|  |  |  |
| --- | --- | --- |
| Team No：15 | Team Name：TING & TING | |
| Project Title：小艾走不出去的迷宮 | | |
| Name：留鈺婷 | | ID：111062320 |
| Name：陳璟婷 | | ID：111062112 |

**一、 設計概念**

在這次Project中我們希望應用課堂上所學到的東西，設計出一個迷宮小遊戲。

**（一）Background Story**

三隻小艾散步時突然間被神秘的力量帶入一個異世界。眼前豎立起一座古老而神秘的迷宮，必須避開怪物的追蹤，跟循指示找到三把鑰匙，方能從唯一的出口離開。

每個小艾面臨不同的挑戰，他們必須借助玩家的智慧和冒險精神，才能成功逃離這個異世界。小艾們是否能在怪物的追捕中順利取得鑰匙，找到迷宮深處的出口？這全取決於玩家的指引！

**現在，帶領小艾們踏上未知的冒險，逃離異世界！**

**（二）Description**

１、**Title**可以從**Help**查看遊玩方式，或是選擇關卡，只有通關前一關才會解鎖新的關卡。

２、玩家需要用鍵盤操控角色在迷宮中尋找鑰匙、電燈開關和出口，共有三個關卡。

（１）**Stage1: Beginner’s Luck**

　　　一般關卡，只需找到三把鑰匙和出口即可通關。

（２）**Stage2: Live in the Light**

　　　除了玩家和電燈開關周圍，其他地方都是暗的。

　　　玩家必須先找到電燈開關，再找到三把鑰匙和出口才能通關。

（３）**Stage3: Be Free**

　　　有一位 boss會根據玩家位置跟隨玩家，若被抓到會減少一條生命值。

　　　玩家需躲避 boss並找到鑰匙和出口才能通關。

（４）若生命值歸零會進入**Game Over**畫面，通關則會看到**You Win**畫面。

３、通關後可以選擇要進入下一關或是回到**Title**；失敗後可以選擇重新挑戰或回到**Title。**

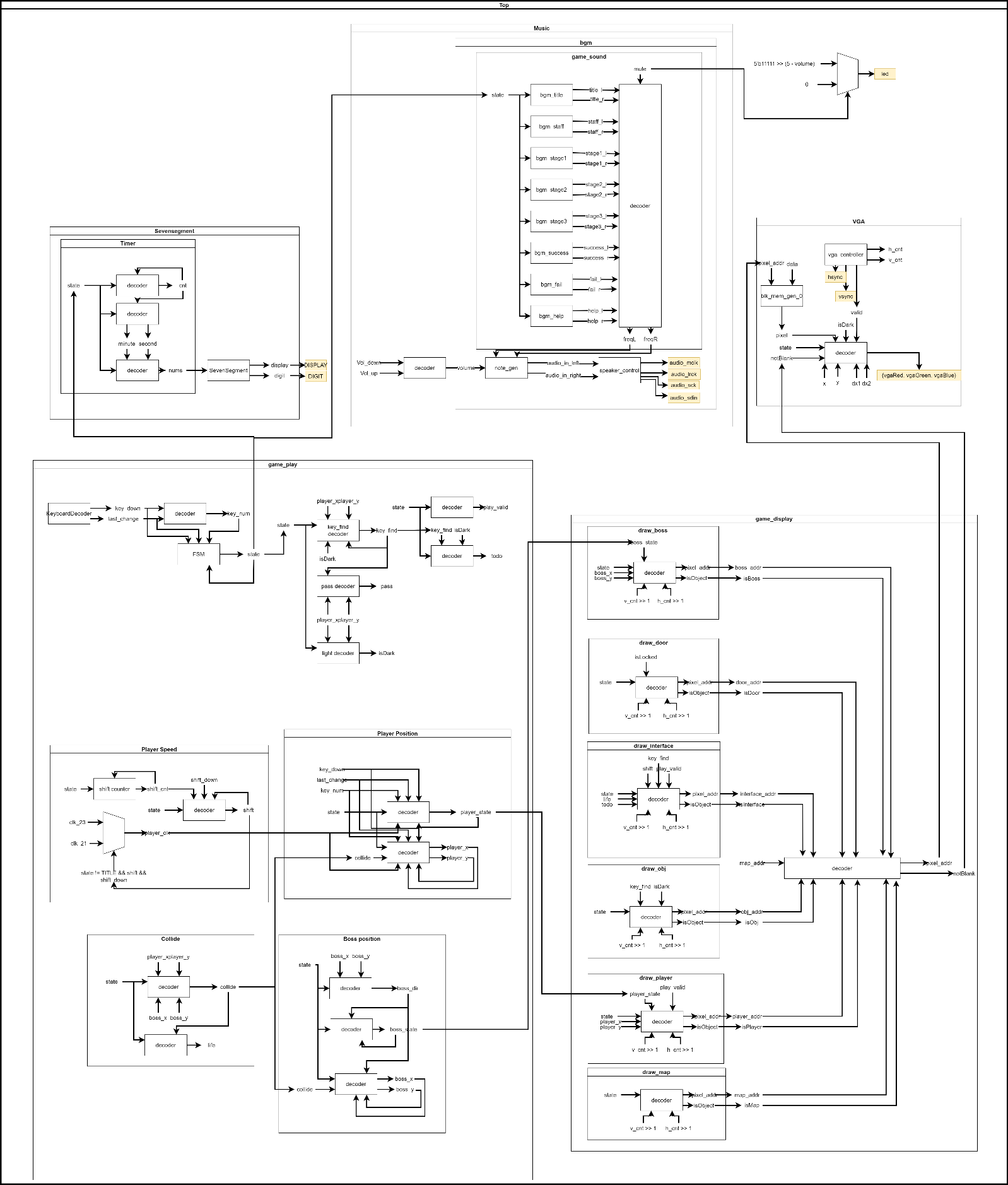
**（三）I/O signal specification**

|  |  |  |
| --- | --- | --- |
| **Connect** | **Name** | **Description** |
