

海戰棋---最終征戰

團隊夥伴：

資訊三甲 林泓劭 D0745530

資訊三甲 馮伯誠 D0745908

目錄

目錄.....	2
一、 簡介：	3
1.1 海戰棋簡介.....	3
1.2 遊戲規則.....	3
1.3 動機.....	3
二、 相關的作品差異：	3
三、 連線方式：	4
四、 子模組、模組之間的關係及界面：	4
五、 客戶端及伺服器運作的流程：	4
六、 客戶端與伺服器之間訊息傳遞循序圖：	5
6.1 登入.....	5
6.2 等待進入房間.....	5
6.3 設置船艦.....	6
6.4 攻擊、判斷輸贏.....	7
七、 成果：	8
八、 結論：	10
九、 參考文獻：	10

一、 簡介：

1.1 海戰棋簡介：

海戰棋是在 20 世紀初發明，由美國的遊戲公司 Milton Bradley 帶領而流行的，由一張紙上相連的 10x10 的格子組成。左邊的格子不僅用來記錄你每艘船的位置，而且同時記錄你的對手所射擊的位置。右邊的空格除了記錄你射擊的位置外，還記錄每一擊的結果。

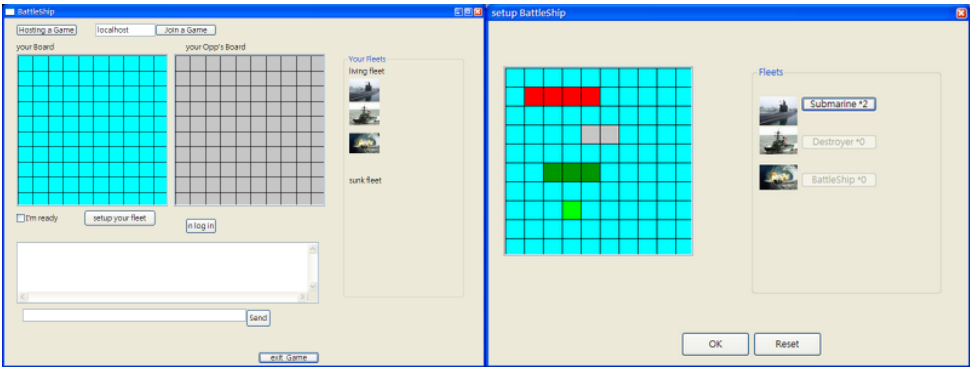
1.2 遊戲規則：

每位玩家在一張空白的格子紙上開始，並在上面放置一些船。通常使用五艘船（航空母艦占有 5 個格子、戰艦占有 4 個格子、驅逐艦及巡洋艦占有 3 個格子、潛艇占有 2 個格子、巡邏艇占有 2 個格子），每艘船的格子不得重複。每位玩家輪流射擊，通過說出他們想要攻擊的方格坐標來射擊，而另一個玩家必須說出射擊是否命中，最先擊沉對方所有船艦者即可獲得勝利。

1.3 動機：

實作電子版海戰棋，避免了紙張的浪費，藉此愛護環境拯救地球，為了世界的環境與森林的保護盡一份心力。

二、 相關的作品差異：



	與他人連線 對戰	版面整潔	可以改變船艦 方向(垂直/平行)
他牌	X	X	X
海戰棋 最終征戰	O	O	O

三、 連線方式：

利用 Java RMI 建構伺服器，以便於資料以物件型態進行傳輸。利用 RMI 建構伺服器同樣具有缺點，RMI 是屬於遠端程序呼叫的類型，因此伺服器無法主動傳送資訊給使用者，因此需要使用者主動呼叫程序更新自身資料。在客戶端方面，為了及時更新部分玩家資訊，且避免程序卡死在某處，因此選擇開啟其它執行緒來協助取得伺服器資料以更新本地端玩家參數與畫面。

四、 子模組、模組之間的關係及界面：

客戶端：

- i. 登入遊戲：向伺服器傳送玩家資訊。
- ii. 要求加入遊戲：傳送配對要求給伺服器。
- iii. 配置完成：傳送玩家自身資訊與船艦配置給伺服器做紀錄。
- iv. 發動攻擊：將攻擊指令傳給 Server 並等待回應。

