

Nell TELECHEA-LOJOU

BUT1 Informatique Fontainebleau

Benjamin BRIBANT

RAPPORT SUDOKU

SAE DEV 2.1

TABLE DES MATIERES

Contenu

- I. Introduction
- II. Description des fonctionnalités
- III. Diagrammes de classes
- IV. Algorithme de résolution du jeu
- V. Conclusion

INTRODUCTION

I. Introduction

Dans ce rapport, nous allons présenter notre projet de Sudoku et l'expliquer pas à pas.

Le Sudoku est un jeu de réflexion dont le but est de remplir complètement une grille de 81 cases, en respectant certaines règles.

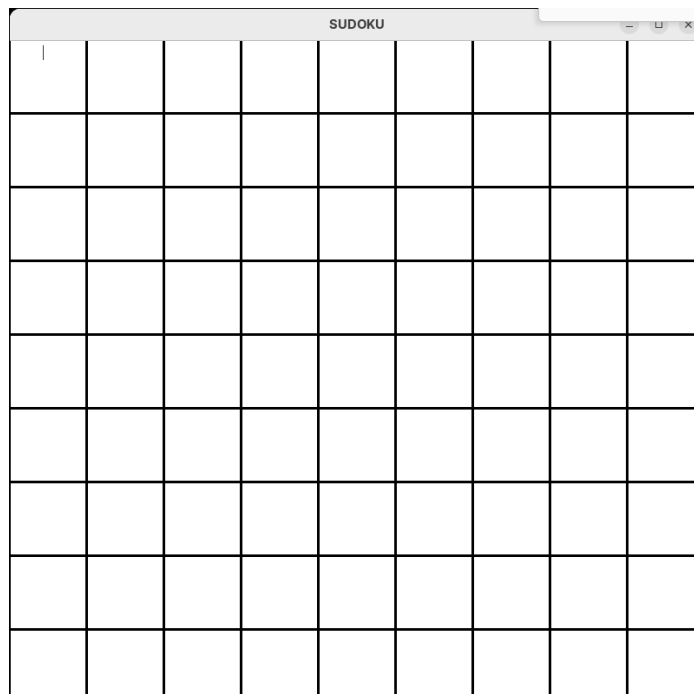
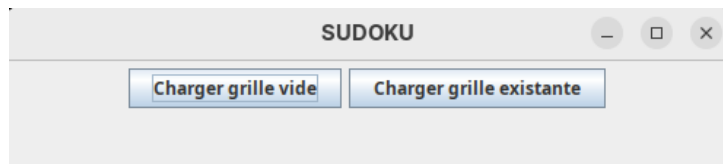
La grille est composée de 9 lignes et 9 colonnes, et est divisée en 9 régions en forme de carré, chacune composée de 9 cases. Chaque case doit accueillir un chiffre entre 1 et 9, mais ce chiffre ne doit pas apparaître plus d'une fois dans une même ligne, une même colonne ou une même région.

En général, certaines cases sont préremplies avant le début du jeu. Le joueur doit utiliser sa logique pour remplir la grille.

II. Description des fonctionnalités

Partie élaboration de la grille de jeu:

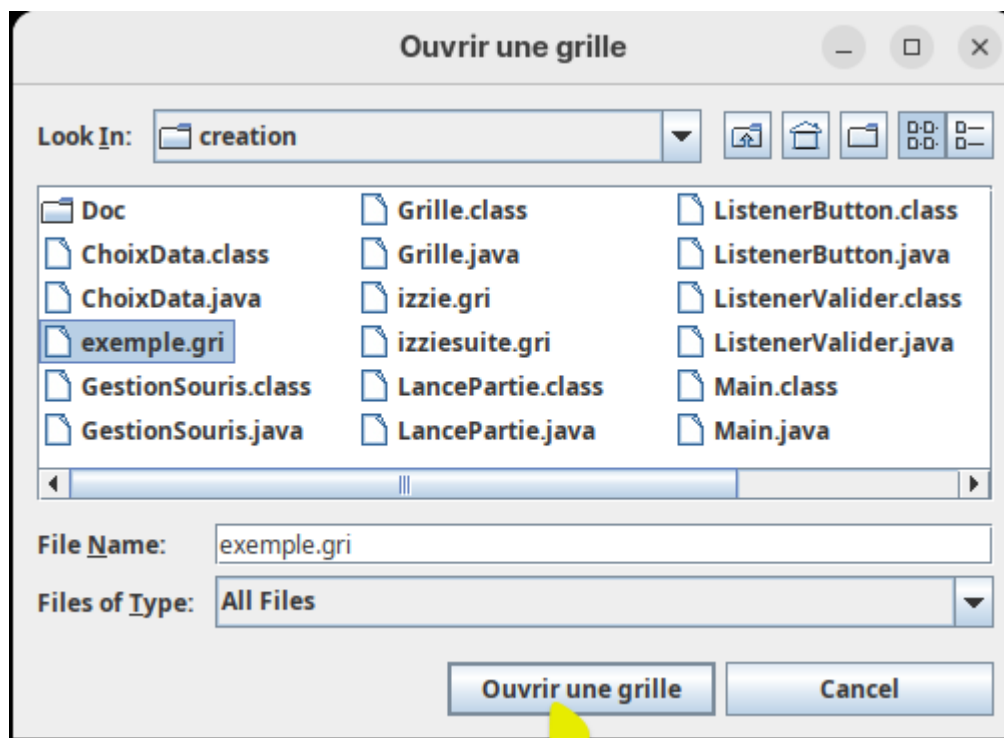
Quand on lance le jeu, une fenêtre avec deux boutons s'ouvre: le premier bouton permet d'ouvrir une autre fenêtre avec une grille complètement vide.



DESCRIPTION DES FONCTIONNALITES

Le deuxième bouton permet de charger une grille à partir d'un fichier d'extension .gri, dans lequel se trouvent 9 nombres sous forme d'entiers, qui correspondent aux chiffres qui apparaîtront dans les cases préremplies ; les 0 correspondent aux cases vides.

Cliquer sur ce bouton ouvre une fenêtre qui permet de choisir le fichier que l'on veut charger.



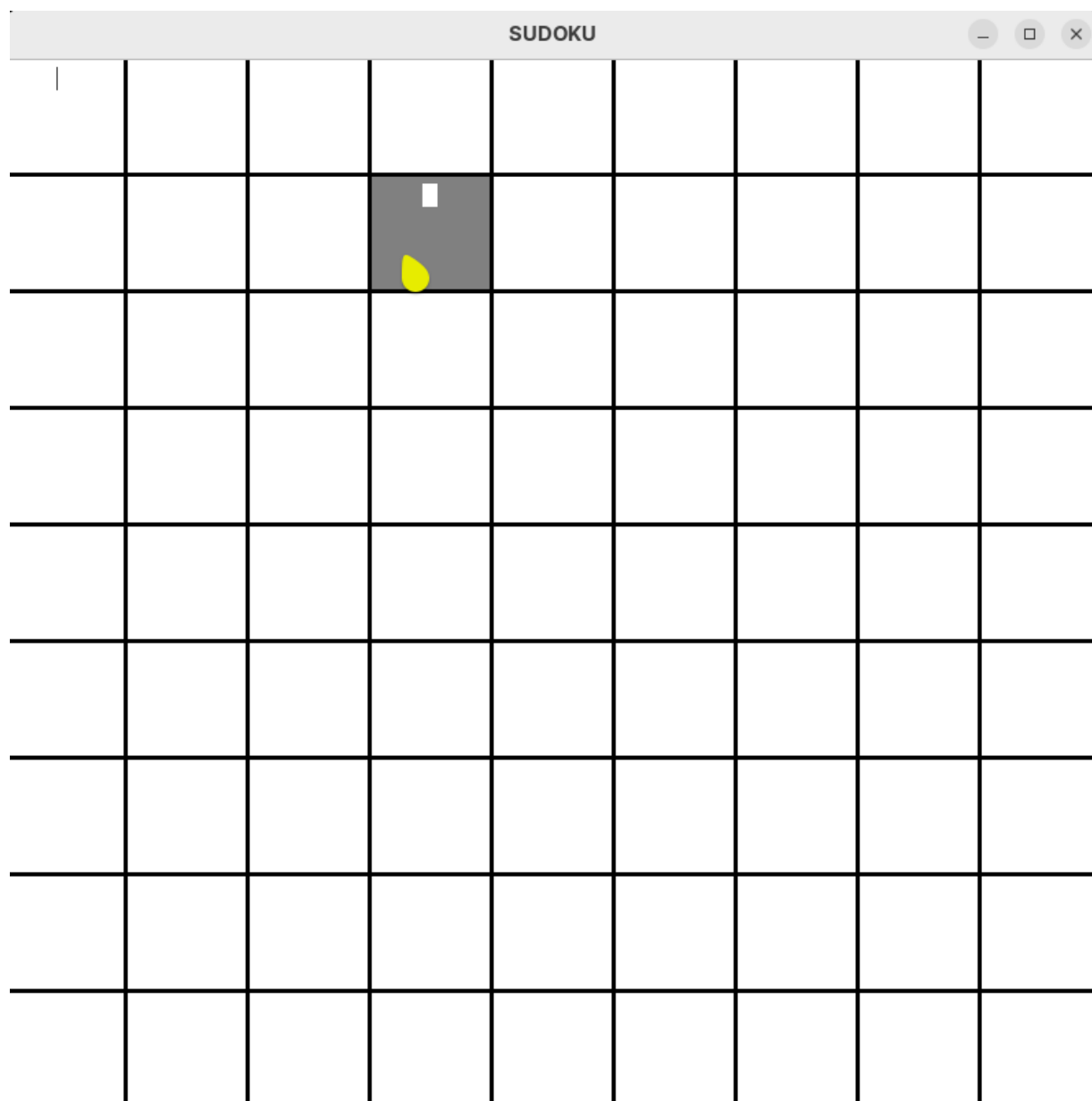
DESCRIPTION DES FONCTIONNALITES

Une fois le fichier choisi, une nouvelle fenêtre va s'ouvrir contenant une grille dont les cases seront préremplies avec les chiffres contenus dans le fichier.

