

# **RAPPORT**

## **Création d'un Sudoku**

### **Sommaire**

Partie 1 : Introduction du jeu

Partie 2 : Description des fonctionnalités

Partie 3 : Présentation de la structure

Partie 4 : Explication de l'algorithme

Partie 5 : Conclusion personnel

# Introduction

Le Sudoku est un jeu qui repose sur la logique et la réflexion. Le jeu se déroule sur une grille de 9x9 cases, divisée en neuf grilles de 3x3 cases. Le Sudoku vise à remplir chaque case de la grille avec un chiffre de 1 à 9, de manière à ce que chaque ligne, chaque colonne et chaque grille de 3x3 cases contiennent une seule fois tous les chiffres. Le jeu commence par certaines cases de la grille préremplies de chiffres, ce qui crée une configuration initiale, ou en créant votre propre grille. D'après ce schéma, le joueur doit faire appel à la logique pour déterminer les chiffres à placer dans les cases vides pour compléter la grille en respectant les règles du Sudoku. Le Sudoku est une activité intellectuelle captivante qui ne requiert pas de compétences mathématiques avancées, mais plutôt une réflexion logique et méthodique afin de trouver la solution originale à chaque puzzle.

# Descriptions des fonctionnalités

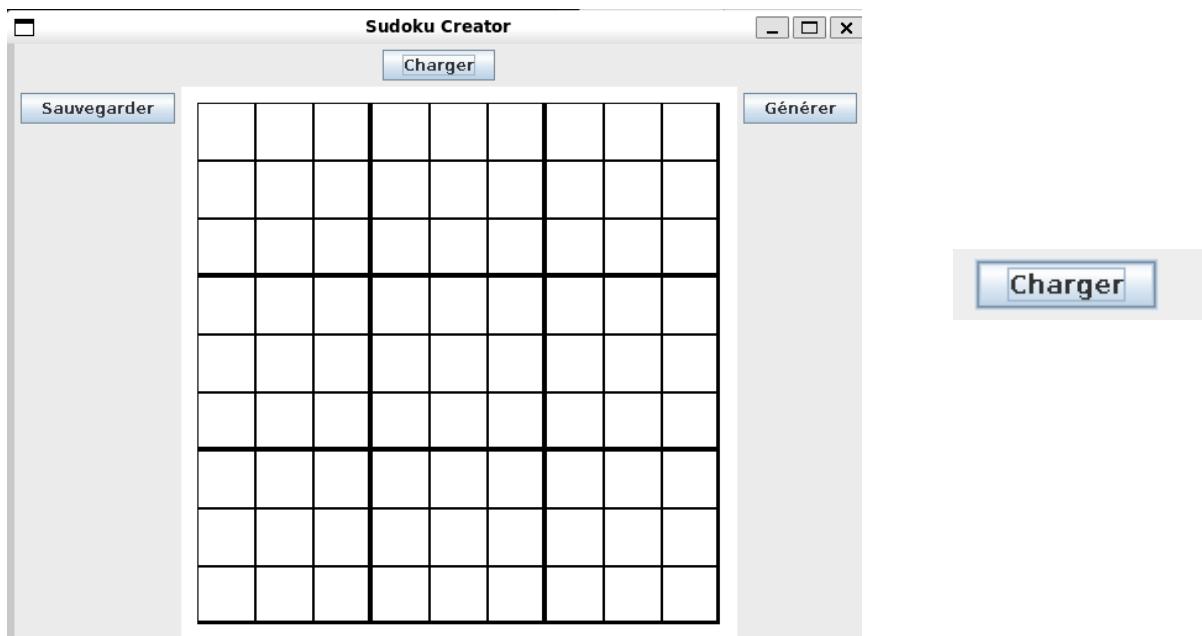
Le jeu Sudoku est composé de deux programmes qui eux ont différentes fonctionnalités.

## Premier programme :

Notre premier programme permet de créer ou ouvrir une grid et d'ajouter des chiffres ou d'en supprimer puis de sauvegarder le fichier grille.

