Fiche d'investigation de fonctionnalité

Fonctionnalité : Search(), méthode de recherche dans une liste de recettes
Problématique : Comparer deux options pour implémenter une méthode de recherche : une utilisant une boucle while et une utilisant la méthode filter() de JavaScript.

Option 1 : Boucle while	
Avantages Offre un contrôle complet sur le flux du programme. Peut être plus performante pour de très grands ensembles de données car elle arrête l'exécution dès qu'une condition est satisfaite	**Inconvénients** Le code est plus long et plus complexe. Le contrôle manuel de l'itérateur peut mener à des erreurs.
Cette fonction utilise une boucle while pour parcourir toutes les recettes. Pour chaque recette, elle fonction vérifie si chaque mot de la chaîne de recherche est présent dans le nom de la recette, la liste des ingrédients ou la description. Si c'est le cas, elle ajoute la recette à la liste des recettes filtrées. Cette fonction utilise une approche manuelle pour itérer sur les éléments et vérifier les conditions.	

Option 2 : Méthode filter()	
Avantages Code plus concis et plus lisible. La méthode filter est spécialement conçue pour ce genre de tâches, donc moins de risque d'erreurs.	**Inconvénients** Moins de contrôle sur le flux du programme. Peut être moins performante pour de grands ensembles de données car elle examine tous les éléments du tableau.
Cette fonction utilise la méthode filter de JavaScript, qui est une fonction intégrée pour les tableaux. Filter crée un nouveau tableau avec tous les éléments qui passent le test implémenté par la fonction fournie. La fonction fournie à filter a la même logique que la boucle while, mais l'itération est gérée automatiquement par la fonction filter. Cela rend cette fonction plus concise et plus facile à lire.	

Solution retenue :

Performances : En termes de performances, la différence entre les deux approches peut dépendre de la taille et de la nature des données. Pour des tableaux ayant un très grand nombre de recettes, très grand nombre d'ingrédients par recette, ou une recherche avec beaucoup de mots, une boucle manuelle while peut être plus rapide en raison de l'overhead supplémentaire de la méthode filter () :
- copie du tableau sur lequel elle est appelée,
- nécessité d'une fonction callback exécutée pour chaque élément du tableau.

Style : En termes de style, la deuxième fonction est plus moderne et utilise des fonctions intégrées de JavaScript qui sont conçues pour travailler avec des tableaux. L'utilisation de filter() rend le code plus déclaratif et facile à comprendre

complexité algorithmique notation Big O :

La complexité est la même pour les deux fonctions, 3 opérations en temps linéaire :
O(n) n est le nombre de recettes
O(m) m est le nombre moyen d'ingrédients par recette
O(p) p est le nombre de mots dans la recherche.
L'opération globale a une complexité de O(n * m * p)

Dans le tableau comparatif en annexe 1, on voit que la différence de performance entre ces deux fonctions est négligeable. L'overhead de la méthode filter() reste très faible et n'a pas d'impact significatif. La clarté et la lisibilité du code sont souvent plus importantes que des différences minimes de performance. C'est pourquoi la solution retenue est l'option 2 (méthode filter().

Annexe 1

Tableaux comparatifs

Comparatifs des fonctions sur JSBench :
https://jsbench.me/81li9zh715/1

Tests de performance entre la boucle native while et la méthode filter de l'objet Array

