

### Sommaire :

- Introduction
- Description des fonctionnalités
- Présentation de la structure de chaque programme
- Explication de l'algorithme qui résout les grilles
- Conclusion

### | ) Introduction :

Dans le cadre de notre 1 ère année de BUT Informatique, notre professeur de développement nous donne un projet à réaliser qui consiste à créer le fameux jeu du Sudoku, voici une brève description du jeu :

Le Sudoku est un jeu de logique et de réflexion qui se joue sur une grille de 9x9 cases, divisée en neuf sous-grilles de 3x3 appelées régions. Le but du jeu est de remplir chaque ligne, chaque colonne et chaque région avec les chiffres de 1 à 9, sans répéter aucun chiffre dans une même ligne, colonne ou région. Au départ, une grille partielle est donnée avec quelques chiffres déjà remplis. Le joueur doit compléter la grille en respectant les règles mentionnées précédemment.

Pour mener à bien ce projet, nous devons réaliser ce projet avec les contraintes suivantes :

- Produire un programme qui servira à l'élaboration des grilles de départ :
  - On partira d'une grille vide ou charger une grille existante depuis un fichier.
  - Le joueur peut choisir de concevoir sa propre grille (en évitant les placements non conformes aux règles du Sudoku)
  - Le joueur sauvegarde / charge les grilles par le biais de la class [JFileChooser](#)

- Un deuxième programme qui va se charger de la résolution du grille :
  - On commence par charger une grille depuis un fichier.
  - Décider si on souhaite la résoudre manuellement ou automatiquement.

## II ) Description des fonctionnalités :

### Fonctionnalités du Programme 1 :

On lance le programme 1 à l'aide de la commande make run1.

Nous avons un bouton qui permet de Charger une grille :



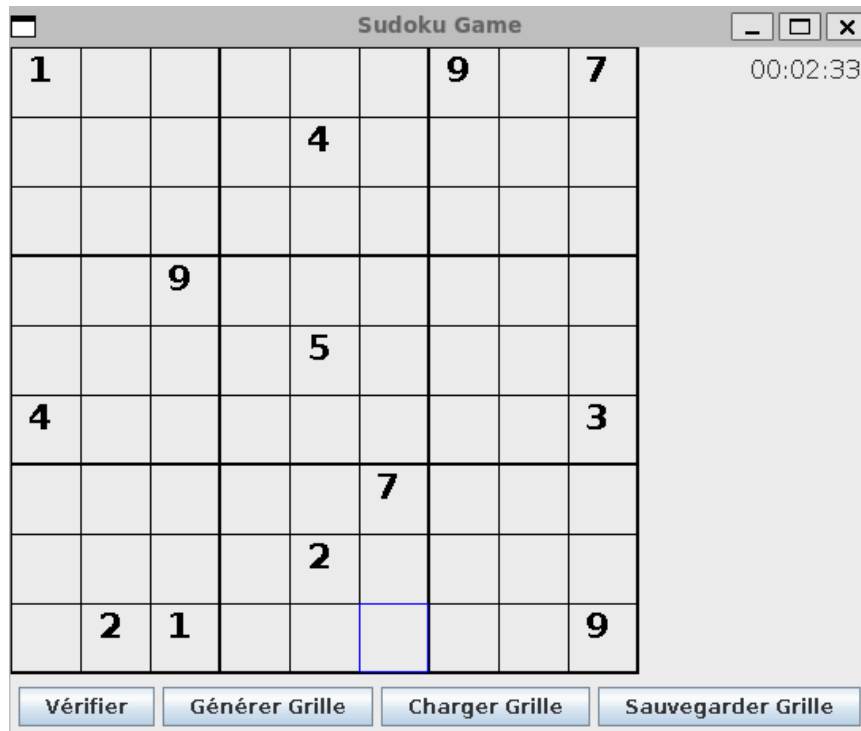
Une fois que ce bouton est cliqué, le joueur navigue dans un système de fichiers pour pouvoir choisir d'importer une grille à partir des fichiers .gri.



