

Final Project Report

學號: 111062202、111062205

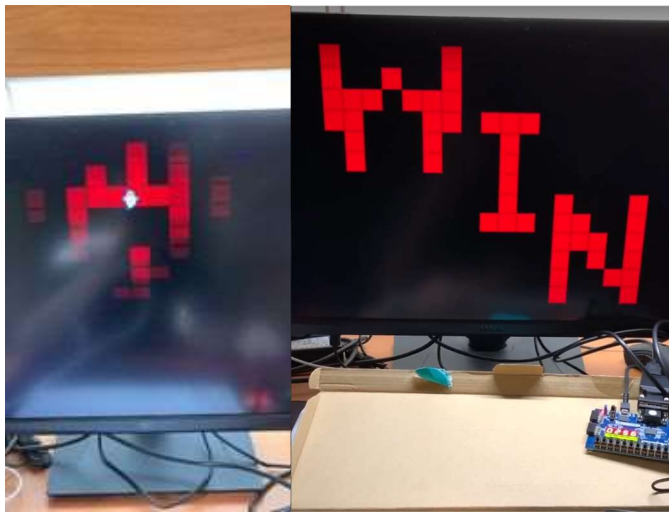
姓名: 陳威傑、鍾秉智

A. 遊戲簡介

我們的 Final Project 是進行遊戲的製作，遊戲名稱為密室逃脫。

在這款遊戲中，玩家的任務是協助一位可愛的小幽靈，在限定的時間內成功逃離地底的迷宮。由於地底的光線微弱且空氣品質不佳，小幽靈的視野相當有限，只能看到自己附近的路線。迷宮中隱藏著通往地面的大門，然而，這扇門並不會一開始就呈現在地圖上。玩家必須搜尋迷宮，找到星星，並逐一將星星擺放到門上，方能成功打開大門。

遊戲的挑戰在於時間有限，若小幽靈無法在指定的時間內成功逃離迷宮，地底的毒氣將淹沒她，結束遊戲。



B. Code 實作

我們的 project 主要由以下幾個.v file 組成，以下分別講解其作用：

Top.v:

Top.v 主要用於處理各 module 的輸入、輸出及接收與傳遞硬體讀到的訊息。另外也有少部分的設計在 Top.v 中，如 state 的控制、倒數計時的控制等。

