

Projet Sudoku

24 janvier 2020

Table des matières

1	Introduction	2
1.1	Installation	2
1.2	Objectif	2
1.3	Instructions	2
2	Fonction principal	2
3	Affichage	2
4	Gestion du sudoku	3
5	Stockage des indices pour chaque case	3
5.1	Description	3
5.2	Fonctionnement	4
6	Modifier une case	4
7	Lecture de fichier et sauvegarde	5
7.1	Description	5
7.2	Fonctionnement	5
7.2.1	Fonction de lecture	5
7.2.2	Fonction d'écriture	5
8	Pile	6
8.1	Description	6
8.2	Fonctionnement	6
9	Fonction trouver	6
10	Extension	6

1 Introduction

1.1 Installation

Tout est expliqué dans le readme.

1.2 Objectif

L'objectif du projet est de programmer un sudoku.

1.3 Instructions

Voici les instructions du jeu

- Cliquez sur une case à l'aide de la souris pour afficher les valeurs possible, attention si vous cliquez sur une case où il y a déjà une valeurs ou en dehors, rien ne se passe.
- Appuyer sur la touche 'u' pour retourner en arrière.
- Appuyer sur la touche 'v' pour résoudre le sudoku.
- Appuyer sur la touche 's' pour sauvegarder.
- Appuyer sur la touche 'q' pour quitter le jeu.
- Lorsque vous avez résolu le sudoku, le programme s'arrête.

2 Fonction principal

La fonction principal initialise les données du jeu, attends le clique ou la touche appuyé et termine le programme quand le jeu est fini.

3 Affichage

En haut, on a le nom du fichier initiale appelé dans le fichier /Make-file. En dessous, on a le sudoku composé de 9*9 cases. Lorsqu'on clique sur une case de travail, c'est-à-dire une case vide ou une valeur de couleur orange, le programme affiche la valeur supérieur (selon les indices affichés en haut de la valeur) ou sinon il y a aucun changement. Les valeurs en rouge sont des valeurs de départ.

Franck Quesette -- UVSQ (2016)								
niveau1.sudoku								
29	4	29	1	256	8	7	3	26
7	12	5	236	236	3	9	128	4
3	6	28	9	7	4	18	5	12
589	379	1	4	369	2	578	78	59
6	279	2479	8	9	1	457	247	3
289	239	23489	5	39	7	6	1248	129
12	5	237	23	4	9	13	6	8
4	139	6	3	1358	35	2	1	7
12	8	23	7	1235	6	1345	9	15

4 Gestion du sudoku

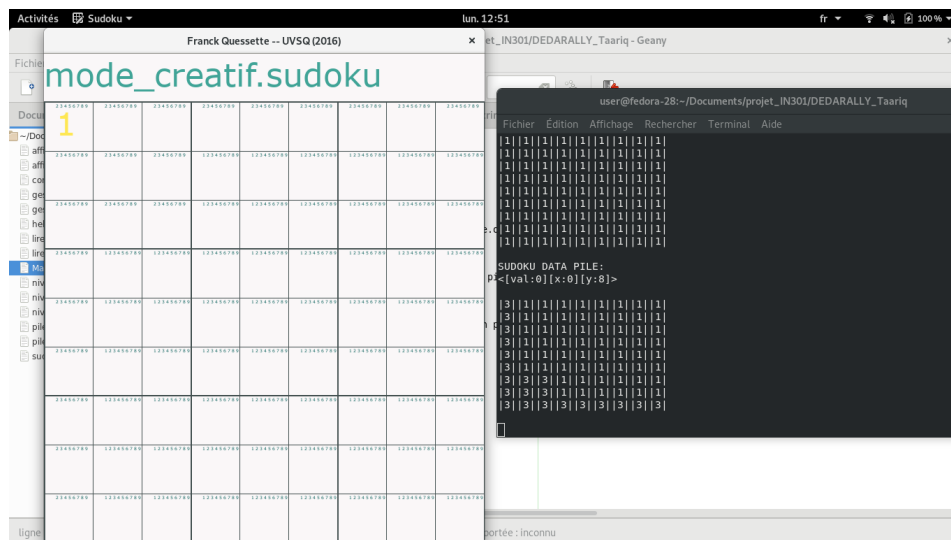
Le programme est composé d'une structure sudoku :

- struct élément qui contient, la valeur de la case, deux booléens si c'est une case de travail, si c'est une case qui est vérifiée. Et contient un entier (unsigned int) où est stockés les valeurs possible pour une case.
- Une pile qui permet de revenir en arrière.

5 Stockage des indices pour chaque case

5.1 Description

Les indices sont sauvegardés dans des entiers. Pour chaque case on a un entier. Dans ces entiers de 10 bits, on sauvegarde dans chaque bits des valeurs entre {0,1}. Un entier pour valeur maximal (quand il y a plus d'indices), 11111111 soit en hexadécimale 3FF. En hexadécimal ça donne un tableau comme ceci :



5.2 Fonctionnement

La fonction valeur possible passe par chaque case et remplit les indices :des colonnes,des lignes et des cases.

