

組合語言與系統程式期末專題報告

第 28 組

108502510 蘇洪瑤 107303001 張凱晴

108502508 莊智凌 108502509 陳昕妤

一、分工

VS 視窗架構：張凱晴

明暗：蘇洪瑤、陳昕妤

放大：蘇洪瑤、張凱晴

縮小：莊智凌、蘇洪瑤

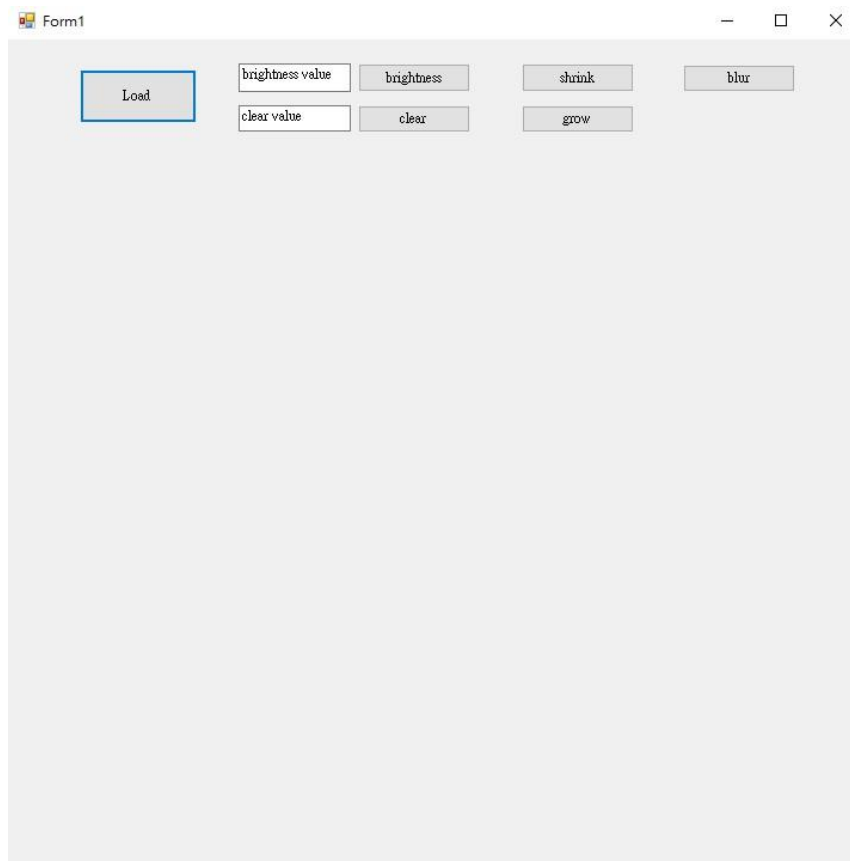
模糊：莊智凌、陳昕妤

填滿：陳昕妤、莊智凌

二、使用函式庫

Visual C++、Irvine32.inc

三、操作說明



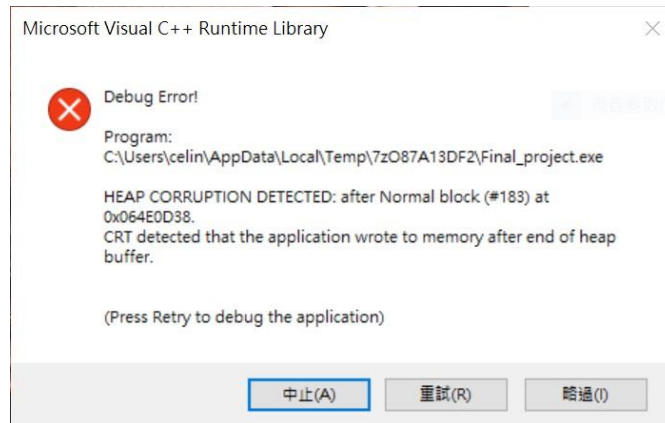
1. 選取 Load 讀取 bmp 圖檔

注意：圖片僅支援 24 位元圖像深度

2. 選取功能

注意：

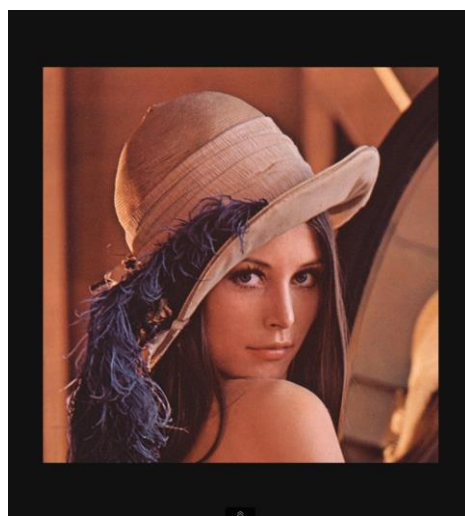
- (1) 模糊功能僅有第一次有效果
- (2) 若 shrink 和 grow 功能加起來按超過 2 次以上，會出現 error 警告視窗如下：



若按略過可以繼續使用。（上網查詢後大概知道是陣列的記憶體問題，但因 visual studio 會自行多開幾個記憶體空間所以執行上不會出問題，我們 debug de 不出來）

- i. brightness：輸入任意數值決定欲調整明暗程度，實際有效數值為-255~255。
 - ii. clear：輸入 0~255 填滿整張圖片
 - iii. shrink：長寬變為本來的一半
 - iv. grow：長寬變為本來的兩倍
 - v. blur：使圖片模糊化
3. 顯示圖片，另存圖片到與原圖相同的資料夾位置

