Final Project

B_N_W

Design Description

B_N_W 是一個音樂遊戲，玩家須按照螢幕上的指示打擊節拍，螢幕配置如圖（一），玩家須在磚塊黑色磚塊碰到打擊線時，壓下對應的鍵盤按鍵，但在白色磚塊碰到時，需放開，掉落的四排磚塊對應到鍵盤上的按鍵分別是 FGHJ（由左至右）。此遊戲含有兩種模式—關卡模式以及無盡模式。遊戲畫面及操作指令圖請見附錄。

關卡模式（Lv 1~3）：

當處於關卡模式時，七段顯示器上將顯示 PLAY 字樣。關卡模式總共分有三關，分別為 60 BPM（beats per minute）、90 BPM 以及 120 BPM。遊戲中總共有三個磚塊藏有星星，星星磚在掉落時，星星會先快速旋轉，轉至方塊中心後停止，當玩家成功按下星星磚後，累計星星數會從遊戲上方跳出來，然後消失，玩家的目標是蒐集滿三顆星星並完成樂曲，完成樂曲或是按錯磚塊則會結束遊戲，遊戲結束時將顯示所得星星數。

無盡模式（Lv 0）：

在此模式中，節拍將越來越快，玩家將挑戰自己的極限，看看自己能夠撐多久。此模式中的節拍數將從 60 BPM 逐秒增加，增加速度後的節拍為上一秒的 1.07 倍。時間將由 FPGA 板上的七段顯示器顯示，左兩位為秒數，右兩位為毫秒數。

設定模式：

只可在遊戲暫停或結束時才能進入，長按 FPGA 板上的中心按紐即可切換到設定模式，設定模式有兩個部分，分別是音量設定及關卡設定。

音量設定：

在音量設定模式中，共包含 100 級音量等級（0~99），將由七段顯示器顯示準確數字，由下方 LED 燈號顯示其十位數，便玩家快速檢視。共有兩種控制方法。

1. 短按：玩家可藉由短按 FPGA 板上的上/下方按紐增加/減少一級音量等級
2. 連續長按：玩家可藉由連續長按 FPGA 板上的上/下方按紐增加/減少音量等級，當連續長按按鈕超過一秒後，音量等級將每 0.2 秒增加/減少一級音量等級

關卡設定：

關卡只可以在遊戲尚未開始時或是結束時設定，短按中間按紐即可切換關卡及模式。

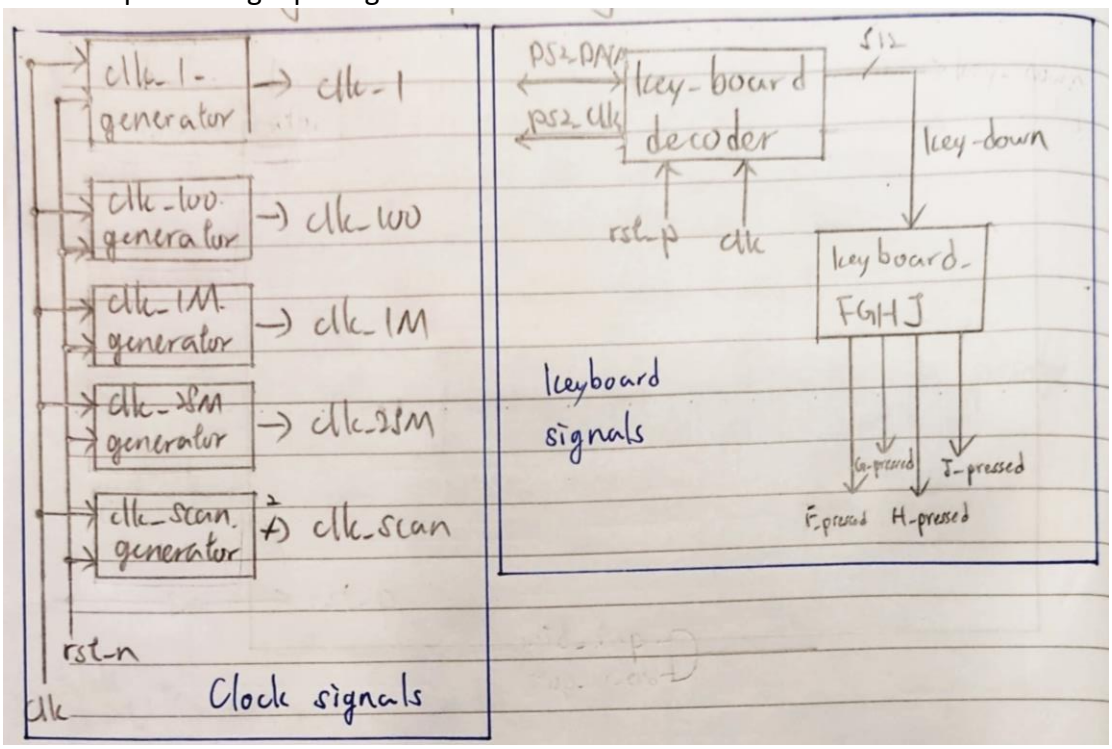
Design Specification

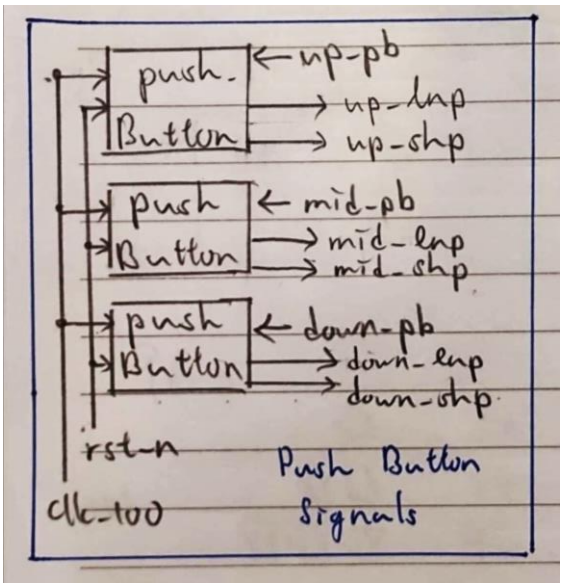
Input/Output Table

Input	Function	Output	Function
	↓Count from left		↓Count from left
Keyboard F	First row	LED * 3	Current mode
Keyboard G	Second row	LED * 1	Start
Keyboard H	Third row	LED * 2	Current level
Keyboard J	Forth row	LED * 10	Current volume
Button T18		SSD	
Button U17		VGA(LCD)	animation
Button U18		speaker	music
Button T17			
DIP switch V17	On/off		
DIP switch V16	Hack		