- Pour remplir les colonnes, on fait varier une valeurs qui part du début de la colonne et va jusqu'à la fin.Pour chaque valeur lue,on ajoute à la valeur d'indice 1 qu'on shift selon la valeur de la case :par exemple pour 5 on détermine une valeur temporaire 0x00000001 qu'on shift de 5 éléments vers la gauche,on aura :1,10;100;1000,10000.Puis on ajoute cette valeur par la valeur indice en appliquant un 'ou' logique :si la valeur d'indice était 00100001 on aura,00110001.
- on fait de même chose pour les lignes.
- Pour remplir les grandes cases(3*3)on prend toutes les valeurs de cette grande case et on les ajoute dans les indices de toutes les cases.Par exemple pour une grande case contenant 0,5,6 on aura pour chaque case(les 9 cases)dans la grande case l'indice :1100001.

6 Modifier une case

Lorsque l'on clique,sur une case,on incrémente la valeur par l'indice supérieur.Pour cela,on shift la valeur de l'indice de cette case par la valeur de la case :si la dernière valeur de l'indice shifté par la valeur est égale a 1,alors, on passe à la valeur suivante,jusqu'à que cette valeur soit égale à 0.Pour cela, on prend la valeur de l'indice shifté par la valeur de la case qu'on compare avec le 'et' logique pour déterminé si la dernière valeur est égale a 1 ou 0.Par exemple si on a le code suivant dans la valeur d'indice,0000000001 et la valeur de la case de départ est égale à 0 et on clique dessus,on shift la valeur d'indice de 0,on compare 1 & 1 égale à 1,on incrémente 1 à la valeur

de la case, et on compare donc, 0 & 1 égale à 0, le programme s'arrête, et on a comme valeur final 1.

7 Lecture de fichier et sauvegarde

7.1 Description

Le jeu contient 3 niveaux et 1 mode créatif (qui permet d'afficher un sudoku vide). Pour que le programme lise les fichiers, il faut changer le Makefile pour appeler le fichier jeu souhaité : niveau1.sudoku, niveau2.sudoku, niveau3.sudoku, mode_creatif.sudoku.

On peut aussi sauvegarder sa partie en appuyant sur 's'. Un fichier sera généré ou écrasé si le fichier est déjà existant. Pour charger la partie, il suffit de l'appeler dans le Makefile. On peut sauvegarder jusqu'à 999 fichiers.

7.2 Fonctionnement

7.2.1 Fonction de lecture

La fonction de lecture lit chaque caractère jusqu'à la fin du fichier (jusqu'à que `fgetc(F) == EOF`) et la sauvegarde dans le struct `element`. Quand on lit le fichier on a 3 possibilités :

- C'est un caractère : on le sauvegarde dans `valeur`. Si il y a une étoile devant la valeur : on passe à la valeur suivante et on la lit. On sauvegarde cette valeur comme n'étant pas une case de travail. Sinon on la sauvegarde comme étant une case de travail.
- C'est un point : on sauvegarde `VIDE(0)` et on la définit comme une case travail.
- C'est un retour chariot : on passe à la ligne suivante.

7.2.2 Fonction d'écriture

Pour la fonction d'écriture : on commence par créer le nom du fichier de sauvegarde, si c'est un fichier chargé on ajoute 1 au numéro du nom du fichier (exemple : niveau1.009.sudoku va devenir niveau1.010.sudoku) sinon on crée un fichier avec comme numéro de sauvegarde 001 (exemple : mode_creatif.sudoku va devenir mode_creatif.001.sudoku). Puis on écrit sur le fichier (devant la valeur de travail on ajoute *, sinon on ajoute rien), afin qu'on puisse lire le fichier dans ce programme quand on l'aura chargé.

8 Pile

8.1 Description

La pile sert à sauvegarder :les valeurs et les coordonnées de x et y compris entre $\{0,9\}$,à chaque clique.Lorsque l'utilisateur clique la valeurs qui était dans cette case est sauvegardée.Quand l'utilisateur appuie sur le bouton 'u', on dépile et on restitue la valeurs qui était dans la dernière case où l'utilisateur avait cliqué. La pile est supprimée lorsque l'on quitte le jeu.

8.2 Fonctionnement

On commence par créer une pile.Cette pile contient une structure composée des coordonnées et de la valeur de la case et une adresse.À chaque fois qu'on clique, on sauvegarde les données dans une structure qui elle pointe vers une adresse.Quand,on dépile,on supprime la structure(et les valeurs qui la compose).

9 Fonction trouver

Vérifie si une seul valeur est possible pour chaque case et met la seul valeur possible pour cette case,si ce n'est pas le cas elle renvoie FAUX.

10 Extension

Voici une liste des différentes améliorations possibles du sudoku :

- Permettre au joueur de sélectionner un sudoku avec plus de case($16*16$)ou moins de case($3*3$).
- Lorsqu'on appuie sur 'c',on charge la dernière sauvegarde existante.
- Permet au joueur de sélectionner directement dans le jeu le niveau qu'il souhaite.
- Lorsqu'on appuie sur 'h',le jeu affiche un texte qui explique comment jouer au sudoku.
- Pouvoir créer sa propre grille.
- Afficher un chronomètre et mettre le jeu sur pause.
- Améliorer la fonction trouver