



VaultofCodes

Final Project

By
Thatha Keerthi

Table Of Contents

- Introduction
- Working
- Uses
- Advantages & Disadvantages
- Future scope
- Conclusion



Introduction

- **Overview:**
 - Tic-Tac-Toe is a simple two-player game where players take turns marking spaces in a 3x3 grid.
 - The objective is to align three of your symbols (either 'X' or 'O') horizontally, vertically, or diagonally.
- **Objective:**
 - Develop a simple, text-based Tic-Tac-Toe game using Python.
 - Enhance understanding of basic programming concepts and logical thinking.



Working

Step by step process:

- a. **Board Initialization:** A 3x3 grid is initialized with empty spaces.
- b. **Player input:** players are prompted to choose a position (1-9) to place their marker.
- c. **Game Loop:** The board is updated after each move, and the game checks for the winner or a draw.
- d. **Winning Condition:** The game checks if the current player has won by matching any of the possible winning combinations.
- e. **Draw Condition:** The game checks if all positions are filled without a winner, resulting in draw.
- f. **Replay Option:** After the game ends, players can choose to play again or exit.



Uses

- **Educational Tools:**
 - Helps beginners learn basic python programming concepts.
 - Demonstrates logic, conditional ,loop and functions in a practical concept.
- **Entertainment:**
 - Provides a simple and engaging game to play.
- **Educational Tools:**
 - Enhances logical thinking and strategy development.



Advantages & Disadvantages

- **Advantages:**
 - **Simple and Easy to Understand:** The game logic is straightforward, making it accessible to beginners.
 - **Interactive:** Provides immediate feedback and interaction with the user.
 - **Customizable:** Easy to modify and extend with additional features.
- **Disadvantages:**
 - **Limited Functionality:** The game is text-based and lacks a graphical interface.
 - **Basic AI:** No AI implementation for playing against the computer; requires two human players.
 - **No Persistence:** The game doesn't save progress or maintain a history of games played.



Future Scope

- **Improvements:**
 - **Graphical User Interface (GUI):** Implement a GUI using libraries like Tkinter or Pygame.
 - **Single-Player Mode:** Add AI functionality to allow single-player games against the computer.
 - **Score Tracking:** Implement a scoring system to track wins, losses, and draws over multiple rounds.
 - **Online Multiplayer:** Extend the game to support online multiplayer mode using networking libraries.



Conclusion

- The Tic-Tac-Toe game in Python is a simple project that effectively demonstrates the basics of programming.
- It's a stepping stone for more complex projects and helps solidify foundational concepts.



THANK YOU!!!

VaultofCodes

Final Project

Create a basic game like a Tic-Tac-Toe in Python Programming

Code:

```
def print_board(board):
    """Function to print the game board."""
    print("\n")
    print(f" {board[0]} | {board[1]} | {board[2]} ")
    print("---+---+---")
    print(f" {board[3]} | {board[4]} | {board[5]} ")
    print("---+---+---")
    print(f" {board[6]} | {board[7]} | {board[8]} ")
    print("\n")

def check_winner(board, player):
    """Function to check if a player has won."""
    win_conditions = [(0, 1, 2), (3, 4, 5), (6, 7, 8), # rows
                      (0, 3, 6), (1, 4, 7), (2, 5, 8), # columns
                      (0, 4, 8), (2, 4, 6)]           # diagonals

    return any(board[a] == board[b] == board[c] == player for a, b, c in win_conditions)

def check_draw(board):
    """Function to check if the game is a draw."""
    return all(spot != ' ' for spot in board)

def get_player_move(board, current_player):
    """Function to get a valid move from the current player."""
    while True:
        move = input(f"Player {current_player}, choose a position (1-9): ")

        if move.isdigit() and 1 <= int(move) <= 9:
            move = int(move) - 1
            if board[move] == ' ':
                return move
```

```

        return move
    else:
        print("This position is already taken. Try another one.")
    else:
        print("Invalid input. Please enter a number between 1 and 9.")

def play_game():
    """Main function to play the Tic-Tac-Toe game."""
    board = [' '] * 9
    current_player = 'X'

    while True:
        print_board(board)
        move = get_player_move(board, current_player)
        board[move] = current_player

        if check_winner(board, current_player):
            print_board(board)
            print(f"Player {current_player} wins!")
            break

        if check_draw(board):
            print_board(board)
            print("It's a draw!")
            break

        current_player = 'O' if current_player == 'X' else 'X'

def main():
    """Main function to run the game loop."""
    while True:
        play_game()
        replay = input("Do you want to play again? (y/n): ").lower()
        if replay != 'y':
            print("Thanks for playing! Goodbye!")
            break

if __name__ == "__main__":
    print("Welcome to Tic-Tac-Toe")
    main()

```

Output:

```
Welcome to Tic-Tac-Toe
```

```
  |  |  
--+--+  
  |  |  
--+--+  
  |  |
```

```
Player X, choose a position (1-9): 9
```

```
  |  |  
--+--+  
  |  |  
--+--+  
  |  | X
```

```
Player O, choose a position (1-9): 5
```

```
  |  |  
--+--+  
  | O |  
--+--+  
  |  | X
```

```
Player X, choose a position (1-9): 1
```

```
X |  |  
--+--+  
  | O |  
--+--+  
  |  | X
```

```
Player O, choose a position (1-9): 2
```

```
X | O |  
--+--+  
  | O |  
--+--+  
  |  | X
```

```
Player X, choose a position (1-9): 8
```

```
X | O |  
--+--+  
  | O |  
--+--+  
  | X | X
```

Player O, choose a position (1-9): 4

```
X | O |  
---+---+---  
O | O |  
---+---+---  
   | X | X
```

Player X, choose a position (1-9): 7

```
X | O |  
---+---+---  
O | O |  
---+---+---  
X | X | X
```

Player X wins!

Do you want to play again? (y/n): n

Thanks for playing! Goodbye!

...Program finished with exit code 0

Press ENTER to exit console.