

ANALYSE DU PROJET NFA031

Zain ZAFAR (N° 100224623)

zainzafarsial@gmail.com

1. Méthode Main

Pile	Tas									
	0	1	2	3	4	5	6	7	8	9 (dic.length - 1)
dic →	A	A	A	B	A	C	A	A	S	...
	0	1	0	1	2	4	5	0	1	2 3 4 5
										...
										...
										...
tirage →
	0	1	2	3	4	5	6	7	8	9
input →	Saisie au clavier									
	0	1	...							(input.length-1)
rep	Saisie au clavier									

2. Déclarations(Méthode main):

```
char[][] dic ;
```

```
char[] tirage ;
```

```
char[] input ;
```

```
char rep;
```

3. Méthode de calcul:

- Méthode **afficherTab** prend un tableau **unidimensionnel** comme paramètres et l'affiche.
- La méthode **upperCase** prend le mot proposé(**input**) par l'utilisateur et s'il est minuscule, il le convertit en majuscule.
- La méthode **donnerTirage** nous donne un **tableau de char** contenant **dix** lettres aléatoires.
- La méthode **getDictionnaire** lit le fichier de dictionnaire '**dico_nfa031**' et le convertit en un tableau de char à **deux dimensions**.
- Tout d'abord, nous utilisons la méthode de **compareInputDic** qui prend comme paramètres le mot et le dictionnaire proposés. Cette méthode parcourt le tableau à deux dimensions du dictionnaire(**dic**) jusqu'à trouver un mot de même longueur que le mot proposé(**input**). Ensuite, il compare les deux mots lettre par lettre. À chaque fois que les lettres sont égales, le **counter** de variables est augmenté. Lorsque la boucle pour la comparaison se termine, il voit si le **counter** est égal à la longueur des deux lettres. S'il est égal, alors nous avons trouvé le mot et le résultat est renvoyé sous la forme d'un booléen (**vrai**). Si non, le variable **counter** est réinitialisé et le processus continue. S'il ne trouve pas le mot, il envoie **faux**.
- Ensuite nous avons la méthode **compareAvecTirage**, elle compare le mot proposé(**input**) avec le **tirage** donné en utilisant deux boucles **for** et **while**. Il prend la première lettre du mot proposé(**input**) et le compare à chaque lettre de **tirage** jusqu'à ce qu'il trouve la lettre exacte ou la boucle terminée. S'il trouve la

même lettre, il enregistre la position de la lettre tirage dans un tableau de char **temp** et le remplace par "~". Sinon, il place **X** à sa place. À la fin, il envoie le tableau de char **temp** comme résultat.

- Nous utilisons la petite méthode **afficherComparaison** juste pour afficher les trois tableaux **tirage** avec sa **numérotation**, le mot proposé(**input**) et le résultat de la méthode **compareAvecTirage**.
- La méthode **compareInputTirage** parcourt le résultat de la méthode **compareAvecTirage** et recherche si elle contient des '**X**' si elle en trouve envoie le résultat comme **faux**, sinon le résultat est **vrai**.
- La méthode **tousLesMotsPossible** utilise le même principe que la méthode **comparaison** mais prend tirage et dictionnaire comme paramètres, elle trouve tous les mots possibles avec le tirage indiqué et les stocke dans un **tableau à deux dimensions** et l'envoie comme résultat.
- la méthode **motsPlusLong** prend le résultat de la méthode **tousLesMotsPossible** et calcule la **longueur** du mot le plus long. Ensuite, il affiche tous les mots de cette **longueur**.

4. Méthodes