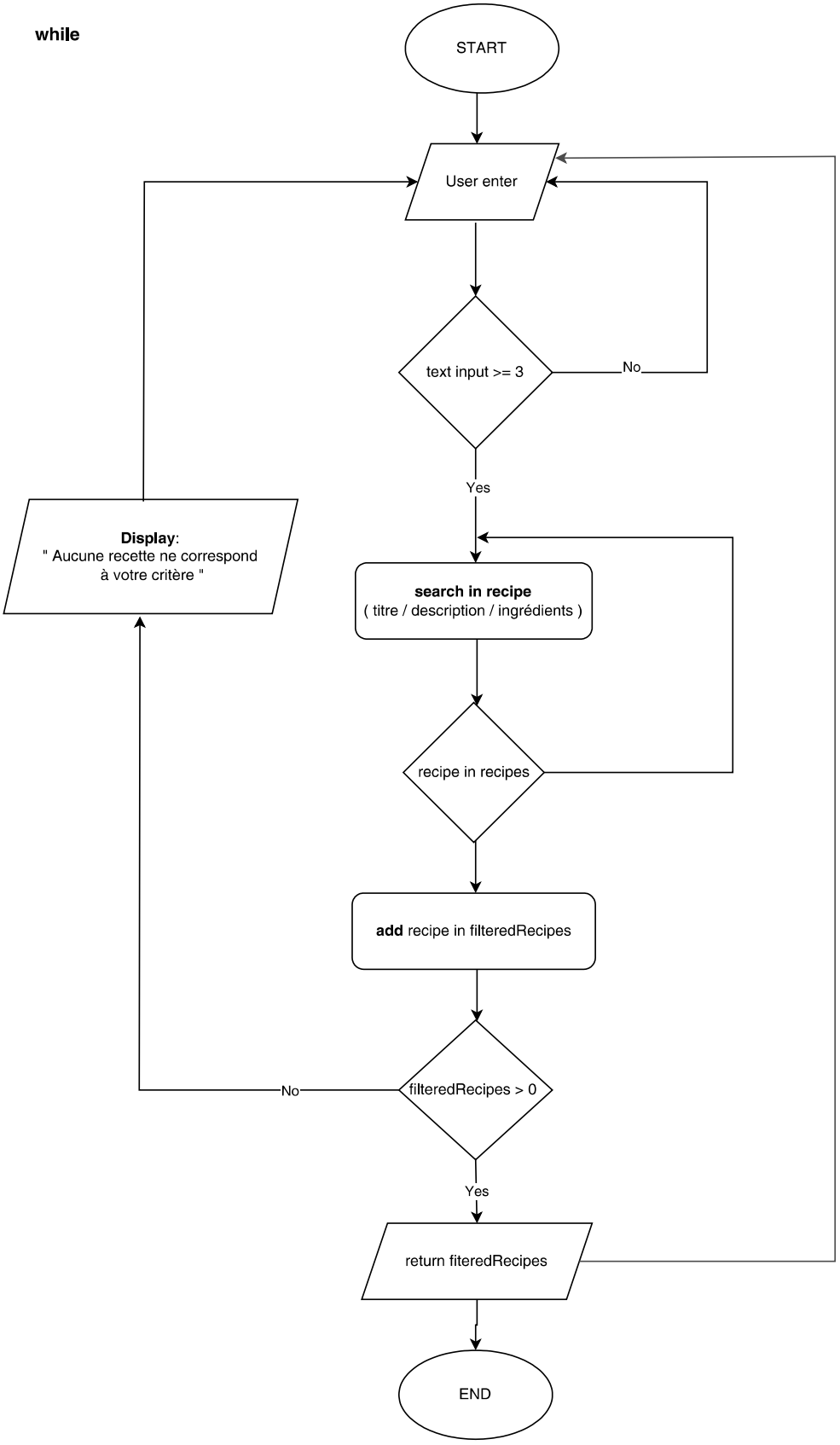
Tableau recipes.js	Boucle while	Méthode filter()
	4,3 k ops/s ± 0.53%	4,2 k ops/s ± 0.84%

Tableau fictif : Nb de recettes	Boucle while	Méthode filter()
10	94 k ops/s ± 2.69%	97 k ops/s ± 1.02%
50	20 k ops/s ± 1.52%	20 k ops/s ± 1.45%
100	9,9 k ops/s ± 1.37%	9,9 k ops/s ± 1.68%
500	2,1 k ops/s ± 0.57%	2 k ops/s ± 0.59%
1000	998 ops/s ± 0.55%	1 k ops/s ± 1.06%

Annex2

Algorigrammes

Algorigramme fonction 1 : Boucle while



Algorigramme fonction 2 : Méthode filter()