Assignment 4 C212 FA22

Last modified 2022-09-28 11:59 PM

Learning Objectives

After completing this assignment, students will be proficient in utilizing 2D Arrays, ArrayLists, and the logic used to manipulate them to solve complex problems.

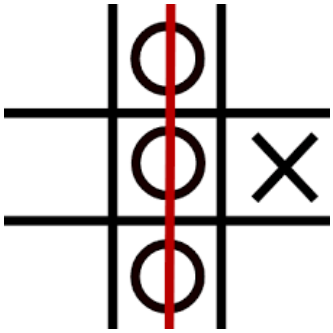
Summary

The gist of this assignment document's contents are as follows:

- Program Description: Tic-Tac-Toe with 1 or 2 players and replay viewing
- Hints, edge-cases, and design considerations
    - You get starter code this time… here's how to use it (or not)
    - You must use 2D arrays and ArrayLists
    - Why do we have static variables/methods?
- Deliverables & Reminders
    - Use a project named `Assignment4`, and a class named `TicTacToe`, preferably by starting with the starter code project folder.
    - Submit a .zip file that is your compressed project folder.
- Rubric
    - Also visible on Canvas assignment page
- Sample program output
- Changelog

Program Description

Your little brother achieved some amazing results thanks to your program. So you decide to reward him by making a game for him. In particular, he loves to play tic-tac-toe with you so you decide to build that.



Furthermore, you want him to be able to compete against the computer as well as you. Hence, your game is going to have two modes: (i) single player mode (against the computer) and (ii) two player mode.

In two player mode, your program should first ask the players their names and toss a coin to decide who goes first. Then the program should ask the users to take turns playing the game. Ensure that each move is legitimate, and print the game result when someone wins or when the game ends in a draw.

In one player mode, your program should first ask for the difficulty as well as the player's name. There can be 3 modes: easy, medium, or hard. Then it should toss a coin to decide who gets to go first.

In easy mode, the computer just randomly decides a row and column (really dumb) and keeps doing that until a valid move is found. In hard mode, your program should be good enough so that the human never wins; instead, the game should end in a "draw" or "the computer wins." Here is the algorithm you should implement for the hard AI:

1. Check if there is a winning move available, and if yes, make that move and win.
2. If not, check if there is a winning move available for the opponent, and if yes, take that spot on the board.
3. If not, then check to see if the center spot is empty. If yes, take that spot on the board.
4. If not, then see if there are any open corners on the board. If yes, take the first open corner found on the board.
5. If not, then find any remaining empty spot on the board and take that spot.

Medium should be in between these two. It should not be dumb but not as good as the hard level. It's up to you how you want to implement the AI for this difficulty as long as you follow this guideline.

There should be a main menu option to display the last match. As you'll see in the starter code, this can be achieved by storing each turn of the game in an ArrayList.

Player one and player two refer to the order in which the players enter their names. Player one gets the 'X' character while player two gets the 'O' character. Player one does NOT refer to the player who takes the first turn, since it's based on a coin-flip. When playing one-player mode, the AI will always be player two.

Lastly, the menu should repeat unless the user selects the quit option.

Hints, edge-cases, and design considerations

This is a challenging assignment with many moving parts! It would be a great idea to start early to leave ample time to visit office hours and get help when you're stuck.

To get you started, this week we provide starter code in the form of an IntelliJ project folder, which has all the headers of the methods you'd need to implement to create the program as well as more specific hints for doing so. You are not required to use this starter code and you may modify it however you wish, such as by adding additional methods or changing return and parameter types.

You can unzip the starter code file to get a project folder. Then, open this in IntelliJ and work with it as you would any other project.

Make extensive use of methods to organize your code and break the problem down into manageable pieces. The starter code contains all the method headers you would need for this – the solution we wrote has all these methods and no more than these methods – but feel free to add, modify, or remove them as long as it helps you.

Start with the two-player game, since that'll be most similar to what you're familiar with and makes testing easier. After you have that, the only challenge with the one-player game will be implementing the AI algorithms, as the rest of the game logic will carry over and can be reused.

You must use an ArrayList to store the game history, and you must use a 2D char array to store a game state.

The starter code has three static variables in the TicTacToe class, which are used to centrally store what symbols in the game represent the empty space, the player one symbol, and the player two symbol. Throughout your code, you must refer to these static variables rather than hard-coding the game to rely on ' ', 'X', and 'O'. You may not add additional static variables to the class. These are effectively used as configuration options for the game.

Certain methods in the starter code are marked as private. Since we're only using one class, there is no functional difference. Later on in the course you'll see why some have their access from other classes restricted in this way. All methods are static. You'll also see why when we cover what it means for methods and variables to be static in detail.

