PLC 與人機介面設計實作

作品名稱：PLC 踩地雷

作者：吳子龍

課程：機電整合實習

日期：2022 年 6 月 30 日

目錄

1 前言

1.1 作品簡介

這一次的作品我還擇了一個每個人都玩過的小遊戲 - 踩地雷，雖然看起來簡單，但在 PLC 上要實作相當的困難，也是一項挑戰，我這一次選用 9x9 的地圖，因為這樣的大小可以放滿人機介面的螢幕，而且在 PLC 的運算上也不會太遇吃緊。

這個作品分為兩個部分，第一個部分是 PLC，還用三菱電機 FX-3U，這也是我在學校學習到的第一顆 PLC，這也是最重要的東西，所有的程式都燒錄在裡面，以及輸入輸出都是由這裡設定的，第二個部分是人機介面，選用士林電機所出產的人機介面，帶有觸控功能。

1.2 研究動機

在課程上我學習到很多的邏輯設計，像狀態機就是一種常見的設計方式，我在學習到狀態機之前，如果我要設計一些比較複雜的物品，經常需要很多邏輯開關來進行設計，這樣設計相當的花費時間，而且要除錯也相當不容易，所以我在學習 PLC 的初期會覺得很吃力，而且當時是用書寫器進行書寫，比較沒有那麼直觀，但在後來我們開始用電腦來灌程式，除了直觀很多，而且在除錯也快上許多，但這個時候我還是一直有一些不太容易的點，像一開始寫習的階梯圖就真的非常的不容易，因為我在之前有學習過別的程式語言，但在使用上總是要考慮到很多我以前不曾注意到的細節。後來我在自學的過程中，我學習到 ST 語言，這是一門像 C++ 這種純文字的語言，我在編程上面變的非常的得心應手，因為我想要靠這個做出一項實際的做品。

1.3 踩地雷規則

遊戲的目地是開所有**非地雷**的格子，若玩家覺得該格有地雷的話，也可以選擇旗子來防止誤觸。

若玩家的點到的格子周圍 8 格內有 1 以上（含）的地雷，則需要在該格顯示周圍 8 格的地雷數。

若玩家點到的格子周圍沒有任何的地雷，則需要連同所有相連的空白格子也都打開，直到沒有空白格子為止。該遊戲所比的就是在最短的時間內打開所有格子。

1.4 課程運用

能成功的做出這個作品，也是因為我們的機電整合 課程裡面有這些設備，而知道如何使用這些設備，也是因為在上學期的可程式控制實習 課程中有學習到 PLC，給了我 PLC 的基楚，像是怎麼燒錄、IO 控制與如何與人機介面連接。

2 實作

2.1 硬體架構

2.1.1 程式語言

因為我 C++ 及 Java 這兩個我在之前有開發的經驗，因為這一次的開發我選擇的不是階梯圖，而是選用 ST 語言，雖然這門語言在學校沒有教學，但我還是花了一段時間，大該清楚他的邏輯，以及與 C++ 這種電腦的語言差別在哪。

2.1.2 PLC 版本

我當初的 PLC 有兩種選擇，一種是比較新的 FX-5U，這是比較新的機種，但因為當時我所學習的都是 FX-3U 為主，雖然 3U 的功能較 5U 少一些，但在編程上比較容易一些，但這也代表有一些功能設要自己設計。

2.2 軟體架構

2.2.1 輸入輸出 IO

輸入與輸出大部分使用 PLC 內部接點，除了方便與人機介面溝之外，也因為本次的作品使用的 IO 數量較多，外部接點只使用兩個燈號指示。

2.2.2 內部運算

因為踩地雷這種遊戲算有運用到「圖」的遊戲，因為在設計上最初就需要陣列來儲存，但在 3U 上面能選擇的只有一維的陣列，因為我還必需自己實作一些資料結構。有了地圖還是遠遠不夠的，我還需要產生地圖，這部分我自己實作了亂數產生，以及當我按下時需要做出的反應，可能是一次打開一整片，也可能是只開啟一個，更可能是跳到地雷，因為這些都需要複雜的設計，也因為我之前有學習到「有限狀態機」，最後也決定用這種設計方式進行實作。

2.3 遇到的困難

在做這項作品的時候因為有很多東西我還沒有學過，因為很多時候我必需要自己想出解法，這些困難也對我後面在寫別的程式的時候有更多的啟發。

1. 地圖如何每一次生成都是隨機的？
2. 如何讀取地圖與改變地圖的值？
3. 如何將大量的 IO 轉換為數值？
4. 根據遊戲機制，如果玩家點選到空白的格子，那麼需要連帶打開周圍是空的格子，但要如何做到？
5. PLC 計算能力與電腦不能相比，PLC 經常死當，該如何解決？
6. 人機介面與 PLC 該如何進行模擬？