伺服器端：

- v. 接收玩家：賦予呼叫模組之玩家序號，並回傳給客戶端。
- vi. 接收加入遊戲要求：檢查是否有他人同時在線等待，無則等待，並建立房間編號，回傳等待信號；有他人在線等待則進行配對，更新玩家資訊，並回傳且遊戲開始信號。
- vii. 接收船艦配置資訊：接收玩家資訊與船艦配置並更新伺服器資料，根據更新結果回傳成功或失敗。
- viii. 接收攻擊指令：接收攻擊指令，確認結果並向呼叫者回傳攻擊成功或攻擊失敗。此外更新受攻擊者資訊，以便另一方玩家更新資訊。若有決定出勝負，也會即時更新房內相關參數資訊。
- ix. 檢測遊戲結束：檢測當局遊戲是否結束，結束後則更改房間內贏家參數，並等待雙方玩家抓取資料。
- x. 取得遊戲狀態：能供玩家根據身分取得自身的最新資訊。查看目前是誰的回合。
- xi. 回合控制：於第一回合前決定先攻方，並設定其參數。當玩家攻擊完後，將攻擊方改為另一使用者。

五、 客戶端及伺服器運作的流程：

客戶端：

- 等待玩家開啟遊戲，輸入姓名後即可登入並等待對方加入遊戲。

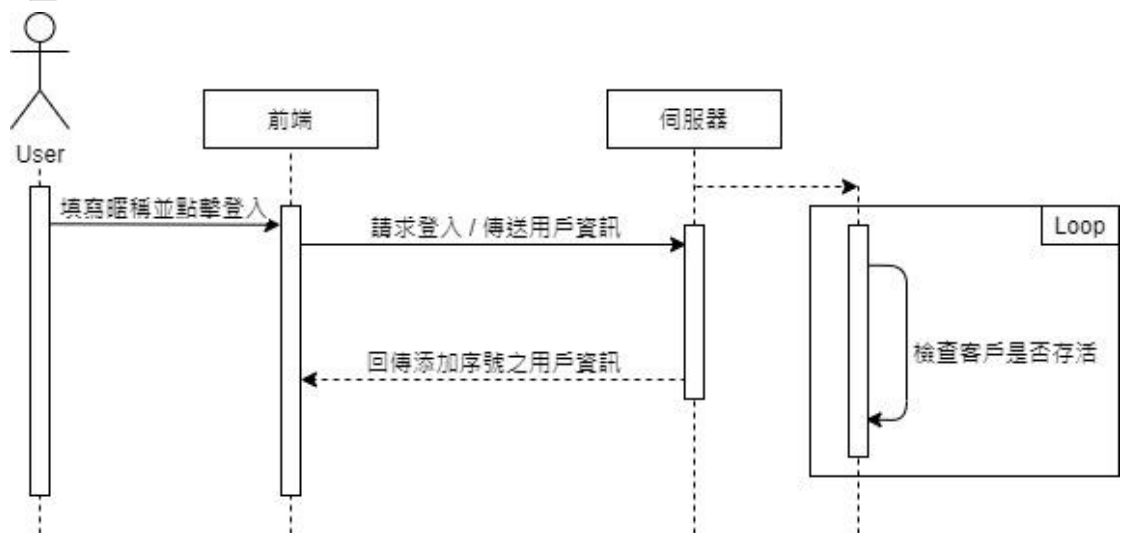
- 開始遊戲後，會看到兩張 10 x 10 的方框，左側為使用者的地圖，右側為對手的地圖。並另開兩條執行序，第一個是傳送客戶端還存活的訊號；第二個是獲取伺服器中玩家的狀態，以利及時的更新。
- 進行船艦設定，在使用者的地圖設定完後，傳送給伺服器。
- 開始進行攻擊。