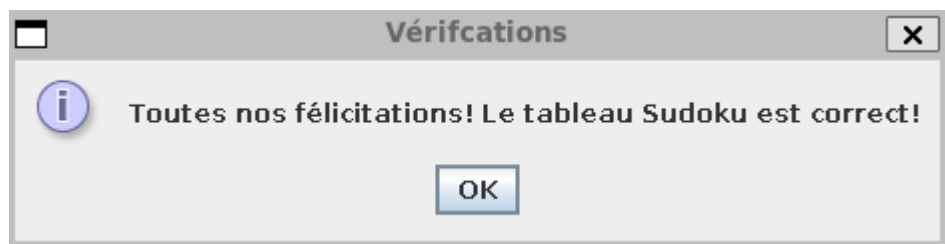
Dès que le joueur sélectionne le exemple.gri et qu'il clique sur "Open", il a sa grille directement sur le plateau :

				9	5			4
5	3		4		8	7		2
			7			6		3
9				3	4		8	
	4			1			7	
	2		5	7				6
4		9			2			
6		7	9		3		2	1
2			6	5				

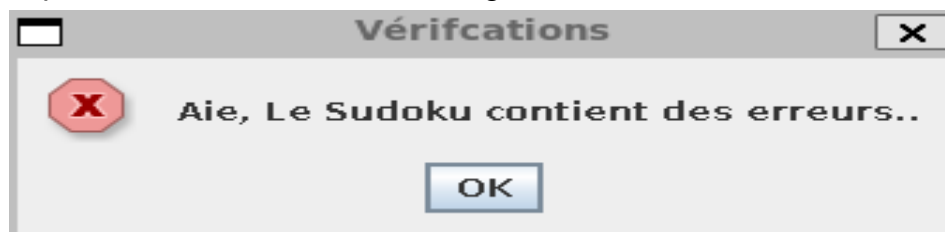
Le joueur peut aussi créer sa propre grille, pour cela il a tout simplement à mettre des chiffres où il désire puis clique sur le bouton “Sauvegarder Grille”, il a également un bouton Vérifier, pour éviter les placements contradictoire dans la mise en conception de sa grille :



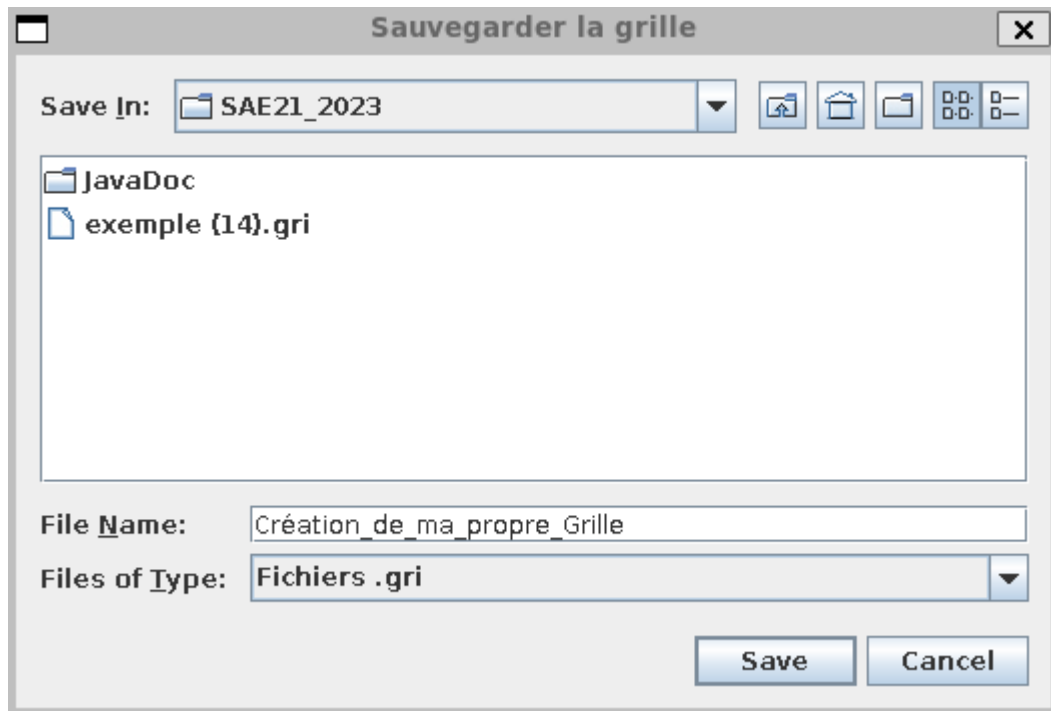
Si placement conforme aux règles du Sudoku :



