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**

[ㄧ、簡介 4](#_Toc137324698)

[1. 動機 4](#_Toc137324699)

[2. --分工 4](#_Toc137324700)

[二、遊戲介紹 4](#_Toc137324701)

[1. 遊戲說明 4](#_Toc137324702)

[2. 遊戲圖形 6](#_Toc137324703)

[3. 遊戲音效 10](#_Toc137324704)

[三、程式設計 11](#_Toc137324705)

[1. 程式架構 11](#_Toc137324706)

[2. 程式類別 12](#_Toc137324707)

[3. 程式技術 13](#_Toc137324708)

[四、結語 15](#_Toc137324709)

[1. --問題及解決方法 15](#_Toc137324710)

[2. --時間表 15](#_Toc137324711)

[3. 貢獻比例 16](#_Toc137324712)

[4. 自我檢核表 17](#_Toc137324713)

[5. 收穫 17](#_Toc137324714)

[6. 心得或感想 18](#_Toc137324715)

[7. 對於本課程的建議、想說的話:如果有的話 18](#_Toc137324716)

**圖目錄**

[圖 1海戰棋遊戲首頁頁面 6](#_Toc137324057)

[圖 2海戰棋遊戲模式選單 6](#_Toc137324058)

[圖 3單人模式困難度選單 7](#_Toc137324059)

[圖 4玩家一進行**擺放階段** 7](#_Toc137324060)

[圖 5玩家一完成**擺放階段** 7](#_Toc137324061)

[圖 6玩家一完成**擺放階段** 7](#_Toc137324062)

[圖 7玩家二完成**擺放階段** 8](#_Toc137324063)

[圖 8玩家一與電腦控制的敵方輪流進行**回合進行階段** 8](#_Toc137324064)

[圖 9玩家一與玩家二輪流進行**回合進行階段** 8](#_Toc137324065)

[圖 10 開啟**作弊快捷鍵**，可以一覽雙方棋譜 8](#_Toc137324066)

[圖 11 遊戲進行直到進入**遊戲結束階段** 9](#_Toc137324067)

[圖 12 遊戲進行直到進入**遊戲結束階段** 9](#_Toc137324068)

[圖 13 遊戲結束，可以選擇重玩或者離開遊戲。 9](#_Toc137324069)

[圖 14 遊戲 Class Diagram 10](#_Toc137324070)

**表目錄**

[表 1 第九組分工表 4](#_Toc137324598)

[表 2 遊戲圖形與對應介紹表 6](#_Toc137324599)

[表 3 陳列在遊戲中的音效資源： 10](#_Toc137324600)

[表 4 陳列在遊戲中的程式類別名稱以及對應說明： 12](#_Toc137324601)

[表 5 說明(1 至 17 週)撰寫遊戲的時數、工作項目與整學期的總時數。 15](#_Toc137324602)

[表 6貢獻比例表 16](#_Toc137324603)

[表 7自我檢核表 17](#_Toc137324604)

ㄧ、簡介

1. 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。

1. 分工
2. 張庭瑋
   * 客製化類別與函式
   * 通用遊戲架構
     1. 棋盤擺放與隨機配置
     2. 回合機制
     3. 音效
     4. 作弊模式
   * 單人模式
   * 報告
3. 李子儀
   * 雙人模式
   * 棋譜衍伸類別建置
   * UI
   * 報告

二、遊戲介紹

1. 遊戲說明

**ㄧ、遊玩方式：**

1. 遊戲提供單人模式和雙人模式可供玩家選擇。

2. 單人模式中，玩家將面對電腦控制的敵方。雙人模式中，兩位玩家互相對抗。

**二、遊戲規則：**

1.**擺放階段：**

1. 使用滑鼠點擊以選取船隻，將船隻放置在棋盤上的任意位置，不可重疊。
2. 按下 R（或 r）鍵可旋轉船隻。
3. 或按下 "Random Board" 按鈕，電腦將自動擺放船隻。
4. 可選取已在棋盤上的船隻，並調整至新的擺放位置。
5. 確認好棋盤船隻陣形後，按下 "Game Start" 按鈕，即開始遊戲，另外，雙人模式則是輪到玩家二擺放，當第玩家二按下 "Game Start" 按鈕，遊戲即開始。

