

```
<!DOCTYPE html>
```

```
<html lang="zh">
```

```
<head>
```

```
  <meta charset="UTF-8">
```

```
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
  <title>陰府棋局：十二生肖的命運之戰</title>
```

```
  <style>
```

```
    body {
```

```
      font-family: Arial, sans-serif;
```

```
      text-align: center;
```

```
    }
```

```
    #chessboard-table {
```

```
      border-collapse: collapse;
```

```
      margin: 20px auto;
```

```
    }
```

```
    #chessboard-table td {
```

```
      width: 50px;
```

```
      height: 50px;
```

```
      text-align: center;
```

```
        border: 1px solid black;

        font-size: 18px;

    }
```

```
#game-log {

    width: 80%;

    height: 200px;

    overflow-y: scroll;

    border: 1px solid black;

    padding: 10px;

    margin: 20px auto;

    text-align: left;

}
```

```
#input-area {

    margin: 20px;

}
```

```
#command-input {

    padding: 5px;

    width: 300px;

}
```

```
        button {

            padding: 5px 10px;

            margin-left: 10px;

        }

    </style>

</head>

<body>

    <h1>陰府棋局：十二生肖的命運之戰</h1>

    <p>玩家 1（藍色） vs AI 玩家（紅色） 遊戲開始！玩家 1 先行。</p>

    <table id="chessboard-table">

        <tr><td id="A8"></td><td id="B8"></td><td id="C8"></td><td
id="D8"></td><td id="E8"></td><td id="F8"></td><td id="G8"></td><td
id="H8"></td></tr>

        <tr><td id="A7"></td><td id="B7"></td><td id="C7"></td><td
id="D7"></td><td id="E7"></td><td id="F7"></td><td id="G7"></td><td
id="H7"></td></tr>

        <tr><td id="A6"></td><td id="B6"></td><td id="C6"></td><td
id="D6"></td><td id="E6"></td><td id="F6"></td><td id="G6"></td><td
id="H6"></td></tr>

        <tr><td id="A5"></td><td id="B5"></td><td id="C5"></td><td
id="D5"></td><td id="E5"></td><td id="F5"></td><td id="G5"></td><td
id="H5"></td></tr>

        <tr><td id="A4"></td><td id="B4"></td><td id="C4"></td><td
id="D4"></td><td id="E4"></td><td id="F4"></td><td id="G4"></td><td
```

```
id="H4"></td></tr>
```

```
    <tr><td id="A3"></td><td id="B3"></td><td id="C3"></td><td
id="D3"></td><td id="E3"></td><td id="F3"></td><td id="G3"></td><td
id="H3"></td></tr>
```

```
    <tr><td id="A2"></td><td id="B2"></td><td id="C2"></td><td
id="D2"></td><td id="E2"></td><td id="F2"></td><td id="G2"></td><td
id="H2"></td></tr>
```

```
    <tr><td id="A1"></td><td id="B1"></td><td id="C1"></td><td
id="D1"></td><td id="E1"></td><td id="F1"></td><td id="G1"></td><td
id="H1"></td></tr>
```

```
</table>
```