SUDOKU								
				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				

DESCRIPTION DES FONCTIONNALITES

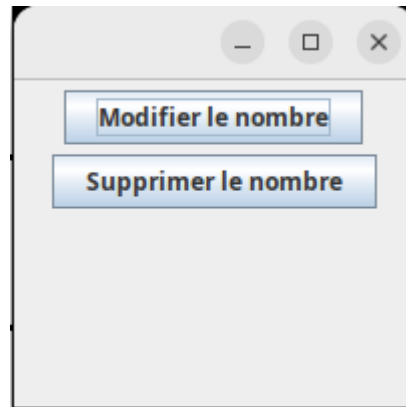
Quand on passe la souris sur une case de la grille, celle-ci devient grise.



DESCRIPTION DES FONCTIONNALITES

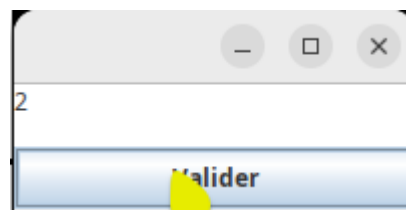
Si on clique dessus, une fenêtre s'ouvre.

Celle-ci contient deux boutons qui permettent de soit supprimer le nombre se trouvant dans la case, soit ajouter ou modifier ce nombre (s'il n'y a pas de nombre et qu'on choisit supprimer, rien ne se passe).



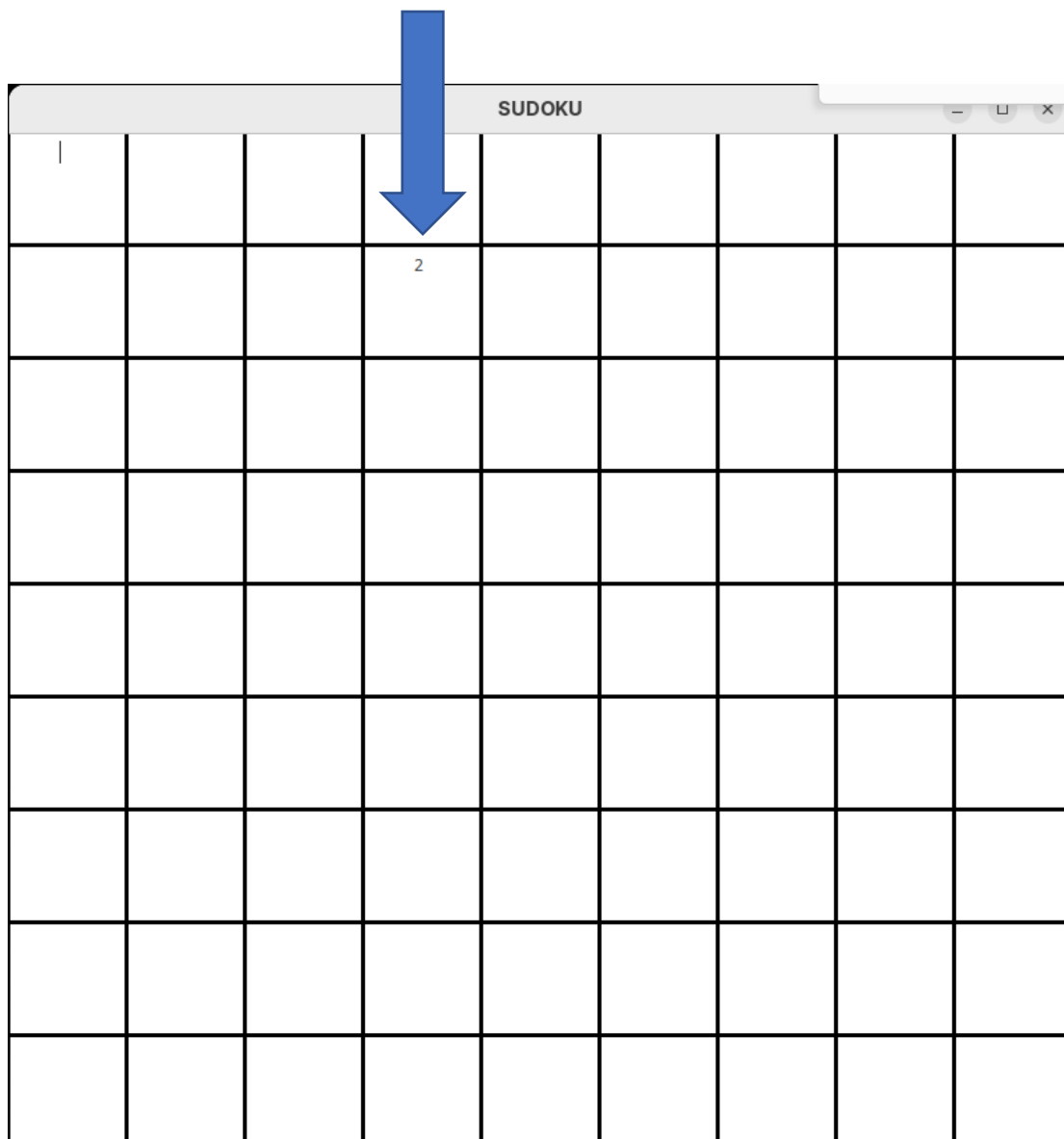
Cliquer sur le bouton permettant de modifier affiche une nouvelle fenêtre: cette fenêtre contient une zone de texte, si on veut ajouter un chiffre dans la case, on peut l'écrire dedans puis appuyer sur le bouton valider.

Le jeu va alors vérifier si ce chiffre respecte les règles décrites plus haut, et, si c'est le cas, le chiffre sera ajouté à la case. Si ce n'est pas le cas, rien ne sera ajouté.



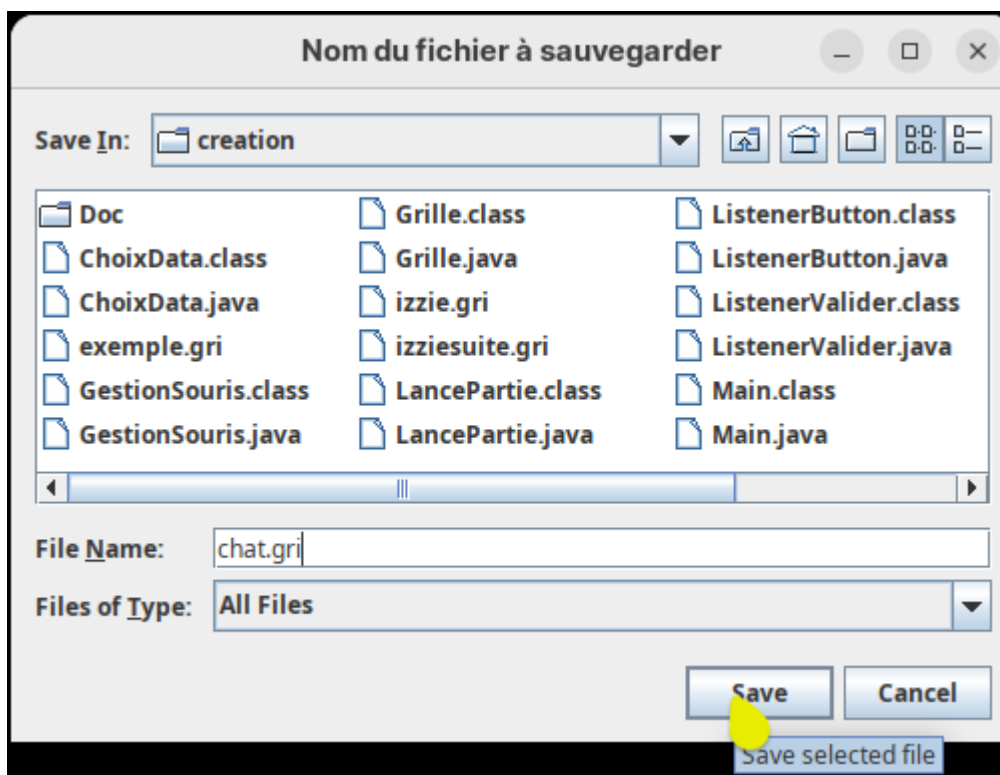
DESCRIPTION DES FONCTIONNALITES

Le 2 a bien été ajouté!



