

# 網際網路架構 期末作業報告

主題：輕鬆 21 點, 成員：王郁誠 n26130825

code: [https://github.com/superglutenman0312/black\\_jack](https://github.com/superglutenman0312/black_jack)

## 應用介紹

現代人生活忙碌，在努力的同時也該放鬆一下，那麼就揪些朋友，用這款輕鬆的 21 點來好好地放個鬆吧~

這是一個可以多人連線玩 21 點的小程式。支援文字介面 & 圖形介面(GUI) 兩種版本。使用 python 的 socket、thread 進行開發，而圖形介面版本則是使用 tkinter 開發。

## 使用方法

要進行遊戲時，client 端需先得知 server 端 IP，並更改 client.py 內 IP 位置。server 端透過 python server.py 建立 server，client 端再透過 python client.py 加入遊戲。而若是想玩圖形版本，則使用 server\_gui.py、client\_gui.py 進行遊戲。

## 遊戲規則

數字 2~10 代表花色對應分數，數字 J, Q, K 代表 10 分，數字 1 可以代表 1 or 11。玩家每回合會有一張初始牌，並且可以決定加牌或停牌，目的是湊到越接近 21 越好。所有玩家停牌後，開始比較分數，在分數不超過 21 點的情況下，分數最高者勝利。

由於是輕鬆版本，因此不採用其他像是莊家機制、過五關等等進階規則。

## 程式邏輯(以 GUI 版本為例)

### - server 端

呼叫 start\_server()，建立整副撲克牌資訊，並維護未抽過的牌堆。使用 socket.bind() 將 server 部屬到本地 IP:Port。

啟動 accept\_connections()，再透過 server.accept 接收玩家連線。當玩家連線後，分配執行緒來執行 handle\_client\_wrapper，處理個別玩家的互動。玩家進入 handle\_client()，紀錄玩家資訊(分數、socket...)，並定義玩家行動(DRAW、STOP)，此階段 client 端和 server 端會頻繁互動。

等所有玩家選擇 STOP 後，呼叫 announce\_winner() 計算贏家，並透過 send() 廣播給所有玩家。

### - client 端

呼叫 start\_client()，和 server 建立連線，並初始化圖形介面和變數。

當玩家點擊 DRAW 按鈕，用 send() 將該行為告知 server，server 發牌過來，client 用 recv() 接收抽到的牌，再用 calculate\_card\_value() 分析該牌分數，並算出目前牌數、總分。接著載入圖片並調整大小，再使用 tkinter 排版抽到的牌。

當玩家點擊 STOP 按鈕時，透過 send() 將玩家分數傳給 server，等 server 宣布冠軍時，再用 recv() 接收資訊並顯示出來。

## 程式展示

### - 文字介面

**server 端**：會顯示玩家行為，包括抽牌、停牌等(牌局開始時，會自動幫玩家先抽一張牌)，在所有玩家停牌後，會計算出冠軍，並宣布給所有玩家。

```
(21_dot) PS D:\Internet_app> python .\server.py
Server started, waiting for connections...
Player 0 connected, his IP address is: ('127.0.0.1', 8288)
Player ('127.0.0.1', 8288) drew 1 card. Remaining 51 cards.
Player 1 connected, his IP address is: ('127.0.0.1', 8291)
Player ('127.0.0.1', 8291) drew 1 card. Remaining 50 cards.
Player ('127.0.0.1', 8288) drew 1h. Remaining 49 cards.
Player ('127.0.0.1', 8288) drew 5d. Remaining 48 cards.
Player 0 has ended his round with a score of 16, waiting for other player
Player ('127.0.0.1', 8291) drew 2c. Remaining 47 cards.
Player ('127.0.0.1', 8291) drew 10h. Remaining 46 cards.
Player 1 has ended his round with a score of -1, waiting for other player
All player is done. Calculating result...
The winner is "player 0" with a score of 16 !
Shutting down server...
```