```
<div id="game-info">
```

```
    <p id="current-player">當前玩家： 玩家 1</p>
```

```
</div>
```

```
<div id="game-log"></div>
```

```
<div id="input-area">
```

```
    <input type="text" id="command-input" placeholder="輸入指令（例如：
move 鼠 A1 to B2）">
```

```
    <button onclick="submitCommand()">提交</button>
```

```
</div>
```

```
<script>
```

```
    let board = {};
```

```
let currentPlayer = "玩家 1";

let lastListed = null;

let aiScriptIndex = 0;

const aiScript = [

    "list 鼠", "move 1", "end_turn",

    "list 牛", "move 1", "end_turn",

    "list 虎", "move 1", "end_turn",

    "list 兔", "move 1", "end_turn",

    "list 龍", "move 1", "end_turn",

    "list 蛇", "move 1", "end_turn",

    "list 馬", "move 1", "end_turn",

    "list 羊", "move 1", "end_turn",

    "list 猴", "move 1", "end_turn",

    "list 雞", "move 1", "end_turn",

    "list 狗", "move 1", "end_turn",

    "list 豬", "move 1", "end_turn"

];

// 初始化棋盤
```

```
function initializeBoard() {

    const player1Pieces = ["鼠", "牛", "虎", "兔", "龍", "蛇", "馬", "羊", "猴", "雞", "狗", "豬"];

    const player2Pieces = ["鼠", "牛", "虎", "兔", "龍", "蛇", "馬", "羊", "猴", "雞", "狗", "豬"];

    const cols = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'];

    for (let i = 0; i < 8; i++) board[`${cols[i]}1`] = `1_${player1Pieces[i]}`;

    for (let i = 0; i < 4; i++) board[`${cols[i]}2`] = `1_${player1Pieces[i + 8]}`;

    board["D1"] = "1_YF";

    for (let i = 0; i < 8; i++) board[`${cols[7 - i]}8`] = `2_${player2Pieces[i]}`;

    for (let i = 0; i < 4; i++) board[`${cols[7 - i]}7`] = `2_${player2Pieces[i + 8]}`;

    board["D8"] = "2_YF";

    appendLog("棋盤初始化完成，棋子已自動擺放。");

    renderBoard();

}
```

```
// 渲染棋盤
```

```
function renderBoard() {  
  
    for (let row = 8; row >= 1; row--) {  
  
        for (let col = 'A'; col <= 'H'; col =  
String.fromCharCode(col.charCodeAt(0) + 1)) {  
  
            const pos = col + row;  
  
            const cell = document.getElementById(pos);  
  
            if (board[pos]) {  
  
                cell.textContent = board[pos].split('_')[1];  
  
                cell.style.color = board[pos].startsWith('1_') ? 'blue' :  
'red';  
  
            } else {  
  
                cell.textContent = "";  
  
            }  
  
        }  
  
    }  
  
}
```

```
// 添加日誌
```

```
function appendLog(message) {
```

```
const log = document.getElementById('game-log');

log.innerHTML += message + '<br>';

log.scrollTop = log.scrollHeight;

}
```

