

Annee 2019 - 2020
Departement Informatique
Groupe 1

Projet Tutoré 2019 – 2020



Louis Brunet & Thibault Barbieri

SOMMAIRE

| | |
|--|---|
| <u>I) Introduction</u> | 2 |
| <u>II) Description des fonctionnalités</u> | 3 |
| <u>III) Structure du projet</u> | 4 |
| <u>IV) Présentation algorithme de résolution</u> | 6 |
| <u>V) Conclusions personnelles</u> | 7 |

I) Introduction

Le sudoku est un jeu consistant à placer un chiffre de 1 à 9 dans chaque cellule d'une grille de 9 x 9 cellules. Au début d'une partie, certaines cellules sont d'ores et déjà remplies et ne peuvent être remplacées par le joueur.

On ne peut pas placer un chiffre dans une cellule si ce même chiffre est déjà présent dans la même ligne, colonne ou le même groupe de 9 cellules que la cellule en question. Une ligne est complète lorsque l'ensemble des cellules qui la composent sont remplies avec les chiffres 1, 2, 3, 4, 5, 6, 7, 8, et 9, même principe pour les colonnes et les groupes carrés de 9 cellules.

La partie est finie lorsque l'ensemble des cellules sont remplies correctement.

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

II) Description des fonctionnalités

Lors du lancement du jeu, on accède à une fenêtre donnant le choix entre consulter les règles ; importer une grille de sudoku à résoudre ; ou bien créer une grille soit même :

- Construction :

une interface se lance et permet au joueur de placer les nombres souhaités sur une grille 9x9 cellules tout en respectant les contraintes du sudoku et ensuite de la sauvegarder dans un fichier. Un bouton retour permet de revenir au menu de départ afin de pouvoir directement accéder à la grille créée.

-Résolution :

le joueur sélectionne une grille de sudoku à résoudre parmi celles disponibles (des grilles déjà générées ainsi que celles construites manuellement), celle-ci se charge puis la partie commence.

Lorsque la partie commence, le joueur a devant lui une grille incomplète qu'il doit compléter en suivant les règles du sudoku avec maximum 4 cellules complétées en même temps dans une même case, lorsque toutes les cellules sont remplies c'est à dire que toutes les nombres insérés dans les cellules respectent les contraintes le joueur a gagné et un écran de victoire s'affiche avec un bouton pour retourner au menu.

Le joueur peut également choisir de laisser l'algorithme résoudre la grille sélectionnée, celui ci affichera à la fin le temps qu'il a fallu pour la finir.

III) Structure du projet

Le diagramme de classes simplifié ne peut pas être lisible sur une page d'un PDF. Il est néanmoins disponible aux formats *.mdj* et *.JPG* sur le dépôt Gogs de ce projet (les fichiers *classes.mdj* et *classes.JPG*).

Ce diagramme a été créé dès le début du projet et mis à jour dans la continuité de l'avancement global du projet. On se basera dessus afin de présenter la structure du projet.

Grille : contient toutes les cellules et gère les contraintes de placement des chiffres

Cellule : contient les caractéristiques d'une cellule

Position : gère la position de chaque cellule

ObservableModel : relie Cellule, Grille et GameModel

ObserverView : vue reliant Cellule et CellulePanel ainsi que Grille et GrillePanel

CellulePanel : gère les cellules

GrillePanel : gère la grille

les classes GameModel, Game screen et ButtonPanel sont des classes abstraites implémentées différemment suivant si on est dans l'interface de résolution de grille ou bien dans celui de construction de grille.

GameModel : sert à modéliser une série d'interactions avec une grille de sudoku et ses actions vont dépendre de l'interface dans lequel on se trouve (résolution ou construction).

GameScreen : affiche l'interface de jeu

SidePanel : gère le numpad servant à remplir les cellules ainsi que les boutons

utilitaires.

