



UNIVERSITÉ  
CAEN  
NORMANDIE

RAPPORT : INFECTIONS

---

# Sécurité et Aide à la décision

---

AUBRY NICOLAS

HU ERIC

L2 Informatique  
Groupe 4B

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Algorithmique et règles du jeu</b>	<b>3</b>
2.1	Algorithmique . . . . .	3
2.2	Règles du jeu . . . . .	3
<b>3</b>	<b>Expérimentation</b>	<b>4</b>
3.1	Nombres de noeuds parcourus . . . . .	4
3.2	Potentiel de victoire . . . . .	5
3.2.1	NegaMax contre AlphaBeta . . . . .	5
3.2.2	NegaMax contre NegaMax . . . . .	5
3.2.3	Bilan rapide . . . . .	6
<b>4</b>	<b>Améliorations</b>	<b>7</b>
<b>5</b>	<b>Conclusion</b>	<b>8</b>

# 1 Introduction

En sécurité et aide à la décision, on nous a demandé de réaliser un programme qui permet à deux algorithmes de s'affronter sur un plateau en 2 dimensions. Le but étant de voir à quelle problématique pouvait répondre un certain algorithme. Nous avons choisi l'algorithme NegaMax, une forme contractée du MinMax et un algorithme AlphaBeta basé sur le MinMax.

Nous ferons donc des expérimentations sur la rapidité de ces algorithmes de recherches et leur potentiel à gagner en le faisant jouer contre un NegaMax ayant des coups d'avance.

## 2 Algorithmique et règles du jeu

### 2.1 Algorithmique

Nos algorithmes NegaMax et AlphaBeta se basent sur le score pour faire leur choix. Le score est calculé comme une différence entre nombres de cases appartenant au joueur courant et son adversaire. Cela a pour but d'inciter les algorithmes à "s'infecter entre eux" plutôt que de simplement se dupliquer. De plus, afin de souligner encore plus cet effet, lorsqu'il ne reste qu'un seul joueur sur le plateau, ce joueur se voit attribuer un bonus de score de 100 points.

### 2.2 Règles du jeu

Lorsqu'il ne reste plus de coup possible à un seul joueur uniquement, nous avons décidé, au lieu d'arrêter la partie, que le joueur auquel il reste des coups devra se dupliquer jusqu'à remplir l'intégralité du tableau. En implémentant ceci, nous avons espéré qu'avec une profondeur suffisamment grande, que si un joueur ne peut pas entièrement infecter l'autre, il aurait pour but de bloquer ses mouvements pour remplir toutes les cases libres et ainsi de créer ou de creuser un écart de score.

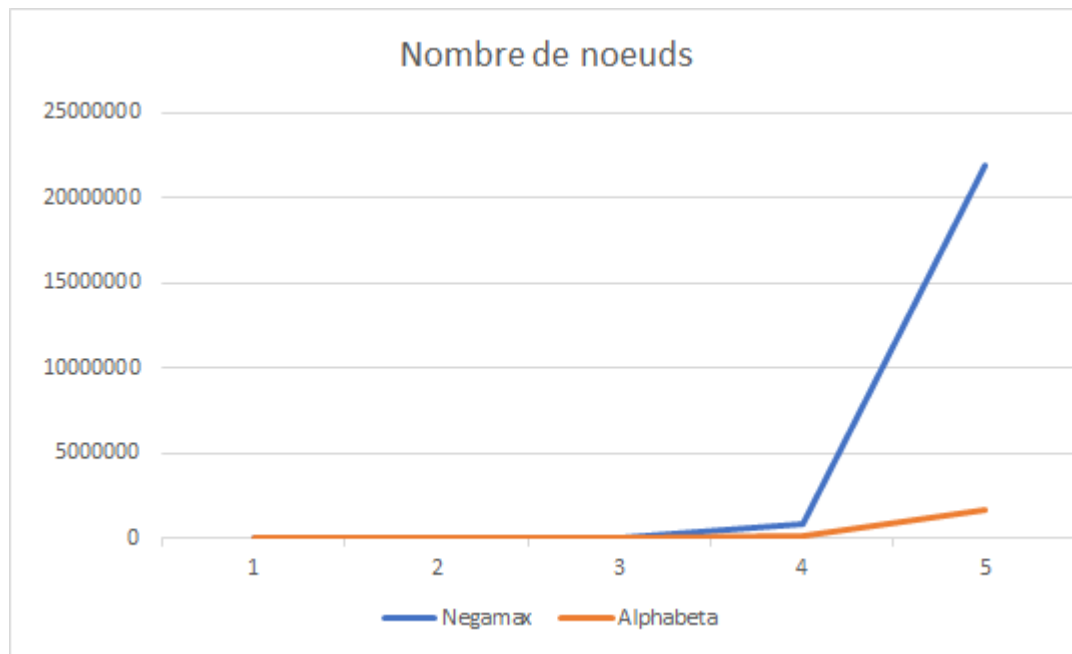
### 3 Expérimentation

Pour la première expérimentation, nous testerons notre programme avec une grille 6X6 et nous compterons le nombre de nœuds parcourus par Alpha-Béta et NémaMax, afin de comparer les résultats entre les deux.

Dans une seconde expérimentation, nous regarderons, en donnant des coups d’avance au joueur blanc et une profondeur de plus en plus grande au joueur noir, à partir de quand le joueur noir peut gagner. Dans cette expérimentation, le joueur blanc est un NémaMax avec une profondeur de 1 et le joueur noir est soit un NémaMax soit un AlphaBeta.

