Rapport SAE 2.1

Lenny FOULOU groupe 3
Hamza BAZZAOU groupe 3

Sommaire

- Introduction
- Fonctionnalités
- Structure des programmes
- Algorithme du Solver
- Conclusion

Introduction

Notre projet consiste à développer un jeu de Sudoku en Java, un défi de logique et de réflexion passionnant. Le Sudoku est un jeu où l'objectif est de remplir une grille de 9x9 cases, divisée en neuf sous-grilles de 3x3 appelées "régions", avec les chiffres de 1 à 9. Chaque ligne, colonne et région doit contenir une fois chaque chiffre, sans répétition.

Ce projet met en avant notre capacité à développer une application Java offrant une expérience de jeu fluide et stimulante. Le Sudoku ne requiert pas de compétences mathématiques avancées, mais plutôt une pensée logique et une méthode de résolution systématique. Nous visons à offrir différentes variantes et niveaux de difficulté, des grilles plus simples pour les débutants à des casse-têtes plus complexes pour les joueurs expérimentés, garantissant ainsi une expérience divertissante pour tous les amateurs de puzzles.

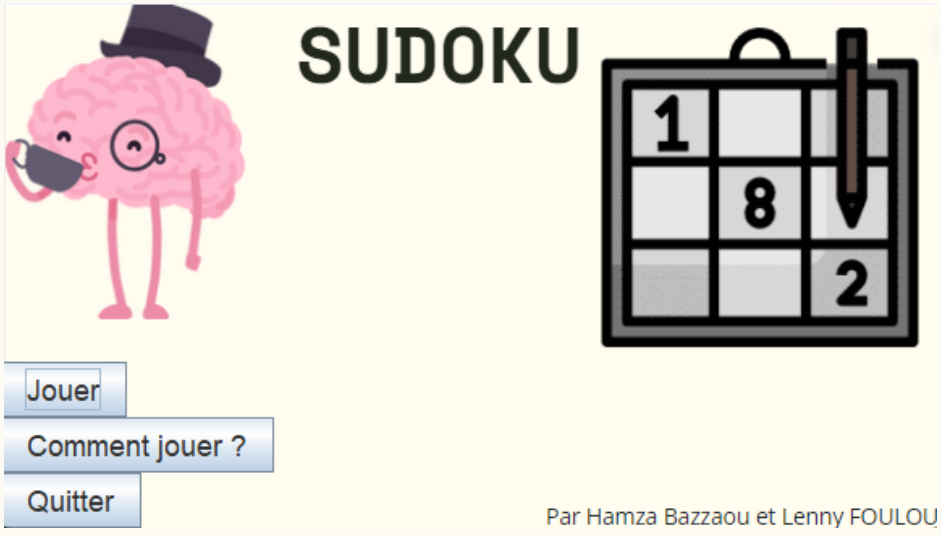
Fonctionnalités

Notre application Sudoku offre une gamme complète de fonctionnalités visant à offrir une expérience de jeu interactive, éducative et agréable. Voici une présentation détaillée de chacune de ces fonctionnalités :

1. Le Menu

Le menu du jeu de Sudoku offre une interface claire et conviviale pour les joueurs. À son lancement, une fenêtre s'affiche avec trois options principales :

Jouer : En cliquant sur ce bouton, les joueurs peuvent accéder aux grilles de Sudoku et commencer à jouer. Différents niveaux de difficulté sont généralement proposés, permettant aux joueurs de choisir celui qui correspond le mieux à leur niveau d'expérience.



