

Rendu du projet d'intro à l'IA

Nicolas Devatine, Julien Guyot

20 avril 2018

1 Introduction

2 Description des problèmes rencontrés au cours du projet

Dans un premier temps, nous avons prévu un modèle assez optimisé, qui était réfléchi pour prendre le moins de place possible, et pour que les opérations sur le plateau soient le moins coûteuses en CPU possibles.

On en était arrivé à un état du plateau qui prenait “moralelement” une dizaine d'octets (c'est à dire hors questionnements spécifiques à java, ou l'on prendrait en compte la taille liée au header de la classe¹

Néanmoins, cette idée nécessitait d'avoir un contrôle assez fin sur les entiers, comme par exemple la gestion des entiers signés, ou une gestion fine des opérateurs bit à bit.

Malheureusement, la gestion des entiers (et/ou des short, byte) de n'a pas permis la réalisation qui était à la base prévu. Cela est dû aux spécificités suivantes :

- Les entiers java sont forcément signés — en tous cas les entiers atomiques (“int” et pas “Integer”).
- Les opérateurs bit à bit traitent uniquement les entiers (c'est à dire ni les bytes, ni les short). Par conséquent, l'opération (byte & byte)
 1. Convertit les bytes en int
 2. Applique l'opérateur
 3. Renvoie un int, qu'il faut re-caster si l'on cherche à calculer un byte

Si, dans un premier temps

¹Je ne connais pas l'implémentation du langage, par conséquent il se peut que je dise n'importe quoi

3 Modélisation du plateau

Chaque pièce est modélisée sous la forme d'un byte, qui contient les quatre caractéristiques définissant la pièce. Ce byte est de la forme
(0000 *couleur hauteur sommet forme*)

Le plateau de jeu est une matrice de byte, où chaque case est initialisée à -1. L'état du tour est matérialisé dans un byte, dont seuls les deux derniers bits sont significatifs. Celui de poids le plus faible dénote de quel type de tour on traite, le second informe de quel joueur joue.

Le plateau stocke une pièce à jouer, qui est la pièce donnée par le joueur.

4 Heuristiques