伺服器：

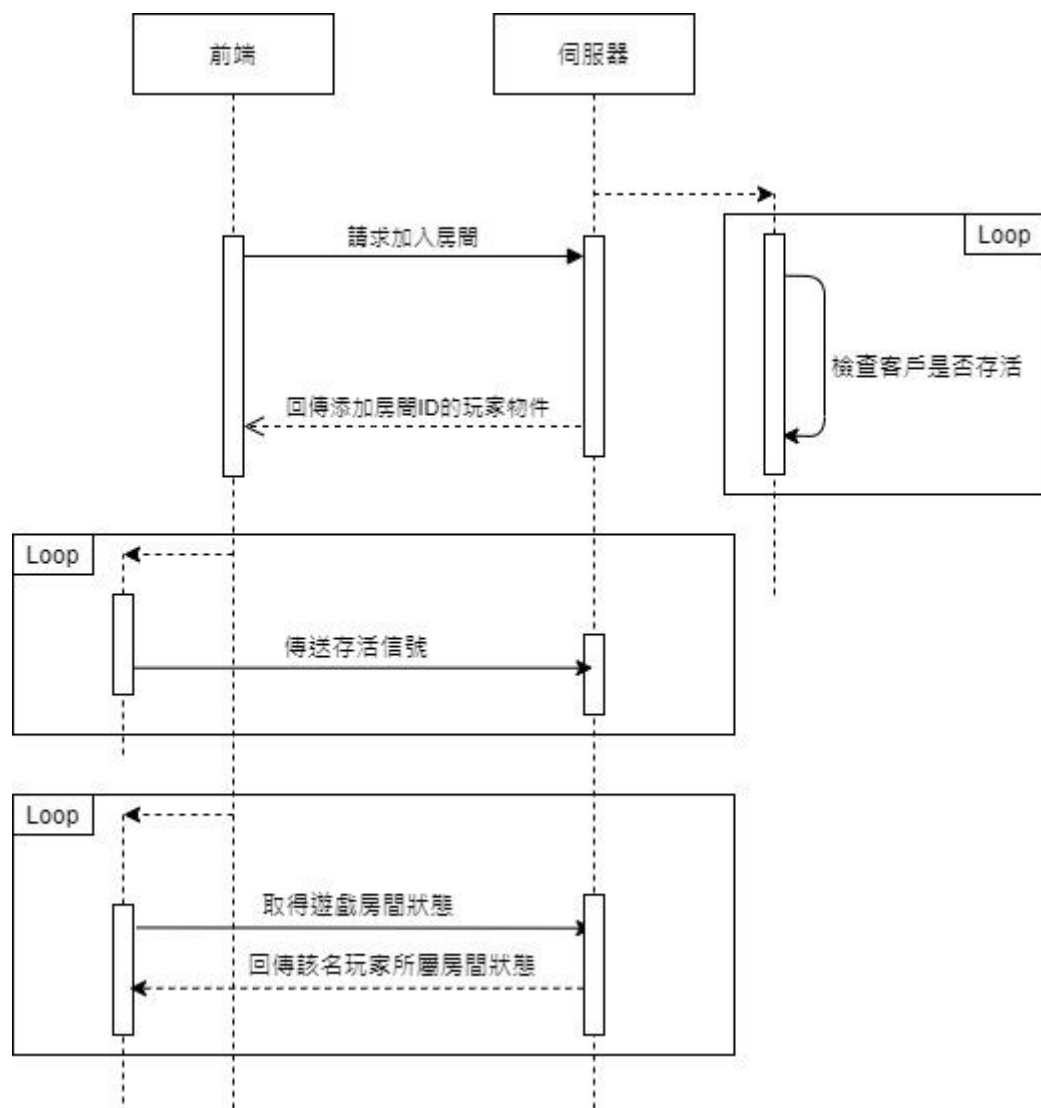
- 啟動 RMI registry 與伺服器，並根據呼叫的指令執行相對應的行為。
- 因為以遠端呼叫連線伺服器，因此開啟第二執行續，以刷新各玩家的狀態變數，以獲知玩家是否結束程序離開遊戲，若玩家離開遊戲，則判定另一方玩家獲勝

