

```

def display_board(board):
    """Prints the current state of the Tic-Tac-Toe board."""
    for row in board:
        print("|", " | ".join(str(x) for x in row), "|")

def is_valid_move(board, move):
    """Checks if a given move is valid on the current board."""
    row, col = move // 3, move % 3
    return 0 <= row < 3 and 0 <= col < 3 and board[row][col] == " "

def check_win(board, player):
    """Checks if the given player has won the game."""
    win_conditions = [[(0, 0), (0, 1), (0, 2)], [(1, 0), (1, 1), (1, 2)], [(2, 0), (2, 1), (2, 2)],
                      [(0, 0), (1, 0), (2, 0)], [(0, 1), (1, 1), (2, 1)], [(0, 2), (1, 2), (2, 2)],
                      [(0, 0), (1, 1), (2, 2)], [(0, 2), (1, 1), (2, 0)]]
    for condition in win_conditions:
        if all(board[row][col] == player for row, col in condition):
            return True
    return False

def is_board_full(board):
    """Checks if the board is full."""
    for row in board:
        for cell in row:
            if cell == " ":
                return False
    return True

def main():
    board = [[" " for _ in range(3)] for _ in range(3)]
    current_player = "X"

    while True:
        display_board(board)
        print(f"Player {current_player}'s turn.")
        move = int(input("Enter your move (1-9): ")) - 1

        if not is_valid_move(board, move):
            print("Invalid move. Try again.")
            continue

        row, col = move // 3, move % 3
        board[row][col] = current_player

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

        if is_board_full(board):
            display_board(board)
            print("It's a draw!")
            break

        current_player = "O" if current_player == "X" else "X"

if __name__ == "__main__":
    main()

```

```

↩
| | | |
| | | |
| | | |
Player X's turn.

```

