CS5004 Lab07 Report

Rong Huang

Lab 7: Tic Tac Toe MVC

1. Reflection

1) MVC Design Pattern:

Implementing the MVC pattern allowed me to segregate the game's logic (TicTacToeModel), user interface (TicTacToeView), and the game flow control (TicTacToeConsoleController). It became clear how each component serves its unique role, making the code more maintainable and scalable.

2) IO Streams & StringBuilder:

Utilizing StringBuilder in the view and Scanner for input in the controller improved my understanding of efficient string manipulation and streamlined input handling in Java.

3) Appendable and Readable:

The FailingAppendable class was a good exercise in understanding that Appendable isn't just for System.out. It could be any writable data destination, which could fail, hence the need to handle exceptions.

4) Exception Handling:

Dealing with IllegalArgumentException and IllegalStateException in the TicTacToeModel gave me a clearer understanding of when and why to throw exceptions. For instance, preventing illegal moves reinforced the integrity of the game state.

2. How does this design encourage future growth?

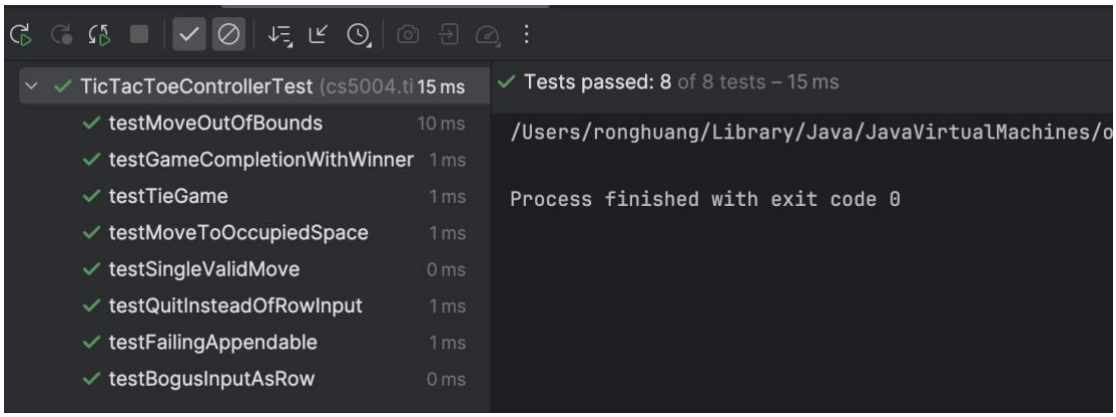
The MVC design inherently supports scalability and future growth. With the model handling the game's logic, the view managing the display, and the controller acting as an intermediary, each component can be modified independently. For instance, the view can be replaced with a graphical user interface (GUI) without altering the game's underlying logic. Similarly, the model can be enhanced to support different game rules while keeping the user interface consistent.