DESCRIPTION DES FONCTIONNALITES

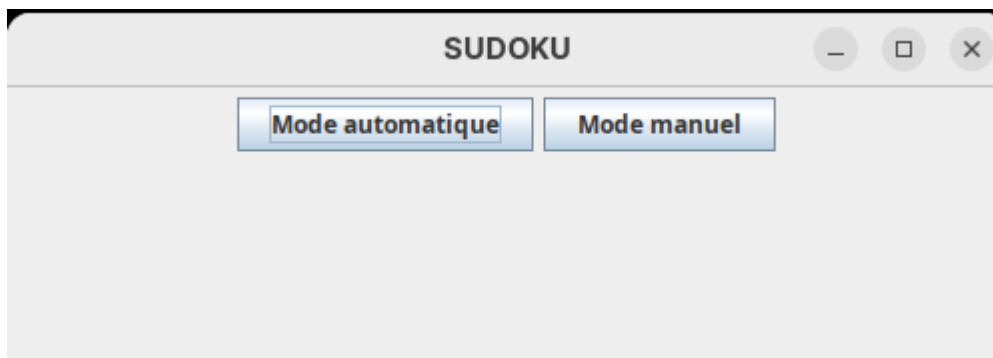
Quand le joueur a fini, il peut fermer la grille et la première fenêtre. Cela ouvrira une fenêtre qui permettra de choisir un fichier dans lequel il veut enregistrer la progression de sa grille.
La grille sera enregistrée dans le même format que décrit précédemment.



DESCRIPTION DES FONCTIONNALITES

Partie résolution de la grille:

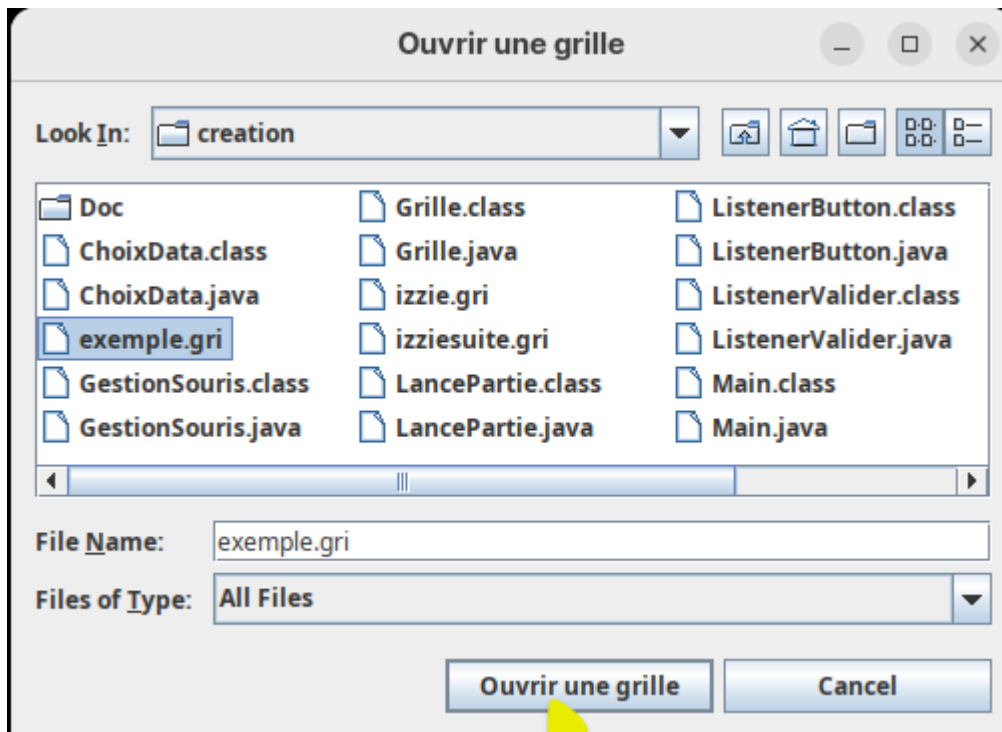
Quand on lance le jeu, une fenêtre avec deux boutons s'ouvre :
le premier bouton permet de résoudre une grille automatiquement et le
deuxième bouton permet de résoudre une grille manuellement en
chargeant une grille à partir d'un fichier d'extension .gri, dans le même
format que décrit précédemment.



DESCRIPTION DES FONCTIONNALITES

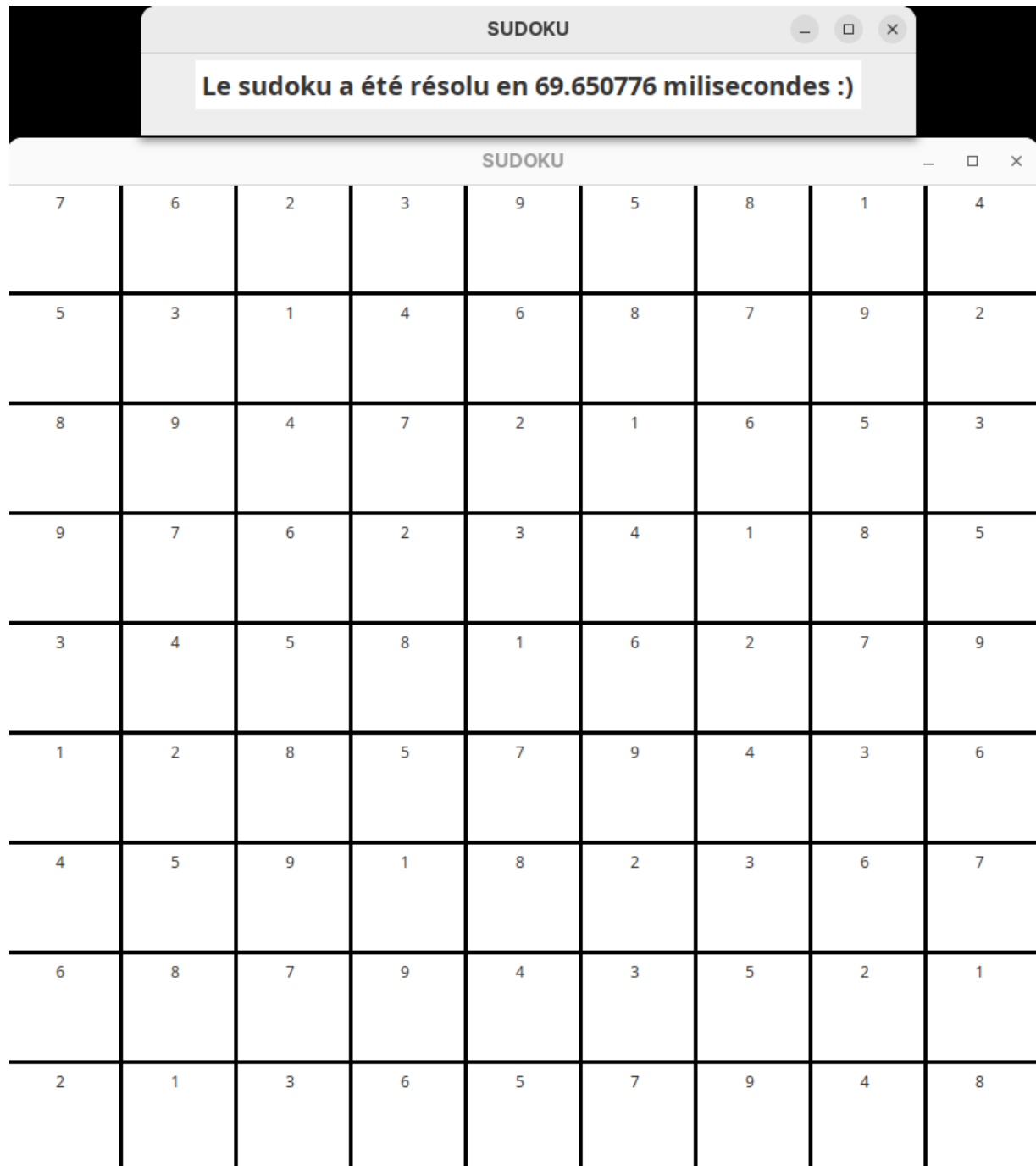
Si le mode automatique est choisi, une fenêtre qui permet de choisir le fichier que l'on veut charger.

Cliquer sur ce bouton ouvre une fenêtre qui permet de choisir le fichier que l'on veut charger va s'afficher.



DESCRIPTION DES FONCTIONNALITES

Une fois la grille choisie, le programme la résout de lui-même puis affiche la grille résolue et le temps pris en millisecondes.



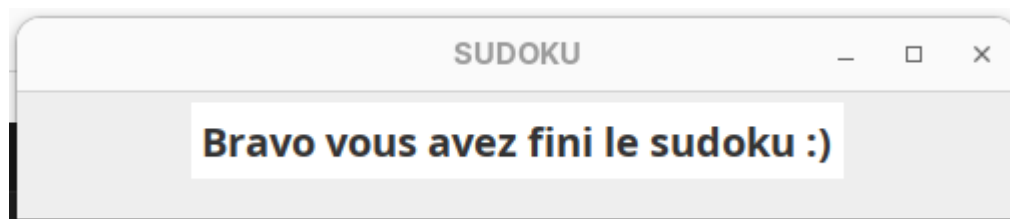
The image shows a screenshot of a web application titled "SUDOKU". A message box at the top states: "Le sudoku a été résolu en 69.650776 millisecondes :)". Below this is a 9x9 grid representing a solved Sudoku puzzle. The grid is displayed in a window titled "SUDOKU".

