Fiche d'investigation de fonctionnalité

Fonctionnalité : Recherche de recettes Fonctionnalité #1

Problématique:

Filtrage des recettes dans l'interface utilisateur, l'utilisateur doit pouvoir accéder rapidement à la recette correspondant à sa recherche

Option 1 : Programmation fonctionnelle (annexe 1)

Utilisation des méthodes de l'objet Array (forEach, Filter...)

Emploi ici de la méthode « filter » qui filtre les recettes suivant la saisie effectuée et les correspondances trouvées dans le nom ou la description ou les ingrédients de la recette.

La recette trouvée est ajoutée à un tableau qui servira à l'affichage des recettes.

De ce tableau, les différentes listes sont mises à jour.

Avantages

- code plus robuste et plus stable

- code plus court
- version plus rapide

Inconvénients

- code moins lisible

Saisie de 3 caractères minimum dans le champ de recherche principal

Option 2 : Programmation native (annexe 1)

Utilisation des boucles (while, for ...).

Ici utilisation de « for » qui itère sur le tableau des recettes et cherche s'il existe une correspondance entre la saisie, et le nom ou la description ou un des ingrédients de la recette. Si oui, la recette en question est ajoutée à un nouveau tableau qui servira à l'affichage des recettes trouvées.

De ce tableau, les différentes listes sont mises à jour également

Avantages

- code plus lisible, plus facile à comprendre

Inconvénients

- code moins stable, plus long
- version plus lente

Saisie de 3 caractères minimum dans le champ de recherche principal

Solution retenue

Tests effectués sur 4000 recettes :

Tests avec console.time:

algorithmPerformance()

- programmation fonctionnelle avec filter : 4139 ms (en moyenne)
- programmation native avec for : 4356 ms (en moyenne)

Notre choix se porte donc sur l'option 1, la programmation fonctionnelle avec Filter