Si placement nonconforme aux règles du Sudoku :



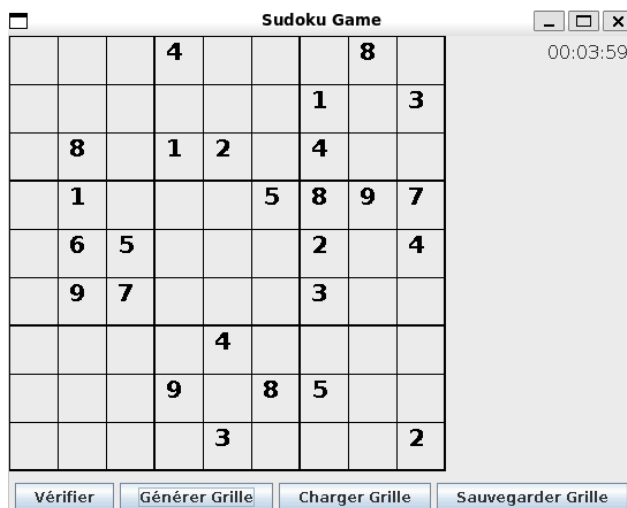
Une fois que le joueur souhaite sauvegarder sa propre grille il a juste à cliquer sur “Sauvegarder Grille” et un système de fichiers s’ouvrent comme pour le chargement des grilles :



Nous avons également ajouter une fonctionnalité qui permet de générer une grille aléatoirement (éligibles aux règles du Sudoku bien sûr), pour cela suffit simplement de cliquer sur “Générer Grille” :

**Générer Grille**

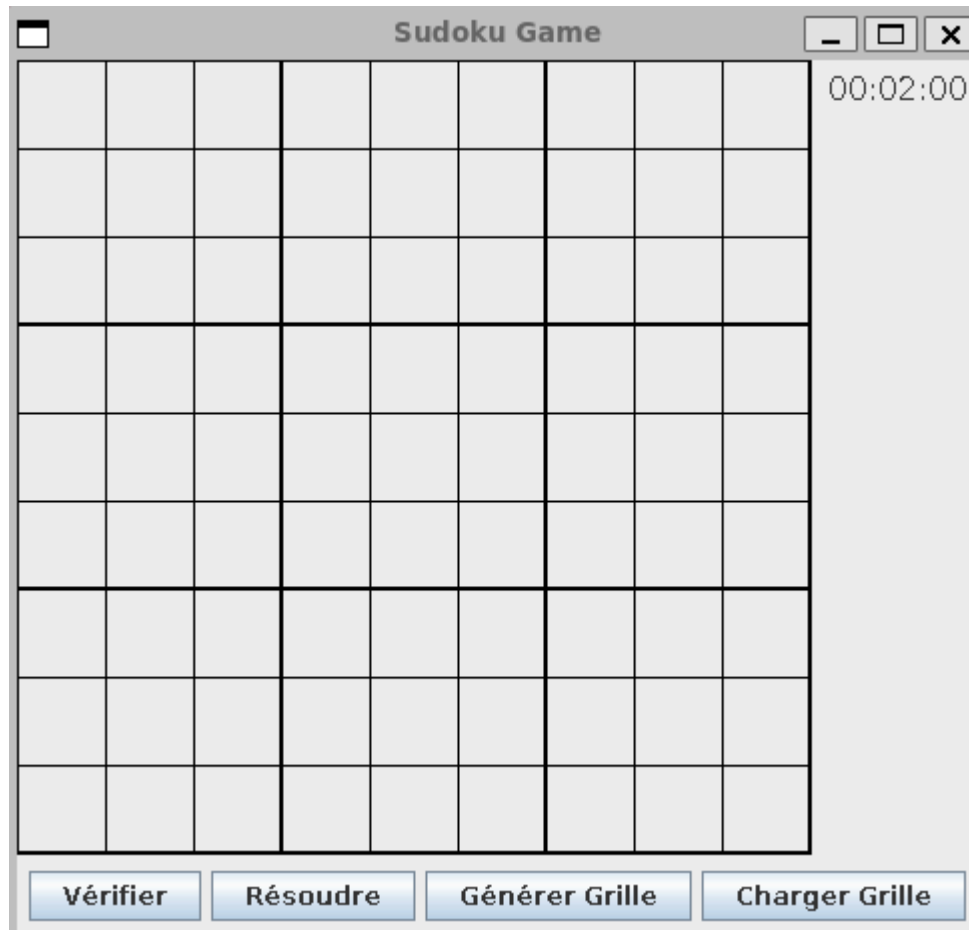
Et on obtient une multitude de Grille de départ qu’on peut également sauvegarder pour pouvoir la charger dans le deuxième programme.



