

Mickaël Albarello

Monpierre Aurélien

Rapport de projet tutoré

Sommaire :

I) Introduction.....	3
II) Description des fonctionnalités du programme.....	4
III) Structure du programme.....	11
IIII) Les Conclusions.....	12

Introduction :

Nous devons créer un programme en java qui avait pour but de proposer une interface conviviale pour concevoir ou résoudre des grilles de Sudoku. Ce jeu de réflexion demande de remplir une grille en respectant une configuration de départ différente à chaque partie et quelques règles simples.

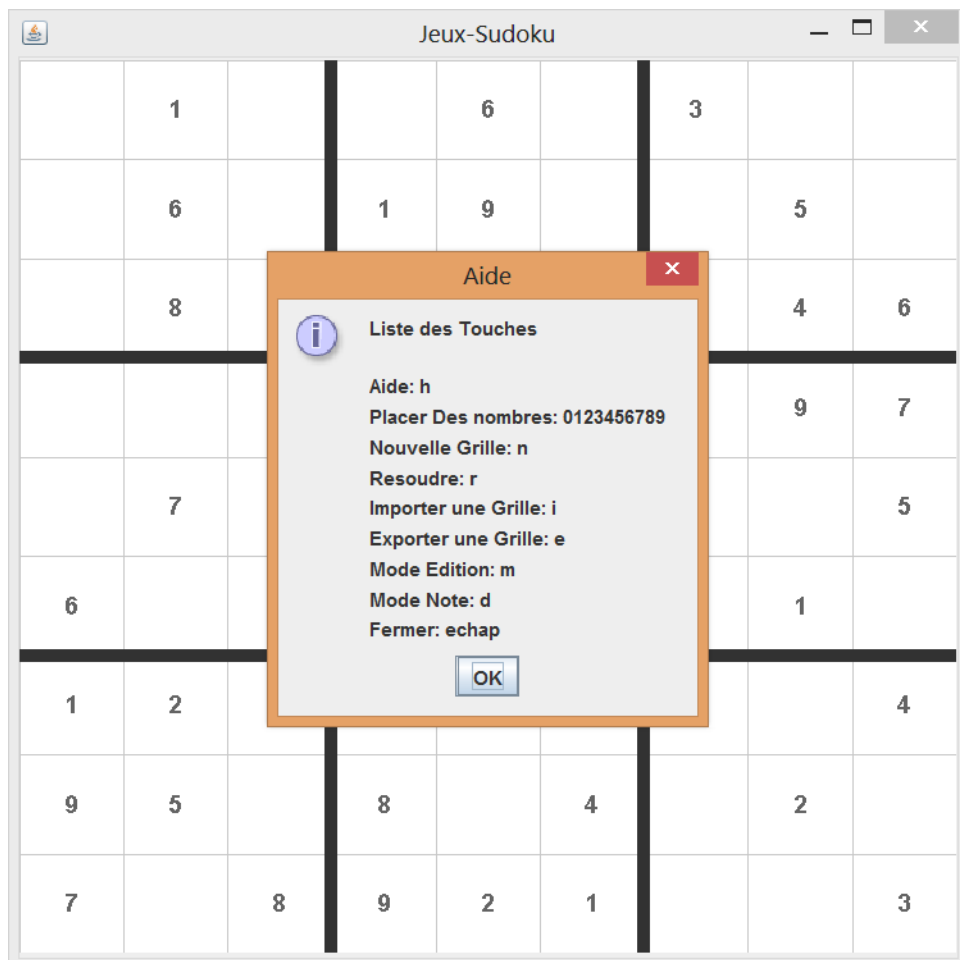
Nous nous sommes donc mis en binômes afin de résoudre ce problème avant le vendredi 10 avril 2020 à 18h00.

On fournit au joueur une grille où seulement quelques cases sont initialement remplies. Sa tâche consiste à remplir entièrement la grille en respectant les contraintes d'unicité. Les grilles de départ sont délicates à construire car elles doivent conduire à une seule solution.

