Faculty of Computers and Artificial Intelligence

Cairo University

OOP Course

Under the Supervision of Dr. Mohamed Elramly

Section : 21

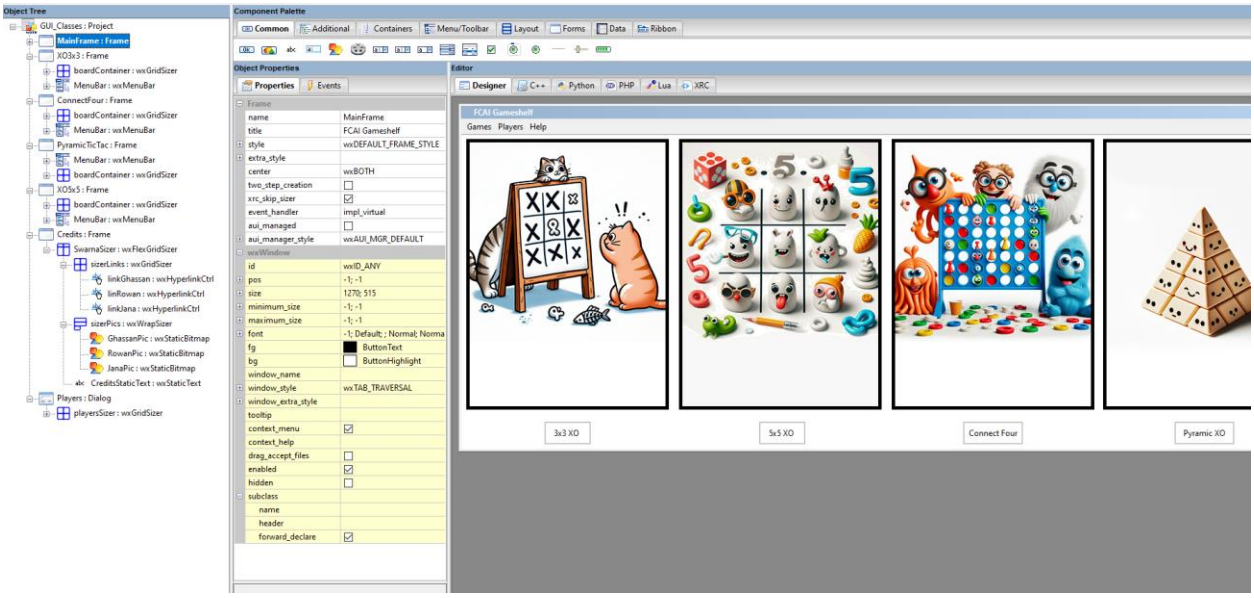
Prepared By:

Name	ID
Ghassan Tarek Elgendy	**20220239**
Jana Mohamed Ramadan	**20220470**
Rawan Ehab Ammar	**20220133**

Pre-phase:

After searching for a good GUI library/framework we decided to use combination between wxWidgets and wxFormBuilder.

In wxFormBuilder we created the initial prototype for the gui and embedded the need buttons , texts and input fields without any functionalities



Then we modified the code in visual studio, and created the needed handlers and functions to run each game and for better usability/compatibility , after developing the main menu and menu bar, each one of the team members took a game to develop it as follows

Implementation phase:

Pyramic XO (Done by Rawan):

After implementing the main frame for the game through the constructor , and creating the gameboard, a function to handle the button press for each cell was created to start each turn and other helper functions.

Constructor (PyramicTicTac::PyramicTicTac):

Initializes a wxFrame (a window)

Initializes a reset button and a grid of buttons for the Tic Tac Toe game.

Binds events such as button clicks and menu selections to their respective handler functions.

Event Handler Functions:

OnInstructions: Displays instructions for the game in a dialog box when the "Instructions" menu item is clicked.

rand_comp_move and **smart_comp_move**: Functions to handle the computer's moves in different difficulty modes.

OnButtonClicked: Handles button clicks on the game grid, allowing players to place their moves and controlling the game flow accordingly.

ResetButton: Resets the game board and enables buttons for a new game when the reset button is clicked.

Helper Functions:

is_winner: Checks if a player has won the game by evaluating the current state of the game board.

is_draw: Determines if the game ends in a draw when there are no more moves possible.

Reset: Resets the gamboard, sets the turn to 0 and enable the buttons back.

Connect 4 (Done by Ghassan):

Constructor (ConnectFour::ConnectFour):

Binds button click events to a function (onCellClick) that handles player moves.

Initializes menu items (like "Instructions") and binds menu events.

Initializes a reset button and binds its click event.

Helper Functions:

reset(): Resets the game board and related variables to start a new game.

ComputerPlay(): Handles the computer's move by generating a random valid move.

