Team members:

Name : Ahmed mahmoud hasaneen Mahdi , id :20220039

Name : doaa ayman mohamed ali, id :20220124

Name : Nour El-dien Mohamed Rabie, id :20220364

Ahmed mahmoud hasaneen Mahdi did the pyramic board and its main and helped at integrating the code boards with each other.

doaa ayman mohamed ali did the concet_four board and its main and helped at integrating the code boards with each other.

Nour El-dien Mohamed Rabie did the 5X5 board and its main and helped at integrating the code boards with each other.

Name : doaa ayman mohamed ali, id :20220124 (report about pyramic board)

pyramic board constructor:

initialize number of moves, rows and columns as number of rows=3 ,columns=5 , moves=0.

update board :

it checks if the value that the user entered is valid or not

if it is valid , increment number of moves and put the symbol in the place that the user has chosen .

but here we can remove the first condition in the line 30.

display board :

display the current state of the board.

but i see it will be better if we added indexs in the cells of the game .

is winners :

it checks if there 3 symbols are equale horizontally ,vertically or diagonally .

is draw :

it checks if the moves became 9 (the maximum number of moves) .

game is over :

it makes the same thing of is draw function

Name : doaa ayman mohamed ali, id :20220124 (report about 5x5 board)

5x5 game report

costructor :

initialze number of rows and columns as number of rows = number of columns = 5

update board :

it checks if the value that the user entered is valid or not

if it is valid , increment number of moves and put the symbol in the place that the user has chosen .

display board :

display the current state of the game .

count function :

this function uses the struct and for loops to count the number of winners either X or O .

is winner :

it calls count function and checks if O or X won , print it and return true if one of them won.

is draw :

it checks if the moves became 24 (the maximum number of moves) and there is no winners .

game is over :

it return true when number of moves >= 24 (the maximum number of moves) .

After reading the code , i did not found any mistakes .

the code is easy to read and formatted correctly .

Name : Ahmed mahmoud hasaneen , id :20220039 (report about 5x5 board)

board_5x5::board_5x5():

The class constructor initializes a 5x5 board with all elements set to 0. It dynamically allocates memory for the board using a two-dimensional array of characters.

Update Board:

The Update Board method allows a player to make a move on the board. It checks if the specified position is within bounds and not already occupied before updating the board.

Display Board:

The Display Board method prints the current state of the board in a formatted manner.

Count Winners and Check Winner:

The Count method checks for winning combinations on the board and returns a `winners` struct containing the counts for 'X' and 'O'.

The Check Winner method determines the winner based on the counts obtained from the `count` method.

Check Draw and Game Over:

The is_draw and game_is_over methods determine if the game is a draw or if it is over based on the number of moves.

Potential Optimization

While the code is well-structured and functional, there is room for optimization. The most significant potential improvement lies in the `count` method, which currently uses nested loops to check for winning combinations. A more efficient algorithm, possibly utilizing a loop with a fixed set of winning patterns, might reduce the time complexity.

Use a predefined set of winning patterns for rows, columns, and diagonals.

Iterate through the patterns and check for matches on the board.

The optimization may involve creating a constant array of winning patterns and iterating through them to count matches. This can reduce the complexity from nested loops to a single loop, potentially improving performance.

Name : Ahmed mahmoud hasaneen,id :20220039(report about conecet board)

connect_four_board::connect_four_board():

The constructor initializes a 6x7 board with all elements set to 0. It dynamically allocates memory for the board using a two-dimensional array of characters.

Update Board:

The Update Board method allows a player to make a move by placing a marker in the specified column. It checks if the column is valid and updates the board accordingly.

Display Board:

The Display Board method prints the current state of the board in a formatted manner.

Check Winner: game_is_over

The is_Winner method checks for winning combinations in horizontal, vertical, and diagonal directions on the board.

Check Draw and Game Over:

The is_draw and game_is_over methods determine if the game is a draw or if it is over based on the number of moves.

Potential Improvements:

1-Bitwise Operations in Update: The check board[i][x - 1] == 0 in the update_board method can be simplified to !board[i][x - 1]. This avoids the need for a comparison with zero.

2-Redundant Diagonal Check Indices: The diagonal checks in the is_winner **method could be made more consistent by adjusting the loop indices to cover the entire board.**

3-Early Exit in Winner Check: Similar to the Tic-Tac-Toe case, consider adding an early exit when a winner is found in the is_winner **method to avoid unnecessary checks.**

4- Optimizing Draw Check: The is_draw method can be simplified by checking if the number of moves is equal to the total number of cells on the board (n_rows * n_cols).

Name : Nour El-dien Mohamed Rabie, id :20220364 (report about concet board)

Connect four Board:

1. **The constractor :**
   the number of rows and columns was assigned as 6 and 7 as in the rule of the game and fill the board with zeroes as beginning.

2. **Update board function:**
   checks if the input cell is valid by checking if input number of column is between 1 and 7 and it's
   empty then it's loop on the every row to get the first empty row in the column and if the whole
   columns are selected it's return false.

3. **Display board:**
   It shows the last updated board.

4. **Is winner:**
   checks if there is four similar symbols in row in every row and every column and every possible diagonal case.

5. **Is draw:**
   checks if there is no winner by checking if all cells are filled and there is no winner.

6. **Game is over:**
   checks if number of moves has reached its maximum to end the game.

Name : Nour El-dien Mohamed Rabie, id :20220364 (report about pyramic board)

Pyramid board:

1. **Constructor :**
   initialize number of moves, rows and columns as number of rows=3 ,columns=5 , moves=0.

2. **Update board :**
   checks if the input number is valid and the chosen cell is empty .but there is no need for the first if condition if(n_moves==9) return false.

3. **Display board :**
   it shows the last updated board .but we can add index number to the cells to make it easy to the user to enter a valid input.
4. **Is winner :**
   checks all the possible wins cases diagonally horizontally and vertically .
5. **Is draw :**
   checks if number of moves has reached its maximum and there is no winner.

Game is over :
checks if number of moves has reached its maximum to end the game.