

Fiche d'investigation de fonctionnalité

Fonctionnalité : Champ de recherche principal	Fonctionnalité #1
Problématique : Afin de pouvoir donner la meilleure expérience de recherche aux utilisateurs, nous cherchons à rendre le plus fluide possible le moteur de recherche de la barre principale dans le but d'afficher les recettes le plus rapidement possible.	

Option 1 : <code>loop `` for...in ``</code>	
Avantages : <ul style="list-style-type: none">- Supportée par tous les navigateurs	Inconvénients : <ul style="list-style-type: none">- Boucle la moins performante sur Chrome et Firefox- Plus difficile à lire- Nécessite plus de ligne de code- N'utilise pas les instructions « <code>break ;</code> » et « <code>continue ;</code> »
Nombre de boucles nécessaires : 4 Nombre de boucles optionnelles : 0	

Option 2 : <code>loop `` for...of ``</code>	
Avantages : <ul style="list-style-type: none">- Plus performante sur Chrome et Firefox- Plus facile à lire- Nécessite moins de ligne de code- Utilise les instructions « <code>break ;</code> » et « <code>continue ;</code> »	Inconvénients : <ul style="list-style-type: none">- N'est pas supportée par Internet Explorer
Nombre de boucles nécessaires : 2 Nombre de boucles optionnelles : 1	

Solution retenue : <p>Nous avons donc retenu l'approche « <code>loop `` for...of ``</code> » pour ces différentes raisons :</p> <ul style="list-style-type: none">- Parmi les boucles natives et fonctionnelles testées, la boucle « <code>for...of</code> » est apparue comme l'une des meilleures solutions selon le score obtenu au Javascript Benchmark.- L'utilisation de la méthode « <code>indexOf()</code> » montre de meilleure performance que les méthodes « <code>include()</code> » et « <code>match()</code> » ;- Mettre la condition concernant le nom de la recette et la description dans le même « <code>if</code> » permet de gagner en performance ;- L'utilisation des instructions « <code>break ;</code> » ou « <code>continue ;</code> » permettent un gain de performance puisque dès qu'une condition est remplie, on passe à l'itération suivante.- La condition « <code>if</code> » placée avant la seconde boucle permet de gagner en performance puisque la boucle parcourt un tableau.
--

