**OOP 期末專題報告**

**題目: LITTLE FUCKER 2 (Dev C)**

**電機1B-0710738-巴斯昊(50%) 電機1B-0710846-林子昊(50%)**

1. **題目描述**
2. 有選單可供玩家選擇遊戲模式(目前僅有對決模式的內容)
3. 進入對決模式後，可讓玩家(分為player 1 , player 2)自由或者隨機選擇角色(共8隻角色)進行遊玩並可選擇電腦角色數量
4. 玩家和玩家以及玩家和電腦角色之間不會重複，若玩家角色重複則後選者無法選擇該角色
5. 選擇角色後可以重新選擇角色、隨機更新電腦角色或回到選擇遊戲模式頁面
6. 進入遊戲後，以鍵盤按鍵的方式控制玩家所選角色
7. 每隻角色可進行移動、普通攻擊及特殊技能施放，每項動作都由其專屬的角色動畫
8. 角色被攻擊後會做出倒地動作，血量也會減少
9. 當其中一方血量歸零時，遊戲將跳回選擇遊戲模式頁面
10. **程式架構**

程式開始執行

getimg\_menu1(int)

\* getimage(char \*, int, int, int)

\*\* getimage(char\*, int, int, int, int, int, int)

getimg\_menu2(int)

getimg\_menu3()

Player1沒按確認鍵

printimg\_m1(int)

presskey\_1()

printimg\_m2(int)

Player1按確認鍵

Player1沒按確認鍵

printimg\_m3(int, int, int, int, int, int)

player1按確認鍵

未選完角色 未選完角色

printplayer(int, int, int, int)

presskey\_1()

presskey\_2()

printnum(int)

printnop\_m3(int)

已選完角色 已選完角色

playgame(int, int)

class名稱.control()

class名稱.getdata()

mode3(眾多參數)

class名稱.setdata()

class名稱.setdata()

class名稱.setdata()

若動畫條件成立 若動畫條件成立

print\_技能2()

print\_技能1()

print\_技能1()

class名稱.print()

print\_many技能2()

print\_many技能1()

print\_defense()

print\_down()

介面當中function的功能：

|  |  |
| --- | --- |
| void\*\* getimage(char\*, int, int, int, int, int, int) | 將大圖片切成一小張讀取 |
| void\* getimage(char \*, int, int, int) | 讀取單張圖片 |
| void getimg\_menu1(int) | 讀取menu1所需圖片 |
| void getimg\_menu2(int) | 讀取menu2所需圖片 |
| void printimg\_m1(int) | 將menu1顯示到螢幕上 |
| void printimg\_m2(int) | 將menu2顯示到螢幕上 |
| void getimg\_info() | 讀取角色資訊圖 |
| void print\_bar(int, int) | 畫出血量、魔量圖 |
| void printimg\_info(int, int) | 把欄位和角色小圖顯示到螢幕上 |
| int presskey\_1() | Player1的按鍵判斷 |
| int presskey\_2() | Player2的按鍵判斷 |
| void getimg\_menu3() | 讀取menu3所需圖片 |
| void printimg\_m3(int, int, int, int, int, int) | 將一些圖片顯示到螢幕上 |
| void printnop\_m3(int) | 將現在選擇選項顯示到螢幕上 |
| void printnum(int) | 將選擇電腦數量顯示到螢幕上 |
| void printplayer(int, int, int, int) | 將player1和player2顯示到螢幕上 |
| int bar\_detect(int\*, int\*, int\*, int\*, int\*, int\*) | 判斷血條狀況 |

