

數位電路實驗期末報告

team04

B09703086 吳庭毅

B09901192 林沛翰

B09901194 陳品睿

1、 主題簡介

我們的期末專題是俄羅斯方塊-雙人對戰。

2、 File structure

team04_final

|-team04_final_report.pdf

|-Dclab_final_presentation.pptx

|-src

 |-DE2_115

 |-DE2_115.sv

 |-Debounce.sv

|-ip

 |-pll_vga.v

|-FSM_Game.sv

|-FSM_Whole.sv

|-Gameplay.sv

|-global.sv

|-keyboard.sv

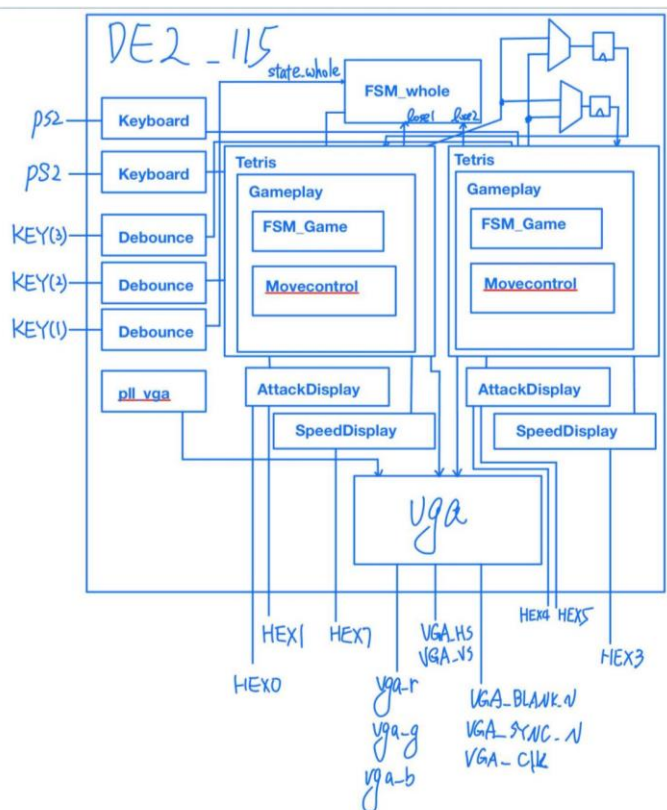
|-Tetris.sv

|-vga.sv

|-SevenHexDecoder.sv

|-MoveControl.sv

3、 System architecture



4、 功能簡介

a. 基礎功能

- 左移
- 右移
- 左旋
- 右旋
- 消行
- 慢速下降
- 瞬間下降(space)

b. 進階功能

- Shadow: 降落位置預覽
- Next five pieces: 下五個方塊預覽
- Hold: 暫存方塊
- Wall kick: 旋轉時若遇到障礙物，方塊會根據 SRS 規則判斷最終抵達的位置
- Lock delay: 當物體使用慢速下降的方式降落到底部時，還有 1 秒的時間可以移動
- Battle: 雙人對戰、進攻

5、 Hardware scheduling

a. DE2_115:

這個 module 主要處理硬體設施。包含了各種如鍵盤、按鍵、VGA 等處理硬體 protocol 的模組、和兩個分別處理兩位玩家的 Tetris 模組、和處理整體流程的有限狀態機 FSM_Whole，和一個處理對戰時互相攻擊的模組。以 player1 為例，當它消行時，會對玩家 2 攻擊，而其攻擊會先抵消玩家 2 對玩家 1 的攻擊，抵銷有剩後才會反過來攻擊，否則會被玩家 2 攻擊。而攻擊的能量條會經過 AttackDisplay 顯示在七段顯示器上。

b. Keyboard

這個模組處理鍵盤資訊的讀取，我們採用 PS2 protocol，另外我們有幾個 state，在 state 為 receive 時這個模組會傳送按下的按鍵的座標供其他 module 使用。

