Projet Résolution de Problèmes

1GI

Jeu de Moulin

Réalisé par :

- AMGHAR Souhail

Encadrant :

- Pr. Malika ADDOU

Plan :

1. Introduction :

1.1 Contexte:

Ce projet s'inscrit dans le cadre du module « résolution de problèmes ». L'objectif est de programmer le jeu de moulin en langage C/C++, et d'implémenter les stratégies de traitement de l'intelligence artificielle IA étudier pendant ce module.

1.2 Présentation du jeu:

Le jeu affronte à deux personnes sur un tableau formé par quatre carrés concentriques reliés au centre de leurs quatre côtés par des lignes perpendiculaires. Le jeu se déroule sur les 24 points du tableau (les 12 coins des carrés et les 12 intersections qu´ils forment avec les lignes perpendiculaires). Chaque joueur a 9 pions (9 hommes de morris), de couleur différente pour chacun d´eux.

1.3 Travail réalisé:

Tout au long de mon travail sur le problème de « jeu de moulin », j'ai pu réaliser un logiciel version console qui utilise différents algorithmes, structures et fonctions pour répondre au travail demandé et je tacherai à décrire leurs rôles et leurs fonctionnement globales.

2. Modélisation :

2.1 Espace d'états :

J'ai représenté les paramètres du problème de moulin dans un nœud état en exploitant les notions de l'orienté objet fourni par le langage C++:

```
class noeud
{
   public:
      int E[7][7];
      noeud();
      noeud(int [7][7]);
      float Heuristique_Attaque(int);
      float Heuristique_Defense(int);
      vector<noeud> noeud_suivant(int);
      vector<int> Remove(noeud);
      bool Etat_Final();
};
```

- **E[7][7]** : Matrice de table.
- **Nœud() et nœud(int [7][7])** : deux constructeurs des nœuds .
- **float Heuristique_Attaque (int) et float Heuristique_Defense (int)**: deux méthodes de calcul de l'Heuristique d'un nœud .
- **vector<noeud> noeud_suivant(int) et vector<int> Remove(noeud)**: deux méthodes de génération des nœuds suivants .
- **bool Etat_Final()**: méthode pour tester si le nœud est final.

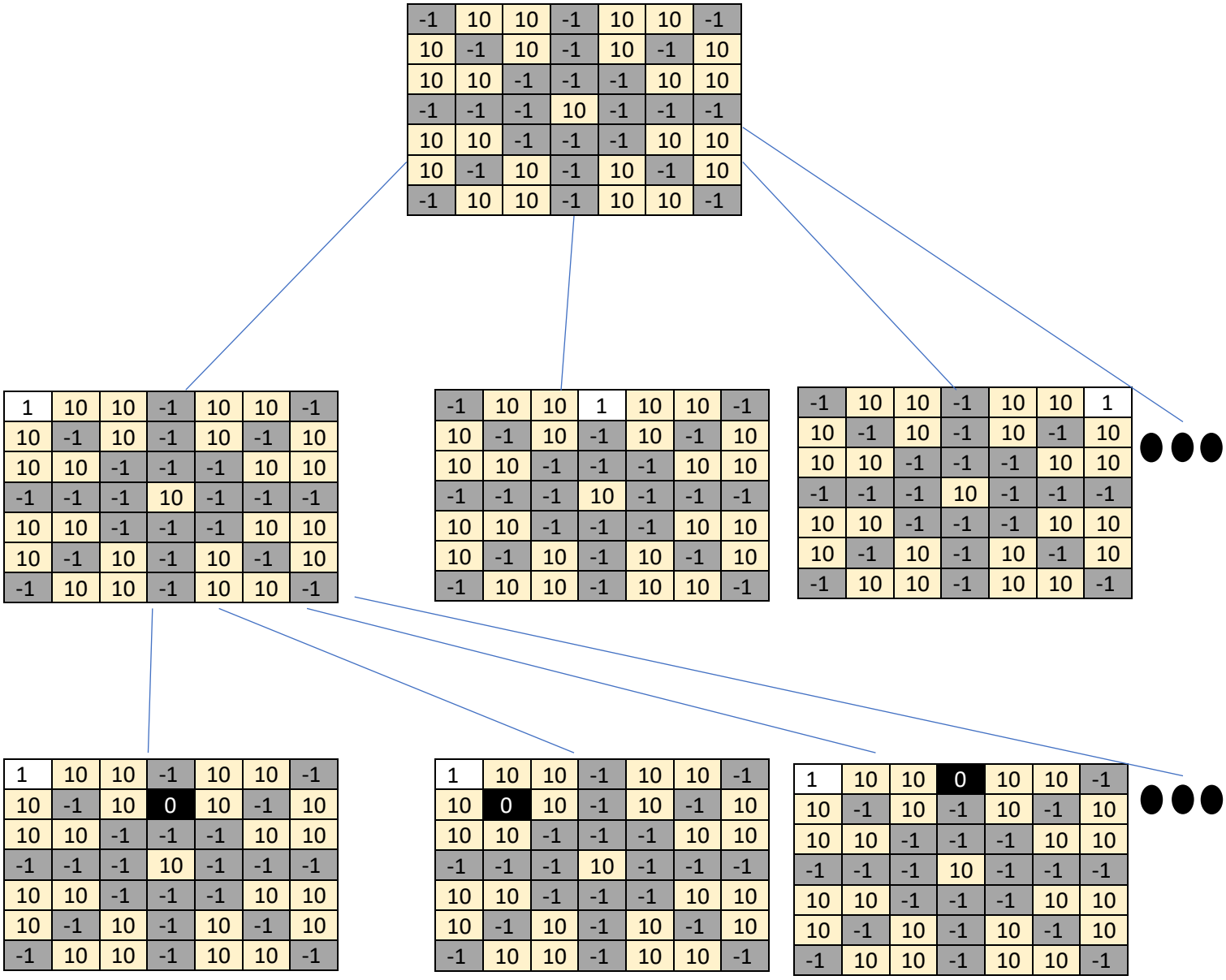
Etat initial :