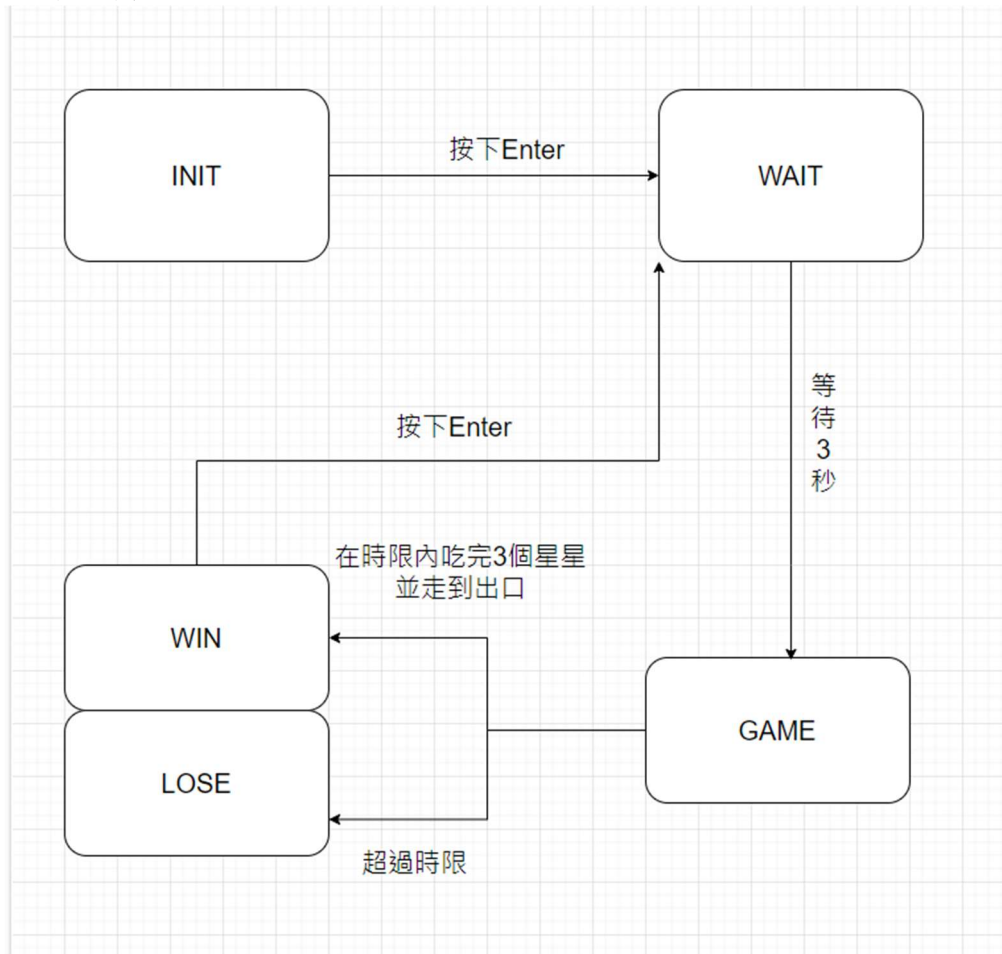
值得一提，本遊戲主要由五個 state 組成，包含 INIT、WAIT、GAME、WIN、LOSE。遊戲最初在 INIT state，按下 enter 鍵後進入 WAIT state，經過三秒的倒數後遊戲就會進入 GAME state。在 GAME state 中，如若玩家的血量歸零，或是倒數計時歸零，遊戲就會進入 LOSE state，反之，如若玩家成功抵達終點，遊戲就會進入 WIN state。而在 WIN 和 LOSE state 中，如若玩家按下 enter 就會重新返回 WAIT state。

以下是 Top.v 的部分 code 截圖：

```
module Top (
    input clk,
    input rst,          //btn_c
    input volUP_btn,    //btn_up
    input volDOWN_btn,  //btn_down
    inout wire PS2_DATA, //keyboard
    inout wire PS2_CLK,  //keyboard
    output [15:0] LED,
    output [6:0] DISPLAY, //7 seg
    output [3:0] DIGIT,   //7 seg
    output [3:0] vgaRed,   //screen
    output [3:0] vgaGreen, //screen
    output [3:0] vgaBlue,  //screen
    output hsync,          //screen
    output vsync,          //screen
    output audio_mclk,      //voice
    output audio_lrck,      //voice
    output audio_sck,       //voice
    output audio_sdin       //voice
);

//state control
always @(posedge div_hsec)
    three_sec_cnt <= ((state == WAIT) ? (three_sec_cnt + 1) : 1'b0);
always @(posedge clk or posedge rst) begin
    if(rst) state <= INIT;
    else begin
        case (state)
            INIT: begin
                if(key && (key_num == ENTER)) state <= WAIT;
                else state <= state;
            end
            WAIT: begin
                if(three_sec_cnt >= 6) state <= GAME;
                else state <= state;
            end
            GAME: begin
                if(character_h>16*13 && character_h<16*14 && character_v>16*11 && cha
                else if(!time_left[0] && !time_left[1] && !time_left[2])
                || (curr_hp == 0))
                    state <= LOSE;
                else state <= state;
            end
        endcase
    end
end
```

另外以下是 state 圖:



Clock_divider.v:

負責處理撥放音樂、7-segment、畫面顯示以及按鈕所需的不同 clk，另外也包含一個使用 counter 計算的精準半秒 clk。

Button.v:

音量的放大及縮小由按鈕操作，由這個檔案負責按鈕的 debounce 以及 one_pulse。但是 rst 的按鈕沒有特別 debounce 和 one_pulse。