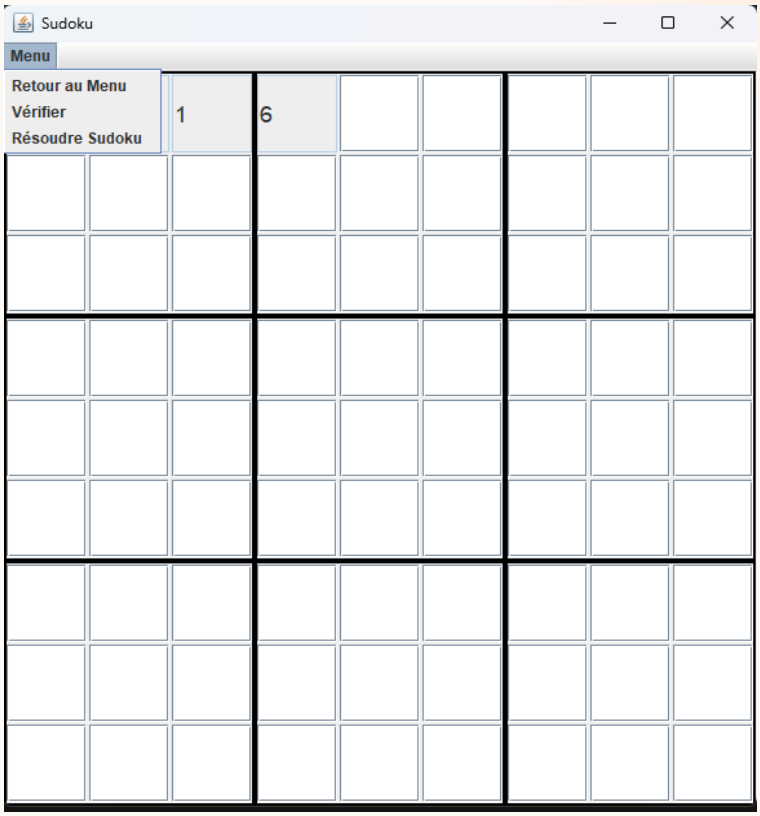
Comment jouer ? : Ce bouton fournit aux joueurs des instructions et des conseils sur la façon de jouer au Sudoku. Il explique les règles de base du jeu et offre éventuellement des stratégies pour résoudre les puzzles plus efficacement.

Quitter : En cliquant sur ce bouton, les joueurs peuvent quitter le jeu et revenir à leur bureau ou à l'écran d'accueil de leur appareil.

2. Interface Utilisateur Conviviale :

L'interface utilisateur a été conçue pour être conviviale, intuitive et facile à utiliser, même pour les débutants. Voici quelques éléments clés de notre interface :