c. Debounce

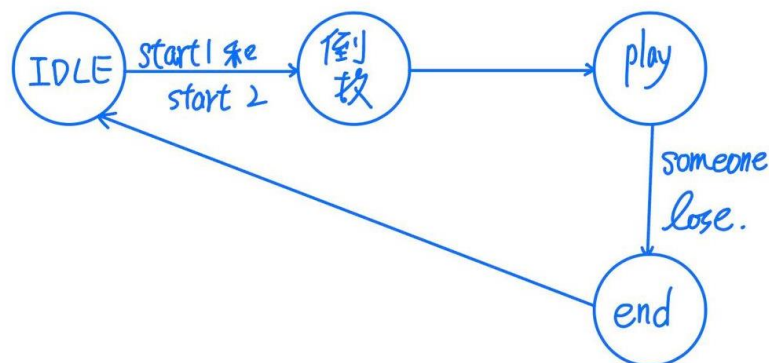
這個模組處理按鍵的傳輸訊號。

d. Pll_vga

這個模組會產生 25MHz 的頻率供 vga 使用，其產生方法是使用 quartus 的 IP catalog。

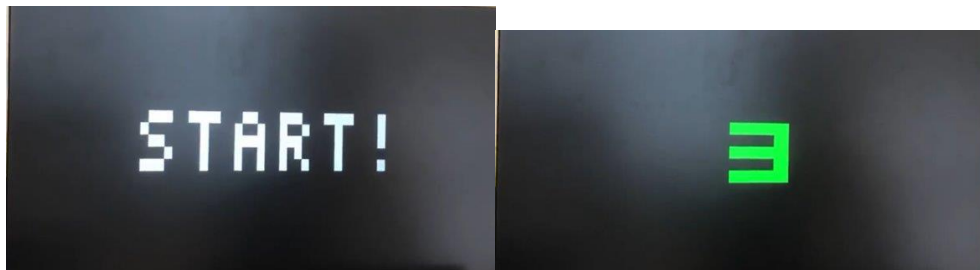
e. FSM_whole

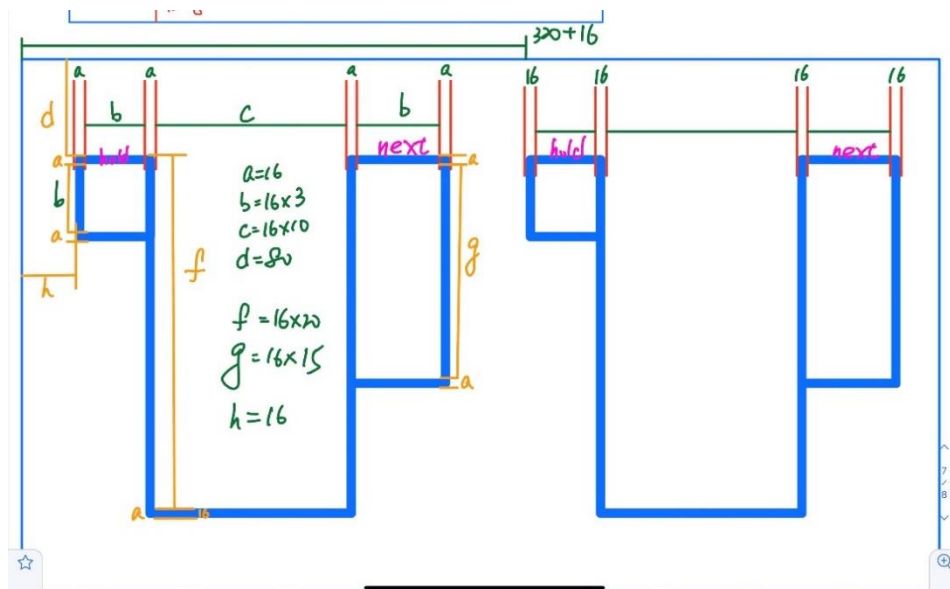
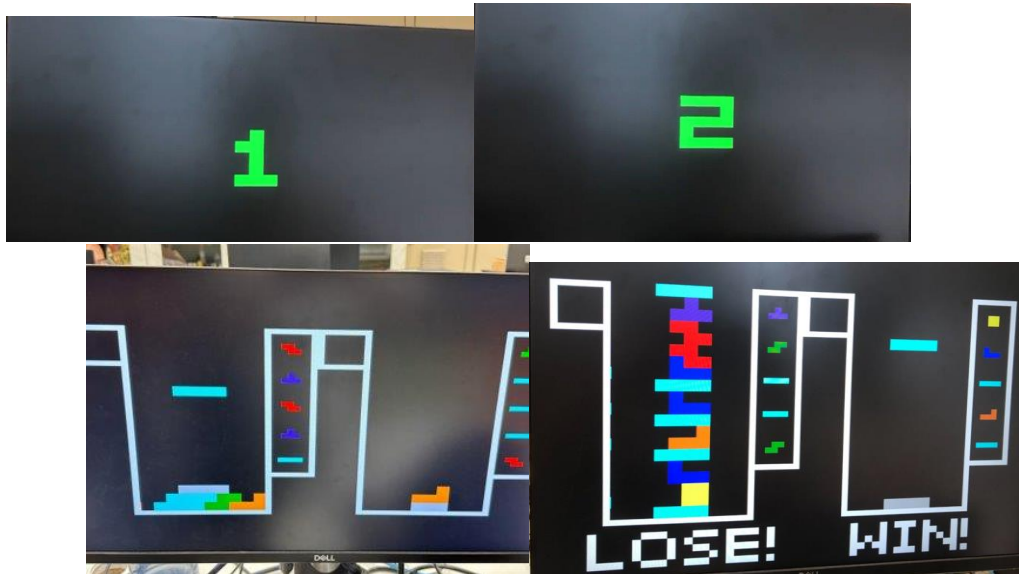
這個有限狀態機處理了整個遊戲的總體流程。在按下 reset 後，會先回到 IDLE 狀態，代表等待按下開始鍵(KEY[1])以開始，而此時畫面會顯示”Start!”字樣以告示使用者按下 start 按鍵。按下 KEY[1]後，會進入 CTWN 狀態，代表倒數 3 秒進入遊戲，此時畫面會根據倒數的秒數分別顯示 ”3”、”2”、”1”。倒數完成後，會進入 PLAY 狀態，此時玩家可以開始進行對戰直到有一方輸。當有人輸了後，會再進入 ENDS 狀態，顯示結束畫面，此時輸下方家會有”LOSE”，贏家下方會有”WIN”。