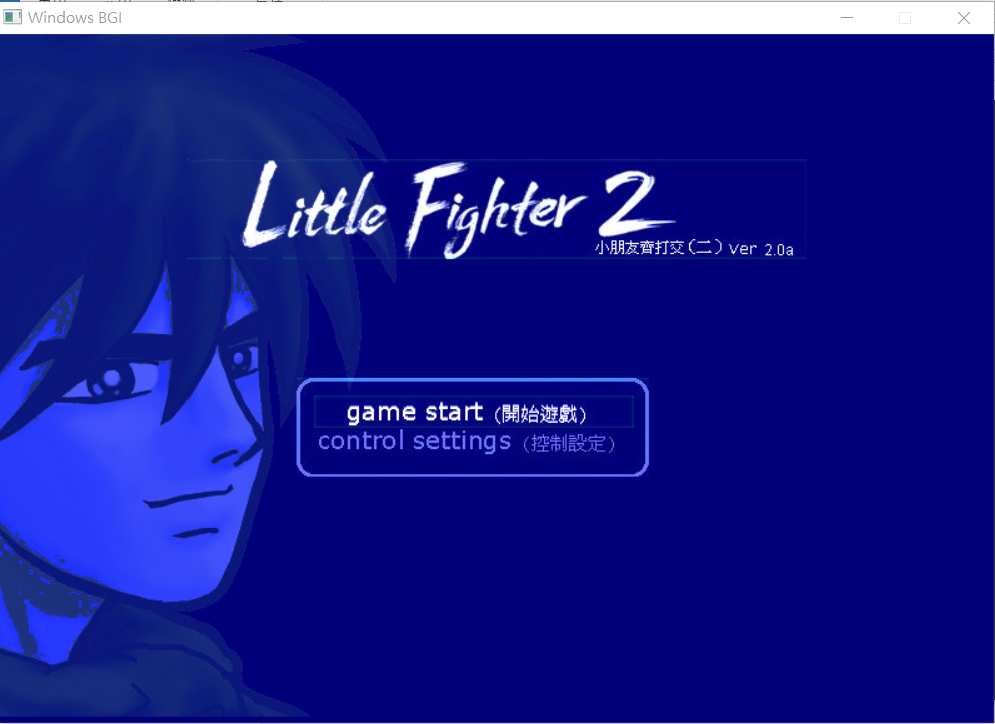
角色Class當中fuction的功能：

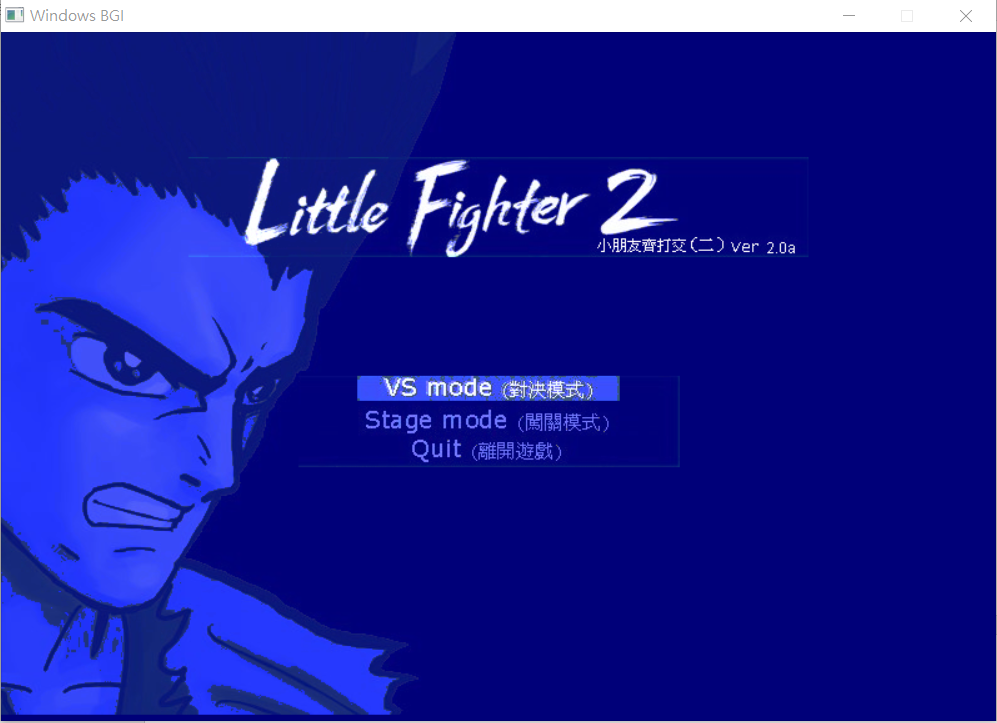
|  |  |
| --- | --- |
| void set(int); | 1.判斷當前選擇該角色的玩家為player1或player2，若為Player1則使他的角色控制鍵為’w’ ‘s’ ‘a’ ‘d’ ‘c’ ‘f’且角色的起始方向為向右  Player2則使他的角色控制鍵為’上’ ‘下’ ‘左’ ‘右’ ‘/’ ‘;’且角色的起始方向為向左  2.決定角色的起始位置 |
| Class名稱(); | 1. 此為constructor，用來初始化各個位於class當中的變數，讓接下來程式的運行不會發生錯誤。 |
| void control(); | 1. 判斷當前鍵盤上的那些按鍵被按下 2. 判斷角色當前為向左還是向右 3. 根據按鍵的排列組合，以及角色當前的方向來打開相對應的旗標，例如，當角色的方向向右且同時按下上鍵及特殊功能鍵的話，則會打開角色動畫旗標及招式動畫旗標，讓之後的void print()函式知道接下來要做出那些動畫 4. 紀錄按鍵被按個數 |
| void recordimage(); | 1. 讀取該角色各種動畫的圖檔，以進行角色的動畫 |
| void print(); | 1. 製作出角色的移動、普攻動畫，另外在此函式中也會呼叫特殊技能動畫以及角色被擊倒動畫的函式，故在主程式當中，只要呼叫該函式即可做出所有動畫的判定 |
| void print\_技能(); | 當角色特殊技能被觸發時，即可透過這個函式將角色的動畫產生，主要架構為三部分   1. 角色做出技能的動作 2. 技能產生的動畫 3. 角色技能動畫結束的條件判定 |
| void skills \_recoed(); | 1. 當角色重複使用同一招技能時，此函式會繼承一開始使用的技能的數據，例如該技能的座標、最終應該飛向哪裡及目前應該要顯示哪一張圖片，讓接下來的print\_many技能();函式能夠打前一個的動畫繼續做完 |
| void print\_many技能(); | 使角色能夠重複投擲出技能(讓角色同一招技能可以重複出現在場上)，主要的架構為：   1. 根據接收到的技能數據做出技能動畫 2. 角色技能動畫結束的條件判定 |
| void print \_down(); | 1. 當角色被另外一位玩家的技能所攻擊到時，此時這個函式會把做出被擊倒動畫的旗標打開，並且開始做出一連串被擊倒的動畫，若動畫做完，則把相關的變數設為初始值，以便接下來遊戲的運作不會出現問題。 |
| void print \_defense(); | 1. 當角色被另外一位玩家的普通攻擊所攻擊到時，此時這個函式會把做出抵禦攻擊動畫的旗標打開，並且開始做出一連串防禦的動畫，若動畫做完，則把相關的變數設為初始值，以便接下來遊戲的運作不會出現問題。 |
| void getdata(許多參數); | 1. 由於每一個角色都是獨立的class，若要讓兩個角色對於彼此的相互關係有所反應，例如何時該被擊倒及扣血，則需要從class當中取出data至main主程式當中，以便於判斷該讓角色做出那些動畫，至於該如何去判斷，則會在主程式當中有額外的函式去執行。 |
| void setdata(int,int); | 1. 當判斷完角色之間的相互關係後，改變後的參數值會經由此函式送至class當中，以改變class當中的許多旗標，例如判斷完角色被技能擊中後，會經由此函式打開class當中被擊倒的旗標，以便class做出被擊倒的動畫。 |