- Block diagram

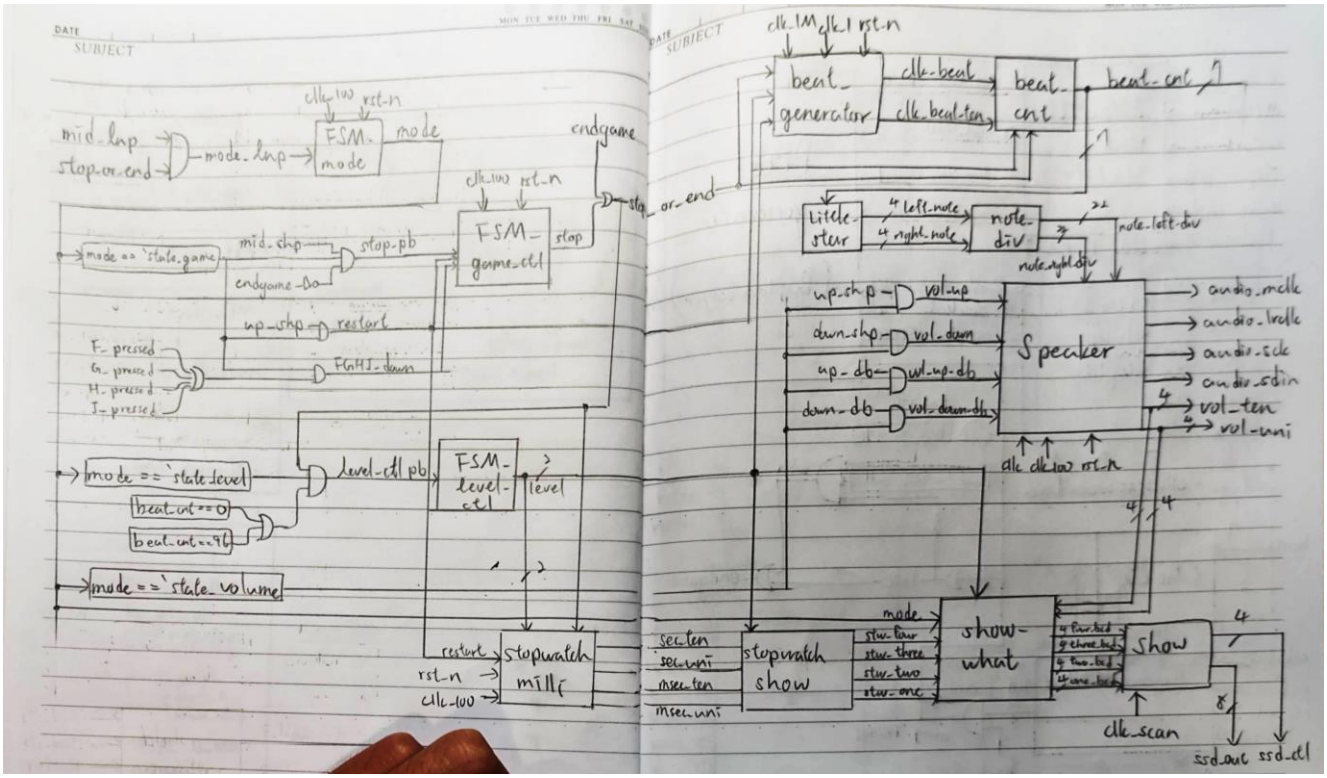
Top module
Modules processing input signals





