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

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

Names of the Reviewer: Parker Link (30045201)

Name of the developer being reviewed: William Ledingham (<https://github.com/William-Ledingham/ENSF409_Lab4/tree/master/Ex3_TicTacToe/src>)

|  |  |  |
| --- | --- | --- |
| **Category** | **Comments/questions about of the reviewing group about the design documents** | **Responses by the developer (if any)** |
| Spelling Mistakes | * No observable spelling mistakes were present. |  |
| Naming Issues | * The method name testForBlocking of BlockingPlayer.java is not inherently clear what it is. * The methods addHyphens/addSpaces of Board.java should be called print\_\_\_. Inconsistent naming between displayColumnHeaders and addHyphens. * The create\_player() method in Game.java uses an underscore instead of camelCase. |  |
| SOLID Principle Violations | * Single Responsibility: All classes effectively follow a single responsibility. Better implementation of this could be employed by following the refactoring notes below, though. * Open/Close Principle: Not directly applicable here, but the general guidelines are followed by allowing configurability with the constants defined in Constants.java. * Interface Segregation: The one interface used (Player.java) was implemented effectively, as it allows the developer to only implement the methods they need. By following the refactoring notes, further minor improvements can be made. * Dependency Inversion Principle: The higher-level player does not depend on lower-level modules, but rather depends on its children. |  |
| Lack of documentation | * The checkWinner method of Board.java contains no inline comments. * Throughout all the code, very few inline comments (//) are used, making the intention of various sections quite difficult. |  |
| Consideration of “Overly-Complicated” code | * The checkWinner method of Board.java could be implemented in a more clear way, potentially breaking its functionality into subroutines/methods. * The create\_player method of Game.java is quite long, and could be broken into separate methods, with more methods used for human output. * Having a SPACE\_CHAR constant is highly unnecessary, considering it wasn’t even used in locations such as the addSpaces() method of Board.java. * Excellent use of javadocs overall, especially for member variables/attributes/fields. |  |
| Refactoring Suggestions (Avoid Code Reuse) | * testForWinning() of SmartPlayer.java contains redundant code from checkWinner of Board.java and testForBlocking of BlockingPlayer.java. Better inline documentation would help making moving this code into some shared methods, possibly placed in the Player superclass. * RandomGenerator class should be developed as a static math library-style, instead of requiring object instantiation. |  |