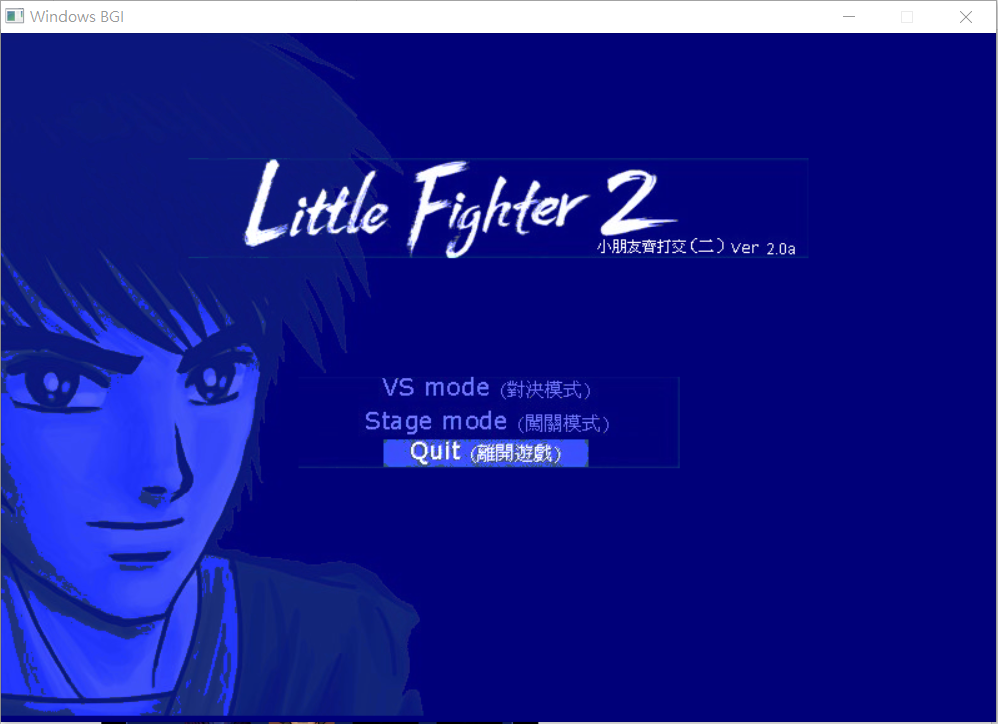
判定角色是否被擊倒的function的功能：

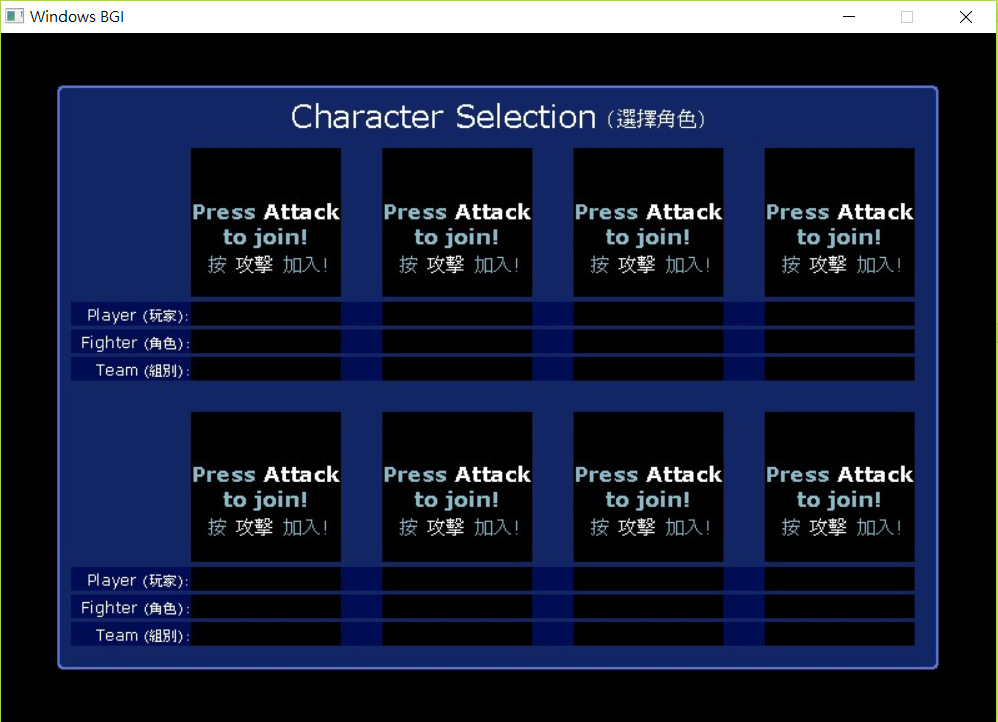
|  |  |
| --- | --- |
| int mode3(參數); | 1. 接收由主程式傳來的參數來判定角色與招式之間是否有接觸，即角色是否有被擊倒，若角色被擊倒則會改變某些主程式當中的參數，在透過class當中的   void setdata();來改變class當中的旗標，使被擊倒的動畫能夠順利進行。 |

