

Faculty of Computers and Artificial Intelligence  
Object Oriented Programming

## Assignment #3

### Tic-Tac-Toe Versions

Under the supervision of:

**Dr. Mohamed El-Ramly**

**By:**

<b>Name</b>	<b>IDs</b>
Ahmed Ashraf	20230606
Nada Abobaker	20220715
Alaa Eldin Hassan	20230615

## Work table

Ahmed Ashraf	Nada Abobaker	Alaa Eldin
Game #2	Game #1	Game #3
The integrated app	-----	Report

# **The code analysis report**

## **Introduction:**

This project comprises implementations of four distinct board games: XO (3\*3 Tic-Tac-Toe), Pyramic Tic-Tac-Toe, Four in a Row, and Five by Five Tic-Tac-Toe. Each game is encapsulated within a set of classes, including board representations, player types, and game managers.

## **Background:**

The code aims to create playable instances of popular board games, incorporating human and computer players. Each game adheres to its specific rules and winning conditions, utilizing classes to manage game flow, player moves, and board updates. The structure suggests a modular approach, fostering code reusability and maintainability.

## **Process:**

### **1. XO Game:**

- **GameManager class:**
  - Manages the flow of the game, alternating player turns.
  - Checks for a winner or a draw after each move.
- **RandomPlayer class:**
  - Generates random moves for the computer player.

### **2. Pyramic Tic-Tac-Toe:**

- **Tic\_Tac\_Toe\_board class:**
  - Represents a triangular board structure.
  - Validates moves based on the triangular pattern.
  - Checks for a winner using specific triangular conditions.

### **3. Four-in-a-row:**

- **FourIAR\_board class:**
  - Represents a 6x7 board for Four-in-a-row.
  - Updates the board, checks for a winner, and manages moves.

- Uses an array to track used squares in each column.

#### 4. **Five by Five Tic Tac Toe:**

- **FiveBF\_board class:**
  - Represents a 5x5 Tic Tac Toe board.
  - Validates moves and checks for a winner using specific patterns.
  - Determines the winner based on the number of moves.

### **Issue Resolution:**

Noteworthy issues include the complexity of winner-checking logic and the potential for inaccurate results in determining winners based on move count. Resolving these concerns would involve refactoring common functionalities, enhancing code readability through comments, and thorough testing to validate the correctness of game logic.

### **User Experience:**

The user experience is influenced by aspects such as error handling, user interface clarity, and feedback during gameplay. Suggestions include refining user prompts, implementing robust error handling for user input, and enhancing the overall user interface for a more enjoyable gaming experience.

### **Effectiveness:**

Effectiveness is assessed based on the code's ability to fulfill its intended purpose – facilitating the gameplay of various board games. Opportunities for improvement include code consistency, better organization, and the adoption of best practices to enhance the maintainability and extensibility of the codebase.

### **Suggestions:**

#### **1. Refactoring:**

- Explore opportunities for code reuse, especially in winner-checking logic.
- Consider creating a common base class for shared functionalities.

#### **2. User Interface Enhancement:**

- Refine user prompts for clearer instructions.
- Provide informative feedback during gameplay.

#### **3. Error Handling:**

- Implement robust error handling for user inputs to enhance code reliability.

#### 4. Code Consistency:

- Ensure consistent coding style and formatting across different classes.

### **Conclusion:**

The code exhibits effective management of various board games, with potential areas for improvement in terms of code organization, user experience, and code consistency. Addressing these aspects will contribute to a more refined and maintainable codebase, ensuring an optimal gaming experience. Thorough testing is advised to validate the correctness of the implemented game logic.

# The code quality of the three games

## The code checklist

### Requirements

- ☐ Have the requirements been met?
- ☐ Have stakeholder(s) approved the change?

### Code Formatting

- ☐ Is the code formatted correctly?
- ☐ Unnecessary whitespace removed?

### Best Practices

- ☐ Follow Single Responsibility principle?
- ☐ Are different errors handled correctly?
- ☐ Are errors and warnings logged?
- ☐ Magic values avoided?
- ☐ No unnecessary comments?
- ☐ Minimal nesting used?

### Maintainability

- ☐ Is the code easy to read?
- ☐ Is the code not repeated (DRY Principle)?
- ☐ Is the code method/class not too long?

### Performance

- ☐ Is the code performance acceptable?

### Architecture

- ☐ Is it secure/free from risk?
- ☐ Are separations of concerned followed?
- ☐ Relevant Parameters are configurable?
- ☐ Feature switched if necessary?

### Testing

- ☐ Do unit tests pass?
- ☐ Do manual test plans pass?
- ☐ Has been peer review tested?
- ☐ Have edge cases been tested?
- ☐ Are invalid inputs validated?
- ☐ Are inputs sanitised?

### Documentation

- ☐ Is there sufficient documentation?
- ☐ Is the ReadMe.md file up to date?

### Other

- ☐ Has the release been annotated (GA etc)?

## Game #1:

### Requirements

- ☒ Have the requirements been met?
- ☐ Have stakeholder(s) approved the change?

### Code Formatting

- ☒ Is the code formatted correctly?
- ☒ Unnecessary whitespace removed?

### Best Practices

- ☒ Follow Single Responsibility principle?
- ☒ Are different errors handled correctly?
- ☐ Are errors and warnings logged?
- ☒ Magic values avoided?
- ☒ No unnecessary comments?
- ☒ Minimal nesting used?

