期末專案書面報告

組名:第 3 組驚奇四超人

組員:楊家綺B10701001, 黎子婷B10701032, 孔慶媛B10701016, 陳一伶B10701020

一、 主題

我們做的遊戲名稱是 Flunk You，這是一款射擊遊戲，遊戲的背景是小傑是一位大學生，正在面臨被當掉的危機，因為決定開始發憤圖強。所以我們要一起幫助小傑協助他畢業。

這個遊戲的玩法是玩家是小傑，天上掉下來很多 59 分的考卷，玩家需要發射「及格」這個子彈去消滅那些 59 分的考卷。玩家按空白鍵可以發射子彈，而方向鍵上下左右可以一棟小傑的位置，每打到一個 59 分及得 2 分，若小傑被 59 分打到會倒扣 5 分，玩家共有 30 秒可以幫助小傑，如果在時間內打到 60 分即過關，如果失敗就明年跟學弟妹再玩一次，可按任意鍵開始遊戲。我們有分兩個難度簡單跟困難兩種，兩個的差別在於 59 分下降的速度，按 R/r 鍵可以看遊戲規則。

我們遊戲的背景包括歡迎界面、勝利及失敗的畫面、再玩一次畫面、結束畫面都是藉由輸出字母符號印出的。以歡迎界面為例，上圖為程式碼輸出的符號，下圖為開始遊戲的歡迎界面。

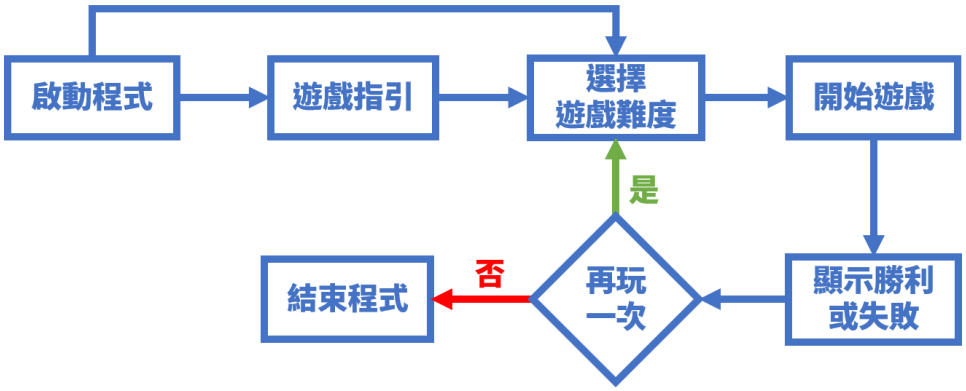
```
gotoxy(x, y  ); cout <<" ____    _      _   _   _ _ __   __  __  _____   _   _   _  ";
gotoxy(x, y+1); cout <<" |___|  | |    | | | | | \\  | | | / /  \\ \\/ / | ___| | | | | | |    ";
gotoxy(x, y+2); cout <<" | |__  | |    | | | | | |\\\\ | | |/ /    \\  /  | |_  | | | | | |    ";
gotoxy(x, y+3); cout <<" |  __| | |    | | | | | | \\\\| | |  /     /  \\  |  _| | | | | | |    ";
gotoxy(x, y+4); cout <<" | |    | |___ | |_| | | |  \\\\| | |\\ \\    /    \\ | |___| |_| | |_|    ";
gotoxy(x, y+5); cout <<" | |    |  ___| |  _  | | |   \\  | | \\ \\  /      \\|  ___|  _  |  _     ";
gotoxy(x, y+6); cout <<" |_|    |_____| |_| |_| |_|    \\_| |_|  \\_\\/       |_____| |_| |_|    ";
gotoxy(40, y+9); cout <<" developed by 驚奇4超人 C++. - 2019";
gotoxy(31, y+12); cout << "Press R/r to see game guide. Press other keys to play.";
```