f. VGA

我們是根據下方的架構，手刻出我們的開始、結束畫面以及整個 tetris 遊戲畫面。





VGA 會以 25MHz 的頻率來掃過整個 640x480 pixel 的螢幕，若掃過的 pixel 座標在指定的條件內，則會傳 VGA 的水平 and 垂直座標出去，並把該座標位置的 RGB 顏色傳出。

-

結束畫面則會根據 lose1、lose2 誰輸的訊號決定哪方被顯示 "WIN!"，哪方被顯示 "lose!"。

g. Tetris

a) Gameplay:

這個 module 包含了遊戲細部運作方式及 FSM_Game 和 MoveControl。

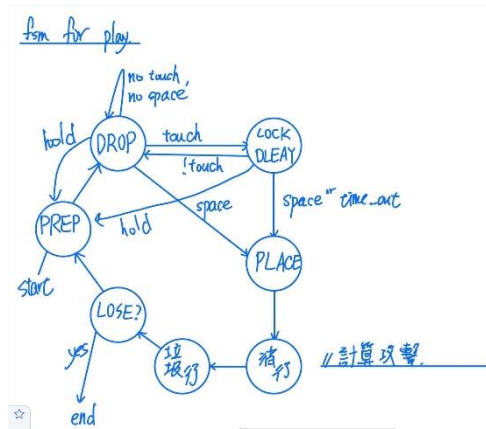
Hold 功能：每個掉落的回合 hold 只能使用一次，其功能是讓當下比較不好放的方塊可以先暫存留到之後使用。因此，當第一次 hold 被按下後，hold_enable 會變為 0，則之後再按的 hold 都將無效。切換後原本正在掉落的方塊會與 hold 方塊交換，若是最一開始 hold 還是空的狀態，則直接將當下的方塊存進 hold 並載入下一個方塊。當方塊確定放置準備進入下一個循環後，則會將 hold_enable 重置。

至於 space 可以直接降到最底部的方法，我是使用當按下 space 後 space_delay 會一直為 1，進而一直產生降落訊號，直到確定放置後才停止。而因為降落速度太快(50MHz)，所以肉眼不會發現其快速降落(影子的產生方法也是一樣)。

產生下幾個方塊的方法，我是使用 LFSR 的方式，讓其一直產生隨機方塊，並在需要時才去擷取值。擷取值時會先傳給下面第五個方塊，而第四個則會接收第五個，依此類推，而下一個方塊則會載入給即將掉落的方塊。