Led.v:

負責在 WAIT 時閃爍還有在 GAME 時顯示玩家血量及音量大小。

Seven_segment.v:

用來顯示 7-segment，要顯示的東西由 top.v 輸入(top.v 會輸入一個 16bit 的資訊，7-segment 會將其每 4 個 bit 轉成一個顯示元素)。7-segment 主要負責顯示完成迷宮的所剩時間。

Keyboard.v:

```

module Keyboard ( //use "w" "a"
    input rst,
    input clk,
    inout PS2_DATA,
    inout PS2_CLK,
    output reg [2:0] key_num,
    output reg key
);
  
```

處理鍵盤傳輸的數據，分別有 W、A、S、D 還有 Enter，透過 key_num 傳輸給 top,另外 key

是用來判斷按鍵的狀態，被壓住的話 **key** 會變成 **1**，放開的話會變成 **0**，以此來追蹤按鍵的觸發。

Map.v:

負責地圖的數據，將不同 **state** 下的地圖數據傳給 **screen.v** 去輸出。另外，**WIN/LOSE** 的畫面也是讓地圖去輸出的。

Character.v:

負責處理角色的移動，方向，在那些 **state** 需要顯示等等。至於輸出畫面的部分是由 **screen.v** 負責。

Screen.v:

```
module Screen (  
    input rst,  
    input div_2,  
    input [2:0] state,  
    input [0:899] map,  
    input [8:0] charactor_h,  
    input [8:0] charactor_v,  
    input charactor_dir,  
    output all_star_collect,  
    output reg star_countA,  
    output reg star_countB,  
    output reg star_countC,  
    output [3:0] vgaRed,  
    output [3:0] vgaGreen,  
    output [3:0] vgaBlue,  
    output hsync,  
    output vsync  
);
```

負責所有的畫面輸出，包含地圖、角色、星星等等。還有角色、地圖、星星和 **vga_controller** 的 **module** 在裡面，角色、地圖、星星的 **module** 會將該輸出的畫面 **serial** 的傳輸給 **screen**，再由 **screen** 作篩選，角色最優先顯示、星星第二、地圖在最下面。另外，判斷是否吃到星星也是由 **screen** 來操控的。

Voice.v:

負責遊戲的背景音樂還有吃到星星時的音效。

Constraints.xdc:

Constraints file 是仿造過去 **lab** 所使用的 **constraints file** 進行製作，此處不做截圖。

C. 問題與解決

1. Map 的繪製與 Screen 的顯示:

原先我們考慮直接使用外部 **jpg** 圖片進行地圖製作，不過後來考慮到仍須設定角色的可移動範圍，且須顧及記憶體的资源，我們更改計畫為將螢幕切塊進行地圖的製作。在實作中，我們將 **320*240** 的螢幕切成 **20*15** 的格子，每格為 **16*16** 的大小。我們僅需紀錄一格的地圖資料以及 **20*15** 的地圖陣列即可控制整張地圖的顯示。(透過上述方法製作地圖，我們不僅能快速調整地圖，也能輕易達成製做多張地圖的目標)

2. Character 的移動與 Screen 的顯示:

由於地圖是以切割的方式在螢幕上顯示，角色也很自然地以 **16*16** 進行製做。不過此時我們碰上兩個大問題，其一為該如何讓角色在地圖的切割線上移動(不會受限於切割線而只能進行跳躍式移動)，其二為該如何將角色顯示在地圖之上(該如何將不規則的角色顯示在地圖上，又不干擾到背後的地圖顯示)?

