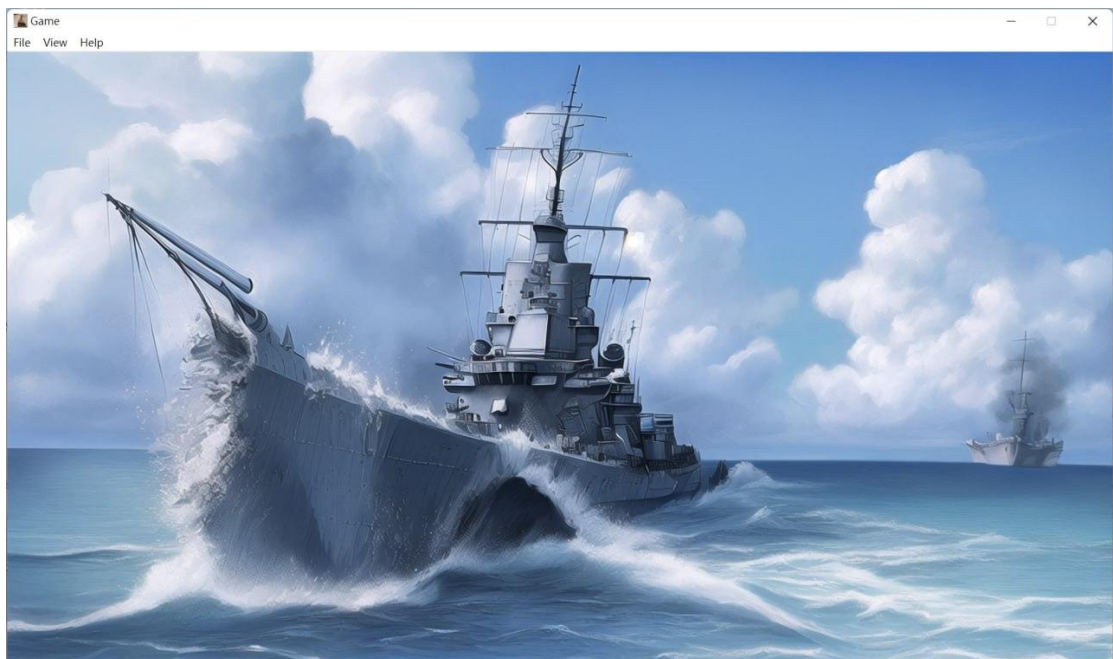


國立臺北科技大學

2023 Spring 資工系物件導向程式實習

期末報告

海戰棋



第九組

目錄

壹、 簡介	3
一、動機	3
二、分工	4
貳、 遊戲介紹	5
一、遊戲說明	5
二、遊戲圖形	7
三、遊戲音效	11
參、 程式設計	12
一、程式架構	12
二、程式類別	13
三、程式技術	14
肆、 結語	15
一、問題及解決方法	15
二、時間表	17
三、貢獻比例	19
四、自我檢核表	19
五、收穫	19
六、心得或感想	21
七、對於本課程的建議、想說的話:如果有的話	22

圖目錄

圖 1 遊戲首頁頁面	7
圖 2 遊戲主選單	7
圖 3 單人模式困難度選單	7
圖 4 玩家一進行擺放階段	8
圖 5 玩家一完成擺放階段	8
圖 6 雙人模式下玩家二完成擺放階段	8
圖 7 回合進行階段	9
圖 8 回合進行階段	9
圖 9 遊戲進行直到進入遊戲結束階段	9
圖 10 遊戲結束，可以選擇重玩或者離開遊戲。	10
圖 11 作弊模式下顯示所有船隻	10
圖 12 遊戲 CLASS DIAGRAM	12

表目錄

表 1 遊戲圖形與對應介紹表	7
表 2 陳列在遊戲中的音效資源：	11
表 3 陳列在遊戲中的程式類別名稱以及對應說明：	13
表 4 時間與進度表	17
表 5 貢獻比例表	19
表 6 自我檢核表	19

壹、 簡介

一、 動機

本專題選擇開發海戰棋遊戲作為物件導向程式設計的主題，旨在探索並應用基礎程式設計與物件導向程式設計，帶給我們的演算邏輯及物件概念和技術。海戰棋遊戲，不單是一款僅頻運氣隨機就能取勝的遊戲，更是一款具有豐富的規則性和策略性的遊戲，玩家需思考佈局、推測對手的船艦位置和分析對手的行動模式等，這將提供思考演算法和設計，不同的遊戲關卡難度，也將會有對應不同的應對規則。遊戲的規則和算法將進一步，促使深入應用與理解物件導向觀念，透過良好的物件導向程式設計架構的建立，將遊戲與算法元素轉化為物件，並利用封裝、繼承和多形等概念來組織創建整體遊戲。