2.**回合進行階段：**

1. 玩家一先行，選擇棋盤上的一格進行猜測。
2. 若猜中敵方戰艦（以擊中圖表示），則可以連續繼續猜測；否則（以藍色圖表示）：單人模式輪到電腦控制的敵方 ; 雙人模式輪到玩家二。
3. 電腦控制的敵方或玩家二猜測完畢後，及猜測至出現沒猜中船隻（以藍色圖表示），則再輪到玩家一。
4. 輪到玩家一時，右邊棋盤背景以棕色標示，輪到玩家二時則以灰色標示。反之亦然，輪到玩家二時，右邊棋盤背景以棕色標示，輪到玩家ㄧ時則以灰色標示。
5. 玩家需運用推理和策略來猜測敵方船艦的位置，以最快速度擊沉所有敵方船艦為目標。

3.**遊戲結束階段：**

1. 先擊沉所有敵方船艦獲得勝利，當有一方擊沉所有敵方船艦，遊戲結束。

**三、特殊技能：**

1. 單人模式中，可以加入電腦的演算系統，提供不同難度的對手，挑戰玩家的戰術能力。
2. 雙人模式中，玩家可以與朋友一起對戰，比拼推理和策略的能力。

**四、遊戲流程：**

1. 開始遊戲，選擇單人模式或雙人模式。
2. 若為單人模式，玩家進行擺放階段，放置自己的船艦。
   * 可手動擺放船艦或選擇自動擺放，擺放後仍可微調修改。
3. 若為雙人模式，依次讓玩家一和玩家二輪流進行**擺放階段**。
4. 雙方完成擺放後，進入**回合進行階段**。
5. 玩家一先行，選擇棋盤上的一格進行猜測。
   * 若猜中敵方戰艦，則繼續猜測；否則輪到電腦控制的敵方或玩家二。
6. 電腦控制的敵方或玩家二進行猜測完畢後，再輪到玩家一，並進行回合進行，直到回合結束。
7. 遊戲持續進行，直到一方擊沉所有敵方船艦回合結束，進入**遊戲結束階段**。

**五、****作弊快捷鍵：**

* 1. 使用Konami Code作弊鍵：「上上下下左右左右BA」開啟作弊模式。
  2. 作弊模式下，所有船隻的位置將顯示出來，玩家可以快速了解敵方的佈局，進而加快回合進行階段，進入遊戲結束階段。
  3. 作弊模式還提供特殊背景音樂和啟動動畫，為遊戲增添樂趣，也提示使用者知道目前在作弊模式。

2. 遊戲圖形

表 2 遊戲圖形與對應介紹表

（左側欄位代表單人模式，右側欄位代表雙人模式，置中大圖代表兩者之共同）

|  |  |  |
| --- | --- | --- |
|  | | |
| 圖 1海戰棋遊戲首頁頁面 | | |
|  | | |
| 圖 2海戰棋遊戲模式選單 | | |
| 以下進入**單人模式** | 以下進入**雙人模式** | |
|  |  | |
| 圖 3單人模式困難度選單 |
|  | | |
| 圖 4玩家一進行[**擺放階段**](#規則：擺放階段) | | |
|  |  | |
| 圖 5玩家一完成[**擺放階段**](#規則：擺放階段) | 圖 6玩家一完成[**擺放階段**](#規則：擺放階段) | |
|  |  | |
| 圖 7玩家二完成[**擺放階段**](#規則：擺放階段) | |
|  |  | |
| 圖 8玩家一與電腦控制的敵方輪流進行[**回合進行階段**](#規則：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ㄧ） | 圖 9玩家一與玩家二輪流進行[**回合進行階段**](#規則：回合進行階段)  （其中，此圖左棋盤格呈現亮棕色，為提示輪到玩家二） | |
|  | | |
| 圖 10 開啟[**作弊快捷鍵**](#作弊快捷鍵)，可以一覽雙方棋譜 | | |
|  | |  |
| 圖 11 遊戲進行直到進入[**遊戲結束階段**](#規則：遊戲結束階段) | | 圖 12 遊戲進行直到進入[**遊戲結束階段**](#規則：遊戲結束階段) |
|  | | |
| 圖 13 遊戲結束，可以選擇重玩或者離開遊戲。  （重玩即回到[**圖2海戰棋遊戲模式選單**](#圖2海戰棋遊戲模式選單)**）** | | |

3. 遊戲音效

表 3 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 (**player\_hit**玩家) | “Hit\_Marker.wav” |
| 擊中音效 (**bot\_hit**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” |
| p1\_wins打敗電腦控制的敵方且難度不是 "dark\_soul"音效 | “Defeat\_NotDS.wav” |
| p1\_wins打敗電腦控制的敵方且難度是 "dark\_soul"音效 | “Defeat\_Dark\_Soul.wav” |
| p1\_wins敗北時失敗音效A（A/B擇一） | “Defeated.wav” |
| p1\_wins敗北時失敗音效B | “Sad\_violin.wav” |
| 作弊模式啟動音效 | “CheatModeActivated.wav” |