1. FSM_Game

這個有限狀態機處理遊玩過程時方塊的運作。其狀態圖如下所示。



當遊戲開始，會先進入 **PREP** 狀態，這個狀態會準備載入下一種方塊並將正在掉落的方塊的位置回到起始位置（中上位置）。接著下一個 cycle 會到下一個狀態 **DROP**，此時畫面會有掉落的方塊，並且可根據鍵盤對應的按鍵做左移、右移等操作。此時，若 space 被按下，則方塊會直接降落到底部並且沒有 LockDelay

的機會，即直接進入 PLAC 狀態。而若是透過自由降落(drop 鍵)或是慢速降落(down 鍵)抵達底部，那會先進入狀態 LKDY，此時方塊還能左右移動，直到 lockdelay 時間到或按下 space 時會進入狀態 PLAC。而狀態 PLAC 只會持續一個 cycle，在這個狀態會將掉落中的方塊會根據其 x、y 座標存進 memory 對應的位置裡面，成為被放置好的方塊(memory)的一部分。

下一個 cycle 會進入狀態 Elim，在這個狀態會執行行數消除，即某一欄都被填滿的話則該欄會被消除，該欄上面的欄數都會被往下順移。不像後面提及的垃圾行是全部移同樣數目，消除的不同位置會移動不同的行數。因此，我使用的消行方法是會有兩個指標，一個是從下面開始填入新的資料，一個是指向要被填入資料原本的行數。在每個 cycle 中，後者的指標會指向下一個不用被消除的列數並繼承它，而因為一次最多消 4 行，所以只要檢查上四行是否會被消除即可。而從下面開始填資料則可以避免後填入的資料不會因為前面先行改正而使資料有誤。直到第一行也被更新完成，則會進入下一個狀態：GARB。除此之外，ELIM 狀態的第一個 cycle 也會先計算在狀態 GARB 中要被攻擊的行數，即 actual_attacked。

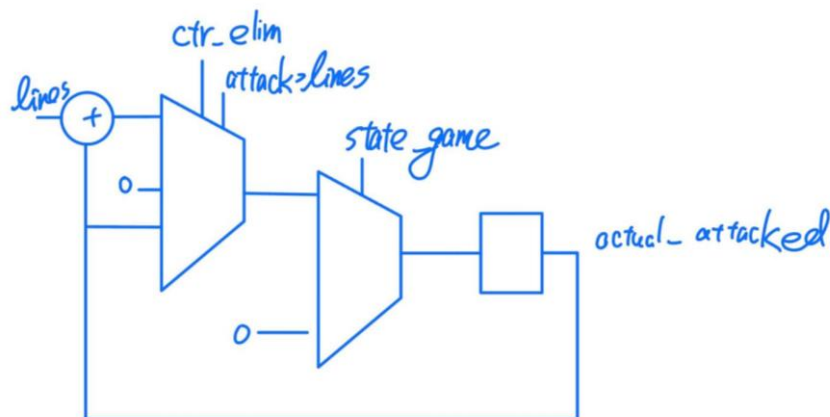
在 GARB 狀態中，會根據 actual_attacked 決定有幾行是垃圾行。而其計算方法是如果對手累積的攻擊數比本人這次消的行數還多，則受到的攻擊是對手累積攻擊數扣除本人這次消除的行數；若本人這次消的行數比對手累積攻擊數多，則會反過來對對手造成攻擊。獲得攻擊數後，所有列數會因為下面塞入垃圾行而被往上推。因此，與消除不同的是，GARB 會從最上面開始更新，在垃圾行數上的列數會根據 actual_attacked 決定要塞入下幾列的資料。而應被塞入垃圾行的列數則會被塞入垃圾列。而每列垃圾列只有一格是空格，其他皆是被占據狀態，當該空格被塞入時垃圾列才會被消除。而決定哪個位置為空格的方式是透過 LFSR。



垃圾行也被處理完後，則會進入 LOSE 的一個待機狀態並準備進行下一個循

環，即開始擺放下一個方塊。

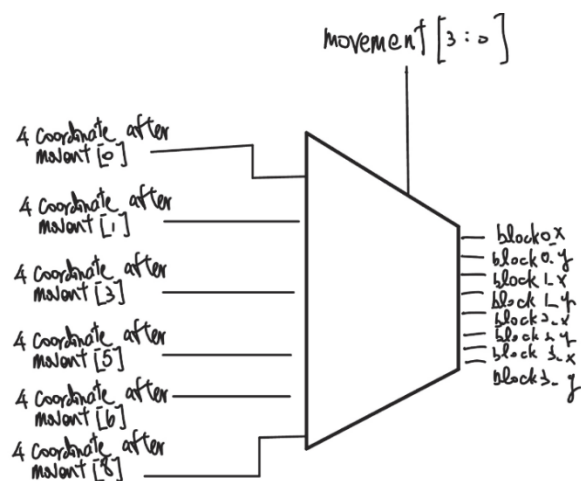
而 `actual_attacked` 的邏輯是只有在 `elim` 的第一個 `cycle` 時該循環該消除的行數