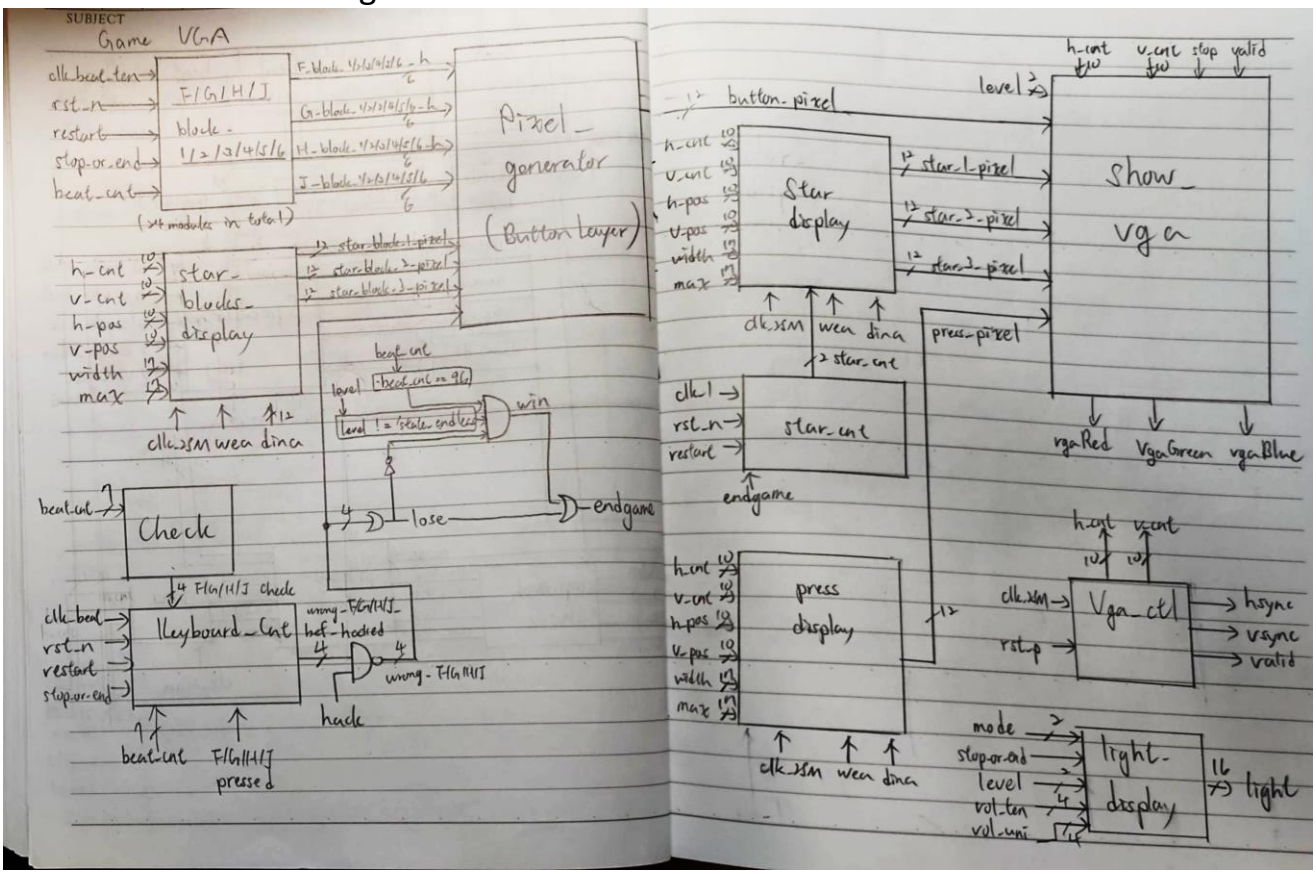
輸入的訊號會先經過上述的訊號處理來作為後續模組的利用

Modules related modes and beat controls



此部分的模組將會控制遊戲的進行，音量設定、等級設定等，以及七段顯示器的訊號和聲音模組的訊號。

Modules related to the game and the VGA



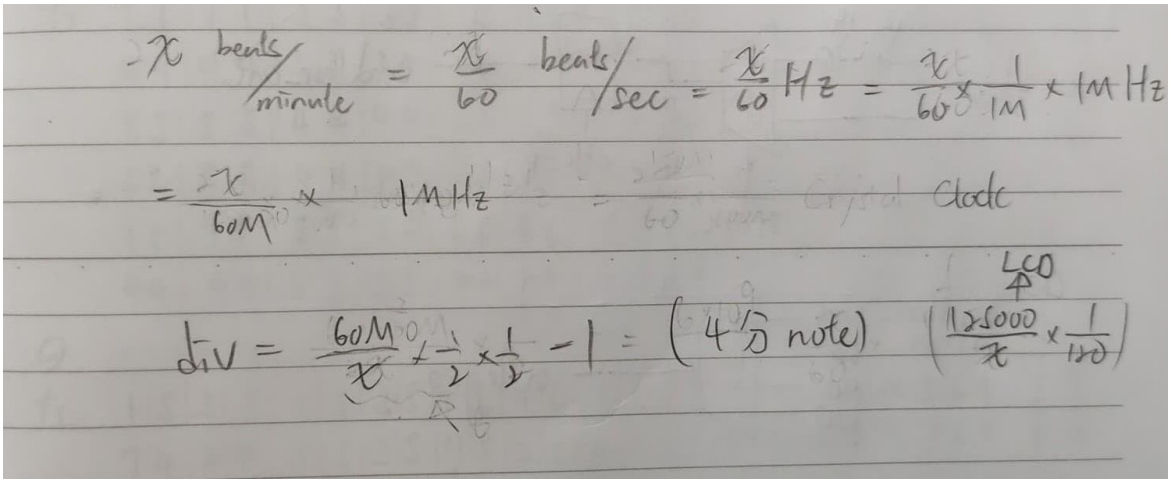
此部分模組將會決定遊戲是否結束，並分別製造出底層及各層的 VGA 的訊號，最後再利用多公器選擇輸出訊號。

Design Implementation

在此份報告中將不再贅述先前 lab 中已經實作過的功能

Beat_generator
此模組將分為兩個主要部分，首先將會由當前的 level 決定對應 BPM，若為無盡模式則利用一個 flip-flop 去紀錄及產生當下的 BPM 和下一秒的 BPM。

接著將產生的 BPM 換算成對應的數字供除法器分割時鐘，由於若使用 100MHz 的時鐘，被除數將會過大，因此，這邊使用 1MHz 的時鐘處理，處理後先產生 120 倍的時鐘供 LCD 顯示使用，再將此時鐘分割成所需的節拍（這樣子做才不會造成延遲，因為延遲會隨著樂曲的進行累積，而一個音樂遊戲若是音樂及畫面對不到拍，將不會是一個好的音樂遊戲）。計算過程如下：



Beat_cnt
此模組會計算遊戲開始後經過的拍數

聲音訊號

Little star
此模組會根據當前的 beat_cnt 產生對應的音階，傳到 note_div 裡面後轉換為對應的數字供 speaker 中的 buzzer ctl 發出對應的音階。