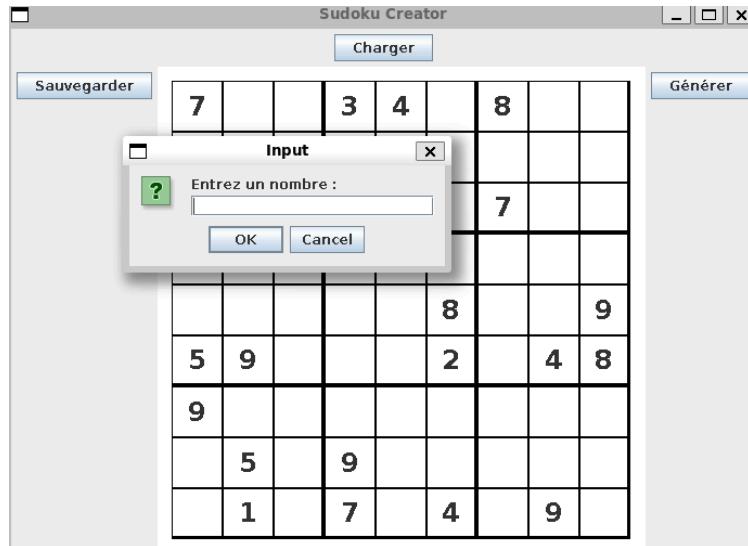
### Choix de grille :

Lorsque que nous lançons notre premier programme, on arrive sur une page avec une grille vide ou nous avons la possibilité de charger une grille.



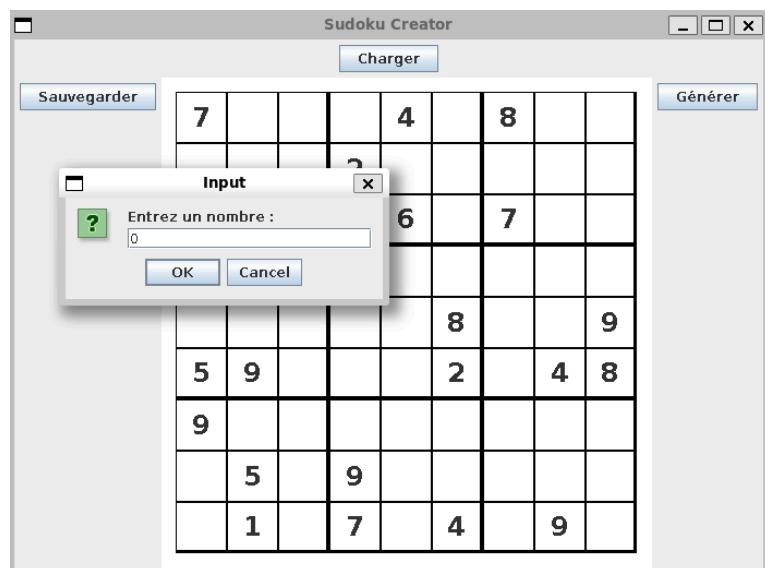
## Ajout de chiffres dans les cases :

Sur la page on peut ajouter en cliquant sur la case en question ,vous pouvez confirmer ou annuler l'ajout si on essaye d'ajouter un chiffre qui contraint les règles d'unicité cela mettra en rouge le chiffre dans la case.