#### 3.1 Nombres de nœuds parcourus

FIGURE 1 – Graphique pour le parcours de nœuds :



On peut remarquer que l’algorithme NémaMax a un nombre de nœuds qui monte de façon quasi-exponentielle tandis que pour AlphaBeta le nombre de nœuds augmente de façon plus linéaire. C’est normal, car AlphaBeta va élaguer les branches qu’il trouve inutile et de ce fait, va éviter le parcours de

ces nœuds, ce qui n'est pas le cas de NegaMax qui va parcourir tous les nœuds. On peut donc conclure que plus la profondeur souhaitée est importante, plus il est intéressant d'utiliser AlphaBeta.

## 3.2 Potentiel de victoire

### 3.2.1 NegaMax contre AlphaBeta

FIGURE 2 – Tableau de résultats entre profondeur et coups d'avance avec un NegaMax de profondeur 1 contre un AlphaBeta :

Profondeur Pour le joueur noir		Coups d'avance pour le joueur blanc								
		1	2	3	4	5	6	7	8	
	1									
	2									
	3									
	4									

Le résultat de ce tableau est très peu équilibré, notre joueur noir (ici un AlphaBeta, ne gagne qu'une seule fois). Nous pensons que cela est dû au fait qu'il élague beaucoup trop et se retire ainsi la possibilité de jouer des coups qui le ferait gagner.

FIGURE 3 – Tableau espéré des résultats entre profondeur et coups d'avance :

Profondeur Pour le joueur noir		Coups d'avance pour le joueur blanc								
		1	2	3	4	5	6	7	8	
	1									
	2									
	3									
	4									

On pourrait plus s'attendre à ce genre de résultats. Notre premier tableau est donc erroné, nous n'avons pas les résultats attendus. Néanmoins, le nombre de nœuds parcourus par chacun des algorithmes est cohérent.

### 3.2.2 NegaMax contre NegaMax

Lorsque l'on compare les scores, on peut voir plusieurs fois que l'écart entre les deux est supérieur à 100. Cela veut dire que le joueur noir s'est fait

FIGURE 4 – Tableau de résultats entre profondeur et coups d’avance avec un Negamax de profondeur 1 contre un NegaMax :

Score : joueur1 / joueur2	Coups d’avance pour le joueur Blanc							
Profondeur pour le joueur Noir	1	2	3	4	5	6	7	8
1	28/-28	135/-35	28/-28	135/-135	135/-35	135/-135	28/-28	135/-35
2	4/-4	10/-10	110/-10	112/-12	110/-10	112/-12	110/-10	112/-12
3	6/-6	-4/4	6/-6	16/-16	6/-6	26/-26	26/-26	8/-8
4	12/-12	14/14	26/-26	14/-14	18/-18	10/-10	16/-16	24/-24

totalelement "infecter". Et que le "comportement" que nous avons donné aux algorithmes est bien respecté.

Cependant, plus le joueur noir a une profondeur importante, plus il est difficile pour le joueur blanc d’anéantir complètement son adversaire.

Comme pour AlphaBeta, on aurait pu s’attendre à un tableau plus "équilibré".

### 3.2.3 Bilan rapide

En regardant les tableaux NegaMax contre AlphaBeta et NegaMax contre NegaMax, on se doute que donner des coups d’avance est bien trop fort, car même en ayant une profondeur bien supérieure, le joueur noir ne gagne jamais. Mais, que ce soit un NegaMax ou un AlphaBeta, le joueur noir gagne lorsque qu’il a une profondeur de 3 et son adversaire a 2 coups d’avance. Mais il ne gagne pas avec une profondeur de 4. Nous pouvons alors nous demander s’il y a une configuration spéciale dans ce cas-là. Dans le cas général, le joueur noir essaie de diminuer l’écart de score, il essaie de "gagner le plus possible", surtout lorsque c’est un NegaMax. Ceci est déjà un exploit, surtout lorsque le joueur blanc commence avec plus de 5 coups d’avance, car il occupe alors au moins 1/6 des cases du jeu avant que le joueur noir ne puisse commencer à jouer.

## 4 Améliorations

Nous avons rencontré plusieurs problèmes lors de l'élaboration de notre jeu. Le premier est que nous ne gérons pas l'effet de bord, c'est-à-dire que dès que les deux algorithmes vont s'affronter et vont se retrouver dans une position où ils vont chacun jouer le coup qui leur semble le meilleur, mais ne feront que rejouer les mêmes coups en boucle. D'une autre part nous pensons que notre NegaMax n'est pas aussi "puissant" qu'il devrait être, en effet lors de différents tests contre un RandomPlayer, nous avons remarqué qu'il perdait dans les cas où nous avions fait en sorte qu'il gagne. Heureusement, ces cas sont rares. Et finalement, nous aurions pu mettre un "cache" pour que ni NegaMax ni AlphaBeta rejouent des suites de coups qui donnent un même "state" afin d'améliorer la rapidité d'exécution du programme.



## 5 Conclusion

Nos algorithmes de recherche ne gagnent jamais contre un algorithme avec un ou plusieurs coups d'avances. Le fait d'apprendre à utiliser les algorithmes de décisions comme AlphaBeta ou NegaMax permet de comprendre les stratégies utilisées et de voir aussi la complexité que prend juste un plateau 6X6 en terme de choix de coups possible. Après nos expérimentations, nous pouvons déduire qu'Alphabeta est plus intéressant si nous voulons un algorithme rapide. Tandis que NegaMax sera plus intéressant si l'on veut un algorithme qui gagne le plus possible.