

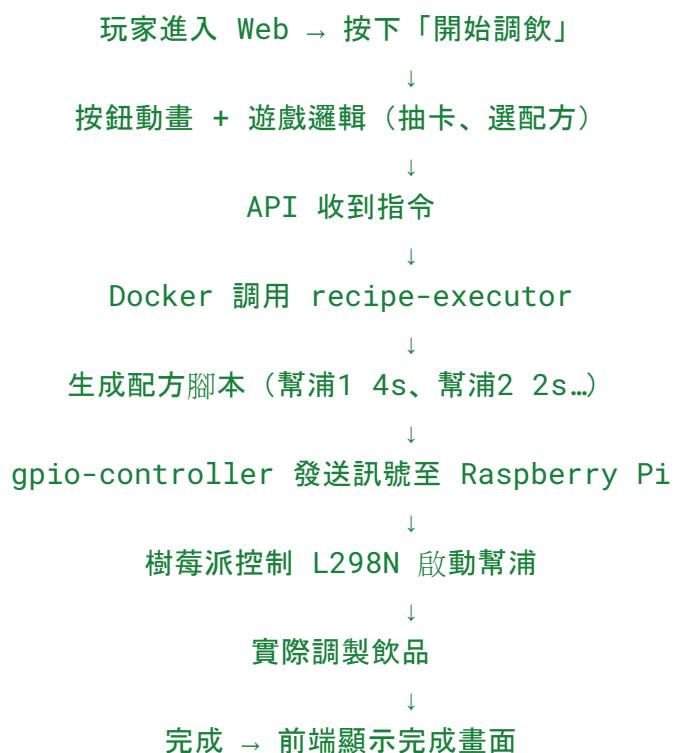
# 初步想法

### <連線調酒桌遊>

- 為什麼要做這個主題？韓國遊戲有個喝酒的傳統，我們想利用這個調酒機增加遊戲的趣味性。
- 跟LSA的關聯：一個系統化、nginx、API、Docker 服務

### Linux 飲料機專題方案：

- 專題概念  
打造一台「調酒飲料機」，用 Linux 控制整個流程。用戶透過網頁畫面去操作控制，調配飲料。
- 控制方式  
利用網頁畫面去控制。傳訊號給樹莓派，來控制幫浦流量。
- 系統架構



### 流程

1. 每個 client 可以連進 server。開始進入遊戲進行調酒配方。
2. API 接收指令，查詢配方。
3. Docker 容器執行配方腳本。
4. 控制幫浦流量完成飲料。
5. 完成時，畫面顯示完成(可能可以寄信)

- 硬體設備

**Raspberry Pi 4 / Zero 2** 系統核心、控制 GPIO

**L298N 馬達驅動板** 控制幫浦(雙 H 橋輸出)

**3–6V 微型幫浦** 注入不同飲料

**工業電源 (Mean Well)** 穩定供電 5V / 12V

- Linux 技術整合

會建立一個系統，有分成client端與server端，經由client端連進去我們的server  
( NGINX反向代理 Web UI/API)

當我們要調酒的時候

技術用法 Docker每個飲料配方是一個容器，快速切換不同飲料

提供 HTTPS 與權限控制 iptables / firewall防止未授權操作

- 滴定控制

每個幫浦出水量固定: 1 秒 = 8 ml。Linux 控制毫秒級開關。例: 12 ml → 開 1.5 秒。可加「慢速滴定模式」呈現化學反應感。

- 實作步驟

硬體組裝: 幫浦/閥門固定在一個箱子上。

Linux 系統準備: 安裝 Raspberry Pi OS、Docker、NGINX、Python Flask。

GPIO 控制程式: 控制幫浦開關。Web UI 開發: 飲料選單、配方滑桿、開始停止按鈕。

整合測試: 校正幫浦流量、配方執行順序。

架一個server

-----  
多人協作->在網頁上操作

個人操作->指定手勢來操作

-----  
如果, 暗黑()