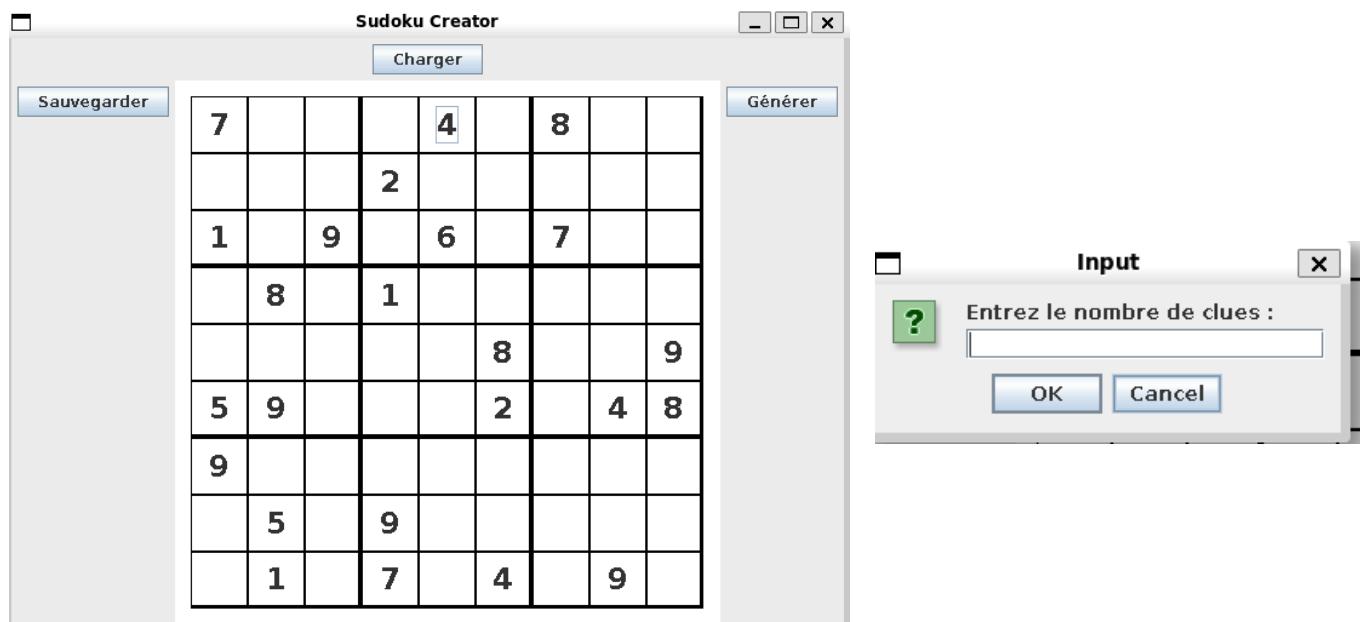
## Suppression des chiffre dans les cases :

Sur la page, on peut ajouter en cliquant sur la case en question. Vous pouvez confirmer ou annuler l'ajout. Si on essaie d'ajouter un chiffre qui contredit les règles d'unicité, cela mettra en rouge le chiffre dans la case.



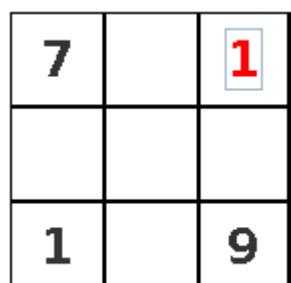
## Génération de grille automatique :

Il y a aussi une fonctionnalité de génération automatique de grille qui remplit automatiquement la grille tout en respectant les contraintes d'unicité en appuyant sur le bouton “Générer”.



## Fonctionnalité d'unicité :

Cette fonctionnalité permet de mettre en rouge si l'ajout d'un chiffre enfreint la contrainte d'unicité en vérifiant s'il est déjà présent dans la ligne, la colonne et la région concernées.



## Sauvegarde :

Après avoir créé, modifié ou encore généré une grille, on peut sauvegarder la grille afin de l'utiliser dans notre second programme grâce au bouton "Sauvegarde".