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

DESCRIPTION DES FONCTIONNALITES

Si le mode manuel est choisi, on choisit aussi une grille, elle s'affiche et on peut commencer à jouer.

Le jeu suit les règles décrites précédemment, avec les mêmes fonctionnalités d'ajout et de suppression que pour l'élaboration des grilles. Une fois la grille finie, un message de félicitations apparaît.



Vous pouvez fermer la fenêtre et recommencer à jouer autant que vous le voulez.

DIAGRAMMES DE CLASSES

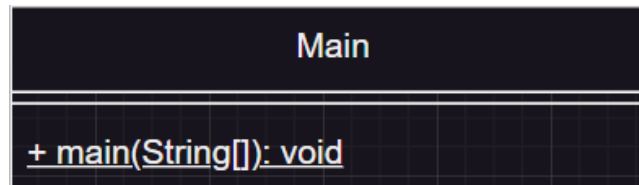
III. Diagrammes de classes

Partie élaboration :

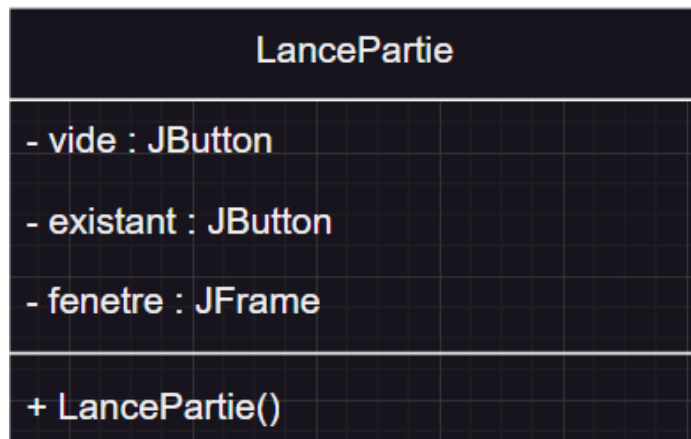
Nous allons dans un premier temps présenter les classes détaillées individuellement.

Ensuite nous présenterons le diagramme avec les relations entre les classes.

Classe Main :

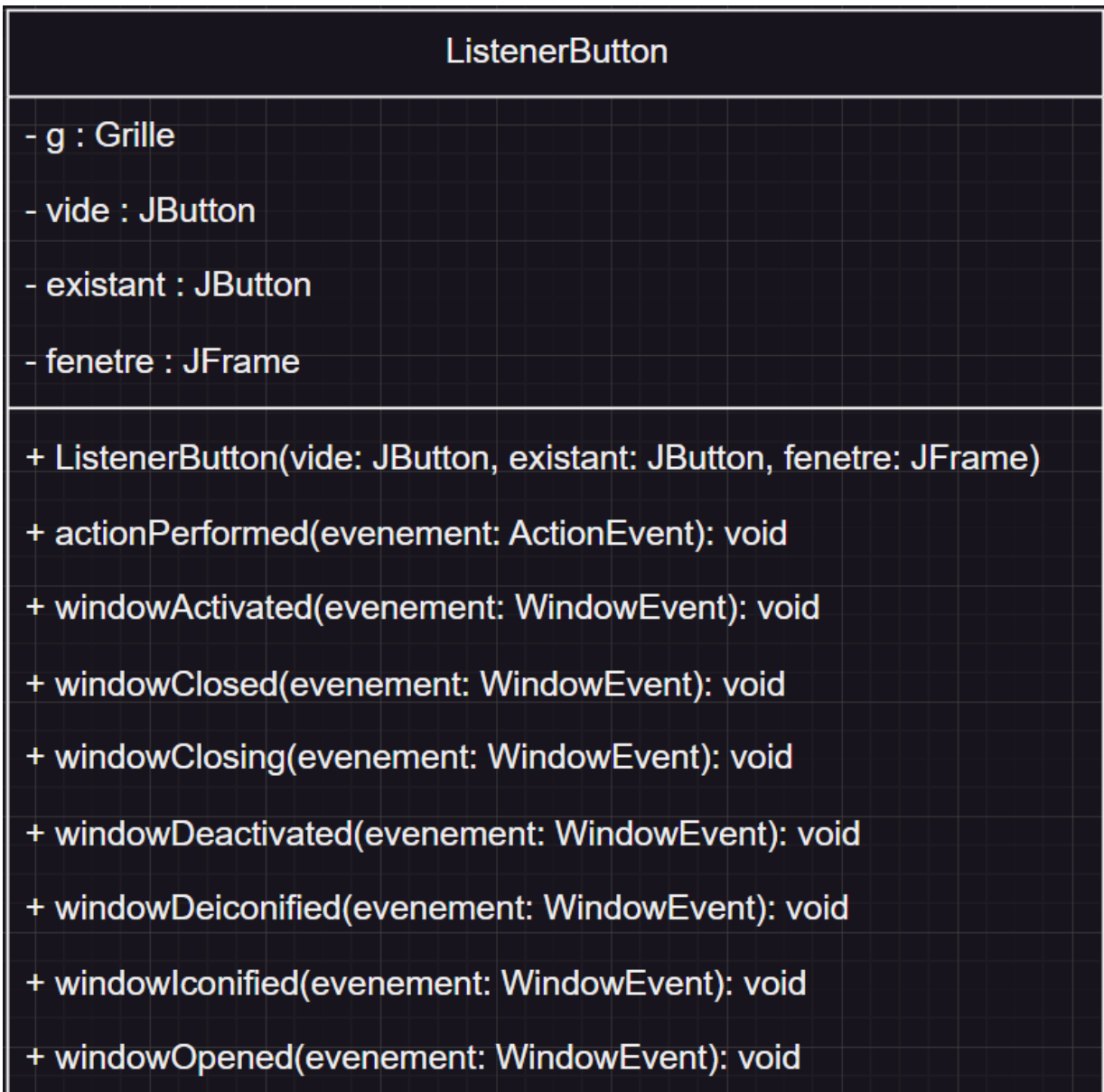


Classe LancePartie :



DIAGRAMMES DE CLASSES

Classe ListenerButton :



DIAGRAMMES DE CLASSES

Classe Grille :

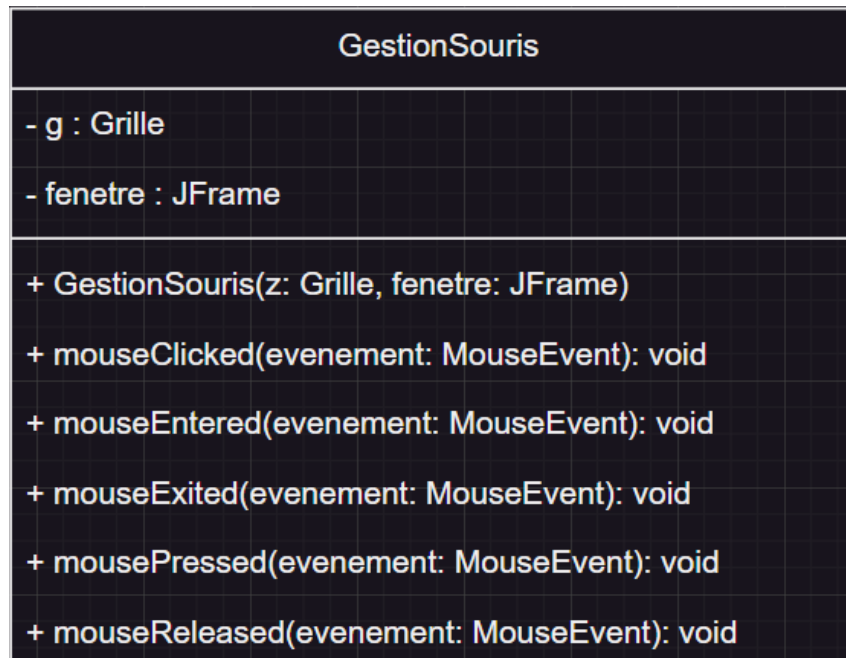
Grille
- grille : int[][]
+ Grille()
+ initGrilleVide(): void
+ initGrilleFichier(nomFichier: String): void
+ afficheGrille(): void
+ majGrille(val: int, i: int, j: int, etat: boolean):
+ saveGrille(nomFichier: String): void

Classe MyPanel :

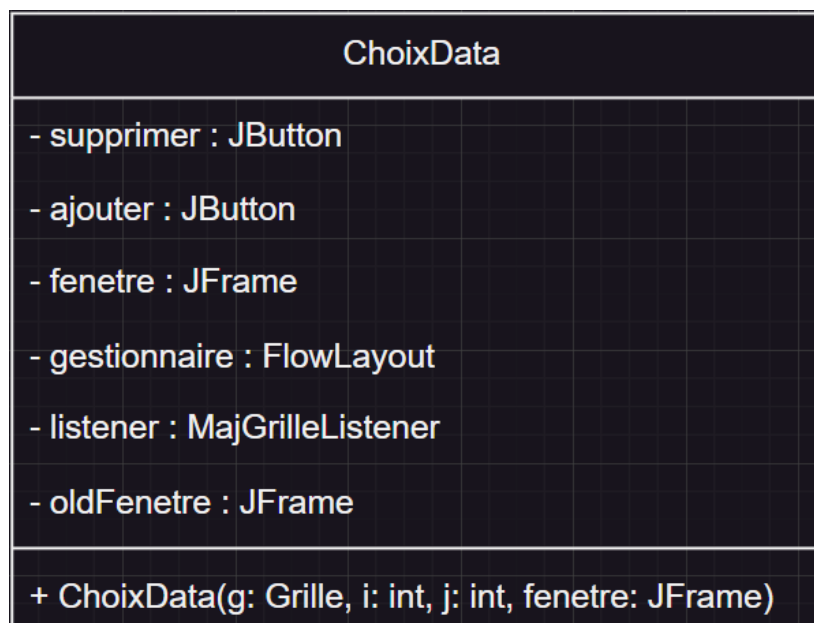
MyPanel
- i : int
- j : int
+ MyPanel(i: int, j: int)
+ getI(): int
+ getJ(): int