## Fonctionnalités du Programme 2 :

On lance le programme 2 à l'aide de la commande make run2.

Une fois la commande lancée on tombe sur interface comme celle ci-dessous.



On a donc le choix encore une fois de "Charger Grille" ou "Générer Grille" :



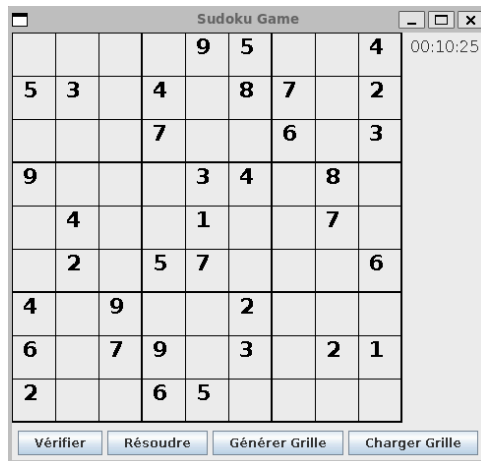
Une fois une des deux options choisies, le joueur peut enfin commencer à jouer au Sudoku, pour cela il a le choix de sélectionner des cases avec la souris ou bien avec les touches directionnelles, tout dépend ce avec quoi il est plus à l'aise !

Le joueur est contraint de choisir uniquement des chiffres entre 1 et 9, si le joueur souhaite supprimer un chiffre d'une case, il remplace le chiffre en question par un 0.

Nous possédons également un bouton “Résoudre” qui est destiné aux personnes qui donnent leur langues aux chats ou encore pour faciliter la correction du professeur.



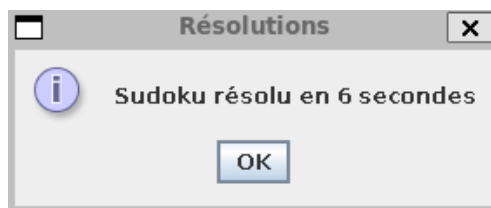
Grille avant la résolution :



Pour rendre le jeu un minimum réaliste, nous avons décidé que la résolution de la grille se fera sur une durée de 5 à 10 secondes.

Ce qui se passe quand je clique sur “Résoudre” :

