Network Programming HW1
蔡懷恩 112550020

1. System Architecture

系統由四個元件組成：
1. Lobby Server：使用 TCP，負責帳號註冊、登入、登出、遊戲結果儲存等功能。
2. Player A：發送邀請的先手玩家。UDP 掃描並發送邀請，對方接受後再開 TCP server 供連線。
3. Player B：等待邀請的後手玩家，監聽 UDP 回傳等待，等之後接收到 TCP 連線後進入遊戲。
4. DB（JSON 檔模擬）：儲存帳號資訊與勝負場數等統計。

2. Communication Details

通訊協定使用 JSON 格式。以下為具體範例：

1. [註冊] Player → Lobby Server (TCP)

```
{
   "action": "register",
   "username": "username",
   "password": "password"
}
```

回應：

成功
```
{
   "status": "REGISTER_SUCCESS"
}
```

名稱重複
```
{
   "status": "REGISTER_FAIL", "reason": "User exists"
}
```

2. [登入] Player → Lobby Server (TCP)

```
{
   "action": "login",
   "username": "username",
   "password": "password"
}
```

回應:

成功

```
{
   "status": "LOGIN_SUCCESS",
   "win": 2,
   "draw": 1,
   "lose": 3
}
```

找不到

```
{
   "status": "LOGIN_FAIL", "reason": "Wrong credentials"
}
```

已登入

```
{
   "status": "LOGIN_FAIL", "reason": "User already logged in. Please logout first"
}
```

3. [看戰績] Player → Lobby Server (TCP)

```
{
   "action": "get_stats",
   "username": "username",
   "password": ""
}
```

回應:

```
{
   "status": "STATS",
   "win": 2,
   "draw": 1,
   "lose": 3
}
```

4. [掃描] Player A → Player B (UDP)

```
{
   "action": "scan"
}
```

回應:
"WAITING"

5. [邀請] Player A → Player B (UDP)

```
{
   "action": "invite",
   "from": "PlayerA username"
}
```

回應:

接受
```
{
   "status": "ACCEPT", "username": "PlayerB username"
}
```

拒絕
```
{
   "status": "REJECT"
}
```

6. [遊戲中的下棋動作] Player A/B → 對方 (TCP)

```
{
   "action": "move",
   "row": 1,
   "col": 2
}
```

7. [更新戰績] 最後下棋的一方 → Lobby Server (TCP)

```
{
   "action": "update_stats",
   "updates": [
      {"username": "username", "stats": {"win": 1, "draw": 0, "lose": 0}},
      {"username": "opponent_username", "stats": {"win": 0, "draw": 0, "lose": 1}}
   ]
}
```

回應:

成功

```
{
   "status": "UPDATE_SUCCESS"
}
```

一方成功

```
{
   "status": "PARTIAL_SUCCESS",
   "failed": ["b"]
}
```

8. [登出] Player → Lobby Server (TCP)

```
{
   "action": "logout",
   "username": "username",
   "password": ""
}
```

回應:

成功

```
{
   "status": "LOGOUT_SUCCESS"
}
```

失敗

```
{
   "status": "FAIL", "reason": "User not found"
}
```

3. The game play

本作業實作的是一個線上的「井字棋（**Tic-Tac-Toe**）」雙人對戰遊戲。

遊戲規則如下：
- 玩家 A 為先手，下 "X"；玩家 B 為後手，下 "O"
- 雙方輪流下棋，共有 3x3 的九宮格
- 任一玩家橫排、直排或對角線達成三子連線即獲勝
- 若九格皆填滿仍無勝負則為平手

遊戲流程：
1. 玩家登入後選擇角色（A or B）
2. A 掃描 UDP 範圍尋找 B，並發送邀請
3. 若 B 接受，雙方建立 TCP 連線並開始對戰
4. 每回合交換 "move" 指令與棋盤狀態
5. 結束時由 最後下棋的一方 回報勝負至 Lobby Server
6. 雙方回到選角畫面，可進行下一場或登出