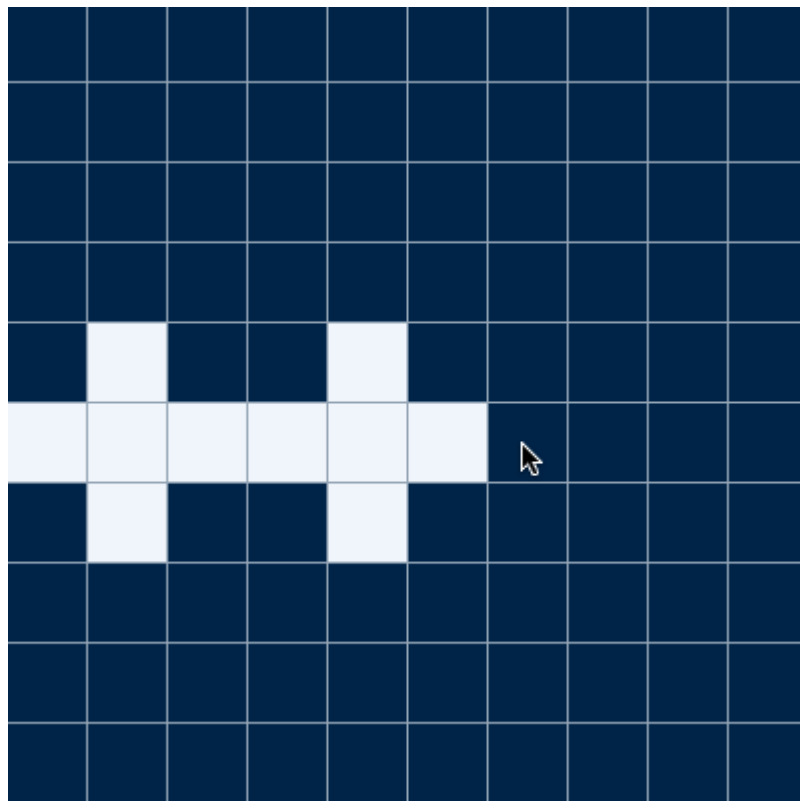


E1: 開關燈 (たこ焼き)

- 滑鼠點擊的該格以及其上下左右相鄰的格子會觸發開關燈事件
- 原本開燈的會關燈，原本關燈的會開燈



規格說明

```
int cols = 10  
int rows = 10
```

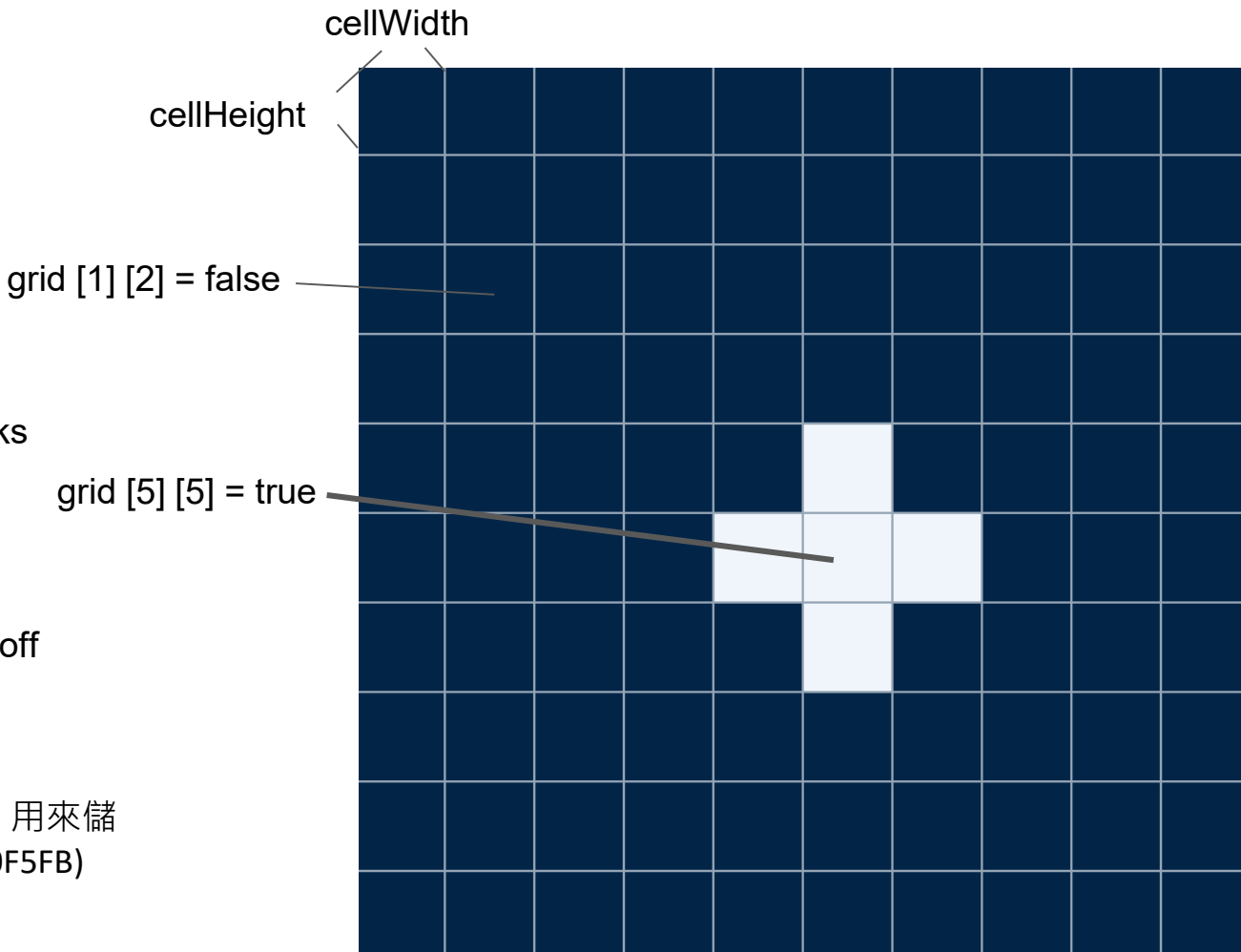
```
boolean[][] grid ; // 10x10 blocks
```

