs renseignées doivent strictement être égales, l'étudiant n'est pas trouvé dans la promo.

### Fonctionnalité n°3 :

La seconde fonctionnalité s'exécute lorsque l'utilisateur renseigne la valeur "3". Elle permet d'afficher les données de tous les étudiants de la promotion. Le programme appelle la fonction principale (main) de la classe **ListeEtudiants** afin d'afficher à tour de rôle, les étudiants de la promotion numérotés, ainsi que le nombre de total d'étudiants dans la promo.

### Trace d'exécution :

```
Choix :
3
Nombre de lignes : 5
Etudiant #1 : Afritt Barack
Etudiant #2 : Aboisdormant Abel
Etudiant #3 : Balmaske Alonzo
Etudiant #4 : Bu Amede
Etudiant #5 : Cape Andy
Il y a : 5 etudiants

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
```

Une fois l'affichage terminé, le programme renvoie automatiquement l'utilisateur au menu principal.

### Fonctionnalité n°4 :

La dernière fonctionnalité s'exécute lorsque l'utilisateur renseigne la valeur "4". Elle permet d'arrêter le programme. Un message informe l'utilisateur que le programme s'est arrêté et aucune valeur ne peut être renseignée tant que l'utilisateur n'exécute pas une nouvelle fois le programme.

## Trace d'exécution :

```

/*****/
Selectionner :
1 - Tirer des élèves de la promo au hasard
2 - Vérifier si un(e) étudiant(e) est dans la promo
3 - Afficher les élèves de la promo
4 - Quitter
/*****/

Choix :
4

Arrêt du programme,
merci de l'avoir utilisé.
```

Le programme s'arrête.

## Conclusion :

Cette SAÉ nous a permis, sur un domaine plaisant, de pouvoir coopérer et de nous coordonner en binôme afin de réaliser le projet demandé. Le travail de groupe était ici principalement mis en avant puisqu'il s'agissait de programmer et comprendre ensemble un même algorithme, plutôt que chacun essaie individuellement d'en faire une partie.

Cette SAÉ nous aura également permis de percevoir de nouvelles méthodes et techniques en Java, afin de pouvoir optimiser les programmes et de traiter des problématiques plus larges concernant nos futurs projets.