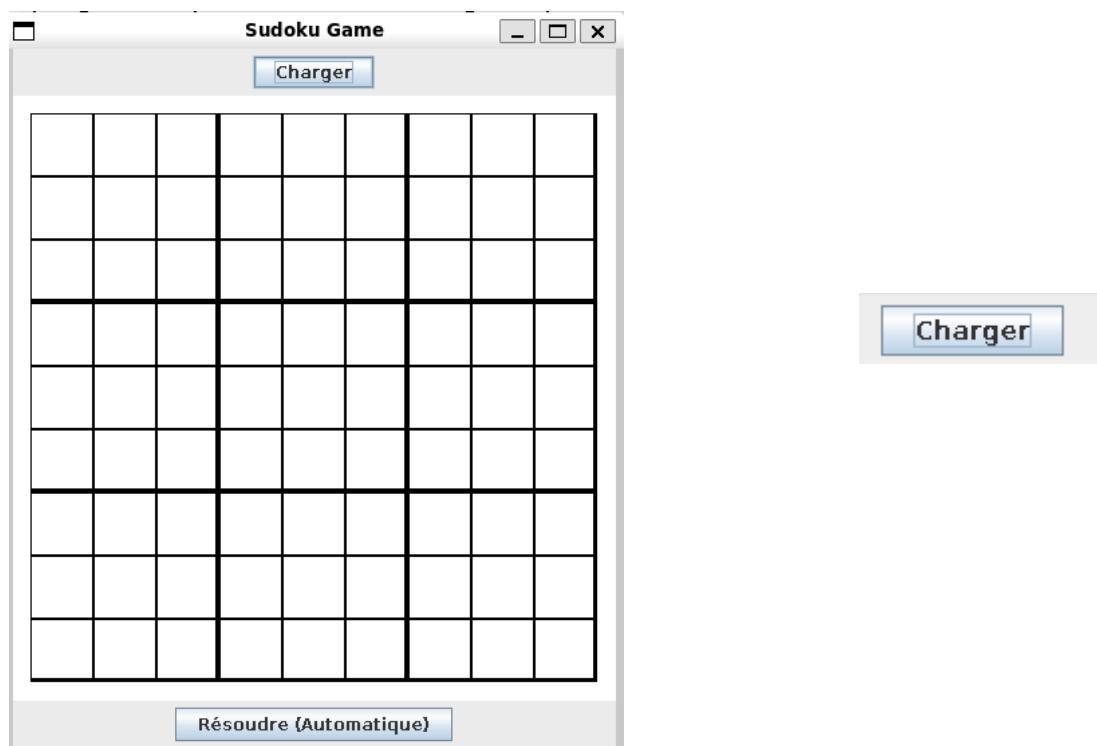


## Second programme :

Notre second programme permet d'ouvrir une grille créée avec notre premier programme. Il permet de choisir entre un mode automatique ou manuel. En mode manuel, on peut ajouter et supprimer des chiffres dans les cases, mais on ne peut pas supprimer les chiffres qui étaient déjà sur la grille. En mode automatique, on peut résoudre automatiquement la grille et voir le temps pris pour la résoudre.

### Ouverture de fichier :

Au lancement du programme, nous devrons ouvrir un fichier contenant une grille que nous aurons créée grâce au premier programme. Cette grille nous permettra de jouer plus tard au jeu.



## Choix entre manuel et automatique :

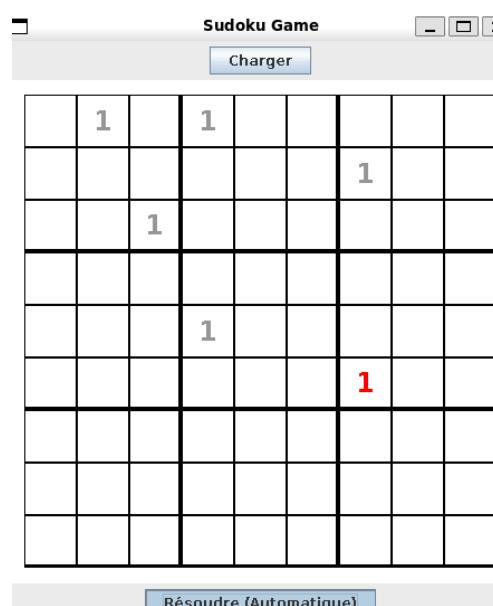
Après avoir ouvert un fichier contenant une grille nous arrivons sur une page qui nous offre de choisir le mode manuel et le mode automatique :

### Mode manuel :

Lorsqu'on appuie sur le bouton "Manuel" cela nous ouvre une page sur laquelle il y a une grille avec plusieurs fonctionnalités tels que :

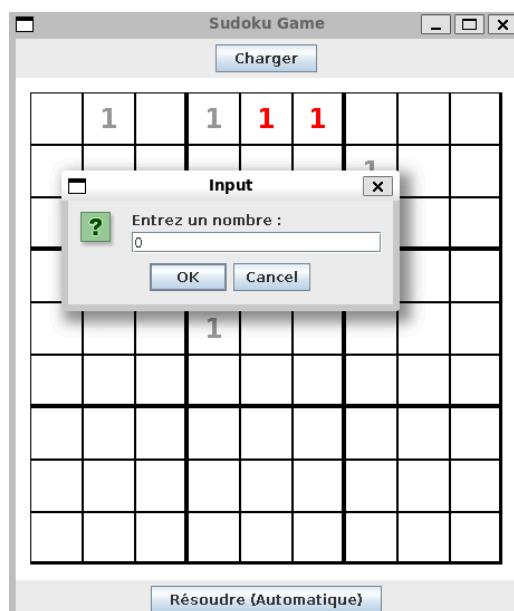
### Ajout de chiffres dans les cases :

