

```

def get_player_prefix(player_name):
    """ 根據玩家名稱返回棋子代號前綴 # 陰府棋局：十二生肖的命運之戰
    (簡化文字版) """
    return "1_" if player_name == "玩家 1" else "2_"

def 初始化棋盤():
    """ 初始化棋盤，設定棋子初始位置 (需玩家手動在實際棋盤上擺放) """
    # 在這裡，我們只需要設定棋盤的邏輯結構，不需要視覺化棋盤
    board = {} # 使用字典來表示棋盤，key 為座標 (例如 "A1", "B2")，value
    為棋子代號或 None
    # 假設玩家手動在實際棋盤上擺放好初始棋子
    print("請在你的棋盤上，按照以下初始設定擺放棋子：")
    print("  玩家 1 (假設為先手): 陰府棋子放在 D1， 其餘生肖棋子放在第
    一、二排 (例如 鼠 A1, 牛 B1, 虎 C1...)")
    print("  玩家 2 (假設為後手): 陰府棋子放在 D8， 其餘生肖棋子放在第
    七、八排 (例如 鼠 H8, 牛 G8, 虎 F8...)")
    return board

def 顯示棋盤狀態(board):
    """ 顯示棋盤的文字狀態 (簡化版，只顯示棋子位置) """
    print("\n--- 棋盤狀態 ---")
    for row in ['8', '7', '6', '5', '4', '3', '2', '1']:
        line = row + " | "
        for col in ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']:
            pos = col + row
            piece = board.get(pos, " ") # 若該位置沒有棋子，顯示空白
            line += piece + " | "
        print(line)
    print(" -----")
    print("      A  B  C  D  E  F  G  H\n")

def 取得玩家輸入(player_name):
    """ 取得玩家的文字指令輸入 """
    指令 = input(f"{player_name} 請輸入你的指令 (例如: move 鼠 A1 to B2, list
    鼠, attack 牛 C3 D4, help): ").strip().lower()
    return 指令

def 顯示指令說明():

```

```

""" 顯示可用的文字指令說明 (更新版) """
print("\n--- 指令說明 ---")
print("list [棋子代號]                                - 列出指定棋子的可移動位置並編號 (例如: list 鼠)")
print("move [數字]                                    - 選擇編號移動位置 (需先使用 list 指令，例如: move 1)")
print("move [棋子代號] [起始位置] to [目標位置]        - 移動棋子到指定位置 (例如: move 鼠 A1 to B2)")
print("attack [攻擊棋子代號] [目標位置]                - 攻擊目標位置的棋子 (例如: attack 牛 C3 D4)")
print("end_turn                                          - 結束當前回合")
print("help                                              - 顯示指令說明")
print("--- 請注意 ---")
print(" * 棋子代號：鼠, 牛, 虎, 兔, 龍, 蛇, 馬, 羊, 猴, 雞, 狗, 豬, YF (陰府)")
print(" * 位置：西洋棋盤座標，例如 A1, H8, C5")
print(" * 使用 'list' 指令後，可以使用 'move [數字]' 快速選擇移動目標。")

```

```

def pos_to_coords(pos):
    """ 將棋盤位置 (例如 A1) 轉換為座標 (例如 (0, 0)) """
    col = ord(pos[0]) - ord('A')
    row = int(pos[1]) - 1
    return col, row

```

```

def coords_to_pos(coords):
    """ 將座標 (例如 (0, 0)) 轉換為棋盤位置 (例如 A1) """
    col_char = chr(coords[0] + ord('A'))
    row_char = str(coords[1] + 1)
    return col_char + row_char

```

```

def is_valid_move(piece_name, start_pos, end_pos, board, current_player):
    """ 判斷從起始位置到目標位置的移動是否合法 (簡化規則) """
    start_col, start_row = pos_to_coords(start_pos)
    end_col, end_row = pos_to_coords(end_pos)

    # 檢查目標位置是否超出棋盤邊界

```

```

if not (0 <= end_col < 8 and 0 <= end_row < 8):
    return False

# 檢查目標位置是否已被我方棋子佔據
if end_pos in board and
board[end_pos].startswith(get_player_prefix(current_player)):
    return False

piece_type = piece_name # simplified, piece_name itself is the type for now

if piece_type == "YF":
    # 陰府棋子移動規則：八方各一步
    col_diff = abs(start_col - end_col)
    row_diff = abs(start_row - end_row)
    if max(col_diff, row_diff) == 1: # 只能移動一步
        return True
elif piece_type in ["鼠", "牛", "虎", "兔", "龍", "蛇", "馬", "羊", "猴", "雞", "狗",
"豬"]:
    # 生肖棋子移動規則：直線或斜線各一步 (暫定規則)
    col_diff = abs(start_col - end_col)
    row_diff = abs(start_row - end_row)
    if max(col_diff, row_diff) == 1: # 只能移動一步
        return True

return False # 其他情況皆為不合法移動

def 列出可移動目錄(piece_name, start_pos, board, current_player):
    """ 列出指定棋子的所有合法移動目標位置 (更新版 - 加入編號) """
    possible_moves = []
    for col in ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']:
        for row in range(1, 9):
            end_pos = f"{col}{row}"
            if is_valid_move(piece_name, start_pos, end_pos, board,
current_player):
                possible_moves.append(end_pos)

    if possible_moves:

