Semestre Septembre 2021-2022



**RAPPORT DE PROJET--LU3IN005**

*Exploration/Exploitation et Puissance 4*

**Groupe 1 Binôme 6**

Guo Youheng 28711440

Yang Kai 21107392

**Présentation du projet**

Dans ce projet, notre objectif principal est d’implémenter un jeu (puissance 4) pour qu’on puisse le jouer entre deux joueurs et d’optimiser les algorithmes pour des joueurs (IAs) de jeu.

Ce projet consiste à faire réaliser des stratégies de joueur en utilisant des différents algorithmes. Des algorithmes principales étaient à implémenter : algorithme de Monte-Carlo, l’algorithme aléatoire, l’algorithme greedy, l’algorithme e-greedy, l’algorithme UCB et l’algorithme UCB adapté aux arbres de jeu.

**Description de l’architecture du code**

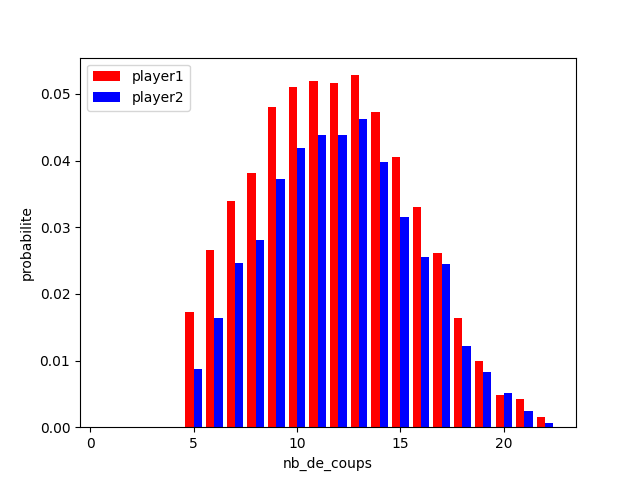
**Explication des statégies et analyse des résultat**

Partie 1 : Combinatoire du puissance 4

Tout d’abord, on a obtenu une liste de 69 quadruplets de cases (pour le plateau 6 \* 7) après avoir implémenté la fonction qui calcule des cas gagants.

Ensuite, nous avons implémenté le moteur de jeu dans la classe « Game » et un joueur dans la classe « Player » pour qu’un joueur puisse jouer une partie classique avec un autre joueur (les joueur jouent aléatoirement sans statégie). On peut bien obtenir un résultat de la partie, ça nous permettra d’analyser le nombre de coups avant une victoire lorsque les deux joueurs

jouent aléatoirement. Après 10000 parties, on a un graphique qui décrit la distribution de nombre de coups.

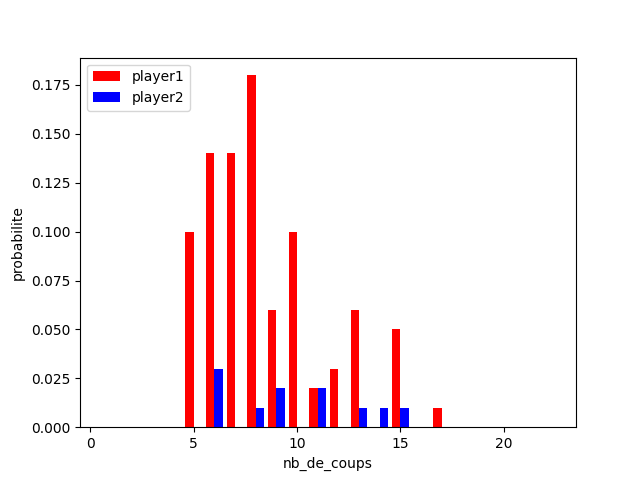


*La distribution du nombre de coups*

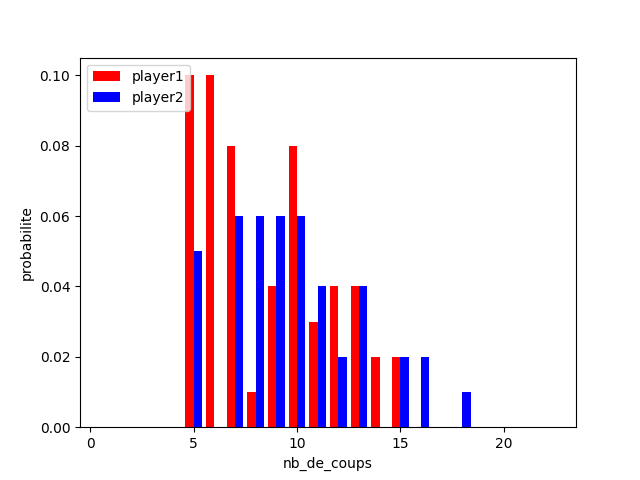
玩家1赢的概率更大，随着步数的增多这种差距逐渐减小，这种情况正常，因为玩家1先手。图像符合泊松分布，因为 。。。

我们可以让两个玩家随机对战无穷多次，得到平局的概率

Partie 2 : Algorithme de Monte-Carlo



MonteCarloPlayer VS RandomPlayer



MonteCarloPlayer VS MonteCarloPlayer

Partie 3 : Bandits-manchots