// 獲取玩家前綴

```
function getPlayerPrefix(playerName) {

    return playerName === "玩家 1" ? "1_" : "2_";

}
```

// 位置轉座標

```
function posToCoords(pos) {

    const col = pos.charCodeAt(0) - 'A'.charCodeAt(0);

    const row = parseInt(pos[1]) - 1;

    return [col, row];

}
```

// 檢查移動是否合法

```
function isValidMove(pieceName, startPos, endPos, board, currentPlayer) {
```



```
const [startCol, startRow] = posToCoords(startPos);
```

```
const [endCol, endRow] = posToCoords(endPos);
```

```
if (endPos in board &&  
board[endPos].startsWith(getPlayerPrefix(currentPlayer))) return false;
```

```
const pieceType = pieceName.split('_')[1];
```

```
const colDiff = Math.abs(endCol - startCol);
```

```
const rowDiff = Math.abs(endRow - startRow);
```

```
if (pieceType === "鼠") {
```

```
    return (colDiff <= 2 && rowDiff === 0) || (colDiff === 0 &&  
rowDiff <= 2) || (colDiff === rowDiff && colDiff <= 2 && colDiff !== 0);
```

```
    } else if (pieceType === "牛") {
```

```
        return (colDiff === 0 && rowDiff > 0) || (rowDiff === 0 && colDiff  
> 0);
```

```
    } else if (pieceType === "虎") {
```

```
        return (colDiff === 2 && rowDiff === 1) || (colDiff === 1 &&  
rowDiff === 2);
```

```
    } else if (pieceType === "兔") {
```

```
        return (colDiff <= 3 && rowDiff === 0) || (colDiff === 0 &&  
rowDiff <= 3) || (colDiff === rowDiff && colDiff <= 3 && colDiff !== 0);
```

```
    } else if (pieceType === "龍") {

        return (colDiff <= 3 && rowDiff <= 3) && (colDiff !== 0 ||
rowDiff !== 0);

    } else if (pieceType === "蛇") {

        return (colDiff <= 2 && rowDiff === 0) || (colDiff === 0 &&
rowDiff <= 2);

    } else if (pieceType === "馬") {

        return (colDiff <= 3 && rowDiff === 0) || (colDiff === 0 &&
rowDiff <= 3);

    } else if (pieceType === "羊" || pieceType === "猴" || pieceType === "
狗" || pieceType === "YF") {

        return (colDiff <= 1 && rowDiff <= 1) && (colDiff !== 0 ||
rowDiff !== 0);

    } else if (pieceType === "雞") {

        return (colDiff <= 2 && rowDiff === 0) || (colDiff === 0 &&
rowDiff <= 2);

    } else if (pieceType === "豬") {

        return (colDiff <= 1 && rowDiff === 0) || (colDiff === 0 &&
rowDiff <= 1);

    }

    return false;

}
```

```
// 檢查攻擊是否合法
```

```
function isValidAttack(attackerPieceName, targetPos, board, currentPlayer)
{
```

```
    if (!(targetPos in board) ||
board[targetPos].startsWith(getPlayerPrefix(currentPlayer))) return false;
```

```
    const attackerPosList = Object.entries(board).filter(([pos, piece]) =>
piece === attackerPieceName).map(([pos]) => pos);
```

```
    if (!attackerPosList.length) return false;
```

```
    const attackerPos = attackerPosList[0];
```

```
    const [startCol, startRow] = posToCoords(attackerPos);
```

```
    const [endCol, endRow] = posToCoords(targetPos);
```

```
    const colDiff = endCol - startCol;
```

```
    const rowDiff = endRow - startRow;
```

```
    const absColDiff = Math.abs(colDiff);
```

```
    const absRowDiff = Math.abs(rowDiff);
```

```
    const attackerPieceType = attackerPieceName.split('_')[1];
```

```
    if (attackerPieceType === "鼠") {
```

```

        return colDiff in [1, -1] && rowDiff === 1;

        } else if (attackerPieceType === "牛" || attackerPieceType === "兔" ||
attackerPieceType === "馬" ||

        attackerPieceType === "羊" || attackerPieceType ===
"YF") {

        return false;

        } else if (attackerPieceType === "虎") {

        return (absColDiff === 2 && absRowDiff === 1) || (absColDiff ===
1 && absRowDiff === 2);

        } else if (attackerPieceType === "龍" || attackerPieceType === "蛇" ||
attackerPieceType === "猴" ||

        attackerPieceType === "狗" || attackerPieceType === "
豬") {

        return (absColDiff <= 1 && absRowDiff <= 1) && (absColDiff !== 0
|| absRowDiff !== 0);

        } else if (attackerPieceType === "雞") {

        return colDiff === 0 && rowDiff === 1;

        }

        return false;

    }

    // 提交指令

```

```
function submitCommand() {  
  
    const input = document.getElementById('command-  
input').value.trim().toLowerCase();  
  
    executeCommand(input, currentPlayer);  
  
    document.getElementById('command-input').value = "";  
  
    renderBoard();  
  
}
```

// 執行指令

```
function executeCommand(command, player) {  
  
    appendLog(`${player} 輸入指令: ${command}`);  
  
    const parts = command.split(' ');
```

```
    if (parts[0] === "help") {
```

```
        appendLog(`
```

--- 指令說明 ---

list [棋子代號] - 列出指定棋子的可移動位置並編號 (例如: list 鼠)

move [數字] - 選擇編號移動位置 (需先使用 list，例如: move 1)

move [棋子代號] [起始位置] to [目標位置] - 移動棋子 (例如: move 鼠 A1 to B2)

attack [棋子代號] [目標位置] - 攻擊目標位置 (例如: **attack** 牛 C3)

end_turn - 結束當前回合

help - 顯示指令說明

--- 請注意 ---

* 棋子代號：鼠, 牛, 虎, 兔, 龍, 蛇, 馬, 羊, 猴, 雞, 狗, 豬, YF (陰府)

