

# TRAVAUX PRATIQUES INTELLIGENCE ARTIFICIELLE ET JEUX ADVERSARIAUX EN SECURITE ET AIDE A LA DECISION

## JEU D'INFECTION

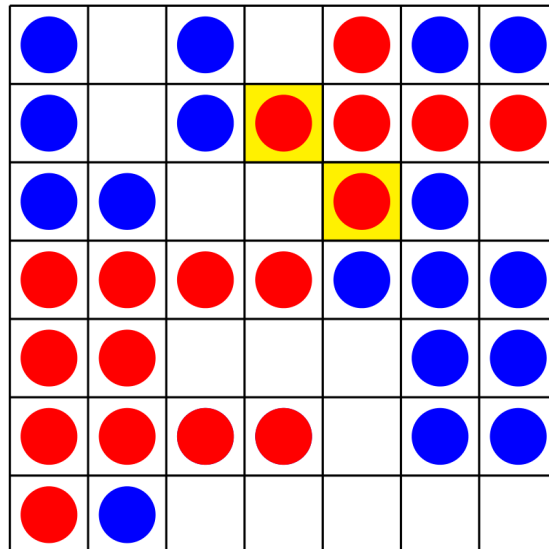
VU Nguyen Phuong Vy - 21911658

BOUCHHEFA Badis - 21914662

ADOUMA Hassan Brahim - 21901741

BENELAM Kamel - 21913417

1<sup>er</sup> mars 2021



# Table des matières

0.1	Introduction . . . . .	2
0.2	Diagramme . . . . .	2
0.3	Expérimentation . . . . .	3
0.3.1	minmax . . . . .	3
0.3.2	alphabeta . . . . .	4
0.4	Raisonnement de la profondeur . . . . .	5
0.5	Conclusion . . . . .	5

## 0.1 Introduction

Après avoir réalisé le jeu et d'implémenter tous les expérimentations nécessaires nous vous expliquons le déroulement du jeu et de ces expérimentations.

## 0.2 Diagramme

Voici notre diagramme de classe UML, Afin de bien simplifier et de comprendre notre code :