六、 客戶端與伺服器之間訊息傳遞循序圖：

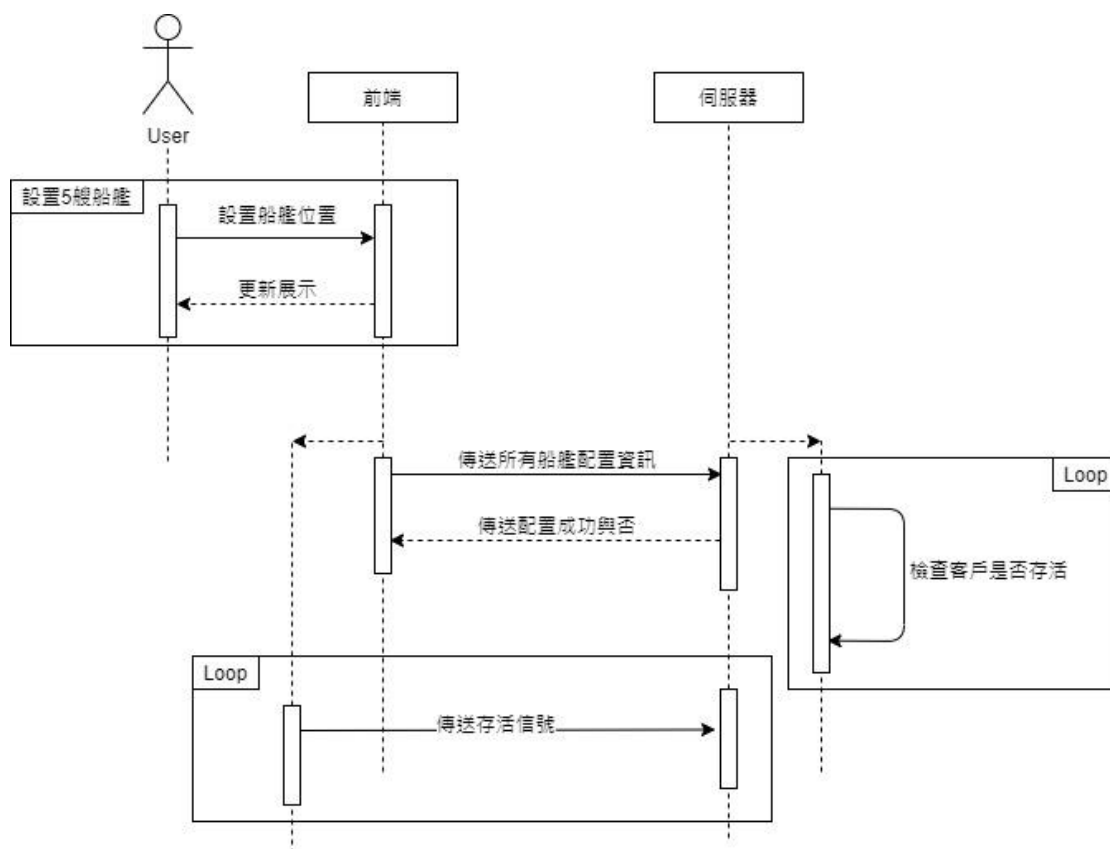
6.1 登入



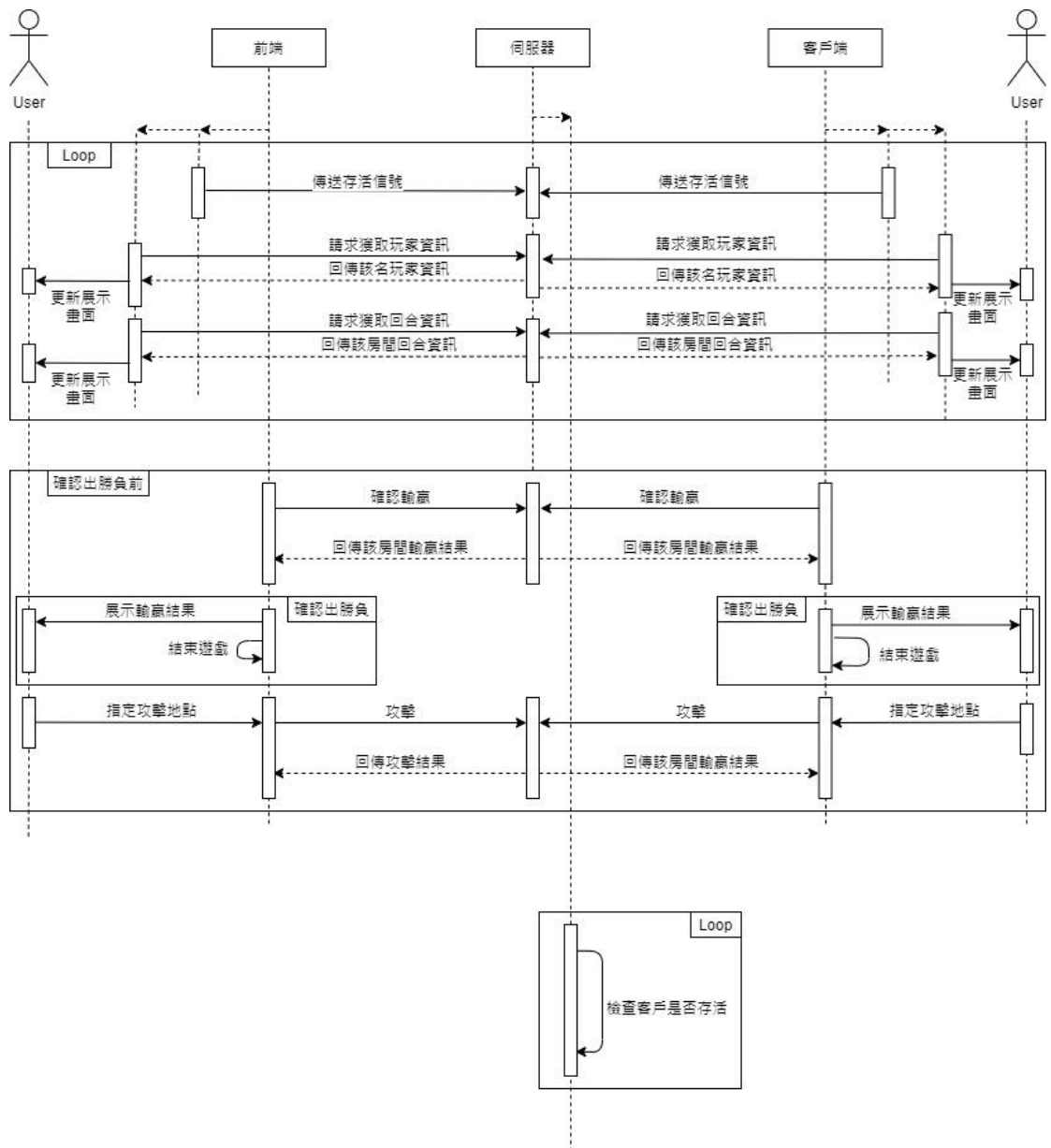
6.2 進入房間等待



6.3 設置船艦

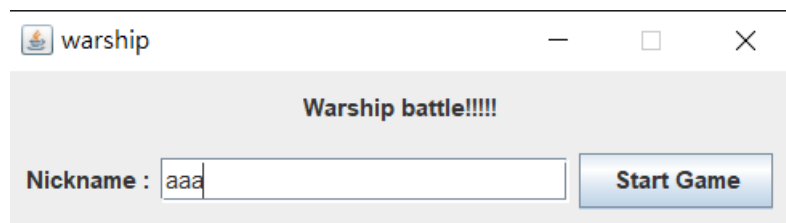


6.4 攻擊、判斷輸贏

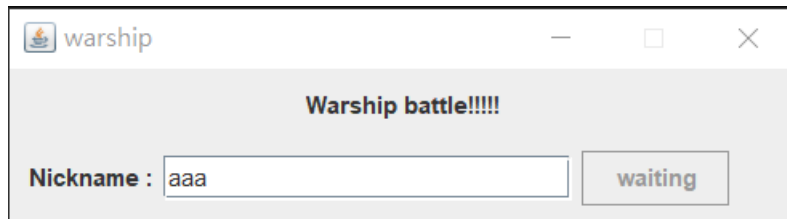


七、 成果：

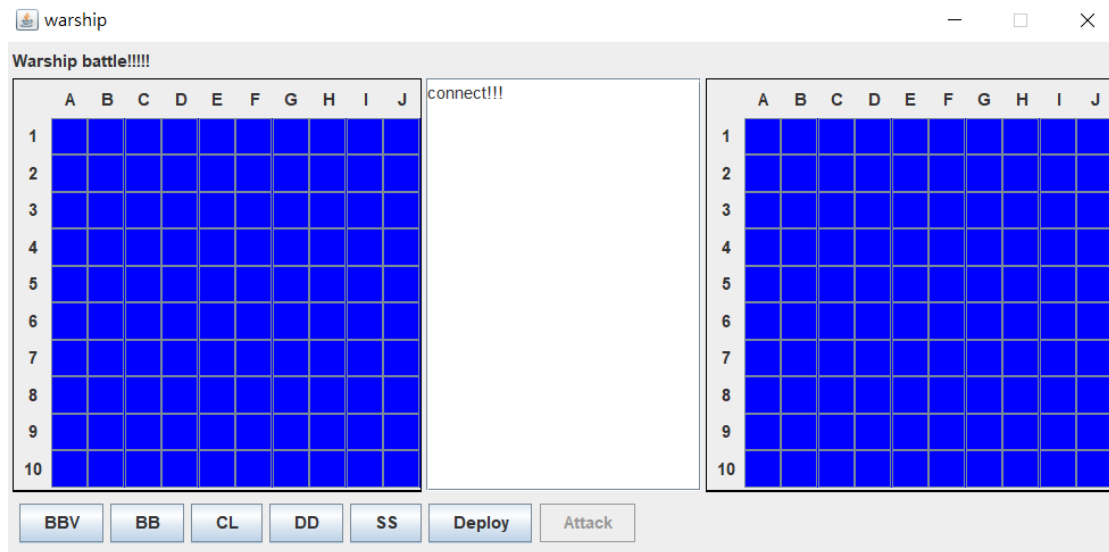