AIMove(): Handles AI logic for making moves in the game.

isWinner(): Checks if there's a winner by examining the game board for winning patterns.

isDraw(): Checks if the game ends in a draw when there are no more moves possible.

endGame(): Handles the end of the game, setting appropriate messages and disabling further moves.

Event Handling Functions:

OnInstructions(): Displays game instructions in a dialog box.

OnResetBtn(): Triggers a reset of the game when the reset button is clicked.

onCellClick(): Handles a player's move when a cell on the game grid is clicked.

OnCharEvent(): Listens for keyboard events to enable keyboard input for gameplay (e.g., pressing 'R' for reset).

5x5 XO (Done by Jana):

Constructor: Initializes the game window, buttons, menu items, game status display, and layout.

onCellClick: Handles the button clicks for player moves and switches between 'x' and 'o' based on the current turn.

isWinner: Checks for winning conditions by examining rows, columns, diagonals, and reverse diagonals.

rand_comp_move: Generates a random move for a computer player ('o') when playing against a human player.

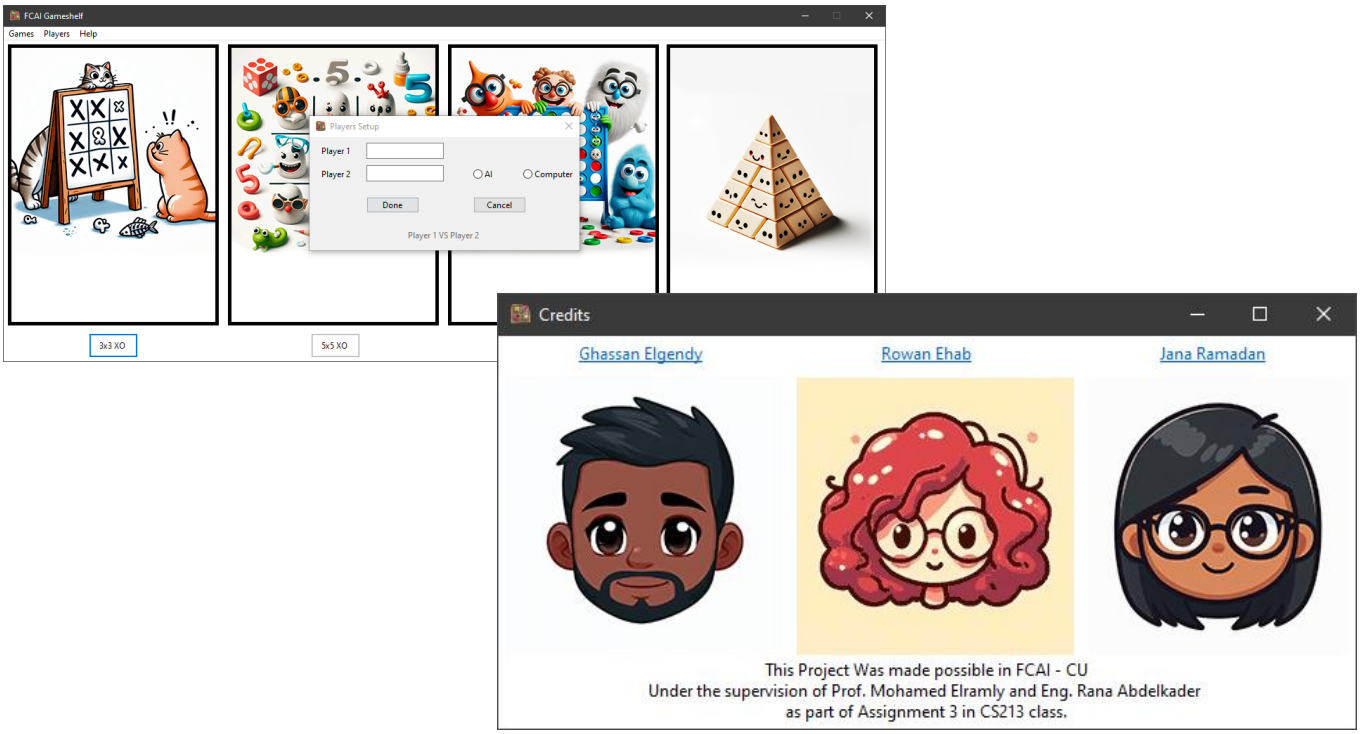
OnInstructions: Displays game instructions in a dialog box.

OnResetBtn: Resets the game board, enabling buttons for a new game, resetting the turn and move count, and resetting the game status display.

Integration phase:

After each one of the team members has finished the game, we integrated all the games in the main menu using button handlers for a button with the game's name, initiating the constructor for the game.

Screenshot of the app:



Proof of Git history: