Code Review Template

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

Names of the Reviewers: **Nathan Jack**

Name of the developer being reviewed: **Jordan Joorisity**

Category	Comments /questions about of the reviewing group about the design documents	Responses by the developer (if any)
Spelling Mistakes	• None apparent.	
Naming issues	• All vars named appropriately.	
SOLID Principle Violations	• HumanPlayer.Java makeMove()method handles user input/screenoutput/input checking/move validation/and making the move (or looping as needed) this could be 2-3 separate methods. • Blocking Player testPlay() = 70 lines of code. Splitting them up into checkDiag/CheckHoriz/CheckVert might make that chunk easier to handle.	Agreed, there is also some violation of the DRY principle here as the testPlay() method is related to the checkWin() in the Board class.
Lack of documentation	• Player.Java play() and makeMove() method have IO Exceptions but no explanation of what would cause the exception. Exception not explicitly handled. Exception without documentation present in all player classes.	Agreed, error handling needs improvement throughout.
DRY	• Recreating a random generator for each player in game() seems repetitive. RandomPlayers on creation should create their own generator, perhaps within the constructor. Meaning all subclasses would inherit the ability to	Agreed, this is repetitive but also allows control over coupling, as the random generator could be passed with association/aggregation over

	create a generator. Making the random generator as an interface might also alleviate this problem. ● Blocking player makeMove() rewrites the logic for randomPlayer makeMove(). Better to use super.makeMove() if the test for blocking fails. ●	composition. Using an interface is the best option. I should have created a randomMove method that could be inherited from the Random Player. Having the strategies intermingled is sloppy.