| W5 | clk | Clock |
| BtnC | \_rst | Reset |
| BtnU | \_volUP | Volume Up |
| BtnD | \_volDOWN | Volume Down |
| SW 14 | \_mute | Mute |
| LED 0 ~ 4 | led | Volume Indicator |
| Pmod JB 1~4 | audio\_mclk、audio\_lrck、  audio\_sck、audio\_sdin | Pmod I2S |
| 7-Segment | DISPLAY、DIGIT | Time Record |
| pin N19, J19, H19, G19 | vgaRed | Screen Display |
| pin D17, G17, H17, J17 | vgaGreen |
| pin J18, K18, L18, N18 | vgaBlue |
| pin P19 | hsync |
| pin R19 | vsync |
| C17 | PS2\_CLK | Keyboard |
| B17 | PS2\_DATA |

**二、 架構細節及方塊圖**

**（一）Block Diagram**

１、Block Diagram



２、Explanation

Top module中主要分為遊戲控制、螢幕顯示、音樂和時間顯示。

（１）遊戲控制：game\_play

**game\_play**利用keyboard轉換state和改變player\_position，shift切換player\_clk以變換player的速度；key\_find decoder中更新key\_find數量，並以此判斷是否pass；light decoder判斷stage2 中的isDark的值。在stage3中，偵測player是否與boss相撞，以改變player和boss的位置以及遊戲狀態。

一張含有 文字, 螢幕擷取畫面, 圓形, 字型 的圖片

自動產生的描述

螢幕轉換的機制

在TITLE中，如果按下[1]並且STAGE1是解鎖的，則畫面跳到STAGE1；如果按下[2]並且STAGE2是解鎖的，則畫面跳到STAGE2；如果按下[3]並且STAGE3是解鎖的，則畫面跳到STAGE3；如果按下[H]，則畫面跳到Help。

