

UE Projet informatique

Projet « Sudoku »

Introduction

Le sudoku est une forme de puzzle très connue. On dispose d'une grille de 3 par 3 blocs, eux-mêmes de dimension 3 par 3. La plupart des 9 par 9 cases sont vides, mais certaines contiennent initialement un chiffre de 1 à 9. Le joueur doit compléter entièrement cette grille tout en respectant les règles suivantes :

- Chaque ligne de la grille ne doit pas contenir le même chiffre 2 fois.
- Chaque colonne de la grille ne doit pas contenir le même chiffre 2 fois.
- Chaque bloc de 3 par 3 ne doit pas contenir le même chiffre 2 fois.

Pour ce projet, nous commencerons par des grilles composées de $N = 2$ par 2 blocs, eux mêmes de dimension 2 par 2, et remplies de chiffres allant de 1 à 4. Votre programme final devra faire les choses suivantes :

- Afficher une grille.
- Vérifier si une grille respecte bien les règles du sudoku.
- Résoudre une grille.
- Générer une grille incomplète pour le joueur.

Méthodes de base : grille et affichage

Il est important de commencer par les éléments les plus simples. Tout d'abord, sous quelle forme votre programme va-t-il encoder la grille du sudoku ? Si cette grille contient des nombres, comment gérer simplement le cas où une case ne contient pas encore de valeur ?

Une fois ces décisions prises, écrire les fonctions suivantes :

- Une fonction qui prend en paramètre la grille et l'indice d'une ligne, et affiche une ligne à l'écran.
- Une fonction qui prend en paramètre la grille et l'affiche à l'écran.

Tentez de rendre votre affichage le plus confortable possible. En particulier, prenez soin de permettre une distinction entre les blocs de la grille. Pour chacune des fonctions que l'on rédige, il faut les tester. Initialisez une grille de taille 4 par 4 avec des valeurs de test. Pour chacune de vos fonctions, faites usage des fonctions précédemment définies si cela est possible. Pour le reste du projet, c'est à vous de déterminer quelles fonctions sont nécessaires. Découpez toute tâche trop complexe en plusieurs fonctions ; cela permet de rendre le code plus facile à lire et à écrire.

Vérification d'une grille

Une fois une grille initialisée et affichable, écrivez une fonction qui prend en paramètre une grille, et vérifie que les règles du sudoku sont bien respectées. Pour cela, nous vous recommandons les étapes suivantes :

- Une fonction qui vérifie qu'une ligne d'indice donné est valide (n'a pas de chiffre en double).
- Une fonction qui vérifie qu'une colonne d'indice donné est valide.
- Une fonction qui vérifie qu'un bloc de coordonnées données est valide.
- Une fonction qui vérifie toutes les lignes.
- Une fonction qui vérifie toutes les colonnes.
- Une fonction qui vérifie tout les blocs.
- Une fonction qui vérifie toute la grille.

Évitez au maximum la redondance et appelez les fonctions déjà définies dès que possible.

Résolution d'une grille

Résoudre le sudoku est la tâche principale de ce projet. L'ensemble des autres tâches (comptage de solutions ou génération de grilles pleines et incomplètes) ne sont que des variations de l'algorithme que vous allez implémenter pour résoudre un sudoku. Pour cette tâche, vous devez implémenter une fonction qui prend en paramètre une grille **G** (qui peut être déjà partiellement complétée) et la résout, si cela est possible. L'algorithme exécuté par cette fonction est un algorithme récursif de *backtrack*, dont le pseudo-code est le suivant :

- si **G** est invalide, renvoyer **None**
- si **G** est complète, renvoyer **G**
- soit **x**, **y** les coordonnées du premier emplacement libre dans **G**
- pour chaque **valeur** possible de 1 à N^2 :
 - mettre **valeur** aux coordonnées **x**, **y** de **G**
 - résoudre récursivement **G**
 - si le résultat n'est pas **None**, renvoyer ce résultat
 - effacer **G** aux coordonnées **x**, **y**
- renvoyer **None**