

Enseignant : Jin-Kao HAO

Résolution de problèmes

-

Mickaël FARDILHA – Raphaël PILLIE



Sommaire

INTRODUCTION	3
I – STRUCTURE DU PROJET	3
II – PRESENTATION DE L'APPLICATION	3
III – ALGORITHME DE RESOLUTION	6
IV – RESULTATS & PERFORMANCES	6
V – EXTENSIONS	6
CONCLUSION	6

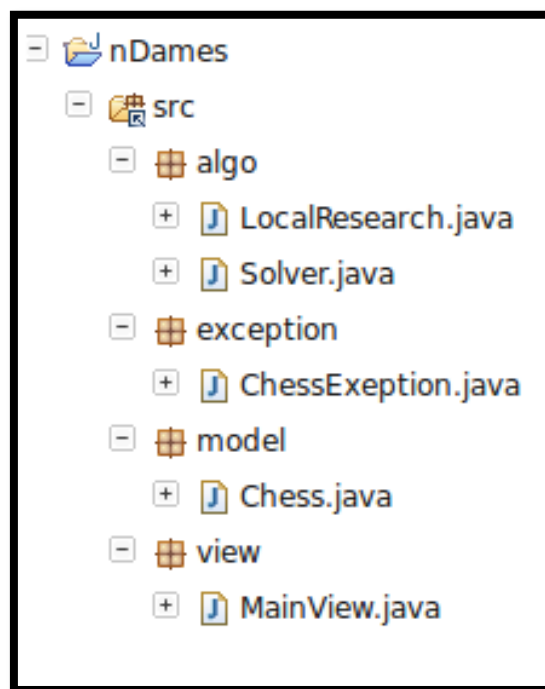
Introduction

Le master 1 informatique d'Angers propose deux options à choisir parmi quatre propositions. Pour cette première option, nous avons choisi la résolution de problèmes.

Ainsi, suite au cours qui nous a été donné, nous avons mis en place un algorithme permettant de résoudre le problème des N-dames.

I – Structure du projet

Afin de répondre le mieux possible au sujet, nous sommes passés par une phase d'analyse afin de définir une structure propre pour ce projet. En effet, nous avons découpé l'ensemble du code pour séparer la partie algorithme de la partie modèle de donnée, exceptions, ou encore de la partie interface graphique de l'application.



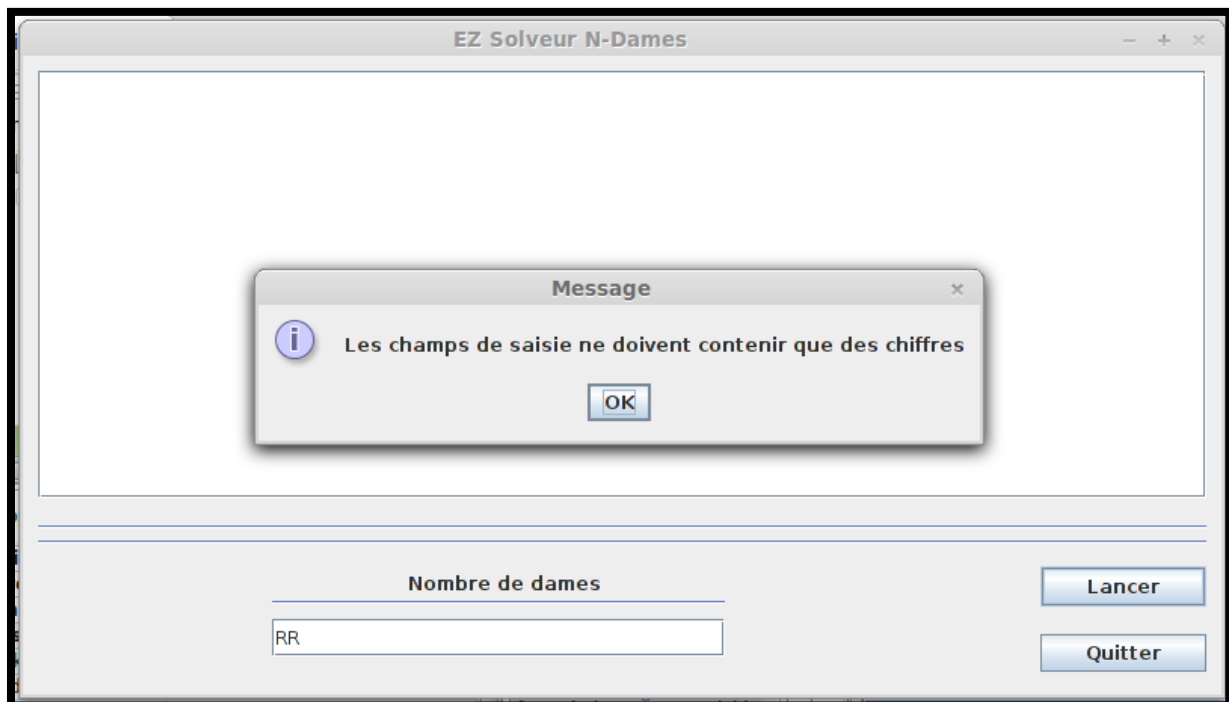
II – Présentation de l'application

L'application que nous avons mise en place possède une interface graphique permettant de lancer l'algorithme pour le nombre de dames souhaité :