在每個STAGE中，如果通關（pass），則跳至其對應的SUCCESS畫面；如果在STAGE3中失敗（fail），則跳至FAIL畫面。

在SUCCESS1、SUCCESS2畫面中，如果按下[N]，則跳至下一個STAGE畫面；在SUCCESS3畫面中，如果按下[N]，則跳至STAFF畫面。

在任意一個非TITLE的畫面中，只要按下[B]，會跳回TITLE畫面。

（２）螢幕顯示：VGA、game\_display

**game\_display**使用**draw\_[obj]**查找（h\_cnt, v\_cnt）上是否有需要顯示的object，並輸出isBlank和pixel\_addr。**blk\_mem\_gen\_0**則根據pixel\_addr讀取圖片的顏色讓螢幕顯示。

|  |  |
| --- | --- |
| draw\_player | 玩家控制的角色 |
| draw\_boss | Stage3中會追逐玩家的角色 |
| draw\_map | 迷宮的地圖（牆壁） |
| draw\_door | 出口（Locked & Unlocked） |
| draw\_obj | 鑰匙和燈泡 |
| draw\_interface | 標題、遊戲指示等遊戲介面或其他互動物品 |

（３）音樂：Music

**bgm\_[state]**根據音樂進度輸出note對應的頻率，**game\_sound**則依照state選擇不同的bgm，最後再用**speaker\_control**輸出音樂。另外由vol\_down、vol\_UP、mute調整音量並在led顯示。