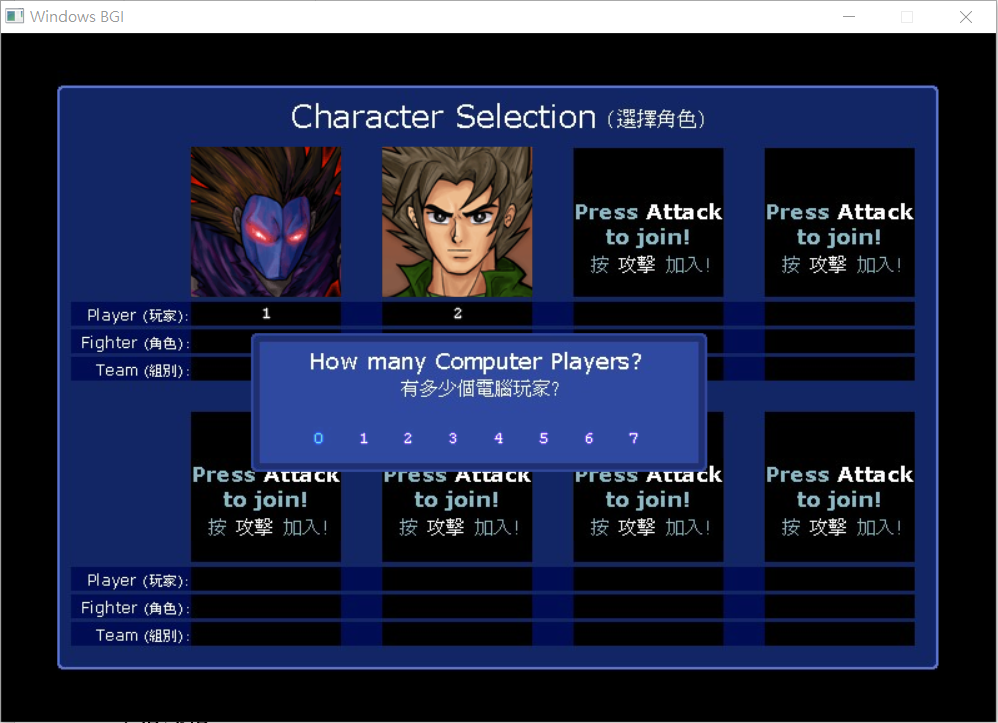
1. **執行結果**

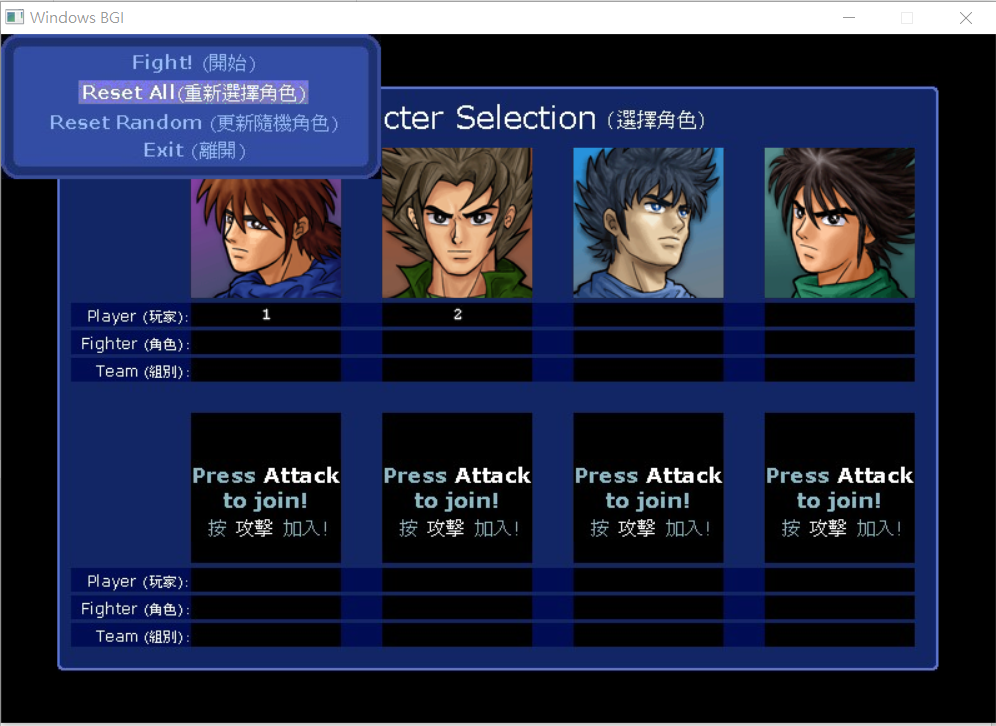
程式剛執行時(控制設定目前沒東西)

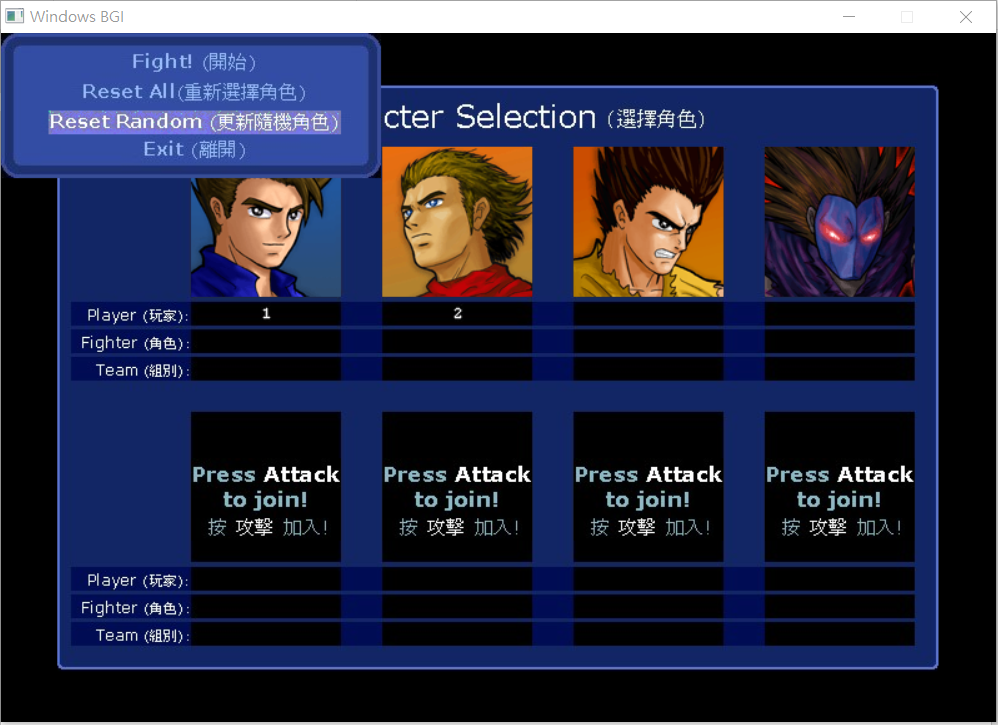
 按下“開始遊戲”後即可選擇遊戲模式(目前僅有對決模式)