和BT meeting後

接螢幕  
建立一個手機當作觸控螢幕  
ipad可以綁在包裝好的  
上面跑個網頁  
我們的設備會有一個互動環節

## <<遊戲邏輯初步規劃>>

酒會遊戲  
簡單的遊戲  
書的人會要喝下那杯酒  
兩個人的互動遊戲  
分數差來決定酒精濃度  
會有懲罰緩節

## 遊戲邏輯：

哪些過程在機器上做的

玩家會進入一個遊戲畫面

會有一個喝酒流程

遊戲要加深背景

先選人數、骰色子，最贏\的幫最輸的調酒

騙子酒館，一個人會發五張牌，會發二十張，jka2Q，這回合只能喝  
吹牛

遊戲邏輯!!!類似這種元素，就可以進入一個調酒環節

遊戲拖久一點再喝

結合遊戲，就可以派上用場  
我們要設遊戲，可以用nginx來幫忙

先做一個簡單的遊戲，在滾動的時間(要常要短)

可以再多做兩三個功能

調整酒精濃度  
酒要加多少  
這類細節我們可以自己決定

## <<系統部屬>>

情境不是站在機器前  
機器在桌子旁邊  
平板放在桌上  
五個人中間放平板  
先輸入幾個人  
可以輪流點一下

主機跟設備分開

用網頁的方式去聯  
在佈署方面可以加入docker

將決定主機在另一台  
調酒機也是用網路連線那台主機  
設備也是連到那台主機  
才要用reverse proxy(用上)  
(與linux)

兩個device  
client: 一個ipad,  
server: 除了做網頁之外，也身兼調酒器的controller

client 透過網路連進server  
controller可以透過網路連進server

那這樣就可以同時有很多使用者進入我們的伺服器

這時候才會需要我們的reverse proxy

小插曲：  
買一棟房子喝一杯

現在只有一個遊戲，客人會覺得很無聊  
可以增加遊戲功能

(買一個電路板)

## 我們要多注意的點

hardware enabling 需要一點功夫  
同時要接很多科抽水馬達  
要看怎麼控制

遊戲跟酒結合  
十點半  
發一張牌，大家會覺得自己是否會贏  
抽種十種



# 《互動式智慧調酒伺服系統》專題計畫書(重寫專業版)

# 🎯 《互動式智慧調酒伺服系統》專題計畫書 (重寫專業版)

---

## 1 專題背景與動機(升級版)

韓國娛樂節目與遊戲文化裡常見「懲罰酒」、「調酒賭局」等玩法，參與者需要依照遊戲結果喝下由大家選的特調。

我們希望把這種 *high* 氣氛 做得更科技、更安全、更好玩——

- 把原本需要「人手調酒」的遊戲，改成由一台智慧飲料調製機完成。
- 透過自動控制幫浦、飲料比例與玩法流程，提升遊戲效果。

而且我們不是單純做飲料機，我們要做的是：

「一個可多人連線、可定義配方、可自動執行的雲端式智慧調飲平台」

玩家可以透過網頁進入同一個遊戲房間，由伺服器依「遊戲結果」自動叫出配方，再控制硬體完成飲料。

這不只好玩，也展示：

- 全端系統整合
- API 設計
- Docker 架構
- Linux 控制硬體
- 多人協作式互動網站

同時非常符合 LSA(Linux Security & Application) 的精神：  
「用 Linux 架構可信賴的自動化服務平台」

---

## 2 專案與 LSA 課程的關聯(增強版)

這個專案完整落實 LSA 的核心概念：

### (1) Server 建置(Linux)

- 建立一台後端服務伺服器
- 使用 NGINX 作 reverse proxy
- 導入 HTTPS 與 API 驗證

## (2) Docker 化

飲料配方、控制程式、WebServer 全部容器化：

- 環境可重複
- 安全隔離
- 易於多人協作

## (3) API 設計

- Client / Game UI → Server API
- Server API → 指令派送 → 控制幫浦 → 完成飲料

## (4) 安全控制

- iptables 防止未授權存取
- Docker network 隔離
- Web Token 驗證

## (5) 使用者介面

- 多人同時進入網站遊戲
- 伺服器派送配方
- 裝置端顯示狀態、回饋結果

► 總結：

這個專案是“完整的 Linux-based IoT 自動化系統”，非常符合 LSA 技術重點。

---

## 3 系統架構設計



流程細節：

1. 使用者進入網頁 → 選遊戲 → 系統抽配方
2. API 收到配方 → 查詢資料庫
3. Docker 容器負責「執行配方腳本」
4. Raspberry Pi GPIO 控制幫浦出水
5. L298N 控制電機或幫浦開關
6. 工業電源供應幫浦與主機
7. 網頁最後顯示完成(也可同步寄信通知、推播消息)