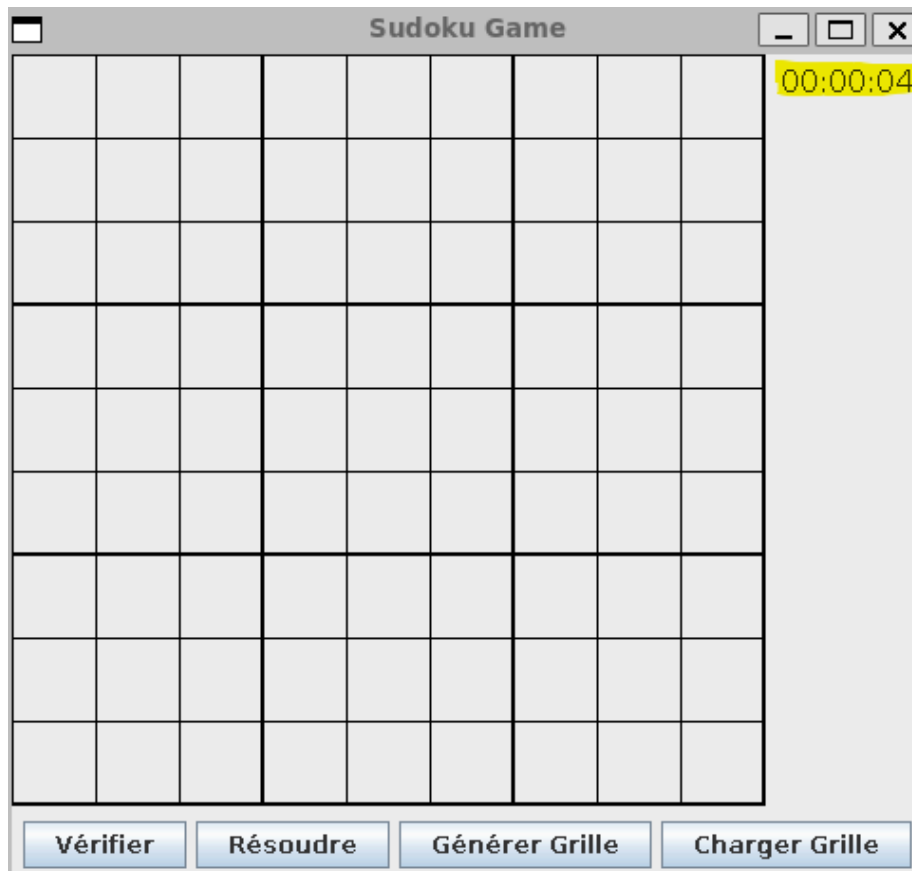
La fenêtre suivante s’ouvre :



Avec la grille entièrement résolu :

7	6	2	3	9	5	8	1	4
5	3	1	4	6	8	7	9	2
8	9	4	7	2	1	6	5	3
9	7	6	2	3	4	1	8	5
3	4	5	8	1	6	2	7	9
1	2	8	5	7	9	4	3	6
4	5	9	1	8	2	3	6	7
6	8	7	9	4	3	5	2	1
2	1	3	6	5	7	9	4	8

Nous avons également ajouté un Timer pour pouvoir permettre au joueur d'évaluer ses performances et pour qu'il s'améliore au fil du temps. Ce Timer se situe en haut à droite de la fenêtre swing.



Le bouton vérifier a deux options :

Si grille valide quand je clique sur "Vérifier" :



Sudoku Game									
1	2	3	4	5	6	7	8	9	
4	5	6	7	8	9	1	2	3	
7	8	9	1	2	3	4	5	6	
2	1	4	3	6	5	8	9	7	
3	6	5	8	9	7	2	1	4	
8	9	7	2	1	4	3	6	5	
5	3	1	6	4	2	9	7	8	
6	4	2	9	7	8	5	3	1	
9	7	8	5	3	1	6	4	2	

Vérifier Résoudre Générer Grille Charger

si je modifie le 5 au dessus du bouton “Résoudre” par un 4 et que je vérifie :

Vérifications

X

Aie, Le Sudoku contient des erreurs..

