|  |  |
| --- | --- |
| **C:\Users\Far\Pictures\New Picture.jpg** | **Code Review Template** |

**This template is to be completed and submitted by the *Reviewer*.**

Names of the Reviewers: Ziad Chemali

Name of the developer being reviewed: Lotfi Hasni

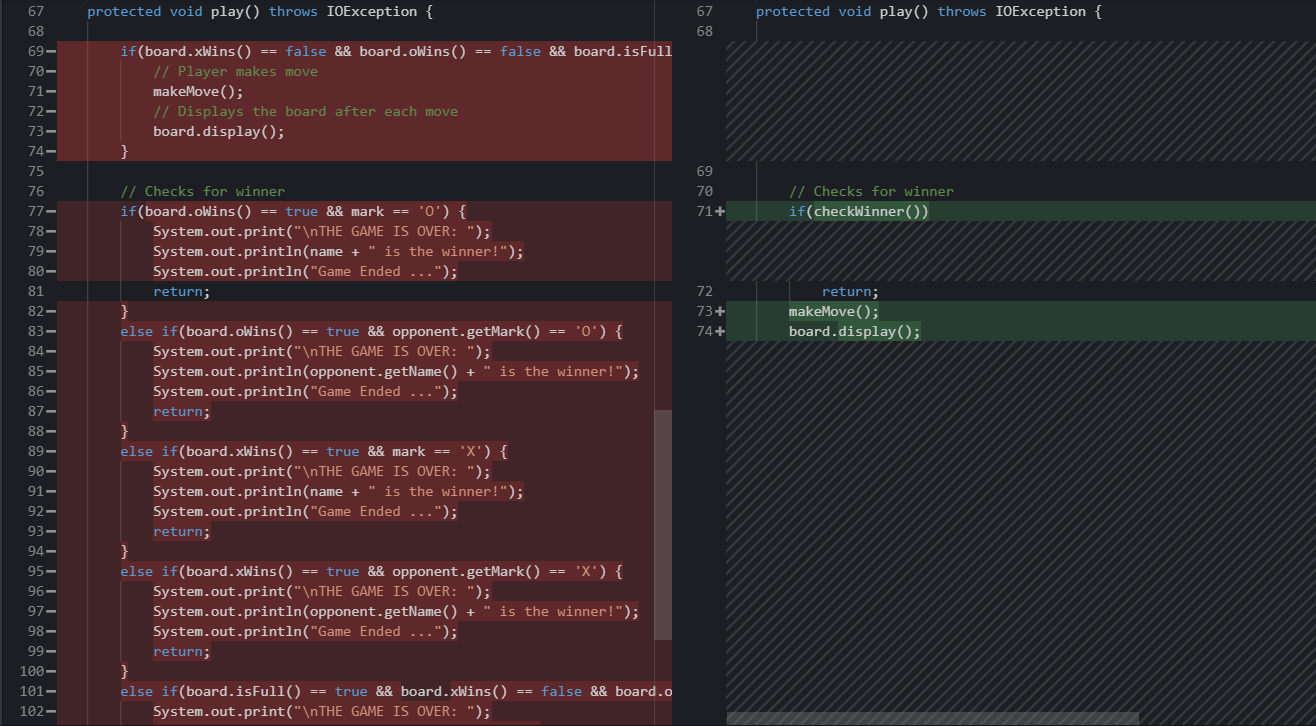
|  |  |  |
| --- | --- | --- |
| **Category** | **Comments /questions about of the reviewing group about the design documents** | **Responses by the developer (if any)** |
| Spelling Mistakes | N/A |  |
| Naming issues | N/A |  |
| SOLID Principle Violations | N/A |  |
| Lack of documentation | N/A |  |
| Conditional statements | Some if statements conditions can be re-written, ex: instead of if(board.xWins()==false) can be rewritten as if(!board.xWins()) |  |
| OOP | RandomPlayer has dependency to RandomGenerator instead make the relationship between these two classes aggregation or composition, to avoid creating new RandomGenerator objects each time makeMove is called |  |
| Method structure | Divide the tasks for the method, example makeMove() can be divided in to another 3 methods, getRow(),getCol(), and addMark(). This way it is easier to debug code |  |
| Base Class Methods | Since the logic of checking for winner is the same among all sub classes of Player it makes sense to make a method in Playerclass that checks for winner , then each sub class can invoke it. |  |

# Changes:

## i)HumanPlayer.java:

### 1)play method:

Note: CheckWinner() is a new method added to Player class, that contains all the logic for checking for winner



### 2)makeMove()

|  |  |
| --- | --- |
| Before | After |
|  |  |

## ii)RandomPlayer.java:

