



INSTITUT SUPERIEUR POLYTECHNIQUE DE MADAGASCAR
TP JAVA HS04 – Jeu Morpion

Objectifs :

Savoir gérer une grille 2D, manipuler des événements Swing, implémenter une IA par heuristique simple.

- 1) Implémenter **UN** des codes sources suivants (ou les deux), dans un projet JAVA, placer le package convenablement :

MorpionConsoleIA.java

```
1  package morpion12;
2  import java.util.*;
3
4  public class MorpionConsoleIA {
5      static final int N = 3;
6      static final char VIDE = '.';
7      static final char HUMAIN = 'X';
8      static final char IA = 'O';
9      static final Scanner sc = new Scanner(System.in);
10     static char[][] board = new char[N][N];
11
12     public static void main(String[] args) {
13         init();
14         System.out.println("=== MORPION (vous = X, IA = O) ===");
15         System.out.println("Entrez une case 1..9 comme ceci :");
16         System.out.println(" 1 2 3\n 4 5 6\n 7 8 9\n");
17
18         while (true) {
19             afficher();
20             tourHumain();
21             if (win(HUMAIN)) {afficher(); System.out.println("Bravo ! Vous avez gagné."); break; }
22             if (plein())      {afficher(); System.out.println("Match nul."); break; }
23
24             tourIA();
25             if (win(IA))      {afficher(); System.out.println("L'IA gagne. Courage !"); break; }
26             if (plein())      {afficher(); System.out.println("Match nul."); break; }
27         }
28     }
29
30     // ----- Jeu -----
31     static void tourHumain() {
32         while (true) {
33             System.out.print("Votre coup (1..9) : ");
34             if (!sc.hasNextInt())
35                 { sc.next(); System.out.println("Veuillez entrer un nombre."); continue; }
36             int pos = sc.nextInt();
37             if (pos < 1 || pos > 9) { System.out.println("Entre 1 et 9, svp."); continue; }
38             int r = (pos - 1) / 3, c = (pos - 1) % 3;
39             if (board[r][c] != VIDE) { System.out.println("Case déjà occupée."); continue; }
40             board[r][c] = HUMAIN;
41             return;
42         }
43     }
44
45     static void tourIA() {
46         int[] m;
47         // 1) Gagner si possible
48         m = coupGagnant(IA);
49         if (m != null) { board[m[0]][m[1]] = IA; return; }
50         // 2) Bloquer l'humain
51         m = coupGagnant(HUMAIN);
52         if (m != null) { board[m[0]][m[1]] = IA; return; }
53         // 3) Prendre le centre
54         if (board[1][1] == VIDE) { board[1][1] = IA; return; }
55         // 4) Prendre un coin
```

```

56     int[][] coins = {{0,0},{0,2},{2,0},{2,2}};
57     m = premierLibre(coins);
58     if (m != null) { board[m[0]][m[1]] = IA; return; }
59
60     // 5) Prendre un côté
61     int[][] cotes = {{0,1},{1,0},{1,2},{2,1}};
62     m = premierLibre(cotes);
63     if (m != null) { board[m[0]][m[1]] = IA; return; }
64     // 6) Fallback (ne devrait pas arriver)
65     for (int r = 0; r < N; r++)
66         for (int c = 0; c < N; c++)
67             if (board[r][c] == VIDE) { board[r][c] = IA; return; }
68 }
69
70 // ----- Outils IA -----
71 static int[] coupGagnant(char joueur) {
72     for (int r = 0; r < N; r++) {
73         for (int c = 0; c < N; c++) {
74             if (board[r][c] != VIDE) continue;
75             board[r][c] = joueur;
76             boolean gagne = win(joueur);
77             board[r][c] = VIDE;
78             if (gagne) return new int[]{r, c};
79         }
80     }
81     return null;
82 }
83
84 static int[] premierLibre(int[][] cases) {
85     for (int[] rc : cases) {
86         if (board[rc[0]][rc[1]] == VIDE) return rc;
87     }
88     return null;
89 }
90
91 // ----- Logique de plateau -----
92 static void init() {
93     for (int r = 0; r < N; r++) Arrays.fill(board[r], VIDE);
94 }
95
96 static boolean plein() {
97     for (int r = 0; r < N; r++)
98         for (int c = 0; c < N; c++)
99             if (board[r][c] == VIDE) return false;
100     return true;
101 }
102
103 static boolean win(char p) {
104     // Lignes/colonnes
105     for (int i = 0; i < N; i++) {
106         if (board[i][0] == p && board[i][1] == p && board[i][2] == p) return true;
107         if (board[0][i] == p && board[1][i] == p && board[2][i] == p) return true;
108     }
109     // Diagonales
110     if (board[0][0] == p && board[1][1] == p && board[2][2] == p) return true;
111     if (board[0][2] == p && board[1][1] == p && board[2][0] == p) return true;
112     return false;
113 }
114
115 static void afficher() {
116     System.out.println("\nPlateau :");
117     for (int r = 0; r < N; r++) {
118         for (int c = 0; c < N; c++) {
119             System.out.print(" " + board[r][c]);
120         }
121         System.out.println();
122     }
123     System.out.println();
124 }
125 }

