

Project #3: Tic-Tac-Toe

CS0007

Due: July 15th, at 11:59 PM

1 Instructions

The goal of this project is to test your ability to use the knowledge of Java that you have obtained up to this point in the course. The focus of this project is decision structures, looping structures, arrays, and methods. The goal is to create a classic tic-tac-toe game. The user will be playing against a computer opponent that will pick an open spot on the board.

2 Program Functionality

2.1 Program Steps

1. Display the current board state.
2. Take in a row and column as a position on the board for the user's move.
 - (a) If the user enters anything besides a valid row and column, tell the user it is invalid.
 - i. Start over from step 2.
 - (b) If the user enters a position that already has an "X" or an "O", then tell the user that the position is occupied.
 - i. Start over from step 2.
 - (c) If the user enters a valid, unoccupied row and column, this should be considered the user's move.
 - i. Display the board, now updated with the player's move
 - ii. If the user's move results in a win, tell the user that she has won.
 - A. Terminate the program.
 - iii. If the user's move results in a tie (cat), tell the user that the game has ended in a tie (cat).
 - A. Terminate the program.
 - iv. The computer opponent should then randomly choose an open spot on the board.
 - v. Display the board, now updated with the computer's move
 - vi. If the computer's move results in a win, tell the user that the computer has won.
 - A. Terminate the program.
 - vii. Go to step 2.

2.2 Details

- The board should be represented by a 3 x 3 (two dimensional) array of characters.
- It should be obvious to the user how to enter coordinates and what the state of the board is at any time (See “Example Execution” Section).
- The computer’s moves should be completely random at the least. If you want to do more intelligent move making, you may receive extra points.
- You are REQUIRED to write 3 methods (NOT including the main method). My implementation has 6 (Note: Yours could very well be different):

```
– public static void printBoard(char[] [] board)
– public static boolean isValidMove(String move)
– public static boolean submitMove(String move, char[] [] board, char player)
– public static boolean isWinner(char [] [] board, char player)
– public static boolean isCat(char[] [] board)
– public static String generateMove()
```

- In my example, the user goes first. If you want to randomly select which player goes first, this is also acceptable and will get you bonus points.
- All the normal rules to tic-tac-toe apply to this game. If you do not know the rules of tic-tac-toe, or have a specific question on the rules, feel free to ask the instructor.

3 Example Execution

```
*****
Tic-Tac-Toe!
*****
```

Please enter the column and then row of your move.

```
  A B C
1 | | | |
-----
2 | | | |
-----
3 | | | |
```

Player Move (X): banana

Invalid Input: Please enter the column and row of your move (Example: A1).

Player Move (X): A1

```
  A B C
1 |X| | |
-----
```

```
2 | | | |
-----
3 | | | |
```

Computer Move (0): C2

```
  A B C
1 |X| | |
-----
2 | | |O|
-----
3 | | | |
```

Player Move (X): C2
The space entered is already taken.

Player Move (X): B2

```
  A B C
1 |X| | |
-----
2 | |X|O|
-----
3 | | | |
```

Computer Move (0): B1

```
  A B C
1 |X|O| |
-----
2 | |X|O|
-----
3 | | | |
```

Player Move (X): A3

```
  A B C
1 |X|O| |
-----
2 | |X|O|
-----
3 |X| | |
```

Computer Move (0): A2

```
  A B C
1 |X|O| |
-----
```

```
2 |0|X|0|
-----
3 |X| | |
```

Player Move (X): C3

```
   A B C
1 |X|0| |
-----
2 |0|X|0|
-----
3 |X| |X|
```

You win!

4 Concluding Remarks

Since this is a project, I will leave interface design decisions to you. A portion of the grade will be on the design of the interface. Your program should be easy to use and understand. I highly suggest giving the user instructions when the program starts and gently reminding her when she inputs something invalid. You are required to have sufficient internal documentation for your program as discussed in class.

This is meant to be AN INDIVIDUAL ASSIGNMENT. Also, taking large segments of code from other sources without citing is plagiarism. The majority of this assignment should be YOUR OWN ORIGINAL WORK. I recommend getting a small storage device such as a flash drive to save your work on. You can get these for under \$20 at any Radio Shack, and many other stores. When you are done submit your project by zipping up the project folder and FTPing it to the drop box for this course as described in class and here: <http://www.cs.pitt.edu/~eth13/cs0007/submissionGuidelines.html>. On the course webpage there is a grading rubric which I will be grading from. Keep this in mind while doing the assignment.