PX111-info (1) - Jouons au sudoku

Grenoble INP – Esisar – Academic year 2025-26 • *Compiled on 30 novembre 2025*

Version du 30 novembre 2025

Objectif pédagogique

Les 4 séances de PX-info seront sur le même modèle, 2 à 3 interfaces de jeu seront réalisées.

- Pratiquer l'algorithmique et la programmation C
- Gagner de l'autonomie dans la conception et les tests logiciels
- Pratiquer le vocabulaire de lecture de programmes
- Pratiquer la fiche de syntaxe C "niveau 1".

Ni le code ni les réponses ne sont évaluées, l'objectif est de **pratiquer**.

Vous compilerez toujours à la ligne de commande Vous utiliserez la fiche de syntaxe du C niveau 1. Il est interdit d'utiliser autre chose que des tableaux statiques.

Description du jeu et objectif de programmation

Le sudoku (cf https://sudoku.com/fr) est un *puzzle* utilisant une grille carrée composée de 9 régions (représentées en gras sur la figure ci-dessous) ; chaque région est elle-même composée de 9 cases. On part d'une grille partiellement remplie et on doit la compléter en respectant les trois règles suivantes :

- Chaque colonne doit contenir les 9 chiffres de 1 à 9 (tous distincts)
- Chaque ligne doit contenir les 9 chiffres de 1 à 9 (idem)
- Chaque région (3x3) doit contenir les 9 chiffres de 1 à 9 (idem)

Une grille est résolue lorsqu'elle est pleine (toutes les cases ont été remplies) et satisfait les conditions précédentes.

Dans cet exemple `s[0][0]` vaut 5, `s[2][7]` vaut 6, `s[0][3]` vaut 0, etc. Si l'on considère la case vide où on a dessiné un petit rectangle, il n'est pas possible de poser 4,8,3,1 à cause de la ligne, et 7,9,6,2,1,8 à cause de la colonne. Il n'y a donc qu'une seule possibilité (la valeur 5).

Si l'on considère la case vide où on a dessiné un petit disque, il n'est pas possible de poser 5,3,7 à cause de la ligne, et 2 à cause de la colonne, et 6 à cause de la région, il y a donc encore 4 possibilités (les valeurs 1,4,8,9)

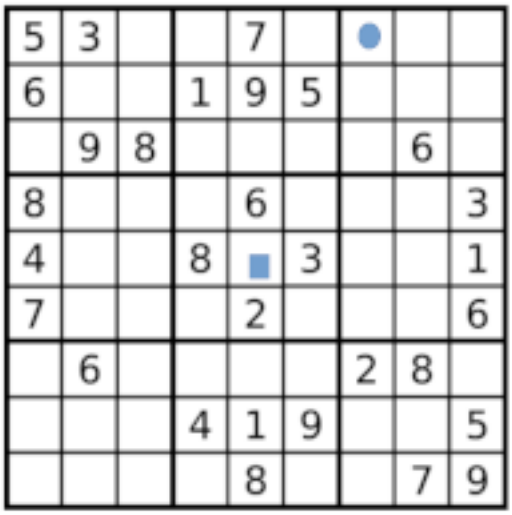

Figure 1: Grille. La case (0,0) est en haut à gauche

Objectif: réaliser un jeu interactif de résolution d'une grille de sudoku.

Une grille vous est fournie, et vous devez réaliser une interface pour qu'une personne puisse essayer de résoudre cette grille.

Le code fourni comporte un unique fichier `.c` que l'on compile de la manière habituelle.

Codage d'une grille

Une grille de sudoku est représentée par un tableau d'entiers de 9 lignes et 9 colonnes : `typedef int sudoku[9][9];`

Les cases d'un sudoku en cours de résolution prennent les valeurs suivantes:

- 0 si la case est vide
- entre 1 et 9 sinon.

Étapes à réaliser

1) Comprendre le code fourni
2) Compléter le jeu
3) Bonus : écrire un algorithme de résolution automatique d'une grille.

Étape 1 : questions pour comprendre le code fourni - à rédiger avant de programmer

- Comment (et où) la grille est-elle déclarée ? initialisée ? À l'aide de quelle fonction pourra-t-on visualiser cette grille ?
- Dans le `main` écrire un appel à la fonction `lireAction` de la façon suivante:

```
int entreeX,entreeY,entreeV;
bool res = lireAction(&entreeX,&entreeY,&entreeV);
printf("%d,%d,%d\n", entreeX, entreeY, entreeV);
```

puis tester. Comment fonctionne `lireAction` ?

Note importante : le “passage de paramètres par adresse” n’a pas été vu en cours, plus de détails seront donnés en 2A.

- Quelle sera la fonction utilisée pour vérifier qu’une grille est complétement remplie ?
- Comment décrire les 9 cases de la région contenant la case (x,y) ? (indice : utiliser la division entière, et *faire un dessin*)

Étape 2: complétion du jeu

On pourra commencer:

- soit par écrire la boucle de jeu qui appelle les fonctions fournies (mais non encore implémentées complétement)
- soit par écrire les implémentations des fonctions, puis la boucle de jeu.

On fera attention à réaliser des tests au fur et à mesure de l’écriture du code.

Pour la boucle de jeu, on pourra écrire ce qu’on appelle quelquefois “boucle réactive”:

```
bool fini = false;
while (!fini){ // ou boucle infinie
    demande_info(); // interaction avec l'utilisatrice.
    calcule();      // le jeu lui même
    if (condition) fini = true;
}
```

Si vous avez du temps

Concevoir, puis programmer:

- Une façon (simple) de résoudre une grille de sudoku.
- Une interface de lecture d’une grille vide.