---

## 4 硬體架構總覽(含你照片中的實物)

### (1) 控制核心: Raspberry Pi

- 執行 Linux
- 控制 L298N
- 執行 Docker
- 提供 API server

## (2) 幫浦(你照片中的 370 微型氣泵 )

功能: 抽取飲料、控制注入量

- DC 3–6V
- 流量 < 1.5 L/min
- 由 L298N 控制開/關與方向

## (3) L298N 馬達驅動板

- 控制幫浦與其他 12V 裝置
- 支援雙通道 → 可控制多個幫浦

## (4) 工業電源(第二張照片)

- Meanwell LRS-50 系列
  - 提供 12V/5V 輸出
  - 供應給 Raspberry Pi + 幫浦 + L298N
- 

## 5 功能設計

### A. 多人遊戲模式 (Web 介面)

玩家可以透過手機連線同一伺服器:

- 投票調酒配方
- 隨機抽飲料
- 遊戲決定誰喝
- 自動啟動調酒機

### B. 個人控制模式

單人模式可以:

- 自選配方
- 手動調整比例
- 或利用「手勢控制」操作

## C. API 功能

- `/recipe/list`
- `/recipe/{id}`
- `/machine/start`
- `/machine/status`
- `/drink/make`

## D. Docker 容器負責執行配方

每一杯飲料是一組獨立腳本：

- 控幫浦 1:5 秒
- 控幫浦 2:3 秒
- 控 LED 閃爍
- 攪拌 2 秒

---

## 6 流程示意圖(重寫更專業)

Client (網頁)



Server (API)



配方管理 (DB)



Docker Recipe Executor



Raspberry Pi

↓

L298N

↓

幫浦運作

↓

完成 → 回傳狀態 → 網頁顯示完成（可寄 email）

---

## ⑦ 若要加入「暗黑模式」(你提到的)

你可以把它做成：

- 暗黑飲料(隨機苦味、鹹味)
- 賭輸的人會被伺服器「惡搞配方」
- 伺服器加密該配方 → 限定一次使用

會讓展示超有戲劇效果。

# ★《多人互動式雲端智慧調飲系統》

# ★《多人互動式雲端智慧調飲系統》

## 完整專題計畫書(系統版 / 重寫強化版)

---

### 1 專題背景與動機

在現代聚會、派對、韓國遊戲文化中，「調酒」常作為懲罰或遊戲的一環，而過程中最麻煩的就是——要有人負責調，而且常常調錯、量不準、有爭議，也不夠公平，不夠科技感。

我們希望改善這點，打造一個：

### ★ 讓玩家透過網頁進入遊戲，由系統自動調製飲品的雲端互動平台

所以我們的重點不是“飲料機”，而是：

\*\*✓ 一個可多人連線

- ✓ 有可視化互動畫面
  - ✓ 由伺服器自動調度配方
  - ✓ Raspberry Pi 控制幫浦實際調製
  - ✓ Docker + API + Linux 整合
- 的完整智慧調飲系統\*\*

此系統讓娛樂活動更公平、流程一致、具科技展示效果，並能擴充更多玩法。

---

### 2 系統主軸(本專題的核心價值 )

本專題旨在打造一套「多人互動式雲端智慧調飲系統」，讓使用者透過 Web UI 參與遊戲，由伺服器調度配方，並由 Raspberry Pi 控制硬體自動完成飲品調製。

系統強調：

- 前端互動性：即時顯示按鈕、動畫、調製狀態

- 後端可控性: API 管控流程, 保障安全
  - 多服務分工:Docker 微服務
  - 硬體執行可靠:L298N + 幫浦精準控制
  - 安全性:iptables + NGINX 反向代理
  - 可擴充性:日後可加遊戲、感測器、RFID、配方管理系統
- 

## 3 系統完整架構(技術面)

整體分為三大層:

---

### (一) 前端互動層:多人 Web UI

玩家可以用手機或電腦進入:

- 加入遊戲房間
- 查看配方選單
- 按下按鈕觸發事件
- 看到調製過程的動畫(進度條 / LED 模擬)
- 調製完成畫面

► 所有玩家同步看到相同畫面(**WebSocket**)