Pour construire une grille, le joueur pouvait partir d'une grille vide ou charger une grille existante depuis un fichier. Il devient alors possible d'ajouter ou d'enlever des numéros dans la grille. Une fois la grille achevée, elle sera sauvegardée dans un nouveau ou un ancien fichier.

Description des fonctionnalités du programme :

Le jeu commence sur une fenêtre de bienvenu puis d'aide ou l'on indique au joueur les touches qui vont l'aider à utiliser notre sudoku. Le joueur trouve alors un sudoku aléatoire crée par le programme. Il a la possibilité de le remplir avec les numéros 123456789 et 0 remettant la case vide.

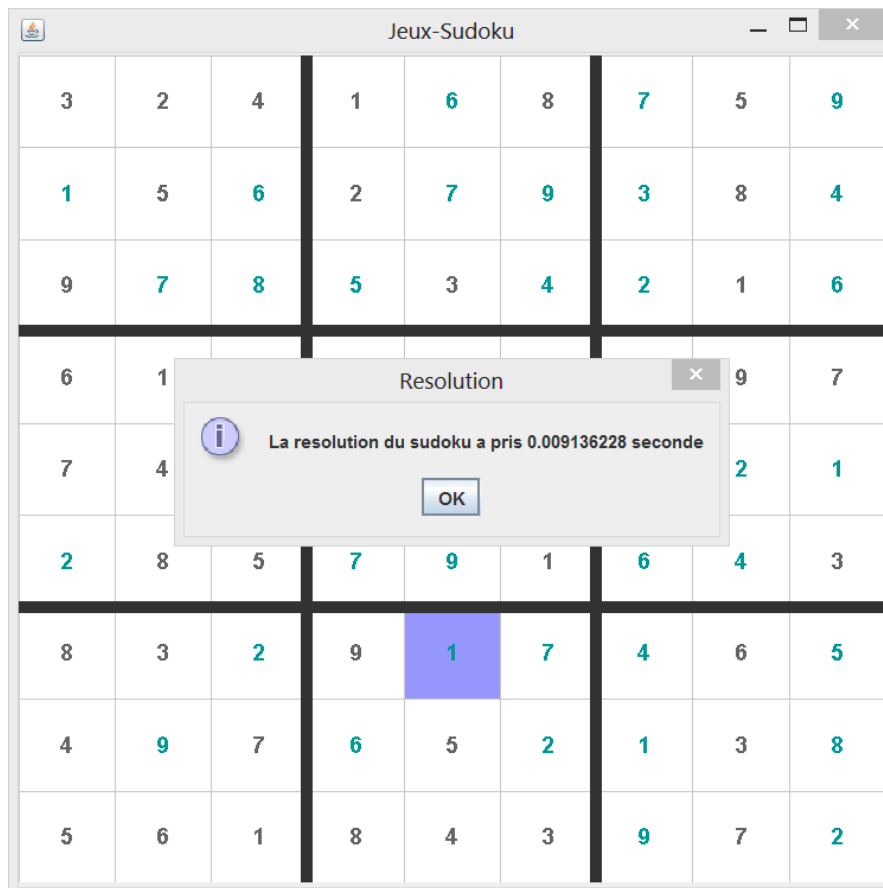


Lorsqu'on arrive sur le Sudoku, on lance le mode joueur. C'est à dire que la personne a la possibilité de remplir le sudoku mais pas de modifier les chiffres prérempli. Le sudoku est d'ailleurs déjà rempli pour qu'il soit faisable par l'utilisateur.

