

"Tic Tac Toe"

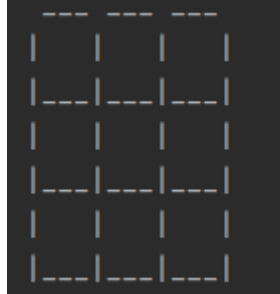
Ou le jeu du Morpion

Ce TP a été le premier long exercice sur lequel j'ai pu travailler en Java.

**Il nous a été donné par Mr Lefebvre Pierre,
Professeur de programmation Python, Java et C.**

Présentation d'une partie :

Mise en place de la partie :



Début de partie :

```
--- --- ---
|  |  |  |
|---|---|---|
|  |  |  |
|---|---|---|
|  |  |  |
|---|---|---|

tour n°1
Joueur 1
donnez la ligne (1-3):
1
donnez la colonne (1-3):
1

--- --- ---
| 0 |  |  |
|---|---|---|
|  |  |  |
|---|---|---|
|  |  |  |
|---|---|---|

tour n°2
Joueur 2
donnez la ligne (1-3):
```

Le joueur 1 à demandé une case déjà sélectionnée :

```

  ---  ---  ---
  | 0 | X |   |
  |__|__|__|
  |   | 0 |   |
  |__|__|__|
  |   |   | X |
  |__|__|__|
tour n°5
Joueur 1
donnez la ligne (1-3):
1
donnez la colonne (1-3):
2
case déjà prise, essaie encore.

  ---  ---  ---
  | 0 | X |   |
  |__|__|__|
  |   | 0 |   |
  |__|__|__|
  |   |   | X |
  |__|__|__|
tour n°5
Joueur 1
donnez la ligne (1-3):
```

Le joueur 2 comprend qu'il va perdre car le joueur 1 a deux possibilités :

```

  ---  ---  ---
  | 0 | X |   | |
  |  |  |  |  |
  | 0 | 0 |   |
  |  |  |  |  |
  |  |   | X |
  |  |  |  |  |
```

Fin de partie :

```

  _ _ _ _ _
| 0 | X |   |
|_ _ |_ _ |_ _|
| 0 | 0 | 0 |
|_ _ |_ _ |_ _|
| X |   | X |
|_ _ |_ _ |_ _|
Bravo Joueur 1, vous avez gagné.

```

Le code source et leur action

La classe Main :

```
1  import java.util.Scanner;
2  |
3  ▶ public class Main {
4      static Scanner scan = new Scanner(System.in);
5      static final String empty = " ";
6      static final String PLAYER1 = "O";
7      static final String PLAYER2 = "X";
8
9  ▶  public static void main(String args[]) {
10     String player = "Joueur 2";
11     String symbol = PLAYER2;
12     String[][] tabCase = initialize(empty);
13     play(tabCase, player, symbol);
14
15     }
```

- Déclaration des méthodes réutilisées durant tout le code
- Mise en place des méthodes player, symbol et du tableau tabCase :
player équivaut au joueur dont c'est le moment de jouer
symbol : X ou O
tabCase : le tableau de jeu sur lequel la partie se déroule
- Appel la fonction initialize() avec paramètre empty
- Appel la fonction play

La fonction play :

```
public static void play(String[][] tabCase, String player, String symbol) {
    int tour = 1;
    String win = " ";
    while (win != "Joueur 1" & win != "Joueur 2" & tour != 9) {
        String caseEmpty = player;
        player = (player == "Joueur 1") ? "Joueur 2" : "Joueur 1";
        symbol = (symbol == PLAYER1) ? PLAYER2 : PLAYER1;
        while (caseEmpty != empty) {
            display(tabCase);
            System.out.print("tour n°");
            System.out.println(tour);
            System.out.println(player);
            System.out.println("donnez la ligne (1-3):");
            int ligne = scan.nextInt() - 1;
            System.out.println("donnez la colonne (1-3):");
            int colonne = scan.nextInt() - 1;
            if (tabCase[ligne][colonne] == empty) {
                caseEmpty = empty;
                tabCase[ligne][colonne] = (player == "Joueur 1") ? PLAYER1 : PLAYER2;
            } else {
                System.out.println("case déjà prise, essaie encore.");
                caseEmpty = player;
            }
        }
        win = victory(tabCase, player, symbol);
        tour++;
    }
}
```