---

### (二) 伺服器後端:API + 遊戲邏輯 + NGINX

伺服器負責:

- 管理玩家、遊戲房間
- 接收前端指令

- 查詢配方
- 控制調飲流程
- 管理日誌、權限
- API Token 驗證
- 使用 NGINX 作反向代理與流量統一入口

### NGINX 代理三項服務：

URL	代理服務
-----	------

/	前端 Web UI
---	-----------

/api/	API Server
-------	------------

/recip e/	Recipe Executor(Docker)
--------------	-------------------------

---

### (三) 後端微服務層 (Docker 架構)

每個功能模組獨立運作，不互相干擾：

- `web-frontend`: 前端檔案伺服器
- `api-service`: 後端 API(Flask/FastAPI/Node)
- `recipe-executor`: 配方執行器，負責生成「幫浦時序」
- `gpio-controller`: 最終控制 Raspberry Pi GPIO
- `db`: 存配方、玩家紀錄

→ Server → Docker → Pi → L298N → 幫浦

---

### (四) 硬體執行層 (Raspberry Pi)

樹莓派負責：

- 接收 Docker 命令
- 控制 GPIO
- 透過 L298N 啟動幫浦
- 控制流量(例如 1 秒  $\approx 8mL$ )
- 控制 LED、攪拌馬達
- 接收工業電源供電(穩定)

Raspberry Pi 是整個硬體端的「控制中心」。

---

## 4 使用硬體設備(清單)

硬體	功能
Raspberry Pi 4 / Zero 2	系統核心、控制 GPIO
L298N 馬達驅動板	控制幫浦(雙 H 橋輸出)
3–6V 微型幫浦	注入不同飲料
工業電源(Mean Well)	穩定供電 5V / 12V
LED 燈條 / RGB	顯示調製演出效果
小馬達(攪拌)	混合飲料
壓克力或 3D 外殼	硬體機構

---

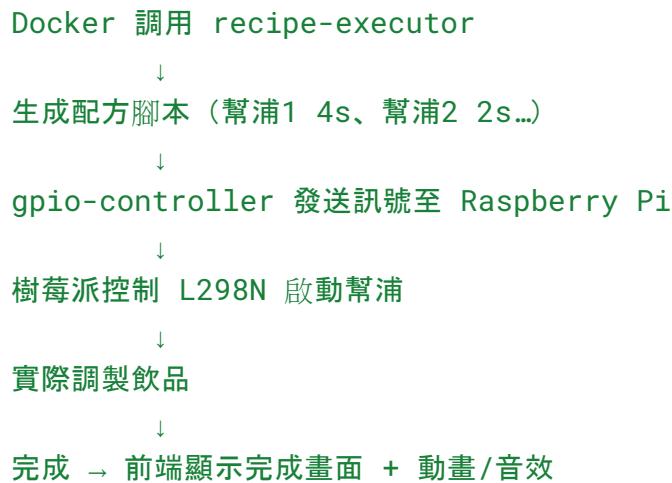
## 5 系統流程設計(玩家角度)

玩家進入 Web → 按下「開始調飲」

↓  
按鈕動畫 + 遊戲邏輯(抽卡、選配方)

↓  
API 收到指令(/api/drink/start)