DIAGRAMMES DE CLASSES

Classe GestionSouris :

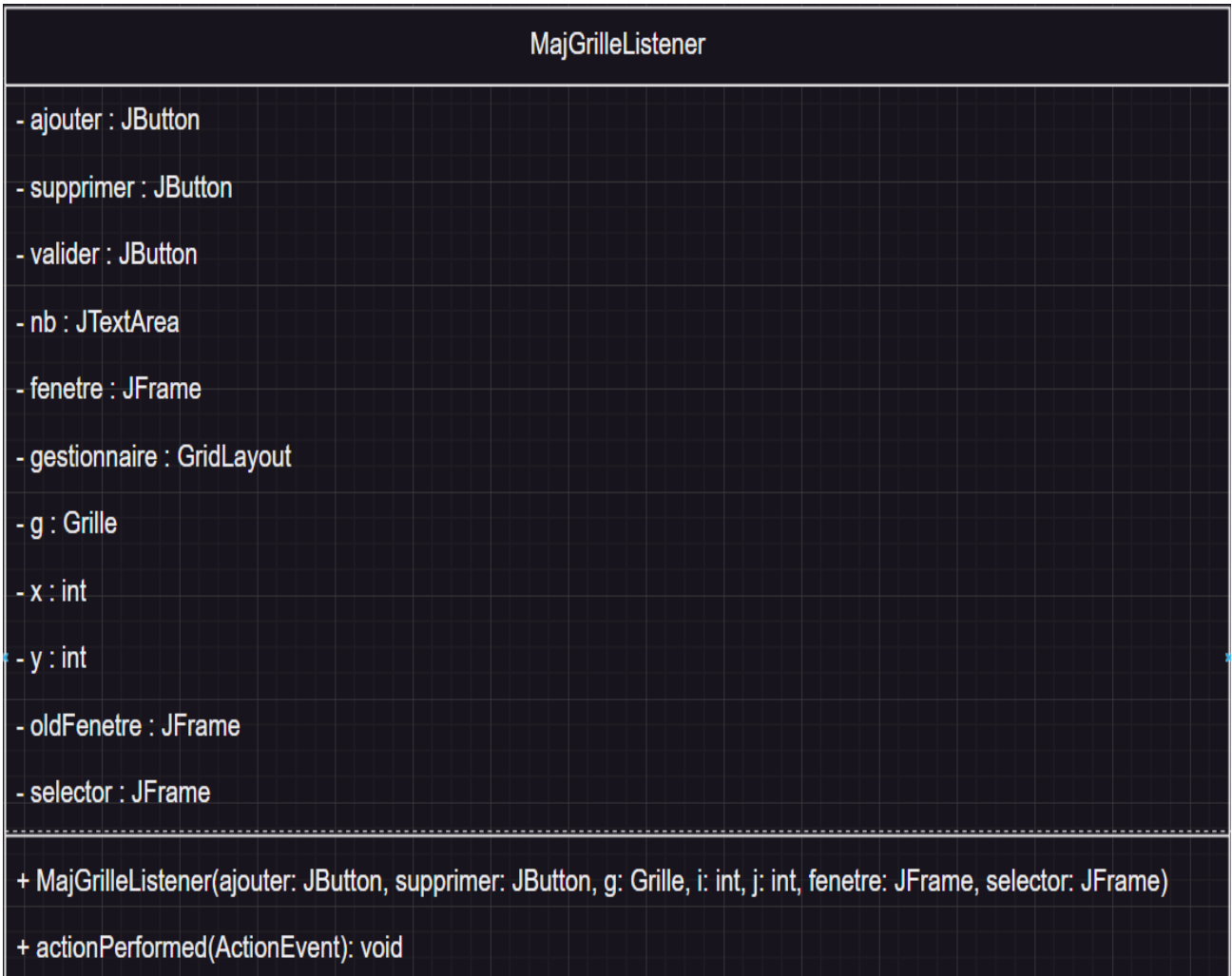


Classe ChoixData :



DIAGRAMMES DE CLASSES

Classe MajGrilleListener :

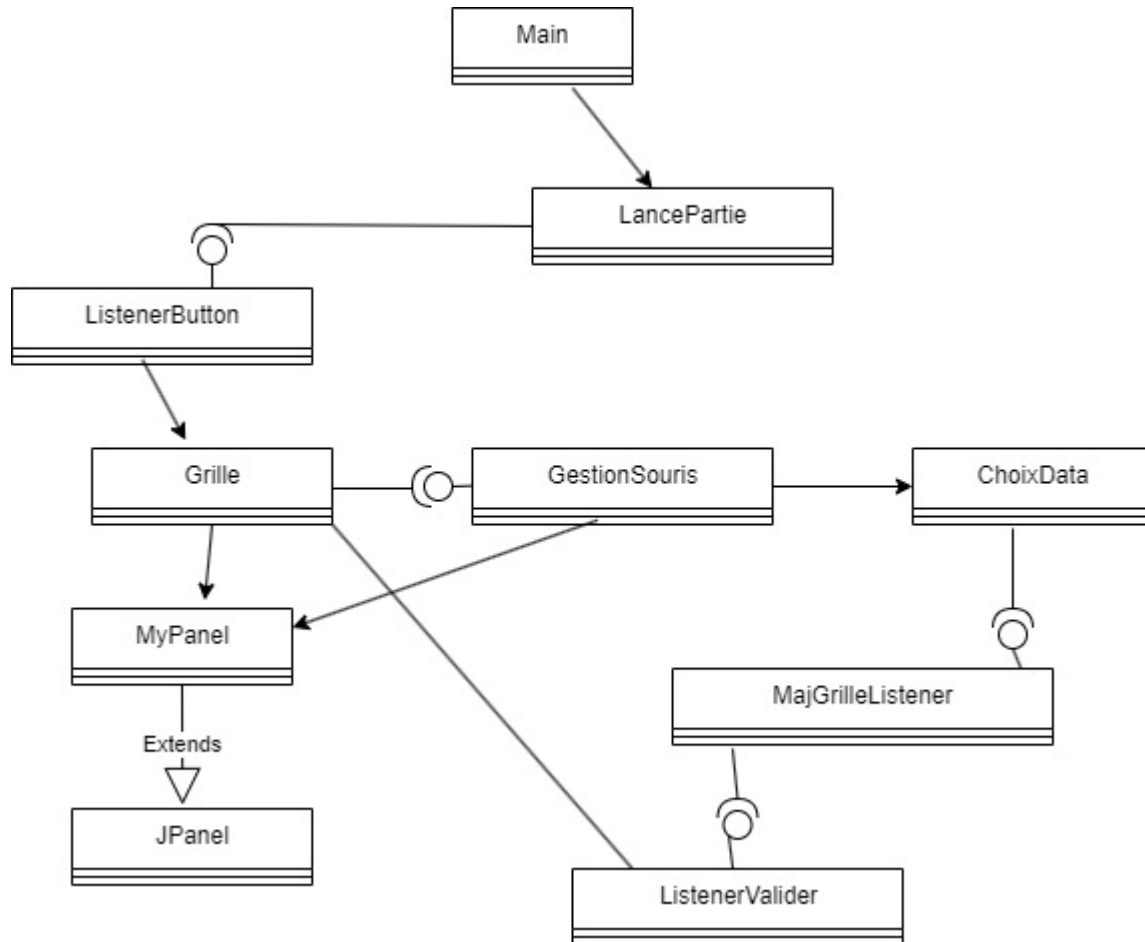


DIAGRAMMES DE CLASSES

Classe ListenerValider :



DIAGRAMMES DE CLASSES



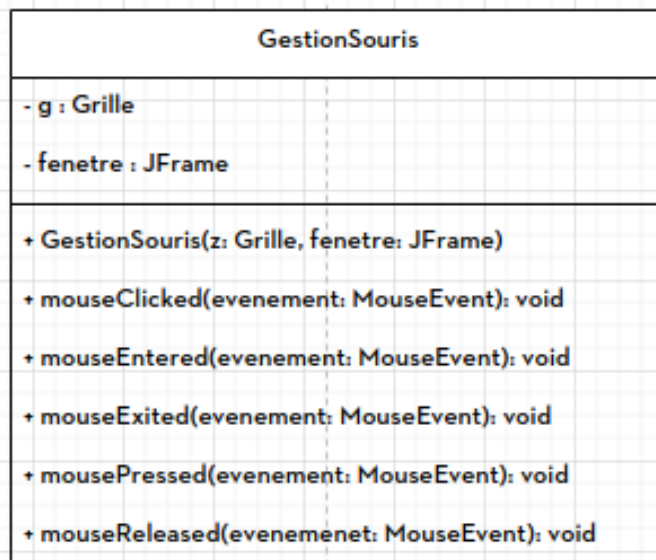
DIAGRAMMES DE CLASSES

Partie résolution :

Nous allons dans un premier temps présenter les classes détaillées individuellement.