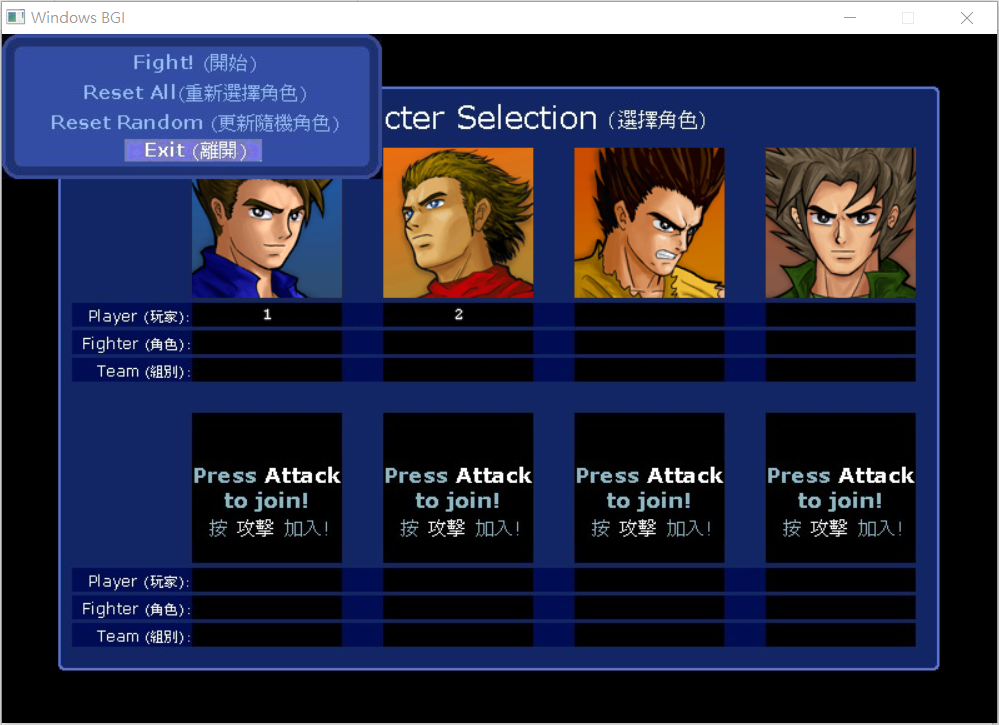
 按下“離開遊戲”即可關閉程式

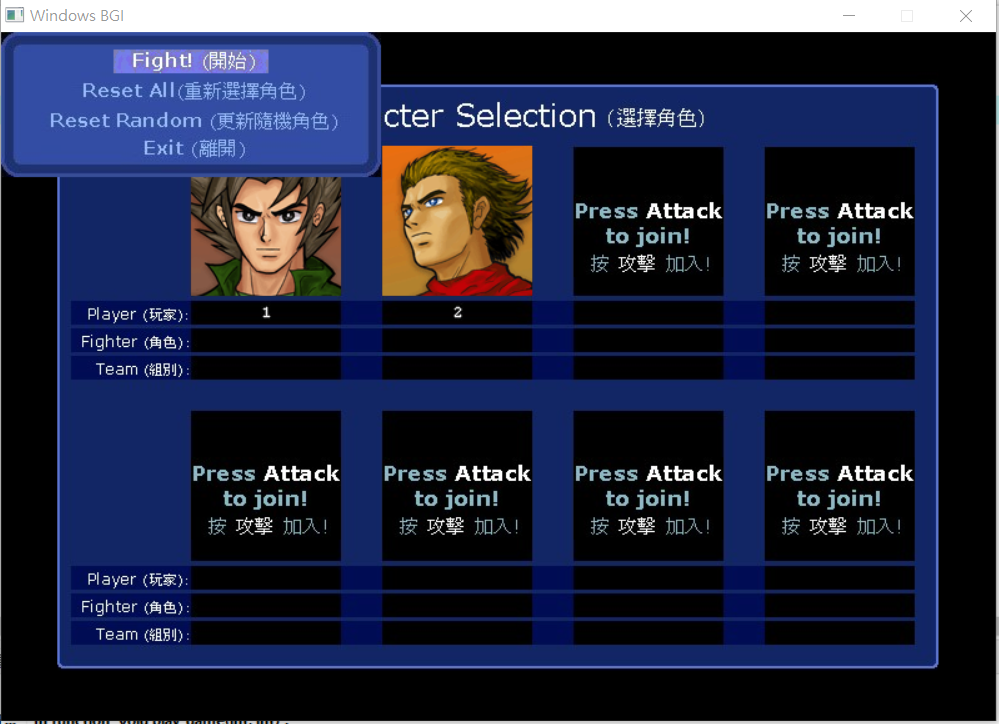
 按下“對決模式”後即可開始選擇角色

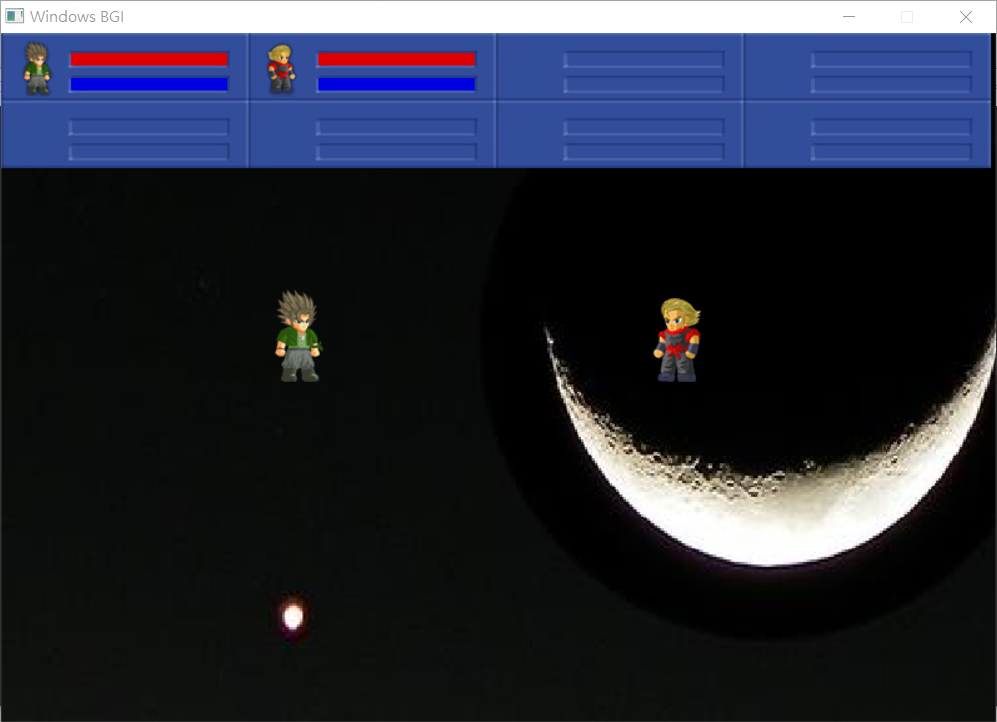
 當player1和player2選完角色後，可選擇電腦角色數量(目前僅供選擇，遊戲中並不會出現)