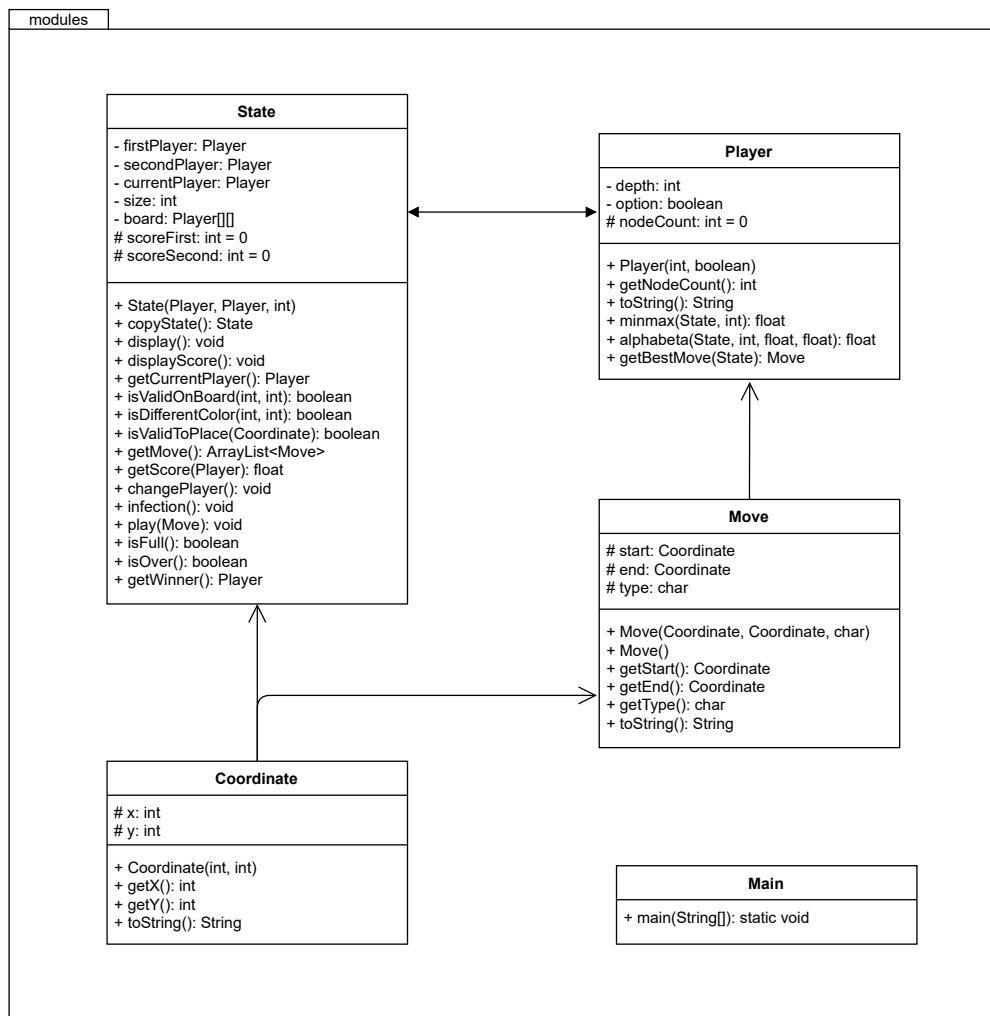


FIGURE 1 – La photo du diagramme

## 0.3 Expérimentation

L'expérimentation du jeu de ce jeu nécessite deux algorithmes très importants à signaler qui sont :

### 0.3.1 minmax

L'idée de cet algorithme est de développer complètement l'arbre de jeu, de noter chaque feuille possède sa propre valeur, puis de faire remonter ces valeurs avec l'hypothèse que chaque joueur choisit le meilleur coup pour lui.

L'algorithme minmax amène donc à développer un nombre exponentiel des branches selon la profondeur à laquelle on s'intéresse.

Le graphe ci-dessous représente le nombre de nœuds explorés par minmax :