（４）時間顯示：SevenSegment

**Timer**計算通關時間（秒、分鐘），再透過**SevenSegment**顯示。

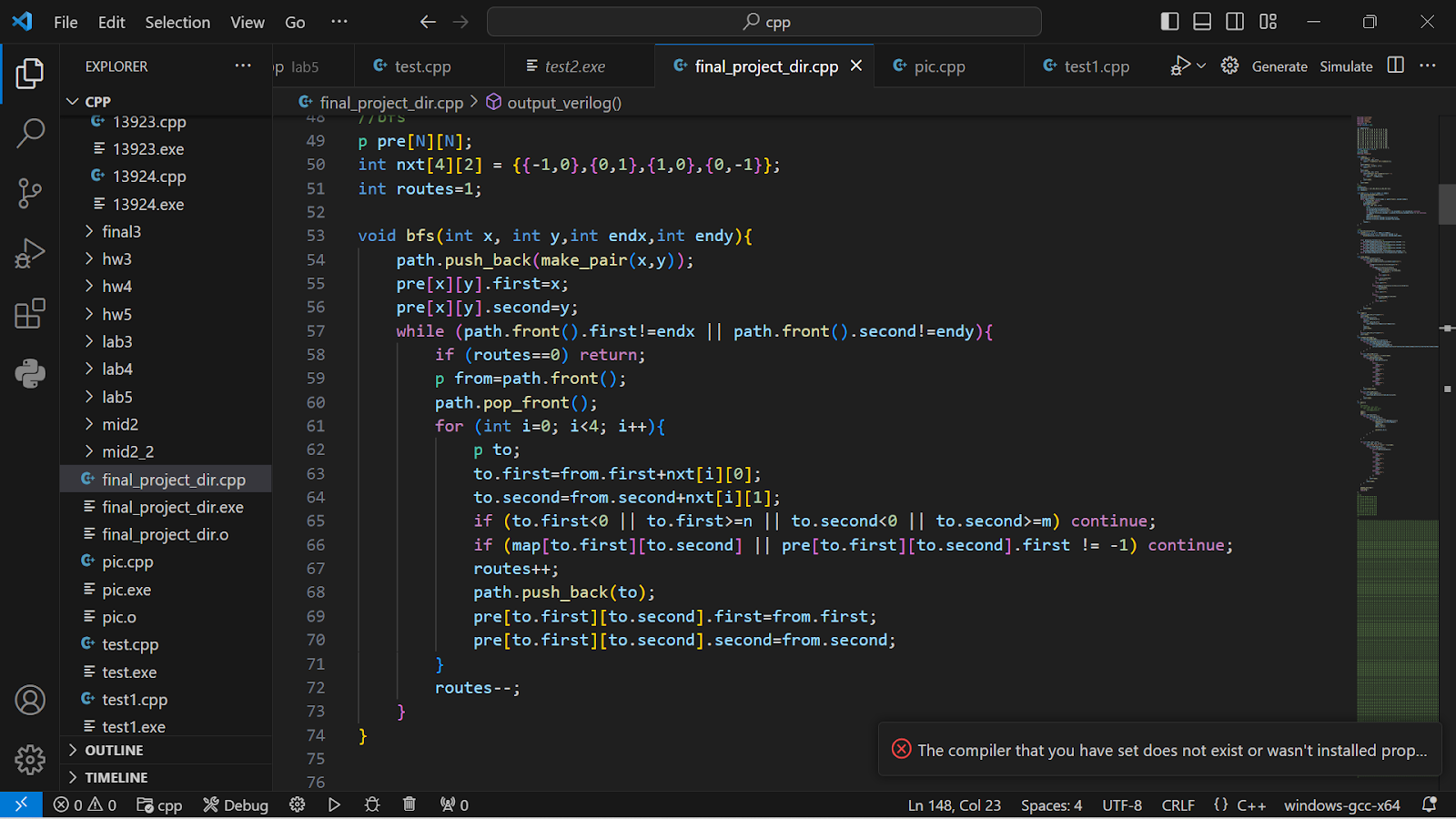
**（二）Game Implementation**

　　除了之前lab做到的鍵盤與螢幕的互動等，這次Project中我們有一些比較重要的實作。

１、Boss movement

　　在C++中，先利用一個（13\*13）的地圖，用BFS演算法求任兩點間的最短路徑path，並存其起始點的方向。再將此陣列帶回到verilog中，以shortest\_dir儲存其值。

一張含有 文字, 軟體, 電腦圖示, 多媒體軟體 的圖片

自動產生的描述

　　利用shortest\_dir陣列存在任意boss位置和任意player位置時，boss應該要往哪一個方向前進，其中，第13ｍ+ｎ行代表player在map[ｍ][n]（13\*13）時，boss在各點的追蹤方向，有上下左右四個方向，以兩個bit為一個單位（00：Up、01：Down、10：Right、11：Left），因此每一個陣列的元素會有338（169\*2）個bit。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

