

JEU A 2 JOUEURS ET MINMAX – APPLICATION AU TIC-TAC-TOE.**Jeu du tic-tac-toe**

Vous considérerez donc des plateaux de taille NxM, et des alignements à réaliser de P pièces.

La machine joue les 'X', le joueur les 'O'.

Après l'identification de qui commence, les adversaires mettent à tour de rôle un 'X' ou un 'O' sur le plateau. La partie est terminée lorsqu'il y a P 'X' ou P 'O' alignés et contigus sur une même ligne, colonne ou diagonale. Il y a alors un gagnant.

A défaut, il y a match nul si le plateau est plein (ou si aucun alignement gagnant ne peut plus être effectué).

Par exemple, pour NxM=7x9 et P=4, si la machine commence, et au bout de 5 coups joués, on obtient :

			O		X			
			O	X	X	X	O	
				O	X			
					O			

...et le joueur ('O') est mal parti puisque c'est à la machine ('X') de jouer !!!

Conditions générales de réalisation du projet*Constitution des équipes*

Equipes de 3 ou 4 personnes, selon le tableau ci-dessous :

	Groupes TD			
	A	B	C	D
Nombre d'étudiants	33	36	37	30
Equipes de 3 étudiants	3	0	3	2
Equipes de 4 étudiants	6	9	7	6

Seul un changement confirmé par la scolarité du nombre d'étudiants dans un groupe TD pourra entraîner un changement dans ce tableau. Ceci ne pourra de toute façon être décidé que par l'enseignant.

Les étudiants devront fournir la constitution de leurs équipes à leur délégué de groupe TD, qui devra lui-même envoyer la constitution des équipes à l'adresse herve.barbot@efrei.fr au plus tard le vendredi 14/02 à 23h59.

Un numéro sera fourni à chaque équipe.

Langage de programmation

C ou C++.

Rendu du travail

Par email : herve.barbot@efrei.fr

Un seul envoi par équipe.

Titre email : L3 – AD – Numéro équipe

Pièces jointes :

Code source (fichiers .c, .cpp, .h, .hpp) et traces d'exécution (fichiers .txt) uniquement.

Tous les fichiers préfixés par votre numéro d'équipe (par exemple « 09-nomdufichier.cpp »).

Traces d'exécution en copier/coller du contenu de la fenêtre d'exécution de votre programme (pas d'image, jpeg ou autre, « copie de fenêtre »).

Aucun fichier binaire (notamment, pas de fichier « projet » extrait des répertoires CodeBlocks ou Visual, pas de copie d'écran au format image).

Pas de fichier archive.

Corps du message :

Rappel des noms des membres de l'équipe ainsi que du groupe TD.

Le cas échéant, remarques sur les problèmes éventuels de votre code (dans le cas où ça ne fonctionne pas...)

TOUTES CES CONSIGNES ONT UNE RAISON D'ETRE. UN NON RESPECT D'UNE D'ENTRE-ELLES PEUT ENTRAINER PAR EXEMPLE QUE VOTRE EMAIL NE SOIT PAS TRAITE CORRECTEMENT LORS DE SA RECEPTION, ET SOIT PERDU.

TOUT MANQUEMENT A CES REGLES SERA DONC PENALISE.

1^{ère} étape – TP #1 – Minmax « basique »

Cette 1^{ère} étape se termine lors de la 1^{ère} séance de TP. Un email contenant vos premiers résultats devra être envoyé à l'adresse `herve.barbot@efrei.fr` en fin de séance.

Votre programme doit à ce stade offrir les fonctions suivantes :

- Interaction avec l'adversaire humain :
 - o Choix de la taille du plateau,
 - o Choix de qui (humain ou machine) commence la partie,
 - o Choix de la profondeur d'analyse par la machine (équivalent au choix d'un niveau de force de la machine),
 - o Affichage du plateau de jeu,
 - o Sélection du coup joué par l'humain ;
- Analyse du meilleur coup à jouer pour la machine en minmax sans élagage, à une profondeur maximale d'analyse telle que choisie par l'humain, et avec une fonction d'évaluation heuristique simple ;
- A chaque coup joué par la machine, affichage des statistiques concernant la recherche faite par la machine : nombre de positions évaluées à l'aide de la fonction heuristique (feuilles de l'arbre), profondeur d'analyse,
- En fin de partie, affichage de la liste des coups joués.

