Rapport de Liu FRANCOIS

Proffeseur: Jean-Jacques BOURDIN

Contents

1 Le jeu AWAlE **2**
1.1 Les régles 2
1.2 Le programme 2

Chapter 1

Le jeu AWAlE

1.1 Les régles

Le jeu awalé est un tour par tour a jouer a 2 ou chaque joueur a 6 case en rangées devant lui et sur les côtés il y a un grenier, le grenier étant le nombre de points de chacun. Au début du jeu dans chaque case, hormis les greniers, il y a 4 graines. Chaque tour on choisit une de ses cases et on prend toutes les graines, puis on ajoute une a toutes les cases suivantes dans le sens antihoraire sauf dans le grenier du joueur adverse et dans la case initiale si on fait un tour complet. Si la dernière graine qu'on donne est sur notre ranger, on gagne les graines de la case adverse en face de la nôtre que l'on va rajouter à notre grenier. Si la dernière graine qu'on donne est sur la rangée adverse et si elle a 2 ou 3 graine on ajoute ces 2 ou 3 graines dans notre grenier puis on vérifie la case d'avant avec ces même conditions et on répète cela jusqu'à qu'on tombe sur une case qui n'a pas 2 ou 3 graines ou si on est plus sur la rangée adverse. A la fin de notre tour si l'adversaire n'a plus de graines le tour est joué mais le plateau revient à l'état où il était avant ce tour car il ne faut pas affamé l'adversaire. Si l'adversaire n'a plus de graines a la fin de son tour, on est obligé de le nourrir et de mettre au moins 1 graine sur une de ces case pendant mon tour, si aucun coup ne permet de donner des graines la partie s'arrête et les graines restantes vont dans le grenier a qui appartient la ranger avec des graines. Pour gagner il faut avoir 25 graines ou plus.

1.2 Le programme

J' ai une structure noeud qui contient plateau, un tableau de 14 éléments qui est l'état de la partie. Et un tableau de 6 noeud qui sont les noeud des coups possibles par rapport au plateau.

Le plateau est un tableau de 14 élément, je l'ai affiché comme ci dessous ici les nombre sont les indices du tableau, affiché comme ça nous permet que le sens antihoraire soit l'indice

d'après dans la tableau.

La fonction affiche_plateau va afficher comme ci-dessus mais les valeurs à la place des indices. La fonction tour prend en paramètre le plateau, le joueur qui joue entre 0 et 1 et l'indice de la case qu'il veut jouer. On commence par vérifier que la case choisie n'est pas vide, on commence par semer les graines dans l'autre cases, on vérifie si on est sur une de nos cases, si oui on prend les graines de la case en face. Si on est sur une case adverse on fait un boucle while pour qu'on récupère les case avec 2 ou 3graines et regarder les case d'avant avec la même condition while puis on vérifie qu'on a pas affamé notre adversaire sinon on remet le tableau comme on l'a trouvé au début de la fonction.

La fonction verif_fin comme sont nom l'indique elle vérifie sur la partie est terminée, les cas possible sont qu' un joueur a plus de 25 points ou qu' un joueur ne peut pas jouer. On commence par vérifier les 2 greniers qui sont les indice 0 et 7 ou on renvoie 1 si un des 2 est supérieur ou égal à 25 puis on fais un while qui si toute une rangée est a 0 on essaye tous les coups que le joueur peut faire si nb == 7 alors on renvoie 1.

La fonction remplir_tableau qui prend en paramètre le plateau, la profondeur et le joueur. On commence par regarder si la profondeur == 0 ou si la partie est terminée avec la fonction verif_fin.
En théorie il y a maximum 6 coup possibles en partant d'un plateau, on a plateau_temp qui est un 1 tour, puis on vérifie si plateau->plateau et plateau_temp sont identique, si il sont identique alors on le coup joue dans plateau_temp etait un 0. On vérifie si le coup joue n'est pas 0 avec diff et que la rangée du joueur 1 est vide et que c'est la tour du joueur 0 et que la rangée du joueur 0 est vide et que c'est la tour du joueur 1 donc que aucun des 2 joueurs n'a affamé l'autre on ajoute dans plateau->liste le plateau_temp et on ajoute 1 à plateau->liste.

La fonction minimax qui prend en paramètre le plateau, le joueur qui doit jouer et à profondeur, minimax et une fonction récursive ou si la liste d'un noeud est vide ou si la profondeur est a 0 on renvoie le score donc la case 0 la case 1. Sinon si c'est au joueur 1 de joueur donc max on prend le plus grand score de la liste de noeud, si le joueur 0 donc min on prend le plus petit score.

Ai par rapport a la profondeur

Temps pour remplir l'arbre de profondeur 5 et jouer/afficher une partie avec minimax: 2725

Temps pour remplir l'arbre de profondeur 9 et jouer/afficher une partie avec minimax: 2022514

Temps pour remplir l'arbre de profondeur 10 et jouer/afficher une partie avec minimax: 12537066

On voit que pour 10 de profondeur on prend plus de 6 fois plus de temps que pour 9.