　　計算boss和player對應到13\*13 map的位置後，利用shortest\_dir取得boss的追蹤方向，其中

**｛shortest\_dir［ax+ay\*13］［( bx+by\*13 )\*2］，shortest\_dir［ax+ay\*13］［( bx+by\*13 )\*2+1］｝**

代表從（bx, by）到（ax, ay）的最短路徑的方向。

一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

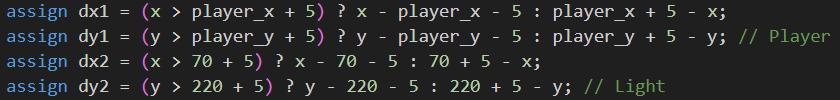
自動產生的描述

２、Light

　　如果state2的燈還沒亮，用dx和dy計算**dx2 + dy2 < r2**，

　　讓人物和燈周邊圓圈範圍以外的圖片都不顯示。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

３、Character

　　用player\_state和boss\_state判斷要顯示的動作圖片。

一張含有 文字, 字型, 螢幕擷取畫面, 數字 的圖片

自動產生的描述一張含有 蠟燭 的圖片

自動產生的描述一張含有 藝術 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

4、Music

　　我們用lab7中的方法實作bgm，但因為我們每一個畫面都有各自不一樣的bgm，為了讓打譜不要浪費太多時間，我們有配合C++設計方便寫音樂的程式。

一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 陳列 的圖片

自動產生的描述

**三、 實作完成度**

**（一）Undone**

１、遊戲暫停

　　原本想設計遊戲暫停畫面，後來改成了back按鍵讓玩家可以直接回到TITLE代替。

２、隨機的鑰匙位置

　　想讓鑰匙的位置可以隨機生成，但是因為怕LUT不夠用，因此捨去。

３、音效

　　因為我們的背景音樂有主旋律和伴奏，已經將兩個聲道都占用了，所以就沒有做音效。

**（二）Additional**

１、Help畫面

　　新增Help畫面介紹遊戲操作方式。

２、返回首頁

　　Stage畫面中可以按B鍵回到Title畫面，讓玩家可以隨時離開遊戲回到Title。

３、新增動畫

　　新增其他畫面的動畫，增加遊戲螢幕的豐富度。

**四、 困難與解決方法**

**（一）Boss 追蹤 player**

　　為了讓boss能夠追蹤player的位置，我們一開始便想到用BFS取其追蹤的最短路徑之方向，但是經過嘗試之後我們發現while不能合成到FPGA中，會合成超久還synthesis failed。

　　我們後來先利用c++語言，將地圖上每個a點到b點最短距離要走的方向存進陣列。如果以一個pixel為單位直接跑BFS的話，計算和陣列空間仍然龐大，並且因為路寬超過一格，最短路徑解容易出現錯誤。

　　所以最後我們便將迷宮簡化為路和牆皆為一格單位（13\*13），再用簡化後的路線找最短路徑。在verilog中，就可以直接透過目前player和boss對應到13\*13地圖的位置，直接查找陣列決定boss下一步要走的方向。

**（二）LUT 不夠**

　　最後我們在各個畫面加上人物動畫時，發現LUT會超過限制。所以我們盡量將每個module中不必要的計算和變數去除掉，但作用微乎其微。最後我們決定將**Title**其中一個小動畫，也就是Help鍵旁邊原本會出現boss的動畫刪除，才終於讓LUT低於100%，使其正常運作。因此也打退我們原本想新增更多功能的念頭。

**五、 心得討論**

　　這是我們第一次製作沒有template的project，雖然每個功能大概知道如何實作，但是整個程式的架構都還不清楚。一開始花了很多時間思考和整理硬體之間與modules之間的關係，再根據先前lab的經驗大致寫出modules的template，好的開始是成功的一半，之後實作細節時就可以很順利的進行。

　　在過程中我們也遇到了一些困難，像是雖然boss追蹤player的方法可以用C++解決，但實際用在遊戲裡有很多boundary condition，難度不高卻需要細心花時間處理細節。後面做得越來越上手，我們新增了一些平常玩遊戲會有的功能，讓我們的project可以更接近一般遊戲體驗。不過在快結尾時發現LUT不夠用，即使有想新增的功能也不太能實作（我們認為主要是螢幕顯示的計算和記錄boss的陣列佔了不少），但自認有讓原本簡單的小遊戲多了更清楚明瞭的UI（也更可愛）。