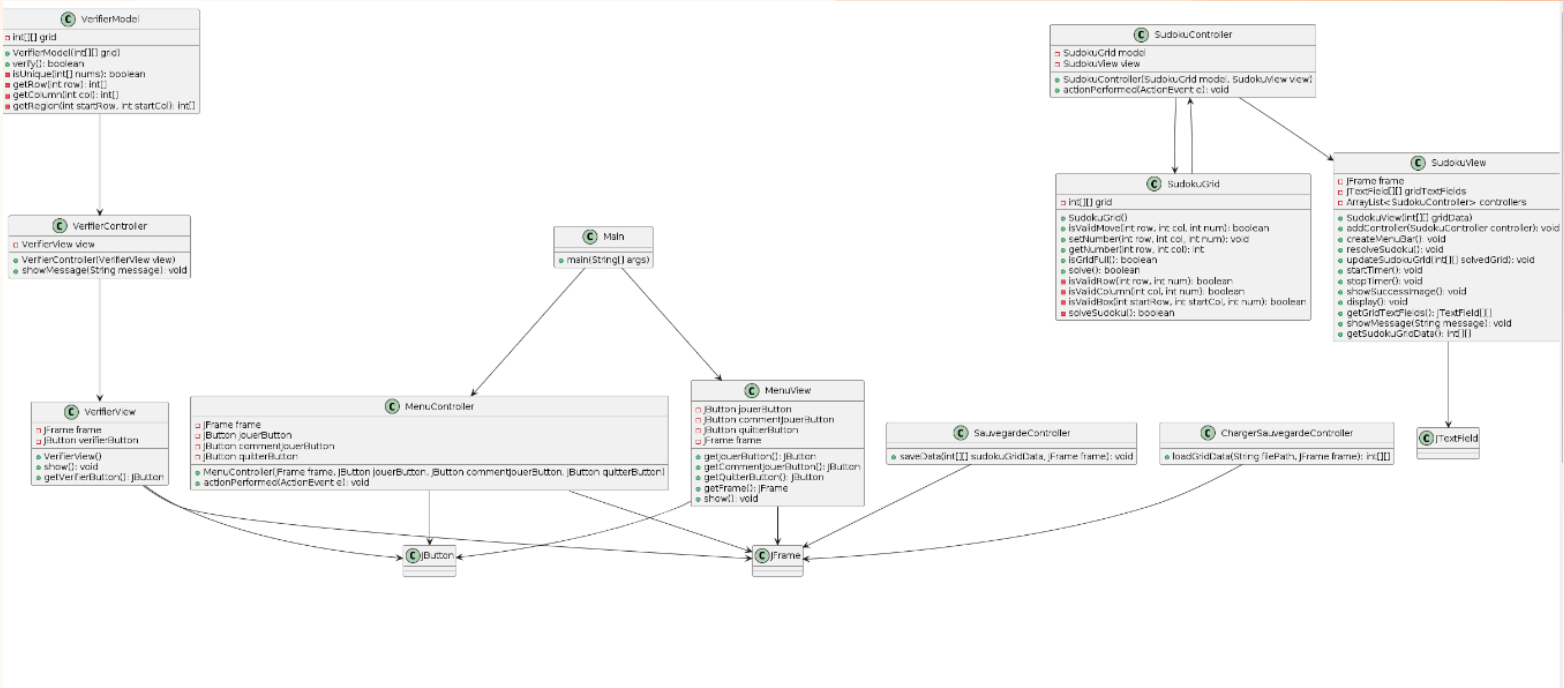
Grille de Sudoku Interactive : Une grille de Sudoku 9x9 est affichée à l'écran, avec des cases interactives permettant aux utilisateurs de saisir les chiffres.



Boutons d'Interaction : Des boutons sont disponibles pour effectuer différentes actions, telles que la vérification de la solution, la résolution automatique de la grille et le retour au menu principal.

Structure des programmes

Programme Concepteur



Main : Cette classe contient la méthode principale **main(String[] args)** qui est le point d'entrée de l'application. Elle est responsable de démarrer l'application en créant et en affichant la vue du menu.

MenuView : Cette classe représente la vue du menu principal de l'application. Elle contient les boutons pour jouer, obtenir des instructions sur le jeu et quitter l'application. La classe fournit également des méthodes pour accéder à ces boutons et pour afficher la vue.

MenuController : Ce contrôleur gère les actions effectuées dans la vue du menu. Il est associé à la vue du menu et aux boutons correspondants. En fonction de l'action de l'utilisateur, comme cliquer sur le bouton "Jouer" ou "Quitter", ce contrôleur prend des mesures appropriées telles que lancer le jeu ou fermer l'application.

