Untitled2

September 28, 2025

Tic-Tac-Toe – terminal game

References I checked while learning: 1) Python Official Docs – input/print, lists, loops: https://docs.python.org/3/tutorial/index.html *2) Real Python – basics of building small console games:* https://realpython.com/python-rock-paper-scissors/ *3) GeeksforGeeks – examples of checking win conditions in Tic Tac Toe: https://www.geeksforgeeks.org/tic-tac-toe-gui-in-python-using-pygame/ (logic ideas) 4) Invent with Python – simple game structure and functions:* https://inventwithpython.com/chapter9.html (tic-tac-toe chapter)

Note: I didn’t copy code directly; I used these to understand how to structure the board, check winners, and handle user input.

[2]:

```python
def new_board():
    return [" "] * 9

def print_board(b):
    # Show numbers on empty cells so the player knows where to play
    cells = [str(i+1) if b[i] == " " else b[i] for i in range(9)]
    print()
    print(f" {cells[0]} | {cells[1]} | {cells[2]}")
    print("---+---+---")
    print(f" {cells[3]} | {cells[4]} | {cells[5]}")
    print("---+---+---")
    print(f" {cells[6]} | {cells[7]} | {cells[8]}")
    print()

def winner(b):
    lines = [
        (0,1,2),(3,4,5),(6,7,8),
        (0,3,6),(1,4,7),(2,5,8),
        (0,4,8),(2,4,6)
    ]
    for i, j, k in lines:
        if b[i] != " " and b[i] == b[j] == b[k]:
            return b[i]
    return None

def board_full(b):
    return all(c != " " for c in b)
```

```python
def ask_move(b, mark):
    while True:
        raw = input(f"{mark}'s move (1-9): ").strip()
        if not raw.isdigit():
            print("Please type a number 1-9.")
            continue
        idx = int(raw) - 1
        if idx < 0 or idx > 8:
            print("Number must be from 1 to 9.")
            continue
        if b[idx] != " ":
            print("That cell is taken. Choose another.")
            continue
        return idx

def random_ai_move(b):
    free = [i for i,c in enumerate(b) if c == " "]
    return random.choice(free)
```

[3]:

```python
def play_pvp():
    b = new_board()
    turn = "X"
    print_board(b)
    while True:
        idx = ask_move(b, turn)
        b[idx] = turn
        print_board(b)
        w = winner(b)
        if w:
            print(f"  {w} wins!")
            break
        if board_full(b):
            print("  Draw!")
            break
        turn = "O" if turn == "X" else "X"

def play_vs_ai():
    b = new_board()
    print("You are X. Computer is O (random moves).")
    turn = "X"
    print_board(b)
    while True:
        if turn == "X":
            idx = ask_move(b, "X")
        else:
            print("Computer is thinking...")
            idx = random_ai_move(b)
        b[idx] = turn
        print_board(b)
        w = winner(b)
        if w:
            if w == "X":
                print("  You win!")
            else:
                print("  Computer wins!")
            break
        if board_full(b):
            print("  Draw!")
            break
        turn = "O" if turn == "X" else "X"
```

[ ]:

```
def main():
    print("=== Tic-Tac-Toe ===")
    print("1) Player vs Computer (random AI)")
    print("2) Player vs Player")
    while True:
        choice = input("Choose 1 or 2: ").strip()
        if choice == "1":
            play_vs_ai()
            break
        if choice == "2":
            play_pvp()
            break
        print("Please type 1 or 2.")

if __name__ == "__main__":
    main()
```

```
=== Tic-Tac-Toe ===
1) Player vs Computer (random AI)
2) Player vs Player

Choose 1 or 2:  1

You are X. Computer is O (random moves).

 1 | 2 | 3
---+---+---
 4 | 5 | 6
---+---+---
 7 | 8 | 9
```