Mohamed **(Game 2)**	**Constructor: Initializes the game board as a 6x7 grid with blank spaces.** **update_board: Updates the game board with the player's mark (X or O) at the specified position if the move is valid.** **display_board: Prints the current state of the game board to the console, showing the positions of the marks.** **is_winner: Checks if a player has won the game by examining rows, columns, and diagonals for a sequence of four marks.** **is_draw: Determines if the game is a draw, checking if the board is full and no winner has been declared.** **game_is_over: Checks if the game has ended by evaluating if the maximum number of moves has been reached or if there's a winner.** **This code ensures that the game functionalities (updating the board, checking for a winner, determining draws, and ending the game) work properly within the rules of a Connect Four-like game on a 6x7 grid.**
Ammar **(Game 1)**	**Constructor: Initializes the game board with a 3x5 grid, setting all cells to blank spaces.** **update_board: Updates the game board with the player's mark (X or O) at the specified position if the move is valid.** **display_board: Prints the current state of the game board to the console, showing the positions of the marks. The board structure reflects a pyramid shape.** **is_winner: Checks for winning patterns specific to the game's unique rules, considering various possible winning configurations involving adjacent cells.** **is_draw: Determines if the game has resulted in a draw by verifying if the maximum number of moves has been reached and no winner has been declared.** **game_is_over: Determines if the game has ended by checking if the maximum number of moves has been reached.**

	This code allows for playing and verifying wins or draws in a specialized version of Tic Tac Toe with a distinct board structure.
Mohab (Game 3)	**Constructor: Initializes the game board with a 5x5 grid, setting all cells to empty.** **update_board: Updates the board with the player's mark at the specified position if the move is valid.** **display_board: Prints the current state of the board to the console, displaying the positions of the marks in a 5x5 grid format.** **is_winner: Checks for winning patterns in rows, columns, and both diagonals for 'X' and 'O'. It increments counters for 'X' and 'O' whenever a winning sequence is found and determines the winner based on the counter values.** **is_draw: Checks if the game has resulted in a draw, indicating that all 24 moves have been made, and 'X' and 'O' have an equal number of winning sequences.** **game_is_over: Checks if the game is over by verifying if the maximum number of moves (24) has been reached.** **Additionally, the is_winner function prints the winner and the counters for 'X' and 'O' if the game has concluded, and one player has more winning sequences than the other.**