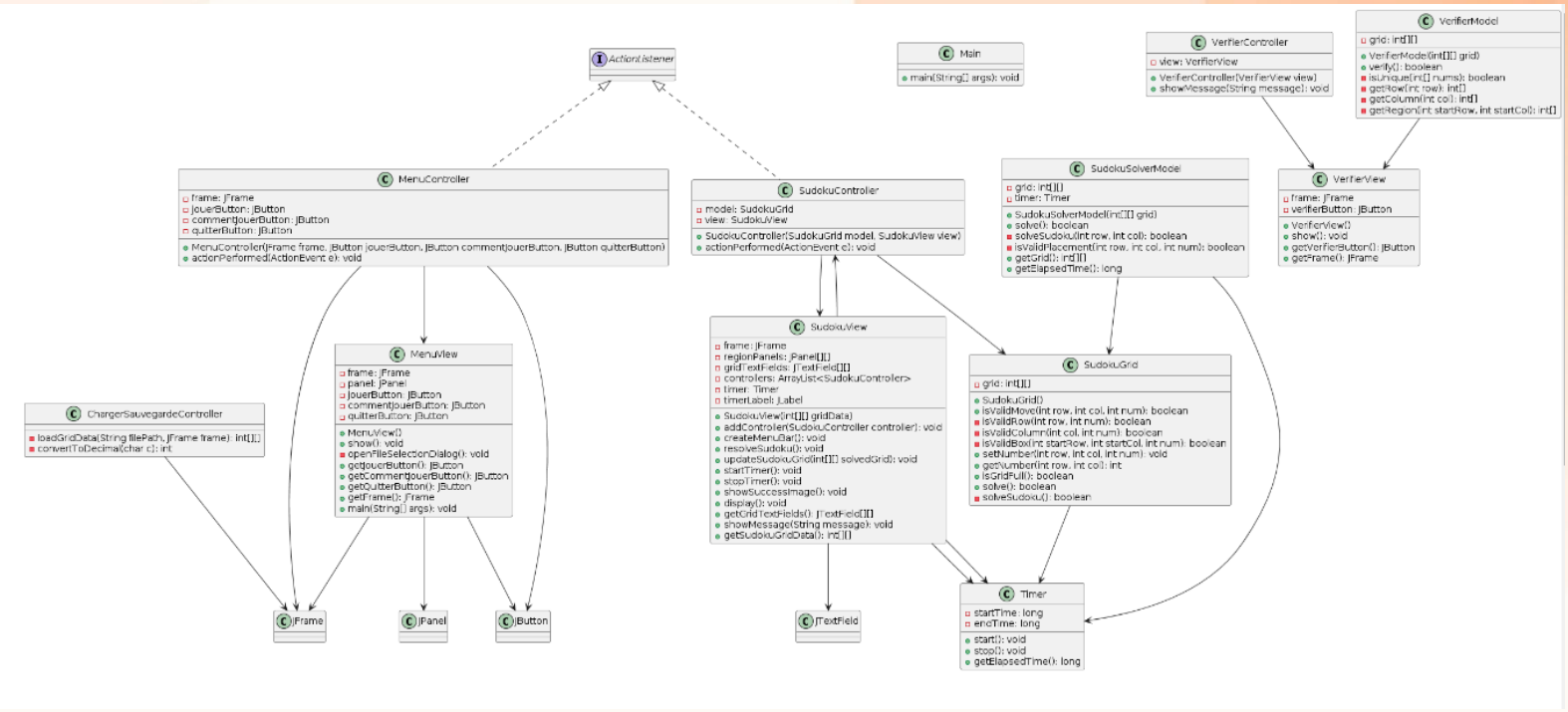
SudokuController : Ce contrôleur est responsable de gérer les interactions entre la vue et le modèle du jeu de Sudoku. Il écoute les événements sur la grille de Sudoku et met à jour le modèle en conséquence.

SudokuGrid : Cette classe représente le modèle du jeu de Sudoku. Elle contient la logique pour vérifier si un mouvement est valide, placer des nombres dans la grille, vérifier si la grille est pleine et résoudre la grille. Elle utilise également des méthodes privées pour valider les mouvements dans les lignes, colonnes et régions.

SudokuView : Cette classe représente la vue du jeu de Sudoku. Elle affiche la grille de Sudoku où les joueurs peuvent saisir les nombres et fournit des fonctionnalités telles que sauvegarder, vérifier et résoudre la grille. Elle utilise des contrôleurs pour gérer les actions effectuées par l'utilisateur.