Volume ctl
此模組大致上與 lab7.2 中一樣，改動的點在於我新增了一個連續長按的功能，連續長按的功能是利用模運算實現的，當按鈕按下時，將會開始計數（100Hz），當計數器超過 100，並且除以 20 餘 2 時，將會加一，由於聲音控制只有 0~99，因此最高連續長按秒數為 99/5 約等於 20 秒，因此計數器最大需可記錄 20*100 = 2000 的數字，也就是 11 位數

遊戲判定

遊戲判定的難處在於，不僅僅要在黑色磚塊掉落時持續按下鍵盤，也要在白色磚塊掉落時放開，因此，最後決定利用黑磚出現的前綴和來判定

Check
此模組將輸出與輸入（beat_cnt）對應的前綴和，並傳送給 keyboard_cnt 檢驗

Keyboard cnt
此模組利用四個計數器分別記錄每排當下的數目，使用 posedge 觸發 flip-flop，當對應的鍵盤按下時加一，否則維持原本的數字。
利用輸入的 F/G/H/J check 檢查是否與當下的計數器相同，若不同則輸出 wrong_F/G/H/J_bef_hacked。

輸出的 wrong_F/G/H/J_bef_hacked 將與輸入的 hack 訊號做 NAND 運算，意即若 hack 等於 1，系統將不會報錯。

VGA 輸出

F/G/H/J_block_1/2/3/4/5/6

此二十四個模組分別計算每排出現的黑磚塊位置。由於螢幕最多同時可以顯示四個方塊，且紀錄的位置為磚塊的底部位置，因此需多備兩塊，作為上方即將掉落的磚塊及下方正在沉下去的磚塊。每排每個方塊都是獨立計算的，這樣才能做到將原黑磚塊改變為星星磚塊的操作。

一開始，除了第一拍跟第二拍的磚塊顯示於螢幕上以外（有一個預備拍），其他磚塊皆沉在螢幕下方，當節拍來到第 n-3 拍時，第 n 拍需抵達打擊線的黑色磚塊位置就會被設為螢幕上方並開始掉落。

Star_blocks_display

這邊總共含有三個模組，分別代表三個不同的星星磚（因為所在排數不同，若使用同一個模組將會出錯），每一個星星磚的 coe 檔含有 9600 個數字，也就是寬 80 乘以長 120，當星星磚距離打擊線還有兩格以上時，我將他的 pixel address 設為 120 減掉欲顯示的原黑磚塊高度（每個磚塊長 120pixels），以做到反轉的效果，並當原黑磚離打擊線距離小於兩格時，將他的 pixel address 設為欲顯示的原黑磚塊高度減掉 120，如此便可以在他距離等於兩格時定住，停止旋轉。

Pixel generator

此模組用於製照出最底層的圖層，會依據輸入的磚塊高生成黑色磚塊或是星星磚（在關卡模式當 beat_cnt 數到第 32、63、95 拍時出現），當黑色磚塊或是星星磚超過打擊線時則顯示藍色的磚塊，當玩家按錯時，則將按錯的那排抹上一層半透明的紅色，此外，此模組還會產生兩旁的背景以及打擊線。

Star display

這邊總共含有三個模組，分別代表三個不同的星星。星星位置有所不同，最中間星星最高，兩旁的星星較矮。此模組根據當前的 h_cnt 跟 v_cnt 輸出對應的 pixel。

Star_cnt

此模組用於計數，使星星在結束時可以一顆一顆的跳出來

Press_display

此模組將輸出字樣”Press any button to start”

Show_vga

這個模組將會把所有的圖層疊起來，先從最上面的圖層顯示，若最上面的圖層沒有要顯示東西，才會顯示下面的圖層，此外，我將”Press any button to start”的 pixel 與 1Hz 的時鐘做 and 運算，以達到閃爍的效果。

在此模組中我還做了去背的功能，首先需將欲去除的部分改為某一不用的顏色後轉成 coe 檔，（這邊我是把它改成白色），讀入時若是為白色，則顯示下面的圖層。

- I/O pin assignment

Input

System inputs

clk	rst_n	Hack
W5	V17	Should only be known by the designer

Buttons

mid_pb	up_pb	down_pb
U18	T18	U17

Inout

Keyboard

PS2_CLK	PS2_DATA
C17	B17

Output.

Speaker

audio_mclk	audio_lrck	audio_sck	audio_sdin
A14	A16	B15	B16

Seven-segment display

ssd_out 7	ssd_out 6	ssd_out 5	ssd_out 4	ssd_out 3	ssd_out 2	ssd_out 1	ssd_out0
W7	W6	U8	V8	U5	V5	U7	V7
ctl 3	ctl 2	ctl 1	ctl 0				
W4	V4	U4	U2				

VGA

vgaRed 0	vgaRed 1	vgaRed 2	vgaRed 3
N19	J19	H19	G19
vgaGreen 0	vgaGreen 1	vgaGreen 2	vgaGreen 3
D17	G17	H17	J17
vgaBlue 0	vgaBlue 1	vgaBlue 2	vgaBlue 3
J18	K18	L18	N18

Lights

Light 15	Light 14	Light 13	Light 12	Light 11	Light 10
L1	P1	N3	P3	U3	W3

Light 9	Light 8	Light 7	Light 6	Light 5	Light 4	Light 3	Light 2	Light 1	Light 0
V3	V13	V14	U14	U15	W18	V19	U19	E19	U16

```
light [15] = ( mode ==`state_game );
light [14] = ( mode ==`state_volume )
light [13] = ( mode ==`state_level )
light [12] = ~stop_or_end
light [11] = level [1]
light [10] = level [0]

light [9] = ( vol_ten >= 9 ) && ( vol_ten != 15 )
light [8] = ( vol_ten >= 8 ) && ( vol_ten != 15 )
light [7] = ( vol_ten >= 7 ) && ( vol_ten != 15 )
light [6] = ( vol_ten >= 6 ) && ( vol_ten != 15 )
light [5] = ( vol_ten >= 5 ) && ( vol_ten != 15 )
light [4] = ( vol_ten >= 4 ) && ( vol_ten != 15 )
light [3] = ( vol_ten >= 3 ) && ( vol_ten != 15 )
light [2] = ( vol_ten >= 2 ) && ( vol_ten != 15 )
light [1] = ( vol_ten >= 1 ) && ( vol_ten != 15 )
light [0] = ( vol_ten != 4'd0 || vol_uni != 4'd0 )
```