二、 分工

張庭瑋

- I. 客製化類別與函式
- II. 通用遊戲架構
 - i. 棋盤擺放與隨機配置
 - ii. 回合機制
 - iii. 音效
 - iv. 作弊模式
- III. 單人模式
- IV. 報告

李子儀

- I. 雙人模式
- II. 棋譜衍伸類別建置
- III. UI
- IV. 報告

貳、 遊戲介紹

一、 遊戲說明

開啟遊戲時會進入遊戲首頁（圖 1），在此頁面點擊左鍵即可進入主選單。

主選單（圖 2）提供玩家兩種模式可供選擇，分別是單人模式和雙人模式。單人模式共有四種難度（圖 3）：菜鳥、困難、專家與大師。在選擇完難度後進入擺放階段。

在擺放階段（圖 4、圖 5）中，玩家可以根據個人策略進行手動擺放船艦，也可以選擇自動擺放功能，由系統自動為其擺放船艦。不論是手動還是自動擺放，玩家在擺放完成後仍有機會進行微調修改，以確保自己的船艦佈局最為理想。

另一方面，如果玩家選擇雙人模式，則玩家一和玩家二將輪流進行擺放階段（圖 4、圖 5、圖 6）。

當雙方完成擺放後，遊戲進入回合進行階段（圖 7、圖 8）。在每輪遊戲中，玩家一將先行，從棋盤上選擇一格進行猜測。玩家可以選擇不同的位置進行猜測，以找出敵方的戰艦位置。

