Rapport de relecture de code

Production soumise à la relecture : **TP7 - Sudoku Graphique**  
Projet/code produit par **Elsa Guillet**  
Relecture opérée par **Max Ducoudré** le 13 janvier 2023  
  
**Concision et propreté des rendus**  
Les productions transmises sont particulièrement bien structurées et la précision du nommage  
ne laisse aucun doute sur le contenu des répertoires et fichiers.  
  
Rien à dire sur les noms des fichiers, ils sont explicites. Attention à ne pas envoyer les fichiers compilés .o dans le rendu final. Les fichiers pourraient être subdivisés dans des répertoires. Peut-être ajouter un readme pour expliquer comment lancer le programme, en effet, si on ne met pas d'argument au programme, il segmentation fault apparaît.  
  
**Installation et compilation des sources transmises**  
Un makefile propre, lisible et complet permet la compilation de toutes les sources sans aucun warning  
et avec les drapeaux -Wall et -ansi (voire -pedantic). Le makefile gère correctement les dépendances  
et intègre une règle pour le nettoyage des produits de compilation.  
  
Le makefile est lisible et aucune erreur se lève à la compilation. Il aurait peut-être été intéressant de rajouter une règle "make run" pour faire un lancement du programme sans erreurs (cf. critique précédente)  
  
**Fonctionnalités implantées dans les rendus**  
Les programmes démontrent, durant leurs exécutions, qu'ils réalisent correctement l'intégralité  
du cahier des charges.  
  
Le programme réalise tous les éléments du cahier des charges hormis les bonus.  
  
**Utilisations et sécurité des programmes produits**  
Un cas d'utilisation normal fait échouer un programme (erreur, segfault, etc...).  
  
A aucun moment il est expliqué comment utiliser l'exécutable. En effet, lancer l'exécutable seul sans argument provoque une erreur segfault. Il aurait été intéressant d'ajouter un fichier pour expliquer les arguments ou envoyer une erreur plus explicite à la console si aucun n'argument n'est donné au programme.  
  
**Nommage des éléments dans les productions rendues**  
Les identifiants suivent des règles de nommages sur tout le code de manière cohérente.  
  
Les fonctions, leurs arguments, les constantes et les noms de variables sont assez explicite et logique. Par instants, la faible présence de commentaires dans le projet peux rendre la lecture du code de certaines fonctions compliqué.  
  
**Efficacité et performance des productions rendues**  
Les programmes sont basés sur des algorithmes attendues.  
  
Le programme utilise les structures de données attendues à partir du TP6. C'est-à-dire un tableau d'entier à double entré.  
  
**Commentaires dans les sources des productions rendues**  
Un passage trop rusé ou surprenant manque de commentaires ou encore un commentaire est présent mais non-éclairant .  
  
Le nom des variables & des fonctions sont explicites & le code est bien subdivisé en petite tâche, la maintenance par autrui ne s'avère donc pas trop compliqué malgré la très faible, voir inexistante, présence de commentaires sur l'ensemble du programme pour lever certaines ambiguités.  
  
**derniers commentaires :**  
Ce sudoku graphique respecte bien le cahier des charges, il est fonctionnel et son code source est bien organisé. Pour résumer les améliorations possibles : -Ajouter un make run (pour l'exemple) -Gérer toutes les cas de lancement de l'exécutable pour éviter de segfault. -Retirer les fichiers compilés à l'envoi -Ajouter des commentaires -Ajouter une fonction "print usage" qui affiche l'utilisation normale de l'exécutable en cas d'erreur de lancement par l'utilisateur.