Discussion

- Implementing process

這次一樣先做了top module 然後才著手做其他模組，由於對VGA輸出不太有把握，因此，我先做了其他的功能，先做的功能有模式轉換、音量控制、音量輸出、等級控制、碼表計時（無盡模式的），這邊一如往常的出現了bug並在多次嘗試及更改後，才將所有的錯誤修掉，其中發現最大的問題就是我的計時器出現了問題，由於一開始，我使用的秒數計時器是用一赫茲的時鐘，而毫秒的是用一百赫茲的時鐘，雖然一開始是正常的，但隨著時間的增加，累積出來的延遲也越來越多，最後使得他們不同步，雖然說本來就知道會有延遲差異，但一開始時做的時候想說延遲很小應該不影響，然而，此種會累計的延遲會像滾雪球一樣越滾越大，還好改完之後就沒什麼問題了。

值得慶信的是，在這個時候出現了這樣子的問題，提醒了我，對於聲音輸出跟螢幕輸出需要使用一樣的時鐘，否則隨著音樂的撥放，其中的延遲也將會越來越多。

在寫VGA輸出時，我改了很多次，首先是視覺暫留問題，我本來讓我的方塊一次下降12 pixel，然而，這樣的速度會因為視覺暫流的關係使玩家的頭很暈，後來我便將移動速率改為1 pixel 每次，改完以後的感覺好多了。再來是遊戲性的問題，本來對於節拍的精準度為100%準確，但這樣其實會讓玩家很挫折，因為真的太難了，後來在多次測試後，誤差冗忍率從0%改為了2.5%，又改為了5%，最後將他改為了10%，改為10%以後對於玩家的遊戲體驗最佳，如果再更大就會照成太簡單。

在來是一開始磚塊的畫面跟遊戲判定對不上，經過無數次的觀察後，我發現是因為我的clk_beat的初始值是0，因此只經過了半個週期變會觸發beat_cnt（因為我這個部分是使用posedge trigger），後來將初始值設為1以後就正常了。

在確認底層的圖層沒問題後，我開始研究如何使用coe檔貼上圖片，之後先貼了方形含背景的圖片以後，才慢慢加了去背的效果，後來的會旋轉至定位星星磚，和遊戲暫停或開始時的閃爍指導語。

最後，當我發現自己手不夠靈活的時候，才加上了hack開關，這樣才能讓我 ~~成功拿到三顆星~~ demo 所有的功能。

Conclusion

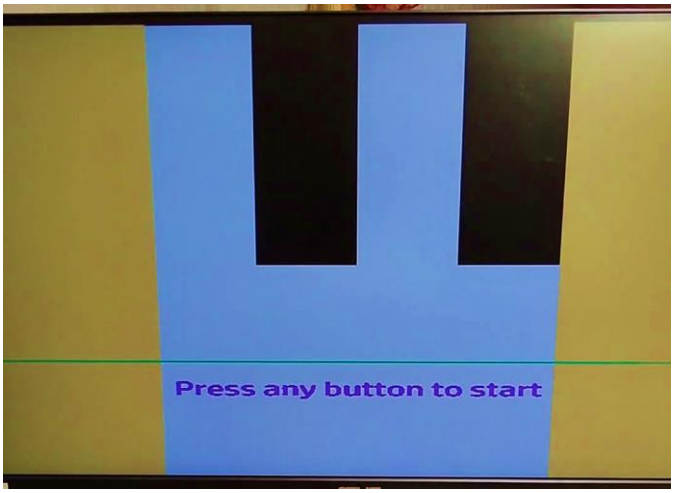
我覺得期末專題讓我學到了很多，也讓我很有成就感，因為有前面的每次lab的訓練，在這次的專題中，我一開始的架構就寫得夠大，寫得夠活，供我後續做修改及增加更多功能。在這次的專題中，我把所有學過的東西都用了進去，也甚至自學了如何輸出VGA，在成功輸出的那一刻真的很開心，因為螢幕帶給我的資訊量及視覺享受遠大於FPGA上面只有亮跟暗的顯示。雖然我不是很擅長音樂遊戲，但我還是玩了我的遊戲好多次，拿了好多次的一顆星。

尚未解決的問題：按暫停的時候不一定會跟螢幕移動的clock切齊，因此會在按鈕中間出現一條白白的（矯正節拍後的結果），目前想到的改善方法是將clock切的更細，讓肉眼無法分辨出來，但是在無盡模式中，速度會越來越快，最後再短暫的時間也會在螢幕上被放大。

