
Algorithmique et Programmation 1 – TD - TP 11

JEU DU MORPION CORRECTION

Le **morpion**¹ est un jeu de réflexion à deux joueurs se jouant sur une grille carrée de 3x3 cases. Les joueurs doivent remplir chacun à leur tour une case de la grille avec le symbole qui leur est attribué : 'O' ou 'X'. Le gagnant est celui qui arrive à aligner trois de ses symboles, horizontalement, verticalement ou en diagonale.

On suppose dans un premier temps avoir les fonctions suivantes :

- Une fonction `init_grille()` permettant d'initialiser la grille. Au début de la partie, les cases ne sont remplies par des espaces (' ')
- Une fonction `affiche_grille(grille)` permettant d'afficher la grille
- Une fonction `grille_pleine(grille)` permettant de tester si une grille est pleine
- Une fonction `case_libre(grille, ligne, col)` permettant de tester si une case est libre
- Une fonction `deposer_symbole(symbole, grille, ligne, col)` permettant d'inscrire un symbole sur la grille
- Une fonction `coup_gagnant(grille, ligne, col)` permettant de tester si un coup, joué dans la grille, est gagnant

On souhaite tout d'abord jouer humain contre humain. Voici la fonction permettant de lancer ce jeu :

```
1 def jeu2joueurs() :
2     """None --> None
3     Lance une partie morpions à 2 joueurs humains"""
4     grille = init_grille() #Initialisation de la grille
5     fin_partie = False #Booléen permettant de savoir quand finir la partie
6     tour = 1 #Compteur du tour
7     while not(fin_partie):
8         affiche_grille(grille) #Afficher la grille
9         if tour % 2 != 0: #Si le tour est pair, c'est au joueur 'X' de jouer
10             symbole = 'X'
11         else :
12             symbole = 'O' #Sinon, c'est au joueur 'O'
13         print("Tour", tour, "c'est au joueur", symbole, "de jouer")
14         # Récupération du coup du joueur
15         ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
16         col = int(input("Indiquez la colonne où déposer votre symbole : "))
17         while not(case_libre(grille, ligne, col)) :
18             #Vérification que le coup est valide, sinon on recommence
19             print("Cette case est déjà occupée")
20             ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
21             col = int(input("Indiquez la colonne où déposer votre symbole : "))
22         deposer_symbole(symbole, grille, ligne, col) #le coup est joué sur la grille
23         if coup_gagnant(grille, ligne, col) : #Vérification si le coup est gagnant
24             affiche_grille(grille)
25             print("Bravo, joueur", symbole, "a gagné !")
26             fin_partie = True
27         if grille_pleine(grille) : #Vérification si la grille est pleine
28             affiche_grille(grille)
```

1. ou **Tic-Tac-Toe** dans sa version anglaise

```

29         print("C'est un match nul, la grille est remplie!")
30         fin_partie = True
31         tour = tour + 1 #Incrémentation du tour

```

On souhaite à présent pouvoir jouer contre l'ordinateur.

1. Nous allons dans un premier temps programmer une fonction permettant à un joueur humain de jouer contre un joueur aléatoire, c'est à dire un joueur artificiel qui tire un coup au hasard parmi l'ensemble des coups possibles. Pour cela :

- (a) Programmez une fonction `liste_coups_possibles(grille)` qui retourne la liste des coups possibles. Utilisez une compréhension de listes.

```

def liste_coups_possibles(grille) :
    """List --> List
    Retourne la liste des coups possibles"""

    return [[i, j] for i in range(3) for j in range(3) if grille[i][j] == " "]