是正確的，所以在該 `cycle` 就會計算在 `GARB` 狀態時收到的實際攻擊。

2. MoveControl

這個 submodule 主要是控制每個 tetris block 的移動，包括六個 movement---「下」、「左」、「右」、「左旋」、「右旋」、「維持不動」。此外，每個 tetris 共佔四個格子，因此這個 submodule 會根據哪種 movement 來給定對應的四個 output 座標。



`movement[0]`:左

`movement[1]`:右

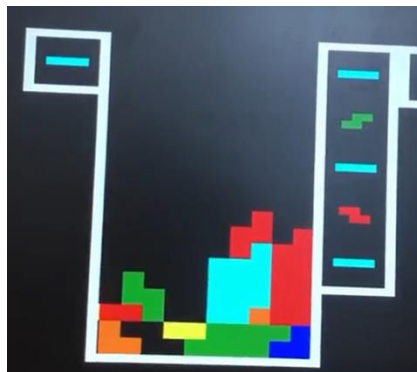
movement[2]:下

movement[5]:左旋

movement[6]:右旋

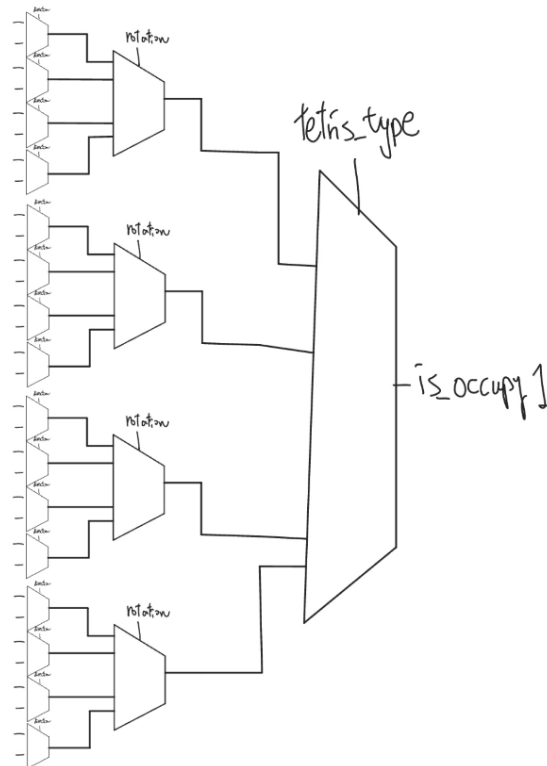
movement[8]:不動

值得注意的是，當我們在執行左旋或右旋時，如果撞到牆，根據 tetris 的規則，除了方塊 tetris，其餘的 tetris 我們會執行”wall kick”規則。



如右圖，當綠色方塊進行右旋時，會撞到紅色方塊，但是當根據 wall kick 規則，綠色方塊實際可以進行右旋並往下一格，像是右圖的樣子。實際上，每個方塊在進行旋轉時，都會進行五個 wall kick 的位置測試，如果其中一個可以進行旋轉，則旋轉成功；反之，則保持原位。

如下圖，就是負責處理 wall kick 的電路，由於每個 tetris type 都有五種旋轉位置測試，以下電路會有五份，分別判斷五種 test case 使否有通過，通過與否用 is_occupy_1, is_occupy_2, is_occupy_3, is_occupy_4, is_occupy_5 五個線路傳輸。



{is_occupy_1, is_occupy_2, is_occupy_3, is_occupy_4, is_occupy_5} =

5'0xxxx: test1 通過

5'10xxxx: test2 通過

5'110xxx: test3 通過

5'1110x: test4 通過

5'11110: test5 通過

5'11111: 都沒通過，tetris 保持原位置

以上這五種情況，我們就用 priority mux 去對應到轉後 tetris 應該要輸出的位置。

除此之外，我們有使用 memory 存每個格子的方塊種類，但是因為只有 7 種方塊、200 個格子，所以只需要 $3 * 200 * 2$ (兩位玩家) 個位置，所以我們使用 register file 的方式當作記憶體，也比較好處理。