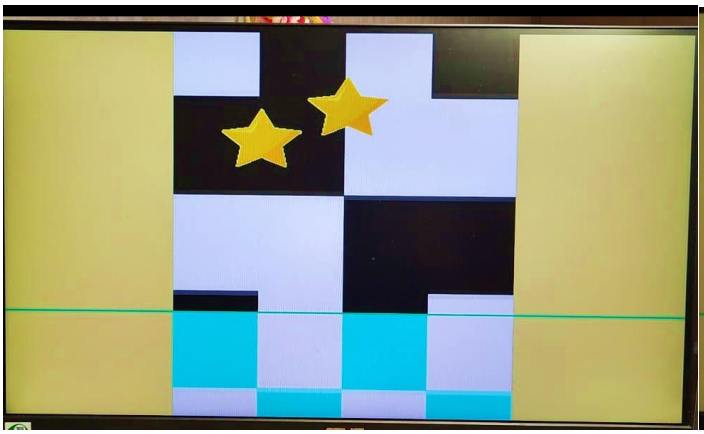
附錄

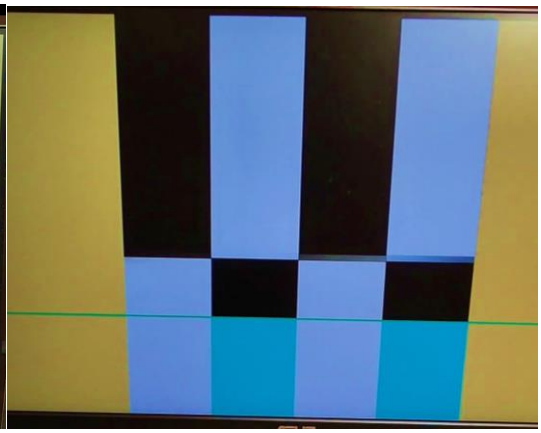
遊戲畫面

開始畫面



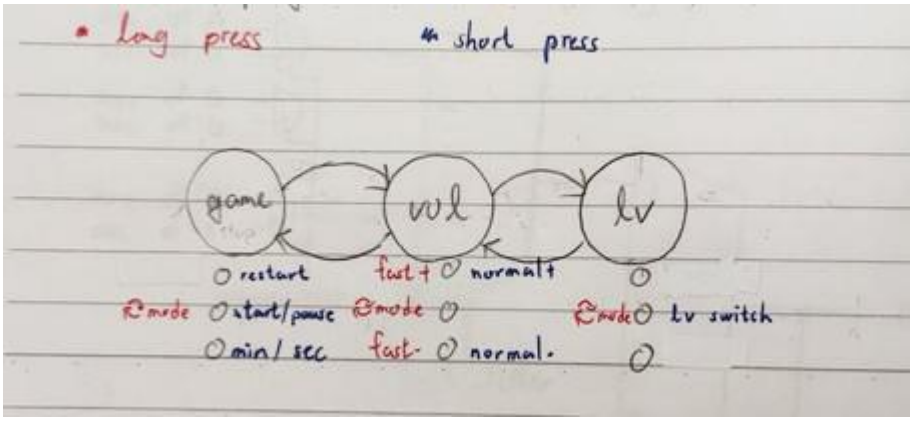
遊戲進行畫面



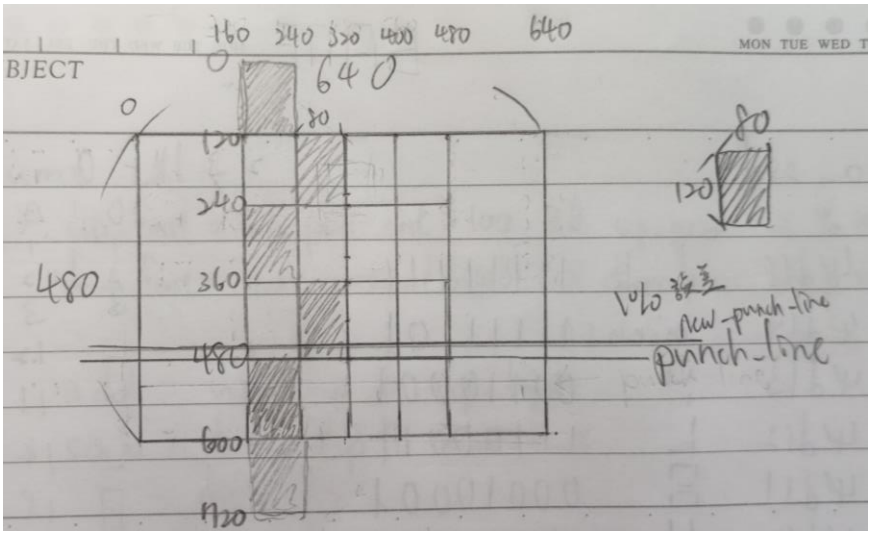


遊戲結算畫面

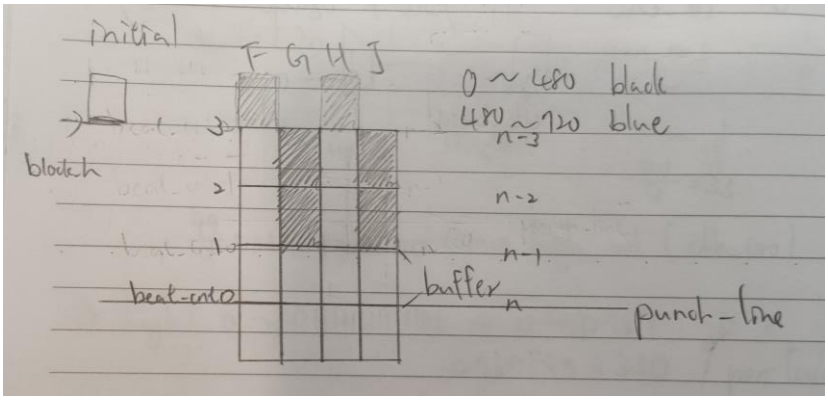
遊戲操作指令圖



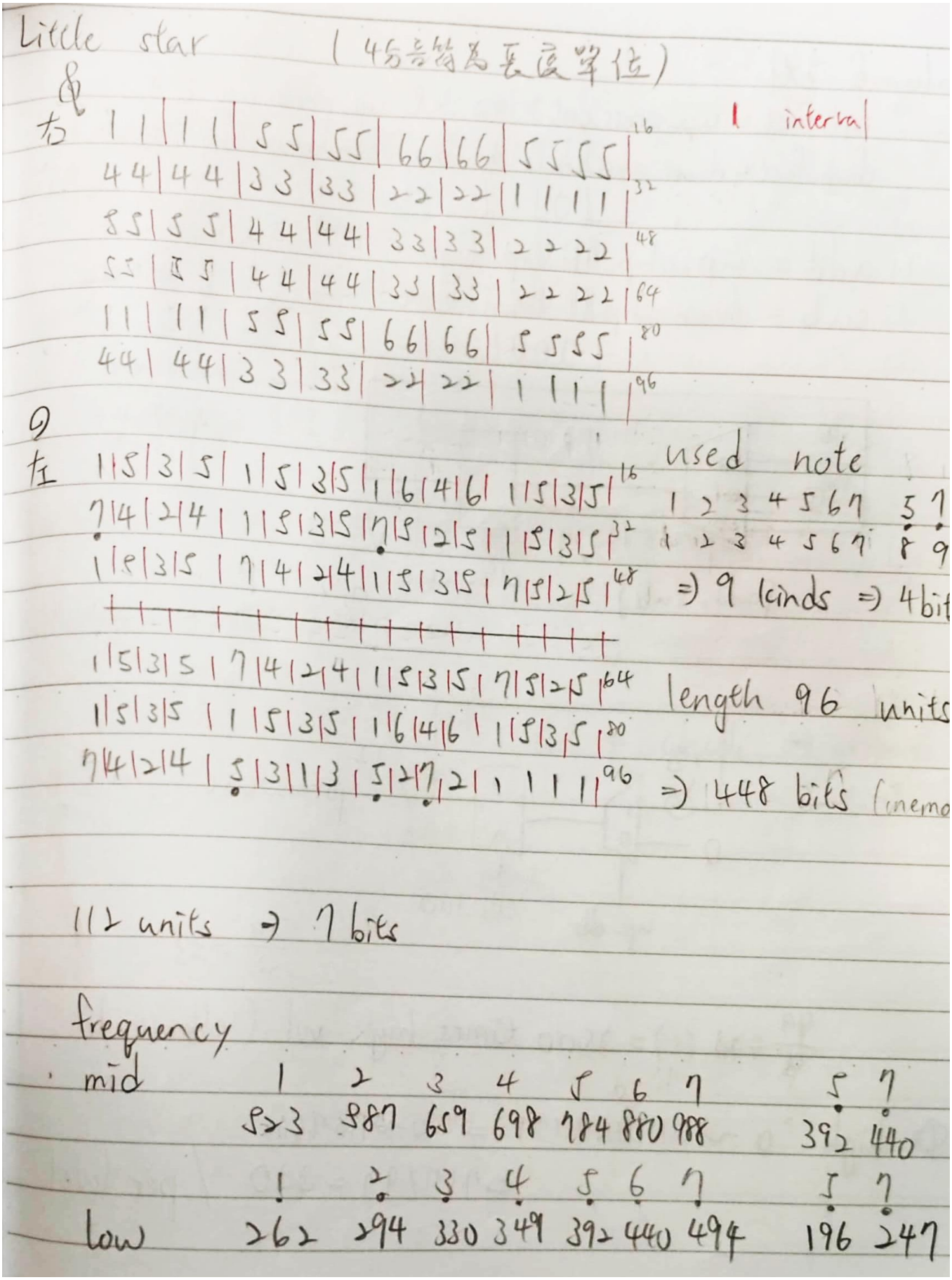
遊戲設計圖及自定義座標軸



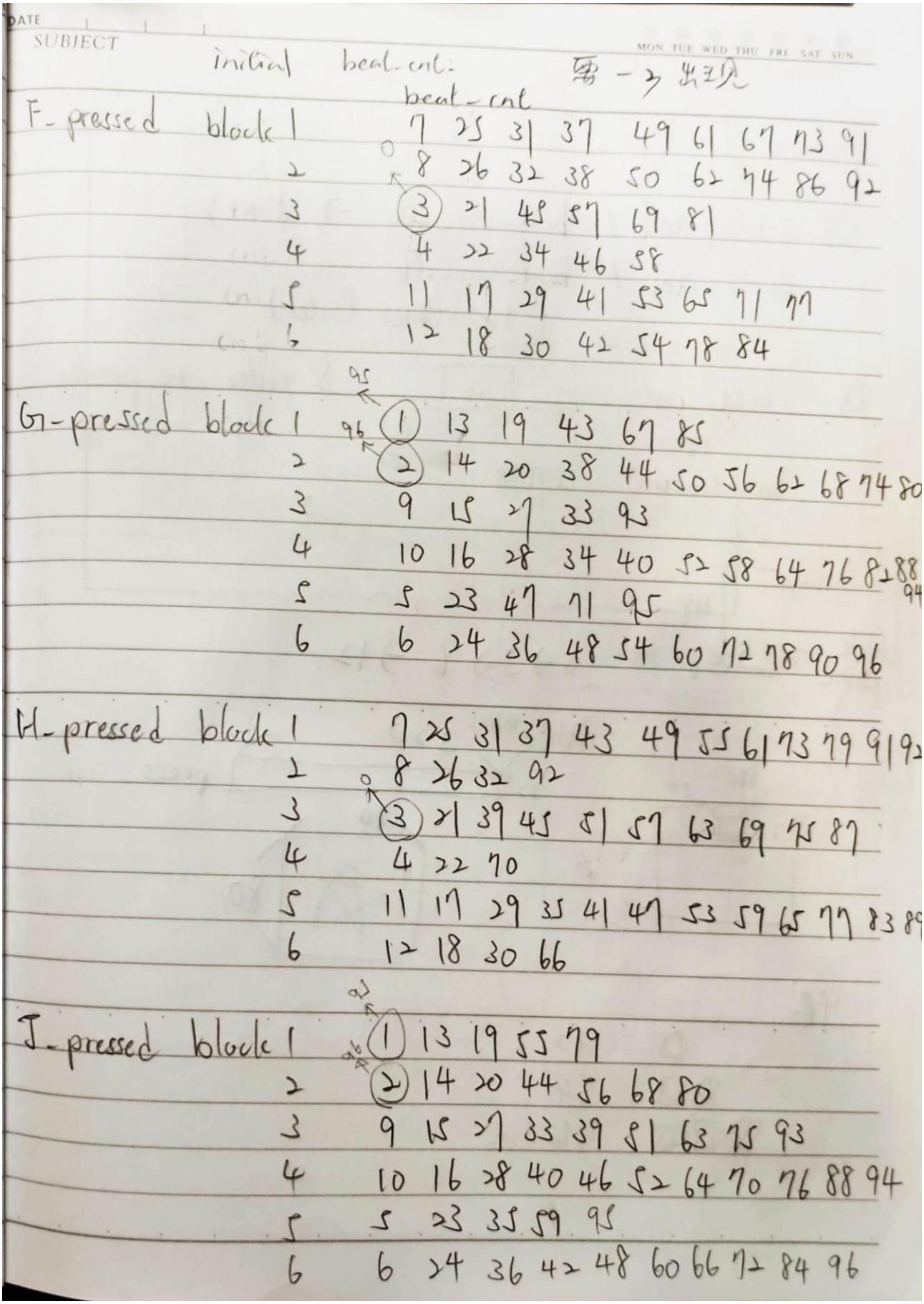
遊戲初始畫面設計



使用音樂之樂譜及對應記號和頻率



磚塊出現時機設計


DATE
SUBJECT
MON TUE WED THU FRI SAT SUN
initial
beat_cnt:
beat_cnt
F-pressed block 1 7 25 31 37 49 61 67 73 91
2 8 26 32 38 50 62 74 86 92
3 3 21 45 57 69 81
4 4 22 34 46 58
5 11 17 29 41 53 65 71 77
6 12 18 30 42 54 78 84
G-pressed block 1 1 13 19 43 67 85
2 2 14 20 38 44 50 56 62 68 74 80
3 9 15 27 33 93
4 10 16 28 34 40 52 58 64 76 82 88
5 5 23 47 71 95
6 6 24 36 48 54 60 72 78 90 96
H-pressed block 1 7 25 31 37 43 49 55 61 73 79 91
2 8 26 32 92
3 3 21 39 45 51 57 63 69 75 87
4 4 22 70
5 11 17 29 35 41 47 53 59 65 77 83
6 12 18 30 66
J-pressed block 1 1 13 19 55 79