- Déclaration des paramètres tour et win :
tour compte le nombre de tours dans une partie
win peut être vide ou contenir joueur 1 ou joueur 2
- La première boucle While continue la partie tant que personne n'a gagné.
la condition tour sert de sécurité
- caseEmpty permet de connaître si un joueur a déjà placé un pion dessus ou si la case est vide
- player est un opérateur ternaire pour changer de joueur
- symbol est aussi un ternaire, il change le symbole
- La deuxième boucle While écrit les instructions et vérifie que la case n'est pas déjà prise
- Appel de la fonction win() avec pour paramètres tabCase, player et symbol
- Tour incrémente de 1

Fonction Victory :

```
52 public static String victory(String[][] tabCase, String player, String symbol) {
53     String win = null;
54     for (int i = 0; i < 3; i++) {
55         if (tabCase[i][0] == symbol) {
56             if (tabCase[i][1] == symbol) {
57                 if (tabCase[i][2] == symbol) {
58                     display(tabCase);
59                     System.out.println("Bravo " + player + ", vous avez gagné.");
60                     win = player;
61                 }
62             }
63         }
64         if (tabCase[0][i] == symbol) {
65             if (tabCase[1][i] == symbol) {
66                 if (tabCase[2][i] == symbol) {
67                     display(tabCase);
68                     System.out.println("Bravo " + player + ", vous avez gagné.");
69                     win = player;
70                 }
71             }
72         }
73     }
74     if (tabCase[0][0] == symbol) {
75         if (tabCase[1][1] == symbol) {
76             if (tabCase[2][2] == symbol) {
77                 display(tabCase);
78                 System.out.println("Bravo " + player + ", vous avez gagné.");
79                 win = player;
80             }
81         }
82     }
83     if (tabCase[0][2] == symbol) {
84         if (tabCase[1][1] == symbol) {
85             if (tabCase[2][0] == symbol) {
86                 display(tabCase);
87                 System.out.println("Bravo " + player + ", vous avez gagné.");
88                 win = player;
89             }
90         }
91     }
92     return win;
93 }
```

La fonction victory permet de définir s'il y a un vainqueur ou non
il retournera le joueur gagnant ou null s'il n'y en a pas

Fonctions initialize et display :

```
117 @ public static String[][] initialize(String initChar) {
118     return new String[][] { { initChar, initChar, initChar }, { initChar, initChar, initChar },
119                             { initChar, initChar, initChar } };
120 }
121
122 @ public static void display(String[][] tabCase) {
123     System.out.println(" --- --- --- ");
124     for (int i = 0; i < tabCase.length; i++) {
125         for (int j = 0; j < tabCase[i].length; j++) {
126             System.out.print(" | ");
127             System.out.print(tabCase[i][j]);
128         }
129         System.out.println(" |");
130         System.out.println(" |___|___|___|");
131     }
132 }
133 }
```

La fonction initialize définit le contenu du tableau

La fonction display affiche le tableau de jeu ainsi que les pions s'il y en a.

Ce que ce projet m'a apporté :

Avec cet exercice, j'ai pu commencer à appréhender ce qu'est un projet, ses difficultés autant en termes de temps que de code.

Je n'étais pas vraiment habitué à coder en java et ça n'a pas été simple mais c'était gratifiant d'arriver à faire un jeu par soi-même.

Ce qui pourrait être amélioré :

- Refactoriser de façon orientée objet
 - Donner la possibilité de se donner un pseudo
- Faire apparaître les valeurs des abscisses et ordonnées
- Passer l'abscisse de chiffres à lettres pour plus de compréhension
 - Mettre en place un système de nouvelle partie
 - Compter les points et les conserver (score)
 - Enregistrer les pseudos et leurs points
 - Faire une interface utilisateur