Jeux-Sudoku								
	1			6		3		
	6		1	9			5	
	8		5				4	6
					5	6	9	7
	7	4	6			8		5
6			3	8	7		1	
1	2	3				9		4
9	5		8		4		2	
7		8	9	2	1			3

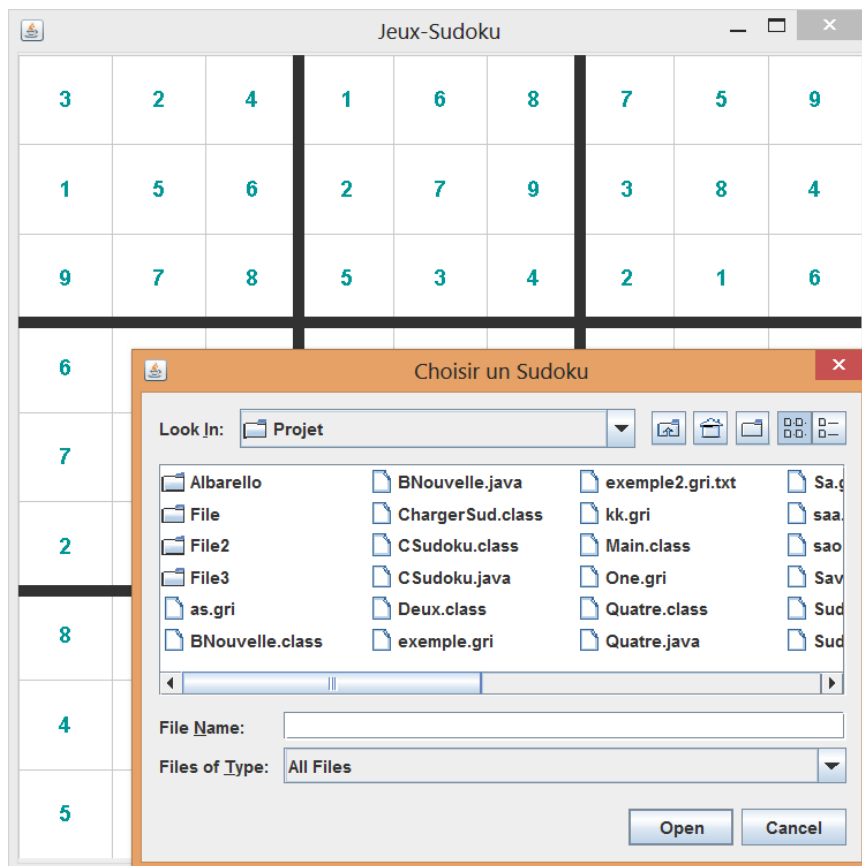
Lorsqu'on appuie sur la touche r on obtient alors la grille résolue et son temps en seconde.

Grille résolue :



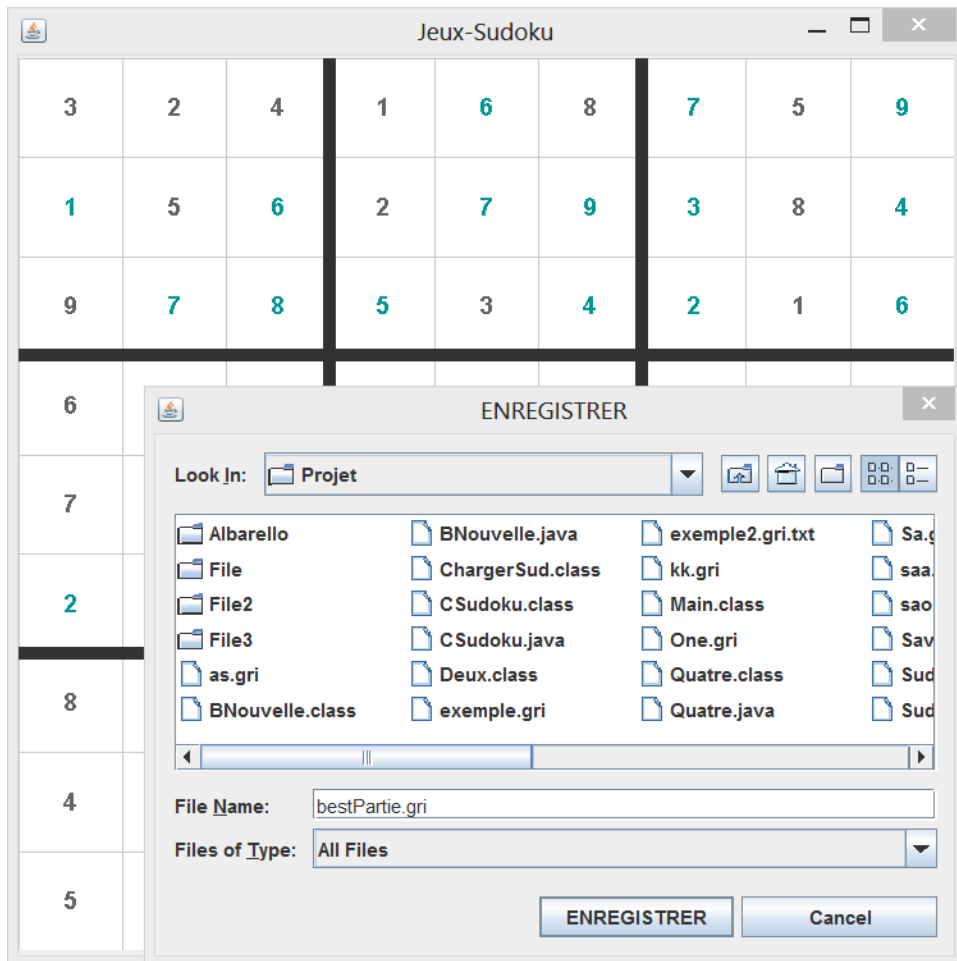
Dans notre jeu, nous avons la possibilité de charger une grille grâce à la touche i que le joueur a pu commencer (ou pas).