```

- (b) Modifiez la fonction `jeu2joueurs()` de façon à remplacer le joueur 'O' par un joueur qui joue un coup au hasard parmi l'ensemble des coups possibles

```

def jeu_ia_random() :
    """None --> None
    Lance une partie de morpions contre un joueur aléatoire"""
    grille = init_grille() #Initialisation de la grille
    fin_partie = False #Booléen permettant de savoir quand finir la partie
    tour = 1 #Compteur du tour

    while not(fin_partie):
        affiche_grille(grille) #Afficher la grille
        if tour % 2 != 0: #Si le tour est pair, c'est à l'humain de jouer
            symbole = 'X'
            print("Tour", tour, "c'est au joueur", symbole, "de jouer")
            # Récupération du coup du joueur
            ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
            col = int(input("Indiquez la colonne où déposer votre symbole : "))
            #Vérification que le coup est valide, sinon on recommence
            while not(case_libre(grille, ligne, col)) :
                print("Cette case est déjà occupée")
                ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
                col = int(input("Indiquez la colonne où déposer votre symbole : "))
            else : #C'est à l'ordi de jouer
                symbole = 'O'
                print("Tour", tour, "c'est à l'ordinateur de jouer")
                #calcul de tous les coups possibles
                coups_possibles = liste_coups_possibles(grille)
                #tirage aléatoire d'un coup
                coup_choisi = random.choice(coups_possibles)
                ligne = coup_choisi[0]
                col = coup_choisi[1]
                deposer_symbole(symbole, grille, ligne, col) #le coup est joué sur la grille
            if coup_gagnant(grille, ligne, col) : #Vérification si le coup est gagnant
                affiche_grille(grille)
                if symbole == 'X':
                    print("Bravo, vous avez gagné !")
                else:
                    print("Dommage, l'ordinateur vous a battu")
                fin_partie = True
            if grille_pleine(grille) : #Vérification grille pleine

```

```

    affiche_grille(grille)
    print("C'est un match nul, la grille est remplie!")
    fin_partie = True
    tour = tour + 1

```

2. Nous souhaitons à présent améliorer ce programme, en permettant au joueur humain de jouer contre un joueur artificiel (légèrement) plus élaboré. Pour cela, nous allons implémenter une intelligence artificielle très simple, qui suivra l'algorithme suivant :

- (a) Vérifie s'il existe un coup à jouer gagnant. Si oui, le jouer pour gagner
- (b) Vérifie si le joueur adverse a un coup gagnant. Si oui, le jouer pour éviter de perdre
- (c) Joue la case centrale si elle est libre²
- (d) Sinon, jouer au hasard parmi les coups possibles

Modifier la fonction précédente pour implémenter ce joueur.

```

def jeu_ia() :
    """None --> None
    Lance une partie de morpions contre une IA légèrement plus élaborée"""
    grille = init_grille()
    fin_partie = False
    tour = 1

    while not(fin_partie):
        affiche_grille(grille)
        if tour % 2 != 0:
            symbole = 'X'
            print("Tour", tour, "c'est au joueur", symbole, "de jouer")
            ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
            col = int(input("Indiquez la colonne où déposer votre symbole : "))
            while not(case_libre(grille, ligne, col)) :
                print("Cette case est déjà occupée")
                ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
                col = int(input("Indiquez la colonne où déposer votre symbole : "))
            else :
                symbole = 'O'
                print("Tour", tour, "c'est à l'ordinateur de jouer")
                coups_possibles = liste_coups_possibles(grille)

                #On regarde si un prochain coup est gagnant
                gagnant_possible = False
                nb_coups = len(coups_possibles)
                i = 0
                while i < nb_coups and not(gagnant_possible) :
                    coup = coups_possibles[i]
                    deposer_symbole('O', grille, coup[0], coup[1])
                    if coup_gagnant(grille, coup[0], coup[1]):
                        gagnant_possible = True
                        coup_choisi = coup
                    deposer_symbole(" ", grille, coup[0], coup[1])
                    i = i + 1
                if not(gagnant_possible) : #il n'y a pas de coup gagnant
                    #On regarde si un prochain coup de l'adversaire est gagnant
                    perdant_possible = False
                    i = 0
                    while i < nb_coups and not(perdant_possible) :
                        coup = coups_possibles[i]
                        deposer_symbole('X', grille, coup[0], coup[1])
                        if coup_gagnant(grille, coup[0], coup[1]):

