

Portfolio One Requirements

CSC 122-002

Grading

We decided as a group that portfolio one will consist of two parts:

1. A Required Tic-Tac-Toe Submission (Outlined Below)
2. An Optional Extra Credit Submission

We decided as a group to NOT do the portfolio add-ons this time around.

Tic-Tac-Toe

Client Request

I want you to write a game of tic-tac-toe that satisfies the following requirements:

- I want you to implement a human vs. human game of tic-tac-toe that can be played on the terminal.
- This should be a traditional game of tic-tac-toe
 - a. Players take turns marking cells on a 3x3 board
 - b. You can win by getting three marks in a row vertically, horizontally, or diagonally
- At the end of the game, I want to know which player won or if the game ended in a tie.
- Your program should not crash, even if I give it weird inputs.
- You are welcome to work together over the next week on this project. However, each person should submit their own code in their own GitHub repository.

Agreed Upon Acceptance Criteria

These are the acceptance criteria that I will be grading for the portfolio. You are welcome to implement more features if you feel like that will help you accomplish the criteria listed below (e.g. add a main menu). Be aware that if you add a feature in a portfolio, you **MUST** keep it maintained and functional in all of the future portfolios.

- Player 1 is always “X” and Player 2 is always “O”.

- All of the win conditions and rules should be programmed so that the game can tell the user when a game is over and the status of the endgame.
- The full, 3x3 tic-tac-toe board should be visualized in the terminal
 - The cells of the board should be indexed in some way (e.g. 1-9, a-i)
 - The board should be formatted neatly and look good
- A player should be able to make a move by selecting an index on the board. The game will keep track of who's turn it is, what symbol to mark the board with, and the status of the game.
- When a player makes a move, the game will check to ensure that the move is valid and re-prompt the player for a move if it is not:
 - Reject moves in the same space
 - Reject moves that are out of bounds
 - Reject moves where the user enters a random character or string
 - Reject any moves that occur after the game ends
- When the game ends, a closing message will be printed on the terminal. This message can be as simple as "X won", "O won", or "Tie".
- When the game is over, the program should exit gracefully. The program does NOT need to prompt the player(s) to play again.