Pour être certain de pouvoir travailler et envoyer votre programme, n'hésitez pas à venir avec votre PC.

2^{ème} étape – TP #2 / soutenance – Elagage et heuristique renforcée

Une soutenance se déroulera durant la 2^{nde} séance de TP / Projet, prévue fin mars ou début avril selon les groupes. Vous devrez y présenter votre code, et effectuer une démonstration de votre programme.

A ce stade, votre programme devra avoir été amélioré par rapport à la 1^{ère} étape, et devra inclure l'élagage alpha-bêta (en plus de la profondeur d'analyse), et si possible une heuristique plus performante.

Il vous sera demandé d'envoyer votre travail avant la soutenance, à une date qui vous sera communiquée ultérieurement. Cette date sera la même pour tous les groupes TD, quelle que soit la date de soutenance.

Comme pour la 1^{ère} séance, il est aussi conseillé de venir avec son PC afin de parer à toute éventualité.

Idées complémentaires

Fonction d'évaluation

Le point essentiel du MinMax est la fonction d'évaluation d'une position du jeu.

Cette fonction rend des valeurs comprises entre « moins l'infini » (situation où le joueur gagne) et « plus l'infini » (situation où la machine gagne).

Une égalité des chances correspond à la valeur 0.

Une solution simple de mettre en œuvre cette fonction est de :

- compter le nombre de lignes, colonnes ou diagonales où il est encore possible d'aligner P 'X' contigus (Nb1) ;
- compter le nombre de lignes, colonnes ou diagonales où il est encore possible d'aligner P 'O' contigus (Nb2) ;
- et de faire la différence Nb1-Nb2.

On vérifie que si la différence est nulle, la machine et son adversaire ont bien le même nombre de chances de gagner.

Il faut bien entendu tenir compte des 'X' et des 'O' déjà placés, sinon les résultats sont faussés...

Optimisation :

Si, sur une ligne par exemple, il est possible de faire plusieurs alignements, cette ligne doit avoir une valeur supérieure à 1, puisqu'elle donne plusieurs possibilités de vaincre. On peut considérer qu'elle compte pour 2 si deux alignements peuvent être réalisés sans utiliser les mêmes cases, ce qui n'est évidemment pas possible pour un plateau 7x7 et des alignements de 4.

Une valeur comprise entre 1 et 2 (ou plus) peut être choisie en fonction des chevauchements entre les alignements.

Pour des alignements de (P=) 4 :

4 cases = 1

5 cases = 1,25

6 cases = 1,5

7 cases = 1,75

8 cases = 2

...

Quelques valeurs pour quelques configurations de lignes :

1,25					O	X		
2	X				O			
2,25	X							X
0				O			O	

On peut aussi compter le nombre d'alignements, même si ceux-ci se chevauchent. Ce qui donne pour les lignes ci-dessus : 2, 2, 6 et 0.

A vous de décider ce qui est le mieux. Pour décider, vous pouvez vous amuser à mettre votre programme en concurrence avec lui-même avec 2 fonctions d'évaluations différentes.

Calcul des alignements

Un parcours de toutes les lignes, colonnes et diagonales est possible, avec les compteurs appropriés. C'est simple pour les lignes et colonnes, et juste un peu plus compliqué pour les diagonales.

Vous pouvez aussi définir des « masques » correspondant aux alignements possibles, que vous comptabilisez si les cases correspondantes ne contiennent pas de 'O' (mais peuvent contenir des 'X' ou être vides).

Si par exemple votre plateau est représenté par un tableau à une dimension de NxM cases, les masques peuvent être (pour NxM=7x9) :

(0,1,2,3), (1,2,3,4), ... (5,6,7,8) ligne 1

(9,10,11,12), (10,... ligne 2

...

(0,9,18,27), (9,18,27,36), ... colonne 1

...

(0,10,20,30), (10,20,30,40), diagonale 1

(1,11,21,31), (11,21,31,41),... diagonale 2

...

(3,11,19,27) diagonale 7

...

(8,16,24,32), (16,24,32,40),... diagonale 12

...

Pour voir cela, se référer aux indices dans le tableau suivant des cases du plateau. Quelques exemples de masques sont illustrés par des couleurs.

0	1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16	17
18	19	20	21					26
27	28	29	30	31				35
36								
45							52	
54								62

Bien entendu, plus le plateau est grand, plus le nombre de masques à créer et utiliser est grand !