Assignment #8
2-D Arrays: TicTacToe
Due: Sunday, 11/13/16, 11:59pm

(75 pts) You will write a program that plays the game Tic-Tac-Toe.

First, prompt the user to find out if he/she wants to play with one or two players. If the user wants to play with only one player, then the computer must play the one player. You can choose whatever algorithm you want for the computer, i.e. picking random places to put the piece or intelligently selecting your move based on player 1’s selection. However, you mustn’t ever select a position that has already been selected at any time!!!

After you determine how many players are playing the game, then you can determine what character each player wants to choose. You have to make sure that Player 2 or the computer doesn’t choose the same character as Player 1. Print the empty board, and then prompt the player(s) for their position on the board, printing the board after each turn. Make sure that the player(s) chooses valid positions on the board!!!

Example Tic-Tac-Toe:
Do you want 1 or 2 players? 2
Player 1: What character do you want? X
Player 2: What character do you want? O

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Player 1: Where would you like to put your X? 0 0

X |   |
---|---|
|   |   |
Player 2: Where would you like to put your O? 1 0

X |   |
---|---|
O |   |
Player 1: Where would you like to put your X? 1 1

X |   |
---|---|
O | X |
Player 2: Where would you like to put your O? 2 0

Player 1: Where would you like to put your X? 2 2

Congratulations Player 1, you are a winner!!!

Program Input:
- Determine whether the user wants to play 1 or 2 players for Tic-Tac-Toe
- The character/game piece each player wants.
- Player’s choice of position on the board.

Program Output:
- The view of the board after each player’s turn, along with an initial empty board.
- A prompt asking for the player’s selection on the board.

(25 pts) Program Design
Begin by designing your program using these steps, and write steps 1, 2, and 4 on paper or in a text editor. Then, implement the program using Python.

- Step 1: Problem Analysis. (5 pts)
  a. Comments about the problem to aid in understanding it.
  b. Description of the knowledge base (this list would include what you would be expected to know to follow the solution).

- Step 2: Program Design. (15 pts) List the specific steps needed to play the game of Tic-Tac-Toe. Remember, you have to be very explicit here to make sure the computer can accomplish the task using your directions.
  1. Define all the functions needed.
  2. What are the preconditions, post conditions, and return values?
    ....

- Step 3: Program Implementation. This is the Python code that plays TicTacToe.
- Step 4: Program Testing. (5 pts)
  Create a Test Plan with several test cases including the average and extreme cases.

(10 pts) Extra Credit:
Tic-Tac-Toe Error Handling:
- Player’s choice is not appropriate, such as a non-positive int for row/column
- Player chooses a position that is not on the board, i.e. row 5, column 2.
- Player chooses a position that is already occupied.
- Player 2 chooses Player 1’s character.
Extra Credit 2:
All your functions, including main(), must not have over 15 lines of code, this doesn’t include comments or blank lines.

Some functions you might want to include are an initialize_board(), which initializes the board to spaces, a determine_player_choice() that allows players to pick their pieces, i.e. ‘X’ or ‘O’, fill_board(), which fills the board with the player’s choice, a print_board() that prints the board to the screen after each user’s turn, is_full() to check if the board is full, a check_for_winner(), which checks to see if there is a winner, and a print_winner_results() that prints the results of the game to the screen.

Electronically submit your implementation (.py file) and program design (.pdf) by the assignment due date, using TEACH:
https://secure.engr.oregonstate.edu:8000/teach.php?type=want_auth