Your program should conform to these edge-case expectations regarding user input:

- In every menu, check whether the user's choice of menu option is valid and inform them if their choice isn't valid.
- When inputting positions on the board, you do not need to handle non-numerical or non-integer inputs. Assume they will only input integers. However, you should check that the integers they inputted are valid. This means their input should both be within the 3x3 grid and should be an unoccupied space. Inform the user if their input was invalid and which of these two rules it broke.

Deliverables & Reminders

The usual deliverables and reminders apply. Compress and submit your project folder to Canvas before the deadline (or after with penalty, per the syllabus). The more serious warnings are listed below, but of course past assignments have more detailed instructions.

Ensure your submission contains the necessary file(s) and is the correct version of your code that you want graded. This is your warning. You will not be allowed to resubmit this assignment after the due date if you submit the wrong file.

Remember that this is graded on accuracy and you are not allowed to collaborate on your solutions with others. Plagiarism or cheating will be dealt with seriously.

Rubric

A rubric is also available on the assignment page on Canvas, but here is an outline of the point breakdown for different categories.

- (10pts) Code cleanliness, conventions, and comments
  - (10/10pts) for >95% correctness and all required comments
  - (7.5/10pts) for repeated minor mistakes or one egregious mistake, or for missing either overall or inline comments
  - (5/10pts) for code that is sloppy and hard to understand or is missing all comments
  - (<=2.5/10pts) for code that has no comments, is hard to understand, and consistently does not follow Java conventions
- (25pts) Two-player game
  - (1pts) for coin-flip that decides who goes first randomly
  - (2pts) for displaying the past game state before a player's turn
  - (8pts) for tracking game history correctly using an ArrayList of game states
    - (-2pts per) for minor oversights like not adding the initial game state to the game history without handling that design decision in the replay, etc.
  - (10pts) for ensuring that an inputted move is within game bounds and is unoccupied, re-prompting for input if invalid, and displaying the reason why the move is invalid (either out of bounds or already occupied)
  - (2pts) for game ends on a win, displaying winner player's name
  - (2pts) for game ends on a draw
- (30pts) One-player game
  - (10pts) for adapting general game behavior from two-player game
    - (1pt) for random first turn
    - (2pts) for displaying game state before player turn
      - (-1pt) for displaying game state before AI turn
    - (2pts) for tracking game history using an ArrayList of game states
    - (3pts) for validating user move inputs, warnings, etc.
    - (2pts) for game ending on a win or draw, displaying winner name
  - (5pts) for easy AI that makes a random valid move
  - (5pts) for medium AI smarter than easy but worse than hard
  - (10pts) for hard AI that follows algorithm correctly to never lose
- (20pts) Replay last game
  - (7.5pts) for correctly displaying the names of the users and the symbols they had in the game
    - (-5pts) if the symbols associated with the players aren't always right

    - (1.25pts) for correctly displaying the names of each person who took the turn in the replay
    - (1.25pts) for correctly displaying the game outcome at the end
    - (10pts) for iterating over the ArrayList to display each turn
- (10pts) Menu functionality throughout
    - (2pts) for a menu system that takes user input, executes the appropriate code on valid entry and warns the user of invalid entry, repeating until user ends program
    - (3pts) for correct names input collection – the first name should be player one, the computer should be player two and be given a reasonable name, for example
    - (3pts) for not executing the replay unless a past game exists
    - (3pts) for repeatedly asking for difficulty before starting the one-player game
- (5pts) for using static/global variables correctly throughout
    - (-1pt) per instance where a symbol was hardcoded rather than referring to appropriate static class char variable
    - (0pts) for this category if additional static variables are added

