

# Devoir surveillé : IA pour les jeux (45 minutes)

Nom et Prénom :

Groupe :

## Présentation

---

- La classe `Board` vous est fournie en annexe. Elle implémente une grille de jeu pour Puissance 4.
- Vous pouvez utiliser les fonctions de la classe `Board` pour manipuler le plateau de jeu.
- Répondez directement sur ce document.

## Exercice 1.

---

En analysant la classe `Board` :

1. Quel est le facteur de branchement du jeu de Puissance 4 ?

Réponse :

2. Quelle est la profondeur maximale de l'arbre de jeu de Puissance 4 ?

Réponse :

## Exercice 2.

---

Le but de cet exercice est d'implémenter un algorithme MinMax à profondeur limitée.

1. Sans écrire le code complet de la méthode, proposez une heuristique `eval_board` permettant d'évaluer un plateau de jeu. Donnez la signature de la méthode et une explication de son fonctionnement.

Réponse :

2. Nous allons utiliser votre heuristique pour implémenter une évaluation minimax de l'arbre de jeu. Complétez le code de la méthode `minMax` ci-dessous (la fonction `undo_move()` défait la mouvement passé en paramètre).

```
1 def minMax(b, depth=3):
2
3
4
5     for col in
6         if b.is_valid_move(col):
7             b.make_move(col)
8
9
10
11         b.undo_move(col)
12
```

3. Quelle(s) autre(s) fonction(s) doit-on définir pour avoir une IA qui utilise MinMax pour jouer ? Donner juste la (ou les) signature(s) de la (ou des) fonction(s) ainsi qu'une explication concise de ce qu'elle(s) retourne(nt).

Réponse :

### Exercice 3.

---

On souhaite tester notre IA basée sur l'algorithme MinMax.

1. Ecrire le code pour implémenter un joueur qui joue aléatoirement.

Réponse :

2. Ecrire le code pour implémenter un match entre un MinMax à profondeur maximum 4 et un joueur aléatoire, le joueur MinMax commence.

Réponse :

```

1  class Board:
2      def __init__(self):
3          self.board = [" " for _ in range(7)] for _ in range(6)]
4          self.current_player = 1 # Player 1 starts
5          self.name = "Connect4"
6      def get_valid_moves(self):
7          moves = []
8          for j in range(7):
9              if self.board[0][j] == " ":
10                 moves.append(j)
11          return moves
12      def make_move(self, move):
13          for i in reversed(range(6)):
14              if self.board[i][move] == " ":
15                  self.board[i][move] = "X" if self.current_player == 1 else "O"
16                  self.current_player = 3 - self.current_player # Switch player
17                  break
18      def game_over(self):
19          # Check rows, columns, diagonals for a win or if board is full
20          # Return: (Boolean, Winner)
21          # Check horizontal locations for win
22          for r in range(6):
23              for c in range(4):
24                  if (self.board[r][c] == self.board[r][c+1]
25                      == self.board[r][c+2] == self.board[r][c+3] != " "):
26                      return True, self.board[r][c]
27          for c in range(7):
28              for r in range(3):
29                  if (self.board[r][c] == self.board[r+1][c] == self.board[r+2][c]
30                      == self.board[r+3][c] != " "):
31                      return True, self.board[r][c]
32          for c in range(4):
33              for r in range(3):
34                  if (self.board[r][c] == self.board[r+1][c+1] == self.board[r+2][c+2]
35                      == self.board[r+3][c+3] != " "):
36                      return True, self.board[r][c]
37          for c in range(4):
38              for r in range(3, 6):
39                  if (self.board[r][c] == self.board[r-1][c+1] == self.board[r-2][c+2]
40                      == self.board[r-3][c+3] != " "):
41                      return True, self.board[r][c]
42          if self.get_valid_moves() == []:
43              return True, None
44          return False, None
45      def is_game_over(self):
46          return self.game_over()[0]
47      def print_board(self):
48          for row in self.board:
49              print("|", end="")
50              for cell in row:
51                  symbol = cell if cell else " "
52                  print(f" {symbol} |", end="")
53          print("\n" + "-" * 29)

```