首先第一個問題，我們很快就達成共識不能讓角色僅進行跳躍式的移動。藉此我們換了個思維，我們維持使用 320×240 的座標紀錄角色的位置，並將地圖的座標乘以 16 回到原本的 320×240 的座標，由此反過來判斷角色處於哪個地圖格子內，並判斷角色可否待在該格子內。

至於如何將不規則角色顯示在地圖上，同時不影響旁邊地圖的顯示，我們為此思索多日。最終結論如下，我們在 screen 進行掃描時先判斷座標是否處於角色所在的區塊，接著再判斷該區塊中的該座標使否有角色的部分需要描繪，如若否則改進行 map 的繪畫，反之維持角色的繪畫。

3. Character 的速度控制:

在遊戲中，如若持續按壓方向鍵(W、A、S、D)不放，角色會持續進行移動，這是我們對於角色移動的規劃。在實際設計時，我們很快便決定以 counter 的方式進行計算，按壓時間每經過一定數值，角色會向某方向移動一步(320×240)。然而設計時由於一再出現移動速度的 bug，使我們一度懷疑這樣的設計能否滿足我們的需求，也曾嘗試以不同的方式進行設計，不過最終還是以最初的方法完成設計。

4. 視野限制的製作:

在顯示螢幕時，去計算當前顯示的位置與角色的距離，我們計算距離的方式是以 16×16 的格子為單位，若當前顯示的位子離角色 3 單位的話，那我們就將 R、G、B 三種顏色分別減掉 3×2 (我們設定的色域為 $0 \sim 16$)，或是將顏色減至 0 為止。設計時，由於顏色的位元只有 4bit，所以若顯示的位置離角色很遠(超過 15 單位的話)，由於一開始計算的方式是直接讓顏色減去距離再檢查是否大於 0，這樣的話因為 register 的運作方式讓遠處的顯示產生 bug，所以我們後來限制了距離的最大值，才解決了這個問題。

D. 分工

完成(鍾秉智、陳威傑):

Code 架構建設(規劃 module、.v file 等)

Button、Keyboard、Led、Voice、Seven_segment、State 等實作

最基礎的 character、map、screen 等實作(角色可正常移動於地圖內)

Win、Lose 的設計

Screen 視野限制、吃星星等

美化地圖

另外值得一提，我們是使用 github 統整 code 以進行遊戲製作，以下附上 Github 連結:

https://github.com/sevenworm7/Hardware_Final_Project

E. 實做完程度

Level 1: 我們預計從基本的迷宮進行製作，目標是角色能正常地移動於地圖中，且不會出圖。該部份我們有確實達成。

Level 2: 接著我們希望能增加少數的陷阱、解謎內容等，以下以條列式說明:

需要觸碰星星後終點才會顯示在地圖中。

我們有確實達成以上敘述。

視野限制，玩家無法看到距離角色過遠的地圖內容。

我們有確實達成。

減時陷阱，會偽裝成星星的模樣，如若觸碰則會減少所剩時間。

星星及觸碰的機制皆可使用舊有的 code，製作難度低，但尚未完成。

製作多關迷宮。

舊有的 module 都可直接使用。未完成。

製作傳送門。

僅需計算地圖兩格之座標，符合則將角色傳送至另一點。未完成

Level 3: 最後我們希望能加入較為困難的內容，以下以條列式說明:

加大地圖，如若角色走出螢幕外，則會切換到另一張地圖的顯示。

沒有達成。

製作密碼鎖，需要透過鍵盤輸入特定密碼以解鎖門。

未達成。

製作會追擊角色的敵人。

高難度，未達成。

F. 結語

就結果而言，我們認為本遊戲尚有許多可改進的地方。首先，遊戲難度過於簡單，迷宮的地形不夠複雜，陷阱、解謎的元素也仍舊不夠多。其次，遊戲的沉浸感不足，如遇上陷阱時的音效變化，移動到特定地點後的背景樂變化等都能再加改進。

另外就我們目前使用的記憶體空間而言，我們依舊有充足的空間增加新元素(原先有預計這應該會是一個很燒記憶體的 **project**，在記憶體的空間規劃上有做非常充分的準備)。時間的不足大概是我們本次 **project** 最大的遺憾，雖然我們有盡可能在段考間擠出時間進行 **project** 的製作，成果和預期仍有一定落差，值得檢討。