若玩家一猜中了敵方的戰艦，他將繼續進行猜測，直到未猜中為止。然後，輪到電腦控制的敵方（如果是單人模式）或玩家二進行猜測，依此類推。

在每個回合結束後，回合進行權將再次回到玩家一手中，並依此輪流進行，直到其中一方擊沉了對方的所有船艦（圖 9），此時進入遊戲的結束階段。

在結束階段（圖 10）中有兩個按鈕。分別提供回到主選單與離開遊戲的功能。

此外，遊戲配有作弊模式（圖 10）。使用作弊碼「上上下下左右左右 B A」以開啟作弊模式。當作弊模式啟用時會切換背景音樂，在回合階段中顯示雙方所有船隻。

二、 遊戲圖形

表 1 遊戲圖形與對應介紹表

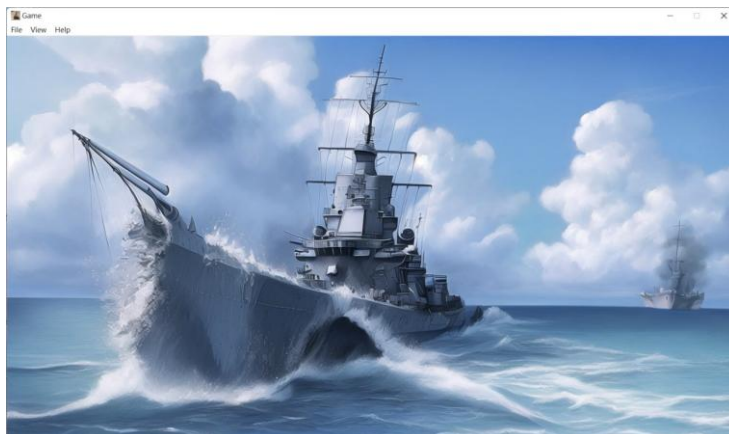


圖 1 遊戲首頁頁面

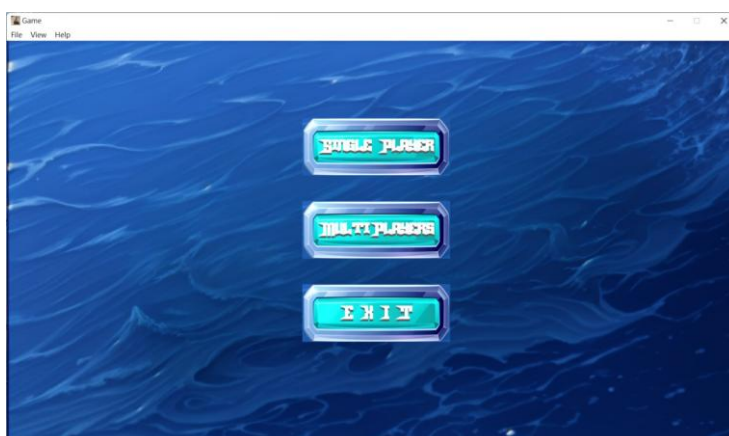


圖 2 遊戲主選單

