**Algorithme utilisés**

* 1. Arbre de Monte Carlo / algorithme min-max / Algorithme
  2. Arbre de Monte Carlo / UCTS / RAVE / AMAF

**Idées pour l'IA**

Position qui assure un lien (bloquer automatiquement lorsque l'utilisateur essaye de casser ce lien ?) :

* 1. Losange (entre deux pions)
  2. Ziggurate (avec le bord)

Position ou un seul coup est possible :

* 1. En défense (si il y a défaite assuré si un coup n'est pas joué)

Jouer automatiquement si victoire à 100% ou s'il y a un seul coup possible pour éviter la défaite

Highlight Certaines connaissance peuvent ne pas être bénéfique pour l'IA (éviter d'implémenter des stratégies manuellement) car elles ne prennent pas en compte tous les pièges qui peuvent être posés

**Gestion des données**

**Cases mortes**

Existence des cases mortes (pas besoin d'être évalué) => tableau de booléen

**Pions**

|  |  |  |  |
| --- | --- | --- | --- |
| Idées | 1 seule tableau pour tous les pions avec 3 valeurs possible par cases :   * 1. Vide = 0   2. Blanc = 1   3. Noir = 2 | 2 tableaux (1 par couleur) :   * 1. On rentre que les pions qui ont déjà été placé | 2 Listes / Piles (1 par couleur) |
| Avantages | Plus adapté à la méthode du circuit électrique ? | Adapté à la méthode du circuit électrique | Permet de tester récursivement si un joueur gagne |
| Inconvénients | Plus long pour différencier les pions blancs/noirs |  | Stack Overflow possible |

**Connexions**

Enregistrer les connexions virtuelles / semi-virtuelles / réelles dans 3 tableau => évite de les recalculer à chaque fois

Highlight Ne pas oublier de les supprimer si elles sont détruites

**Fonction d'évaluation**

**Méthode du circuit électrique :**

Contact Semi-connexions / connexions (voir PDF fonction d'évaluation)

Definition Connexion virtuelle : noir peut relier ses deux pièces même si blanc commence

Definition Semi-connexion virtuelle : noir peut relier ses deux pièce s'il commence

Definition La profondeur d'une connexion est la profondeur de l'arbre requis pour établir la connexion

Position initiale = Semi connexion (car stratégie gagnante)

**H-search pour trouver les connections virtuelles/semi-virtuelles**

On considère et deux connexions virtuelles avec et deux cases noirs, et deux ensemble de cases vide qui relie deux cases ( et pour et et pour ) On prend et et

Definition AND :

* 1. Definition Si est noir, alors et sont reliées par une connections virtuelle
  2. Definition Si est vide, alors et sont reliées par une connexion semi-virtuelle

Definition OR : Si et sont reliés par plusieurs connexions semi-virtuelles qui ne se chevauchent pas, alors et forment une connexion virtuelle

On utilise les propriétés AND et OR pour trouver une première générations de liens, on fait ça récursivement jusqu’à ce qu'il n'y ai plus de liens ou un lien gagnant

**Fonction d'évaluation**

2 circuits électrique dans le jeu (un pour les noirs et un pour les blancs)

Chaque case est associé à une résistance (exemple pour les noirs) :

* 1. 1 si la case est vide
  2. 0 si la case contient un pion noir
  3. si elle contient un pion blanc

Connexion virtuel traité comme cases voisines

Connexion entre 2 vrai voisin = plus fort que des connexions virtuelles => résistance moins forte

Plus la profondeur de la connexion est grande, plus la résistance est élevée

Formule Fonction : avec la résistance des noirs et la résistance des blancs