二、系統設計

Flunk You 的系統設計可以分為兩個部分，一個是為了讓遊戲更完整的整體系統設計，另一個是射擊遊戲本身，我們也會在此節介紹其他程式設計細節：

1. 整體系統設計

Flunk You 整體系統設計的大架構可以用圖一的流程圖表示：。



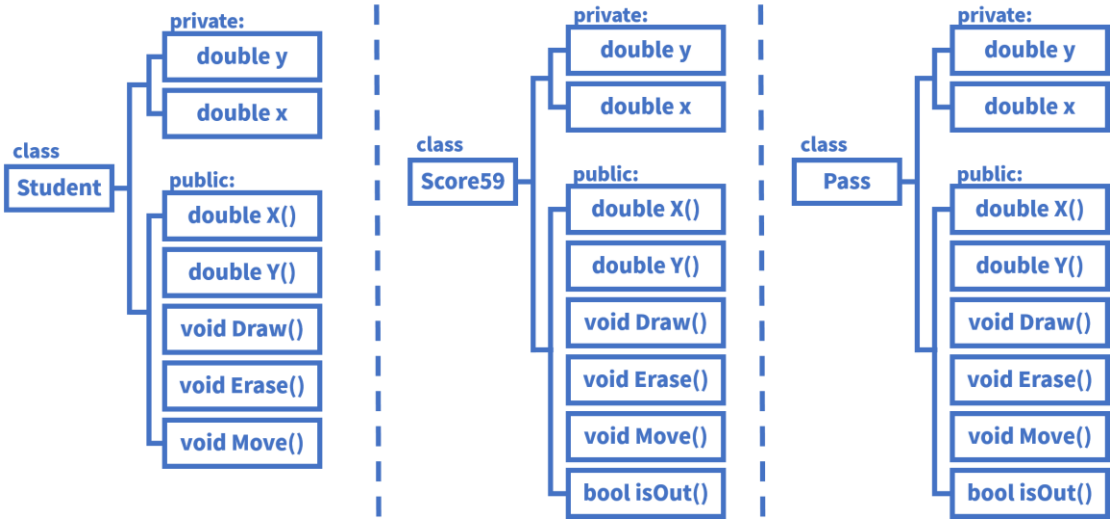
（圖一）Flunk You 整體系統之設計

因為射擊遊戲本身比較複雜，放在 main function 裡會讓 main function 顯得非常冗長，因此我們決定將射擊遊戲本身整理成回傳 bool 值的 function StartGame()。

程式開始執行後，會先讓使用者選擇要直接開始遊戲，或是要看遊戲指引，真正開始遊戲之前會讓使用者選擇遊戲難度，遊戲結束後，透過 StartGame()回傳的 bool 值決定終端機要顯示過關或失敗的畫面，接著程式會詢問使用者要不要再玩一次，如果要，程式會跳到選擇難度的地方，接者再一次進行遊戲……如此反覆直到使用者決定不要再玩，程式才會結束。

2. 射擊遊戲本體設計(bool StartGame()的設計)

StartGame()是整個程式裡最重要的 function，呼叫這個 function 等於開始進行射擊遊戲。這個遊戲裡我們設計了三種 class 物件，分別是 Student、Score59 和 Pass，對應到一般遊戲的角色是使用者控制的主角、攻擊主角的人或物，和主角可以發射的子彈；對應到 Flunk You 裡人物指的是小傑、59 分和及格。圖二是 Flunk You 的 class diagram：

