

CS 5010 Fall 2021 Section 8 (Vancouver)

(/courses/152)

Fall 2021 Section 19263 : Coria Mendoza at M 6:30 PM - 8:30 PM / W 4:00 PM - 6:00 PM (PT) (Lecture)

You are a **student** of this course.

[Teams \(/courses/152/teamsets\)](/courses/152/teamsets)[Assignments \(/courses/152/assignments\)](/courses/152/assignments)

This will be an individual response.

▼ src/

▼ tictactoe/

Main.java

Player.java

TicTacToe.java

TicTacToeConsoleController.java

```
1 package tictactoe;
2
3 import java.io.InputStreamReader;
4
5 /**
6  * Run a TicTacToe game interactively on the console.
7  */
8 public class Main {
9     /**
10      * Run a TicTacToe game interactively on the console.
11      */
12     public static void main(String[] args) {
13         Readable input = new InputStreamReader(System.in);
14         Appendable output = System.out;
15         new TicTacToeConsoleController(input, output).playGame(new TicTacToeMc
16     }
17 }
18
```

Implementation

1. Did you add any additional fields to the TicTacToeConsoleController class? (1 point)

Tag one of the additional field declarations

File: Line number:

2. Did you implement any additional methods in your TicTacToeConsoleController? (2 points)

Tag one of these additional methods, if it exists

File: Line number:

Tag another of these additional methods, if it exists

File: Line number:

Tag a third one of these additional methods, if it exists

File: Line number:

3. Explain how you handled invalid input in the controller. (E.g., user entering "two" as a row or column number for a move.) (5 points)

Tag 1 - explain how this tag supports your response

File: Line number:

Tag 2 - explain how this tag supports your response

File: Line number:

Tag 3 - explain how this tag supports your response

File: Line number:

4. Explain how you handled an invalid row or column number entered by the user (e.g., 5) (5 points)

Tag 1 - explain how this tag supports your response

File: Line number:

Tag 2 - explain how this tag supports your response

File: Line number:

Tag 3 - explain how this tag supports your response

File: Line number:

Testing

5. Do you have at least one test to validate that the Controller throws an `IllegalArgumentException` if the model passed to `playGame` is invalid? (1 point)

Yes

No

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

6. Do you have at least one test to validate that the Controller works correctly when given invalid input for a row? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

7. Do you have at least one test to validate that the Controller works correctly when given invalid input for a column? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

8. Do you have at least one test to validate that the Controller works correctly when given a row that is out of bounds? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

9. Do you have at least one test to validate that the Controller works correctly when given a column that is out of bounds? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

10. Do you have at least one test to validate that the Controller works correctly when a "q" is given for the row? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

11. Do you have at least one test to validate that the Controller works correctly when a "q" is given for the column? (1 point)

Tag the first line of the relevant JUnit assertion

File: -- select a file -- Line number:

Optional explanation

12. Do you have at least one test to validate that the Controller works correctly when given a valid move?
(1 point)

Tag the first line of the relevant JUnit assertion

File: -- select a file -- Line number:

Optional explanation

13. Do you have at least one test to validate that the Controller works correctly when given an move to a space that is already occupied? (1 point)

Tag the first line of the relevant JUnit assertion

File: -- select a file -- Line number:

Optional explanation

14. Do you have at least one test to validate that the Controller continues to run after an invalid move is given? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

15. Do you have at least one test to validate that the Controller works correctly when the game ends with player X winning? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

16. Do you have at least one test to validate that the Controller works correctly when the game ends with player O winning? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

17. Do you have at least one test to validate that the Controller works correctly when the game ends in a tie? (1 point)

Tag the first line of the relevant JUnit assertion

File: Line number:

Optional explanation

Student notes

Bottlenose copyright © 2012-2017 Benjamin Lerner, Nat Tuck. Licensed under the GNU Affero GPL (/agpl-3.0.txt) v3 or later. Source at GitHub (<http://www.github.com/CodeGrade/bottlenose>).