CS213: Object Oriented Programming

Assignment 3

Faculty of Computers and Artificial Intelligence

Cairo University

CS213: Programming II

Year 2023-2024

First Semester

Assignment 3

Team Members:

Ahmad Yahia Hamada – 20220048

Saif El-Din Shady – 20220160

Ahmad Ismail Ali – 20220010

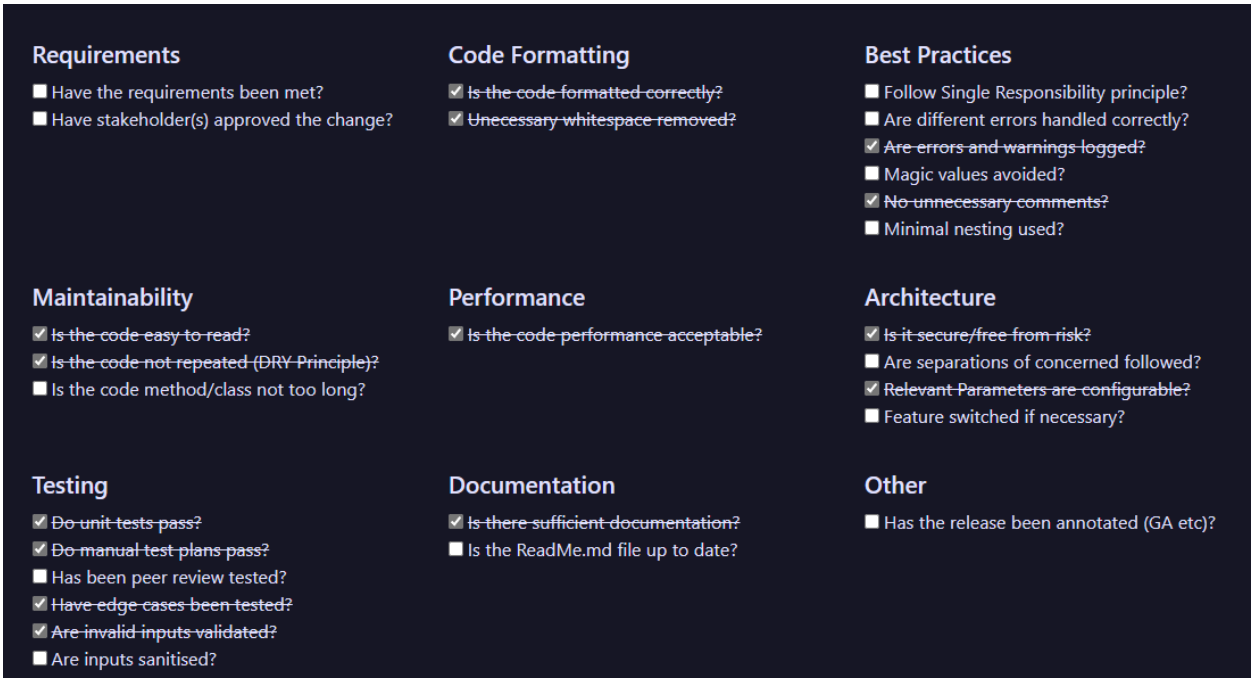
Work Breakdown Table:

Pyramid XO Game	Ahmad Yahia Hamada - 20220048
Connect-4 Game	Saif El-Din Shady - 20220160
Five-by-five XO Game	Ahmad Ismail Ali - 20220010

Quality Report:

The code review has shown the following:

1- The code is well-written, although complex.
2- Some methods are too long. Cases of this include:
   a- The int Board25::wins(char symbol) method on line 189 of the implementation file, titled “full_implementation.cpp”
   b- The PyramBoard::PyramBoard() constructor method on line 69 of the implementation file, titled “full_implementation.cpp”
3- The code’s performance is more than acceptable.
4- The code is formatted correctly using the Prettier extension offered by the Visual Studio Code IDE.
5- The code is not secure from risk due to the incomplete adaptation of the Random Player class, defined on line 67 of the header file, titled “board_Game.hpp”.
6- The code deals safely with errors and validates invalid inputs by asking for them again.



