Validation Prolog 6/11/19 à 12h

Jeu général :

* Coups possibles : tableau/ Liste des coups possibles en coordonnée, mise à jour à chaque coup ex ((0,1),(0,2),(0,3), …. Au début. On ajoute +1 à la ligne quand on joue sur la colonne x quand c’est égal à 7 on l’enlève du tableau.

Tableau de jeu : Liste liste ou matrice avec 0 si rien, 1 ou 2 si un joueur

Implémenter l’aléatoire choisi un coup aléatoire dans la liste des coups possibles

Test fin du jeu : (liste des coups possibles est vide, victoire : test avec le dernier coup joué (4 alignés))

Idée d’heuristique :

* Aléatoire

De base :

* 1 coup d’avance : à chaque coup possible test s’il y a un pion du joueur à coté

|\_> améliorations prendre le coup qui permet l’alignement le plus long

* 2 coup d’avance : idem plus haut mais on regarde si n pourra mettre un autre pion après pour continuer la même ligne

<https://www.christian-schmidt.fr/puissance4>

Principe du jeu a implémenter :

(Ensuite on mettra en paramètre l’heuristique et le nombre de coups d’avance que l’on souhaite)

PDF de ce matin : <https://www.irif.fr/~boutier/ens/IP1/Projet-Puissance4.pdf>

Page LIRIS sur PROLOG et les ALGOS de JEU : <https://perso.liris.cnrs.fr/christine.solnon/Site-PPC/session4/e-miage-ppc-sess4.htm>

#### Min-Max

Principe :Soit Max l'agent que l'on cherche à faire gagner et Min son adversaire. L'algorithme Min-Max à comme but l'élaboration d'une stratégie optimale pour le joueur Max. **À chaque tour, le joueur Max va choisir le coup qui va maximiser son score, tout en minimisant les bénéfices de l'adversaire.** Ces bénéfices sont calculés par une méthode d'évaluation qui, en fonction de l'heuristique choisie, permet de calculer le poids d'une feuille de l'arbre.

Pour des raisons de temps de calcul et de mémoire, nous limitons le nombre de coups en avant par une profondeur de l'arbre.

#### Alpha-Beta

L'algorithme a-b est une optimisation du MiniMax. L'algorithme Min-Max peut être coûteux en terme de complexité et donc en terme de temps d'exécution. La solution proposée par l'algorithme Alpha-Beta consiste donc à ne **parcourir que les noeuds pertinents,** et à réduire la profondeur de jeu. On parle d'élagage de l'arbre de jeu. Plus la profondeur de l'arbre est importante, plus l'intelligence artificielle sera difficile à battre. ( cf AAIA - cf schéma de IDA \* on ne continue à parcourir la branche qui maximise/minimise l’heuristique)

### 

### Système de points pour évaluer les coûts d’après et faire un min-max:

Source : <http://fabien.noirot.free.fr/puissance4.html>

Résumé:

* Loi de gravité comme règle du jeu pour éviter des parcours de cases trop en haut
* Evaluation avec min-max et une fonction des points->Attribution des points à un coup futur en fonction de critères:
* **Création de la fonction d’évaluation :**

- points de position :

Il s’agit de favoriser la création d’alignements.On regarde dans chacune des 4 directions (/, \, |, \_) qui passent par le coup joué, dans le but de trouver d’autres cases de même couleur déjà jouées. La valuation se fait alors en fonction de la distance du coup joué vers la plus proche case voisine, ainsi que du nombre des cases déjà placées par le joueur dans les 4 directions.

Ex : 2e case : +10 distance à la plus proche : 1 case : +20

3e case : +100 2 cases: +10

4e case : +1000 (victoire) 3 cases : +5

- points de blocage :

Il s’agit là de gêner l’adversaire dans sa tentative de création d’alignements. Il faut repérer des alignements consécutifs de cases adverses (2 ou 3 cases, car 4 cases impossible : défaite) et jouer à coté. Il s’agit avant tout d’éviter les cas de victoire de l’adversaire. ( 3 cases alignées)

Ex : 3 cases : +500

2 cases : +50

1 case : +5

Petit point utile de stratégie : <https://www.ilemaths.net/sujet-strategie-optimale-puissance-4-379114.html>

## Technique de Monte Carlo :

On joue beaucoup de parties aléatoires et on attribue à chaque coup des points en fonction du gain moyen à partir de ce coup.

Expliqué ici à page 22 : <http://www.alinehuf.fr/master1-info/heuristiques_GGP_alinehuf.pdf>

## Tableau des alignements possibles

Source :<https://www.christian-schmidt.fr/puissance4>

Résumé : on crée un tableau de la grille de jeu où chaque case(x,y) contient le nombre d’alignement possibles en (x,y) . On ajoute pour le pion max et on soustrain pour min.

## 

## Tuto solving puissance 4 explication strategie (long mais complet) :

<http://blog.gamesolver.org/>

## Stratégie NegaMax :

implémentation simplifié de min-max en version récursive, on considère qie le score de l’adversaire est l’opposé du sien.

La variante Negamax de MinMax est une simplification de la mise en œuvre, tirant parti du fait que le score d’une position du point de vue de votre adversaire est le contraire de celui de la même position de votre point de vue. (c’est à dire que **max(a, b)=-min(-a, -b)** )

Ainsi, vous pouvez implémenter une version unique de la fonction récursive pour calculer le score d'une position et ne plus avoir à faire la différence entre vous et votre adversaire.

## Autre source :

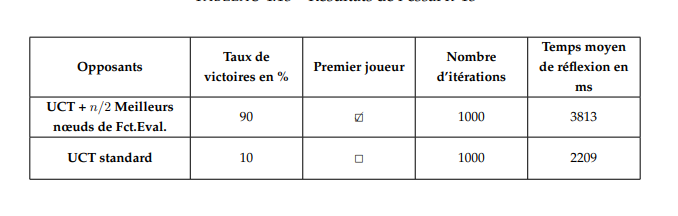
<https://en.wikipedia.org/wiki/Minimax> : regarder les schémas pour comprendre les min max

<https://towardsdatascience.com/creating-the-perfect-connect-four-ai-bot-c165115557b0>

Diapo sur prolog : <http://web4.ensiie.fr/~gacogne/pro3-4.pdf>

Explication min-max diapo 45 : <http://www-lisic.univ-littoral.fr/~guibadj/IA1/cours4.pdf>

<http://ratheil.info/myPublicThesis/masters/2016_GaelAglin--Houndji.pdf>



Rq : Le seul premier coup gagnant est celui de la colonne centrale. Un premier coup dans les colonnes adjacentes permet au second joueur d’obtenir une partie nulle, et un premier coup dans l’une des quatres autres colonnes extérieures permet même au second joueur de remporter la victoire s’il joue de façon optimale