Puis on arrive sur une page avec une grille où l'on peut ajouter en cliquant sur la case en question, vous pouvez confirmer ou annuler l'ajout si on essaye d'ajouter un chiffre qui contraint les règles d'unicité cela met en rouge les cases qui enfreint les contraintes d'unicité contrairement au premier programme vous ne pouvez pas modifier ce qui est déjà grillé chargé.



### Suppression des chiffre dans les cases :

Sur cette page on peut aussi supprimer les chiffres en cliquant sur le chiffre est en mettant 0, vous pouvez confirmer ou annuler la suppression, contrairement au premier programme vous ne pouvez pas supprimer ce qui est dans la grille chargé.

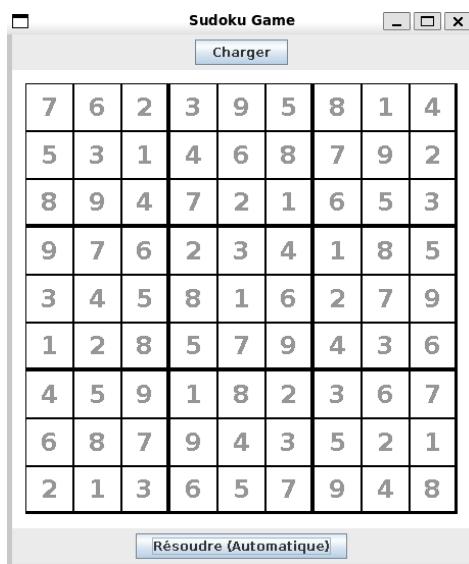


### Mode automatique :

Lorsqu'on appuie sur le bouton “automatique” cela nous ouvre une page sur lequel il y a une grille avec plusieurs fonctionnalité tels que :

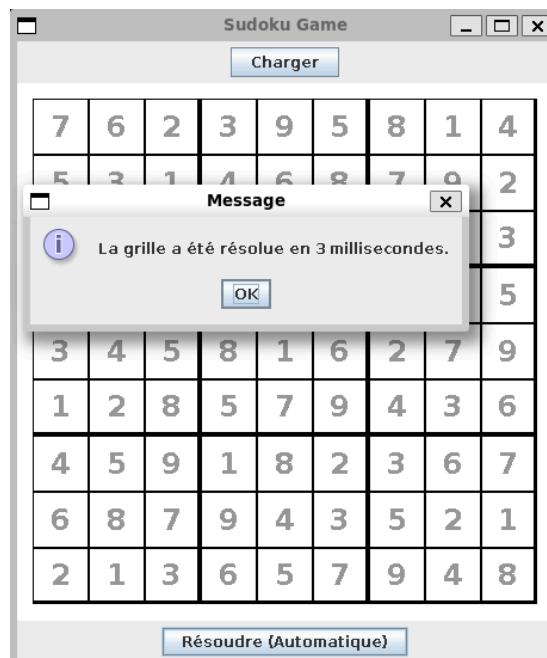
## Résoudre la grille :

Grâce à cette fonctionnalité, vous pourrez résoudre la grille automatiquement grâce à un bouton “Résoudre”, cela permettra de trouver la solution à la grille.