↓



## 6 安全性設計

### **iptables:**

- 限制 API port 只能被 Docker 或內網呼叫
- 封鎖所有未使用 port
- 防止 DoS(流量限制)
- SSH 防暴力登入

### **NGINX:**

- 全部流量統一入口
  - 防止直接訪問內部服務
  - 可加 HTTPS
  - API 驗證
- 

## 7 與 LSA 課程的關聯

本專題完整應用了 LSA 所教授的技術：

- Linux 伺服器架設
- Docker 容器化部署
- NGINX 反向代理
- iptables 防火牆
- API 設計
- 系統安全
- 自動化控制

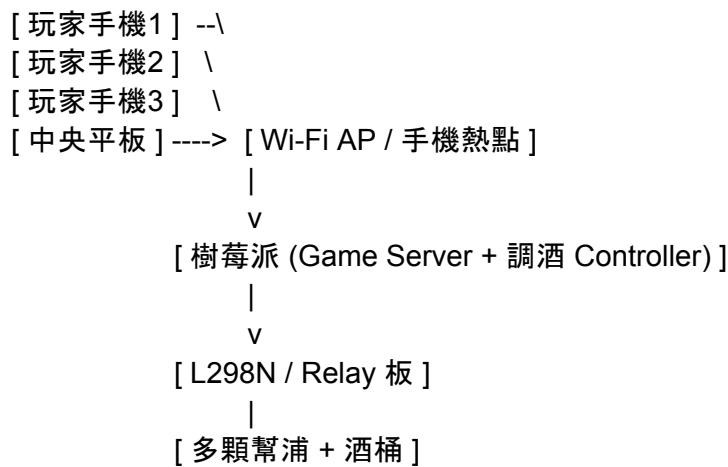
這是一套真正的「系統工程專題」，不只是硬體展示。

---

## 8 預期成果

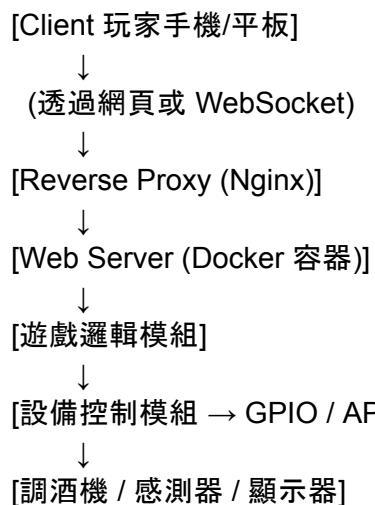
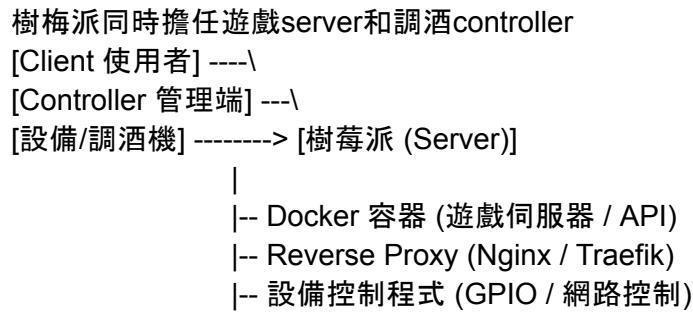
- 一套能容納多人同時操作的智慧飲品系統
- 一個可視化、即時同步的調飲網頁
- 一台由 Raspberry Pi 驅動的智慧飲品調製中心
- 完整的 Linux 後端 + 微服務架構
- 安全性完整的雲端系統

# 系統架構



說法可以是：

- 所有參與者(台下被選到的 4–6 人)用自己手機或中央平板連到同一個 Wi-Fi。
- 樹莓派連到同一個 Wi-Fi, 跑 Web 服務與遊戲後端。
- 手機 / 平板透過瀏覽器連到樹莓派的 IP(例如 <http://192.168.x.x>), 進入「醉加損友、喝不shot了」遊戲頁面。
- 樹莓派同時負責：
  - 管理遊戲邏輯、積分、模式。
  - 控制 L298N + 幫浦, 實際把酒打出來。



## 為什麼需要 Reverse Proxy

- **Client** 透過網路連進 **Server**  
→ 使用者端(Client)會發送請求到伺服器。
- **Controller** 可以透過網路連進 **Server**  
→ 控制端(可能是管理介面或後台)也會透過網路進入伺服器。
- 同時有很多使用者進入伺服器  
→ 當多人同時使用系統，伺服器需要能夠處理並分配流量。
- 這時候才會需要 **Reverse Proxy**  
→ Reverse Proxy 的作用：
- 接收所有外部請求，然後分配到正確的服務或容器。
- 提供負載平衡(Load Balancing)，避免單一服務過載。
- 增加安全性(隱藏內部伺服器 IP、提供 SSL/TLS)。
- 統一入口(Gateway)，讓 Client 不需要知道後端的複雜結構。

遊戲server與調酒controller機器分開

[Client 使用者] ---\

[Controller 管理端] ---\

\

---> [樹莓派 A (Server)]

|

|-- Docker 容器 (遊戲伺服器 / API)

|-- Reverse Proxy (Nginx / Traefik)

[設備/調酒機] -----> [樹莓派 B (Device Controller)]

|

|-- 設備控制程式 (GPIO / 網路控制)