**client 端**：會告訴玩家他的 player 號碼。當玩家抽牌時，系統會告訴玩家抽到的牌，對應的分數，共抽了幾張牌，以及目前的總分等等(Ace 可以當作 1 or 11，系統會考慮這點並提醒玩家，如右圖)。當玩家爆牌時，系統會直接終止你的回合(如下圖)

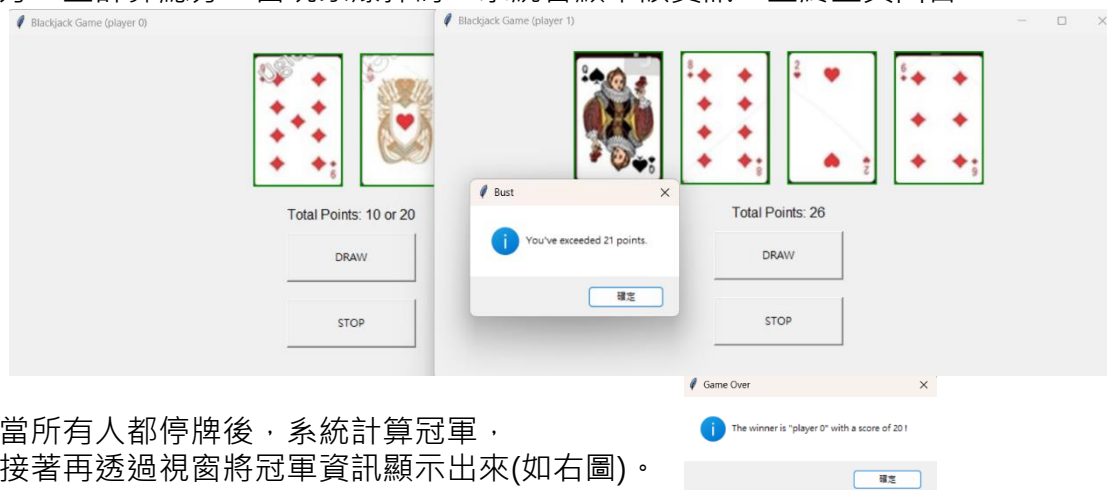
```
Drew 10h
This card value = 10
You have 3 cards, total card value = 22
Bust! You've exceeded 21 points.
```

```
(21_dot) PS D:\Internet_app> python .\client.py
You are "player 0"
Enter 'DRAW' to draw a card, 'STOP' to stop.
Your initial card is 11s
This card value = 10
You have 1 cards, total card value = 10
> DRAW
Drew 1h
This card value = 1 or 11
You have 2 cards, total card value = 11 or 21
> DRAW
Drew 5d
This card value = 5
You have 3 cards, total card value = 16
> STOP
Your total points is 16.
Waiting for other players...
The winner is "player 0" with a score of 16 !
```

### - 圖形介面

**server 端**：開場不會幫玩家抽牌，其他行為和文字介面時相同。

**client 端**：每位玩家會有一個操控介面，介面左上角會標記 player 編號，並用 DRAW、STOP 按鈕來進行遊戲。系統會將玩家抽到的牌顯示於介面上方，並計算總分。當玩家爆牌時，系統會顯示該資訊，並終止其回合。



當所有人都停牌後，系統計算冠軍，接著再透過視窗將冠軍資訊顯示出來(如右圖)。

## Future work

1. **合併 server 端**：文字、圖形介面版本的 server 端做的事情差不多，最大差別就是文字版遵循原始規則，進遊戲系統會先發一張牌給你。而圖形版則是玩家要從頭開始抽。因為 GUI 是 bottom driven 的，還沒弄好要怎麼加進去。等將這點融入後就可以整合成同一份 server code 了。
2. **增加局數機制**：目前設計是玩一局就結束，未來感覺可以加入局數機制，一路玩到牌庫抽光，再根據小局冠軍選出最終的冠軍。
3. **客戶端 GUI 改善**：目前客戶端 GUI 只能看到自己的牌，正常來說要能看到對方的牌才合理。然後系統開局發給玩家的牌則是只有自己能看到。若能改善這部分會讓遊戲的真實感更上層樓。