

IN104 - Solveur de Sudoku - TP3

Ugo Vollhardt

CEA LIST

15 mai 2018

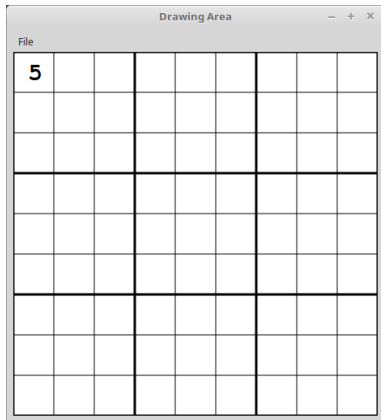
Déroulement prévisionnel :

1. Rappels sur Git, mise en place versionnement, structure donnée et premières fonctions de mises à jours de possibilités.
2. Rappels/introduction à la notion de récursivité, poursuite du travail de la première séance et mise en place des premiers algo de résolution.
3. Fin d'implémentation des premiers algo de résolution + rappels sur sujet au choix.
4. Introduction à la librairie GTK+ pour Python et C, implémentation d'une interface minimale.
5. Séance en autonomie.
6. Revu des points précédents et approfondissement si besoin.

Amélioration des discrimination des possibilités

- Méthode par exclusion : Pour une section donnée (Colonne, Ligne ou carré), si une possibilité n'est présente que sur une seule case (même si cette case en a d'autre), alors la valeur de cette case est cette possibilité.
- Méthode par n-uplet exclusive : Pour une section donnée, si n cases contiennent exactement les même n possibilité, alors ces n possibilités peuvent être retirées des possibilités de toutes les autres cases.

Interface graphique minimal



Exemples de codes sur le GitHub dans le dossier `examples/`

- `Canvas/` : exemple de l'utilisation de l'objet `canvas` pour `TKinter` et `Drawing area` pour `Gtk+`. Support pour dessiner.
- `menu_bar/` : exemple d'utilisation des classes/types permettant de faire un menu

Objectifs de la séance

1. Finir les implémentations simple et avec forçage.
2. Ajouter d'autres algorithmes de discrimination pour réduire le recours au forçage.
3. Designer une interface graphique minimale (affichage grille, petit menu ou boutons).