6、Fitter summary

Flow Summary	
Flow Status	Successful - Sat Jun 17 18:34:19 2023
Quartus II 64-Bit Version	15.0.0 Build 145 04/22/2015 SJ Full Version
Revision Name	DE2_115
Top-level Entity Name	DE2_115
Family	Cyclone IV E
Device	EP4CE115F29C7
Timing Models	Final
Total logic elements	50,929 / 114,480 (44 %)
Total combinational functions	50,889 / 114,480 (44 %)
Dedicated logic registers	2,238 / 114,480 (2 %)
Total registers	2238
Total pins	125 / 529 (24 %)
Total virtual pins	0
Total memory bits	0 / 3,981,312 (0 %)
Embedded Multiplier 9-bit elements	0 / 532 (0 %)
Total PLLs	1 / 4 (25 %)

7、Time analyzer

Unconstrained Paths			
	Property	Setup	Hold
1	Illegal Clocks	0	0
2	Unconstrained Clocks	0	0
3	Unconstrained Input Ports	8	8
4	Unconstrained Input Port Paths	2179	2179
5	Unconstrained Output Ports	68	68
6	Unconstrained Output Port Paths	187	187

Multicorner Timing Analysis Summary						
	Clock	Setup	Hold	Recovery	Removal	Minimum Pulse Width
1	▼ Worst-case Slack	-14.953	0.174	N/A	N/A	9.238
1	CLOCK2_50	N/A	N/A	N/A	N/A	16.000
2	CLOCK3_50	N/A	N/A	N/A	N/A	16.000
3	CLOCK_50	-14.953	0.174	N/A	N/A	9.238
4	pll_vga_inst altpll_component auto_generated pll1 clk[0]	2.228	0.183	N/A	N/A	19.707
2	▼ Design-wide TNS	-1999.276	0.0	0.0	0.0	0.0
1	CLOCK2_50	N/A	N/A	N/A	N/A	0.000
2	CLOCK3_50	N/A	N/A	N/A	N/A	0.000
3	CLOCK_50	-1999.276	0.000	N/A	N/A	0.000
4	pll_vga_inst altpll_component auto_generated pll1 clk[0]	0.000	0.000	N/A	N/A	0.000

Slow 1200mV 0C Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	30.95 MHz	30.95 MHz	CLOCK_50	
2	42.52 MHz	42.52 MHz	pll_vga_inst altpll_component auto_generated pll1 clk[0]	

Slow 1200mV 85C Model Fmax Summary				
	Fmax	Restricted Fmax	Clock Name	Note
1	28.61 MHz	28.61 MHz	CLOCK_50	
2	38.73 MHz	38.73 MHz	pll_vga_inst altpll_component auto_generated pll1 clk[0]	

8、心得

- 為了要達成和真實 tetris 有相同的功能，我們有去實作"wall kick" 的功能，然而這個功能要考慮的 test case 十分的多，有非常多的判斷情況，導致實作的過程要寫很多判斷式 (if-else, case)。這個部份雖然邏輯簡單，但卻要花很多時間和耐心，去把程式刻出來。
- 為了測試 keyboard 可以正常使用，我們找 PS2 的 protocol 測試了蠻久的，而且為了達成雙人對戰的功能，我們本來也要研究 USB keyboard，幸好最後在網路上找到二接一的線，就可以一次用兩個 PS2 keyboard 了。
- 能夠自己親手做出自己喜歡的遊戲，真的很有趣。最後 debug 的時候，我們也都是邊玩遊戲邊 debug，是個很新奇的體驗。
- Final project deadline 前幾天，我們和其他組組員幾乎都住在實驗室，雖然當下覺得很辛苦，但是現在回首看那幾天共同的努力，真的是不可忘懷的回憶。
- 因為我們很喜歡玩俄羅斯方塊這款遊戲，因此一直想實作自己的俄羅斯方塊看看。而也剛好有這個機會讓我們可以真正做出一款屬於自己的俄羅斯方塊，前面的努力都值得了。
- 此外，刻 VGA 的部分也花了我們很多時間，我們幾乎是一個一個 pixel 去做判斷，並把畫面呈現出來。