La matrice E[7][7]= {{ -1, 10, 10, -1, 10, 10, -1},
{ 10, -1, 10, -1, 10, -1, 10},
{ 10, 10, -1, -1, -1, 10, 10},
{ -1, -1, -1, 10, -1, -1, -1},
{ 10, 10, -1, -1, -1, 10, 10},
{ 10, -1, 10, -1, 10, -1, 10},
{ -1, 10, 10, -1, 10, 10, -1}};

10 pour les cases inutilisables
-1 pour les cases vides

Etat final : la matrice E[7][7] ne contient que 2 pions de MAX ou de MIN et le nombre de coup joué est supérieur à 18.

2.2 Arbre de Recherche :



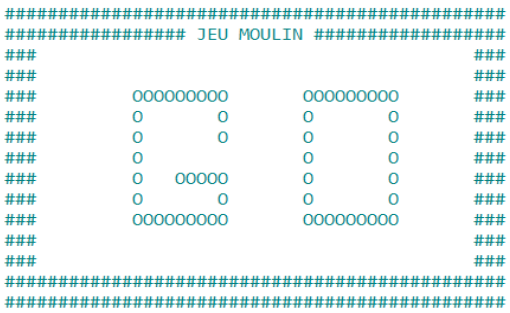
2.3 Les règles :

- Le jeu commence avec un plateau vide entre deux joueurs Max et Min.
- Les joueurs Max et Min doivent placer à tour de rôle leurs pions (neuf pions au maximum) sur un point d'intersection libre du plateau.
- Le but est de faire un « moulin : trois pions alignés ». Si un joueur forme un moulin, il retire un pion à son adversaire (en dehors d'un moulin éventuel).
- Quand les pions sont tous posés, les joueurs peuvent les glisser d'un point d'intersection à un autre point de la ligne (un mouvement à la fois).
- Quand un joueur n'a que trois pions, il peut sauter d'un point à un autre.
- Les pions formant un moulin sont intouchables par l'adversaire. Mais s'ils sont déplacés par le propriétaire, ils deviennent vulnérables.
- Le jeu continue ainsi jusqu'à ce qu'un joueur n'a que deux pions en sa possession. Dans ce cas, son adversaire est déclaré gagnant.

Stratégie suivie :

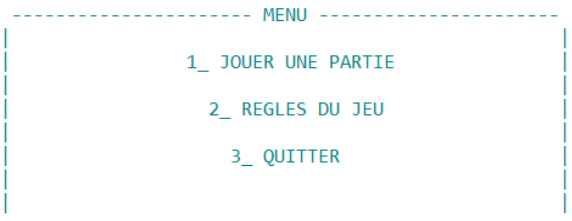
- Stratégie AO « min&max »
  ➔ Heuristique « H » donne un cout de valeur {1/2 ;1 ;2 ;3 ;4} selon le nœud et la probabilité de vaincre sur ce chemin .
  ➔ Heuristique « H » est décomposée en deux parties :
  -Heuri_Def : représente la probabilité de se défendre contre Max si ce nœud est joué.
  -Heuri_Att : représente la probabilité de vaincre Max si ce nœud est joué

