

**Due** Nov 26, 2022 by 11:59pm**Points** 2**Available** after Nov 21, 2022 at 10:30am

## Controller and its Uses

[Home \(https://northeastern.instructure.com/courses/123246/pages/home\)](https://northeastern.instructure.com/courses/123246/pages/home)[Modules \(https://northeastern.instructure.com/courses/123246/modules\)](https://northeastern.instructure.com/courses/123246/modules)

(https://northeastern.instructure.com/courses/123246/assignments/1509845?module\_item\_id=7739424)

### Canvas Exam Security Enhancements



Canvas has enhanced their Files tool to ensure files uploaded to an assignment [default to hidden status](#)

<https://community.canvaslms.com/t5/Deploys/Canvas-Deploy-Notes-2020-11-18/ta-p/421112#toc-hld-821974231>. This helps to ensure

files uploaded to exams and quizzes are secure and only viewable when the assignment or quiz is made available.

**Instructors who were setting release dates in the files area no longer need to do this**, though they will need to upload the

Don't show this again

Helpful?  

#### 1.1 A Controller

The purpose of this exercise is to implement the Model-View-Controller pattern.

In the starter code (at the end of the lab), you will find the starter code for **Tic Tac Toe**, with

**TicTacToeController** interface. Put your new controller code in the same package alongside your Tic Tac Toe model from the previous exercise as it will depend on the model. You are also given a class **Main** with a **main()** method that will allow you to test your game interactively.

You will need to create one additional class: a public class named

**TicTacToeConsoleController** that implements **TicTacToeController**, with a single public

constructor that takes two arguments, a **Readable** and an **Appendable** (in that order). You will fill in the fields and the method definitions as appropriate. You may also define other classes at your option as needed.

The controller will output game state and prompts to the **Appendable**, and read inputs from the **Readable** corresponding to user moves. The `append()` method on **Appendable** throws a checked exception, **IOException**. Your `playGame()` method should not throw this exception. If it occurs, your `playGame()` should catch it and throw an **IllegalStateException**.

A single move consists of two numbers, specifying the row and column of the intended move position. Board positions for these moves are numbered from 1. For example, to mark **X** in the upper left cell, the user would enter `"1 1"` at the first prompt. To mark **O** in the upper right cell on the second move, the user would enter `"1 3"`. To quit a game in progress, the user can enter **q** or **Q** at any time.

The game state is the output of the model's `toString()` method, followed by a carriage return (`\n`). The move prompt is

```
"Enter a move for " + model.getTurn().toString() + ":\n"
```

(where `model` is an instance of your Tic Tac Toe Model).

If a non-integer value is entered, it should be rejected with an error message. If an invalid move is entered, namely, two valid integers, but the proposed move was deemed invalid by the model, the controller should give an error message. The message text is up to you, but should end with a carriage return.

At the end of the game

- A final game state
- `"Game is over"` the outcome

If the user quits, the controller

```
"Game quit! Ending"
```

and end the `playGame()`

## Canvas Exam Security Enhancements

Canvas has enhanced their Files tool to ensure files uploaded to an assignment [default to hidden status](#)

(<https://community.canvaslms.com/t5/Deploys/Canvas-Deploy-Notes-2020-11-18/ta-p/421112#toc-hld-821974231>). This helps to ensure

files uploaded to exams and quizzes are secure and only viewable when the assignment or quiz is made available.

**Instructors who were setting release dates in the files area no longer need to do this**, though they will need to upload the

Don't show this again

Helpful?  

## 1.2 Testing

We have supplied you with some basic JUnit tests as part of the starter code. Use these to verify that your implementation is correct. Write additional tests of your own: Some of the additional cases you should consider are listed as comments in the test class you are given.

## 1.3 Notes to Keep in Mind

- You will likely need a `while()` loop; be aware that you can use the break statement to `break` out of a loop prematurely (before the loop condition is `false`).
- You will need to use the built-in `Scanner` class. See the lecture notes examples and explore the Oracle JavaDoc for this class for more information on how to use it.
- Avoid duplicating code as much as possible. Consider using non-public methods as means of creating reusable pieces of functionality.
- Be sure to use access modifiers, `private` and `public`, as well as `final`, appropriately.
- Include JavaDoc for your classes and constructors as appropriate. You do not need to repeat JavaDoc already existing in a superclass or interface when you override a method.

## 1.4 To Turn In

Submit your solution to Gradescope as usual. Your grade will be determined by code style and JUnit correctness. We will not be doing manual grading, but if you would like feedback on your design, and specifically your test coverage, ask a member of the TA team to review it during office hours.

### Lab 6 - rubric

Full Credit - Pass
2 pts
Both solutions pass automated tests.

### Canvas Exam Security Enhancements

Canvas has enhanced their Files tool to ensure files uploaded to an assignment [default to hidden status](https://community.canvaslms.com/t5/Deploys/Canvas-Deploy-Notes-2020-11-18/ta-p/421112#toc-hld-821974231) (<https://community.canvaslms.com/t5/Deploys/Canvas-Deploy-Notes-2020-11-18/ta-p/421112#toc-hld-821974231>). This helps to ensure files uploaded to exams and quizzes are secure and only viewable when the assignment or quiz is made available.

**Instructors who were setting release dates in the files area no longer need to do this**, though they will need to upload the

[Main.java \(https://northeastern.instructure.com/courses/123246/files/16688237/download?download\\_frd=1\)](https://northeastern.instructure.com/courses/123246/files/16688237/download?download_frd=1) ↓

[TicTacToeController.java \(https://northeastern.instructure.com/courses/123246/files/16688252?wrap=1\)](https://northeastern.instructure.com/courses/123246/files/16688252?wrap=1) ↓

[TicTacToeController.java \(https://northeastern.instructure.com/courses/123246/files/16688252/download?download\\_frd=1\)](https://northeastern.instructure.com/courses/123246/files/16688252/download?download_frd=1) ↓

Don't show this again

Helpful?



[download\\_frd=1\)](#)      [TicTacToeControllerTest.java](#)

[\(https://northeastern.instructure.com/courses/123246/files/16688250?wrap=1\)](#)    ↓

[\(https://northeastern.instructure.com/courses/123246/files/16688250/download?](#)

[download\\_frd=1\)](#)      [FailingAppendable.java](#)

[\(https://northeastern.instructure.com/courses/123246/files/16688251?wrap=1\)](#)    ↓

[\(https://northeastern.instructure.com/courses/123246/files/16688251/download?](#)

[download\\_frd=1\)](#)

## Canvas Exam Security Enhancements



Canvas has enhanced their Files tool to ensure files uploaded to an assignment [default to hidden status](#)

<https://community.canvaslms.com/t5/Deploys/Canvas-Deploy-Notes-2020-11-18/ta-p/421112#toc-hId-821974231>). This helps to ensure

files uploaded to exams and quizzes are secure and only viewable when the assignment or quiz is made available.

**Instructors who were setting release dates in the files area no longer need to do this**, though they will need to upload the

Don't show this again

Helpful?

