

1.請解釋下編譯指令，每一個參數代表的意涵 (10%)

```
g++-8 -std=c++17 -O2 -Wall -Wextra -fPIC -I./ -shared AITemplate/Porting.cpp  
-o ./build/a1.so
```

G++-8: G++是 GNU 的編譯器，用來編譯 C++的，把原始文字檔經過一些步驟變成可執行的檔案

-std=c++17: 因為 C++也是會有版本問題的，像是我們常用的 `nullptr` 也是要 C++11 才有的，所以在編譯指令加入-std=c++17 就可以使用 C++17 版本的東西

-O2: 做最佳化

-Wall:用來取得我們這次 G++所發出的所有 warning

-Wextra:他會讓你取得一些新的警告訊息

-fPIC:這是一種 code generation function，適合用於 dynamic link(適合用於 shared library 的是-fpic)

-I./:代表我們的目錄的位置

-shared AITemplate/Porting.cpp:這是一種我們的連接方式，優點是生成文件比較小，但必須使用動態庫連接

-o ./build/a1.so:這是當我們要生成目標檔案的位置跟名稱，代表我們會在 build 資料夾生成 a1.so 選項

2.請解釋 Game.h 裡面 call 函數的功能 (5%)

執行一個函式()，call 會計算這個程式執行的時間，如果超過一定時間(m_runtime_limit)，就會直接跳出去，然後我們 demo 就會當成失敗了

3.請解釋什麼是 Shared library，為何需要 Shared library，在 Windows 系統上有類似的東西嗎？(10%)

在編譯程式的時候，如果使用 static library，linker 會把.o 檔跟 static library 中需要用到的部分一起轉成.exe 檔，也就是.exe 檔本身就包含了 library。但若是使用 shared library，在生成.exe 檔時不會把 library code 一起寫進去，當.exe 檔在執行的時候才會把需要的部分拿進進程裡使用，所以 shared library 可以被很多不同的程式共用。

使用 shared library 可以避免多個.exe 檔有重覆的 object code，因此能節省硬碟空間。當被多個進程共用時，shared library 的程式碼只佔用一份空間，所以可以節省記憶體空間。當需要替換 library 時，不需要重編程式。在開發時能節省 linking 時間。

在 windows 中，DLL 是一個可由多個程式在同一時間使用的動態連結程式庫(dynamic linking library)，但在編譯時還是需要 link 一個 static library(.lib 檔)，這個.lib 檔會標明需要用的哪部分的程式庫。

4.AI Algorithm (10%)

搜索模式：

由於最小船艦的長度為 3，所以我們採用以 3 為單位的隨機走法去探查敵方船隻。

攻擊模式：

若前一發子彈被判定為 hit，則將進入攻擊模式，我們的 AI 會先找出船隻的某一邊，在借由邊的尋訪找出一台或多台的邊長，再透過計算準確找到敵艦的中心點，給他們一級必殺，直接誇爪，而當處理完發現的船艦，則會再次進入搜索狀態，尋找下一個可憐的受害者，直到遊戲結束。

移動模式：

將船隻的上下左右分別從敵人上一輪攻擊中，分別取出離它們最近的點，然後由這些最近點中，找到離船隻最遠的那個最近點，決定我們要往哪個方向走。這麼做的好處是，可以遠離那些從右上射到左下之類的 AI，但還是無法逃過宇宙最強的 random 就是了。

5.分工與進度規劃 (5%)

涂嘉哲：AI 搜尋模式、report 第四題

梁軒豪：AI 攻擊模式、Debug 超級久

陳亮孔：Gamerunner、AI 船隻移動模式、report 前 2 題

姚瀚宇：開始動畫、AI 部分 function、report 第三題

6.心得 (10%)

陳亮孔：我們最長從前一天下午 2.就拚到了隔天早上 10.，然後最後出現的 bug 因為同學的 code 太長看不懂的時候，也不知道怎麼幫上忙，挺無力的，結果 dl 前沒辦法把我們想實現的功能做出來，我做了一點幫助 debug 的功能但好像也沒有太大用處。但就算我們來不及趕上 dl，最後還是因為軒豪大大的努力，能讓我們在 demo 前做出了一個超屌的 AI。

姚瀚宇：雖然開始做之前決定這次 project 不要躺分，但真正有貢獻的部分好像就只有遊戲開始的動畫跟一些更新地圖的 function，結果最後還是被學弟 carry，感謝軒豪一直很努力 debug 還有嘉哲很多創意的想法，demo 的時候還算順利。

在電資院讀了兩年還是第一次因為打 code 看日出，這種比較大的程式需要想的很周全，只要個小差錯就要花很多時間找，感覺寫大程式的能力還有非常大的進步空間，希望將來有機會能夠寫出讓人再三回味的遊戲。

梁軒豪:雖然已經歷無數個打扣打到天明的日子，不經自問，人生還有多少扣打，但這次與組員一起熱血奮鬥的經驗真的很棒，感謝助教設計了這個分工的 project，讓我知道自己硬幹、debug 是多費時費力的事，也很感謝組員幫忙買東西，一起研究 project，讓這段時光值得回憶，但很抱歉這次在完成 project 的最後階段，我自己的 bug 沒順利的找出來，拖累整組的進度，導致嘉哲的搜尋模式沒有實作上去，下次再實作前要更努力的規劃程式架構，才不會多了很多 case 要處理或設了一堆變數產生難以發現的 bug，不過至少最後研究了一番還是有把理想實現出來，還算是給這次期末一個交代。

涂嘉哲: 這一次的 final project 是我第一次跟別人合作打扣，感覺挺奇妙的，每個人都有負責的部分，我們懶得切割成好幾個檔案去做，所以就用一個雲端的編輯器，好幾個人一起做 AI 的部分，我覺得，還是不要這樣比較好。因為雲端編輯器，我們常常有人刪減行數的時候，造成行數的變動，友人的畫面就會跑掉，真的超級白癡，會要切割檔案的方式來做是有道理的。這次真的學到這個經驗，下次的專案管理會更有效率