TicTacToe

Professor Caleb Fowler
June 5, 2023

Problem.

You are going to program a game of tic-tac-toe. The human will play the computer. Use the following pseudocode to guide your design:

```
Create an empty Tic _ Tac _ Toe board
Display the Game Instructions
Determine who goes first
Display the board
While nobody has won and it's not a tie
        If it's the human's turn
                Get the human's move
                Update the board with the human's move
        Otherwise
                Calculate the computer's move
                Update the board with the computer's move
        Display the board
        Switch turns
Congratulate the winner or declare a tie
```

Requirements.

- Use whatever method is expedient to generate the computer moves; this isn't an AI class. Random numbers work fine.
- Use functions whenever possible to modularize your code. Use function prototypes and code the functions under main().
- Use at least 1 array to keep track of the game board.
- Create a main event loop to replay the game over and over. Once a game ends, reprompt to play again or exit. You do not need to rerun the introductory menu.
- Re-prompt if incorrect data was entered. Keep re-prompting until the client get's it right.
- Include a void ProgramGreeting() function. This will run automatically once when the program first starts. This function should display (on individual lines):

  – A welcome message.
  – Your name (as author).
  – System date. Format this as MonthName day, year. Example:

June 30, 1988.

- Store the results of all computations.
- Include your specification comments above the main portion of your code where you implement the specification. No credit if this is missing!
- Use white-space and comments to make your code more readable.
- Put a Source File Header at the top of your source file.
- Include a ProgramGreeting() function.
- Use function prototypes for all functions. Functions go below main().
- Do not use c (.h) style libraries. Use C++ libraries instead.
- Your program must compile in C++ on Ubuntu.
- Your program must generate logically correct output.
- Program activities are split into logical 'chunks' or paragraphs. I'm expecting paragraphs for input, processing (if any), and output operations.
- If there is non integer output, force the computer to always display 3 places to the right of the decimal.

Specifications.

// Specification C1 - Random Start
Determine which player will start the game - randomly. Make sure you tell the client who's starting.

// Specification C2 - Simple Display
Display in a 2D table X, O and empty spaces.

// Specification C3 - AI
Comment the start of your AI code which determines where the computer will go during it's turn. You can make this as easy or complex as you like.

// Specification B1 - Start Menu
Create a simple menu, 1 - Explain Game, 2 - Play, 3 - Quit. This menu will run once every time the program runs. Don't go into excruciating detail explaining the game. Explaining the game is NOT the same as your program greeting.

//Specification B2 - Valid Menu Choices
Filter the input for valid numbers. Re-prompt incorrect entries.

// Specification B3 – Valid Move
Only allow the various players to make valid moves. This is more advanced input validation.

// Specification A - Reflection

Using the following prompts, generate feedback on your assignment using ChatGPT. Often shift-return will generate a blank line without submitting the prompt.

Analyze this student's code in relationship to commonly accepted C++ programming practices and standards. This is an assignment from an Introductory C++ programming course. Indicate if this code is likely to compile or run correctly in addition to your other feedback.
<Copy and Paste your source code here>

Review and reflect on the feedback the system gives you. Write this up in 250 words or more. Include your write up as a block comment at the bottom of your assignment. **Also indicate the number of words in your write up, as well**. You may wish to:

Comment on the overall quality of the assessment. Was it accurate? Did it make sense? Did you find it useful? Does it align with what you coded? You may wish to discuss one or two main themes the AI identified in relationship to your coding.

Memorialize your reaction to the feedback. Do you find it easier to get feedback from a computer or a human? Was there any advice in particular which was helpful to you? Can you think of a better prompt to generate the information you need? You can also use this as coding notes to yourself to help you remember some of the hard won lessons from this project.

I am NOT interested in the feedback from the generative AI. Do NOT copy and paste that in your program. I am interested in your thoughts about it however. You are free to use multiple prompts as well. I do not use this tool to grade your assignments - it’s not accurate enough for my purposes. I will not grade the quality of your content. I want this to be useful to you and not worry about saying something “grade worthy”. I suggest this is the last step you perform before you turn in your assignment. You can do it earlier if you wish, but the feedback will not be as useful. You are free to revise your code in light of the feedback you get, but remember, the assignment is what I grade to, not ChatGPT. Make sure you confirm your code runs before you turn it in.