# 遊戲邏輯

# 醉加損友、喝不shot了

遊戲設計 >> 與調酒機搭配後，可以根據遊戲邏輯，增加酒精濃度

server放在樹梅派

連線方式：可以讓多個設備可以連進server。

調酒機：

1. 連線到遊戲server
2. 與遊戲server

我們找人：隨機在台下找人(四~六)

## 多模式版本：(有兩個模式)

系統功能：抽籤系統、五回制、多人版、積分版

### 遊戲流程

#### 1、玩家進入遊戲前，先選擇要哪種模式

模式	規則	
無乃酒鬼模式	不要有積分制	答錯直接喝
闔家歡模式	積分	5回合結算一次

(1) 酒鬼模式：主要以遊戲的規則決定喝酒，沒有積分機制，只要輸了就要喝

(2) 闔家歡模式：由遊戲規則與積分，決定怎麼分這杯酒

每 5 回合結算一次積分，結束後：

積分最少的喝

積分最多的可以再加一種酒(或遊戲的方式)，給輸家喝酒

🔥 (可以做成小遊戲 EX: 按5秒來翻牛排，前面會提示1-3秒來讓人心算，後面秒數隱藏，如果積分一樣就系統隨機抽)

回合結束後如果積分一樣，贏的各加一種酒、輸的平分酒(交杯酒?)

積分計算：答對+1 / 答錯-1 / 沒抽到 0

#### 2、設置人數

選擇完模式後，選擇輸入人數，開始轉轉盤，決定哪位玩家，選擇酒的基底。

(遊戲方式: 每個人利用感應偵測或點擊平板去擲骰子)

### 3、在遊戲真正開局前，先要選一杯酒當基底、固定容量

系統會出現一個手動選或系統自動轉輪盤的方式，決定要由誰決定第一杯酒要選甚麼。

(也就是在平板上，會有一個選擇介面，可以控制我們的調酒機，加入到我們的杯子中)

### 4、遊戲邏輯

使用兩顆6面骰子，如果骰的數值加起來是：

#### (1)酒鬼模式

骰子相加的點數	種類	執行的是
三	12	電腦隨機指定一人跟骰到的人玩我有你沒有遊戲(其中一人要喝)
四	13	電腦隨機指定一種酒
五	14、23	實體掰手腕 電腦隨機指定
六	15、24	黑白切 (實際互動遊戲)  這個是只要抽到這個會有一個電腦隨機匹配畫面，匹配後就可以進行輸的喝
七	16、34、25	自己選酒倒
八	26、35	電腦隨機指定一種酒倒
九	36、45	射龍門(撲克牌1~13自己先抽兩張牌當龍柱，第三張牌介於前兩張牌點數之間)
對數	11、22、33、44、55、66	喝喝
十、十一	46、65	真心話大冒險

(2) 開家歡模式遊戲邏輯

骰子相加的點數	種類	執行的是
三	12	LSA 問答(選擇題) 會出現在使用者的手機 讓使用者去回答 回答成功: 積分+1 回答失敗: 積分+
四	13	電腦隨機指定一種酒
五	14、23	LSA 問答(選擇題) 回答成功: 積分+1 回答失敗: 積分+
六	15、24	黑白切 (實際互動遊戲)  這個是只要抽到這個會有一個匹配畫面，匹配後就可以進行 輸了: -1 贏了:+1
七	16、34、25	自己選酒倒 系統會出現酒的選項，可以自己調整
八	26、35	電腦隨機指定一種酒倒
九&對數	36、45 11、22、33、44、55、66	喝喝 因為太慘了，積分 +1
十、十一	46、65	真心話大冒險 回答結束後，會有一個結束的按鈕

(待考慮)

一個模式: 喝系統特調酒(直接喝)

最終積分的輸家，如果積分扣太多，可能會有加倍除罰

例如:- 5 分 就是正常

-10分~ -15 分 : 加倍處罰

可以想想要不要加入此功能~!