Importation :



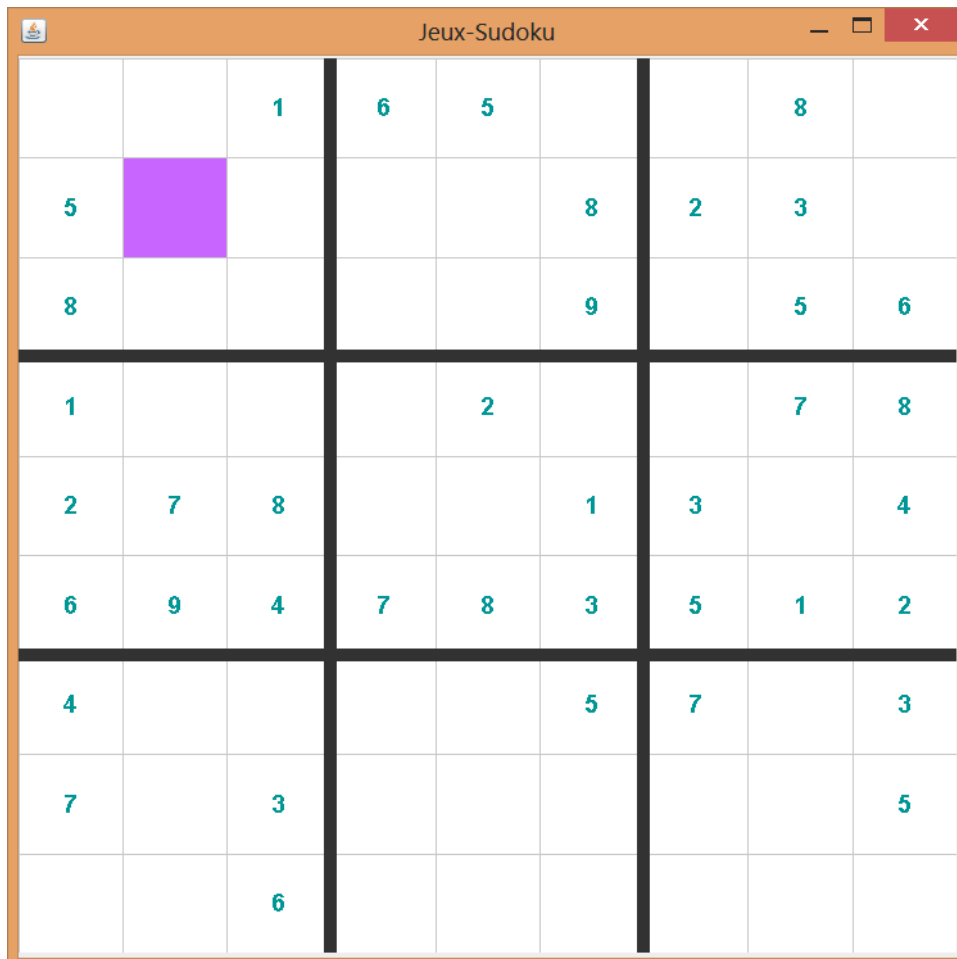
Le joueur a aussi la possibilité de pouvoir enregistrer sa grille dans un fichier .gri à tout moment du jeu grâce à la touche e.