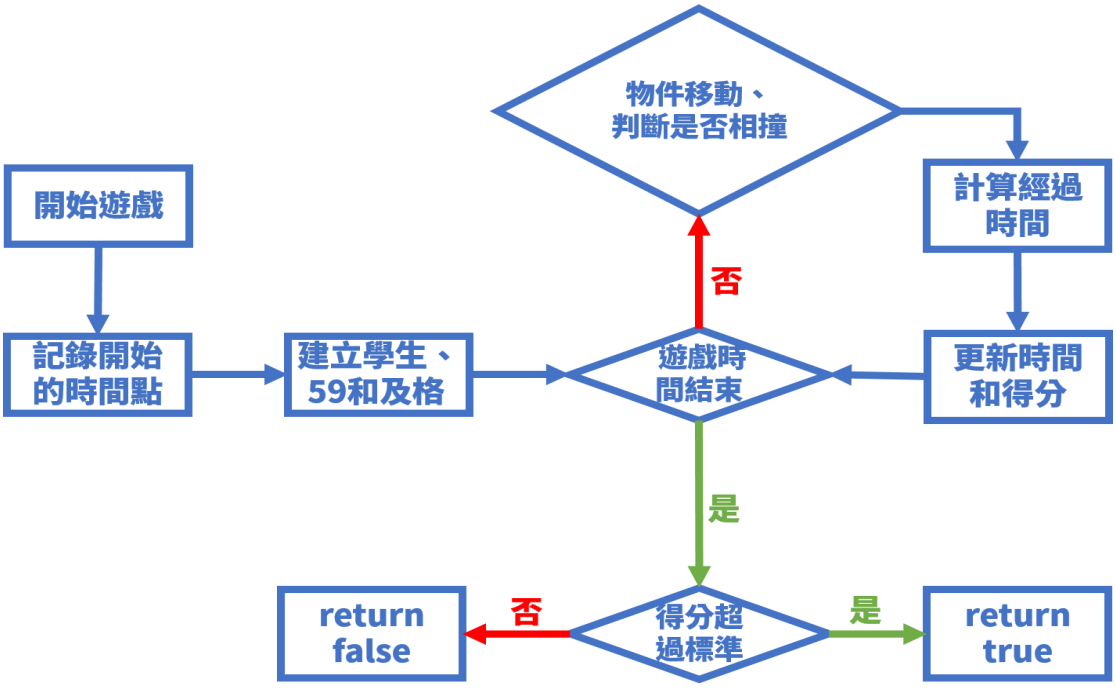


（圖二）Flunk You 的 class diagram

這個遊戲裡的 class 結構類似，都包含物件的 XY 座標、物件的繪製、清除

和移動，但是因為不同物件長相和運用方式都不太一樣，所以分成三種 class。

StartGame()的結構大致可以用圖三的流程圖表示，如欲查看 StartGame()完整的運作過程，請參見圖四：



（圖三）StartGame()的設計

Flunk You 的遊戲規則是若 30 秒內得 60 分以上，遊戲獲勝，反之則失敗，所以我們需要一個可以計算遊戲經過時間的東西，這次我們選用 C++11 的標準函式庫裡用來處理時間和日期的<chrono>，運用<chrono>裡的 seconds 和 steady_clock::now()等指令幫助我們計算時間。

遊戲一開始程式就會建立一個 Student 物件(小傑)，和用來放置 Score59 物件(59 分)和 Pass 物件(及格)的容器，因為遊戲視情況會新增 59 分和及格，而這個遊戲裡當 59 分和及格碰到邊界以後，物件應該要消失在螢幕上，甚至程式應該要刪除該物件，加上我們不會隨機存取物件，所以我們使用 C++標準函式庫裡的 list 為 Score59 物件和 Pass 物件的容器。

如果遊戲時間還沒有結束，程式就會反覆的跑迴圈，每一個迴圈程式就會讓及格移動、59 分移動、檢查 59 分有沒有撞到學生或及格、檢查玩家是否移動或發射及格、更新剩餘時間和得分等等，雖然這些動作在迴圈裡是依序執行，但因為每一個迴圈的執行速度非常快，所以使用者在終端機看到的畫面還是會覺得這些動作是同時進行的。

如果遊戲時間結束，使用者的得分大於等於 60 分，StartGame()回傳 true，代表遊戲獲勝，否則回傳 false，代表遊戲失敗。

3. 其他程式設計細節

Flunk You 的遊戲的呈現原理有點像是在畫定格動畫，就是利用在螢幕上輸

出空白或其他字元，把每個遊戲切換畫面印在螢幕上，而射擊遊戲裡每一時刻，幾乎都是先利用輸出空白，清空終端機的畫面，再每個物件該在的位置輸出在螢幕，塑造出物件在移動的效果。