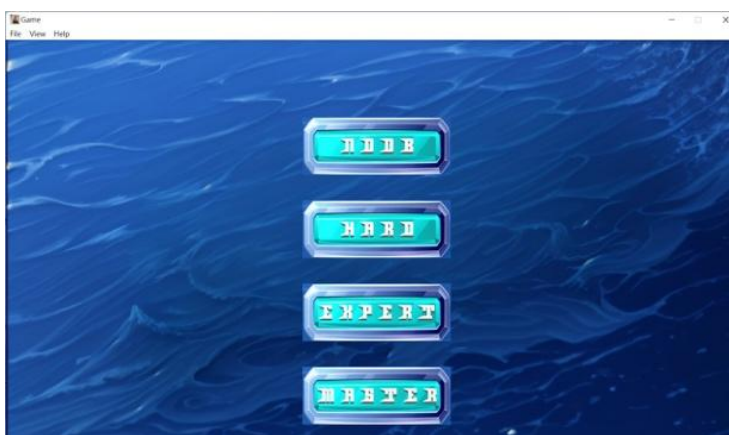


圖 3 單人模式困難度選單

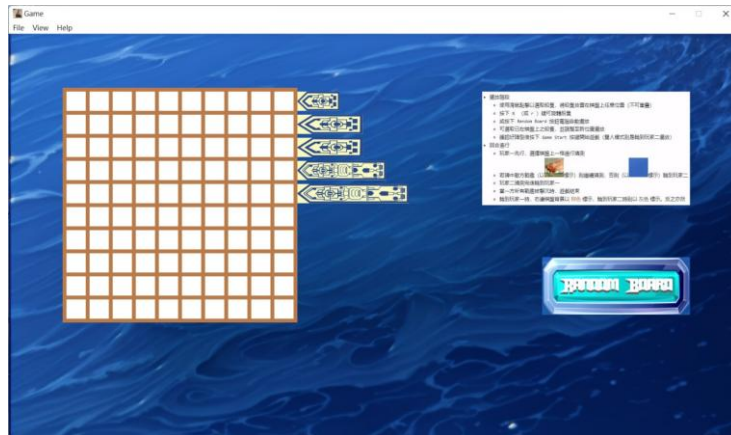


圖 4 玩家一進行擺放階段

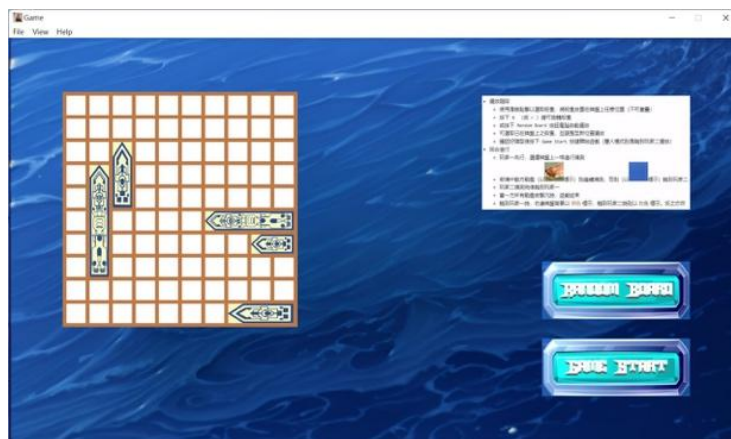


圖 5 玩家一完成擺放階段

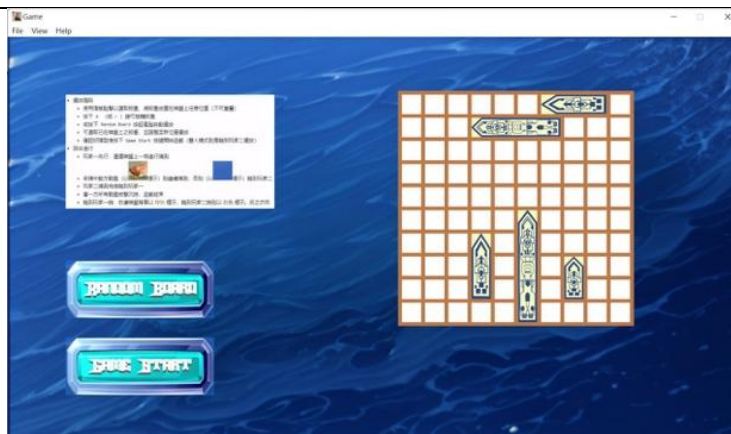


圖 6 雙人模式下玩家二完成擺放階段

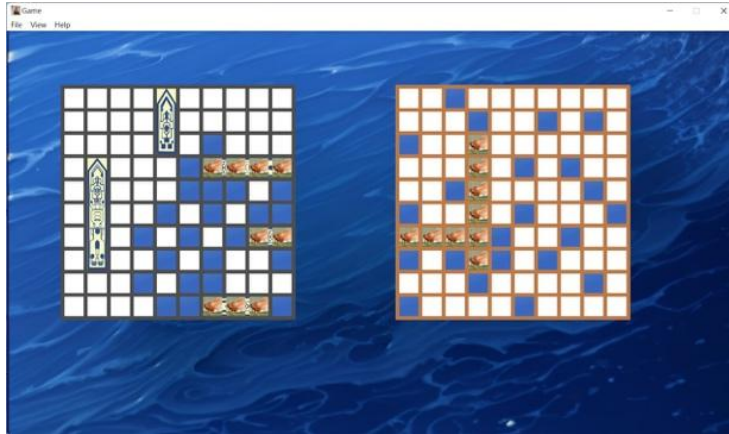


圖 7 回合進行階段

(其中，此圖右棋盤格呈現亮棕色，為提示輪到玩家一)

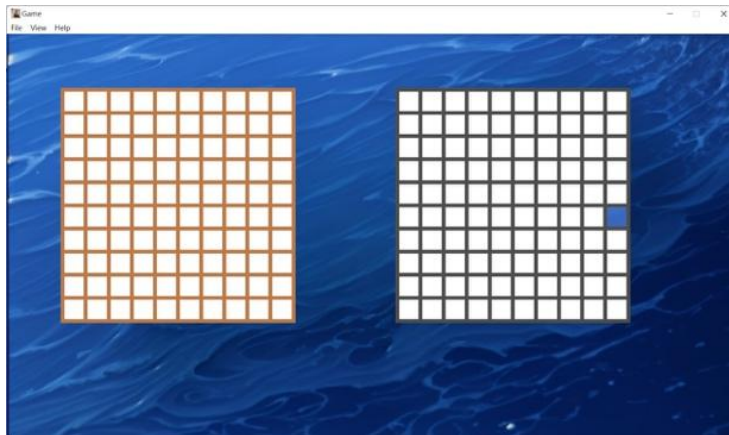


圖 8 回合進行階段

(其中，此圖左棋盤格呈現亮棕色，為提示輪到電腦/玩家二)

