



Implémentation Rust de `cgvg`

Contexte

`cgvg` est un outil très pratique pour explorer une arborescence de fichiers textuels, typiquement le code source d'un logiciel. Le premier utilitaire, `cg`, est un programme qui va rechercher une chaîne de caractères récursivement dans un répertoire, en indexant les résultats trouvés (en leur assignant un numéro). Le second utilitaire, `vg`, est un programme qui permet d'ouvrir un résultat d'une recherche de `cg` dans votre éditeur de texte. `vg` s'utilise en lui indiquant un numéro d'index généré par `cg`.

Objectifs du projet

Le but de ce projet est de faire une nouvelle implémentation en `Rust` de `cgvg`. Ce besoin est motivé par le manque de certaines fonctionnalités dans le `cgvg` historique (comme l'exclusion par la ligne de commande de certains fichiers ou répertoires lors d'une recherche `cg`, ou la compatibilité de `vg` avec les éditeurs de texte récents) ainsi que par le langage de programmation de `cgvg` (perl).

Voici quelques sous-objectifs de ce projet.

- Prendre en main Rust, si nécessaire.
- Comparer les performances de trois versions de `cgvg` : celle historique, une version développée pendant ce projet qui appelle `grep -R` en interne, et une version développée pendant ce projet qui réalise toute la recherche elle-même.
- Mettre en place de quoi maintenir une qualité au nouveau logiciel : tests unitaires, tests en exécution sur un système de fichier contrôlé et automatisation de ces tests via du `GitLab CI`.
- Développer un prototype de programme interactif en ligne de commande pour `vg` basé sur du `fuzzy matching`.