 選擇完後，如果不滿意當前玩家角色可按“重新選擇角色”就能再選一次角色

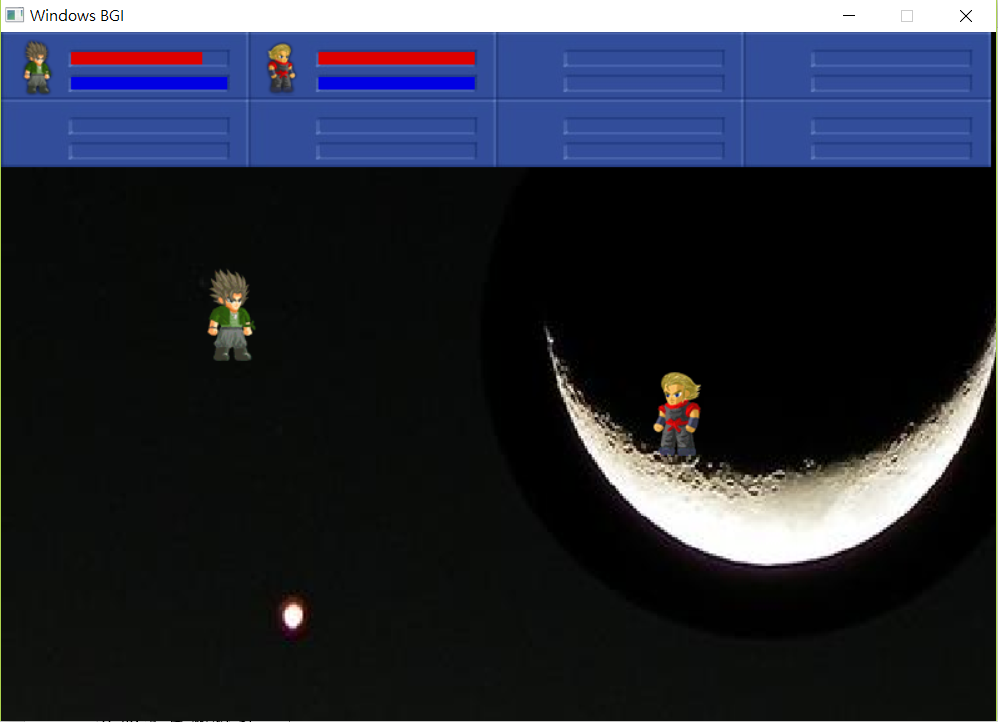
 如果不滿意電腦角色可按”更新隨機角色”來重新搭配對戰組合

 按下“離開”則會回到選擇模式的選單

 按“開始”後就可遊玩囉~

 進入遊戲後的畫面(目前遊戲未設邊界)

 每隻角色都有兩個自己獨特的絕招

 角色被攻擊後也會損血

 當其中一方死亡後遊戲將回到選擇遊戲模式的選單

1. **專題關鍵**

介面當中關鍵點：

1. 按鍵的判定: 因為按鍵判定寫在迴圈裡所以可能玩家只按下一次W就會被判定成按了好幾個W，這會導致無法選擇選單選項甚至是選角方面的困難。而我的解決辦法是多設一個變數來保存上一次的按鍵然後當我這次的內容又和上一次相同時我就不再產生相對應的動作，例如: 假設按W代表0而不按代表1，然後玩家按下了W則在下一個迴圈變數會存0可是W這個訊息還會持續好幾個迴圈所以在這幾個迴圈內就不會多跑好幾次W的動作了，當玩家放開W鍵後變數會存1此時如果在按下W就可正常判定為按下兩次W
2. 每一層頁面的轉換: 當初設計的時候是可讓玩家按下返回鍵的，所以要怎麼正常的控制每個頁面之間的轉換就需要一些判斷。我的解決辦法是，把每一頁都編一個數字，所以當我要返回到某一個頁面時我只要透過變換數字就能到達我要的頁面

Class當中的關鍵點(每個class代表一隻角色)：

* 1. 專題製作初期
     1. 角色基本動畫該如何透過鍵盤產生：

由於之前上機練習時，動畫的產生只有對於open哭哭的動圖有所印象，然而open哭哭的動畫只是單純把圖片在不同的座標給顯示出來，因此還不知道該如何將之搭配鍵盤來由盤操控，因此在設計遊戲的最一開始就進入了沉思。

最後我想出的方法，就是設定一個button\_down[ ]的陣列，透過while迴圈來不斷掃描鍵盤的那些按鍵有被按下，當該按鍵被按下時，則button\_down[]會是1反之則為零，因此我就可以透過這些參數來決定要如何操控角色。

