public class Game- the game class will include the project's main method and serve as a tester class. This being said it will contain no methods of its own, only manipulation of the other classes in the project to produce the ending user interface. The main method and in turn the Game class will display the board, as well as the winning or losing message depending on the game’s outcome.

public class Outcome- outcome will include the methods isWinner() which will determine the winning and losing conditions and when to display a winning or losing message via the Game class. The Outcome class will also determine when the game is over and thus when to check for a win or loss. An endgame() method will be used to count the number of pegs and will collaborate with peg and board to see if any acceptable moves are left. When either one peg is left or no more acceptable moves can be made the isWinner()method will count the number of pegs given by the board class to determine whether the outcome is a win or a loss. If one peg is left on the board then the game outcome is a win, if the pegs are two or greater after the game is over then the match is a loss and will tell the tester to display a varying message that depending on the pegs left will display an increasingly degrading message for each additional peg left.

public boolean endgame()-calls board and loops through the array calling canMove() for each peg in the array, if any pegs return canMove() as true then there are still acceptable moves. If no peg returns canMove() as true then there are no acceptable moves, if no acceptable moves are on the board then return true, if there are acceptable moves left on the board then return false.

public boolean isWinner()-if endgame returns true, calls board.getPegs() to get the number of pegs left on the board, if(pegs == 1) then return true, else return false.