

多人骰子吹牛遊戲連線實作

N26130891 林以諾

(一) 引言

本專案設計了一個基於使用 socket 網際連線的骰子吹牛遊戲，模擬現實中的多人桌遊，目標實現多人即時通信的遊戲機制，包括骰子生成、喊話機制、回合切換、遊戲進行判斷、UI 介面等功能。

遊戲規則說明：

1. 玩家數量與骰子分配：
遊戲需 3 名玩家此外每位配 5 顆骰子，骰子隨機生成且僅玩家自己可見。
2. 遊戲流程：
 - 玩家依序輪流喊話，喊話內容為「X 個 Y」，意指場上所有骰子中至少有 X 顆點數為 Y（1 為通配符，可視為任何點數）。
 - 喊話需逐步提高：X 或 Y 必須大於或等於上一位玩家的喊話。
3. 抓謊機制：
 - 玩家除了喊話外可選擇「抓謊」，但只能針對上一位玩家。
 - 若上一位玩家的喊話大於場上實際的骰子數量，則抓謊者獲勝；反之，抓謊失敗，上一位玩家獲勝。
4. 遊戲結束：
當有玩家抓他人說謊則遊戲結束，計算場上總骰子數量，判斷勝負結果。

(二) 系統架構與設計

2.1 系統架構

本系統採用 Client-Server 架構，實現三人即時互動的骰子吹牛遊戲，設計的主要模組包括：

a. 伺服器端 (server.py)

負責管理玩家連線、分發骰子、處理遊戲邏輯（喊話、抓謊）、廣播訊息及判定勝負。

主要模組與功能：

- `game_loop`：控制遊戲主邏輯，負責處理每位玩家的行動並更新遊戲狀態。
- `handle_action`：解析玩家的指令，執行喊話與抓謊的邏輯判斷。
- `count_dice`：統計所有骰子的點數分布，用於判定抓謊的結果。
- `cleanup`：清理連線資源，關閉伺服器與客戶端連線，結束遊戲。

b. 客戶端 (`client.py`)

提供玩家與伺服器的互動介面，顯示骰子點數與遊戲進度，並將玩家的行動傳遞至伺服器。

主要模組與功能：

- `receive_messages`：持續接收伺服器的廣播訊息，更新玩家的介面內容。
- `update_dice_images`：解析伺服器發送的骰子數據，動態更新骰子圖片。
- `send_action`：將玩家輸入的行動指令傳送至伺服器。
- `Tkinter` 介面：包含滾動訊息框、骰子圖片顯示框及行動輸入框，提供用戶友好的操作體驗。

2.2 通訊設計

- 通訊協議：基於 `TCP` 協議，確保數據傳輸的可靠性與順序性。伺服器監聽固定端口，接受來自三個客戶端的連線。
- 數據交互：客戶端通過指令形式發送玩家的行動至伺服器，伺服器處理後以廣播形式向所有玩家發送遊戲狀態更新。

2.3 遊戲邏輯與程式實現

a. 骰子生成與喊話規則：

- 在遊戲開始時，伺服器為每位玩家隨機生成 5 顆骰子。骰子點數透過 `generate_dice` 函數隨機產生，並發送至各自的客戶端進行顯示。
- 玩家喊話的指令格式為「數量 + 空格 + 點數」，如「3 5」，表示場上至少有 3 顆 5。喊話需符合遞增規則。

b. 抓謊判定：

- 當玩家選擇「抓謊」時，伺服器使用 `count_dice` 函數統計場上所有骰子的點數分布，並與上一位玩家的喊話進行比較，判定是否說謊。
- 抓謊結果將通過 `broadcast_message` 廣播給所有玩家，並結束遊戲。

c. 回合切換：

- 遊戲以循環方式切換回合，由 `current_player` 變數記錄當前玩家序號，並由伺服器通知下一位玩家行動。

d. 遊戲結束：

- 遊戲結束時，伺服器使用 `cleanup` 函數清理所有客戶端連線，並向玩家廣播最終結果。

（三） 結果展示與總結



左圖為 server 端，右圖為 client 端（其中一位玩家的介面）

遊戲操作：啟動 server 後等待三位玩家分別啟動 client 連接伺服器，此外也提供 UI 介面使玩家能更直觀的遊玩，系統也會根據回合判斷輸入框是否啟用。

總結來說，此骰子吹牛遊戲專案成功實現了多人即時互動的遊戲系統，結合了基於 TCP 協議的通訊技術和直觀的圖形化介面設計，提供了穩定且流暢的遊戲體驗。

伺服器負責處理遊戲邏輯、廣播消息以及玩家連線管理，確保每一步操作都能準確地同步到所有玩家，而客戶端則提供了簡潔明瞭的介面，讓玩家能輕鬆查看自己的骰子點數、遊戲進度以及進行操作。