université de Caen
Basse-Normandie

# Sécurité et Aide à la Décision
# Expérimentations du TP

Mori Baptiste 21602052
Leblond Valentin 21609038

L2-Info-groupe-4A

# Table des matières

# Introduction

L'objectif de ce TP est de mettre en place deux intelligence artificielles, la première fonctionnant avec l'algorithme Minmax et l'autre avec l'algorithme Alphabeta. Ces intelligences artificielles nous servirons à choisir le meilleurs coup à jouer sur une profondeur donnée.

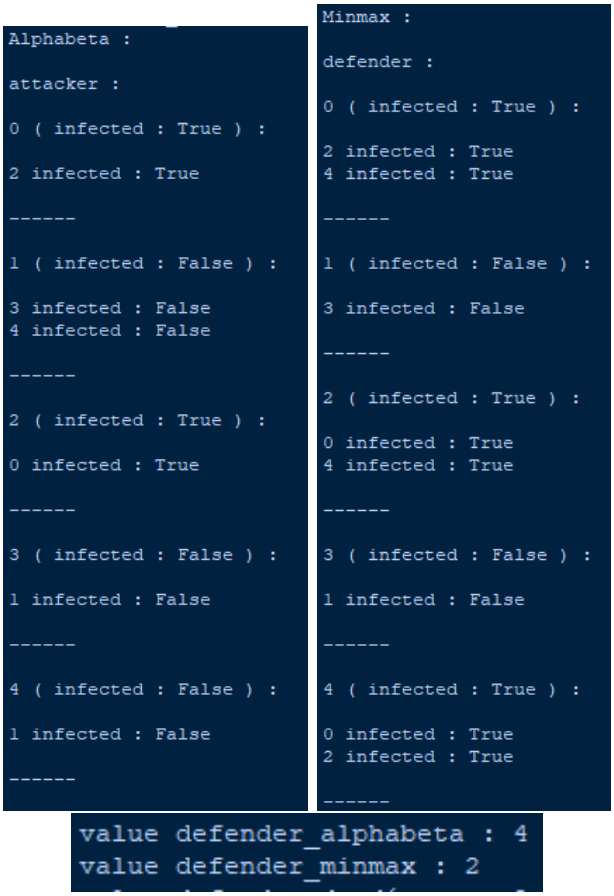
Nous allons procéder à des expérimentation sur le nombre de nœuds que parcours chaque algorithme afin de comparer Minmax et Alphabeta.

# 1 Désaccord sur les choix des deux algorithmes

Nous avons mis les deux algorithme en parallèle sur le même réseaux pour voir si ils avaient les mêmes décisions de coups. Nous avons remarqué que, pour le cas de l'attaque, il arrivait qu'ils soient en désaccord. Comme on peut le voir sur la capture ci-dessous dans l'état actuel Alphabeta choisi d'infecter l'ordinateur 2 alors que Minmax choisi l'ordinateur 3.

```
attacker :

0 ( infected : True ) :

2 infected : False
4 infected : False

------

1 ( infected : False ) :

3 infected : False
4 infected : False

------

2 ( infected : False ) :

0 infected : True
4 infected : False

------

3 ( infected : False ) :

1 infected : False

------

4 ( infected : False ) :

0 infected : True
1 infected : False
2 infected : False

------


choix attacker_alphabeta : 2
choix attacker_minmax : 4
n_ab : 2291
n_mm : 5615
...
```

Les deux algorithmes mènes à deux états finaux différents, pour Alphabeta on a le défenseur qui gagne et pour Minmax c'est l'attaquant qui gagne. Les défenseurs ont également des valeurs différentes, plus importante chez Alphabeta mais la valeur des défenseurs étant différente ne nous dis rien sur le faite que l'un est plus efficace que l'autre car on fait jouer un attaquant Alphabeta contre un défenseur Alphabeta de même pour Minmax.




Nous nous avons donc pensé que Alphabeta favorisé le défenseur et Minmax l'attaquant, nous avons donc tester de faire jouer un Alphabeta attaquant contre un Minmax défenseur et un Minmax attaquant contre un Alphabeta défenseur sur la même génération de graph (étant donnée qu'ils jouent quasiment la même chose on a lancé plusieurs fois le jeu afin de tomber sur un désaccord).

```
attacker :

0 ( infected : True ) :

2 infected : False
3 infected : False
4 infected : False

------

1 ( infected : False ) :

3 infected : False
4 infected : False

------

2 ( infected : False ) :

0 infected : True
4 infected : False

------

3 ( infected : False ) :

0 infected : True
1 infected : False

------

4 ( infected : False ) :

0 infected : True
1 infected : False
2 infected : False

------


choix attacker_alphabeta : 2
choix attacker_minmax : 3
```

Sur les résultats finaux des deux batailles, on remarque que quand Alphabeta attaque (capture de gauche), il gagne quand même mais il a infecté que 3 ordinateurs sur 5 et quand Minmax attaque (capture de droite) il en a infecté 4 sur 5.

```
Alphabeta :

attacker :

0 ( infected : True ) :

2 infected : True
4 infected : True

------

1 ( infected : False ) :

3 infected : False

------

2 ( infected : True ) :

0 infected : True
4 infected : True

------

3 ( infected : False ) :

1 infected : False

------

4 ( infected : True ) :

0 infected : True
2 infected : True

------
```

```
Minmax :

defender :

0 ( infected : True ) :

2 infected : True
3 infected : True
4 infected : True

------

1 ( infected : False ) :


------

2 ( infected : True ) :

0 infected : True
4 infected : True

------

3 ( infected : True ) :

0 infected : True

------

4 ( infected : True ) :

0 infected : True
2 infected : True

------
```

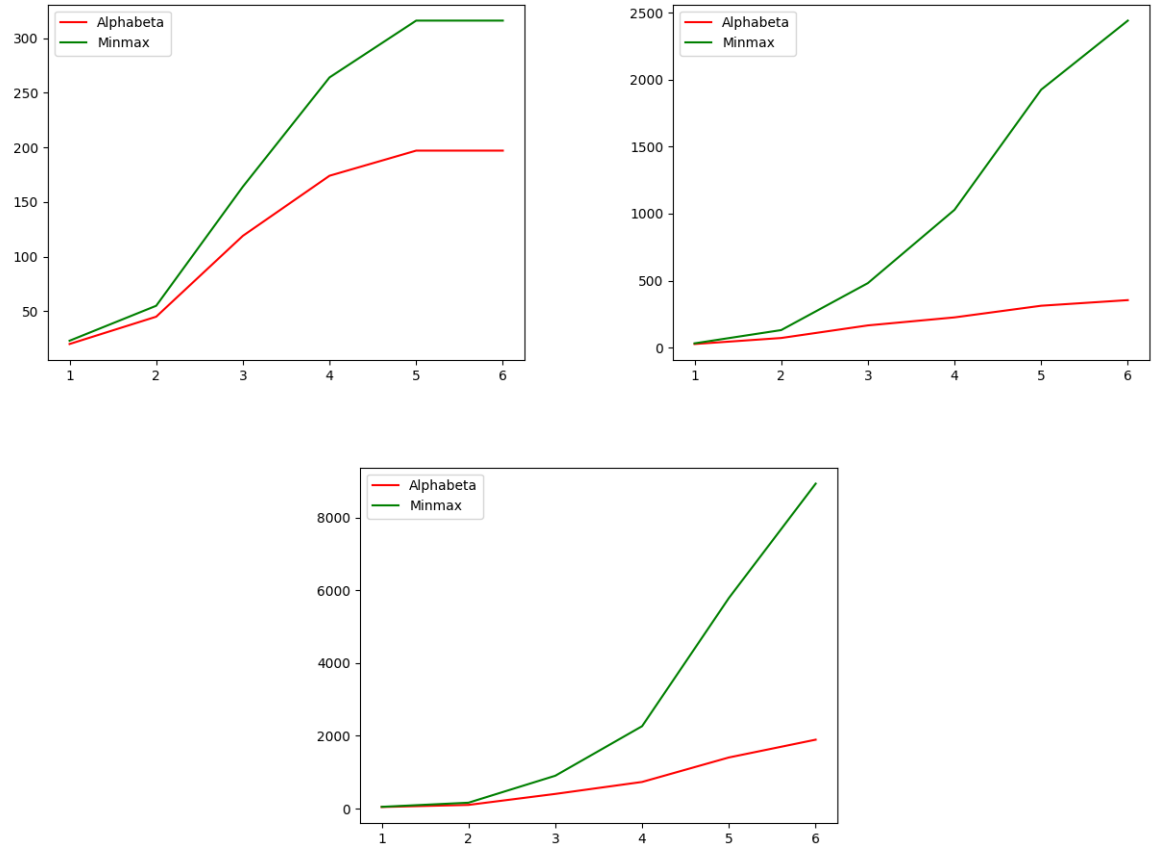
```
value defender_alphabeta : 0
value defender_minmax : 2
```

Malgré ces testes, nous ne pouvons rien dire sur l'efficacité des ses deux algorithme qui semblent presque équivalent en terme de résultat.

## 2 Tests sur le nombre de parcours des nœuds

Nous allons nous intéresser à comparer le nombre de nœuds que parcourt chaque algorithme. Pour cela, nous avons implémenté un compteur dans le constructeur de la classe de L'IA afin de compter chaque passage dans Minmax et dans Alphabeta.