Sample program output

Two players, general case with replay display

```
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: D
No match found.
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: 2
Enter player 1 name: Nazim
Enter player 2 name: Obed
Tossing a coin to decide who goes first!!!
Nazim gets to go first.
   |   |
-----------
   |   |
-----------
   |   |
Nazim's turn:
Nazim enter row: 0
Nazim enter col: 0
 X |   |
-----------
   |   |
-----------
   |   |
Obed's turn:
Obed enter row: 0
Obed enter col: 0
That space is already taken. Try again.
Obed enter row: 0
Obed enter col: 3
That row or column is out of bounds. Try again.
Obed enter row: 1
Obed enter col: 1
 X |   |
-----------
   | O |
-----------
   |   |
Nazim's turn:
Nazim enter row: 1
Nazim enter col: 0
 X |   |
-----------
 X | O |
-----------
   |   |
Obed's turn:
Obed enter row: 2
Obed enter col: 0
 X |   |
-----------
 X | O |
```

```
-----------
 O |   |
Nazim's turn:
Nazim enter row: 0
Nazim enter col: 2
 X |   | X
-----------
 X | O |
-----------
 O |   |
Obed's turn:
Obed enter row: 1
Obed enter col: 2
 X |   | X
-----------
 X | O | O
-----------
 O |   |
Nazim's turn:
Nazim enter row: 0
Nazim enter col: 1
Nazim wins!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: D
Nazim (X) vs Obed (O)
   |   |
-----------
   |   |
-----------
   |   |
Nazim:
 X |   |
-----------
   |   |
-----------
   |   |
Obed:
 X |   |
-----------
   | O |
-----------
   |   |
Nazim:
 X |   |
-----------
 X | O |
-----------
   |   |
Obed:
 X |   |
-----------
 X | O |
-----------
 O |   |
Nazim:
 X |   | X
-----------
 X | O |
```

```
-----------
 O |   |
Obed:
 X |   | X
-----------
 X | O | O
-----------
 O |   |
Nazim:
 X | X | X
-----------
 X | O | O
-----------
 O |   |
Nazim wins!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: Q
Thanks for playing. Hope you had fun!
```

Two players, game ends in tie with replay display

```
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: 2
Enter player 1 name: Nazim
Enter player 2 name: Obed
Tossing a coin to decide who goes first!!!
Obed gets to go first.
   |   |
-----------
   |   |
-----------
   |   |
Obed's turn:
Obed enter row: 1
Obed enter col: 1
   |   |
-----------
   | O |
-----------
   |   |
Nazim's turn:
Nazim enter row: 0
Nazim enter col: 2
   |   | X
-----------
   | O |
-----------
   |   |
Obed's turn:
Obed enter row: 0
Obed enter col: 0
 O |   | X
-----------
   | O |
-----------
```

```
   |   |
Nazim's turn:
Nazim enter row: 2
Nazim enter col: 2
 O |   | X
-----------
   | O |
-----------
   |   | X
Obed's turn:
Obed enter row: 1
Obed enter col: 2
 O |   | X
-----------
   | O | O
-----------
   |   | X
Nazim's turn:
Nazim enter row: 1
Nazim enter col: 0
 O |   | X
-----------
 X | O | O
-----------
   |   | X
Obed's turn:
Obed enter row: 2
Obed enter col: 0
 O |   | X
-----------
 X | O | O
-----------
 O |   | X
Nazim's turn:
Nazim enter row: 0
Nazim enter col: 1
 O | X | X
-----------
 X | O | O
-----------
 O |   | X
Obed's turn:
Obed enter row: 2
Obed enter col: 1
It's a draw!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: D
Nazim (X) vs Obed (O)
   |   |
-----------
   |   |
-----------
   |   |
Obed:
   |   |
-----------
   | O |
-----------
```

```
   |   |
Nazim:
   |   | X
-----------
   | O |
-----------
   |   |
Obed:
 O |   | X
-----------
   | O |
-----------
   |   |
Nazim:
 O |   | X
-----------
   | O |
-----------
   |   | X
Obed:
 O |   | X
-----------
   | O | O
-----------
   |   | X
Nazim:
 O |   | X
-----------
 X | O | O
-----------
   |   | X
Obed:
 O |   | X
-----------
 X | O | O
-----------
 O |   | X
Nazim:
 O | X | X
-----------
 X | O | O
-----------
 O |   | X
Obed:
 O | X | X
-----------
 X | O | O
-----------
 O | O | X
It's a draw!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: Q
Thanks for playing. Hope you had fun!
```

