Intelligence Artificielle

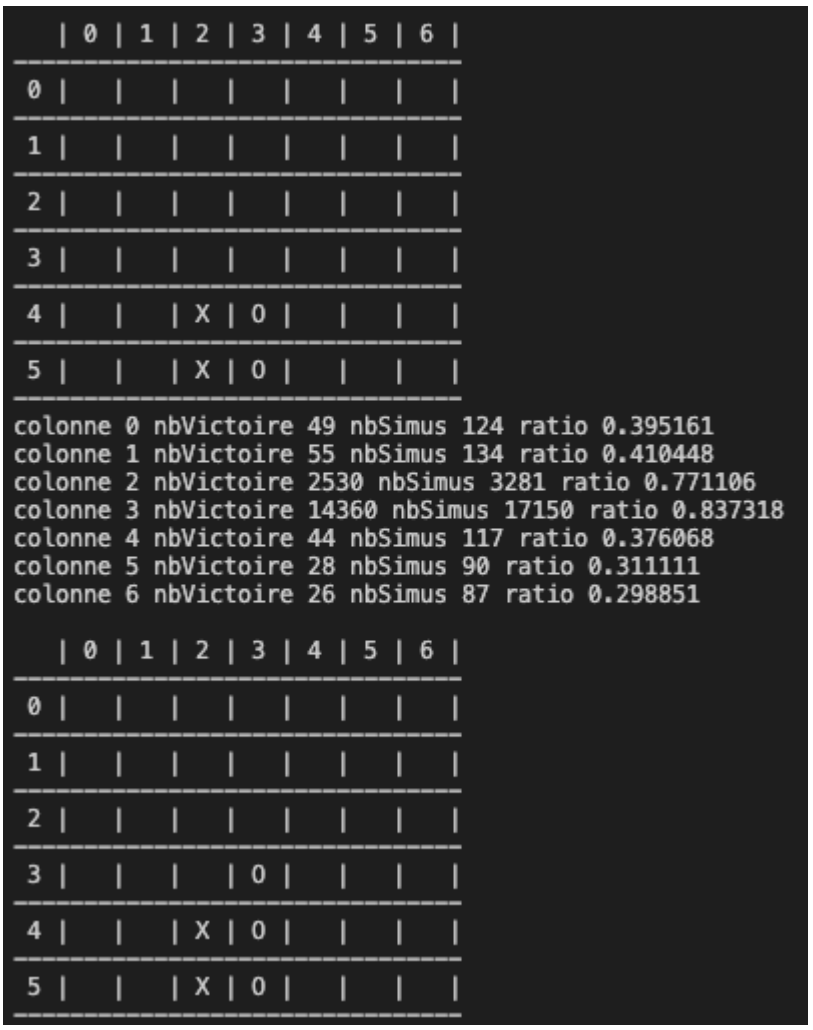
Rendu de Projet

BOUCHER Yohan
Jacquier Timéo

Master 1 - Informatique

Université de Lorraine

Question 1:



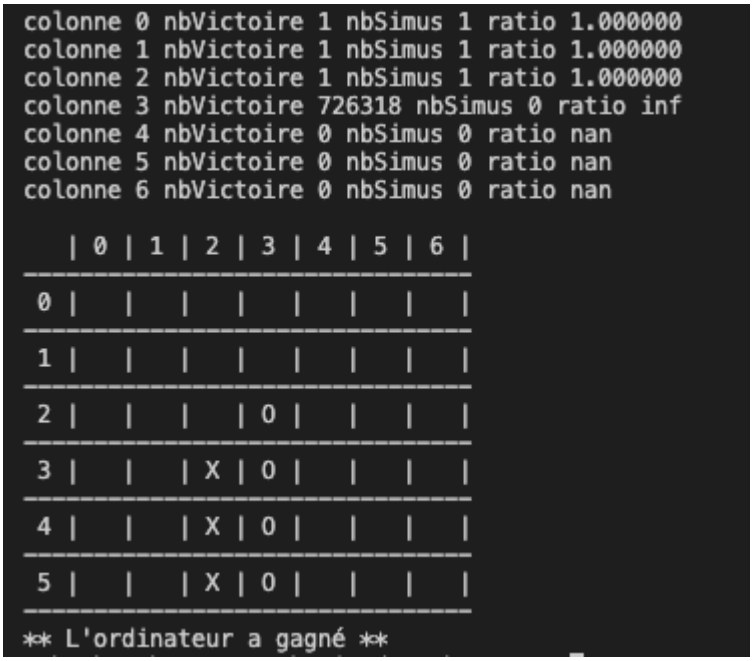
Affichage des données sur la console avec les numéros de colonnes ainsi que le nombre de simulations effectuées sur chaque colonnes, le nombre de victoires et le ratio entre le nombre de victoires et le nombre de simulation.

Question 2:

Le minimum de temps qu'on peut mettre est de 1 seconde, en dessous, l'algorithme n'est pas fonctionnel. À 1 seconde, l'algorithme nous bat à tous les coups

Question 3:

À chaque coup, l'ordinateur teste chaque possibilité, si cette dernière le fait gagner, alors l'IA choisit directement de jouer ce coup et ne testera pas les possibilités suivantes



Ici, l'ordinateur à directement joué sur la colonne 3 (menant à la victoire), les colonnes 4, 5 et 6 n'ont pas été testées.

Question 4:

Le flag "-O3" est un flag d'optimisation proposé par GCC. Avec ce flag, GCC optimise le code machine produit après compilation, ce qui permet une compilation plus rapide, et une optimisation du temps d'exécution, mais aussi de l'utilisation mémoire et processeur. Ce flag en particulier permet de nombreuses optimisations sur les boucles, ce qui est important au vu de la quantité de boucle et de leurs nombres d'instructions dans le programme. Cependant, cela ne semble toujours pas marcher. En effet, lors de l'exécution de la version optimisé et de la version non optimisé, sur un même déroulé de partie, on constate une utilisation mémoire de 2Go pour la version optimisé, et de moins d'1Go pour la version compilée par défaut, avec une utilisation processeur à peu près équivalente.
Les mesures ont été réalisées via le système Ubuntu.

Question 5:

Le critère "max" à l'air de donner de meilleure performance, car ce dernier privilégie les chemins qui donnent le plus de victoires.

Question 6:

Temps de calcul pour jouer le premier coup :

Le calcul dans le pire des cas est de remplir la dernière au dernier moment (lorsque toutes les lignes exceptées la dernière sont remplies). Cela veut dire qu'on a 7 possibilités dans

chaque cases. Ensuite, dans la dernière ligne, le nombre de possibilités décroît au fur et à mesure où la ligne se remplit.

7	6	5	4	3	2	1
7	7	7	7	7	7	7
7	7	7	7	7	7	7
7	7	7	7	7	7	7
7	7	7	7	7	7	7
7	7	7	7	7	7	7

Nombre de possibilités pour chacunes des cases (pire cas)

Le nombre d'itération est donc de 7^36 * 6! = 1,9.10^33