Program Description:

Classes:

- Board class:
    - Houses the data structure which holds the boards of all games offered by the program.
    - Manages the board by variables for the numbers of rows and columns.
    - Holds virtual methods which are to be inherited by children classes to be adapted for each individual game.
- X_O_Board class:
    - 1st child of the previous class.
    - The first game offered by the program: Classical 3x3 Tic-tac-toe.
    - The game stops when one player has formed a three-in-a-row with their symbol, horizontally, vertically, or diagonally.
- PyramBoard class:
    - 2nd child of the Board class.
    - The second game offered by the program: Tic-tac-toe but on a pyramid board.
    - The game board is shaped like a pyramid. Five squares make the base, then three, then one. Players take turns marking Xs and Os as in traditional tic-tac-toe.
- Board42 class:
    - 3rd child of the Board class.
    - The third game offered by the program: Connect 4.
    - Seven columns of six squares each. Instead of dropping counters as in Connect Four, players mark the grid with Xs and Os as in tic-tac-toe.
- Board25 class:
    - 4th child of the Board class.
    - The fourth game offered by the program: 5x5 Tic-tac-toe.
    - Players take turns placing an X or an O in one of the squares until all the squares except one are filled.
    - The player with the higher number of three-in-a-rows wins. Sequences are counted horizontally, vertically, and diagonally. Two sequences overlapping is allowed.

- Player class:
    - Represents one player in the game.
    - Tasked with managing the player's information such as the player's name, their symbol, and the moves they make.
- Random Player class:
    - Child of the previous class.
    - Meant to represent a computer player, which makes random moves.

Methods:

-Board class:

1- Board(): Constructor for the board class.

2-void display_board(): Displays the board, along with each square's index. No side effect.

3-bool is_winner(): Not implemented, defined as virtual for children classes to inherit.

4- bool is_draw(): Same as the above.

5-bool game_is_over(): Same as the above.

6-bool update_board(int x, int y, char symbol): returns true and updates board if the submitted move is valid, otherwise prints an error message and doesn't update board.

-XOBoard class:

1- bool is_winner(): returns true when any player has a three-in-a-row sequence on the board.

2-bool is_draw(): returns true if maximum turns have been played and there is no winner.

3-bool game_is_over(): returns true if maximum turns have been played.

-PyramBoard class:

1-PyramBoard(): Constructor, constructs an empty board of one row of one, another row of three, and a final row of five.

2-bool is_winner(): returns true when any player has a three-in-a-row sequence on the board.

3-bool is_draw(): returns true if maximum turns have been played and there is no winner.

4-bool game_is_over(): returns true if maximum turns have been played.

5-int get_num_columns(int x): used for management due to different number of cells in every row.

-Board42 class:

1-bool is_winner(): returns true when any player has a four-in-a-row sequence on the board.

4-bool is_draw(): returns true if maximum turns have been played and there is no winner.

3-bool game_is_over(): returns true if maximum turns have been played.

5-bool update_board(int x, int y, char symbol): returns true and updates board if the move is valid, otherwise returns false. Different form its parent's counterpart in the conditions of validity.

-Board25 class:

1-int wins(char symbol): Counts the number of three-in-a-rows of a single character on the board.

2-char winner(char c1, char c2): Wrapper for the above, returns the character with more three-in-a-rows.

-Player class:

1-Player(int order, char symbol): Constructor, assigns order and symbol to the player.

2-void get_move(int x, int y): Stores the coordinates of the player's move.

3-string to_string(): Getter for the player's name.

4-char get_symbol(): Getter for the player's symbol.

-RandomPlayer class:

1-RandomPlayer(char symbol, int dd, int ee): Constructor, sets the players name and assigns new symbol.

2-void get_move(int x, int y): Generates random move for the player and stores coordinates in variables x and y.

-GameManager class:

1-GameManager(Board *, Player*playerPtr[2]): Regular constructor, to store a pointer to a board type, or child-of-board type, variable and two pointers to a player type, or child-of-player type, variable.

2-GameManager(Board25, Player*playerPtr[2]): Special constructor for the fourth game option, 5x5 Tic-tac-toe.

3-void run(): Runs the first three games, terminates when game is over and prints winner or draw.

4-void run25(): Runs the 5x5 tic-tac-toe game, terminates when max move count is met.