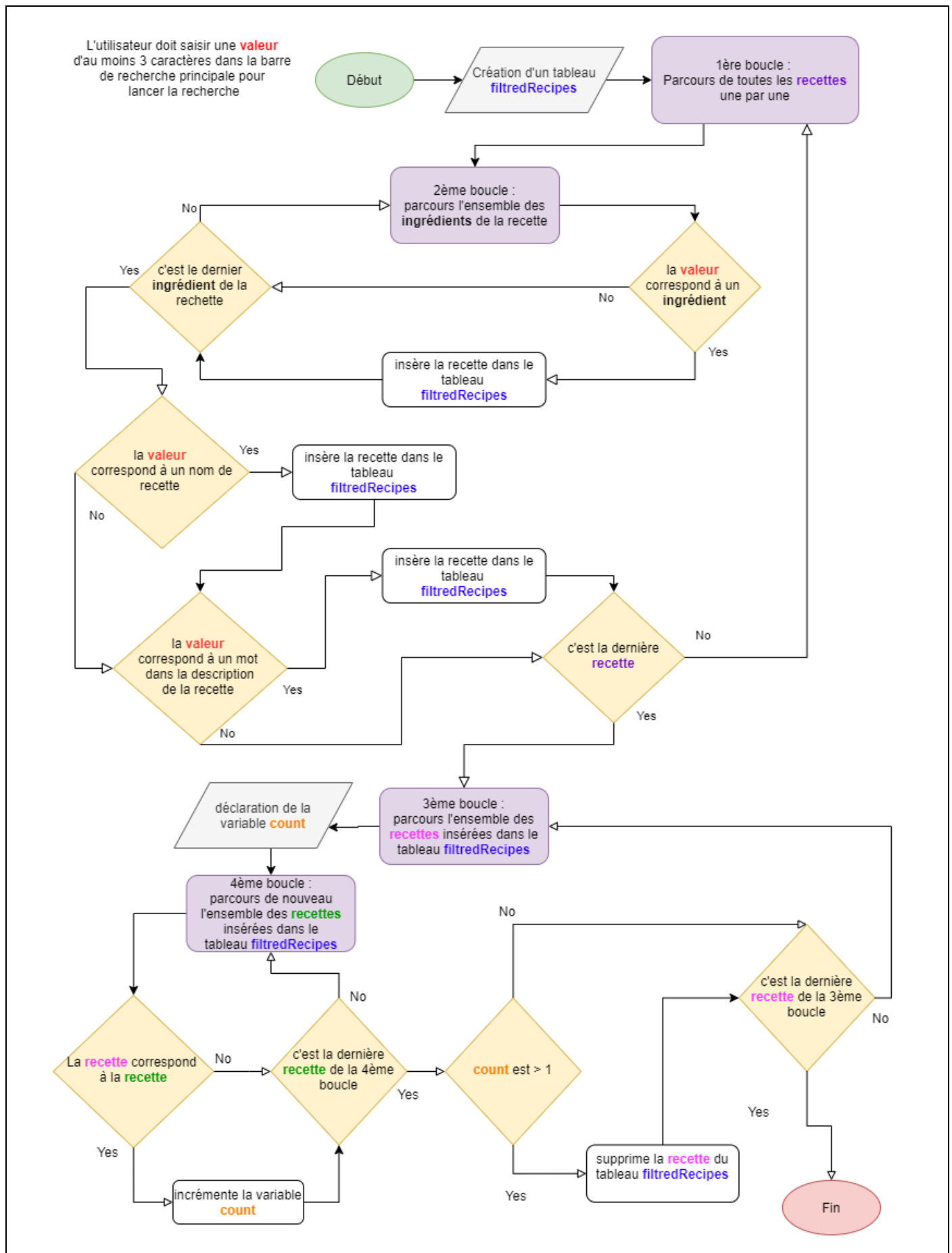


Figure 1 – Diagramme d'activité loop « for...in »

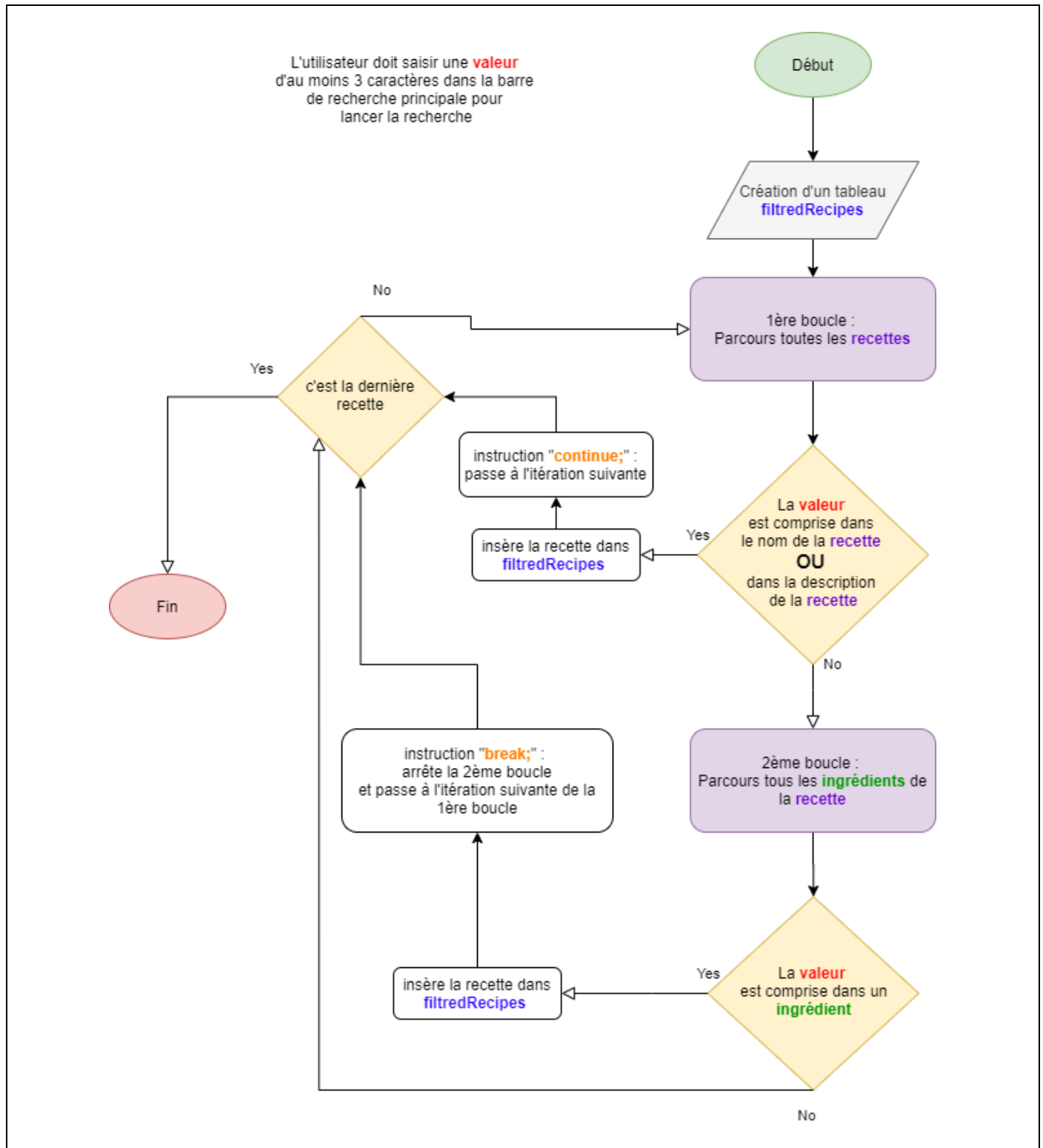


Figure 2 – Diagramme d'activité loop « for...of »