三、程式設計

1. 程式架構

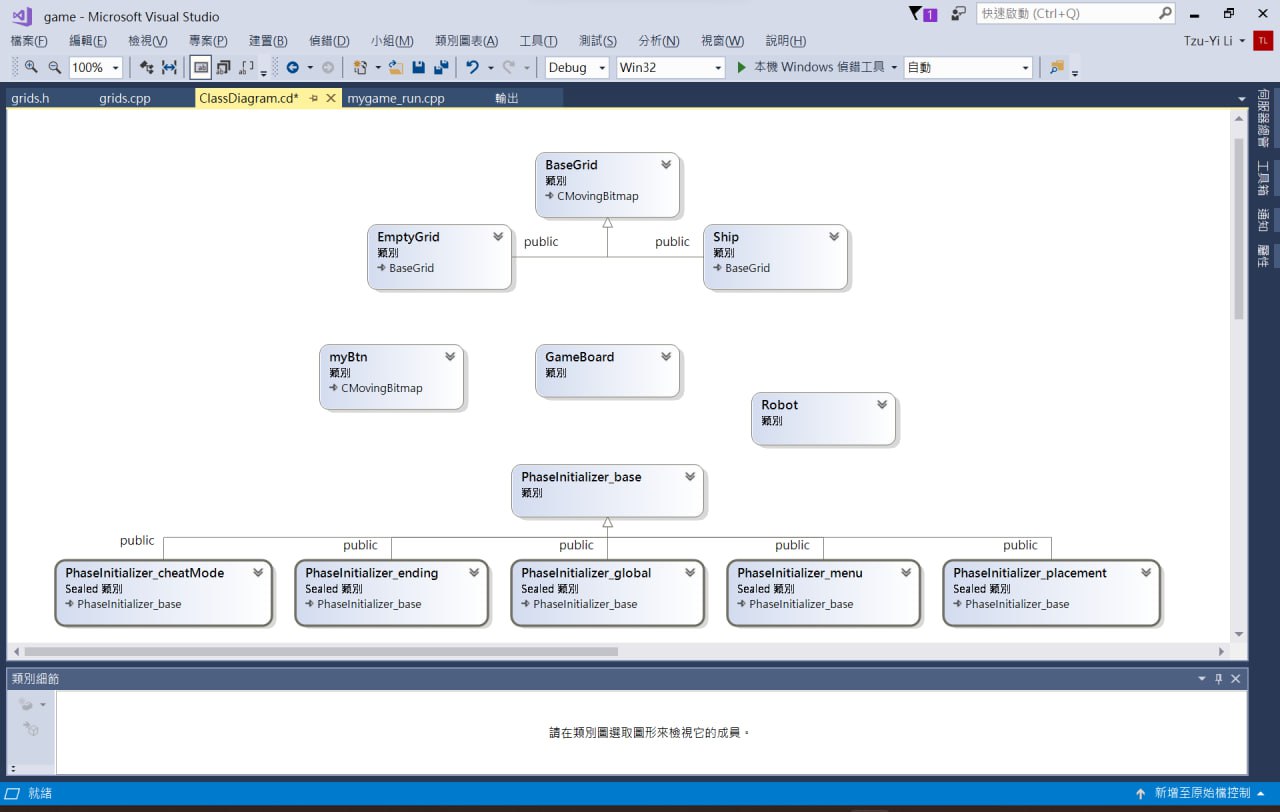


圖 14 遊戲 Class Diagram

2. 程式類別

程式類別:將你自己寫的所有 class 列表，說明用途，並標示各個 class 標頭檔

(.h)及原始碼(.cpp)的行數。

表 4 陳列在遊戲中的程式類別名稱以及對應說明：

|  |  |  |  |
| --- | --- | --- | --- |
| 類別名稱 | 標頭檔  (.h) | 原始碼  (.cpp) | 說明 |
| BaseGrid | 15 | 32 | 實作棋盤基礎方格。成員變數可儲存船隻ID、放置性和顯示旗標等資訊。 |
| EmptyGrid | 2 |  | 代表一個空的棋盤基礎方格，繼承了棋盤基礎方格的功能。 |
| myBtn | 8 | 16 | 實作按鈕。可按下、釋放和顯示按鈕等功能。 |
| GameBoard | 40 | 235 | 實作遊戲棋譜。管理遊戲棋盤的基礎方格、船艦和遊戲進程等。 |
| PhaseInitializer\_base | 20 |  | 實作Phase（遊戲不同階段或狀態）初始化器的基礎類別。（包括：共用的介面，用於初始化各個 Phase。） |
| PhaseInitializer\_global | 10 | 10 | 實作初始化全域 Phase 物件。包括：背景和游標。 |
| PhaseInitializer\_menu | 11 | 16 | 實作初始化選單 Phase 物件。包括：選單按鈕。 |
| PhaseInitializer\_placement | 13 | 19 | 實作始化棋局 Phase 物件。包括：放置時的按鈕、遊戲棋譜和提示。 |
| PhaseInitializer\_ending | 9 | 11 | 實作初始化結束 Phase 物件。包括：重新開始、退出遊戲的按鈕。 |
| PhaseInitializer\_cheatMode | 6 | 26 | 實作初始化作弊模式 Phase 物件。包括：作弊模式開啟時的圖像（如圖十）。 |
| Robot | 50 | 150 | 實作電腦類別。實現了不同難度模式下的下棋策略。 |
| Ship | 28 | 89 | 實作船艦。包含取得船艦相關資訊：方向、重置、浮、沉沒和等動作的方法。 |

1. 程式技術

關卡難度使用到之各種資料結構及演算方式：

1. 資料結構：
2. std::unordered\_set<CPoint> ：無序集合資料結構，用於儲存電腦遇到的敵艦座標，檢查是否已經射中目標。為常數時間複雜度，，新增和搜尋。
3. std::stack<CPoint> ：堆疊資料結構，用於執行深度優先搜尋（DFS）時以搜尋擊船位置（即攻擊敵方戰艦之策略）。
