

網路技術與應用 期末專案

Client-Server 西洋棋

資管三 B09705007 張力升
資管三 B09705012 李彥澂
資管三 B09705027 許圃崧

主題發想

我們當初就是從遊戲為出發點來發想，而最直覺的想法就是做出一個讓玩家能夠連線進行的網路遊戲，但礙於時間以及技術上的困難，要同時滿足 UI 設計精簡以及遊戲邏輯簡單的要求，所以我們能選來做的遊戲其實不多，最後覺得西洋棋是一個不錯的選擇，因為他的棋盤只有 8×8 ，計算不會太多，邏輯上也不複雜，而且棋盤的介面也相對容易設計。所以我們最終的期末專案目標就是做出一個使用 Client Server 架構連線西洋棋。

程式架構

程式架構主要分成三個部分：前端、後端以及連線

前端

前端負責與玩家互動，工作包含「渲染棋盤」、「接收玩家的滑鼠操作」。

當輪到該玩家時，玩家便可以在棋盤上點擊。前端會根據點擊的方式判斷玩家想做的事情，並將指令送到後端：點擊自己的棋子可以預覽可本回合可移動的格子；再次點擊要移動的格子則可以移動；也可以點擊不可移動的格子取消預覽。

後端接到指令後會將運算後的棋盤再送回前端，前端將新的棋盤渲染出來，玩家就可以在進行操作之後看到棋盤發生變化。

另外，在輪到另一位玩家時，前端會跳出“Waiting for opponent”的提示視窗，並且過濾所有等待中玩家的操作。

後端

後端主要負責處理西洋棋的邏輯以及記錄整個棋盤的狀態。

當前端將指令送到後端時，後端會判斷是何種指令。主要分成三種指令：

1. 預覽：後端依據玩家想預覽的棋子種類以及棋盤目前的狀態，計算可以移動的格子，計算完之後將棋盤狀態及預覽一併回傳給前端。
2. 移動：對後端而言，當前端傳來移動指令時，代表該棋子確實可以移動到那邊（不然就變成取消預覽的指令），因此後端將該棋子移到玩家想移動的地方即可。
另外，如果小兵走到另一方的最後一排，那麼要將小兵變身為皇后。
3. 取消預覽：後端將原本的棋盤傳送回前端。

後端最複雜的部份在於處理「預覽指令」，每一種棋子可以走的方式都不相同。而「移動」跟「取消預覽」則相對簡單。

連線

這次我們實作的遊戲是 Client-Server 架構，必須有一台電腦執行 server 的程式來當作伺服器，並在其他電腦上執行 client 端的程式來連線至伺服器。而我們採用的連線方式為使用 http protocol 並使用 WebSocket 來進行連線：

- Server: 建立 http 伺服器，並用這個伺服器建立 WebSocket 連線。Server 在主機 IP 位置的 4000 port 上聆聽，等待接收 client 端的連線。
- Client: 執行 client 端程式，WebSocket 會連線到伺服器主機的 port 4000，當兩部客戶端電腦依序連線成功就可以開始進行遊戲了。

執行方式

前端以 React 實作；後端則是 JavaScript 在 node.js 中執行，要先確定電腦有可以的環境。

另外後端使用的 port 為 4000；前端使用的 port 為 3000，請確保這些 port 上沒有程式運行。

Server 程式

需到 `NWA-CHESS/backend/src/server.js` 中修改 `SERVER_IP`, 改成要跑 server 的電腦的 ip。

接下來在 `NWA-CHESS` 目錄下執行 `yarn backend`, 待跑出 `port on 4000` 即可

Client 程式

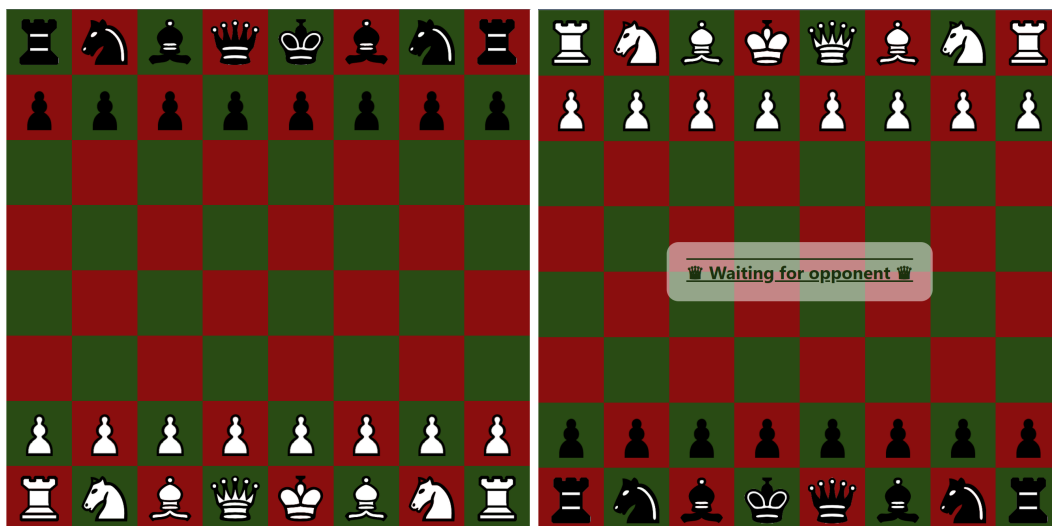
需到 `NWA-CHESS/frontend/src/containers/hook/useChess.js` 中修改 `SERVER_IP`, 改成要跑 server 的電腦的 ip。

