UNIVERSITE DE ROUEN

14 février 2017

L3 Informatique

Projet d'application informatique

Résolveur de sudoku
Rapport

Auteurs :
Loïc Frétard
Paul Hardouin
BOUCHER Laëtitia
BRUN Florian

Responsables :
M. Guesnet

Table des matières

Introduction

Objectif

Ce projet à pour but de réaliser une modélisation d'un résolveur de sudoku en langage Java.

L'origine du sudoku

Le sudoku a été inventé en 1979 par Howard Garns, un pigiste spécialisé dans les puzzles, et publié cette même année pour la première fois dans Dell Magazines sous le nom de Number Place. Après avoir été introduit au Japon, le nom devient Sudoku. En 2004, Le Times publie une première grille puis les autres journaux suivent. Depuis, le phénomène a fait le tour du monde et est arrivé en France. Inspiré du carré latin de Leon- hardt Euler, le but du jeu est que chaque ligne, colonne et région de 3x3 cases contienne chaque chiffre de 1 à 9 une seule fois.

Actuellement, il existe de nombreuses variantes pour le sudoku allant de la plus simple à la plus complexe. Comme exemple, nous avons : changer la taille de la grille et ne plus prendre systématiquement 3x3 cases mais 2x2cases (idéal pour apprendre, se familiariser lorsque c'est la première fois que l'on joue), ou encore 10*10 cases si on aime les défis. Parmi ces variantes nous trouverons également la possibilités de remplacer les chiffres par des symboles ainsi, à la place de compter en base 10, nous pourrions compter en base 16, en hexadécimal, de 0 à F.

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

1 Vocabulaire utilisé

Cette section a pour but de répertorier les différents mots de vocabulaire que nous emploierons dans la suite, pour la partie code, nous utiliserons les équivalents de ces mots en anglais, il sseront noté ci-dessous entre parenthèses.

— sudoku : Terme désignant le jeu en lui-même.
— grille (grid) : Il s'agit de l'ensemble des régions et des cellules.
— secteur/zone/région (sector/zone/area) : Ensemble contenant des cellules.
— cellules (cells) : Ensemble pouvant contenir des possibilités ou une valeur.
— heuristiques : Termes désignant les règles dont nous nous servirons pour créer nos algorithmes.

2 Fonctionnalités

2.1 Fonctionnalités obligatoires

2.1.1 Model

1. prise en charge de grilles de dimensions variables : L'utilisateur peut demander à avoir une grille de n cases, ainsi, si n vaut 2, l'utilisateur pourra utiliser une grille de deux cases sur deux, offrant une grille de 4 cases au total.
2. fichier de chargement pour une nouvelle grille : L'utilisateur peut demander le chargement d'une grille afin de pour pouvoir effectuer l'une de deux actions suivantes : la compléter ou la créer afin d'avoir une nouvelle grille.
3. sauvegarde/chargement d'une partie : Afin de procéder à une sauvegarde, l'utilisateur peut enregister la grille actuelle dans un fichier et, peut la récupérer ensuite gràce à une option de chargement.
4. réinitialisation de la grille : Option permettant de remettre la grille de sudoku telle qu'elle était à son chargement.
5. vérification de la victoire automatique : Système permettant au joueur de savoir si son choix est correcte ou non de manière automatique sans que ce dernier ait besoin de cliquer sur un bouton ou une touche.
6. vérification des conflits à la demande de l'utilisateur L'utilisateur peut demander à ce que l'on vérifie les différents conflits présents dans la grille en cliquant sur un bouton.
7. modification d'une cellule : Une cellule devra être modifiable afin de pouvoir ajouter/retirer des valeurs.