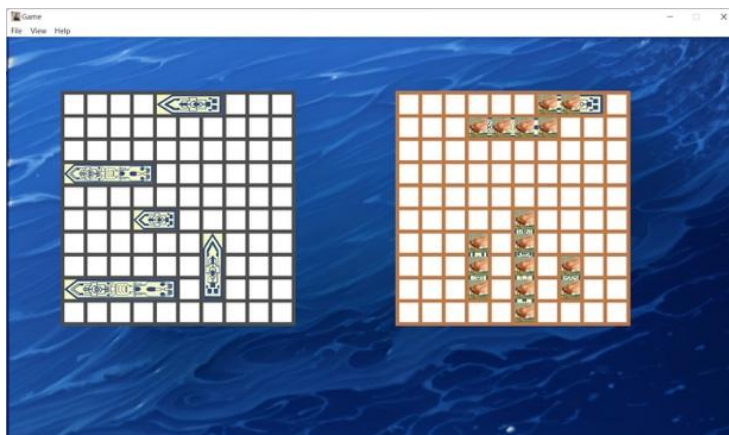


圖 9 遊戲進行直到進入遊戲結束階段

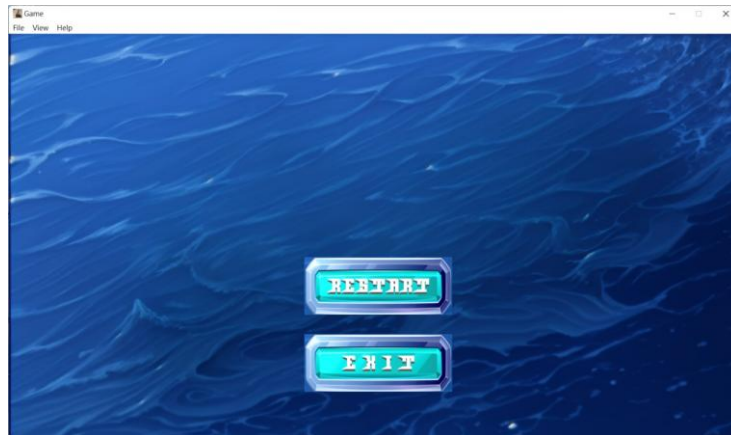


圖 10 遊戲結束，可以選擇重玩或者離開遊戲。
(重玩即回到圖 2 遊戲主選單)



圖 11 作弊模式下顯示所有船隻

三、 遊戲音效

表 2 陳列在遊戲中的音效資源：

音效類別		音效檔案
遊戲主題音樂（背景音樂）		“Theme.wav”
擊中音效（玩家）		“Hit_Marker.wav”
擊中音效（電腦）		“Error.wav”
按鈕點擊音效		“Click.wav”
未命中音效 1		“Missed1.wav”
未命中音效 2		“Missed2.wav”
未命中音效 3		“Missed3.wav”
未命中音效 4		“Missed4.wav”
單人模式下玩家一獲勝	其他難度	“Defeat_NotDS.wav”
	大師難度	“Defeat_Dark_Soul.wav”
單人模式下電腦獲勝		“Defeated.wav”
作弊模式啟動音效		“CheatModeActivated.wav”

參、 程式設計

一、 程式架構

遊戲主要在 mygame.h 與 mygame_run.cpp 中運行並配合其他類別與函式運行。

初始化階段由於含有大量路徑使得 init() 變得十分冗長，故使用 PhaseInitializer 系列物件以減少 mygame_run 中的行數。

在擺放與回合進行階段主要使用 GameBoard 類別的物件管理 BaseGrid 與其子類別物件，並使用 tinyUtils 中自定義的各式函式針對不屬於類別應有的功能(例如複製及生成棋盤等)進行操作。