登入：



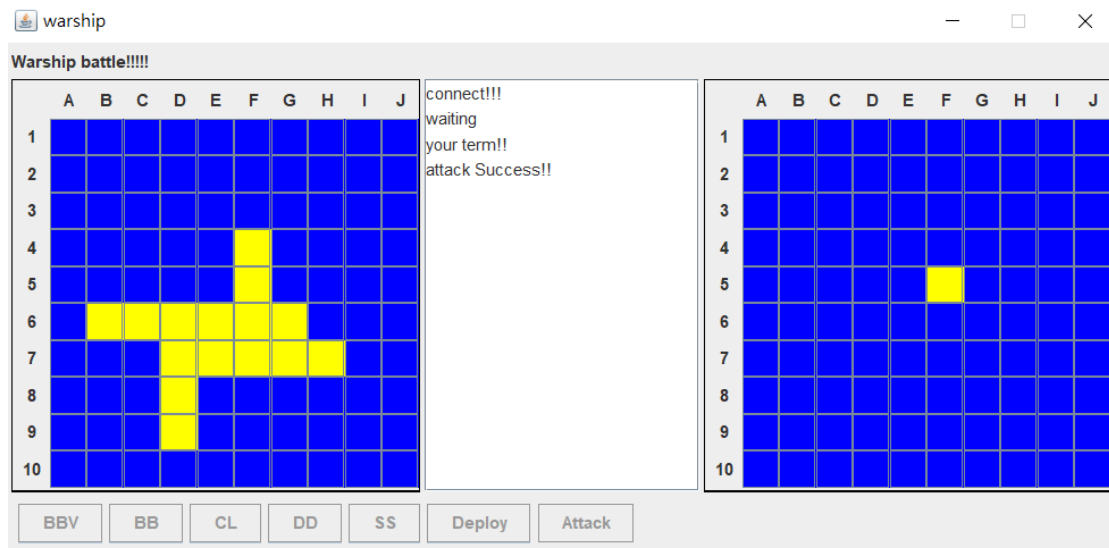
加入房間並等待對方玩家：



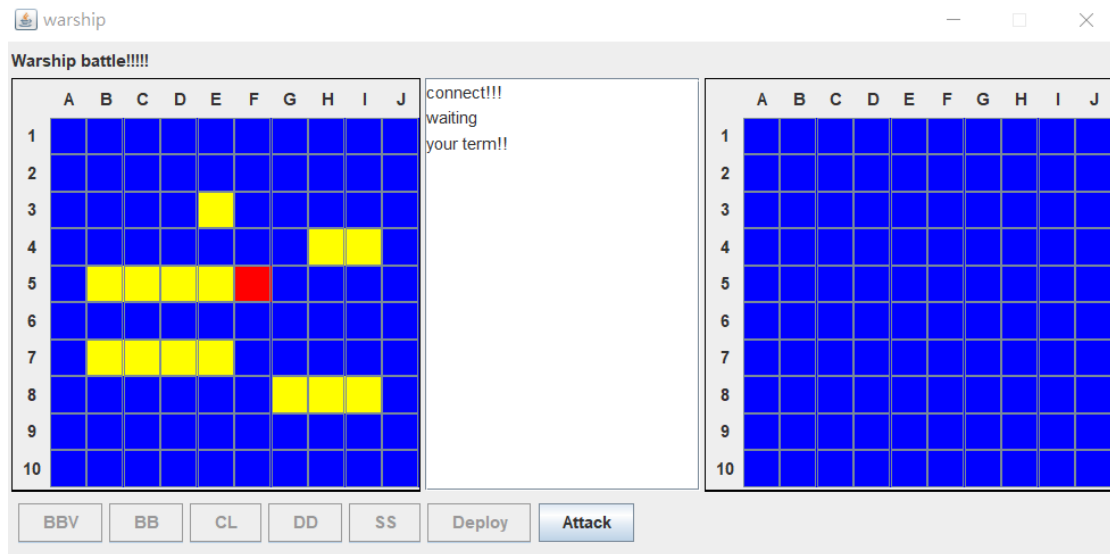
進入遊戲畫面，並部署船艦：



發動攻擊後，回傳攻擊是否成功：



受攻擊時的畫面顯示：



八、 結論：

這個網路程式專題雖然簡單，我還是從中學到了很多，這個專題的我們使用 RMI 的原因是易於使用，並且不需要考慮傳輸或掉包的問題，這讓我們在做專題的過程中省下了不少的時間，GUI 的部份我們則是花的不少的精力來製作，平常在看其他人做好的 GUI 會覺得其實 GUI 也不過如此，在自己做了之後才覺得其實並不容易，在這個時候我們才會發覺自己不足的地方，看似瞭解其實連邊都碰不上，但是在經過了這一次的專題之後，總算對 GUI 有更進一步的了解。

一個專題的製作，除了考驗個人的行動力以及技術力之外，還有考驗團隊的合作與溝通能力，「兄弟齊心，齊力斷金」真的是確有其事，在這次的專題確實的讓我們感覺到合作的重要，除了合作，溝通也是很重要的技能，合作但不溝通很容易南轅北轍，溝通但不合作則會讓團隊進度停滯不前，兩件事都做到才是個合格的團隊，這個專題讓我們學到最多的除了基本的技術之外，就是團隊合作的部分。

九、 參考文獻：

規則介紹：維基百科提供

([https://zh.wikipedia.org/wiki/%E6%B5%B7%E6%88%98_\(%E6%B8%B8%E6%88%8F\)](https://zh.wikipedia.org/wiki/%E6%B5%B7%E6%88%98_(%E6%B8%B8%E6%88%8F)))

作品比較：

Crazylion(<https://crazylion.wordpress.com/category/open-source/java/>)