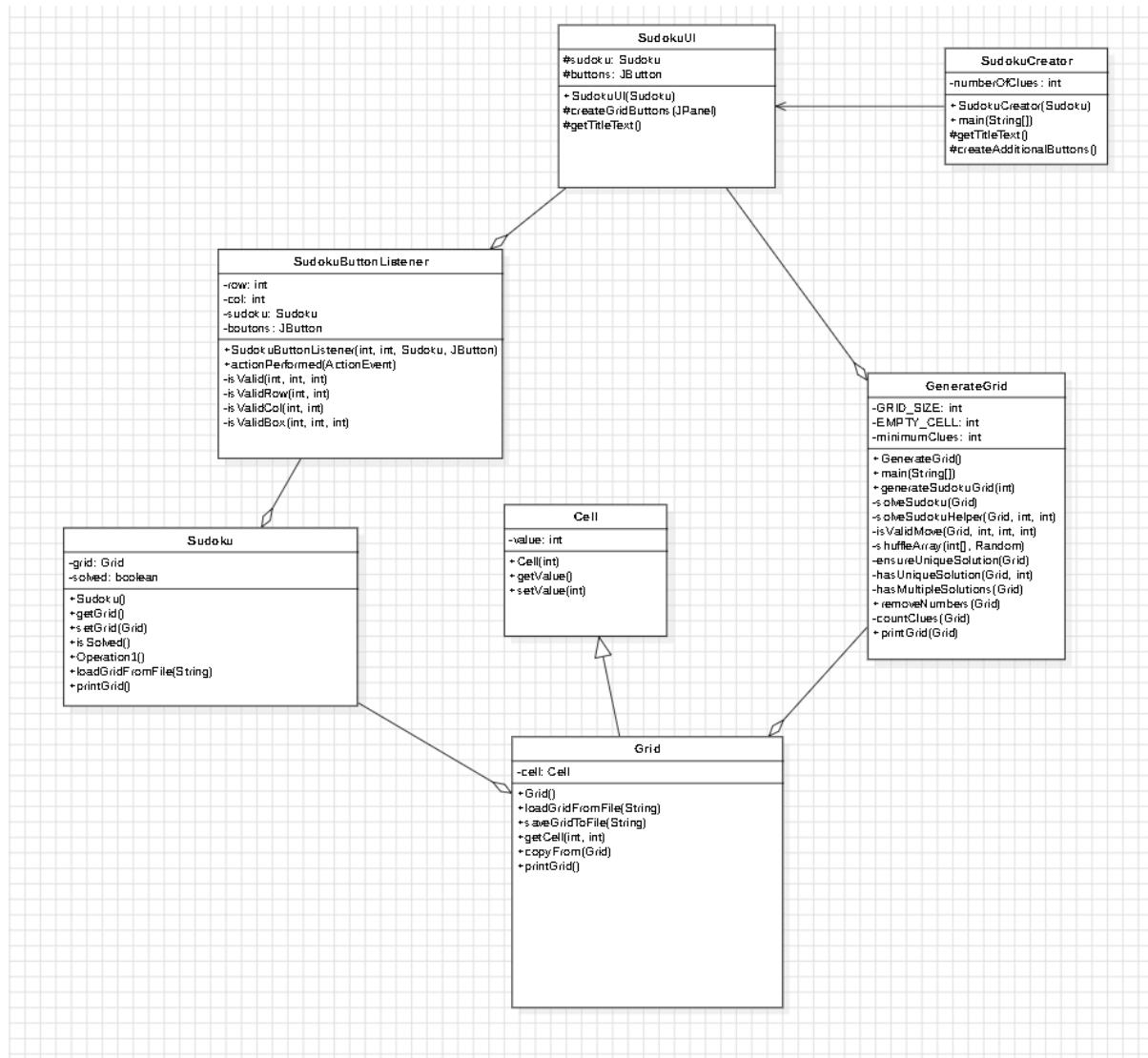
## Temps à résoudre :

Vous pourrez aussi voir le temps que la fonctionnalité a pris pour résoudre la grille.