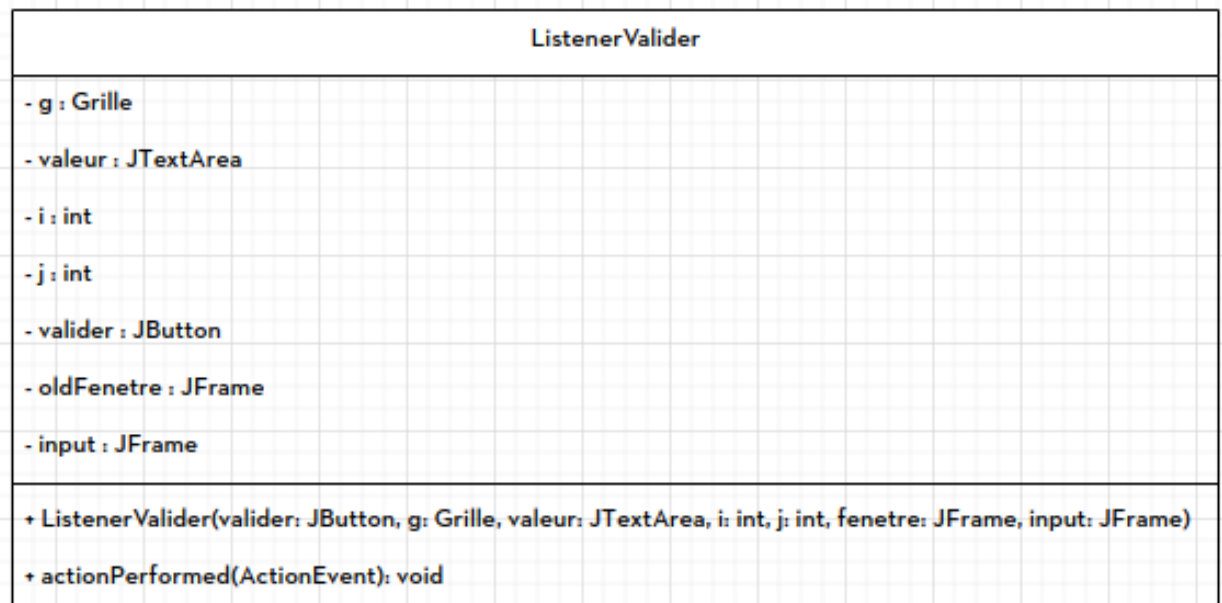
Ensuite nous présenterons le diagramme avec les relations entre les classes.

Classe GestionSouris :

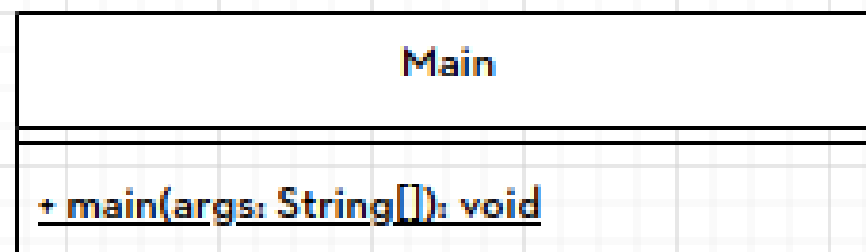


DIAGRAMMES DE CLASSES

Classe ListenerValider :

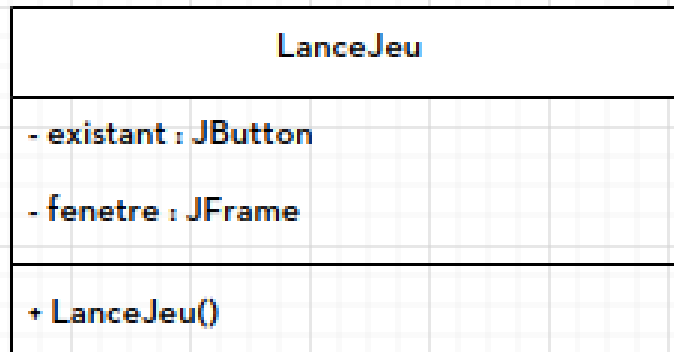


Classe Main :

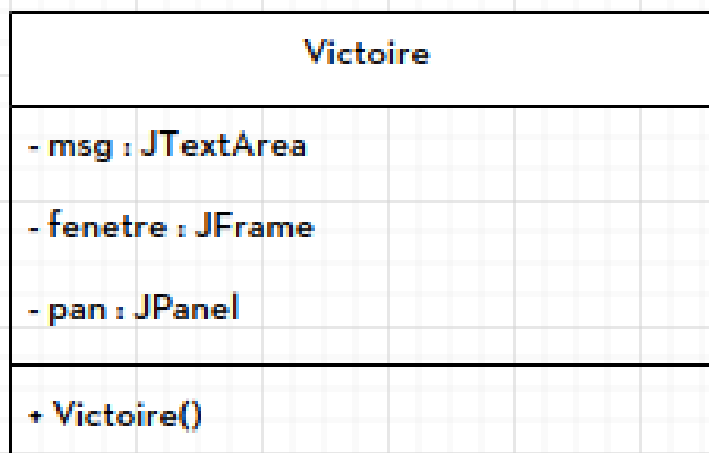


DIAGRAMMES DE CLASSES

Classe LanceJeu :

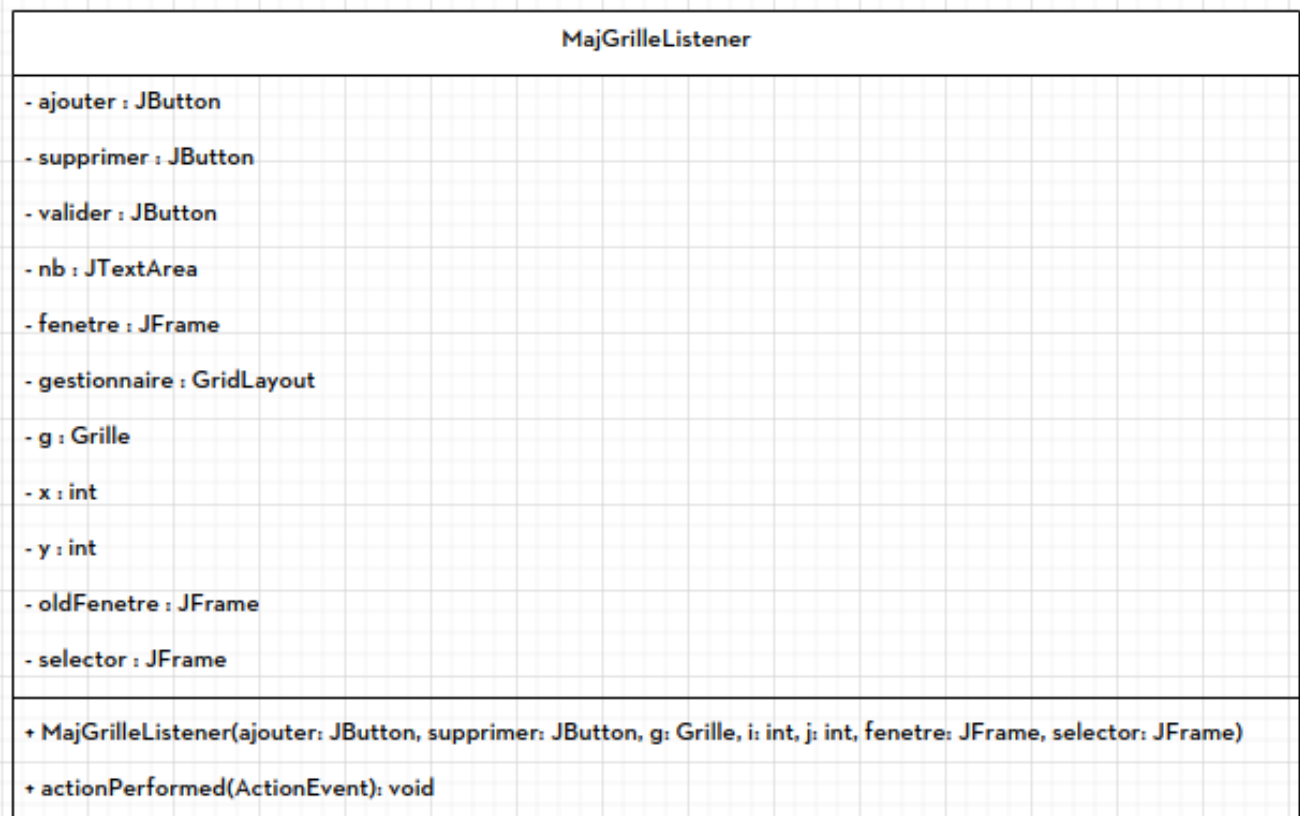


Classe Victoire :



