EL BAKKOURY Yassine
EL HADDADI Yassine

Projet IA - Rapport Binôme 3

LANCER LE JEU :

Pour le lancement du jeu, il faut tout d'abord placer le dossier lib (avec init et game dedans) dans le même dossier que notre jeu. Ensuite, ouvrir l'invite de commande et lancer le serveur dans le bon chemin du fichier Quarto.py avec :

$ python Quarto.py server –verbose

Nous pouvons ensuite lancer les deux clients du même jeu ou d'un jeu différent (il suffit de changer Quarto.py) :

$ python Quarto.py client –verbose PSEUDO

STRATÉGIE :

Deux stratégies sont mises au point dans notre jeu :

1. Tout d'abord, le jeu cherche à donner à son adversaire **la pièce la moins utile**, qui aura moins de chance de trouver des partenaires dans le jeu pour réaliser un quarto.
2. Ensuite, s'il y a possibilité de **réaliser un quarto**, l'IA le réalise.

IMPLEMENTATION :

Tout d'abord, la première stratégie fonctionne comme suit :

- Une **première pièce est choisie au hasard**. Cette pièce a des particularités (dans cet ordre): full/empty, dark/light, high/low, round/square. Les particularités sont analysées de manière indépendante, chaque particularité a une variable propre. Ainsi, la caractéristique est enregistrée dans une variable correspondante en ajoutant +1. Donc si notre pièce est dark, la variable D vaut désormais 1 et la variable L pour light reste nulle.

- Ensuite, une fois que le premier élément a été choisi et que le deuxième tour se lance, la nouvelle pièce est choisie de manière à

ce qu'une de ses caractéristiques soit **peu commune**. Ainsi, si la pièce précédente est full, celle-ci sera (probablement) empty.

N.B. : Les caractéristiques suivant un ordre bien précis, seul le premier minimum est choisi. Donc si light est aussi minimum, il ne sera pas pris en compte et seule une pièce empty sera donnée indépendamment qu'elle soit dark ou light.

- Enfin, quand il ne reste qu'une seule pièce possible à jouer et qu'aucun quarto n'a encore été déclaré, la dernière pièce c'est l'élément zéro, soit **move['nextPiece'] = 0**.

Enfin, la deuxième stratégie fonctionne comme telle : dans une boucle **while** allant jusqu'à i=15 en partant de i=0, **la position i est mise dans le move['pos'].** Ainsi, en parcourant chaque case représentée par i, l'IA vérifie qu'un quarto puisse éventuellement se produire en y mettant la pièce choisie préalablement par l'adversaire. Si un quarto est trouvé, le programme applique le mouvement et le jeu s'arrête.