OK

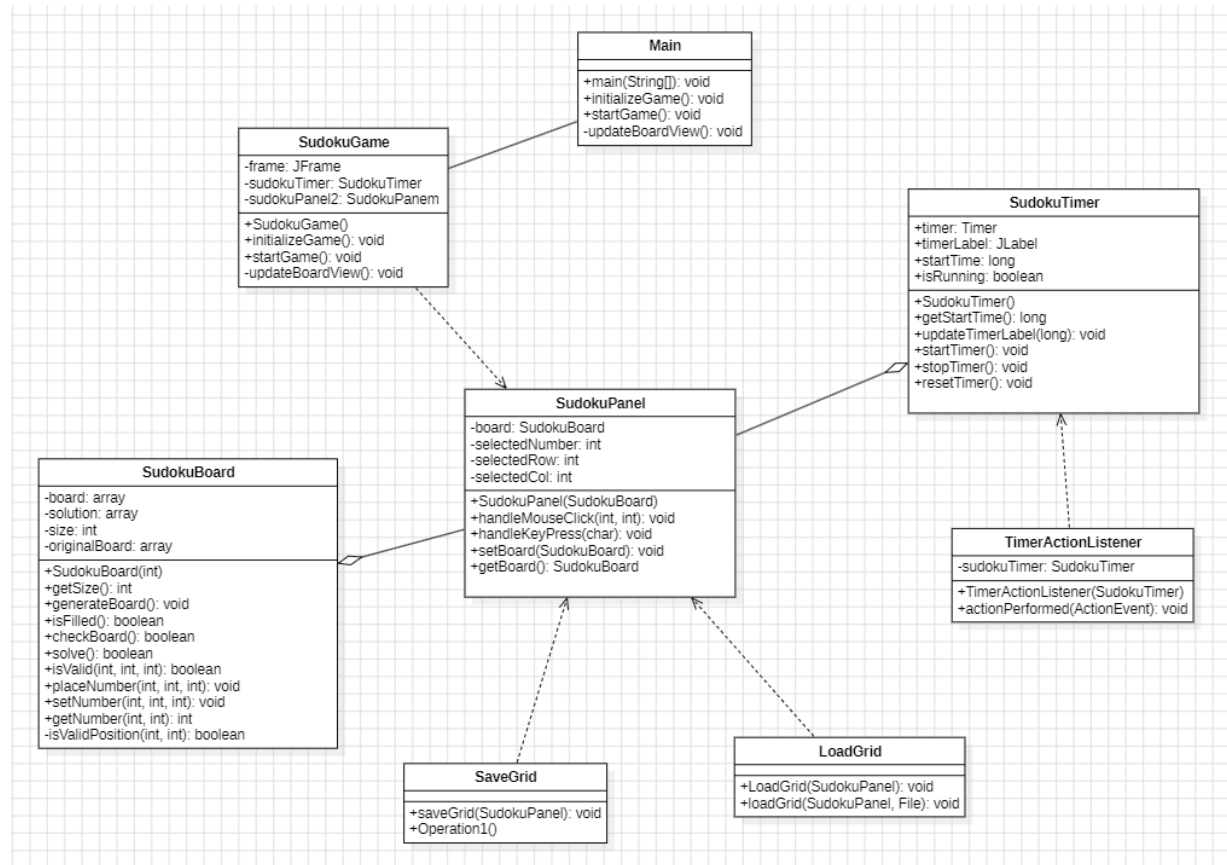
Sudoku Game									
1	2	3	4	5	6	7	8	9	
4	5	6	7	8	9	1	2	3	
7	8	9	1	2	3	4	5	6	
2	1	4	3	6	5	8	9	7	
3	6	5	8	9	7	2	1	4	
8	9	7	2	1	4	3	6	5	
5	3	1	6	4	2	9	7	8	
6	4	2	9	7	8	5	3	1	
9	7	8	4	3	1	6	4	2	

Vérifier Résoudre Générer Grille Charger



### III ) Présentation de la structure de chaque programme :

Pour mieux visualiser et comprendre le programme 1, nous avons conçu le diagramme de classe suivant :



Voici les explications de chaque classe du programme 1 :

**SudokuPanel** : Cette classe est responsable de l'affichage du plateau de Sudoku et de la gestion des interactions utilisateur telles que la sélection de cases, la saisie de nombres au clavier et le dessin des chiffres sur le plateau. Elle sert d'interface entre le plateau de jeu et l'interaction utilisateur.

**SudokuBoard** : Cette classe représente le tableau de jeu du Sudoku, avec des méthodes pour générer un nouveau tableau, résoudre le Sudoku, vérifier sa validité et interagir avec les cases du tableau.