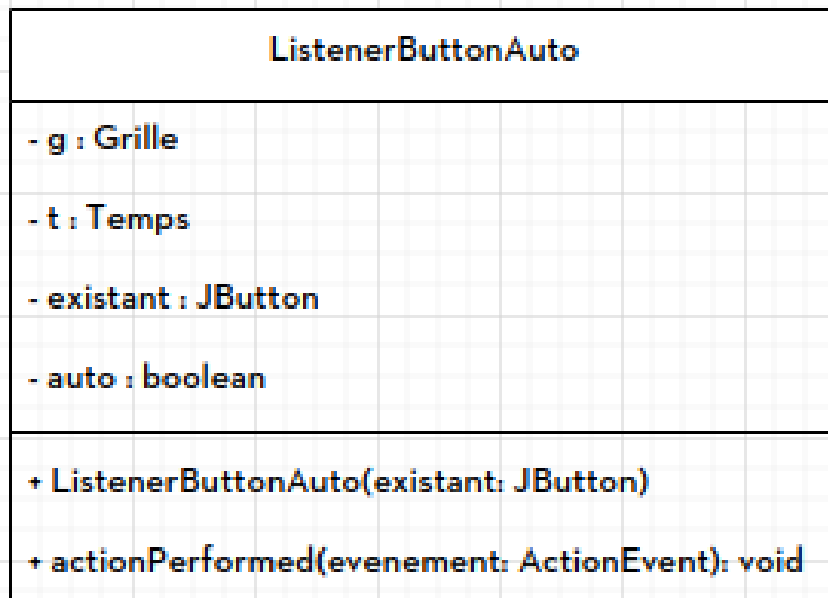
DIAGRAMMES DE CLASSES

Classe MajGrilleListener :

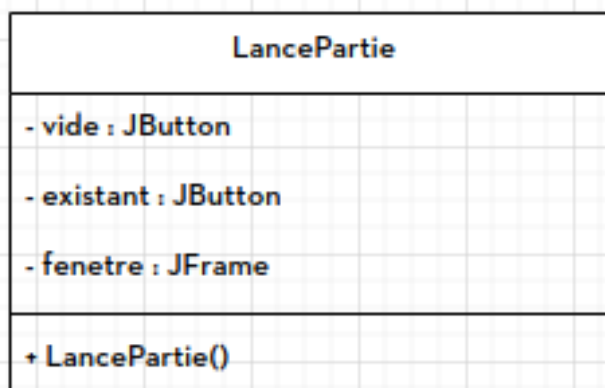


DIAGRAMMES DE CLASSES

Classe ListenerButtonAuto :

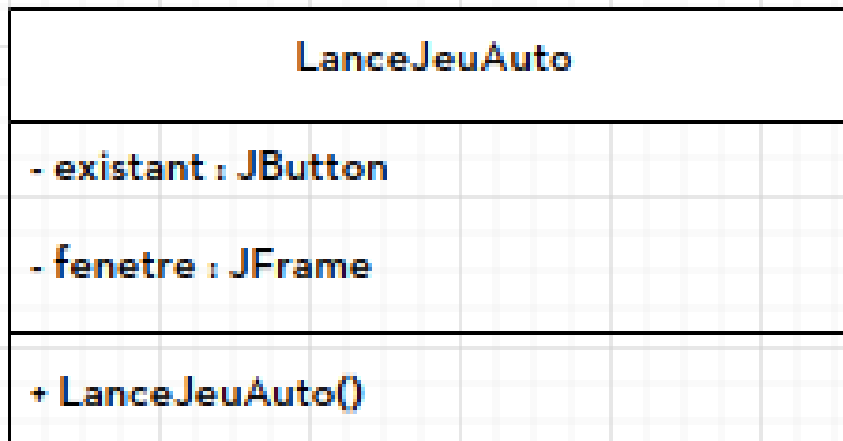


Classe LancePartie :

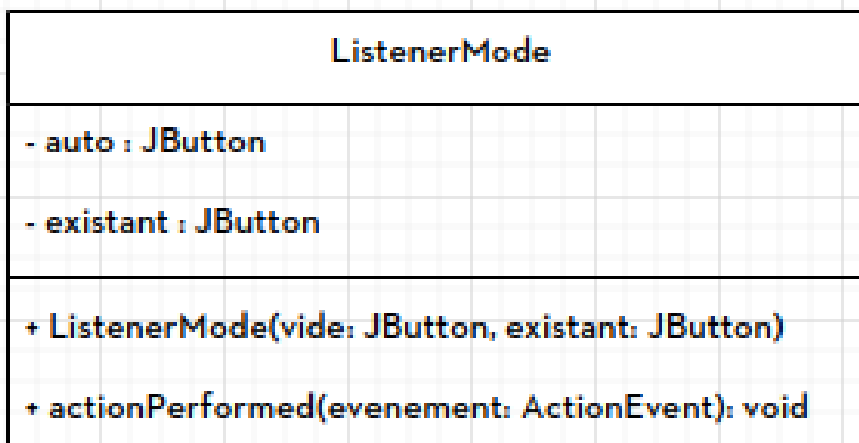


DIAGRAMMES DE CLASSES

Classe LanceJeuAuto :

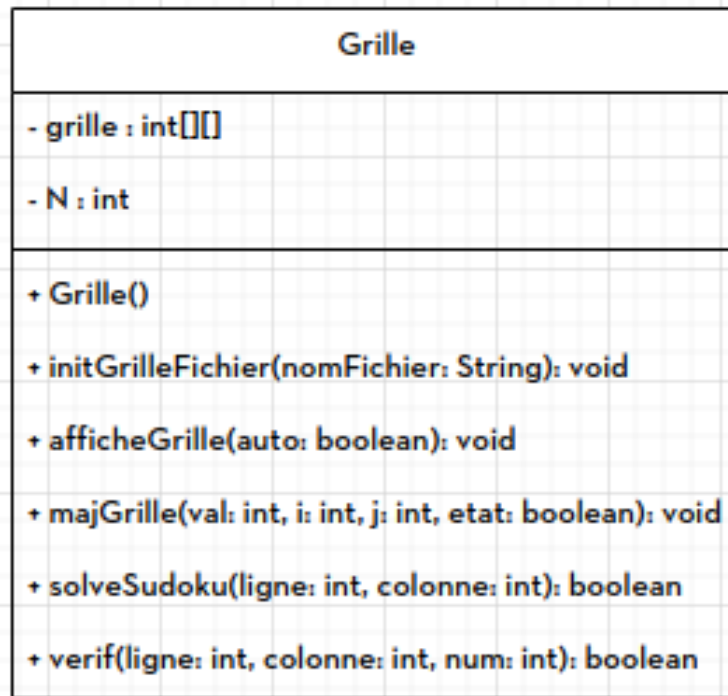


Classe ListenerMode :

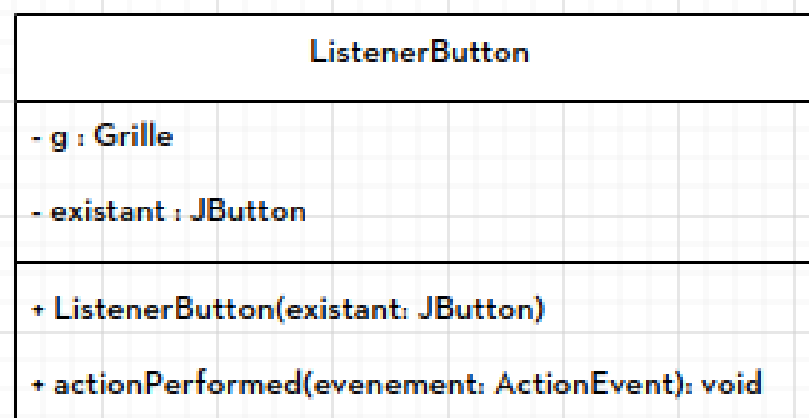


DIAGRAMMES DE CLASSES

Classe Grille :

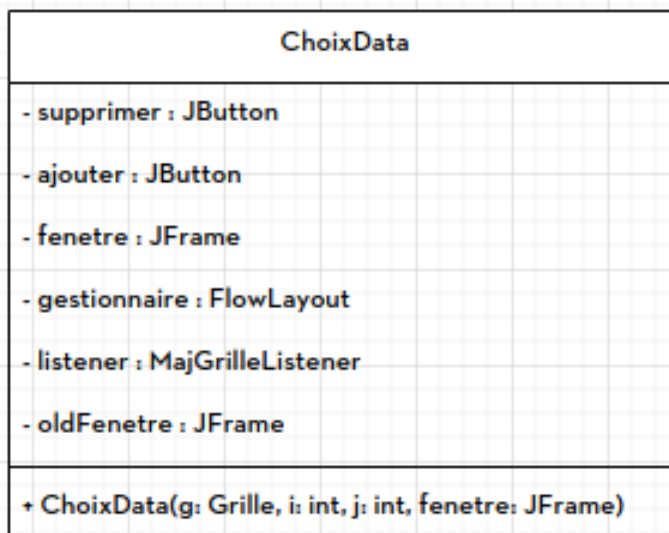


Classe ListenerButton :