```

```

        print(f"{piece_name} (位於 {start_pos}) 的可移動位置：")
        for idx, move in enumerate(possible_moves): # 加入編號
            print(f"    {idx+1}. {move}")
    else:
        print(f"{piece_name} (位於 {start_pos}) 目前無合法移動位置。")

    return piece_name, start_pos, possible_moves # 返回 棋子代號, 起始位置,
    可移動位置列表

def 執行玩家指令(指令, board, current_player, last_listed=None): # 函數加入
last_listed 參數
    """ 執行玩家輸入的指令，並更新棋盤狀態 (更新版 - 支援數字選擇移動)
    """
    指令_parts = 指令.split()
    if not 指令_parts:
        print("無效指令，請重新輸入。")
        return board, last_listed # 返回更新後的 last_listed

    action = 指令_parts[0]

    if action == "help":
        顯示指令說明()
    elif action == "list":
        if len(指令_parts) != 2:
            print("指令格式錯誤，應為：list [棋子代號]")
            return board, last_listed
        piece_name = 指令_parts[1]
        piece_positions = [pos for pos, piece in board.items() if piece ==
f"{get_player_prefix(current_player)}{piece_name}"]
        if not piece_positions:
            print(f"未找到 {current_player} 的 {piece_name}，請確認棋子是
否存在。")
            return board, last_listed
        start_pos = piece_positions[0]
        last_listed_info = 列出可移動目錄(piece_name, start_pos, board,
current_player) # 取得包含 棋子代號, 起始位置, 可移動位置列表 的資訊
        return board, last_listed_info # 返回更新後的 last_listed_info

```

```

elif action == "move":
    if len(指令_parts) == 2 and 指令_parts[1].isdigit(): # 判斷是否為 "move
[數字]" 格式
        if not last_listed:
            print("請先使用 'list [棋子代號]' 查看可移動位置。")
            return board, last_listed
        move_idx = int(指令_parts[1]) - 1
        piece_name, start_pos, moves = last_listed # 從 last_listed 中取出
資訊
        if 0 <= move_idx < len(moves):
            end_pos = moves[move_idx]
            if is_valid_move(piece_name, start_pos, end_pos, board,
current_player): # 再次驗證移動的合法性
                piece = board.pop(start_pos)
                board[end_pos] = piece
                print(f"{current_player} 移動 {piece_name} 從
{start_pos} 到 {end_pos} (編號 {指令_parts[1]})")
            else:
                print("選擇的移動位置不合法，請重新選擇。") # 針對編
號選擇移動，加入合法性檢查提示
        else:
            print(f"無效的移動選項，請輸入 1 到 {len(moves)} 之間的
數字。")

    elif len(指令_parts) == 4 and 指令_parts[2] == "to": # 保留原有的 "move
[棋子代號] [起始位置] to [目標位置]" 格式
        piece_name = 指令_parts[1]
        start_pos = 指令_parts[2] # 原代碼這裡有錯誤，應使用 指令
_parts[1] 作為 piece_name, 指令_parts[3] 作為 end_pos
        end_pos = 指令_parts[3]
        if is_valid_move(piece_name, start_pos, end_pos, board,
current_player):
            if start_pos in board and
board[start_pos].startswith(get_player_prefix(current_player)):
                piece = board.pop(start_pos)
                board[end_pos] = piece
                print(f"{current_player} 移動 {piece_name} 從 {start_pos}
到 {end_pos}")
            else:

```

```

        print("起始位置沒有你的棋子或棋子不存在。")
    else:
        print("移動不合法，請重新輸入。")
    else:
        print("移動指令格式錯誤，應為：move [數字] 或 move [棋子代
        號] [起始位置] to [目標位置]")
    elif action == "attack":
        # ... (attack 指令處理部分保持不變) ...
        pass
    elif action == "end_turn":
        print(f"{current_player} 結束回合。")
        return "end_turn", last_listed # 返回 "end_turn" 標誌, 並傳遞
last_listed
    else:
        print("未知的指令，請輸入 'help' 查看可用指令。")

return board, last_listed # 返回更新後的棋盤狀態和 last_listed

```

```

def 遊戲主循環():
    """ 遊戲主循環，自動設置棋盤 (更新版 - 傳遞 last_listed) """
    board = {}
    player1_pieces = ["鼠", "牛", "虎", "兔", "龍", "蛇", "馬", "羊", "猴", "雞", "狗",
    "豬"]
    player2_pieces = ["鼠", "牛", "虎", "兔", "龍", "蛇", "馬", "羊", "猴", "雞", "狗",
    "豬"]
    cols = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H']

    for i, piece in enumerate(player1_pieces[:8]):
        board[f"{cols[i]}1"] = f"1_{piece}"
    for i, piece in enumerate(player1_pieces[8:]):
        board[f"{cols[i]}2"] = f"1_{piece}"
    board["D1"] = "1_YF"

    for i, piece in enumerate(player2_pieces[:8]):
        board[f"{cols[7-i]}8"] = f"2_{piece}"
    for i, piece in enumerate(player2_pieces[8:]):
        board[f"{cols[7-i]}7"] = f"2_{piece}"

```

```

board["D8"] = "2_YF"

current_player_num = 1
player_names = {1: "玩家 1", 2: "玩家 2"}
game_over = False
last_listed = None # 初始化 last_listed 為 None

while not game_over:
    current_player = player_names[current_player_num]
    顯示棋盤狀態(board)
    指令 = 取得玩家輸入(current_player)
    執行結果, last_listed = 執行玩家指令(指令, board, current_player,
last_listed) # 傳遞和接收 last_listed

    if 執行結果 == "end_turn":
        current_player_num = 3 - current_player_num
    elif 執行結果 == "game_over":
        game_over = True

print("遊戲結束！")

# ... (其餘程式碼部分保持不變) ...

if __name__ == "__main__":
    print("歡迎來到 陰府棋局：十二生肖的命運之戰 (簡化文字版)")
    print("遊戲開始！ 玩家 1 先行。\\n")
    顯示指令說明()
    遊戲主循環()

```