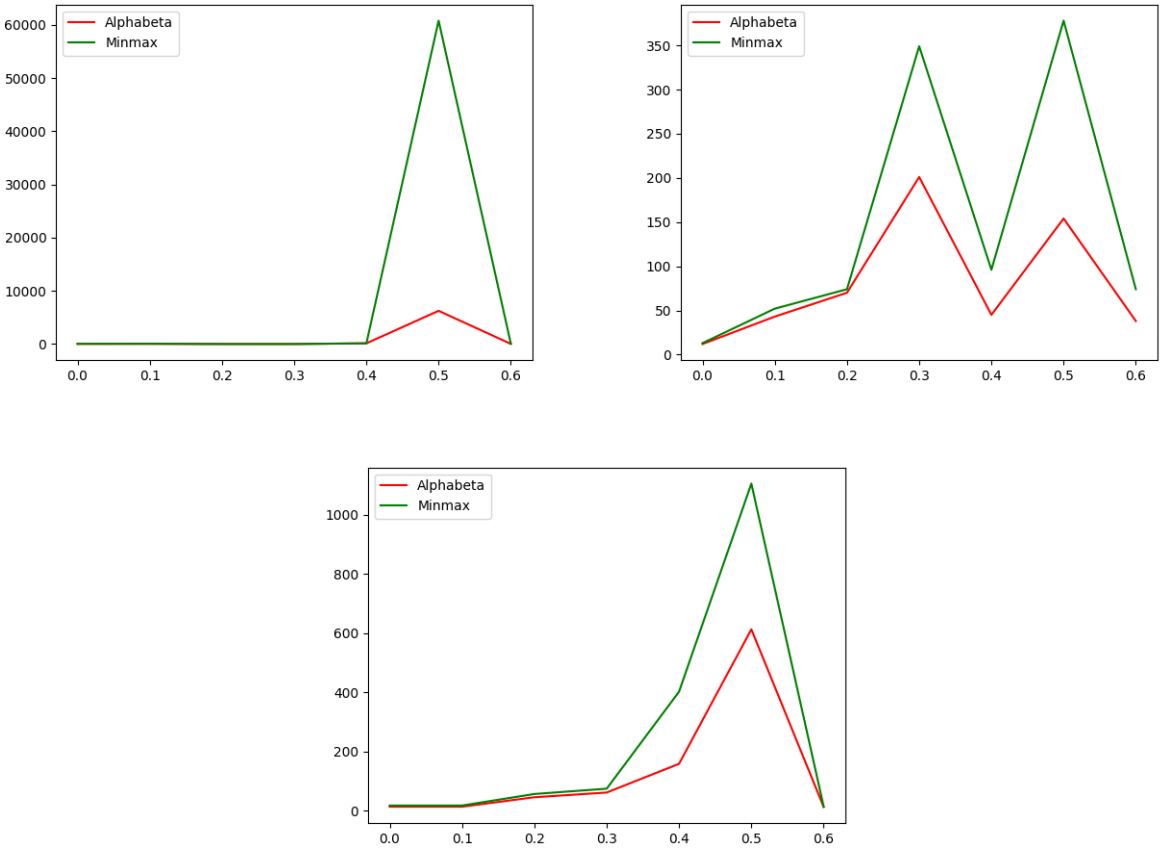
Dans un premier temps, nous allons faire varier le nombre de profondeur auquel les algorithme vont regarder tout en gardant le même graph de départ pour le test de chaque profondeur. Nous utiliseront Matplotlib et Numpy pour construire nos graphiques.




Nombre de nœuds parcourus en fonction de la profondeur

On remarque que pour chaque test, Alphabeta est toujours bien en dessous en terme de parcours que Minmax, même si pour le premier graphique le graph du jeu semble un peux particulier (le passage entre la profondeur 5 et 6 n'a pas changer énormément le nombre de parcours), Alphabeta semble parcourir le moins de nœuds quelque soit la profondeur qu'on lui donne.

Dans un second temps, nous allons faire varier la probabilité que deux ordinateurs soient connectés (cette fois nous recréons à chaque fois un nouveau graph d'où les courbes un peu chaotique).




Nombre de nœuds parcourus en fonction de la probabilité

On remarque de même que Alphabeta parcours moins de nœuds que Minmax et que plus Minmax parcours de nœuds, plus l'écart entre les deux est important, il y a donc une relation de puissance sur le nombre de nœuds parcourus par ces deux algorithmes.

# Conclusion

Alphabeta et Minmax ayant quasiment les mêmes décisions sur les coups à jouer, Alphabeta est celui qui parcours le moins de nœuds et donc qui est le plus rapide dans sa prise de décision.