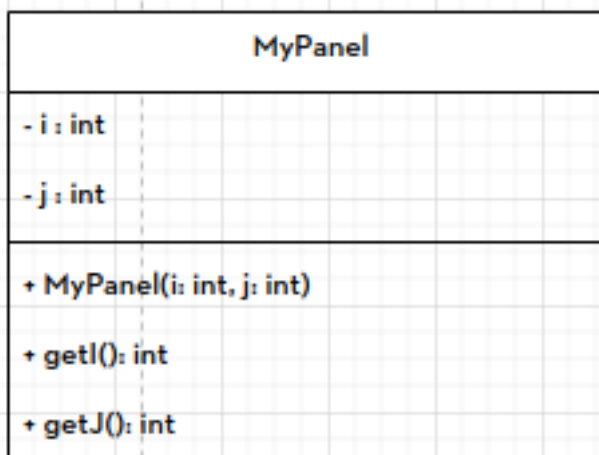


DIAGRAMMES DE CLASSES

Classe ChoixData :

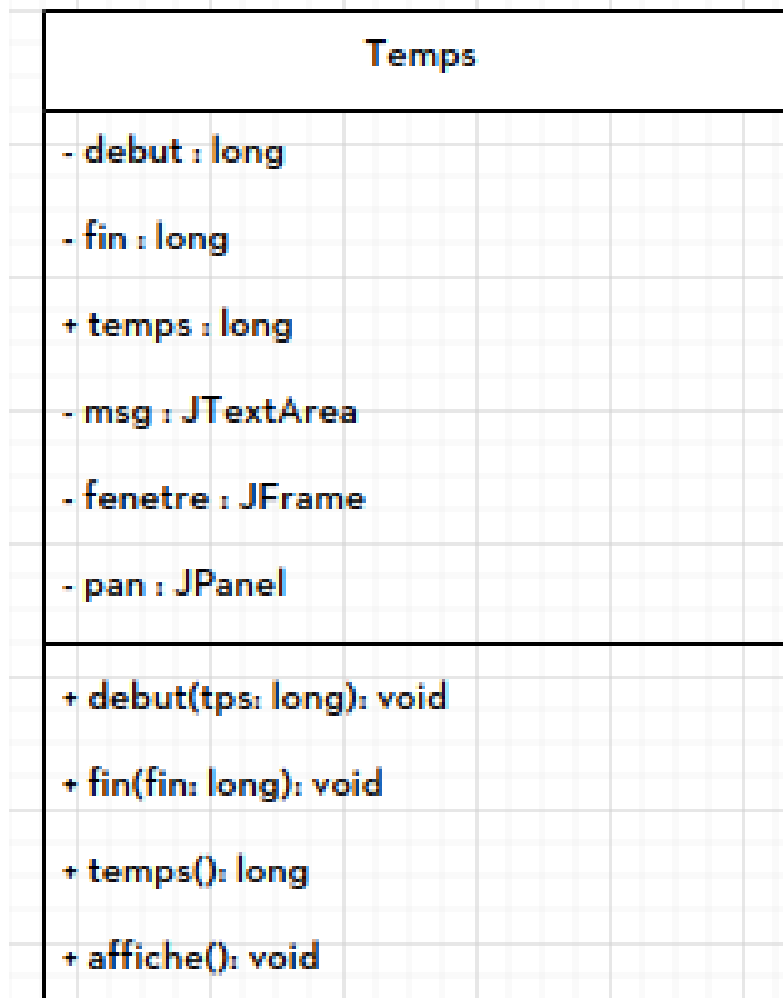


Classe MyPanel :

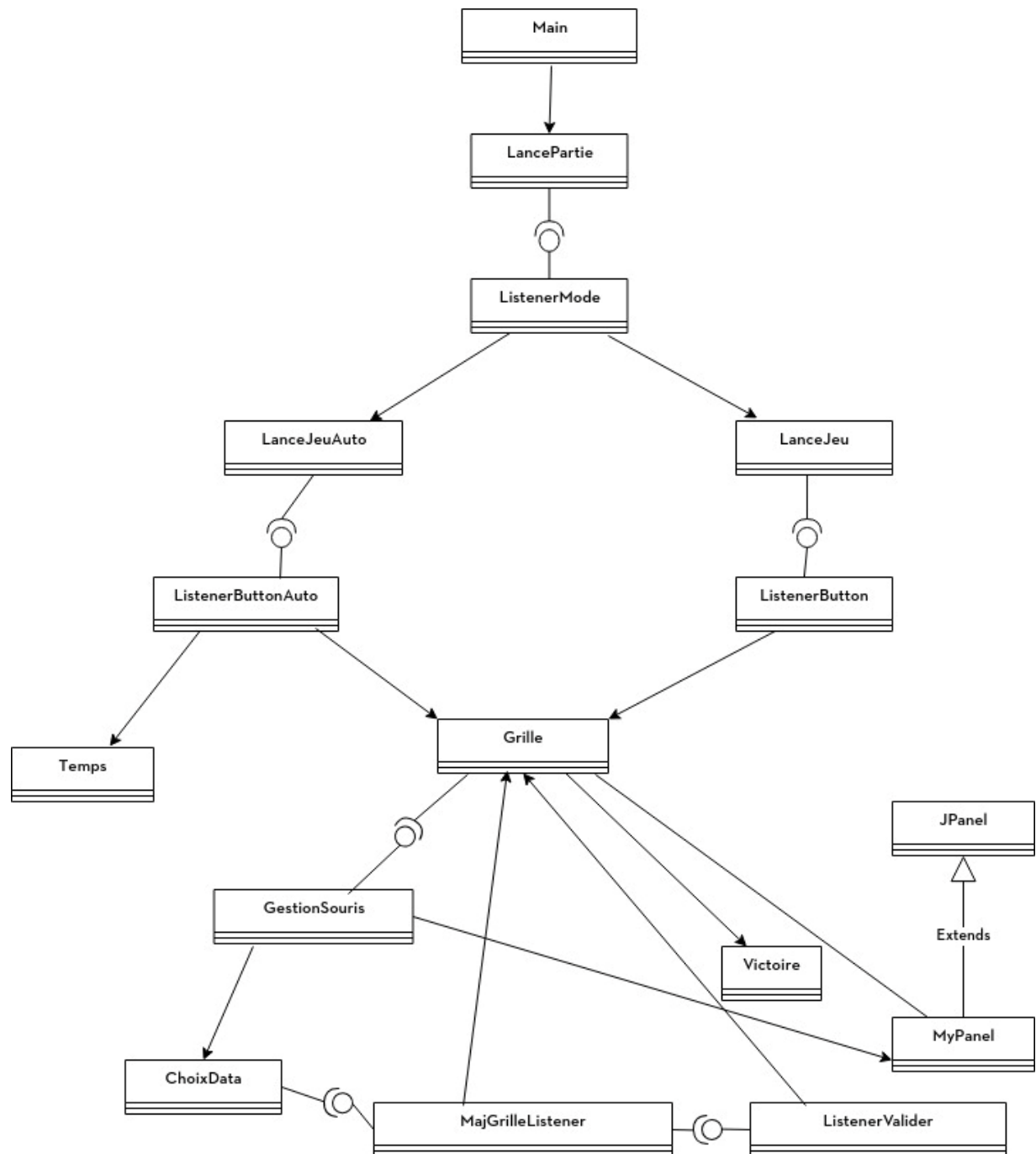


DIAGRAMMES DE CLASSES

Classe Temps :



DIAGRAMMES DE CLASSES



CONCLUSION

IV. Algorithme de résolution du jeu

Notre algorithme est une méthode qui va essayer d'ajouter des chiffres dans chaque case vide en respectant les règles du sudoku.

Tout d'abord, on teste si on arrive à la 9e ligne et 9e colonne, si c'est le cas, la méthode renvoie une valeur qui arrête l'algorithme pour éviter de retourner en arrière.

Ensuite, on teste si on est à la dernière colonne, si c'est le cas, on passe à la prochaine ligne et on retourne à la première colonne.

Puis, à chaque fois, on teste si la valeur de la case est différente de 0, si c'est le cas (la case n'est donc pas vide), on passe à la case suivante.

A chaque case, on vérifie que la valeur n'est pas déjà présente dans la ligne, la colonne, ou le carré de 9 grâce à une autre méthode. Si c'est bon, on met la valeur dans la case.

Puis on vérifie si on peut mettre une valeur dans la case d'après. Les valeurs sont testées de 1 à 9 pour remplir la grille à chaque ligne.

V. Conclusion

Nell Telechea :

Pour ma part, au début plus le projet grossissait, plus j'avais du mal à avancer dans la programmation du jeu. J'ai fini par me rendre compte que je n'avais pas vraiment saisi le concept de la programmation objet et que je me référais trop au langage impératif qu'est le C. Le projet m'a donc permis de m'améliorer en Java. En dehors de ces péripéties, cela m'a quand même plu de voir le jeu prendre forme et à la fin de vraiment pouvoir jouer à un sudoku.

Benjamin Bribant :

De mon côté, je pensais que le délai que nous avions était grandement suffisant et pensais que nous n'avions pas besoin de nous dépêcher, mais plus nous avançons et plus j'ai compris que la taille du travail était bien plus grosse que je ne le pensais. Nous avons d'ailleurs terminé juste dans les temps. En dehors de cela, j'ai quand même apprécié travailler sur ce projet, qui m'a permis de mieux comprendre le Java.

Pour conclure cette conclusion, comme vous l'aurez peut être remarqué, il manque une fonctionnalité demandée: celle des chiffres incertains. Après avoir lu nos conclusions, vous l'aurez sûrement deviné: nous n'avons pas su gérer notre temps correctement pour tout faire.