جـامـعـة ابـن طـفـيـل
ⵜⴰⵙⴷⴰⵡⵉⵜ ⵙⴰⵏ ⴻⵖⵔⴰⵏ
Ibn Tofaïl University

Rapport Mini Project

Sujet : Jeu de Tic-Tac-Toe

Filière : Master Ingénierie des systèmes intelligents (ISI)

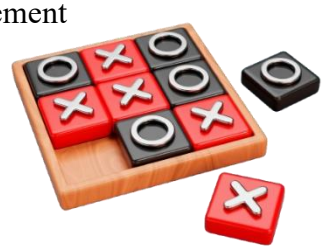
Réaliser Par :

- Tarhri Youssef
- Bouhaddou Mohammed

2024 - 2025

I. Introduction

Le morpion est un jeu stratégique à deux participants se disputant une grille de 3 cases sur 3. L'objectif est de placer trois marques identiques horizontalement, verticalement ou en diagonale. Ce projet propose de concevoir une version numérique avec Prolog pour la gestion des règles du jeu. Un algorithme MiniMax y est intégré afin d'optimiser les décisions de l'adversaire virtuel.



« Figure 1 : Plateau du jeu »

II. Objectifs du projet

- Développer un moteur logique chargé de gérer les règles du jeu, valider les coups joués et identifier les situations de victoire ou d'égalité.
- Intégrer l'algorithme MiniMax afin que l'ordinateur sélectionne la meilleure action possible à chaque étape.
- Mettre en place une interface simplifiée pour visualiser le plateau et permettre au joueur humain d'effectuer ses mouvements.

« Figure 2 : flux de données et interactions »

III. Moteur logique et règles du jeu

Le moteur logique s'appuie sur une structure de données composée de neuf éléments, représentant les cases du plateau, qui peuvent être inoccupées ou marquées par l'un des deux joueurs à l'aide des symboles 'x' et 'o'

3.1. La base de connaissances

Dans ce travail, la base de connaissances Prolog assure le rôle de moteur logique. Elle regroupe l'ensemble des règles qui régissent le déroulement du jeu Tic-Tac-Toe : la structure du plateau, les différentes configurations menant à la victoire, la détection des égalités et la validation des coups joués par les participants.

3.2. Représentation du plateau de jeu

Le plateau de jeu est modélisé sous forme d'une **liste contenant neuf éléments**. Chaque élément représente une case du plateau et peut être vide ou occupé par un symbole ('x' ou 'o'). L'ordre des éléments dans la liste correspond à l'ordre des cases dans la grille du morpion.

Par exemple :

- La première case est à l'indice 0.
- La dernière case est à l'indice 8.

3.3. Définition des conditions de victoire

Huit situations de victoire sont possibles dans le jeu :

- **Trois symboles alignés horizontalement** sur une des trois lignes.
- **Trois symboles alignés verticalement** sur une des trois colonnes.
- **Trois symboles alignés en diagonale**.

Dans la base de connaissances, ces différentes configurations sont définies sous forme de règles logiques. Chaque règle vérifie si trois positions spécifiques du plateau contiennent le même symbole, ce qui indique qu'un joueur a gagné.

3.4. Vérification de la victoire

Pour déterminer si un joueur a gagné, un prédicat spécifique s'appuie sur les règles précédentes. Il vérifie si l'une des huit combinaisons gagnantes est réalisée avec le symbole du joueur en question. Si c'est le cas, la victoire est déclarée.

Cette approche permet d'isoler la logique de détection de victoire et de la rendre indépendante du reste du programme.

3.5. Détection du match nul

Un autre prédicat permet de détecter les situations de match nul. Un match nul se produit lorsque :

- Toutes les cases du plateau sont occupées.
- Aucun des deux joueurs n'a réussi à aligner trois de ses symboles.

La base de connaissances vérifie simultanément ces deux conditions pour conclure à un match nul.

3.6. Validation des mouvements

Avant qu'un joueur puisse effectuer un mouvement, il est nécessaire de s'assurer que la case choisie est bien vide. La base de connaissances contient un prédicat qui vérifie qu'une position donnée est disponible pour un nouveau coup.

Ce contrôle garantit que les joueurs ne peuvent pas jouer sur une case déjà occupée, assurant ainsi le bon déroulement du jeu.

3.7. Réalisation d'un mouvement

Lorsqu'un joueur effectue un mouvement valide, un prédicat construit une nouvelle version du plateau où la case choisie est remplacée par le symbole du joueur. Cette modification se fait sans altérer les autres cases, respectant ainsi l'état du jeu à chaque étape.

3.8. Évaluation de la position

Pour les besoins de l'algorithme MiniMax, il est nécessaire d'attribuer une valeur numérique à chaque situation de jeu :

- Une victoire de l'ordinateur est évaluée positivement.
- Une victoire du joueur humain est évaluée négativement.
- Un match nul est neutre.

La base de connaissances fournit un prédicat qui associe une note à l'état du plateau, facilitant ainsi la prise de décision stratégique de l'algorithme.

IV. Algorithme MiniMax

L'algorithme MiniMax est un algorithme classique d'intelligence artificielle utilisé pour les jeux à deux joueurs avec information parfaite.

Principe :

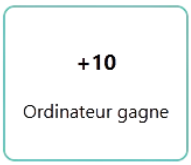
- L'ordinateur cherche à maximiser son score (représentant une position favorable).
- L'adversaire (le joueur humain) cherche à minimiser ce score.

- L’algorithme explore récursivement tous les coups possibles et évalue la qualité de chaque position selon les règles de victoire et de match nul.
- En simulant tous les scénarios, MiniMax choisit le coup qui mène à la meilleure issue pour l’ordinateur, assurant ainsi une stratégie optimale.

4.1. Structure du Plateau

Le plateau est représenté par une liste de 9 cases (indices 0-8) :

4.2. Système d’évaluation



4.3. Algorithme MinMax

Etape 1 : Evaluation

- L'algorithme évalue chaque position possible du plateau.

Etape 2 : Maximation (O)

- L’ordinateur cherche le score maximum pour ses coups.

Etape 3 : Minimisation (X)

- Le joueur cherche le score maximum (du point de vue de l’ordinateur).

Etape 4 : Récursion

- L’algorithme simule tous les coups possibles jusqu’à la fin.

Annexe

1. Outils et environnement de développement

- **Langage Prolog :** SWI-Prolog
- **Langage Python :** Python 3.x
- **IDE utilisée :** Visual Studio Code
- **Tests effectués dans :** Terminal / Console Python

2. Ressources

Voici la liste des ressources consultées et utilisées pour la réalisation de ce mini-projet :

- **Cours de programmation logique (Prolog) :** Notes de cours fournies par le professeur et supports de l'université relatifs à la logique déclarative et aux bases de connaissances.
- **AI (Deep Seek & ChatGPT**) : utilisé comme assistant de compréhension et de documentation pour clarifier certains concepts théoriques liés à l'algorithme MiniMax.

3. Accès au projet

Veuillez cliquer ou copier-coller ce lien dans votre navigateur pour y accéder : Lien vers le projet