程式碼預先完成的部分：
依grid[][]中的true/false決定 on/off

```
color turnOnColor = #F0F5FB ;
```

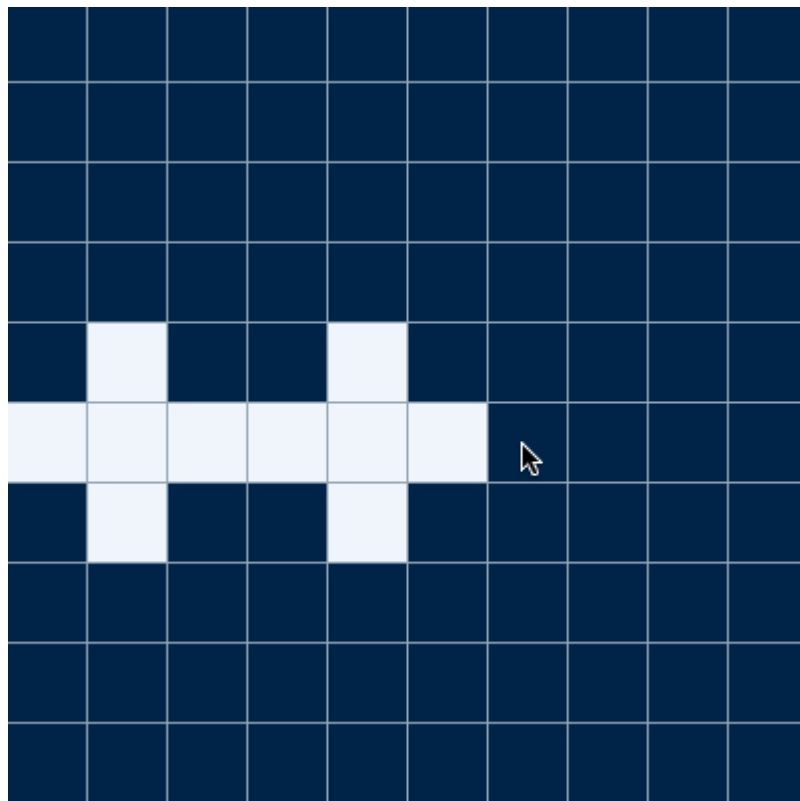
```
color turnOffColor = #022547 ;
```

color 是processing的資料型態，用來儲存顏色，可以存放色碼 (ex.#F0F5FB)