接下來在 `NWA-CHESS` 目錄下執行 `yarn frontend`, 出現網頁畫面(網址是 `http://localhost:3000`)即可, 連上線時後端也會印出 `player connected` 的訊息。

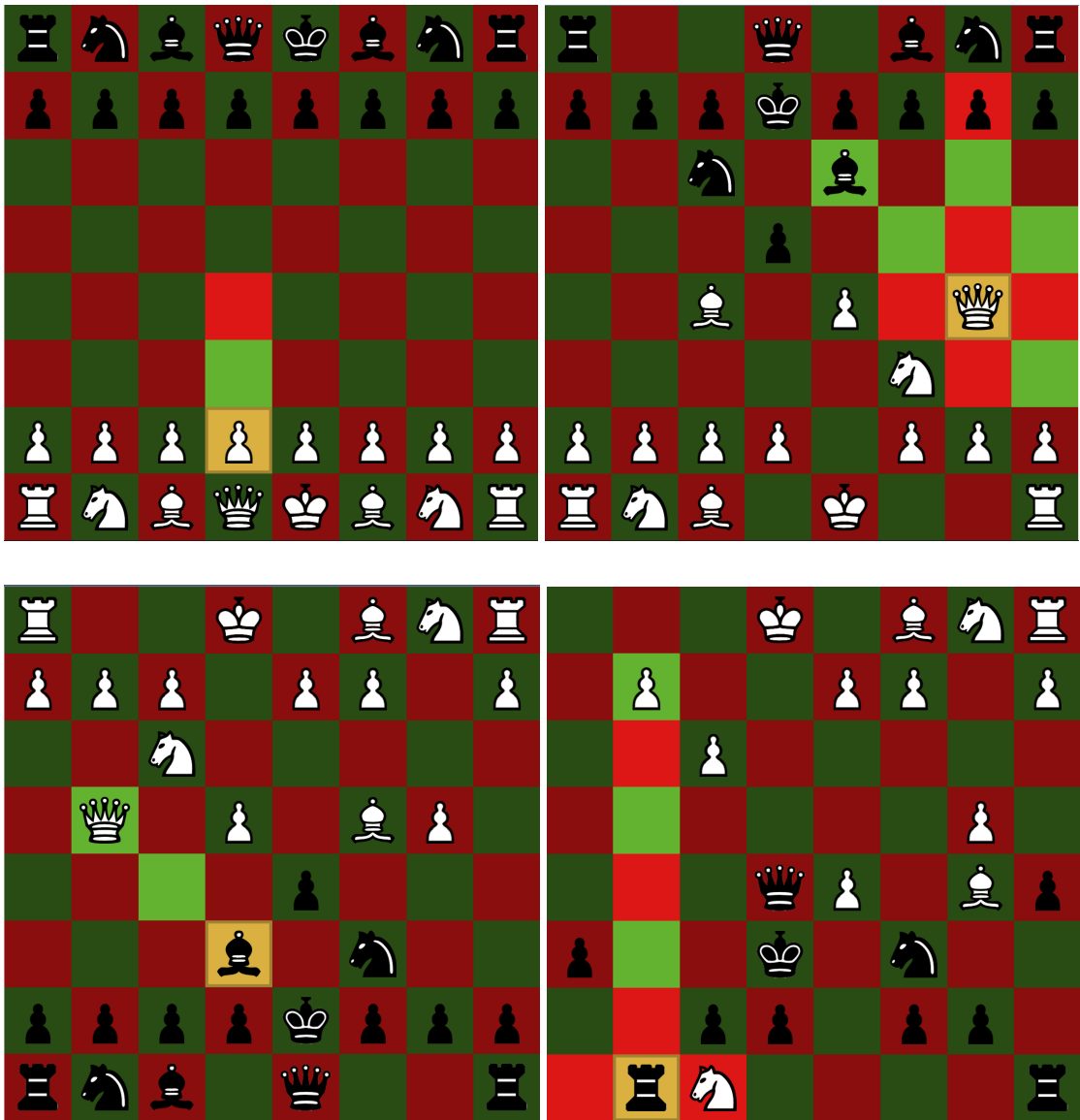
遊戲畫面

遊戲開始

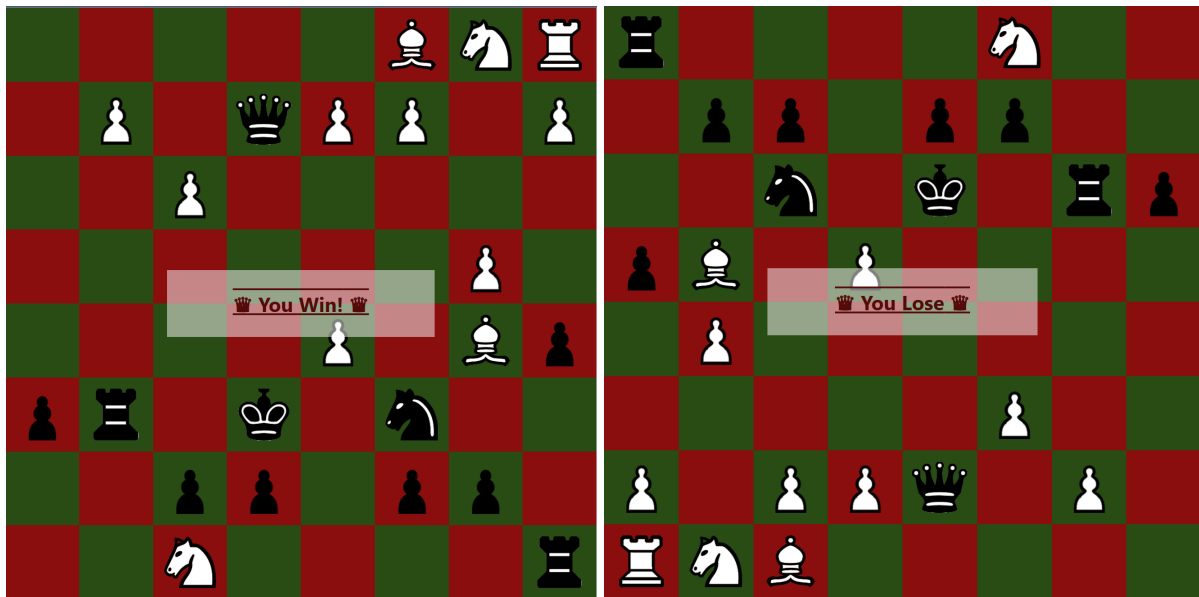
左為白方;右為黑方。第一手是白棋下, 因此黑方螢幕上會出現 “Waiting for opponent” 提示玩家等待對手。



預覽可走格子



遊戲結束



未來改善方向

建立完整遊玩體驗

目前只能開啟一場棋局，且結束之後不能重新開始。希望之後可以讓使用者連上 Server 之後，就可以隨意找人對弈，棋局結束之後可以回到大廳之後重新配對。同時 Server 也要能讓許多人同時上線，同時可以進行多場遊戲。

優化玩家局內體驗

在我們自己試玩的時候，發現對手走完輪到自己時常常會看不出來對手走了哪一步，希望在未來可以用一些方式引導玩家發現。

另外，目前尚未實作避免 check 的預覽判斷，以及某些西洋棋規則尚未實作（例如入堡、吃過路兵等較進階的規則）。未來如果時間充裕，應該很輕易就可以寫出這些判斷，讓棋局可以更貼近真正的西洋棋規則。

前後端邏輯分配優化

目前除了西洋棋預覽邏輯是明確由後端控制之外，其餘部分前後端分工不是很清楚，希望未來能夠讓程式邏輯更乾淨，便於維護。

連線限制放寬

雖然遊戲現在可以透過 IP 位置連線，但是目前僅限於在同一個 private net 中透過 private IP 來連線。外來若將連線方式研究更透徹，希望能將此限制打破，可以從公開網域來進行連線。

Deploy server

經費充足的話希望可以 deploy server, 搭配完整的遊戲體驗, 讓西洋棋最後可以變成是一個可以隨時打開隨時遊玩的遊戲。