

Rapport Projet APO

Sudoku (et variantes) : résolution et génération

Méthodologie

Introduction

L'objectif de notre projet est de développer un programme capable de générer et résoudre des grilles de Sudoku, y compris certaines variantes comme le Multidoku.

Pour mener à bien cette mission, nous avons mis en place une approche collaborative, en nous répartissant les différentes tâches en fonction des compétences et des motivations de chacun.

Répartition des rôles et contributions

Amine Mounib s'est chargé de l'interface graphique, ainsi que de la liaison entre les différentes parties du projet. Son rôle a été essentiel pour assurer la fluidité et la cohérence du programme, notamment en réorganisant le projet et en ajustant les classes. Il a également travaillé sur la lecture et l'importation de grilles de Sudoku depuis un fichier texte, afin de permettre au programme de charger facilement des grilles déjà existantes.

Arthur Joly a pris en charge l'implémentation de l'algorithme de backtracking, qui permet de résoudre les grilles de Sudoku de manière efficace. Il s'est aussi occupé de la génération de nouvelles grilles à partir d'une grille complète, ainsi que de l'implémentation du Multidoku, une variante plus complexe du Sudoku.

Louis Ferret s'est concentré sur l'implémentation des règles de déduction, qui permettent de résoudre certaines parties des grilles sans recourir au backtracking. Il a également travaillé sur la gestion des logs, le débogage du code et l'optimisation générale du programme. Grâce à lui, notre projet est devenu plus stable et performant.

Mouhamed Kandji s'est occupé à la fois de la conception et de l'implémentation d'un solveur entre règles de déduction et du backtracking. Il a également réalisé plusieurs diagrammes pour illustrer et structurer nos idées, ce qui nous a beaucoup aidés à organiser notre travail et à avoir une vision plus claire de l'architecture du projet.

Déroulement du projet

Pour mener à bien notre projet, nous avons structuré notre travail en plusieurs étapes afin d'assurer une progression efficace et organisée.

Nous avons d'abord commencé par une phase de conception, où nous avons défini l'architecture du projet et réparti les responsabilités entre chaque membre de l'équipe. Cela nous a permis d'avoir une vision claire de ce que chacun devait faire et de garantir une bonne coordination.

Ensuite, nous sommes passés à la phase de développement, où chacun a travaillé en parallèle sur son module, tout en veillant à respecter les interfaces définies en amont. Cette approche nous a permis d'avancer rapidement tout en assurant la compatibilité entre les différentes parties du programme.

Concernant la phase de test, on testait directement pendant notre implémentation, puis une fois tout merge dans le main, on demandé a quelqu'un d'autre de pull et essayer ce qu'on venait d'implémenter

Enfin, nous avons consacré du temps à la finalisation et à la documentation. Nous avons commenté le code, rédigé une documentation pour expliquer son fonctionnement...

Pour assurer un bon suivi des versions et faciliter la collaboration, nous avons utilisé Git, ce qui nous a permis de travailler simultanément sur le projet tout en gardant une trace des modifications effectuées par chacun. Cette organisation nous a permis de progresser efficacement et d'assurer une bonne qualité de code tout au long du développement.