Project #4: Object Oriented Tic-Tac-Toe

CS0007

Due: July 31st, 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 on object-oriented programming. The goal is to change your previous, procedural implementation of tic-tac-toe in project #3 into an object oriented one. See project #3's description for how the program should interact with the user. The difference to to create a more object-oriented program. See the Possible Object Design section for one possible valid implementation.

2 Possible Object Design

1. `char[][] Board` - An object representing a tic-tac-toe board

   (a) Attributes

      i. `gameBoard` - Two-dimensional character array representing the board

   (b) `Methods`

      i. Constructors

         A. `Board()` - Initializes the board to an empty board

         B. `Board(char [][] inBoard)` - Initializes the board to `inBoard`

      ii. Accessors

         A. `char[][] getGameBoard()` - returns the game board

         A. `void printBoard()` - prints the board to the screen

         B. `boolean submitMove(String move, char player)` - adds the move to the board, and returns true if the space specified in the move is not taken. If the space is taken, do not add the move and return false.

         C. `boolean isWinner(char player)` - Returns true if the player whose marker is passed as a parameter has won the game, false otherwise.

         D. `boolean isCat()` - Returns true if there is a CAT(tie), false otherwise.

         E. `boolean isMoveValid(String move)` - Returns true if the parameter is a move on the board, otherwise it returns false

2. `Player` - An object representing a player of the tic-tac-toe game

   (a) Attributes

      i. `char marker` - Character representing whether this player is using X or O as her marker

  ii. `boolean isHuman` - Boolean indicating if the player is a human player or a computer player

 (b) `Methods`

   i. Constructors

    A. `Player()` - Initializes the `marker` to X and `isHuman` to true

    B. `Player(boolean inIsHuman)` - Initializes `isHuman` to `inIsHuman` and if `inIsHuman` is true, `marker` to X, otherwise `marker` to O

    C. `Player(boolean inIsHuman, char inMarker)` - Initializes `isHuman` to `inIsHuman` and `marker` to X if `inMarker` is X or x, otherwise `marker` to O

   ii. Accessors

    A. `char getMarker()` - Returns the player's marker

    B. `boolean getIsHuman()` - Returns whether the player is human or not

    A. `String getPlayerMove()` - Returns the move from the player (either from human or computer).

    B. `String getHumanMove()` - Prompts the user for a move, retreieves input from the keyboard, and returns it.

    C. `String generateComputerMove()` - Returns a move randomly generated.

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 (O): 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 (O): 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 (O): A2

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

```
Player Move (X): C3

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

You win!
```

4   Concluding Remarks

You may use your own design of objects for this project, but you are required to write at least two classes, and your program should follow object-oriented programming principles. If you do not want to use your implementation of project #3, you can e-mail your instructor and ask for his implementation for a 10 point penalty. Bonus points will be given if you allow the option for two computer players to play each other, two human players to play each other, a human going first against a computer player, or a human going second against a computer player.

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.