* + 1. 角色動畫的方向以及當前要顯示哪張角色移動動畫圖片的問題：

由於我們必須了解到角色當前的方向為何，才可以決定我的動畫要怎麼樣子去製作，因此我設定了direction[ ]陣列，根據上述的button\_down[]值來改變dorection[ ]當中的值，以確保我當前角色的方向為何；再來就是當前要顯示哪張動畫圖片的問題了，由於當使用者按下方向鍵時，動畫才會產生，反之必須讓角色待在原地不動，這就產生了一個問題──我該如何知道當前要顯示角色的第幾張圖片？最後我分別針對不同角色的動畫，設計了向右及向左的count變數來決定當前應該顯示哪張圖片，搭配上述所說的button\_down[]陣列，加上額外的flag(button\_down有值時將旗標打開，count開始記數，反之則將count初始化，flag歸零)，即可做出判定。

* + 1. 如何讓角色能夠邊移動邊普攻：

在一開始的設計當中，角色的移動及攻擊動畫事實上是分開的，然而，這其實不太符合現今大部分的遊戲設計「角色當然要能夠邊移動邊普攻阿」，所以這讓我思考了一段時間，最後我發現事實上，我先前有設計button\_down[ ]陣列，我可以透過偵測當前有哪些按鍵被按下以及按鍵被按下的數量來決定要如何邊移動邊普攻，因此我算出了，每個邊攻擊邊移動的排列組合，配合之前所設計的direction[ ]和count變數來做出角色的動畫。此外，過程當中當然必須設計許多的flag來幫助我們區別當前要顯示移動的動畫還是攻擊的動畫，這過程當中的許多小巧思真的可以說是讓人一個頭兩個大。

* 1. 專題製作中期
     1. 技能動畫的產生──如何使動作先做完再將技能給投擲出去：

這個問題其實困擾我很久，因為我們程式的動畫是透過while迴圈不斷的去跑來偵測當前應該要做那些動畫，就像上述所說的，動畫製作的方式是透過button\_down[ ]來分辨當前的按鍵是否有被按下，例如移動時，while迴圈就是不斷偵測到button\_down[]的值為1才去顯示角色移動的動畫，一旦button\_down[ ]為0，動畫即停止，然而投擲技能這個指令卻不一樣，當程式偵測到某些按鍵的組合時，它就必須不斷執行投擲技能動作的動畫以及個別技能的動畫，同時間當投擲技能動作還沒結束時，它還必須防止使用者按其他移動指令或是其他技能的指令，來影響整個技能動畫的運行。

最後我想出來的辦法，就是在control();副程式當中增加動作旗標以及技能旗標，當成是偵測到我要發出技能時，旗標全部打開，此時會進到print();副程式當中將動畫給顯現出來，當投擲技能的動作還沒做完時，也就是count還沒到達某些特定值時，角色會不斷地做出動作，此時與我是否有再按其他按鍵無關，當動作做完後，其動作旗標關閉，然而技能旗標還是打開的，所以此時既能將會射出，直到技能大於或小於某些座標時，技能才會被消滅，並將技能旗標給關閉。

* + 1. 如何讓技能與技能之間不被彼此給影響，例如同時做出技能A與技能B的動作：

承上所述，我已經有了不同的技能動作旗標，以及技能旗標，我就可以透過這些技能是否打開來限制其他技能的產生，例如當A技能被執行時，A技能的相關旗標都被設為1，此時當我按下其他技能的相關組合後，程式並不會對其做出反應，這是由於我其他技能必須要在技能A的相關旗標為0時才能動作，因此整個角色的動畫才可以順利進行。

* + 1. 如何投擲出第二發技能：

在說明此問題前，我必須先說明技能如何產生及如何消滅的。當程式偵測到使用者按下某些組合的按鍵時，程式會在void control();副程式當中紀錄角色當前的座標，來判斷當技能座標大於或小於角色投擲技能座標時的某些數字時才將其消滅，此時就產生了一個問題──每當我按下技能時，技能的相關數據都會被初始化，等於是說第一個技能還沒跑完，它的座標又被初始化，再次被角色給投擲出去，這其實是滿荒謬的一件事情的。

所以我就不斷的去思考該如何避免這樣子的一個問題。當初之所以會把單獨角色的技能和單獨角色的其他動畫全部寫在同一個class裡(也就是一隻角色是一個class)，目的就是方便class裡值的傳遞，沒想到卻產生這樣子的一個問題，這真的很讓人頭痛。最後我想出的方法就是建立skills\_record();副程式，以及新增number\_of\_(技能名稱)變數，當我程式偵測到我施放同一招技能時，它會先進去skills\_record();副程式先判斷當前該技能位於場上的數目，若場上的數目=1也就是說我必須讓兩顆技能同時存在於場上時，它就會將前一顆技能的相關參數給記錄下來，讓爾後的print\_many(技能名稱);副程式能夠繼承前一顆技能的數據，並使它繼續於螢幕當中顯示。這也是我們在第二次demo前絞盡腦汁所想出的方法，詳細細節這真的很燒腦。

* 1. 專題製作後期
     1. 如何判定角色被別的角色技能或普通攻擊所攻擊之程式設計構想：

這部分其實也是整個遊戲最關鍵的所在，一旦一個遊戲無法判定甚麼時候被攻擊，那這個遊戲可以說是如同虛設了，要達成這個條件，我們必須運用到兩個位於class當中的副程式

void getdata(參數);和void setdata(int,int);以及額外的副程式void mode3(參數);

首先我們必須先從class當中，把需要被判定的參數值抓到main.cpp當中，再將參數傳遞至int mode3();副程式判定，最後再透過void setdata(int,int);改變class當中參數的值。

接下來討論如何設計程式，由於每個角色的技能都不相同，有些有兩個投擲技能，有些只有一個(所以有投擲技能的角色參數會多很多，因為他可以投擲很多發)，這造成當不同組合的角色進入到遊戲時，每個組別所需要注意到的參數也不同，因此我們在程式設計初期，原本想要使用最笨的方法，也就是使用排列組合，將每組角色的招式判定副程式給寫出來，然而我們有八隻角色，這就說明我們必須寫出28種組合，這實在是不切實際。