```

2. cette case permettant de gagner sur 4 alignements différents

```

        perdant_possible = True
        coup_choisi = coup
        deposer_symbole(" ", grille, coup[0], coup[1])
        i = i + 1
    if not(perdant_possible) : #il n'y a pas de coup perdant
        #Jouer la case centrale si elle est libre
        if case_libre(grille, 1, 1) :
            coup_choisi = [1,1]
        else : #On joue au hasard
            coup_choisi = random.choice(coups_possibles)
        ligne = coup_choisi[0]
        col = coup_choisi[1]
        deposer_symbole(symbole, grille, ligne, col)
if coup_gagnant(grille, ligne, col) :
    affiche_grille(grille)
    if symbole == 'X':
        print("Bravo, vous avez gagné !")
    else:
        print("Dommage, l'ordinateur vous a battu")
    fin_partie = True
if grille_pleine(grille) :
    affiche_grille(grille)
    print("C'est un match nul, la grille est remplie!")
    fin_partie = True
tour = tour + 1

```

Le fichier `morpion.py`, téléchargeable sur Moodle, contient les fonctions mentionnées au début du TP. Mais ce fichier contient quelques erreurs.

3. Commencez par corriger les erreurs de syntaxe en vous appuyant sur les messages d'erreur que vous obtenez en exécutant ce programme
4. Corrigez ensuite les erreurs algorithmiques en testant **exhaustivement et individuellement** chacune des fonctions implémentées. Réfléchissez bien à tous les cas d'applications possibles pour ne rater aucune erreur !
5. Implémentez les deux fonctions vues précédemment pour pouvoir jouer contre l'ordinateur

```

import random
TAILLE = 3

# Fonction permettant d'initialiser la grille
def init_grille():
    """None --> List
    Retourne une grille de morpion (liste de listes) initialisée
    (remplie de " ")"""

    grille = []
    ligne = []
    #ERREUR : ligne doit être initialisé à chaque passage dans la première boucle for
    for i in range(TAILLE) :
        for j in range(TAILLE) :
            ligne.append(" ")
        grille.append(ligne)
    #ERREUR : oubli du return

# Fonction permettant d'afficher la grille
def affiche_grille(grille) :
    """List --> None
    Affiche la grille de morpion passée en paramètres"""

    print("    0    1    2")
    for i in range(TAILLE) #ERREUR : oubli d'un :
        print("    -    -    -")
        print(i, "|", end=" ")
        for j in range(TAILLE + 1): #ERREUR : range(TAILLE)
            print(grille[i][j], "|", end=" ")
        #ERREUR : Oubli du passage à la ligne

# Voici l'affichage attendu par cette fonction :
#    0    1    2
#    -    -    -
#0 |    |    |    |
#    -    -    -
#1 |    | X |    |
#    -    -    -
#2 |    | O |    |

#Fonction permettant de tester si une grille est pleine
def grille_pleine(grille) :
    """List --> Bool
    Vérifie si la grille de morpions passée en paramètre est pleine"""

    pleine = True
    #ERREUR : Oubli initialisation i
    while i <= TAILLE and pleine : #ERREUR : i < TAILLE
        j = 0
        while j < TAILLE or pleine : #ERREUR : test and
            if grille[i][j] == " " :
                pleine = False
            #ERREUR : oubli incrémentation j
            i = i + 1
        return pleine

#Fonction permettant de tester si une case est libre
def case_libre(grille, ligne, col) :
    """List x Int x Int --> Bool
    Vérifie si la case grille[ligne][col] du morpion est libre"""