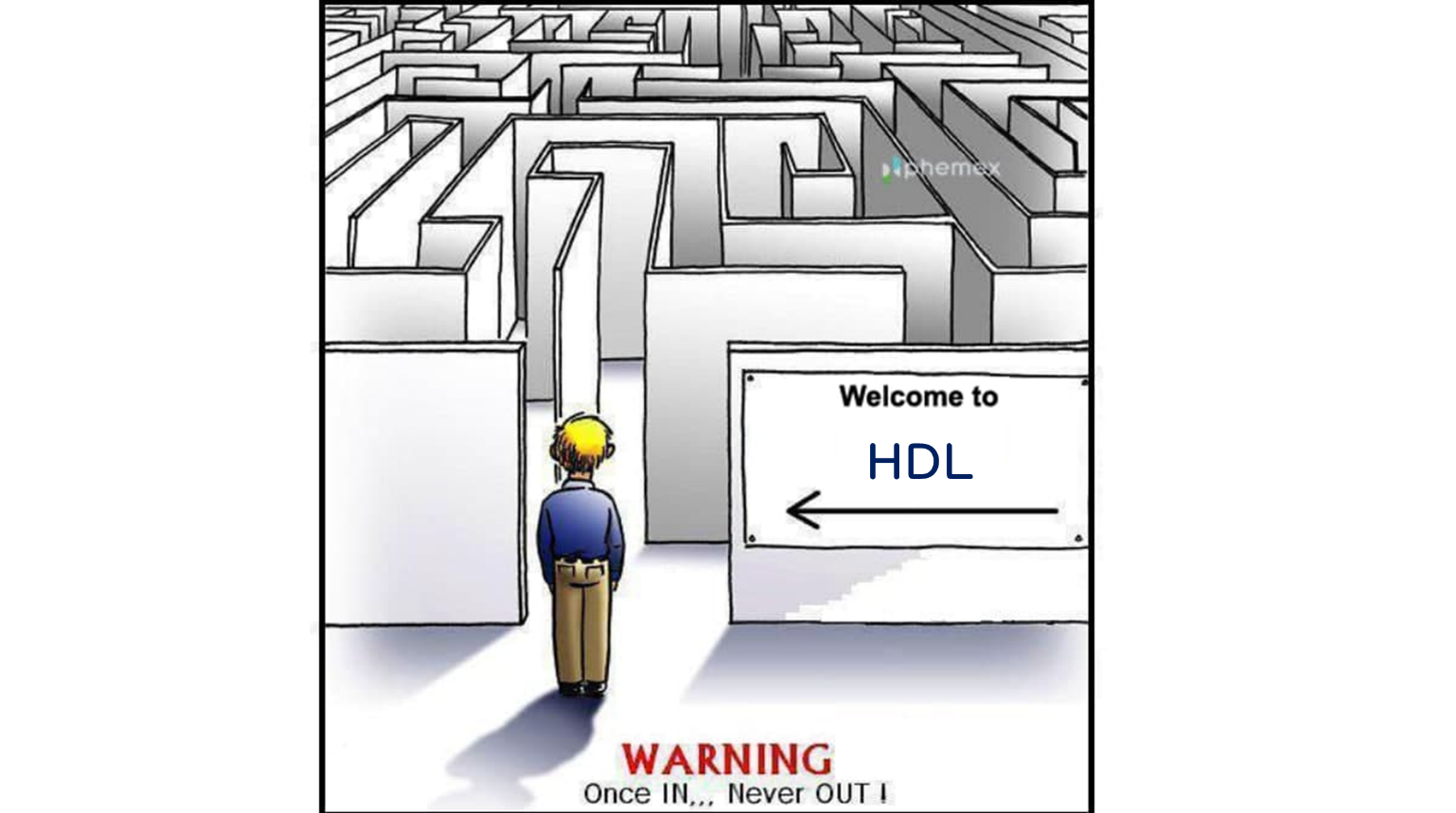
　　雖然每周都在煩惱各科的期末，HDL的project倒是做得很開心。當初預想的進度規劃我們幾乎都有跟上，甚至有時候半夜做上頭了還會心血來潮新增小功能。所以我們就算遇到困難也都有足夠的時間靜下心處理，不會出現同時要處理一堆bug又要趕進度的焦慮狀況。

　　在這次project中，我們學習到如何從零開始製作一個遊戲，包括遊戲構想與程式架構規劃；當然也從每一次的lab中累積（debug）經驗和耐心，讓我們最後可以用養孩子的心態在設計這個Project（辛苦又開心；；）。真的很感謝老師和助教們一學期以來提供的幫助，也希望透過這次的經驗，之後可以更有效率也更完善豐富！

**六、分工**

留鈺婷：

　　- Draw Pictures & Display

　　- Map Design

　　- Boss Direction of Shortest Path

　　- Speed Up

陳璟婷：

　　- Top Module

　　- Screen FSM

　　- Game Control

　　- Music