

# Traitement du tour des joueurs

Temps de lecture : 1 minute

## Gestion des joueurs

Afin d'avoir la notion de joueurs dans le Morpion, l'énumération suivante est créée :

```
public enum Player {  
    FIRST, SECOND  
}
```

Cela nous permet d'ajouter une variable joueur comme nous pouvons le voir dans la méthode main ci-dessous.

## Code de la classe Main

À cette étape, voici le code de la classe Main :

```
public class Main {  
    public static void main(String[] args) {  
  
        final var scanner = new Scanner(System.in);  
        final var game = new TicTacToe();  
  
        var player = Player.FIRST;  
  
        while (true) {  
            System.out.println(game);  
            System.out.println("Veuillez saisir un des chiffres [1-9] :");  
            final var playerInput = scanner.nextInt();  
  
            game.processInput(player, playerInput);  
  
            player = nextPlayer(player);  
        }  
    }  
}
```

```

    }

    private static Player nextPlayer(Player player) {
        if (player.equals(Player.FIRST)) {
            return Player.SECOND;
        } else {
            return Player.FIRST;
        }
    }
}

```

Une variable `player` est ajoutée pour la gestion des deux joueurs. Elle a deux valeurs possible puisque basée sur l'énumération créée précédemment. A chaque fin de boucle, la méthode `nextPlayer` permet de passer de `Player.FIRST` à `Player.SECOND` ou inversement.

À chaque tour de boucle, la méthode `game.processInput` est appelée afin de sélectionner une case de la grille de jeu et de remplir d'un 'X' ou d'un 'O' en fonction du joueur qui joue le tour.

## Code de la classe TicTacToe

À cette étape, voici le code de la classe `TicTacToe` :

```

public class TicTacToe {

    private char[][] grid = new char[][]{
        {'.', '.', '.'},
        {'.', '.', '.'},
        {'.', '.', '.'}
    };

    public void processInput(Player player, int playerInput) {
        final var row = (playerInput - 1) / 3;
        final var column = (playerInput - 1) % 3;
        if (grid[row][column] == '.') {
            if (player.equals(Player.FIRST)) {
                grid[row][column] = 'X';
            } else {
                grid[row][column] = 'O';
            }
        }
    }
}

```

```

    }
}

@Override
public String toString() {
    final var builder = new StringBuilder();
    builder.append("Grille du Morpion : ").append(LINE
_SEPARATOR);
    for (char[] line : grid) {
        for (char cell : line) {
            builder.append(SPACE).append(cell).append
(SPACE);
        }
        builder.append(LINE_SEPARATOR);
    }
    return builder.toString();
}
}

```

Une nouvelle méthode `processInput` est définie pour définir la logique de tour par tour en fonction de quel joueur joue.

## Sélection de la cellule dans la grille

Afin de définir quelle cellule de la grille le joueur a sélectionné, nous avons deux calculs mathématiques successifs :

$row = (playerInput - 1) / 3$  : pour sélectionner l'index de la rangée, nous utilisons une division entière  $/$ . Pour rappel, la division entière permet de retourner le quotient entier d'une division. Par exemple,  $2 / 3 = 0$  ou  $5 / 3 = 1$ . Comme les index débutent à 0 nous devons retirer 1 au choix de la case par l'utilisateur.

Ainsi, s'il choisit la première case en tapant 1, cela nous donne 0 et  $0 / 3 = 0$ . Nous aurons donc bien la première rangée.

$column = (playerInput - 1) \% 3$  : pour obtenir la colonne nous utilisons l'opérateur modulo permettant d'obtenir le reste d'une division. Par exemple,  $2 \% 3 = 2$  ou  $5 \% 3 = 2$ .

Ainsi, par exemple, si le joueur choisit la première case nous aurons bien  $0 \% 3 = 0$  donc la première colonne.

Prenons un autre exemple, le joueur veut sélectionner la case du milieu, donc la 5ème case.

Nous avons donc pour la rangée :  $4 // 3 = 1$ , et pour la colonne  $4 \% 3 = 1$ .

`grid[1][1]` donne bien la cinquième case de la grille.

Ces astuces de calcul nous permettent d'obtenir un code bien plus efficace et concis que l'algorithme suivant que l'on pourrait trouver plus intuitivement :

```
if (playerInput == 1) {  
    row = 0; column = 0;  
} else if (playerInput == 2) {  
    ...
```

## Modification de la grille

Une fois la cellule calculée comme expliqué ci-dessus, on vérifie que celle-ci est bien vide (donc égale à `'.'`).

Ensuite, si le premier joueur est celui qui joue ce tour-ci, on remplit la cellule avec une `'X'`, sinon on la remplit avec un `'O'`.