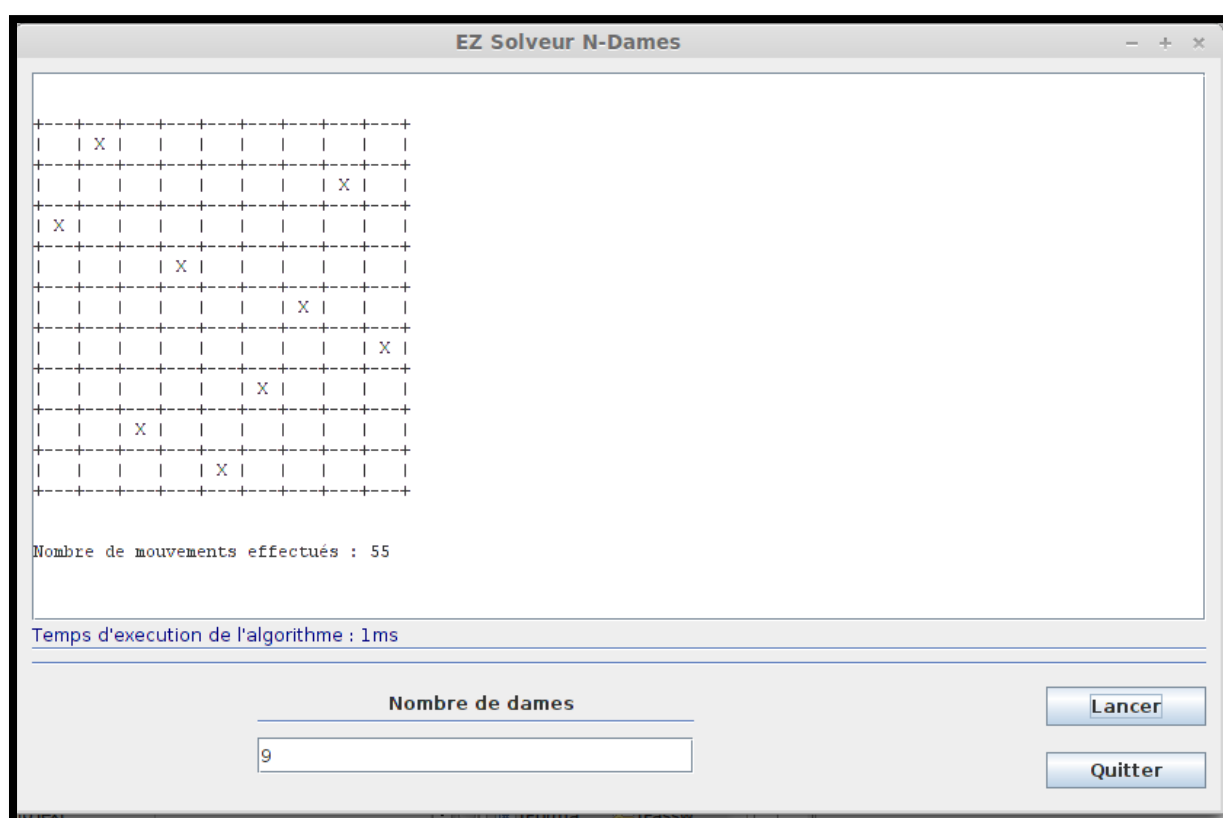
Nous avons choisi de découper cette interface de la façon suivante : la partie du haut permet de visualiser les résultats obtenus. Alors que la partie du bas permet de saisir le nombre de dames, le nombre de résultats à afficher, puis les boutons lancer et quitter permettent de lancer les exécutions des tâches qui leurs sont attribués.

De plus, nous avons mis en place un contrôle de surface permettant de vérifier l'exactitude des paramètres et le bon remplissage des champs obligatoires :





Une fois lancé, nous pouvons observer le résultat affiché :



III – Algorithme de résolution

L'algorithme que nous avons développé se déroule de manière itérative. De cette façon, il tourne tant qu'il n'a pas trouvé de solution et tant qu'il arrive à déplacer des dames. Pour ce faire, il va d'abord rechercher une dame sur l'échiquier qui génère le plus de conflit, puis va la déplacer de façon à ce qu'elle génère le moins de conflit possible. Ensuite, il recommence ce processus jusqu'à avoir une solution viable. Dans le cas où la dame que l'on souhaite déplacer ne dispose pas de solution améliorante, nous choisissons une autre dame jusqu'à ce que l'on puisse en déplacer une. Si l'on ne trouve plus de dames à déplacer, nous en choisissons une aléatoirement pour relancer l'algorithme. Afin d'éviter que l'algorithme ne provoque une boucle infinie, les choix des dames dernièrement évalué sont stocké jusqu'à ce qu'une dame soit déplacée. De plus, une dame ne peut pas être déplacée deux fois de suite. Si l'on se rend compte que l'on a en réserve un nombre de dames non jouable supérieur ou égale à la taille de l'échiquier, nous considérons alors que le problème n'a pas de solutions.

IV – Résultats & performances

L'exécution de l'algorithme augmente de façon exponentielle en fonction de la taille de l'échiquier. Il faut noter que, en fonction de la valeur que prend la valeur « N », le temps d'exécution, même s'il est exponentielle peut varier du fait de la nature aléatoire de l'algorithme. Pour illustrer cela, nous avons lancé plusieurs fois un échiquier avec mille dames et nous avons pu observer une variation du temps d'exécution, allant de trente secondes à une minute.

V – Extensions

Au cours de la réalisation de ce projet, nous avons également mis en place deux classes permettant de gérer les exceptions spécifiques à notre modèle de données. Enfin, nous avons mis en place une interface graphique afin de rendre plus facile l'utilisation de notre application.

Conclusion

Ce projet nous a permis de mettre en pratique les notions importantes vues en cours afin de nous permettre de mieux comprendre les problèmes et leurs résolutions, qui peuvent vraisemblablement devenir un enjeu majeur dans certains domaines de l'informatique.

Ainsi nous avons pu apercevoir la complexité de ce domaine, et commencer à apercevoir les techniques de résolution utilisés à ce jour.