

Conditions de victoire

Temps de lecture : 1 minute

Code de la classe TicTacToe

Nouvelle étape 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';
            }
        }
    }

    public boolean checkWin() {
        for (int i = 0; i < 3; i++) {
            var checkWinLine = grid[i][0] == grid[i][1] &&
grid[i][1] == grid[i][2] && grid[i][2] != '.';
            var checkWinColumn = grid[0][i] == grid[1][i]
&& grid[1][i] == grid[2][i] && grid[2][i] != '.';
            if (checkWinLine || checkWinColumn) {
                return true;
            }
        }
    }
}
```

```

    }
    var checkWinDiagonal1 = grid[0][0] == grid[1][1] &
& grid[1][1] == grid[2][2] && grid[2][2] != '.';
    var checkWinDiagonal2 = grid[0][2] == grid[1][1] &
& grid[1][1] == grid[2][0] && grid[2][0] != '.';
    if (checkWinDiagonal1 || checkWinDiagonal2) {
        return true;
    }
    return false;
}

@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 checkWin est définie pour vérifier si un joueur a réussi à aligner sur la même ligne 3 fois le même symbole.

Pour cela, on vérifie toutes les configurations possibles sur lesquels il est possible d'avoir une ligne :

- les lignes
- les colonnes
- les deux diagonales

Pour chacune de ces configurations, un opérateur logique OU est appliqué (on gagne si une au moins une ligne est vérifiée).

Pour chacune de ces configurations, on utilise l'opérateur logique ET pour vérifier les cellules individuellement (Toutes les cellules d'une même ligne doivent avoir la même valeur qui n'est pas un ' ').

Code de la classe Main

Nouvelle étape 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);  
            if (game.checkWin()) {  
                System.out.println(game);  
                System.out.println("Le joueur " + player +  
" a gagné la partie ! :");  
                break;  
            }  
  
            player = nextPlayer(player);  
        }  
    }  
  
    private static Player nextPlayer(Player player) {  
        if (player.equals(Player.FIRST)) {  
            return Player.SECOND;  
        } else {  
            return Player.FIRST;  
        }  
    }  
}
```

Copier

Il y a un nouvel ajout pour appeler la méthode `game.checkWin`. Si elle retourne `true`, on affiche l'état du jeu et quel joueur a gagné, puis on quitte la boucle grâce à l'instruction `break`. Sinon, rien ne se passe et le jeu continue.