## 有空餘時間:

---

二十一點

ABCD輪流當莊家 ([系統去做決定](#))

Round1 A莊家 : 骰三顆骰子(相加是會爆掉的數字) + 倒酒

假設目標數字是21: 若之後出現超過21的數字，就會爆掉

BCD輪流進行選擇:

1.抽牌加注: 從牌庫抽一張牌(A=1, JQK=10, 其他按牌面)。這張牌面朝上放在你的手牌區，點數加到你的總和中。

2.停牌 (Pass): 停止抽牌，鎖定你當前的總點數，本回合不再行動。

結束條件:

1.如果在抽牌後，你的「當前點數總和」超過了點數總和(例如紅線是 15, 你加到了 17)，立即「爆掉」。喝完莊家倒的酒。

2.當所有人都選擇「停牌」比較剩下玩家的點數。

點數最接近點數總和(但未超過)的玩家獲勝，成為下回合莊家，並可以指定一名玩家(ABCD)喝一杯自己倒的酒。

點數差最多的玩家(點數最小)，喝莊家倒的酒。

Round2 B莊家



酒的供應商

酒的感謝名單：

楷賀

Reg

BT(烈酒)

宣佑

# 酒的配方

葡萄酒 梅酒

威士忌跟可爾必思

# 真心話大冒險&LSA題目

## 真心話(5題)

1. 你會想修 LSA 下嗎?想不想當 LSA 的 TA?
2. 你覺得上週lsa報告的組別裡做最好的是哪一組?
3. 今天在場所有人中, 你覺得誰的穿搭最好看?
- 4.
- 5.

## 大冒險(5題)

1. 自己選在場的一個人喝交杯酒
2. 打開手機的google搜索欄
3. 答應老師下學期的LSA(下)你會修
4. 模仿一位老師, 讓同學猜
- 5.

## lsa題目(5題)

1. 哪一個指令可以用來顯示目前所在的目錄路徑?  
A) cd      B) ls  
C) pwd      D) mkdir  
正確答案:C) pwd
2. 要列出目前目錄下所有檔案(包含隱藏檔), 應使用下列哪個指令?  
A) ls      B) ls -a  
C) ls -l      D) dir  
正確答案:B) ls -a
3. 下列哪個指令可用於切換目錄?  
A) mv      B) cd  
C) cp      D) rm  
正確答案:B) cd
4. 若要複製一個檔案從A目錄到B目錄, 應該使用哪個指令?  
A) cp      B) mv  
C) rm      D) touch  
正確答案:A) cp
5. 下列哪個指令會「永久刪除」一個檔案, 且通常無法輕易復原?  
A) del      B) remove  
C) rm      D) rmdir  
正確答案:C) rm
6. 哪個指令可查看目前的系統時間與日期?  
A) time      B) date  
C) datetime D) clock  
答案:B) date

7. 要查看磁碟空間使用情況, 使用?

- A) df -h
- B) du -h
- C) free -h
- D) ls -h

答案:A) df -h

8. 要清理終端機畫面, 可使用?

- A) clear
- B) cls
- C) clean
- D) reset

答案:A) clear

分頁 10

### 3. 一整局遊戲的系統流程(可以講給老師聽)

你可以這樣描述：

1. 使用者用電腦連上同一個 Wi-Fi, 開啟瀏覽器連到樹莓派。
2. 前端顯示首頁 → 玩家選擇「酒鬼模式」或「閨家歡模式」，並輸入人數。
3. 遊戲開始前，前端開一個「基底酒選擇介面」，玩家從三種酒中選一種當本局基底，容量固定，例如 120 ml。
4. 每一回合：
  - 玩家輪流觸發轉盤／擲骰子。
  - 前端呼叫後端 API, 後端根據兩顆骰子的結果，決定：
    - 這回合何種事件(問答、黑白切、自選酒倒、電腦指定酒、真心話等)。
  - 事件進行中：
    - 若是問答 / 黑白切：根據勝負調整積分(+1 / -1 / 0)。
    - 若是喝喝 / 對數 / 大輸家：後端決定需要調一杯酒。
5. 當需要調酒時：
  - Game Server 呼叫 Drink Engine → 根據模式(酒鬼 / 閨家歡)、分數差、事件類型，決定酒精強度與配方。
  - Drink Engine 輸出配方 → 調酒 Controller 將配方轉成幫浦控制。
  - 調酒 Controller 控制樹莓派 GPIO → L298N → 實際打出酒。
6. 每回合只要有人喝，回合結束，回合數 +1。
7. 每 5 回合：
  - Game Server 檢查積分榜：
    - 積分最少者：觸發一杯懲罰酒。
    - 積分最高者：觸發一個「加一種酒」的 Bonus 動作。
  - 然後進入下一輪 1–5 回合。