Sauvegarde :



Ensuite nous avons le mode édition qui permet de créer son propre sudoku, c'est à dire la possibilité de modifier les cases de départs. La particularité du mode Edition c'est que lorsqu'on sélectionne une case, la couleur de la case est violette ce qui permet de mieux se repérer visuellement par rapport au mode utilisé.

Mode édition :



Erreur :

2	3	8
¹ 4	² 2	1
5	6	9

Lorsqu'un nombre est en incohérence avec un autre (sur une ligne/colonne/région), on l'indique en mettant ce chiffre en rouge.

Comme vous avez pu le voir sur le screen précédent, il y'avait plusieurs chiffres dans la même case, cela est dû au mode note, qui permet de mettre entre 1 et 4 chiffres dans la même case sans gêner l'utilisateur lorsqu'il remplit la case. Lorsqu'on que le mode note est utilisé, la couleur marron de sélection des cases permet de s'en souvenir et de ne pas se tromper.

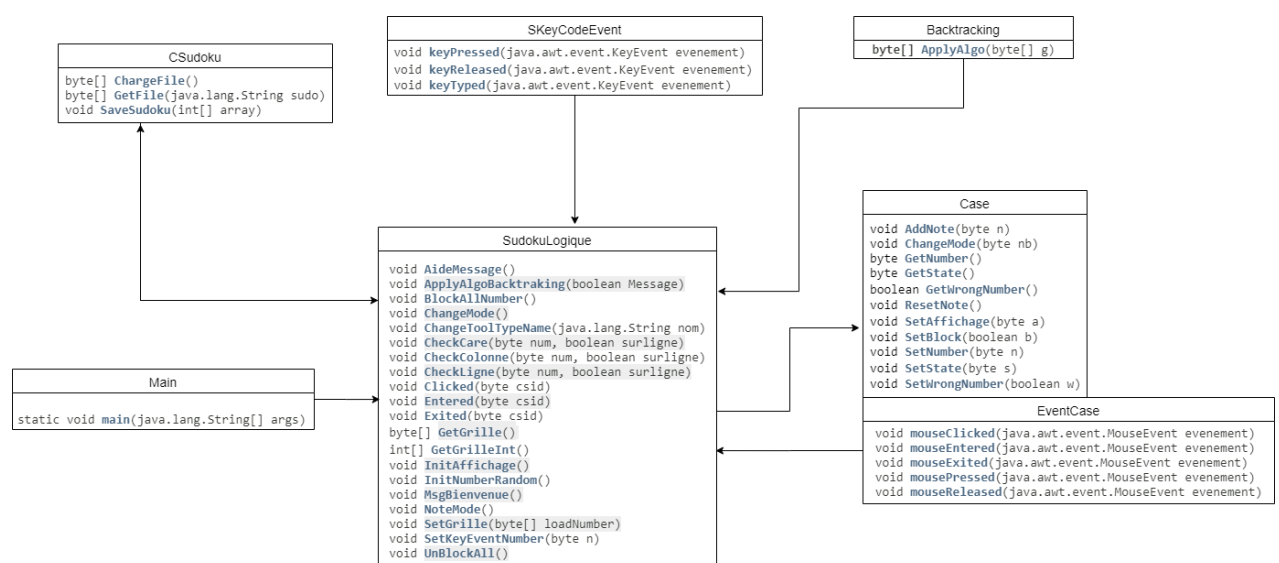
Mode note :

Jeux-Sudoku								
		4				7	8	9
	6	7	8	9		1		
8		1		4				
1			4			8		
4	5							1
7	8				3	6		5
		2		7		9		8
6	7	5	9	2		4		3
9	4	8	6	3	1			2

9		8
4	3 5	3
	1 4	2

Ici on a 4 notes mise dans la même case.