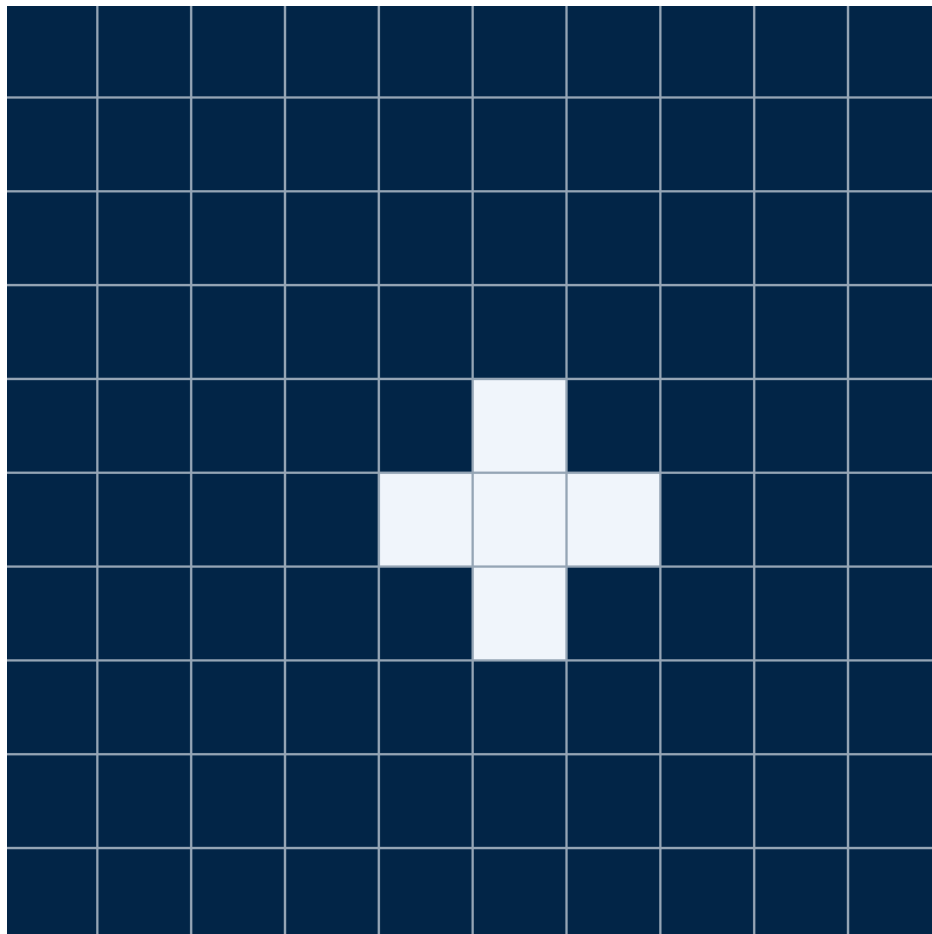
Part 1 要求一

- 滑鼠點擊的該格以及其上下左右相鄰的格子會觸發開關燈事件
- 原本開燈的會關燈，原本關燈的會開燈



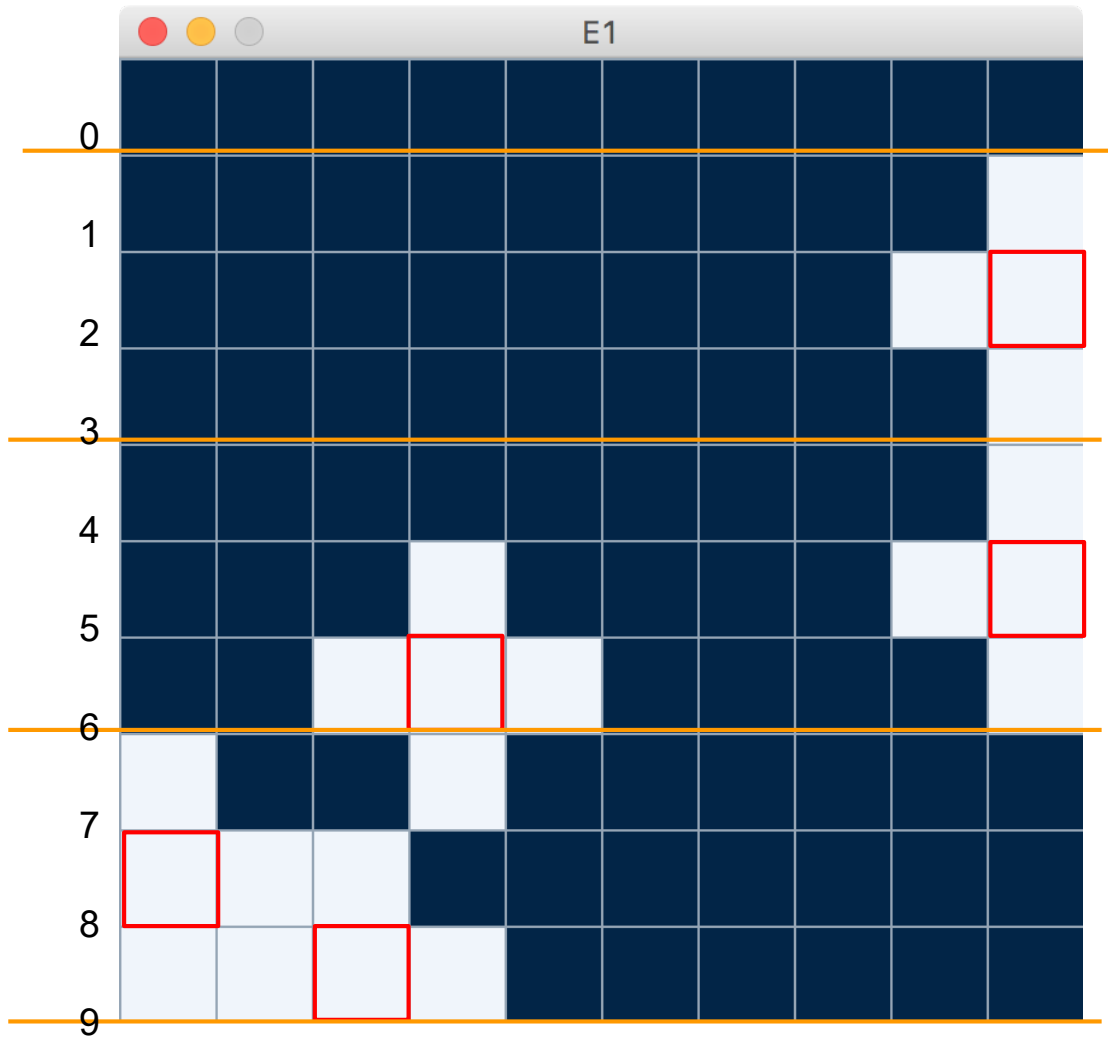
Part 1 提示

- 偵測點擊位置：`mouseX/cellWidth`
- 設定開關燈時須考慮邊界判斷，以免 `ArrayIndexOutOfBoundsException`



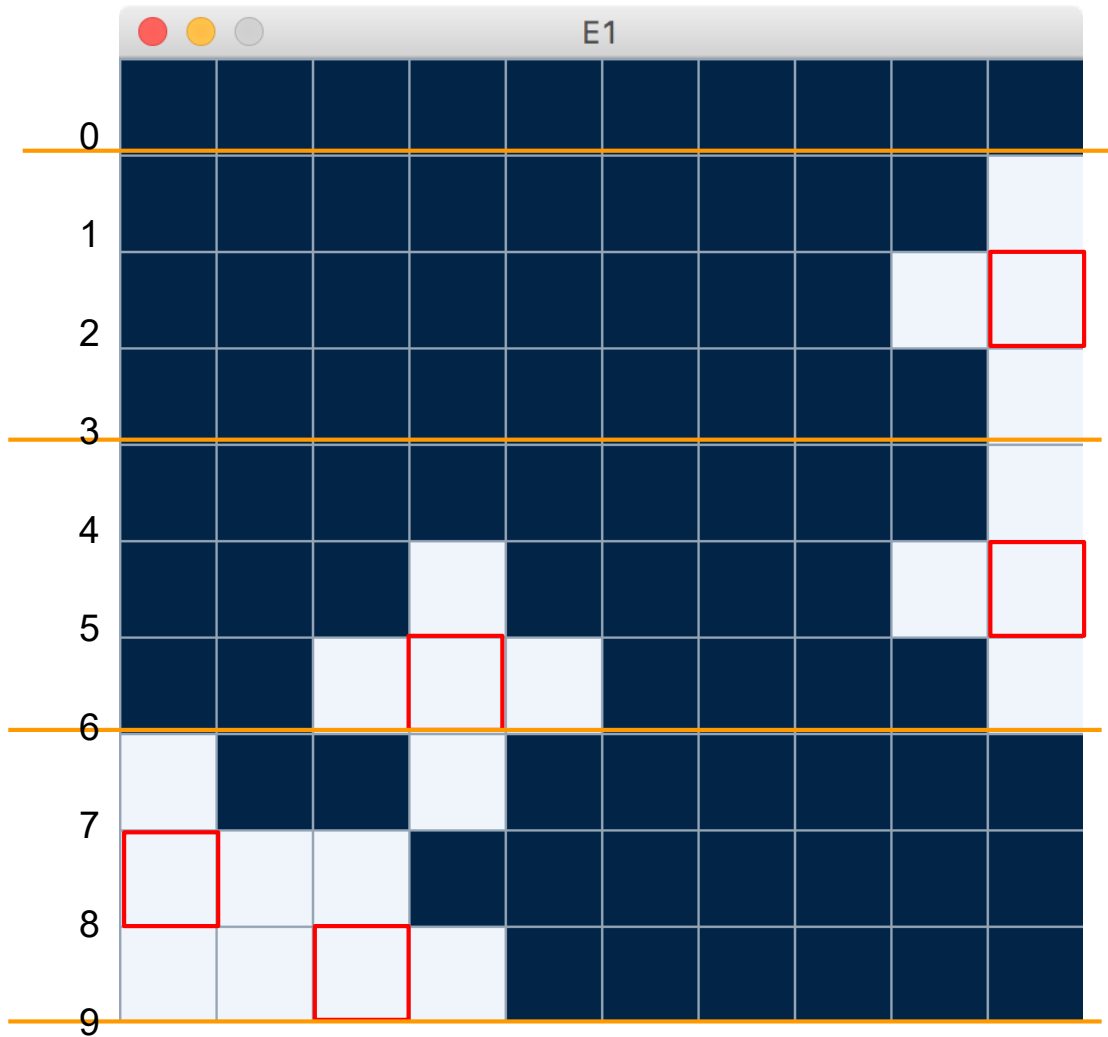
Part 2

- 遊戲開始時，在第1-9列之間，每三列隨機選 1-2格讓進行開關燈的動作



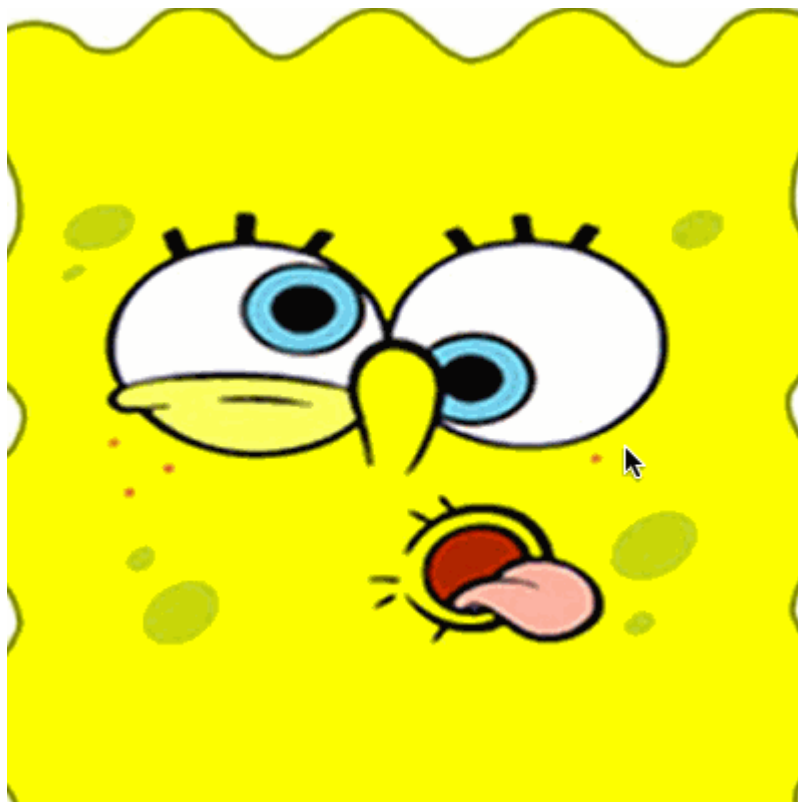
Part 2

- 先隨機決定這一區要點一次還兩次，再隨機決定點的位置