2 2 14 20 44 56 68 80
3 9 15 27 33 39 51 63 75 93
4 10 16 28 40 46 52 64 70 76 88 94
5 5 23 35 59 95
6 6 24 36 42 48 60 66 72 84 96

遊戲圖及前綴和

黑色部分為黑色磚塊，橘色磚塊為掉落的星星磚

右方四排為 F/G/H/J（由左至右）的黑色磚塊數前綴和

	1		1		0	1	0	1
	1		1		0	2	0	2
1		1			1	2	1	2
1		1			2	2	2	2
	1		1		2	3	2	3
	1		1		2	4	2	4
1		1			3	4	3	4
1		1			4	4	4	4
	1		1		4	5	4	5
	1		1		4	6	4	6
1		1			5	6	5	6
1		1			6	6	6	6
	1		1		6	7	6	7
	1		1		6	8	6	8
	1		1		6	9	6	9
	1		1		6	10	6	10
1		1			7	10	7	10
1		1			8	10	8	10
	1		1		8	11	8	11
	1		1		8	12	8	12
1		1			9	12	9	12
1		1			10	12	10	12
	1		1		10	13	10	13
	1		1		10	14	10	14
1		1			11	14	11	14
1		1			12	14	12	14
	1		1		12	15	12	15
	1		1		12	16	12	16
1		1			13	16	13	16
1		1			14	16	14	16

1		1			15	16	15	16
1		1			16	16	16	16
	1		1		16	17	16	17
1	1				17	18	16	17
		1	1		17	18	17	18
	1		1		17	19	17	19
1		1			18	19	18	19
1	1				19	20	18	19
		1	1		19	20	19	20
	1		1		19	21	19	21
1		1			20	21	20	21
1			1		21	21	20	22
	1	1			21	22	21	22
	1		1		21	23	21	23