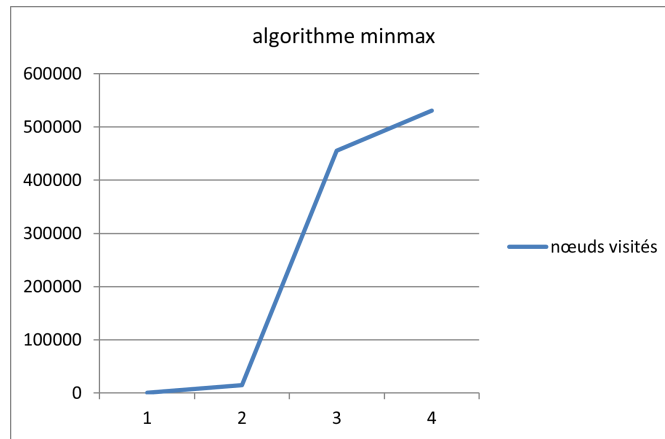


FIGURE 2 – Courbe de l'algorithme minmax avec profondeur à 4

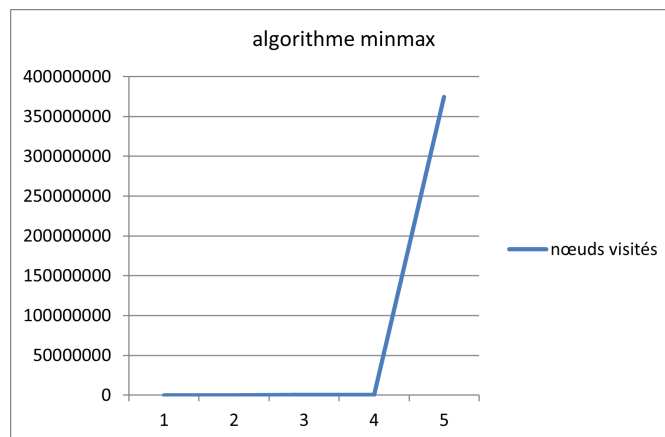


FIGURE 3 – Courbe de l'algorithme minmax avec profondeur à 5

### 0.3.2 alphabeta

Cet algorithme n'est autre qu'une amélioration du minmax sur sa version negamax et l'idée de cet algorithme est d'élaguer certaines branches de l'arbre, car elle coupe les branches inutiles de l'arbre qui sont pour le pion concerné des branches perdantes.

Le graphe suivant représentant le nombre de nœuds explorés par alphabeta :

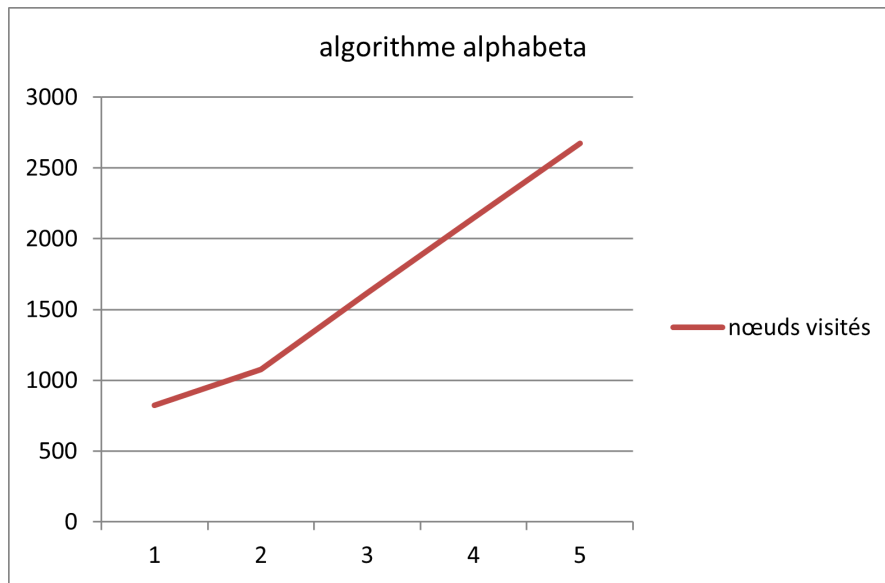


FIGURE 4 – Courbe de l'algorithme alphabeta

Voici le tableau qui résume les deux courbes :

Profondeur	minmax	alphabeta
1	822	822
2	14788	1078
3	455305	1615
4	530578	2147
5	374174846 (très grand nombre)	2675

## 0.4 Raisonnement de la profondeur

Afin d'identifier l'intérêt du raisonnement en profondeur et de mieux analyser les algorithmes minmax et alphabeta. On fixe la taille de la grille (7 x 7) et une profondeur de raisonnement pour les deux joueurs. Car on fixe la profondeur du premier joueur à 2 et on change pour le deuxième joueur, on a utilisé l'algorithme alphabeta qui car il est le plus simple. Les résultats sont :

Profondeur de Joueur 1 <sup>er</sup>	Profondeur de Joueur 2 <sup>ème</sup>	Gagnant
2	1	Joueur 1 <sup>er</sup>
2	2	Joueur 1 <sup>er</sup>
2	3	Joueur 1 <sup>er</sup>
2	4	Joueur 1 <sup>er</sup>
2	5	Joueur 1 <sup>er</sup>

On remarque que le premier joueur a gagné toutes les parties dans ce cas mais dans le cas où la profondeur du premier joueur change donc la gagnant change aussi.

## 0.5 Conclusion

En conclusion, ce jeu marchera d'après la profondeur donner et nous pouvons voir à travers des deux courbes que l'algorithme alphabeta est plus simple que minmax car il est optimal en calcul des nœuds visités.