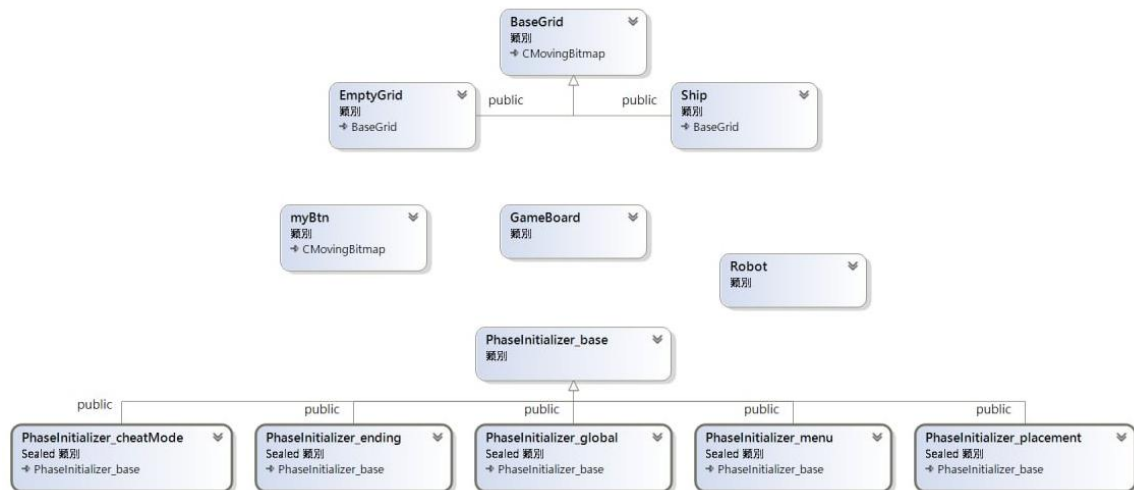


圖 12 遊戲 Class Diagram

二、 程式類別

表 3 陳列在遊戲中的程式類別名稱以及對應說明：

繼承自	類別名稱	標頭 檔行 數	原 始 碼 行 數	說明
CMovingBitmap	BaseGrid	18	45	棋盤上每個格子的基礎類別
BaseGrid	EmptyGrid			空白格子
BaseGrid	Ship	28	90	實作船隻類別 (包含船隻血量、方向等屬性)
CMovingBitmp	myBtn	15	31	實作按鈕類別
無	GameBoard	51	250	實作管理、顯示 棋盤上所有格子的類別
無	Robot	51	150	四種難度的機器 人以模擬單人模 式下的玩家二
無	PhaseInitializer_base	77	91	初始化工具的基礎類別
PhaseInitializer_base	PhaseInitializer_global			各遊戲階段的初 始化工具，負責 初始化各個 CMovingBitmap 的資源與預設幀
PhaseInitializer_base	PhaseInitializer_menu			
PhaseInitializer_base	PhaseInitializer_placement			
PhaseInitializer_base	PhaseInitializer_ending			
PhaseInitializer_base	PhaseInitializer_cheatMode			

三、 程式技術

1. 不同難度的機器人

單人模式下的機器人一共有四種難度。在不重複(Robot 類別中有一個 set 紀錄已經下過的位置，每當試圖下在重複位置時會重新依照各難度的方式重新選取一格)分別使用不同的邏輯進行下棋：

- I. 新手模式：簡單模式下的機器人採取「[猴子打字機](#)」策略，每一次的出手都是隨機選取。
- II. 困難模式：基於新手模式的邏輯，但在成功擊中船隻後會嘗試往上、下、左與右進行猜測。
- III. 專家模式：參考[此](#)網頁的教學，平時以西洋棋盤狀猜測，在擊中船隻時與困難模式一樣，會往四周猜測。
- IV. 大師（作弊）模式：此模式下的機器人會在內部的 `_cheatCount` 歸零前照著專家模式的邏輯進行遊戲，並在歸零後讀取在遊戲開始時取得的玩家船隻座標一一攻擊。

2. 空間換取時間

考量到掉幀對於遊玩體驗的破壞，我們的最佳化策略大略上為使用空間換取時間。我們使用了許多額外變數紀錄船隻位置、前次攻擊結果等資訊，避免 $O(n^2)$ 的掃描棋盤發生。

例如在實作 `GameBoard::dropShip()` 時，棋盤需要檢查是否有足夠的格子能在不超出邊界或是與其他船隻重疊。利用 `BaseGrid` 底下的 `_shipID` 屬性。其預設值是 -1，並在有船隻於此

進行互動時做對應的更動。藉此，我們可以在 $O(\text{船隻大小})$ 的時間下而非遍歷整個棋盤的情況下檢查。

3. 智慧指標

使用現代 C++ 的智慧指標。利用解構式不論是抵達生命週期盡頭或是被例外丟出都會執行的特性解決傳統指標容易產生記憶體洩漏的問題進而增加安全性。

肆、 結語

一、 問題及解決方法

張庭瑋:

這學期的實作由於我們挑選的題目相較其他組沒有那麼複雜，遇到的問題並沒有很多。底下稍微列出幾個困擾較久的：

第一個是環境建置的問題。由於對於 Visual Studio 的操作、介面與他的啟動時間十分感冒，決定還是使用 JetBrains Rider 配上 MSBuild 作為主要開發環境。然而在期中發 Release 前想要把編譯目標從著重快速編譯的 Debug 模式改為注重執行效能的 Release 版本卻遇到了 .vcproj 對於預編譯標頭檔的設定問題，幸虧在一番搜尋後也成功把執行檔縮小。最可惜的是 Toggl 在 Rider 上的擴充元件在學期中的時候壞掉了，以至於後半學期的工作時數都忘了登記。

第二個是當初在做隨機棋盤時原本是假的隨機，建好數個棋盤再透過亂數產生選一個使用。後來經過一番討論後才變成現在的找空位放置。

李子儀:

這學期的實習課是我第一次真正使用，visual studio 做程式的編輯編譯開發，一開始對於龐大的程式碼，需要的理解 trace code、再撰寫、再偵錯 debug。以前對於流程理解上，就是在對於不理解的位置處，以變數儲存值寫輸出去看，然後整份專案重編，偵錯更慘，需要把可能開始錯誤的地方之後都先註解掉，在猜測的位置點後逐一寫輸出，排除問題，而這次我讓我頻繁使用的好工具便是中斷點 breakpoint，他讓我可以清楚看到執行到該中斷點時的之前的所有內容，慢慢理解整個程式的編譯架構流程。舉例來說，記得第一個遇到的問題就是，我在寫最初版本的假 Random Board 時，這涉及的是棋盤上船隻擺放的問題，我一開始執著於看起來像是的名稱片面上的 GameBoard 類別，去做衍伸修改，但很快地就發現怎麼 Board 編起來不是不見就是跑掉，埋頭苦做了一大圈後，偶然間真正的學會了應用 vs 的中斷點、看入類別間的關係、變數儲存的是什麼東西、現在長什麼樣了，之後，我才知道原來本身的問題是，這個棋盤上船隻擺放的問題，涉及了最基礎的棋盤格和衍生船隻，所以後來重頭再來玩，就順利了改出了用預設棋譜隨機生成的船隻擺放功能。雖然後來這個開發涉及了不是真隨機棋譜擺放，所以庭瑋將其改成完整實際實現隨機擺放的棋譜，但這也讓我開始了對於遊戲物件關係的理解。後來，在雙人模式的設計上，我也遇到了一些棋盤位置設計上的問題與不理解，我發現了自己對於程式流程控制的是可以理解的並很快能時做出狀態的更改的部分，但對於狀態後，更改新增物件，常常遇到寫出來樣子跟我理解的相差甚遠，我嘗試藉由在過程中不定的將所有內容的 show 出來，並且放一些不同的 Resource 去強

調提示自己其中隱含的空間概念，雖然花了相對久的時間，做可能相較簡單的部分，但也理解了介面其中的轉換方法與位置數值意義。

二、 時間表

表 4 時間與進度表

周次	張庭瑋		李子儀	
	時數	內容	時數	內容
1	未知	<ul style="list-style-type: none"> 完成框架練習(分支刪除故無連結) 主選單製作 選單按鈕製作 	3	<ul style="list-style-type: none"> 課堂框架練習
2	8:35	<ul style="list-style-type: none"> 擺放階段 - 手動擺放 (1,2,3,4,5,6) 	3	<ul style="list-style-type: none"> 課堂框架練習
3	17:35	<ul style="list-style-type: none"> 製作一個新的函式以解決 CMovingBitmap::IsOverlap() 對於不為初始幀的物件會誤判的問題。 完成手動擺放 完成遊戲結束判定 完成複製棋盤功能 完成偽雙人模式 針對被擊中的船隻給予不同圖示表示 整理程式碼(1,2,3) 	10:33	<ul style="list-style-type: none"> 預設船隻擺放棋譜 建立假隨機產生棋譜 (後由庭瑋真隨機棋譜取代)

4	4:41	<ul style="list-style-type: none"> • 整理程式碼(1,2,3,4) • 隨機產生棋譜 	19:20	
5	8:47	<ul style="list-style-type: none"> • 將傳統指標改為智慧指標 • 完成新手與困難模式電腦 • 完成專家模式電腦 • 整合一些通用函式 • 完成大師模式 • 完成單人模式 	3	<ul style="list-style-type: none"> • 課堂 trace code
6	0	無	9:35	<ul style="list-style-type: none"> • 雙人模式初始模式建立
7	0	無	20	<ul style="list-style-type: none"> • 雙人模式修正使可以開始
8	4:20	<ul style="list-style-type: none"> • 顯示棋盤背景 • 製作重玩機制 	3	<ul style="list-style-type: none"> • 課堂 trace code
9	0	無	3	<ul style="list-style-type: none"> • 新 UI 製作中
10	未知	<ul style="list-style-type: none"> • 製作音效系統(1,2) 	3	<ul style="list-style-type: none"> • 新背景 UI 製作中
11	未知	<ul style="list-style-type: none"> • 修復錯誤(1,2,3) 	14:04	<ul style="list-style-type: none"> • 雙人模式正式完成
12	未知	<ul style="list-style-type: none"> • 整理資源包的路徑 	34 (粗)	<ul style="list-style-type: none"> • 更新 UI 包括船隻外觀
13	0	無	3	<ul style="list-style-type: none"> • 課堂 trace code
14	未知	<ul style="list-style-type: none"> • 完成作弊模式 	3	<ul style="list-style-type: none"> • 課堂 trace code
15			10 (粗)	<ul style="list-style-type: none"> • 報告製作
16	8:20	<ul style="list-style-type: none"> • 報告製作 	20 (粗)	<ul style="list-style-type: none"> • 報告製作
總	51+		184	備:如不包含課堂實作與固定討論則應扣除 68 小時。