分頁 11

網址

一個頁面(web server)

一個網頁伺服器(Server)

web server 被動式：我主動連進去

前端：我們會看到的頁面

後端：會用到 socket：建立一個共通管道，讓server有能力來找你，我主動聯絡道server

用fastapi

語言：python

呼叫東西

你去叫他，他會給你的回應是不是對的

<https://tw.alphacamp.co/blog/postman-api-tutorial-for-beginners>

server要向client溝通

server用socket 後，就可以

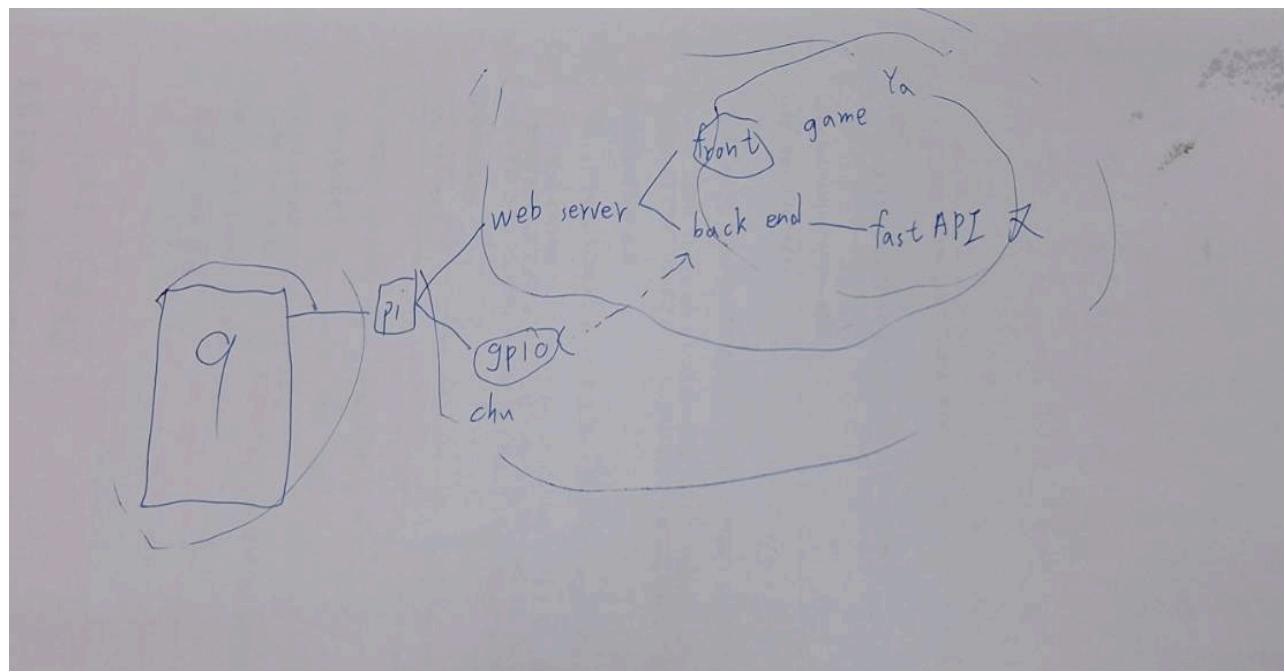
不用!!!

就用request，每秒都發訊息!!

每秒都戳一次

fetch one/all

一個人發request 給 server，每秒都會去確認，有幾個人在線，



**硬體**

	L298N腳位		樹梅派腳位
	第一台	第二台	
第一顆幫浦	in1		GPIO22 (15)
	in2		GPIO27 (13)
第二顆幫浦	in3		GPIO17 (11)
	in4		GPIO4 (7)
第三顆幫浦		in3	GPIO6 (31)
		in4	GPIO5 (29)

	L298N腳位		樹梅派腳位
	第一台	第二台	
第一顆幫浦	in1		GPIO23 (16)
	in2		GPIO22 (15)
第二顆幫浦	in3		GPIO27 (13)
	in4		GPIO17 (11)
第三顆幫浦		in3	GPIO5 (29)
		in4	GPIO6 (31)