4. std::deque<CPoint> ：雙端佇列資料結構，用於儲存敵方艦艇的座標，以提供在"darkSoulModeFire"（我們的魔王關卡）的模式下的電腦方作弊功能。

（以上，CPoint表示二維座標中的一個點。）

2. 演算法：

1. Noob關卡（對應：新手方可嘗試）：函式 infiniteMonkeyModeFire() ，此僅重複用 randomlyPickCoordinate() ，直到找到一個未攻擊過的座標（即不在 \_map 無序集合中），然後將該座標添加到集合中並丟回，亦即，電腦控制的敵方僅亂猜下棋位置。
2. Hard關卡（對應：了解規則的玩家方可嘗試）：函式 normalModeFire() ，檢查DFS堆疊（ \_dfsStack ）是否為空。
   1. 如果為空，它使用 randomlyPickCoordinate() 選擇一個隨機座標並將其設置為上一個座標。
   2. 如果堆疊不為空，它從堆疊中取出頂部座標，將其設置為上一個座標並從堆疊中刪除。然後回傳該座標。
3. Expert關卡（對應：了解相較深入複雜的棋譜擺放的專家方可嘗試）：函式 hardModeFire() ，檢查DFS堆疊 \_dfsStack 是否為空。如果為空，它使用一種算法找到新的攻擊座標：
   1. 如果 \_map 無序集合為空，它使用 randomlyPickCoordinate() 選擇一個隨機座標。
   2. 如果 \_map 不為空，它從集合中選擇一個座標，然後根據特定的策略找到該座標的相鄰座標，這些座標必須是未攻擊過的且在（0, 0）到（10, 10），（有向的）範圍內，同樣回傳找到的座標。  
      而，另外 \_dfsStack 實作是根據getFeedback(const bool& res)

函式攻擊結果來更新

1. Master關卡（對應：想瞭解並實測今天自己的運氣是否爆棚的人，且懂遊戲規則的玩家方可嘗試。）：函式 darkSoulModeFire() 首先檢查 \_cheatCount 是否小於0，並且 \_cheatSheet 不為空。
   1. 其中，\_cheatCount 是一個計數器，表示還有多少次可以使用作弊的機會。當 **\_**cheatCount 減減到小於 0 時，代表電腦已經用完了所有的作弊次數。
   2. 其中，**\_**cheatSheet 是一個儲存敵艦座標的雙向隊列。當 **\_**cheatSheet 不為空時，代表電腦已經獲得了敵艦的座標資訊。  
      而，另外 **\_**cheatSheet實作：gatherEnemyShipCoordinates(const std::deque<CPoint>& pt) – –

