Rapport de relecture de code

Production soumise à la relecture : **TP7 - Sudoku Graphique**  
Projet/code produit par **Mathéo Bourdelat**  
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.  
  
Les fichiers rendus ont des noms explicites & logique.  
  
**Installation et compilation des sources transmises**  
Un makefile permet la compilation de toutes les sources sans grave warning et avec, a minima, le flag -Wall.  
  
Dans le rendu, les fichiers compilés sont déjà présents et il n'y a pas de règle "clean" dans le makefile pour relancer une compilation de zéro. Il est donc difficile de recompiler le programme pour vérifier si des erreurs de compilations sont levés (à part si on supprime à la main les fichiers compilés) De plus, le fichier makefile est très concis, il n'y a aucune règle de dépendances , aucune règle pour "clean" et aucune règle "run" pour lancer le programme.  
  
**Fonctionnalités implantées dans les rendus**  
Les programmes répondent correctement à toutes les tâches qu'ils sont censés effectuer.  
  
Le programme se contente de faire le minimum. Il fonctionne, il est possible de faire une partie de sudoku mais : -Il n'y a pas de gestion de fin de partie -On est obligé de quitter & relancer le programme pour faire une nouvelle partie -L'interface graphique n'est pas très ergonomique (blanc sur fond noir & chiffres en petit) -Il n'y a aucune couleur et animations  
  
**Utilisations et sécurité des programmes produits**  
Un cas d'utilisation normal fait échouer un programme (erreur, segfault, etc...).  
  
Si on lance l'exécutable sans argument, le programme lève un "segfault" mais affiche quand même à la console "grille introuvable" ce qui nous donne un indice sur les arguments à passer. Il serait mieux de mettre fin au programme avant l'arrivée de l'erreur et d'afficher à la console de manière explicite comment lancer l'exécutable comme par exemple "Usage : ./tp7 ".  
  
**Nommage des éléments dans les productions rendues**  
Les identifiants apportent de l'information facilitant la compréhension par autrui.  
  
Tous les éléments du programme ont des noms explicites qui permettent la compréhension par autrui.  
  
**Efficacité et performance des productions rendues**  
Les programmes sont basés sur des algorithmes attendues.  
  
Le programme utilise bien la structure de données utilisé dans le TP6 sans tenter de faire quelque chose de plus optimisé.  
  
**Commentaires dans les sources des productions rendues**  
Des commentaires précisent les parties critiques du code, levant les ambiguïtés.  
  
Les noms des identifiants sont explicites et il y a des commentaires devant toutes les fonctions et devant des parties du programme ambiguë.  
  
**derniers commentaires :**  
En améliorations possible : -Mettre ttoutes les constantes dans les fichiers .h -Compléter le makefile -Trouver une solution pour enlever les variables globales "last\_x" et "last\_y" dans graph.c -Commenter les fonctions dans les fichiers .h plutôt que dans les fichiers .c et expliquer le rôle de chaque arguments (c'est plus simple de les lires dans un fichier .h que dans un fichier .c)