```

MorpionSwing.java

```
1  package morpion12;
2
3  import javax.swing.*;
4  import java.awt.*;
5
6  public class MorpionSwing extends JFrame {
7      private final JButton[][] cells = new JButton[3][3];
8      private char current = 'X';
9
10     public MorpionSwing() {
11         super("Morpion - GridLayout");
12         setDefaultCloseOperation(EXIT_ON_CLOSE);
13         setLayout(new GridLayout(3, 3));
14         initBoard();
15         setSize(320, 360);
16         setLocationRelativeTo(null);
17         setResizable(false);
18         setVisible(true);
19     }
20
21     private void initBoard() {
22         Font big = new Font("Arial", Font.BOLD, 48);
23         for (int i = 0; i < 3; i++) {
24             for (int j = 0; j < 3; j++) {
25                 JButton b = new JButton();
26                 b.setFont(big);
27                 b.setFocusPainted(false);
28                 final int r = i, c = j;
29                 b.addActionListener(e -> jouer(r, c));
30                 cells[i][j] = b;
31                 add(b);
32             }
33         }
34     }
35
36     private void jouer(int r, int c) {
37         JButton b = cells[r][c];
38         if (!b.getText().isEmpty()) return; // déjà joué
39         b.setText(String.valueOf(current));
40
41         if (gagne(current)) {
42             fin(current + " a gagné !");
43             return;
44         }
45         if (plein()) {
46             fin("Match nul.");
47             return;
48         }
49         current = (current == 'X') ? 'O' : 'X';
50     }
51
52     private boolean gagne(char p) {
53         String s = String.valueOf(p);
54         // Lignes et colonnes
55         for (int i = 0; i < 3; i++) {
56             if (cells[i][0].getText().equals(s) &&
57                 cells[i][1].getText().equals(s) &&
58                 cells[i][2].getText().equals(s)) return true;
59
60             if (cells[0][i].getText().equals(s) &&
61                 cells[1][i].getText().equals(s) &&
62                 cells[2][i].getText().equals(s)) return true;
63         }
64         // Diagonales
65         if (cells[0][0].getText().equals(s) &&
66             cells[1][1].getText().equals(s) &&
67             cells[2][2].getText().equals(s)) return true;
68
69         if (cells[0][2].getText().equals(s) &&
```

```

70         cells[1][1].getText().equals(s) &&
71         cells[2][0].getText().equals(s)) return true;
72
73     return false;
74 }
75
76 private boolean plein() {
77     for (int r = 0; r < 3; r++)
78         for (int c = 0; c < 3; c++)
79             if (cells[r][c].getText().isEmpty()) return false;
80     return true;
81 }
82
83 private void fin(String msg) {
84     for (JButton[] row : cells)
85         for (JButton b : row) b.setEnabled(false);
86
87     int choice = JOptionPane.showConfirmDialog(
88         this, msg + " Rejouer ?", "Fin de partie",
89         JOptionPane.YES_NO_OPTION);
90
91     if (choice == JOptionPane.YES_OPTION) reset();
92     else dispose();
93 }
94
95 private void reset() {
96     for (JButton[] row : cells)
97         for (JButton b : row) { b.setText(""); b.setEnabled(true); }
98     current = 'X';
99 }
100
101 public static void main(String[] args) {
102     SwingUtilities.invokeLater(MorpionSwing::new);
103 }
104 }

```

- 2) Expliquer ce qui se passe à la ligne 38 (et 39) du fichier **MorpionConsoleIA.java**.
- 3) Expliquer ce qui se passe à la ligne 29 du fichier **MorpionSwing.java**.
- 4) En vous inspirant de ces deux extraits de codes, implémenter le jeu Morpion sur swing avec IA.
- 5) Refactoriser le code pour être le plus en accord possible avec les principes de la Programmation Orientée Objet (selon votre conception – vous pouvez sauter cette question si vous ne trouvez rien à modifier).
- 6) Améliorer l'IA et/ou personnaliser votre Interface graphique du Jeu (par exemple, mettre la ligne gagnante en surbrillance).