3. Extensions

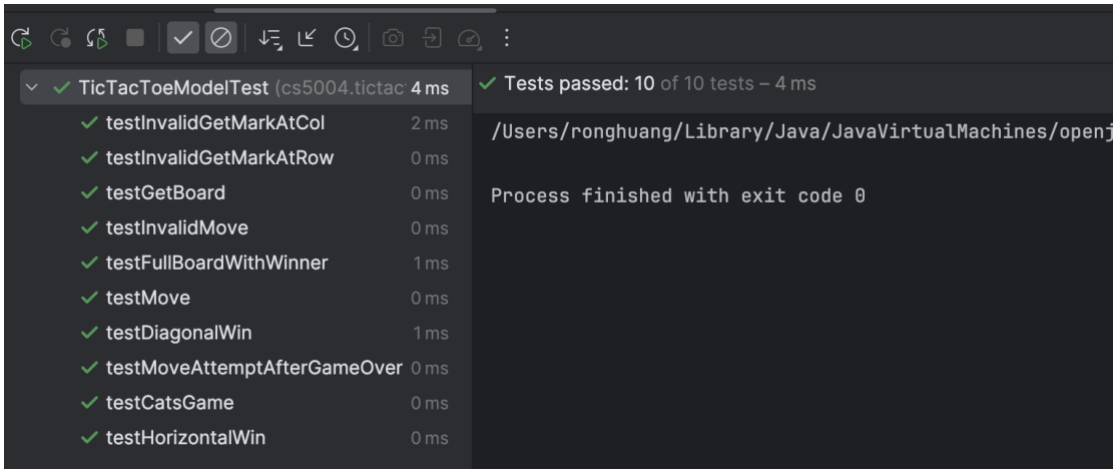
1) add additional tests

I added four additional tests to TicTacToeConsoleController according to the teacher's comments, including:
testGameCompletionWithWinner()

testQuitInsteadOfRowInput()
testMoveOutOfBounds()
testMoveToOccupiedSpace()

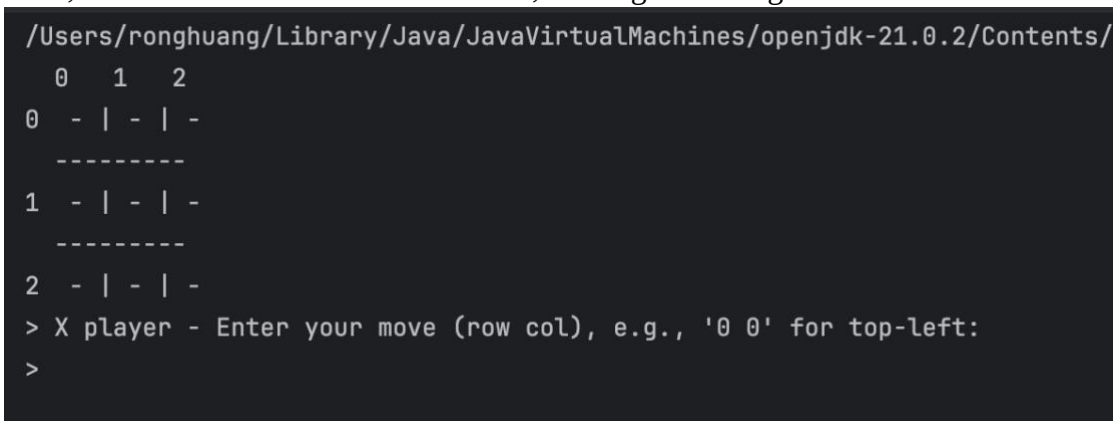


I also added the test in TicTacToeModelTest testFullBoardWithWinner() to test test case where board is full AND there is a winner



2) Creating a driver to test your implementation

I created a main to run the Tic Tac Toe game on the console, which initializes the Model, View, and Controller, and starts the game. You can play the game through this main, and I have tested various situations, The logic of this game is correct



3) Go above and beyond with documentation.

Concise Javadoc English comments are added to each method to improve the readability of the code.

4. Grading Statement

Add the extensions, total 100.

5. Academic Integrity Statement

I understand that my learning is dependent on individual effort and struggle, and I acknowledge that this assignment is a 100% original work and that I received no other assistance other than what is listed here.

Acknowledgements and assistance received:
Professor Molly, TA Will
Google, StackOverFlow

I did not use generative AI in any form to create this content and the final content was not adapted from generative AI created content.

I did not view content from any one else's submission including submissions from previous semesters nor am I submitting someone else's previous work in part or in whole.

I am the only creator for this content. All sections are my work and no one else's with the exception being any starter content provided by the instructor. If asked to explain any part of this content, I will be able to.

By putting your name and date here you acknowledge that all of the above is true and you acknowledge that lying on this form is a violation of academic integrity and will result in no credit on this assignment and possible further repercussions as determined by the Khoury Academic Integrity Committee.

Name:Rong Huang	Date: 04/12/2024