**SaveGrid** : Méthode statique qui permet de sauvegarder la grille Sudoku dans un fichier. Elle ouvre une fenêtre de dialogue pour choisir l'emplacement de sauvegarde, puis parcourt la grille et écrit les nombres dans un fichier avec l'extension ".gri".

**LoadGrid** : La classe LoadGrid facilite le chargement de grilles de Sudoku à partir de fichiers pour les afficher dans un panneau de Sudoku.

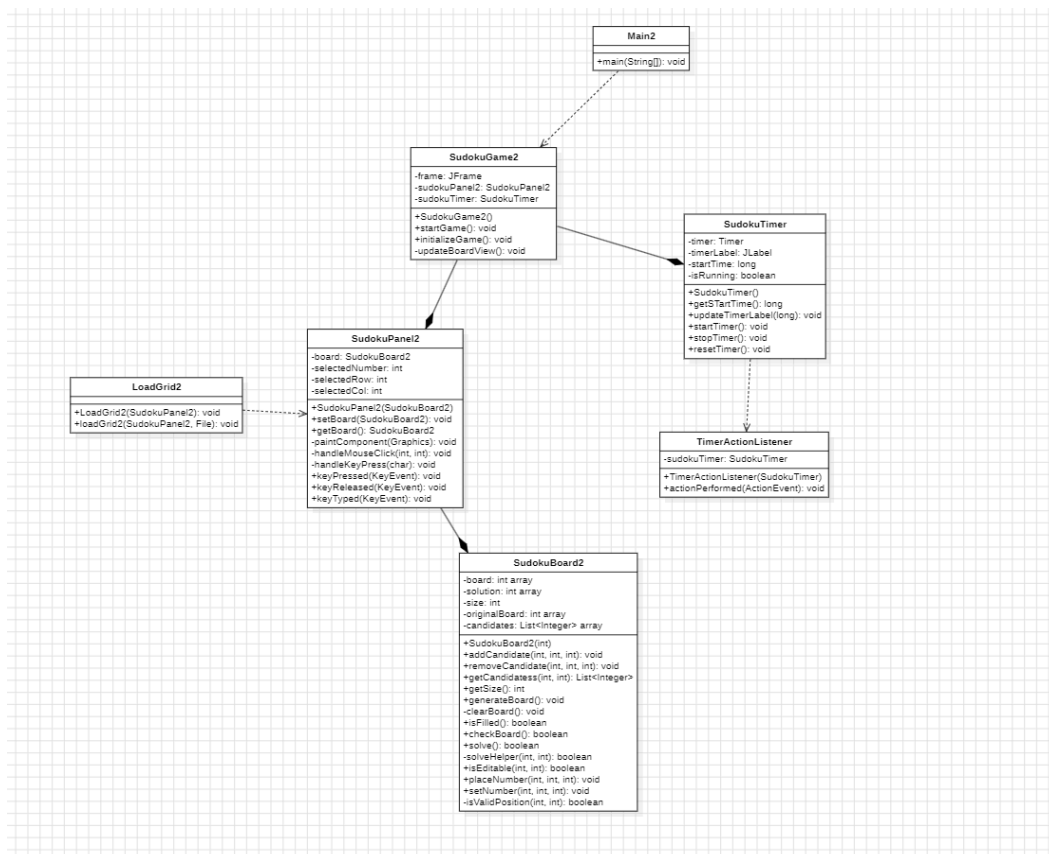
**TimerActionListener** : La classe TimerActionListener est un écouteur d'événements qui réagit aux événements déclenchés par un timer. Elle est utilisée pour mettre à jour l'heure affichée dans un composant SudokuTimer.

**SudokuTimer** : la classe SudokuTimer fournit un composant graphique simple pour afficher le chronomètre dans le jeu Sudoku, avec des méthodes pour démarrer, arrêter et réinitialiser le chronomètre.

**SudokuGame** : La classe SudokuGame est le cœur du jeu Sudoku. Elle gère l'interface utilisateur, les interactions avec l'utilisateur, et l'intégration des différentes fonctionnalités du jeu. Voici une explication détaillée de ses composants et fonctionnalités.

