**Algo greedy :**

1. Pivot carte éjectée

Si random : random.choice

Sinon : opérations suivantes pour chaque pivot

1. Liste de toutes les flèches d’insertion

Si random : random.choice

Contrainte : obligé de modifier le plateau pour calculer tous les chemins accessibles quand on insère telle carte à telle endroit => création d’une copie virtuelle du plateau

1. Génère la liste des chemins possibles étant donné l’insertion de telle carte à telle endroit
2. Calcul des points associé à chaque chemin
3. Choix du maximum.

Renvoie : coup :[ nb rotations, nom flèche utilisée, [chemin]]

Inconvénient : si on ne récupère pas de point, on ne se préoccupe pas de minimiser les gains de l’adversaire

**Algo Min\_Max :**

Basé sur la récursivité.

Entrées : plateau, joueur, joueur copie, alpha, beta, maximizePlayer (le joueur qui appelle), depth (nombre de coups à l’avance)

Retour : flèche, orientation, chemin, gain

1. 1er coup : greedy
2. Si Depth=0 (que le jeu est fini où qu’on ne regarde que jusqu’au coup n°x) :

Return (flèche\*,orientation\*,chemin\*,gain)

Si check\_gagnant :

Return (None,None,None,0) #ca sert à rien de réfléchir

Si Maxplayer False  : #adversaire

Simulation tous pivots toutes flèches

On minimise le gain en considérant que pour chaque coup l’adv peut faire un gain maximal

Si Maxplayer = True : #joueur

Simulation

On maximise le gain

1. Alpha, Beta pour l’élaguage (init à -inf et +inf)

Choix stratég et inconvénient : 1 seul coup à l’avance pour des raisons de temps de calcul

**UCT**

Entrée : Plateau

1. phases : sélection, extension, rollout, retropropagation
2. La situation initiale c’est le coup du joueur N-1

Extension : On génère tous les coups possibles (simul pivot carte et insertion)

Tous initialisés à +inf car pas visités, pas moyen de prendre le maximum. On prend le premier (on pourrait en prendre un random)

1. Rollout :
   1. à partir de la situation initiale : que des coups aléatoires
   2. en epsilon-greedy : un greedy de temps en temps

Renvoie le nb de victoires pour chacun des joueurs pour N parties simulées

1. Rétropropagation : on fait remonter le nombre de victoires.
2. A chaque fois qu’on passe sur un nœud on augmente son nombre de visites

Limites et améliorations :

* Rollout en random
* On initialise tout à +inf donc on privilégie un parcours en largeur plutôt qu’en profondeur. Idée : initialiser à nombre moyen de parties