```

```

    if grille[ligne][col] == " ": #ERREUR : = au lieu de ==
        return True
    else return False #ERREUR : oubli : et indentation

#Fonction permettant d'inscrire un symbole sur la grille
def deposer_symbole(symbole, grille, ligne, col) :
    """Str x List x Int x Int --> None
    Dépose le symbole sur la case de la grille (vérifiée libre ultérieurement)"""
    grille[ligne][col] = symbole #ERREUR indentation

#Fonction permettant de tester si une ligne est gagnante
def ligne_gagnante(grille, ligne) :
    """List x Int --> Bool
    Vérifie si une ligne est gagnante"""

    symbole = grille[ligne][0]
    #ERREUR : oubli initialisation variable gagnant

    #On vérifie si la ligne ne contient que ce symbole
    for j in range(1, TAILLE):
        if grille[ligne][j] == symbole : #ERREUR == au lieu de =
            gagnant = False

    return gagnant

#Fonction permettant de tester si une colonne est gagnante
def colonne_gagnante(grille, col) :
    """List x Int --> Bool
    Vérifie si une ligne est gagnante"""

    symbole = grille[col][0] #ERREUR : vérification ligne au lieu de col (2 fois)
    gagnant = True

    #On vérifie si la ligne ne contient que ce symbole
    for i in range(1, TAILLE):
        if grille[col][i] != symbole :
            gagnant = False

    return gagnant

#Fonction permettant de tester si une diagonale est gagnante
def diagonale_gagnante(grille, ligne, col) :
    """List x Int x Int --> Bool
    Vérifie si l'une des deux diagonales est gagnante"""
    symbole = grille[ligne][col]
    gagnant = True

    if ligne == col : #diagonale (0,0);(1,1);(2,2)
        for i in range(TAILLE):
            if grille[i][i] != symbole #ERREUR oubli :
                gagnant = False
            if ligne == 1 : #ça peut être la seconde diagonale :
                gagnant = True
    elif ligne != col or ligne == 1 : #diagonale (0,2);(1,1);(2,0)
        #ERREUR : if et pas elif : on ne teste pas si ligne == col == 1 sinon
        for i in range(TAILLE: #ERREUR : parenthèse
            if grille[i][TAILLE - 1 - i] != symbole :
                gagnant = False
        return gagnant

#Fonction permettant de tester si un coup est gagnant
def coup_gagnant(grille, ligne, col) :

```

```

"""List x Int x Int --> Bool
Vérifie si le dernier coup joué est gagnant"""

symbole = grille[ligne][col]
gagnant = True

#On vérifie si la ligne est gagnante
if ligne_gagnante(grille, ligne : #ERREUR : parenthèse
    return True
#on vérifie si la colonne est gagnante:
elif colonne_gagnante(grille, col) :
    return True #ERREUR : indentation
#sinon, on vérifie si la case est sur une diagonale
#diagonale si la somme du numéro de la ligne est de la colonne est paire
elif (col + ligne) // 2 == 0 : #ERREUR : // au lieu de %
    return diagonale_gagnante(grille, col) #ERREUR appel fonction
else: #rien n'est gagnant
    return False

#ATTENTION : penser à vérifier ici toutes les lignes, toutes les colonnes et
#toutes les diagonales, en partant de toutes les cases possibles!

#Fonction permettant de jouer humain contre humain
def jeu2joueurs() :
    """None --> None
    Lance une partie morpions à 2 joueurs humains"""
    grille = init_grille() #Initialisation de la grille
    fin_partie = False #Booléen permettant de savoir quand finir la partie
    tour = 1 #Compteur du tour

    while not(fin_partie):
        affiche_grille(grille) #Afficher la grille
        if tour % 2 != 0: #Si le tour est pair, c'est au joueur 'X' de jouer
            symbole = 'X'
        else :
            symbole = 'O' #Sinon, c'est au joueur 'O'
        print("Tour", tour, "c'est au joueur", symbole, "de jouer")
        # Récupération du coup du joueur
        ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
        col = int(input("Indiquez la colonne où déposer votre symbole : "))
        while not(case_libre(grille, ligne, col)) :
            #Vérification que le coup est valide, sinon on recommence
            print("Cette case est déjà occupée")
            ligne = int(input("Indiquez la ligne où déposer votre symbole : "))
            col = int(input("Indiquez la colonne où déposer votre symbole : "))
        deposer_symbole(symbole, grille, ligne, col) #le coup est joué sur la grille
        if coup_gagnant(grille, ligne, col) : #Vérification si le coup est gagnant
            affiche_grille(grille)
            print("Bravo, joueur", symbole, "a gagné !")
            fin_partie = True
        if grille_pleine(grille) : #Vérification si la grille est pleine
            affiche_grille(grille)
            print("C'est un match nul, la grille est remplie!")
            fin_partie = True
        tour = tour + 1 #Incrémentation du tour

#Programme permettant de lancer le jeu
print("Bienvenue sur le jeu du morpion!")
jeu2joueurs()

```