最後我想出了一個通式，由於每個角色扣除掉普通攻擊的話只有兩個技能，雖然有些角色有投擲技能，造成與沒有投擲技能的角色參數比較上的不對稱，然而我卻可以使沒有那麼多投擲技能的角色在參數傳遞時傳一個空值，例如在傳投擲技能座標時，將之設為

(-1000,-1000)，這就不會成角色參數比對數量上的不對稱，且這並不會影響角色被攻擊的判定，透過這個方法，我將原本的28個組合省略成1個，提高了很多效率及姐省很多記憶體空間。

* + 1. 如何判定角色被別的角色技能或普通攻擊所攻擊之演算法：

由於每個角色及每個招式都有其參數的座標，且每個角色的圖檔皆為長方形，所以演算法的構想為，當技能被按下且兩張圖片的圖檔有重疊時，即進行角色跌倒的動畫。

* + - 1. 當招式相對於角色在於右上方時：

0<=當招式x座標減去角色x座標的值<=角色寬度時且

0<=角色y座標減去招式y座標<=招式高度

* + - 1. 當招式相對於角色在於右下方時：

0<=當招式x座標減去角色x座標的值<=角色寬度時且

0<=招式y座標減去角色y座標< =角色高度

* + - 1. 當招式相對於角色在於左上方時：

0<=當角色x座標減去招式x座標的值<=招式寬度時且

0<=角色y座標減去招式y座標< =招式高度

* + - 1. 當招式相對於角色在於左上方時：

0<=當角色x座標減去招式x座標的值<=招式寬度時且

0<=招式y座標減去角色y座標< =角色高度

1. **心得感想**

林子昊：

當初以為LF2應該很好做，結果在做的時候才遇到很多瓶頸，包括一些找不出哪裡有錯的bug還有不小心多重定義到一些變數或函數名稱，而且還要在有其他課業壓力的情況下把專題做出來，實在是耗了我很多的心力。但是能進到決賽我覺得已經是對我的努力有很大的肯定也讓我花的這些時間沒有白費掉，最後也得到了三用電表這個不錯的獎品(私心認為這個比行充實用多了)，整體而言我覺得相當值得而且很有成就感，我也很感謝我的partner，畢竟他做的是這個遊戲的核心部分。

巴斯昊：

首先必須強調的，就是這學期的oop專題真的讓我的大一下生活過得很充實，老實說我們學了那麼多的程式，最後最後也是最為重要的，就是如何去應用它，將所學給實現出來，在我知道蔡媽的oop課程有專題時，我其實有點小興奮，因為終於有機會能夠將自己的知識化為實體，呈現在大家眼前，這不僅是一個能夠表達自己的一個機會，也讓我提前感受到工程師們工作的樣子，就好像將他們的生活寫照投影在自己身上，彷彿我也可以透過這個專題有所成長，像是升級一樣。

萬事起頭難，要邁出如何設計專題的第一步，真的是一個很痛苦的事情，當時每天的工作就是上網找相關遊戲的資料、複習及研究阿法所教的graphic mode、看著自己在上機課時以跟阿扁一樣每秒鐘手抖六下所錄製操控graphic mode的影片，這成為了一個無線迴圈，就像C一樣，你在製作專題的路上一旦判斷式還尚未成立，自己就會不停的繞啊繞，就像輪迴一樣，考驗人的意志力。

最初獲得的成就感，就是在第一次demo前時，我讓我們的第一隻角色可以正常移動及普攻，當它能夠正常運作的瞬間，我真的知道為什麼我媽當初生我時要哭了，因為那滿滿的呵護及成就感真的化為現實，也給我繼續製作專題的動力。雖說獲得暫時性的成果，在當時距離完成專題還很遙遠，第一次demo過後，就產生了和我媽一樣的想法：「我還要把這東西養育成人……。」

爾後的時間，就是不斷地期中及期末轟炸，在被炸之餘，我還要保證我家小孩不會被炸死，真的可以說是為母則強阿。這個專題就像水蛭一樣，不斷的從我身上吸取大學生的青春，當別人在夜衝的時候我要打code、當別人夜唱的時候我也要打code，最可惡的就是當別人在把妹的時候我也要打code，這真的是一個惡魔，總算知道人家為甚麼說求學生涯不要生小孩了，不然真的不會有自己的時間。那時真的覺得自己隨時會爆炸。

基本上期中到期末這段時間我得了產後憂鬱症，一想到要做專題，就覺得人生好難，可是它又是一個自己必須要承擔的一個責任，每天自己內心都在與天人交戰。這段時間的時間分配真的很重要，每天都在想要怎麼把資源的效益發揮到最大，以換取與別科之間的平衡，到最後只能壓縮自己的睡眠時間，不管是心理還是身理，真的都是一大挑戰。

第二次demo過後，我們這組獲得了晉級，看到這個消息其實還滿欣慰的，自己做出來的成果，獲得了助教們的肯定，一定的程度上，告訴自己努力並沒有白費，然而也說明了我們必須在期末最忙的這一段時間內，把作品提前做完，以參加最後的競賽。基本上最後一個星期可以說是每天都可以看的到日出：「今天日出的陽光好像比昨天橘ㄟ」是這一星期的寫照，直到最後一天整天都沒睡，目的就是要及時去完成這個專題，陪它走完最後的一哩路。

競賽當天晚上，自信滿滿地向別人敘述著我們的成果，雖說我們不是最優秀的，但我知道我們已經盡力，不管結果如何，都是最好的結束，至少我有參與過，知道了整個設計的流程，製作專題的學習，可以說是最扎實且有用的。

最後感謝老師給我們這樣子的一個機會來學習這門課程，雖說過程有時真的很艱難，甚至有想過放棄的念頭，但最後也結出了果實，這成就感真的不是其它事物可以輕易比擬的。突破極限才能超越極限，真適合來形容電機人的生活寫照啊。

1. **參考資料**

LF2官方網站 <http://www.littlefighter.com/index_b5.html>

色碼表 <https://www.toodoo.com/db/color.html>