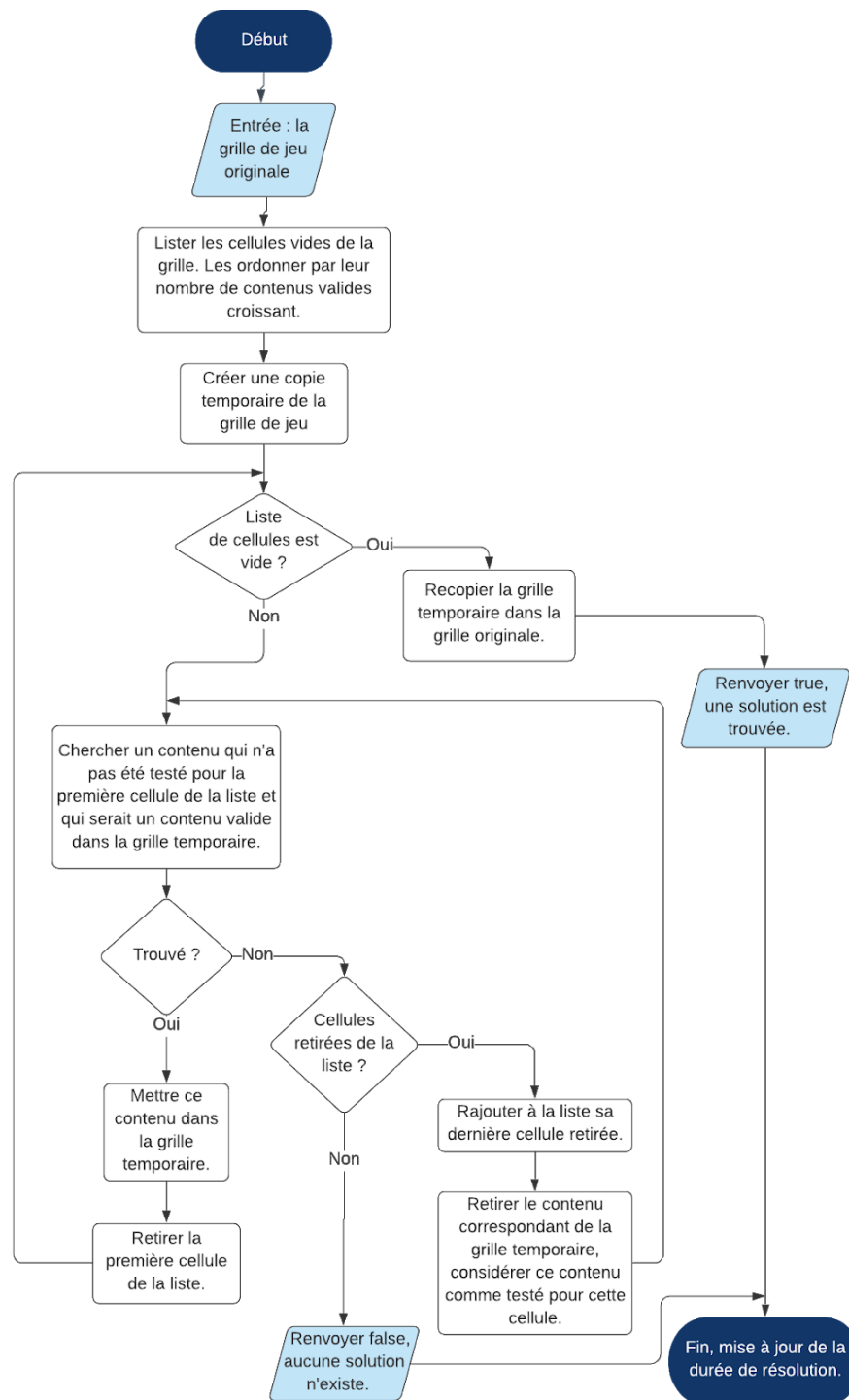
ContentControllerPanel : contient les ActionListener de chaque cellule ainsi que des chiffres.

ButtonPanel :

- dans l'interface construction, les boutons "importer grille" ; "retour"; "exporter" et "réinitialiser" sont affichés.
- dans l'interface résolution, les boutons "importer grille" ; "retour"; "résolution automatique" et "incertitude" sont affichés.

IV) Presentation algorithme de résolution

La fonction `resoudre` de la classe `Resolution` fait appel à la fonction récursive `solved` de la même classe. Le diagramme suivant explicite l'algorithme utilisé durant un appel à `resoudre`.



Entrée :

Une grille de sudoku représentée par la classe `Grille`.

Sortie :

La durée de résolution est mise à jour.

Si la grille donnée en entrée est résoluble:

- modifier son contenu pour qu'elle soit résolue,
- renvoyer `true`.

Sinon, aucune solution n'est possible, renvoyer `false`.

V) Conclusions personnelles

Thibault Barbieri : Ce projet a été grandement facilité par une anticipation des classes à faire ainsi que le dialogue entre nous qui a permis une meilleure compréhension des consignes et de la manière de faire les classes. Le plus dur a été de diviser le projet en chaque classe ainsi que l'algorithme de résolution.

Louis Brunet:

J'ai apprécié travailler sur ce projet, car je ne pensais pas pouvoir réutiliser autant de classes que l'on a pu le faire pour le réaliser. Après réflexion, nous avons remarqué que les deux affichages qu'on allait utiliser étaient très similaires. C'est à ce moment que nous avons fait abstraction de beaucoup de données et classes d'affichage qui ont pu être utilisées pour les deux modes. Notamment, les classes GameModel et GameScreen nous ont permis une progression rapide, car la logique d'interaction et d'affichage d'un mode de jeu a bien été définie dès le début. A mon avis, l'algorithme de résolution, bien qu'il marche comme prévu, est très optimisable. Nous aurions sûrement dû ne pas réutiliser les types Grille et Cellule dans la fonction récursive de résolution, car ils entraînent avec eux beaucoup d'informations inutiles à l'algorithme de résolution.