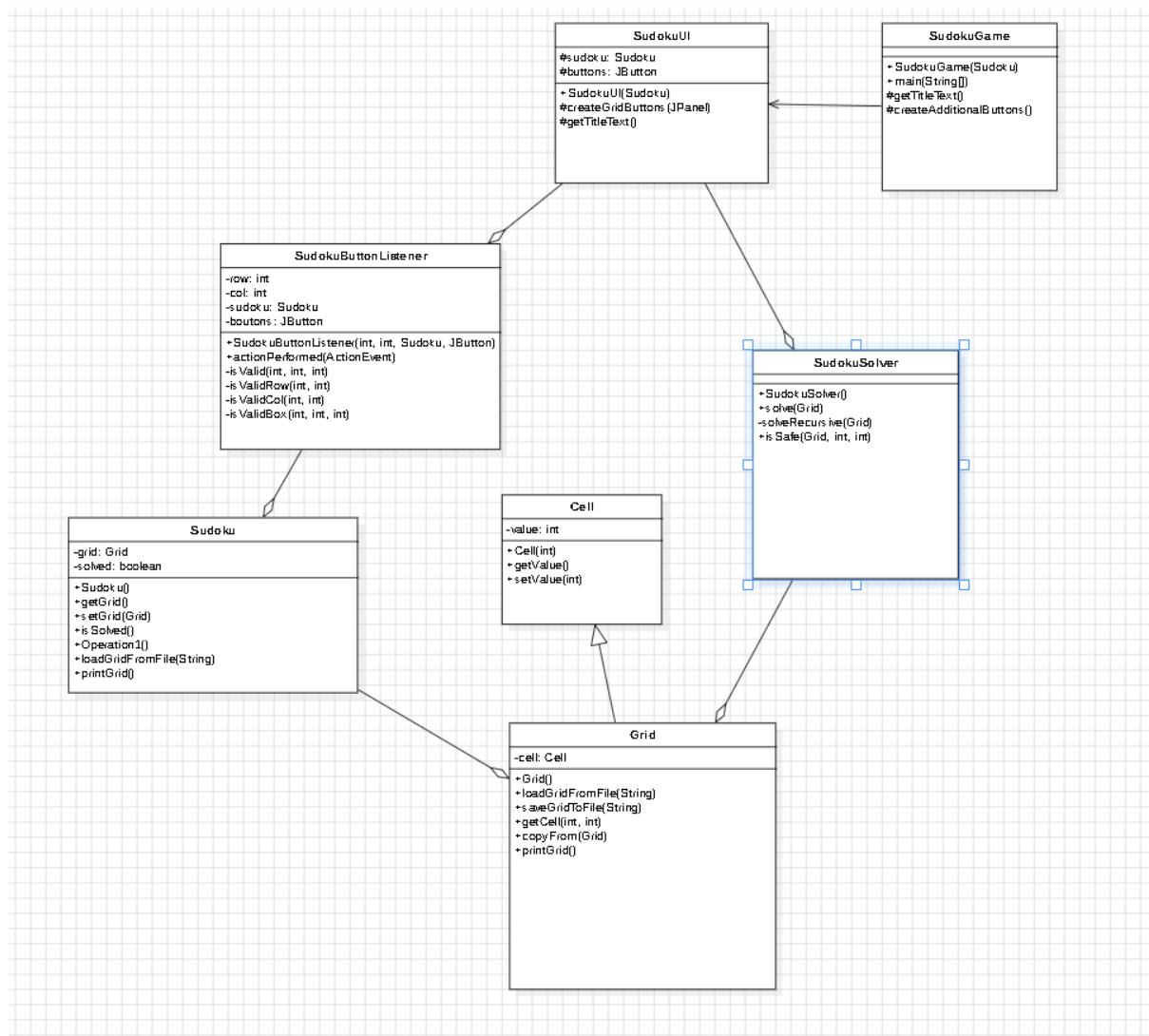


# Présentation de la structure

Voila le diagramme de classe de notre premier programme :



et voila celui de deuxième programme :



Une légère description des Classe :

### Classe Cell :

Notre classe Cell représente une case dans la grille, elle permet de définir et récupérer les valeurs dans les cases de notre grille. Elle permet aussi d'obtenir la valeur d'une case. Elle est utilisée dans toutes les Classes.

### Classe Grid :

La classe Grid représente une grille, elle permet d'initialiser une grille vide, elle sert aussi d'ouvrir une grille depuis un fichier et sa sauvegarde, elle permet aussi de copier une grille ou encore d'afficher la grille. Nous avons fait le choix de mettre toutes ces fonctions dedans car elles ont toutes un lien avec la grille, ça nous semblait logique. Elle est utilisée dans toutes les classes sauf Cell.

### Classe GenerateGrid :

La classe GenerateGrid utilise une grille vide ou chargée pour générer une grille de Sudoku, avec la possibilité de choisir le nombre de case à remplir tout en respectant les conditions d'unicité en les vérifiant. Cette classe est uniquement utilisée dans le programme deux qui est SudokuCreator. Elle colorie aussi en rouge les cases qui ne respectent pas les contraintes d'unicité.

