CS-150-01

Professor Shafaei

11/12/2025

Sebastian Baumann, Tristan Hibbard, William Sanderfer

1. The project our group chose to implement is a **Tic-Tac-Toe game**, a classic two-player strategy game that dates back to ancient times. The modern version involves two players alternately marking spaces in a 3 by 3 grid, one using “X” and the other “O.” The goal is to be the first player to get three of their marks in a row, either horizontally, vertically, or diagonally. This Python implementation recreates the traditional game in a text-based console environment, allowing users to play against each other interactively.
2. The game runs entirely in a command-line interface and uses user prompts for interaction.

   When executed, the program:

   - Displays an empty Tic-Tac-Toe board.
   - Prompts Player 1 to decide whether to go first.
   - Asks the players to choose their symbols (“X” or “O”).
   - Alternates turns between Player 1 and Player 2, prompting for input positions (1–9) corresponding to board spots.
   - Updates and displays the board after each move.
   - Checks for a win or tie after each turn.

- Offers the option to play again after a game concludes.

3. i. The program consists of several defined functions:

- **the_board(board)** – Displays the current state of the Tic-Tac-Toe board.
- **player_key()** – Manages player setup, including who goes first and symbol assignment (“X” or “O”).
- **enter_key(key, bp)** – Handles player turns, board updates, and input validation.
- **game_winner(key, game)** – Checks for winning conditions after each move.
- **game_play()** – The main driver function that initializes the board, starts the game, and handles replay logic.

ii. **Data Types and Collections:**

- **Lists**: The board is stored as a list with 10 elements (index 0 unused), where each index (1–9) represents a position on the board.
- **Strings**: Used for player input, player symbols (“X”, “O”), and printed output.
- **Integers**: Used for move counting and indexing within the board.

iii. **Libraries, Classes, and Functions**

- No external or third-party libraries are used.
- Only built-in Python functions like input(), print(), and range() are used.

4. There are some limitations, one of those being that no AI or single-player mode is implemented; both players must be human users. Another is that we did not accommodate for if the user ever inputs a letter instead of a number, there would be an error. Other than those issues, the game is fully functional.

5. **Some enhancements we could add are:**

I. Single-player mode: Introduce an AI opponent using algorithms such as *minimax* or random move selection.

II. Graphical User Interface (GUI): Use a library like tkinter or pygame to make the game more visually interactive.

III. Network or Online Play: Enable two players to compete over a network connection.

IV. Score Tracking System: Maintain and display a scoreboard across multiple games.

V. Enhanced Input Validation: Implement robust error checking to handle invalid inputs.