Projet

BOUTON Nicolas

1 Sudoku.c

1.1 Initialisation

J' appelle la fonction lire fichier avec le nom qui a été donné dans le makefile et qui initialise le tableau elem de la structure sudoku.
Ensuite j' appelle calcule val possible sudoku qui calcul les valeurs possible pour chaque case qui ne sont pas des cases de travail.
Puis j'initialise le undo.

J' initialise la fenêtre graphique et j'affiche le sudoku. Et je retourne le sudoku.

1.2 Action

La fonction prends en argument le sudoku et le nom du fichier. Je déclare et initialise 5 variables.

Je fais une boucle avec en condition une fonction qui renvoie 1 si on a gagné et 0 sinon. Donc on continue tant qu'on a pas gagné. J' appelle la fonction wait key arrow clic qui renvoie une constante que je compare.

1.2.1 EST CLIC

On joue, donc on appelle la fonction jouer qui change la valeur en mettant la prochaine valeur possible et implemente le undo, puis on recalcul les valeur possibles et on affiche le sudoku.

1.2.2 EST TOUCHE

On regarde quel est le caractère :

- U : On appelle la fonction decremente undo puis on recalcul les valeurs possibles et on affiche le sudoku.

- V : On trouve une solution ou non avec la fonction resolution puis on calcul les valeurs possible et on affiche. Ensuite on test si on a gagné.
- S : On sauvegarde en écrivant dans un fichier avec la fonction ecrire fichier.
- Q : On quitte le jeu en appelant la fonction termine.

1.3 Fin

On quitte le jeu en vérifiant si on a gagné ou non en appelant la fonction verif test termine.

La fonction appelle une fonction qui renvoie 1 si c'est gagné ou 0 si c'est perdu puis on affiche si on a gagné ou perdu dans le terminal.

2 Gestion Sudoku

2.1 Stokage du sudoku

La structure sudoku contient :

- un tableau elem d'une structure element
- Un tableau UNDO qui permet de stocker 100 coups jouer
- un entier pour indiquer la première case vide du tableau UNDO

La structure element qui permet de stocker les éléments d'une case contient :

- un entier qui est la valeur de la case
- un entier qui indique si c'est un case de travail (1 si travail 0 sinon)
- un charactère qui indique si c'est une solution (1 si solution 0 sinon)
- un tableau POSSIBLE qui contient 10 valeurs dont l'indice 0 qui est le vide, et pour chaque indice contient soit 1 si la valeur de l'indice est possible dans cette case et 0 sinon.

2.2 init tab vide

Initialise un tableau vide pour les valeurs possible.

2.3 calcule val possible case

Calcul les valeurs possible pour une case donné.
ligne colonne permet de calculer sur les lignes et les colonnes en même temps en ajoutant 1 a chaque fois en remplacent l'indice des lignes ou des colonnes par ligne colonne.
Et pour calculer sur les grand carré, en divisant par 3 ligne colonne pour les colonne et en faisant modulo 3 ligne colonne pour les lignes nous permet de ne pas sortir des grands carré.

2.4 verif val possible

Renvoie 1 si il existe une valeur possible hormis 0 et 0 sinon.

2.5 calcule val possible sudoku

C'est une boucle qui appelle la fonction calcule val possible case si c'est une case de travail sinon ça initialise un tableau vide.

2.6 gagne

Renvoie 1 si toute les valeurs possible de chaque case sont à 0 sauf à l'indice 0.

2.7 prochaine val possible

Prends la valeurs de la case et la met dans un entier en ajoutant 1. Ensuite on fait une boucle pour chercher la prochaine val posiible et s'arrete quand elle la trouve, et renvoie l'entier modulo 10. Comme la valeur 0, qui est le vide, est toujours possible il n' y a pas de problème.

2.8 implemente undo

A chaque fois qu' on clique et qu'on modifie une case on stock les coordonné(numéro ligne/colonne) et la valeur de la case dans le tableau UNDO et on ajoute un a l'entier qui indique la prochaine case du tableau qui est vide.

2.9 decremente undo

On prend la valeur et les coordonné(numéro ligne/colonne) de la dernière case changé, donc l'indice est compteur-1 du tableu UNDO et on fait l'inverse de la fonction prochaine val possible.

2.10 resolution

Si la case a une seul valeur possible on lui met la prochaine valeur possible et si c'est 0 on lui remet la prochaine valeur possible et on fait la boucle 5 fois.

3 Affichage

J' utilise la librairie uvsq.
J' ai fait des fonctions qui :

- initialise la fenêtre graphique
- fait un quadrillage
- affiche les valeurs
- affiche les valeurs possibles pour chaque case
- affiche le nom du fichier ouvert

Et j'affiche le sudoku comme dans le fichier texte donc je commence en haut.

3.1 appelle acr

Cette fonction appelle affiche carre rouge lorsqu ' il n'y a plus que la valeurs 0, donc le vide, qui est possible dans cette case de travail.

4 Fichier

4.1 Lecture

Ouvre le fichier en mode lecture.
Tant que le caractère est différent de la constante "EOF" qui annonce la fin du fichier et tant que j (qui est l'indice pour les lignes) est inférieure à 9 pour ne pas écrire en dehors du tableau on continue la boucle :

- Si le caractère est un "." (la valeur est 0) ou une "*" (la valeur est le caractère qui suit) alors c'est une case de travail
- Si le caractère est une valeur alors c'est une case de départ

Et on initialse le tableau des valeurs possibles.

4.2 Ecriture

4.2.1 Recherche du nom

On regarde d'abord si après le premier point il y a un "s". S' il y a un "s" alors le fichier dans lequel on va sauvegarder est "exemple.001.suduku". Sinon on récupère le nombre après le premier point avec des chaînes de caractères qu'on met dans un int et on regarde si il est plus petit que 100 ou 10 pour rajouter des 0 dans la chaîne de caractère final et on sauvegarde dans le fichier qui à la même chaîne de caractère que final.

4.2.2 Sauvegarde

On fait l'inverse de se qu' on a fait dans la lecture d'un fichier.