Structure du programme :



Le fichier `backtracking.java` sert à résoudre les grilles de sudoku. Elle regarde d'abord la position sur laquelle elle se situe si elle est à la dernière position elle renvoie la solution si la position de la case ou elle fait c'est testé est égale à 0 alors elle va chercher à trouver un nombre non présent sur la ligne, la colonne et dans la région et modifie la valeur de la grille. Il utilise la récursivité afin de trouver les solutions possibles de la grille cette méthode peut être très rapide comme très lente car certaines grilles peuvent mettre jusqu'à 30 à 40 secondes à se résoudre ce qui peut donner l'impression que le programme a planté, lors de la génération de la nouvelle grille, on génère des nombres aléatoires une très petite quantité puis on applique l'algorithme de résolution et enfin on supprime la moitié des nombres. C'est dans ce fichier qu'on change le tableau à 2 dimensions (9x9) pour en avoir un seul (de 81 valeurs) afin de faciliter l'interaction avec les cases `JComponent` puisqu'en utilisant un seul tableau on peut donner un ID à chaque `EventCase` et donc interagir plus facilement avec le `SudokuLogique`.

Le fichier `Csudoku.java` permet le contrôle des fichiers du sudoku, notamment pour la sauvegarde et le chargement de fichier sudoku. Il prend un fichier grâce à un `JFileChooser` puis le traduit en un tableau pour pouvoir en faire une grille de sudoku. Pareil pour la sauvegarde, il enregistre le tableau de 81 valeurs et l'écrit dans un fichier.

Le fichier `Case.java` est utilisé pour afficher et structurer les cases du sudoku, c'est ici qu'on utilise le `Graphics` `SecondPinceau` afin de colorier la case en fonction du mode sélectionné. On dessine aussi les contours des cases grâce à un `switch` qui nous indique le nombre de côté avec des séparations en noir et on regarde si une case est, survolée, sélectionnée ou non sélectionnée afin de lui donner une couleur et de savoir son état (bloquer ou pas) ainsi que son mode (Edition, note, joueur).

`EventCase.java` permet de détecter les événements sur une case grâce à `MouseListener`, notamment si on clique, entre dedans, sort d'une case.

`Main.java` affiche la `JFrame` fait le lien entre les événements et les cases et le `SudokuLogique` et initialise l'affichage ainsi qu'une grille de départ.

`SkeyCodeEvent.java` permet de détecter les touches de l'utilisateur et d'effectuer des actions sur le Sudoku, c'est ici qu'on règle les actions sur les touches (les nombres de 1 à 9 et les touches indiquées dans le menu d'aide).

`SudokuLogique.java` est le fichier qui s'occupe de contrôler le sudoku et fait le lien entre les événements et les cases du sudoku. C'est ici qu'on initialise les tableaux qui vont permettre de valider un sudoku, de créer un sudoku aléatoirement réalisable par un humain et de modifier l'état d'interaction d'une case. C'est lui aussi qui gère les messages de bienvenue et d'aide, et choisit les modes que l'utilisateur a sélectionnés.

Les Conclusions :

Conclusion personnelle Aurélien :

C'est quand même bien plus sympa d'être en groupe avec quelqu'un qui travaille et qu'on apprécie d'autant plus. Surtout quand il nous lance un "J'ai fini ma partie" à la première semaine du projet. Je vois un peu mieux pourquoi on doit faire ce travail en groupe, on a presque plus galéré à lier mon travail ("mal écrit") au sien qu'à faire le sudoku.

Conclusion personnelle Albarello :

J'ai fait relativement rapidement ma partie pour ne pas avoir de problème de temps sur le projet.