2.4 編程

2.4.1 PLC 看門狗設定

在我製作接進收尾的時候，我發現在電腦模擬並沒有什麼問題，但在 PLC 上卻不斷的當機，我進行算法優化，但計算量還是相當的大，因此我去查找了指令表，發現我可以設定看門狗[1]。因為算法的優化與看門狗，因為這段程式我測試了兩種 PLC，都可以順利的運行。

2.4.2 地圖的儲存

地圖因為是一張「圖」，因此在選用上面當初規畫時，我覺得可以使用二維的陣列來進行儲存，但因為 ST 語言沒有辨法使用二維陣列，因為我只好使用一維陣列來代替，這也讓我後面的計算變的複雜。

2.4.3 亂數產生

遊戲好玩就在於每一次都是不一樣的狀態，如果地圖都是同一張的話，只要多玩幾次就可以背起整張地圖的走法。

因此我決定使用亂數讓每一張地圖都是不一樣的，這裡我花費了很多時間，研究了各種的寫法，因為看起來亂數夠亂了，但實際上放到地圖總是跟我所想的不一樣，最後我決定使用「線性同餘方法」，這樣的亂數產生簡單，而且也足夠的亂。

[1]如果在一定的時間內沒有向看門狗發送訊息，則會讓 PLC 跳出程式

2.4.4 地圖的讀取與寫入

在一維陣列上要輸入二維陣列的內容，這裡我一開始想要直接硬碰硬，但我後來發現這是行不通的，而且這也是主要的問題之一，如果這一步我沒有辦法解決的話，我後面也沒有辦法繼續實作，我最後想出的解法是，寫兩個函式，讓一維的數字 [0 80] 可以與二維數字 [0 8][0 8] 互換。

2.4.5 大量的 IO 轉數值

我最初的想法是使用二進位進行轉換，但因為 IO 數量高達 81 個，因為這樣的方法在實作上會花費大量的時間，因為我去學習到索引暫存器[2]，這是一種我完全沒有使用過的寫法，也是花費了一些時間研究，在這一次的作品的地圖都是使用此方法。

2.4.6 點擊判斷

當玩家點擊到螢幕的時候要做的動作，這一步只需要將步驟轉換，其他的計算在由該步進行。

2.4.7 炸彈爆炸

如果炸彈爆炸的時候，我想要一次翻開所有的格子，這一步需要邊翻格子邊計算該格是否為炸彈，如果為否的話還需要判斷周圍 8 格是否有炸彈，如果有的話就需要算出周圍的炸彈數。

2.4.8 周圍有 8 格炸彈

當玩家點擊到的格子不是炸彈，但周圍 8 格有炸彈，就需要算出周圍的炸彈數並顯示，這裡還需要搭配人機介面使用狀態燈進行設計。

2.4.9 空白格

這一步是最難的一步，也是我花費時間最多的一步，因為當我點擊了一個空白格，根據規則我需要同時打開所有相連的空白格，但因為空白格是隨機的，因為我需要找到一種算法可以一次打開超大量的格子，其實如果這在 C++ 這種語言裡面是一件很簡單的事，但因為在 ST 語言裡面，我沒有遞迴、動態陣列之類的東西，因為在實作上我覺得相當困難，最後

[2]可以儲存暫存器的位置，方便使用數值進行控制

我是選擇自己實作 queue[3]，因為有了這個之後我就可以使用圖論常使用的 BFS 算法[4]，但因為 BFS 算法的效率其實不高，因為這發生了一些我意料之外的事。

2.4.10 輸贏判斷

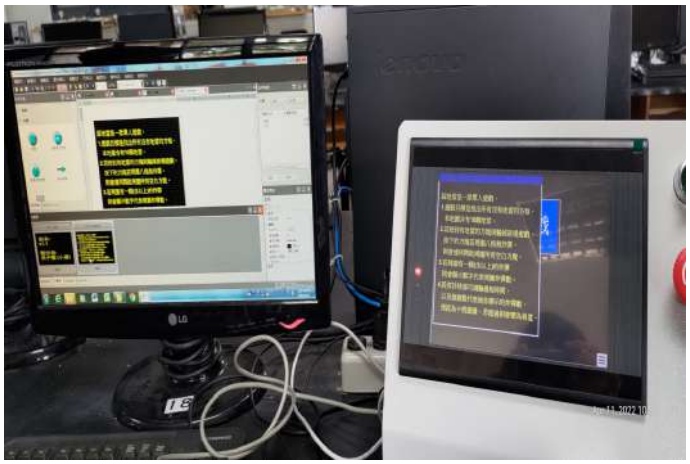
因為遊戲可以進行插旗或者開格子，因此贏的條件有一兩個，一個是開到只剩下地雷，另一個則是所有有地雷的格子都被插上旗子，但實際上的遊戲兩種都有可能發生，因為兩者需要交互判斷。