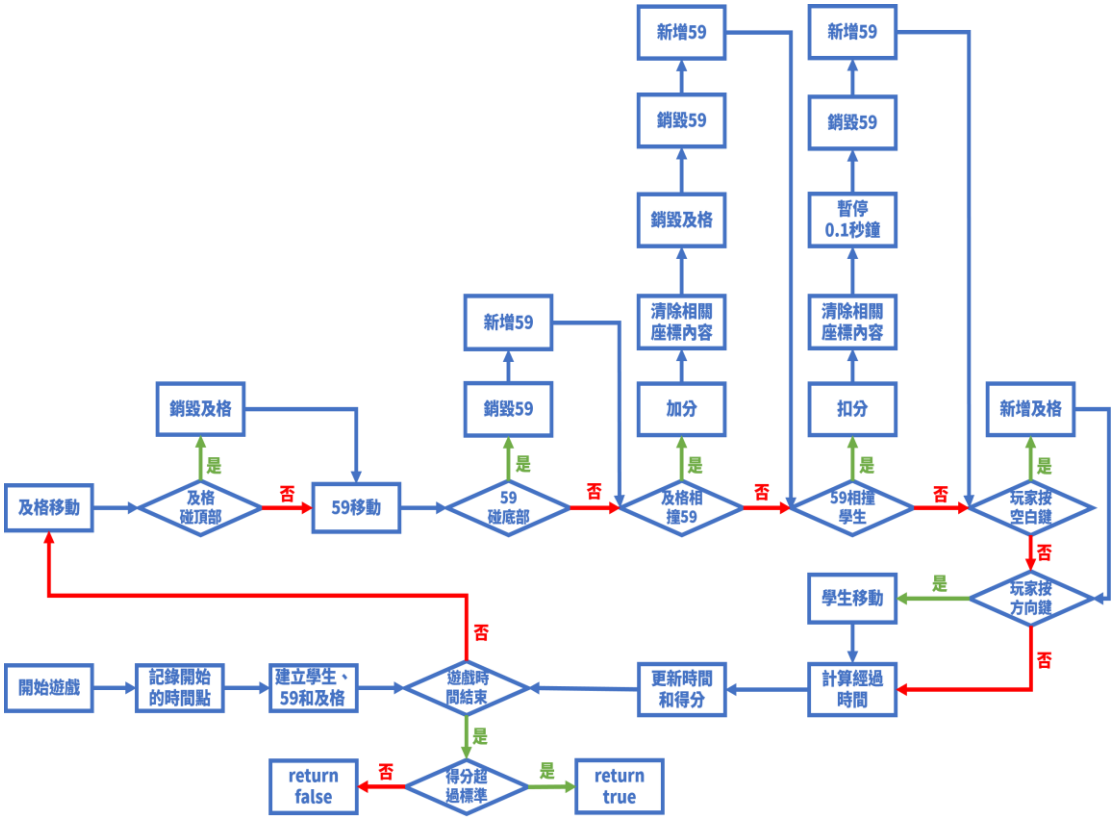
所以程式執行時有個簡單但非常關鍵的技巧是利用<windows.h>裡的 GetStdHandle(STD_OUTPUT_HANDLE)和 SetConsoleCursorPosition(hCon, dwPos)，讓電腦可以依據使用者的操作將終端機裡的游標放置在適當的地方，方便後續在螢幕上輸出東西，和這個技巧的相關操作整合為 void gotoxy(double x, double y)，讓程式碼看起來更簡潔和直覺。

另外一個值得提的細節在於，程式裡需要讓使用者選擇難度或要不要再玩一次遊戲的地方都有防呆的設計，會一直讓使用者輸入，直到使用者按下被定義的按鍵，所以不怕使用者按到不合規則的按鍵，導致程式出現錯誤或讓程式跳到無法預期的地方。

還有一個細節是我們的程式碼盡力做好模組化的工作，雖然部分的組員在一打開程式碼，看到一堆參數、變數、function 或 class 的定義，會覺得不習慣，但是因為有做好這些工作，讓我們可以在測試和維護程式時花費比較少的心力和時間，而且因為一開始就嘗試將程式模組化，所以分工可以更明確有彈性，每個人完成負責的部分之後組合起來的程式碼的衝突和問題也更少。

三、演算法

射擊遊戲本身完整的演算法可以用圖四表示：



（圖四）StartGame()完整的演算法

基本上整個射擊遊戲的演算法是一連串的 if 組合而成，只要遊戲時間還沒有到 30 秒，程式就會不停跑迴圈，要列出迴圈裡需要判斷哪些事情不會太難，對我們來說比較有挑戰的是需要更詳細更全面的拆解我們的想法，比如當 59 分和及格相撞時，我們知道 59 分和及格應該要發生什麼事然後消失在螢幕上，而且應該要加分,但是到底要發生什麼事情呢才會達到我們希望的效果？原來還可以再細分為清空終端機畫面上相撞的 59 分和及格、把該及格從 list 裡刪掉、把 59 分從 list 裡刪掉，和為了保持 59 分的總數相同，需要在 list 裡新增另一個 59 分等動作。而且同樣是相撞，如果是 59 分和學生相撞，需要做的事情又不一樣。

程式利用 for 迴圈和物件的 XY 座標檢查每一個 59 分和每一個及格是否超出邊界。如果是判斷有沒有 59 分和及格相撞，或是 59 分和學生相撞，則是利用將兩個物件的 XY 傳入 bool Collision(double x1, double y1, double x2, double y2)這個 function，透過兩個座標的相減判定兩個物件是否相撞，且由於 XY 座標值的型態是 double，所以我們設定 XY 相差多少以內，即會被程式認定兩物件相撞。