3-Implémentation du logiciel



||||| Souhail |||||

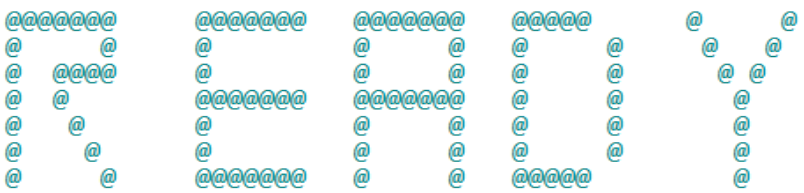
Press any key to continue . . .

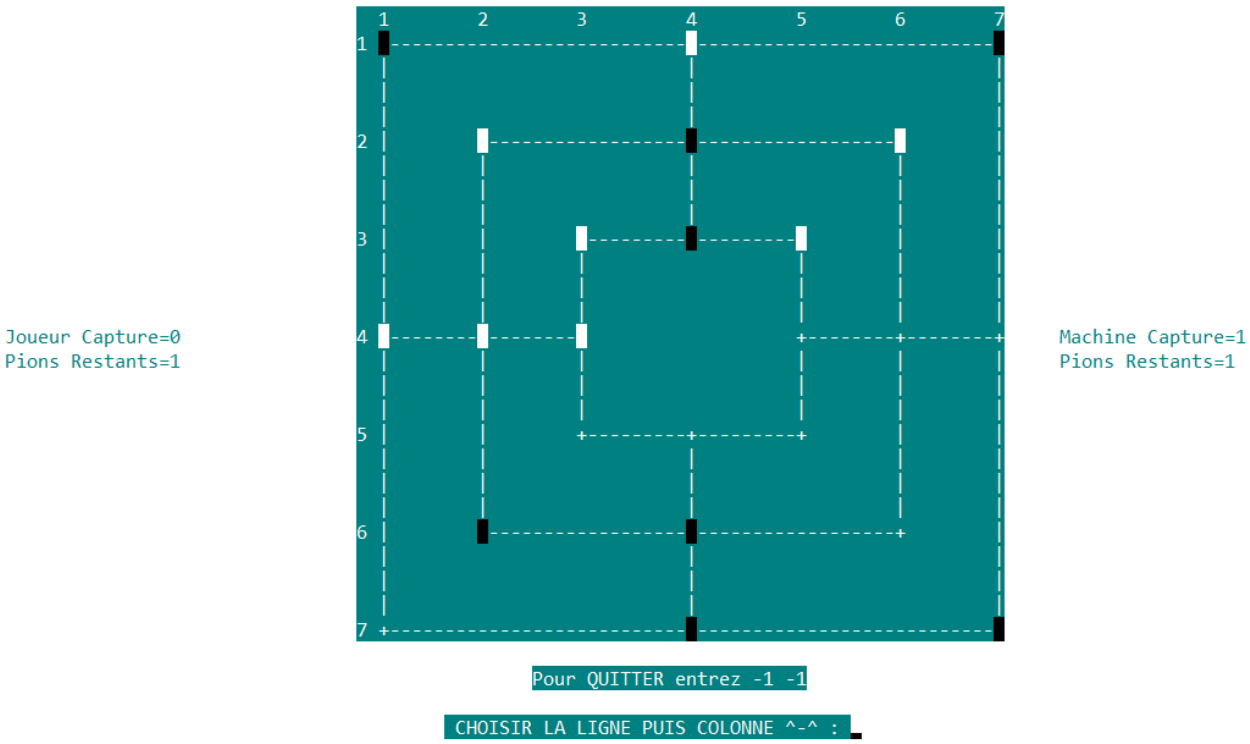


Votre reponse : 1

```
-------------------- MODES DU JEU --------------------
|                                                     |
|                  1_ JOUEUR VS MIN&MAX               |
|                                                     |
|                   2_ JOUEUR VS ALPHA&BETA           |
|                                                     |
|                     3_    RETOUR                    |
|                                                     |
-------------------------------------------------------

                    Votre reponse :
```





4-Conclusion :

Ce projet fut une très bonne expérience, vu qu'il a testé nos connaissances en résolution des problèmes acquises dans le 2er semestre. On a également pu améliorer nos compétences en modélisation et développement des différents algorithmes de recherche. Cela nous a permet de construire un AI que nous même ne pouvons pas vaincre.