- 記住上次點的位置，如果第二次隨機與記住的位置相同就再隨機一次



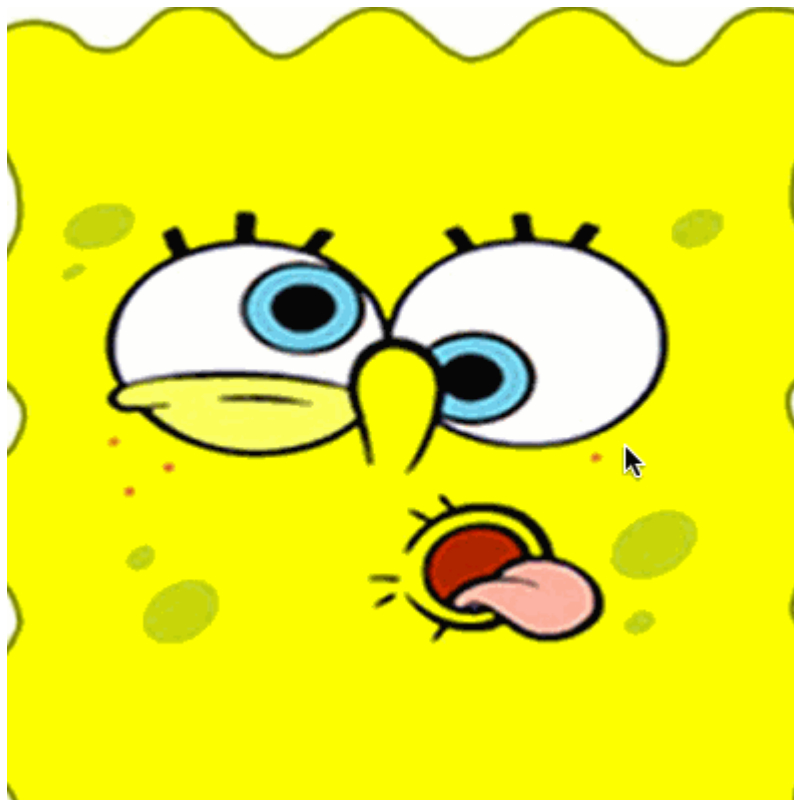
E2：幫海綿寶寶修容吧

- 以滑鼠位置為圓心，做出半徑為十的修容筆刷，取範圍內的畫素顏色平均值為筆刷顏色
- 透過滑鼠點擊著色



E2：幫海綿寶寶修容吧

- [loadPixels\(\)](#)
- [pixels\[\]](#)
- [updatePixels\(\)](#)




E2: 幫海綿寶寶修容吧

pixels[i]

$i = y * \text{width} + x$

$(5, 4) = 4 * 10 + 5 = 45$

pixels[45] => 

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29
30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49