2.5 人機介面設計

2.5.1 開始畫面

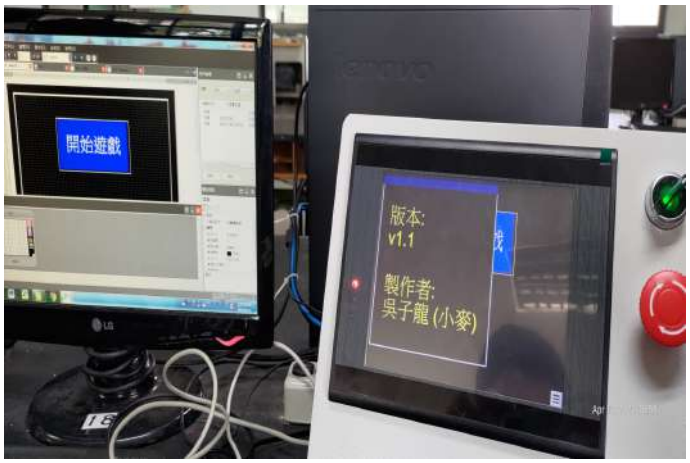
包含有三個按鈕，一個為進入遊戲並且同時 PLC 也會開始初始化，另一個則為教學按鈕，方便新手學習，以及在右下角有一個按鈕是可以顯示作者資訊。

2.5.2 教學



[3]又稱為佇列，屬於先進先出型資料結構
[4]又稱廣度優先算法，相反則為深度優先，會先計算鄰進資瞪，再不斷的往外擴充

2.5.3 作者



2.5.4 遊戲主畫面

這個畫面是最主要遊玩的畫面，左邊有一個最大的區域，這個區域是地圖顯示區，右邊則是功能選擇區。

1. 這張地圖為 9x9 的地圖，屬於簡單難度，在玩遊玩上只要點擊即可操作。
2. 有一個重置按鈕，如果遊戲進行到一半想要放棄的話需要長按 3 秒即可重置遊戲，並再次點選即可重新開始遊戲。
3. 手指和旗子代表現在要使用何種操作，對著格子使用手指即代表翻開該格。使用旗子即標示該格為地雷，若再次點選旗子則會被取消掉。
4. 計時格會在遊戲開始後進行計時，每一秒增加一次。
5. 插旗數為剩於的旗子數，因為一場簡單難度的踩地雷只有 10 根旗子，若數量為 0 則不可插旗。
6. 最右下角為返回開始畫面按鈕。

3 結論

3.0.1 成品

在完成後我自己試玩了一下，也給我朋友們玩，大部分也都覺得還不錯，也沒有什麼重大的 BUG 之類的，整體還不錯。

3.0.2 收穫

因為 PLC 我剛學習不到很久，可能離實際運用還有一段距離，但總是有學習到不少的東西，像 ST 語言也是我為了開發這個而去學習，我也理解到物件導向與函式的運用，這對這個作品的開發幫了不少的忙。

踩地雷這一種需要用到圖論的遊戲我是第一次遇到，因為如此我在開發初期遇到了很大的麻煩，尤其是在儲存和讀取這上面，因為當時並沒有學習到進階的演算法，因為我花費了很多時間在圖論演算法的學習上面，最後才想出屬於自己的解法。

3.0.3 延伸探討

雖然在作品完成後有基本的功能，我也給我朋友們玩過了，他們也反應出一些問題，而我自己在測試與編程的時候也有發現一些問題。

1. 亂數雖然看似足夠的亂，地圖的生成也有一定的水準，但每一次看地圖總是會有一些地方感覺重複了，或者讓人感覺不夠隨機，是否有更好的亂數產生方法？
2. 人機介面在設計要如何讓第一次接觸的人就知道如何操作？這一點在我完成這項作品之後其實也遇到很多第一次接觸的人並不知道如何遊玩與操作，因此在設計上是否要在更容易一點?。
3. PLC 在處理數字較不容易，像計時器只能一秒一次，那能不能每 0.1 秒一次呢？或者在需要更準確的計時有沒有別的方式?。
4. 在需要大量開啟地圖的時候，只能一步一步慢慢開啟嗎？有沒有更好的方式可以一次翻開一大片？
5. 因為在程式設計時運用了過多的迴圈，導致性能降低，能不能在更簡化寫法一些？或者更簡單的寫法？

3.0.4 心得

這是我第一次這麼深入的去寫 PLC，以前在上課的時候可能有做過簡單的東西，但要真的完成一項作品需要把你之前所學到的東西都派上用場，也因為當時沒有學習到演算法，所以在製作的時候非常的不容易，時常會有很多的措折，但每一次總是在我要放棄的時候想到「努力一定會有成果」，因為我總是會不斷的去研究一個我覺得很難的問題，雖然有時候真的沒有辨法給出理想的答案，但這過程中本身就有很多的收穫，也是這一點一滴的收穫才能在以後研究別的課題的有更好的解法。