三、 貢獻比例

表 5 貢獻比例表

張庭瑋	李子儀
85%	15%

四、 自我檢核表

表 6 自我檢核表

	項目	是否完成	說明為何不能完成
1	解決 Memory Leak 的問題	是	
2	自訂遊戲 Icon	是	
3	About 畫面	是	
4	遊戲中說明如何操作	是	
5	發佈(Release)版本	是	
6	承上項，該版本可以正常運作	是	
7	報告字型、點數、對齊、行距、 頁碼等格式正確。	是	

五、 收穫

張庭瑋:

1. Git 使用。與上學期的一人短期 repo 不同，這學期的專案是個長時間維護的大型專案。經過助教的教學文及實際演練一次之後，現在對於 git 不再是胡亂使用，動不動就斷頭。

2. 良好的程式撰寫風格。經由助教的一番指導，學習到了一些清理程式碼與增加可讀性的原則和方法。

3. 智慧指標的用法。在重構的過程中學到的智慧指標解決了過去每次寫指標都要加個 TODO 避免自己忘記釋放記憶體的問題。

李子儀：

1. 學會設計和實現遊戲的規則和算法：海戰棋遊戲開發，其中不僅是隨機的運氣猜測，對於人類或機器該如何思考，好好的下棋，其中是很具有規則和策略的。在流程規則上：我們需要思考如何實現遊戲的擺放階段、回合進行階段和遊戲結束階段等不同的遊戲狀態和轉換。在遊戲策略上：我們需要更進一步思考回合進行階段時，該如何突破，在更短的時間內通過或贏得遊戲。通過對遊戲規則和策略算法的設計和實現，我們學會了思考演算法和設計，並加深了對物件導向觀念的理解。

2. 瞭解物件導向程式設計的優勢：將遊戲元素抽象成物件，可以更好地組織和管理程式碼，提高代碼的可讀性和可維護性。同時，使用繼承和多型等特性可以實現代碼的重用和靈活性，使我們能夠更容易地擴展和修改遊戲的功能。

3. 團隊合作和溝通能力的提升：在專案中，我們需要與團隊成員合作，協調工作進度，討論設計和實現的細節，並解決可能出現的

問題。這個過程提高了我們的團隊合作和溝通能力，並學會了如何有效地與團隊合作，使專案順利進行。

六、心得或感想

張庭瑋:

這次課程是我繼高職畢業專題以來的首個大型專案，在此之中主要學到了如額使用 git 控制專案。那時候我們的專案由於大家都在主枝幹上進行開發，三不五時就斷頭一次，而我們當時總是重新複製一份專案到電腦上再把方才的工作重新做一次還戲稱自己是「接頭藝人」。然而在經過 OOPL 的洗禮後，現在的我可以自信地說對於 git 略懂一二，他不只是個雲端硬碟，而是個方便、強大的版本控制工具。此外有鑑於分組是面照妖鏡，尤其是長期的分組專案更是如此。身邊許多同學也因此受惠，能在開始製作專題前找到與自己步調相近的好組員。

李子儀:

在開發這次下學期實習課的專案—海戰棋遊戲，讓我們更加深入的應用了基礎程式設計和上學期物件導向程式設計等課程的演算概念和開發技術。這個專題是基於物件導向的遊戲，課程原本提供就遊戲框架，因為這使我們更容易踏入專案並知道如何著手，遊戲框架提供的架構，包括遊戲邏輯、圖形界面和用以及遊戲真的可以提供使用者 I/O 體驗的部分涉及到 CMovingBitmap 的部分方面的實現，這些都更具體地讓我們知道並感受到程式開發的內容及意義，綜合以上條列的收穫點，物件導向確實讓我們更

加熟悉了大型程式開發使用物件導向的意義，並且基於應用物件導向的概念，實踐將遊戲元素轉化為物件，並使用封裝、繼承和多型等概念來組織和設計整個遊戲。

七、 對於本課程的建議、想說的話:如果有的話

張庭瑋:

非常感謝助教這一年以來在課內、課外對大家的耐心指導。

李子儀:

謝謝老師、助教。