

1. 執行: type make

- a. Server: ./server 執行 server，會出現一個介面問你要開哪種遊戲的伺服器、人數，依照自己喜好輸入即可。
- b. Client: ./client "ip" 執行 client，會出現一個介面及若干選項，依序為 play，option，about us，story。若要進入遊戲，則選 P，再選擇合種遊戲即可。

2. 設計:

- a. Server: 這個 Server 執行檔是可以依照個人喜好自由創造兩種遊戲的伺服器，裡面的選單介面清楚明瞭，只要依照指示操作即可，但伺服器創造需檔遊戲人數額滿，方可開始遊戲。在 Blackjack 中，須等到玩家結束，莊家才可以開始玩，且在莊家方面不會紀錄玩家的牌卡，故計算金額的地方為在 client 端記錄。
- b. Client: Client 端的使用方式也相當清楚，包括了，play，option，about us，story，幾種選擇。在 about us 中記錄了寫作業的心酸血淚，在 option 中可以調整自己賭金的金額，在 play 當中可以再選擇 guess number 或 Blackjack 的遊戲，進入遊戲後須等伺服器人數額滿。

甲、Guess Number: 此為 10 回合致的遊戲，猜數字的順序依照遊戲者加入伺服器的先後，每人一次輪流，最多 10 回合，若都沒人猜到則結束，若有人先猜到，同回合的人猜到的人都算贏。在賭金計算的方面式由 client 端 send 給 server 端計算，算完再傳回來。

乙、Blackjack: 我有完成檢查所有結果的機制，也就是包誇過五關，三條，同花等等，如果玩到一半 print 出來的 point 變成負的必須按 s，因為這是特殊排出現的訊號，這寫這部分耗費相當大時間。尤其 split 特為難寫。在 server client 的 read write 中很容易不小心讀錯，或多給東西。在賭金方面全部都由 client 計算。

3. 結論: 以上兩個程式，我都沒有用 select 或 multi-thread 寫，純粹用 read write 完成，利用 read 會 block 住的特性，讓每個玩家可以輪流進行，TCP 的連捷方式並不難，難的是寫出剩下的 Blackjack 的過程。很抱歉這學期我的課業量太大沒有如期完成作業，也謝謝助教給我補交的機會，祝您們寒假快樂事事順心。