VerifierController : Ce contrôleur gère les actions effectuées dans la vue de vérification. Il écoute les événements sur le bouton de vérification et affiche un message indiquant si la grille est valide ou non.
VerifierModel : Cette classe contient la logique pour vérifier si une grille de Sudoku est valide. Elle vérifie si les nombres dans les lignes, colonnes et régions sont uniques.
VerifierView : Cette classe représente la vue de vérification. Elle contient un bouton pour vérifier la grille de Sudoku. Elle fournit également des méthodes pour afficher la vue et accéder au bouton de vérification.
ChargerSauvegardeController : Ce contrôleur gère le chargement des données de la grille de Sudoku à partir d'un fichier. Il utilise un objet **JFileChooser** pour permettre à l'utilisateur de sélectionner un fichier de sauvegarde.
SauvegardeController : Ce contrôleur gère la sauvegarde des données de la grille de Sudoku dans un fichier. Il utilise également un objet **JFileChooser** pour permettre à l'utilisateur de sélectionner l'emplacement de sauvegarde.

Programme Joueur



ChargerSauvegardeController : Cette classe est responsable du contrôle du chargement des données de la grille de Sudoku à partir d'un fichier. Elle contient deux méthodes : **loadGridData** pour charger les données de la grille et **convertToDecimal** pour convertir un caractère en décimal.

Main : Cette classe contient la méthode principale **main**, qui est le point d'entrée du programme. C'est là que le programme démarre son exécution.

MenuController : Cette classe est un contrôleur pour la vue du menu. Elle implémente l'interface **ActionListener** pour gérer les actions des boutons du menu. Elle a une méthode **actionPerformed** pour réagir aux événements d'action.

MenuView : Cette classe représente la vue du menu. Elle crée et affiche les boutons du menu comme jouer, comment jouer et quitter. Elle a également des méthodes pour obtenir les boutons individuels et le cadre de la vue.

SudokuController : Ce contrôleur est responsable de la logique métier du Sudoku. Il écoute les actions de l'utilisateur sur la vue du Sudoku et met à jour le modèle en conséquence.

SudokuGrid : Cette classe représente la grille de Sudoku elle-même. Elle a des méthodes pour valider un mouvement, vérifier si la grille est pleine et résoudre la grille.

SudokuSolverModel : Cette classe est utilisée pour résoudre le Sudoku. Elle contient la logique de résolution du Sudoku.

SudokuView : Cette classe représente la vue du Sudoku. Elle affiche la grille de Sudoku et les contrôleurs associés. Elle gère également le chronomètre pour le temps écoulé pendant que l'utilisateur résout le Sudoku.

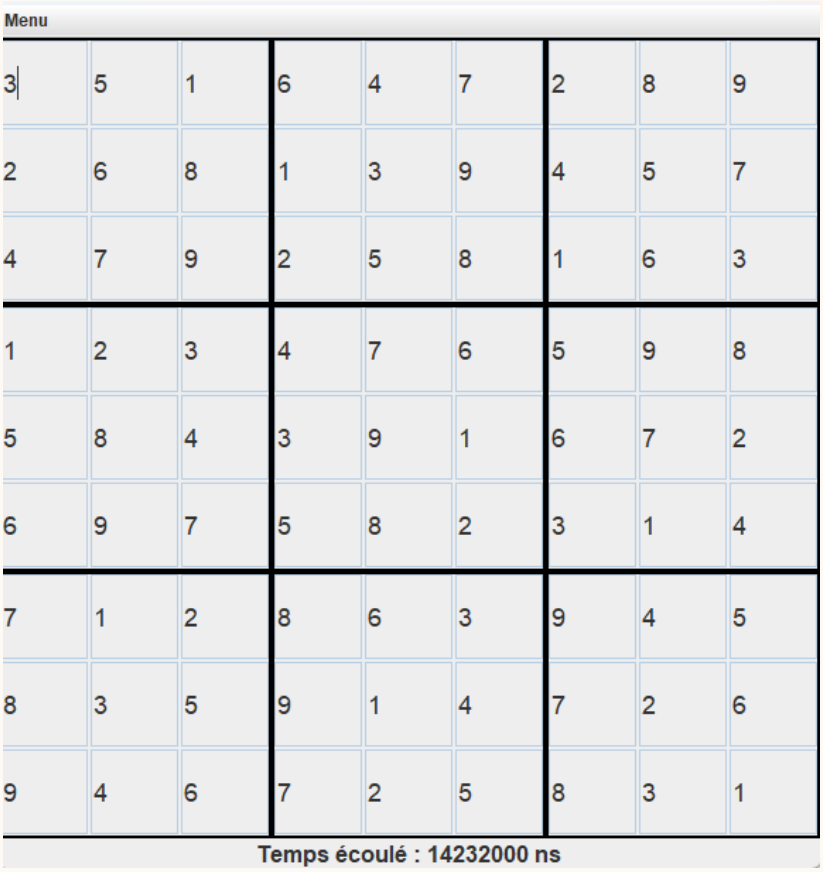
Timer : Cette classe est utilisée pour mesurer le temps écoulé. Elle peut être démarrée, arrêtée et interrogée sur la durée écoulée.
VerifierController : Ce contrôleur est responsable de la vérification de la solution du Sudoku. Il communique avec la vue associée pour afficher le résultat de la vérification.
VerifierModel : Cette classe contient la logique pour vérifier si une solution de Sudoku est valide.
VerifierView : Cette classe représente la vue utilisée pour afficher le résultat de la vérification de la solution du Sudoku.