### Classe SudokuSolver :

La class SudokuSolver permet de à partir d'une grille donné de résoudre la grille en vérifiant les conditions toutes les contraintes d'unicité et calculant le temps pris pour la résoudre. Elle utilise une approche de récursion avec backtracking pour trouver la solution. Cette classe est uniquement utilisé dans le programme deux qui est SudokuGame

### Classe Sudoku :

La classe Sudoku représente un jeu de Sudoku avec une grille associée. Elle fournit des méthodes pour interagir avec la grille, telles que le chargement à partir d'un fichier, l'impression de la grille et la vérification de l'état de résolution.

### Classe SudokuButtonListener :

La classe SudokuButtonListener écoute les événements des boutons dans une interface utilisateur de Sudoku. Lorsqu'un bouton est cliqué, cette classe affiche une boîte de dialogue demandant à l'utilisateur d'entrer un nombre, puis met à jour la grille de Sudoku en conséquence. Elle vérifie si le nombre pour l'ajout est bien un chiffre au contraire ça affichera rien. Ainsi que tous les fonctionnalité comme ajout ou encore la suppression de case dans la grille.

### Classe SudokuUI :

La classe abstraite SudokuUI est une interface graphique pour afficher et interagir avec une grille de Sudoku. Elle fournit les méthodes de base pour la création de l'interface utilisateur.

Nos deux Main classe : SudokuCreator et SudokuGame :

SudokuCreator : La classe SudokuCreator permet tout simplement de lancer notre premier code.

SudokuGame : La classe SudokuGame permet tout simplement de lancer notre premier code.

Je tiens à le préciser SudokuGame utilise pas la classe GenerateGrid et quant a SudokuCreator il n'utilise pas la classe SudokuSolver car il n'ont pas besoin de leur utilisation.

# Explication de l'algorithme

L'algorithme qui permet de résoudre et trouver se nomme “solveRecursive” la solution à la grille utilise tout d'abord une autre fonctionnalité “isSafe” qui nous permet de vérifier les contraintes d'unicités afin de le remplir en vérifiant que sur la ligne le chiffre est unique, de même pour la colonne et aussi pour la région (grille 3x3). Puis il essaye plus possible jusqu'à ce qu'il teste les possibilités si la grille est insoluble le programme renvoie false et true si elle a été résolue. Puis dans notre main pour le programme qui résout donc “SudokuGame” ont créé une deuxième grille qui copie la grille charger pour le jeu si le résolveur arrive à résoudre la copie cela affichera sur la vraie grille la grille résolue sinon cela renverra un message disant que la grille est insoluble. Pour l'utiliser dans notre programme il faut tout simplement appuyer sur le Résoudre dans le programme 2.

## Conclusion personnel

Vincent : Je suis vraiment fier de ce que nous avons réalisé avec ce sudoku en Java. C'était ma première fois à coder un jeu sans utiliser de sources externes ni de bouts de code déjà existants. Ça a été un défi, mais je suis content de voir le résultat final. Bien sûr, il y a eu quelques heures passées à résoudre des problèmes qui auraient pu être évités si j'avais été plus attentif à l'énoncé dès le départ. Mais malgré le manque d'une fonctionnalité, je suis satisfait du travail accompli. Je pense que c'est une bonne expérience d'apprentissage pour l'avenir.

David : J'ai beaucoup aimé faire des projets. J'ai trouvé la phase de compréhension des fonctionnalités assez perplexe du fait de sa complicité, de plus en ayant découvert le jeu Sudoku 1 mois avant la sae je me sentais assez serein sur la difficulté du code en pensant prendre pas énormément de temps à la réussite du projet mais je me suis trompé. A travers ce projet je suis souvent resté bloqué sur des erreurs d'inattention mais malgré cela le projet de d'apprendre plus en java et aussi d'apprendre à faire le Makefile pour du java. J'ai pensé que ça ma encore une fois était une bonne expérience de travail d'équipe et mais pour notre prochain projet on prendra l'initiative de commencer plus tôt.