8. valeur définitive : Dans une cellule, on pourra fixer une valeur définitive qui ne sera plus modifiable par la suite.
9. élimination des possibilité pour la valeur sur la ligne, colonne et région sur lesquels se trouve la cellule : On pourra éliminer plusieurs possibilités d'une cellule par rapport à une région.
10. possibilité : Valeur supposé valide d'une cellule dans l'attente de la faire passer en valeur définitive.
11. effacement : Permet d'effacer une cellule, de la remettre à zéro.
12. demander l'aide : Bouton permettant à l'utilisateur de bénéficier d'une aide afin de résoudre la grille en donnat des indications permettant de ne pas rester bloqués.
13. expliquer la règle : Explication d'une règle de résolution.
14. appliquer la règle : Permet d'utiliser la règle choisie.
15. résoudre la grille pas à pas : Système permettant une résolution de grille de manière pas à pas, c'est-à-dire en résolvant cellule après cellule en appliquant différents algorithmes nécéssitant l'utilisation de règles qui seront indiqués.
16. résoudre complètement : Solution complète de la grille de sudoku.
17. si la grille ne peut pas être résolue : Proposer une réinitialisation avant résolution en indiquant que la grille est irrésolvable.
18. classement par difficultés : Grille classés en fonction des heuristiques nécessaires permettant d'évaluer un système de niveaux.

2.1.2 Vue

1. raccourcis clavier : Gestion des appuies simultanés de touches du clavier générant des possibilités inscrites dans le menu tel les sauvegardes/chargements de nouvelles grilles, l'annulation/réactivation d'une action etc.
2. modification d'une cellule : Une cellule devra pouvoir être modifiable et ceci de façon visible commepar exemple grâce à un changement de couleur ou un changement de taille.
3. clic droit : Sur une possibilité, met la valeur de la cellule à cette possibilité, tandis que sur une valeur, efface la valeur et affiche les possibilités.
4. clic gauche : Sur une possibilité, active/désactive la possibilité tandis que sur une valeur, ne fait rien.
5. demande d'aide : Système permettant à l'utilisateur de bénéficier d'une aide en cas de problème.

6. mettre en surbrillance la/les case(s) concernées : Permet de visualiser les cases concernés grâce à un système de surbrillance mettant en avant certaines cases.
7. surface de la grille fixe :

2.2 Fonctionnalités facultatives

2.2.1 Model

1. timer : Système permettant à l'utilisateur de connaître le temps qu'il a mis à résoudre une grille.
2. pause : Arrête momentanément le timer, ce dernier reprendra son décompte lorsque l'utilisateur aura de nouveau cliquer sur ce bouton.
3. compteur de coups : Indique le nombre de coups joués sur une grille.
4. éditeur/générateur : Possibilités pour l'utilisateur de créer ses propres grilles selon des niveaux de difficultés.
5. samourai : Système de jeu impliquant 5 grilles de sudoku formant une grille géante disposé de telle sorte à ce qu'une grille centrale soit partagé par les 4 autres grilles disposés au coin de cette dernière.

2.2.2 Vue

1. tutoriel : Guide pour tous ceux qui désirent (ré)apprendre comment résoudre un sudoku, quelles sontles règles, comment jouer etc.
2. masquer la grille pendant la pause : Lorsque l'utilisateur appuyera sur le bouton pause, ce dernier verrouillera la session de jeu et ainsi, il ne sera plus possible de jouer tant que l'utilisateur n'aura pas repris le jeu en appuyant sur le bouton reprendre.

3 Implantation

Pour implanter notre sudoku, nous avons choisit de fonctionner selon le modèle MVC, c'est-à-dire le système Modèle-Vue-Contrôleur.

3.1 Model

3.2 Présentation du paquetage

Notre modèle se compose d'une interface ISudoku qui contient toutes les opérations standards nécéssaire à la jouabilité de notre application.

Pour modéliser la grille de jeu, nous avons créer une interface IGrid et, de la même façon, afin de concevoir les cellules, nous avons créer une interface ICell puis, deux autres interfaces afin de différencier les cellules modifiables, des cellules non-modifiables.

Afin de gérer les différentes exceptions pouvant survenir lors d'une phase de jeyu, nous avons créer une classe énumération EnumError ;

3.3 Présentation des différentes interfaces et leurs spécifications

3.4 ISudoku

3.5 IGrid

3.6 ICell

3.7 IModifiableCell

3.8 UnmodifiableCell

3.9 EnumError

3.10 Vue

3.11 Contrôleur