* 此函式接收一個 CPoint 物件的雙向隊列，表示敵艦的座標。
* 它將該雙向隊列賦值給 \_cheatSheet 成員變數。
* 這表示敵艦的座標已經被記錄在 \_cheatSheet 中。
  + 1. 如果滿足以上兩個條件，它從 \_cheatSheet 的前端取出一個座標，將其設置為上一個座標並回傳得到的值。（亦即，電腦方使用 **\_**cheatSheet 中的座標進行攻擊，而不是依靠其他的演算策略。）
    2. 否則，用 hardModeFire() 函式。

四、結語

1. 問題及解決方法（待詢問討論）

整學期所碰到的問題及解決方法。

2. 時間表（填寫時間）

表 5 說明(1 至 17 週)撰寫遊戲的時數、工作項目與整學期的總時數。

|  |  |  |  |
| --- | --- | --- | --- |
| 週次 | 張庭瑋 | 李子儀 | 說明 |
| 1 |  | 3 | * 建立船艦物件（包含：功能及原始形象。i.e. add rotate function, ships' image） * Add myIsOverlap() 以修 CMovingBitmap 建置錯誤 |
| 2 |  | 3 | * 原始的棋盤建置（敵方棋盤先用複製的） * 遊戲現在在一名玩家的船隻全部沉沒時結束 * 船隻被擊中有不同的標記 * 開始清理程式 * 將 class 和functions 分到檔案中 |
| 3 |  | 11 | * 進行清理程式 * 建立phaseManager * 將一些初始化進程移動到類 phaseManager\_XX * 更新框架到v1.1.3 |
| 4 |  | 19 | * 創建新的遊戲棋譜給電腦方或玩家二 |
| 5 |  | 3 | * 繼續進行清理程式 * 玩家現在進行遊戲 * 可以在攻擊後連續猜 * 將所有switch-case 替換為if-elseif。 * 將所有初始化工作移至phaseManager * 隨機生成棋盤。 * Replace C-style pointers to smart pointers. |
| 6 |  | 10 | * Replace pointers involving allocating new memory with smart pointers * 猜測船隻位置功能 * 關卡一（noob）關卡二（hard）建立 * 將 phaseManagers 重命名為 PhaseInitializers 並合併 p1Turn 和 p2Turn |
| 7 |  | 20 | * 更有策略的關卡三（expert）建立 * 更外掛的關卡四（master）建立 * 雙人模式初始模式建立 * 雙人模式修正使可以開始 |

|  |  |  |  |
| --- | --- | --- | --- |
| 8 |  | 3 | * 添加gameBoard的背景 * 為按鈕添加相應的功能 * 更新 README.md |
| 9 |  | 3 | * 背景音樂製作中 * 新背景UI製作中 |
| 10 |  | 3 | * 背景音樂仍在製作中 * 新背景UI仍在製作中 |
| 11 |  | 14 | * 背景音樂完成 * 新背景UI仍在製作中 |
| 12 |  | 34 | * 新背景UI完成 * 多人模式完成 * 從選單中刪除“選項”按鈕 * 更新UI包括船隻外觀 |
| 13 |  | 3 | * 重構資源路徑 * 重構資源路徑 * 修復 ships 的 basePath 路徑錯誤 |
| 14 |  | 3 | * 開啟作弊模式 * 解決綠屏不透明問題   修復了一個導致背景音樂在某種程度上無法停止的錯誤。   * 修復導致背景音樂無法播放的bug |
| 15 |  | 10 | * Add Help/About * 期末報告製作 |
| 16 |  | 20 | * 期末報告製作 |
| 17 |  |  |  |

3. 貢獻比例

表 6貢獻比例表

|  |  |
| --- | --- |
| 張庭瑋 | 李子儀 |
| 60% | 40% |

1. 自我檢核表

表 7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 |  |  |
| 2 | 自訂遊戲 Icon |  |  |
| 3 | About 畫面 |  |  |
| 4 | 遊戲中說明如何操作 |  |  |
| 5 | 發佈(Release)版本 |  |  |
| 6 | 承上項，該版本可以正常運作 |  |  |
| 7 | 報告字型、點數、對齊、行距、頁碼等格式正確。 |  |  |

1. 收穫
2. 學會設計和實現遊戲的規則和算法：海戰棋遊戲咖發，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：我們需要更近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。
3. 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。
4. 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。
5. 心得或感想

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。

1. 對於本課程的建議、想說的話:如果有的話

謝謝老師、助教。