50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69
70	71	72	73	74	75	76	77	78	79

E2：幫海綿寶寶修容吧

image()

把圖片畫到畫面上

loadPixels()

將目前畫面所有像素讀取進入pixels[]陣列中

mousePressed

判斷滑鼠是否壓下，
若 true 使用滑鼠位置判斷哪些像素需要處理
讀取pixels[]計算結果後更新pixels[]對應像素

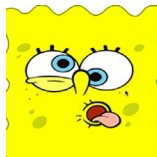
updatePixels()

將pixels[]陣列讀取進入畫面中

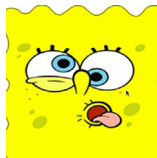
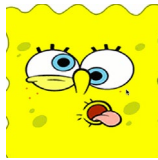
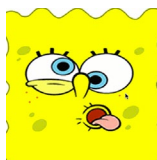
循環

畫面

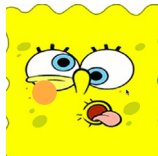
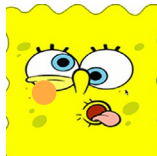
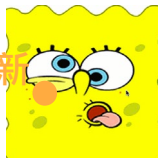
pixels[]



(空)



更新



E2：幫海綿寶寶修容吧

均化效果：

如何平均所有像素顏色？

- 使用`red(color)`, `green(color)`, `blue(color)`取得範圍內每個pixel的RGB
- 平均 = 總和 / 個數，PS: 小心除數不能為零
- 計算完畢後用`pixels[i] = color(r, g, b)`將各個pixel更新

提示：取得半徑範圍內pixels

- 邊界處理：避免讀取到畫布範圍外，以防出現錯誤