1		1			22	23	22	23
1			1		23	23	22	24
	1	1			23	24	23	24
	1		1		23	25	23	25
1		1			24	25	24	25
1	1				25	26	24	25
		1	1		25	26	25	26
	1		1		25	27	25	27
1		1			26	27	26	27
1	1				27	28	26	27
		1	1		27	28	27	28
	1		1		27	29	27	29
1		1			28	29	28	29
1	1				29	30	28	29
		1	1		29	30	29	30
	1		1		29	31	29	31
1		1			30	31	30	31
1	1				31	32	30	31

		1	1		31	32	31	32
	1		1		31	33	31	33
1		1			32	33	32	33
		1	1		32	33	33	34
1	1				33	34	33	34
	1		1		33	35	33	35
1		1			34	35	34	35
		1	1		34	35	35	36
1	1				35	36	35	36
	1		1		35	37	35	37
1		1			36	37	36	37
1	1				37	38	36	37
		1	1		37	38	37	38
	1		1		37	39	37	39
1		1			38	39	38	39
1	1				39	40	38	39
		1	1		39	40	39	40
	1		1		39	41	39	41
1					40	41	39	41
	1				40	42	39	41
		1			40	42	40	41
1			1		41	42	40	42
	1				41	43	40	42
1					42	43	40	42
		1			42	43	41	42
	1		1		42	44	41	43
		1			42	44	42	43
	1				42	45	42	43
1		1			43	45	43	43
1		1			44	45	44	43
	1		1		44	46	44	44
	1		1		44	47	44	45

	1		1		44	48	44	46