**Main** : La classe Main est la classe principale de l'application du jeu Sudoku. Elle contient la méthode main, qui est le point d'entrée de l'application.

Pour mieux visualiser et comprendre le programme 2, nous avons conçu le diagramme de classe suivant :



**SudokuBoard2** : La classe SudokuBoard2 représente un tableau de Sudoku et gère diverses opérations telles que la génération, la résolution et la vérification du tableau de Sudoku. Voici une explication de ses principales fonctionnalités.

**SudokuPanel2** : La classe SudokuPanel2 est un panneau qui affiche le plateau de jeu Sudoku et gère les interactions avec l'utilisateur, notamment la sélection d'une case, la saisie de chiffres via le clavier et l'affichage visuel du Sudoku. Voici une explication de ses principales fonctionnalités

**LoadGrid2** : La classe LoadGrid2 est utilisée pour charger une deuxième grille de Sudoku à partir d'un fichier. Voici une explication de son fonctionnement.

**TimerActionListener** : La classe TimerActionListener est un écouteur d'événements qui réagit aux événements déclenchés par un timer. Elle est utilisée pour mettre à jour l'heure affichée dans un composant SudokuTimer.

**SudokuTimer** : la classe SudokuTimer fournit un composant graphique simple pour afficher le chronomètre dans le jeu Sudoku, avec des méthodes pour démarrer, arrêter et réinitialiser le chronomètre.

**SudokuGame2** : La classe SudokuGame2 gère l'affichage du jeu Sudoku dans une interface graphique Swing, ainsi que les interactions avec l'utilisateur et les fonctionnalités du jeu. Voici une explication de son fonctionnement :

**Main2** : La classe Main2 est utilisée pour lancer une deuxième instance de l'application du jeu Sudoku. Elle crée une instance de SudokuGame2 et initialise le jeu en appelant sa méthode initializeGame().

#### **IV ) Explication de l'algorithme qui résout les grilles :**

L'algorithme utilisé pour résoudre les grilles de Sudoku repose sur une technique de backtracking. Ce processus récursif commence par placer un chiffre valide dans une case vide du tableau. Il teste ensuite chaque possibilité pour chaque case, en avançant de manière récursive. Lorsqu'il rencontre une impasse, c'est-à-dire qu'il ne peut pas placer un chiffre valide dans une case donnée, il revient en arrière (backtrack) pour essayer une autre option. Ce processus se poursuit jusqu'à ce qu'il trouve une solution valide pour l'ensemble du tableau ou qu'il explore toutes les possibilités sans succès. Cette approche est efficace pour résoudre des problèmes complexes comme le Sudoku, où il existe de multiples choix à chaque étape et où la validation de la solution est essentielle à chaque niveau de la récursion.

Nous utilisons cet algorithme dans la classe SudokuBoard2 pour effectuer la résolution des grilles, il est défini grâce à la méthode solve ().

#### **V ) Conclusion :**

##### **William :**

Ce projet de développement pour un jeu de Sudoku a été une bonne expérience pour se familiariser avec le java. J'ai bien aimé la réalisation des diagrammes de classes, du makefile et la séparation propre des classes en plusieurs fichiers. Tout au long du projet, j'ai pu apprécier la complexité et la diversité des tâches, notamment celle de l'algorithme de résolution de grille automatique, pour créer un programme interactif. Ce n'était certainement pas un projet simple mais j'en ressors avec une grande satisfaction d'avoir réussi le challenge.

**Fabio :** Ce projet a été une expérience enrichissante qui m'a permis de mettre en œuvre mes compétences en Java tout en découvrant les subtilités du jeu de Sudoku. Bien que j'aie rencontré quelques difficultés au début pour comprendre parfaitement les consignes, cela m'a poussé à approfondir mes connaissances et à développer mes compétences en programmation. Au final, cette expérience m'a permis de m'améliorer et de mieux appréhender la complexité des problèmes algorithmiques tout en consolidant ma maîtrise du langage Java.