One player, playing a game in each mode and watching some replays

```
1. Single player
2. Two player
```

```
D. Display last match
Q. Quit
What do you want to do: s
's' is not a valid option.
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: 1
E: Easy
M: Medium
H: Hard
Choose difficulty: f
'f' is not a valid difficulty
E: Easy
M: Medium
H: Hard
Choose difficulty: E
Enter player 1 name: Nazim
Tossing a coin to decide who goes first!!!
Easy Computer gets to go first.
   |   |
-----------
   |   |
-----------
   |   |
Easy Computer's turn:
   |   |
-----------
   |   |
-----------
   | O |
Nazim's turn:
Nazim enter row: 1
Nazim enter col: 1
   |   |
-----------
   | X |
-----------
   | O |
Easy Computer's turn:
   |   |
-----------
 O | X |
-----------
   | O |
Nazim's turn:
Nazim enter row: 2
Nazim enter col: 0
   |   |
-----------
 O | X |
-----------
 X | O |
Easy Computer's turn:
   |   |
-----------
 O | X |
-----------
 X | O | O
Nazim's turn:
```

```
Nazim enter row: 0
Nazim enter col: 2
Nazim wins!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: D
Nazim (X) vs Easy Computer (O)
   |   |
-----------
   |   |
-----------
   |   |
Easy Computer:
   |   |
-----------
   |   |
-----------
   | O |
Nazim:
   |   |
-----------
   | X |
-----------
   | O |
Easy Computer:
   |   |
-----------
 O | X |
-----------
   | O |
Nazim:
   |   |
-----------
 O | X |
-----------
 X | O |
Easy Computer:
   |   |
-----------
 O | X |
-----------
 X | O | O
Nazim:
   |   | X
-----------
 O | X |
-----------
 X | O | O
Nazim wins!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: 1
E: Easy
M: Medium
H: Hard
Choose difficulty: M
Enter player 1 name: Nazim
```

```
Tossing a coin to decide who goes first!!!
Medium Computer gets to go first.
   |   |
-----------
   |   |
-----------
   |   |
Medium Computer's turn:
 O |   |
-----------
   |   |
-----------
   |   |
Nazim's turn:
Nazim enter row: 2
Nazim enter col: 2
 O |   |
-----------
   |   |
-----------
   |   | X
Medium Computer's turn:
 O |   |
-----------
 O |   |
-----------
   |   | X
Nazim's turn:
Nazim enter row: 2
Nazim enter col: 0
 O |   |
-----------
 O |   |
-----------
 X |   | X
Medium Computer's turn:
 O |   |
-----------
 O |   |
-----------
 X | O | X
Nazim's turn:
Nazim enter row: 1
Nazim enter col: 2
 O |   |
-----------
 O |   | X
-----------
 X | O | X
Medium Computer's turn:
 O |   | O
-----------
 O |   | X
-----------
 X | O | X
Nazim's turn:
Nazim enter row: 1
Nazim enter col: 1
 O |   | O
-----------
 O | X | X
```

```
-----------
 X | O | X
Medium Computer's turn:
 O | O | O
-----------
 O | X | X
-----------
 X | O | X
Medium Computer wins!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: 1
E: Easy
M: Medium
H: Hard
Choose difficulty: H
Enter player 1 name: Obed
Tossing a coin to decide who goes first!!!
Hard Computer gets to go first.
   |   |
-----------
   |   |
-----------
   |   |
Hard Computer's turn:
 O |   |
-----------
   |   |
-----------
   |   |
Obed's turn:
Obed enter row: 1
Obed enter col: 1
 O |   |
-----------
   | X |
-----------
   |   |
Hard Computer's turn:
 O | O |
-----------
   | X |
-----------
   |   |
Obed's turn:
Obed enter row: 0
Obed enter col: 2
 O | O | X
-----------
   | X |
-----------
   |   |
Hard Computer's turn:
 O | O | X
-----------
   | X |
-----------
 O |   |
Obed's turn:
```

```
Obed enter row: 1
Obed enter col: 0
 O | O | X
-----------
 X | X |
-----------
 O |   |
Hard Computer's turn:
 O | O | X
-----------
 X | X | O
-----------
 O |   |
Obed's turn:
Obed enter row: 2
Obed enter col: 2
 O | O | X
-----------
 X | X | O
-----------
 O |   | X
Hard Computer's turn:
It's a draw!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: D
Obed (X) vs Hard Computer (O)
   |   |
-----------
   |   |
-----------
   |   |
Hard Computer:
 O |   |
-----------
   |   |
-----------
   |   |
Obed:
 O |   |
-----------
   | X |
-----------
   |   |
Hard Computer:
 O | O |
-----------
   | X |
-----------
   |   |
Obed:
 O | O | X
-----------
   | X |
-----------
   |   |
Hard Computer:
 O | O | X
-----------
```

```
   | X |
-----------
 O |   |
Obed:
 O | O | X
-----------
 X | X |
-----------
 O |   |
Hard Computer:
 O | O | X
-----------
 X | X | O
-----------
 O |   |
Obed:
 O | O | X
-----------
 X | X | O
-----------
 O |   | X
Hard Computer:
 O | O | X
-----------
 X | X | O
-----------
 O | O | X
It's a draw!
1. Single player
2. Two player
D. Display last match
Q. Quit
What do you want to do: Q
Thanks for playing. Hope you had fun!
```

Changelog

If any corrections, clarifications, etc. are made to the assignment after its release, they will be listed here. Any changes more substantial than simply fixing typos will get corresponding Canvas announcements. That way, you don't need to keep checking whether updates have occurred.