```
import random
from rich.console import Console
from rich.table import Table

console = Console()
BOARD_SIZE = 5
NUMBERS = [1, 2, 3, 4, 5]

def create_board():
    return [[0 for _ in range(BOARD_SIZE)] for _ in range(BOARD_SIZE)]

def display_board(board):
    table = Table(title="Number Clash")
    for col in range(BOARD_SIZE):
        table.add_column(str(col + 1), justify="center")
    for row in board:
        table.add_row(*[str(cell) if cell != 0 else "-" for cell in row])
    console.print(table)

def is_valid_move(board, row, col, num):
    if board[row][col] != 0:
        return False
    for dr, dc in [(-1,0),(1,0),(0,-1),(0,1)]:
        r, c = row + dr, col + dc
        if 0 <= r < BOARD_SIZE and 0 <= c < BOARD_SIZE:
            if board[r][c] == num:
                return False
    return True

def get_available_moves(board):
    moves = []
    for r in range(BOARD_SIZE):
        for c in range(BOARD_SIZE):
            for n in NUMBERS:
                if is_valid_move(board, r, c, n):
                    moves.append((r, c, n))
```

```
    return moves

def player_move(board, player_marks):
    while True:
        try:
            move = input("マス位置（行 列 数字）：").split()
            r, c, n = map(int, move)
            if is_valid_move(board, r-1, c-1, n):
                board[r-1][c-1] = n
                player_marks.append(n)
                break
            else:
                print("❌ そこには置けません！")
        except:
            print("⚠️ 入力形式は「行 列 数字」(例: 2 3 4)")

def ai_move(board, ai_marks):
    moves = get_available_moves(board)
    if moves:
        r, c, n = random.choice(moves)
        board[r][c] = n
        ai_marks.append(n)
        console.print(f"🤖 AI が（{r+1},{c+1}）に {n} を置いたよ", style="cyan")
    else:
        console.print("🤖 AI は置けませんでした", style="red")

def is_board_full(board):
    return all(cell != 0 for row in board for cell in row)

def game_loop():
    board = create_board()
    turn = 0  # 0 = プレイヤー, 1 = AI
    player_marks = []
    ai_marks = []

    while not is_board_full(board):
```

```
            display_board(board)
            if turn == 0:
                console.print("🎮 あなたのターン", style="green")
                player_move(board, player_marks)
            else:
                ai_move(board, ai_marks)
            turn = 1 - turn

        display_board(board)
        player_score = sum(player_marks)
        ai_score = sum(ai_marks)

        console.print(f"¥n🏁 スコア集計：あなた [bold green]{player_score}[/] | AI [bold
cyan]{ai_score}[/]")
        if player_score > ai_score:
            console.print("🎉 あなたの勝ち！", style="bold green")
        elif player_score < ai_score:
            console.print("🤖 AI の勝ち！", style="bold cyan")
        else:
            console.print("🤝 引き分け！", style="bold yellow")

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