### Maintainability

- ☒ Is the code easy to read?
- ☒ Is the code not repeated (DRY Principle)?
- ☒ Is the code method/class not too long?

### Performance

- ☒ Is the code performance acceptable?

### Architecture

- ☒ Is it secure/free from risk?
- ☒ Are separations of concerned followed?
- ☐ Relevant Parameters are configurable?
- ☐ Feature switched if necessary?

### Testing

- ☐ Do unit tests pass?
- ☐ Do manual test plans pass?
- ☐ Has been peer review tested?
- ☐ Have edge cases been tested?
- ☒ Are invalid inputs validated?
- ☐ Are inputs sanitised?

### Documentation

- ☒ Is there sufficient documentation?
- ☐ Is the ReadMe.md file up to date?

### Other

- ☐ Has the release been annotated (GA etc)?

## Game #2:

### Requirements

- ☒ Have the requirements been met?
- ☐ Have stakeholder(s) approved the change?

### Maintainability

- ☒ Is the code easy to read?
- ☒ Is the code not repeated (DRY-Principle)?
- ☒ Is the code method/class not too long?

### Testing

- ☐ Do unit tests pass?
- ☐ Do manual test plans pass?
- ☐ Has been peer review tested?
- ☐ Have edge cases been tested?
- ☒ Are invalid inputs validated?
- ☐ Are inputs sanitised?

### Code Formatting

- ☒ Is the code formatted correctly?
- ☒ Unnecessary whitespace removed?

### Performance

- ☒ Is the code performance acceptable?

### Documentation

- ☒ Is there sufficient documentation?
- ☐ Is the ReadMe.md file up to date?

### Best Practices

- ☒ Follow Single-Responsibility principle?
- ☒ Are different errors handled correctly?
- ☐ Are errors and warnings logged?
- ☒ Magic values avoided?
- ☒ No unnecessary comments?
- ☒ Minimal nesting used?

### Architecture

- ☒ Is it secure/free from risk?
- ☒ Are separations of concerned followed?
- ☐ Relevant Parameters are configurable?
- ☐ Feature switched if necessary?

### Other

- ☐ Has the release been annotated (GA etc)?

## Game #3:

### Requirements

- ☒ Have the requirements been met?
- ☐ Have stakeholder(s) approved the change?

### Maintainability

- ☒ Is the code easy to read?
- ☒ Is the code not repeated (DRY-Principle)?
- ☐ Is the code method/class not too long?

### Testing

- ☐ Do unit tests pass?
- ☐ Do manual test plans pass?
- ☐ Has been peer review tested?
- ☐ Have edge cases been tested?
- ☒ Are invalid inputs validated?
- ☐ Are inputs sanitised?

### Code Formatting

- ☒ Is the code formatted correctly?
- ☒ Unnecessary whitespace removed?

### Performance

- ☒ Is the code performance acceptable?

### Documentation

- ☒ Is there sufficient documentation?
- ☐ Is the ReadMe.md file up to date?

### Best Practices

- ☒ Follow Single-Responsibility principle?
- ☒ Are different errors handled correctly?
- ☐ Are errors and warnings logged?
- ☒ Magic values avoided?
- ☒ No unnecessary comments?
- ☒ Minimal nesting used?


### Architecture





- ☒ Is it secure/free from risk?
- ☒ Are separations of concerned followed?
- ☐ Relevant Parameters are configurable?
- ☐ Feature switched if necessary?

### Other





- ☐ Has the release been annotated (GA etc)?

# Github

 **OOP-Assignments** Public


 Pin  Unwatch 1  Fork 0  Star 0

## Activity


 All branches  All activity  All users  All time

Showing most recent first


**Tic\_Tac\_Toe\_board.cpp** ...

 iindll pushed 1 commit to `main` • f45d7f4...1fffdce • 1 hour ago


**X\_O\_Board.cpp** ...

 iindll pushed 1 commit to `main` • b74abb9...f45d7f4 • 1 hour ago


**Create TicTacToeGameManager.cpp** ...

 iindll pushed 1 commit to `main` • 380bc6d...b74abb9 • 1 hour ago


**Last update Game #3: 5\*5 Tic Tac Toe** ...

 3laa106 pushed 1 commit to `main` • 63d38bd...380bc6d • 2 hours ago


**X\_O.cpp** ...

 iindll pushed 1 commit to `main` • e5563a2...63d38bd • 2 hours ago


**Tic\_Tac\_Toe.cpp** ...

 iindll pushed 1 commit to `main` • e52325d...e5563a2 • 2 hours ago


**Update BoardGame\_Classes.hpp** ...

 lamoro8800 pushed 1 commit to `main` • f803210...e52325d • 3 hours ago


**Create RandomPlayer.cpp** ...

 lamoro8800 pushed 1 commit to `main` • 79fc929...f803210 • 5 hours ago


**Create Player.cpp** ...

 lamoro8800 pushed 1 commit to `main` • b71da1d...79fc929 • 5 hours ago


**Create GameManager.cpp** ...

 lamoro8800 pushed 1 commit to `main` • 533dbbd...b71da1d • 5 hours ago

**Create FourIAR\_RandomPlayer.cpp** ...

 lamoro8800 pushed 1 commit to `main` • d5a7c7b...533dbbd • 5 hours ago

**Create FourIAR\_Player.cpp** ...

 lamoro8800 pushed 1 commit to `main` • eded38b...d5a7c7b • 5 hours ago