Algorithme du solver

Avant de commencer la résolution, l'algorithme vérifie si la grille de Sudoku est valide. Cela signifie qu'il vérifie s'il n'y a pas de conflits dans les lignes, les colonnes et les régions. Si la grille n'est pas valide, l'algorithme renvoie false, indiquant que la résolution est impossible.

L'algorithme parcourt la grille pour trouver une case vide, c'est-à-dire une case contenant la valeur 0. Si toutes les cases sont remplies, cela signifie que le Sudoku est résolu avec succès, et l'algorithme renvoie true.

Une fois qu'une case vide est trouvée, l'algorithme commence à essayer de placer des chiffres dans cette case. Il teste les chiffres de 1 à 9, en vérifiant à chaque fois si le chiffre peut être légalement placé dans la case en respectant les règles du Sudoku : aucun chiffre identique dans la même ligne, colonne ou région.



Si un chiffre peut être légalement placé dans la case actuelle, l'algorithme procède à la récursion pour tenter de remplir les cases suivantes. Si à un moment donné aucune valeur ne peut être placée dans une case, cela signifie que le chiffre précédemment placé était incorrect. L'algorithme revient alors en arrière (backtracking) pour essayer une autre valeur dans la case précédente. Ce processus se poursuit jusqu'à ce que la grille soit remplie ou jusqu'à ce que toutes les possibilités aient été épuisées.

Une fois que le Sudoku est résolu avec succès, l'algorithme renvoie true et la grille est remplie de manière valide. Si aucune solution n'est trouvée, l'algorithme renvoie false, indiquant que le Sudoku est insoluble.

Conclusion Personnelle

« Cette expérience a été enrichissante pour moi. La conception de ce projet m’a tout d’abord de mener un travail de recherche. Le fait de faire des essais, d’échouer et par la suite de réussir m’a montré, je pense, une facette du métier de développeur. Travailler pendant un longtemps et réussir procure une satisfaction inexplicable et me donne envie de continuer dans cette voie. J’ai également apprécié travailler avec mon binôme. Nous avons entretenu une certaine forme de synergie pour ce projet et j’espère pouvoir à nouveau collaborer avec lui. »

Lenny FOULOU

"La réalisation de ce projet a été une aventure passionnante et enrichissante. Ce défi m'a permis de découvrir les bases du développement de jeux et de consolider mes compétences en programmation. Travailler en équipe a été une expérience gratifiante, où la collaboration et la résolution de problèmes ont été les clés. Cette expérience m'a inspiré à poursuivre dans cette voie et à approfondir mes connaissances.”

Hamza Bazzaou