* 位置：西洋棋盤座標，例如 A1, H8, C5`);

```
    } else if (parts[0] === "list" && parts.length === 2) {

        const pieceName = getPlayerPrefix(player) + parts[1];

        const piecePositions = Object.keys(board).filter(pos =>
board[pos] === pieceName);

        if (!piecePositions.length) {

            appendLog(`未找到 ${player} 的 ${parts[1]}`);

            return;

        }

        const startPos = piecePositions[0];

        const possibleMoves = [];

        for (let col = 'A'; col <= 'H'; col =
String.fromCharCode(col.charCodeAt(0) + 1)) {

            for (let row = 1; row <= 8; row++) {

                const endPos = col + row;
```

```

        if (isValidMove(pieceName, startPos, endPos, board,
player)) {

            possibleMoves.push(endPos);

        }

    }

}

if (possibleMoves.length) {

    let message = `${parts[1]} (位於 ${startPos}) 的可移動位
置 : <br>`;

    possibleMoves.forEach((move, idx) => message += `  ${idx +
1}. ${move}<br>`);

    appendLog(message);

    lastListed = { pieceName, startPos, moves: possibleMoves };

} else {

    appendLog(`${parts[1]} (位於 ${startPos}) 目前無合法移
動位置。`);

}

} else if (parts[0] === "move") {

    if (parts.length === 2 && !isNaN(parts[1])) {

        if (!lastListed) {

            appendLog("請先使用 'list [棋子代號]' 查看可移動
位置。");

```

```

        return;

    }

    const moveldx = parseInt(parts[1]) - 1;

    const { pieceName, startPos, moves } = lastListed;

    if (moveldx >= 0 && moveldx < moves.length) {

        const endPos = moves[moveldx];

        if (isValidMove(pieceName, startPos, endPos, board,
player)) {

            board[endPos] = board[startPos];

            delete board[startPos];

            appendLog(`${player} 移動
${pieceName.split('_')[1]} 從 ${startPos} 到 ${endPos}`);

        }

        } else {

            appendLog(`無效的移動選項，請輸入 1 到
${moves.length} 之間的數字。`);

        }

    } else if (parts.length === 5 && parts[3] === "to") {

        const pieceName = getPlayerPrefix(player) + parts[1];

        const startPos = parts[2].toUpperCase();

```



```
        const endPos = parts[4].toUpperCase();

        if (isValidMove(pieceName, startPos, endPos, board, player)
    && board[startPos] === pieceName) {

            board[endPos] = board[startPos];

            delete board[startPos];

            appendLog(`${player} 移動 ${parts[1]} 從 ${startPos}
到 ${endPos}`);

        } else {

            appendLog("移動不合法或棋子不存在。");

        }

    } else {

        appendLog("移動指令格式錯誤。");

    }

} else if (parts[0] === "attack" && parts.length === 3) {

    const attackerPieceName = getPlayerPrefix(player) + parts[1];

    const targetPos = parts[2].toUpperCase();

    if (isValidAttack(attackerPieceName, targetPos, board, player)) {

        const defeatedPiece = board[targetPos];

        delete board[targetPos];

        appendLog(`${player} 使用 ${parts[1]} 攻擊 ${targetPos}
並擊敗 ${defeatedPiece} !`);
```

```
    } else {

        appendLog("攻擊不合法。");

    }

} else if (parts[0] === "end_turn") {

    appendLog(`${player} 結束回合。`);

    currentPlayer = currentPlayer === "玩家 1" ? "AI 玩家" : "玩家
1";

    document.getElementById('current-player').textContent = `當前
玩家：${currentPlayer}`;

    if (currentPlayer === "AI 玩家") {

        while (aiScriptIndex < aiScript.length) {

            const aiCommand = aiScript[aiScriptIndex++];

            executeCommand(aiCommand, "AI 玩家");

            renderBoard();

            if (aiCommand === "end_turn") break;

        }

    }

} else {

    appendLog("未知指令，請輸入 'help' 查看說明。");

}
```

```
}
```

```
// 初始化遊戲
```

```
initializeBoard();
```

```
appendLog("歡迎來到 陰府棋局：十二生肖的命運之戰 (網頁版)");
```

```
</script>
```

```
</body>
```

```
</html>
```