四、Demo 截圖



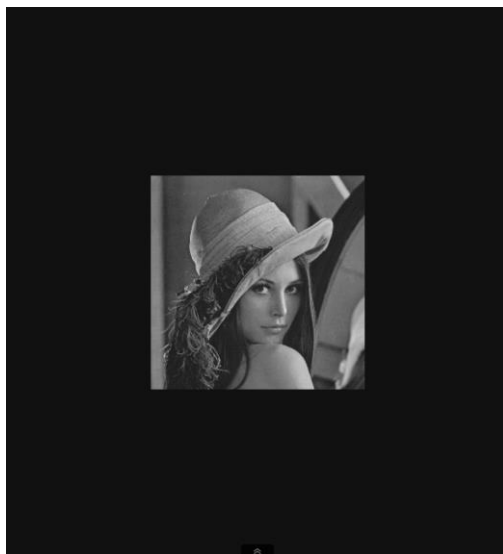
1. 原圖



2. brightness 數值 200 後



3. clear 後



4. shrink 後



5. grow 後



6. blur 後

五、Function 說明

1. brightness

```
4
5 ; STACK
6 ; | ebp
7 ; | return address
8 ; | image1原圖位址
9 ; | width寬度
10 ; | height長度
11 ; | value亮度調整值
12
13 brightness proc
14     push ebp
15     mov ebp, esp
16     push ebx
17
18     mov ebx, [ebp+12] ;存取圖片寬度
19     mov eax, [ebp+16] ;存取圖片高度
20     mul ebx           ;長*寬=總pixel數 設為loop counter
21     mov ecx, eax
22     xor ebx, ebx      ;ebx清零
23     add ebx, [ebp+20] ;存取亮度調整值
24     mov edx, [ebp+8]  ;存取圖片位址
25
26 loopb:
27     xor eax, eax      ;eax清零
28     dec ecx           ;ecx--
29     mov al, [edx]     ;存取當前圖片位址的亮度值(灰階)
30     add eax, ebx      ;更改當前亮度值(灰階)
31     cmp eax, 255      ;若更改後超過255調降至255
32     jle test1
33     mov eax, 255
34
35 test1:
36     cmp eax, 0        ;若更改後低於0調升至0
37     jge forward1
38     mov eax, 0
39
40 forward1:
41     mov [edx], al     ;確定調整後介於0~255再放回去
42     inc edx           ;移到下一個pixel
43     cmp ecx, 0        ;當ecx跑完結束
44     jne loopb
45
46     pop ebx
47     mov esp, ebp
48     pop ebp
49     ret
50 brightness endp
51
52 end
```

傳入 image 的資料、寬、高和輸入欲更改的值。

將資料位址存入 esp 再存進 ebp 裡，width 存入 ebx，length 存入 eax。把 eax 乘上 ebx 計算有幾個 pixel，將之放進 ecx 進行 loop。

ebx 歸零後，存入欲更改的值，edx 存進目前的 pixel。

loopb：將 eax 歸零，ecx--，將目前的 pixel([edx])存進 al，再將 ebx(欲更改的值)加至 eax。檢查 eax 是否超過 255 或低於 0，若是超過 255 則更改為 255，低於 0 則更改為 0。最後將更改完的圖片資料(ebp)存回 esp。

2. clear

```
1  .model flat, C
2
3  .code
4
5  clear proc
6      push ebp
7      mov ebp, esp
8
9      push ebx
10     mov ebx, [ebp+12] ;width
11     mov eax, [ebp+16] ;height
12     mul ebx           ;總像素(width*height), 迴圈次數
13     mov ecx, eax
14     xor ebx, ebx
15     add bl, [ebp+20]  ;設定的油漆桶value
16     mov edx, [ebp+8] ;edx存圖片起始位址
17
18 loop1:
19     dec ecx
20     mov [edx], bl     ;將油漆桶數值填入此pixel
21     inc edx           ;下一個pixel
22     cmp ecx, 0        ;檢查是否填完了
23     jne loop1
24
25     pop ebx
26     mov esp, ebp
27     pop ebp
28     ret
29 clear endp
30
31 end
32
```

傳入 image 的資料、寬、高和輸入欲更改的值。

將資料位址存入 esp 再存進 ebp 裡，width 存入 ebx，length 存入 eax。把 eax 乘上 ebx 計算有幾個 pixel，將之放進 ecx 進行 loop。ebx 歸零後，存入欲更改的值，edx 存進目前的 pixel。

loop1：ecx--，把 bl(欲更改的值)直接存入[edx]（顏色直接覆蓋），edx++(移至下一個 pixel)。將更改完的圖片資料(ebp)存回 esp。

3. shrink

```
1  .model flat, C
2
3  .code
4
5  shrink proc
6      push ebp
7      mov ebp, esp
8      push ebx
9
10     mov ecx, [ebp+20]    ;外圈height
11
12     mov ebx, [ebp+12]    ;縮小完的圖片存ebx
13     mov edx, [ebp+8]    ;edx存圖片起始位址
14
15 looph:
16     dec ecx
17     dec ecx
18     push ecx
19     mov ecx, [ebp+16]    ;內圈width
20     loopl:
21         dec ecx
22         dec ecx
23
24         mov eax, [edx]    ;目前的pixel
25         mov [ebx], eax    ;存進ebx
26         inc ebx
27         inc edx            ;存一個跳一個來縮小(edx+2)
28         inc edx
29
30         cmp ecx, 0
31         jne loopl
32
33     pop ecx
34     mov eax, edx
35     add eax, [ebp+16]    ;原本的跳下一行
36     mov edx, eax
37
38     cmp ecx, 0
39     jne looph
40
41     pop ebx
42     pop ebp
43     ret
44 shrink endp
45
46 end
```

傳入 image1 的資料、image2 的資料、寬和高。

將資料位址存入 esp 再存進 ebp 裡，height 存入 ecx，ebx 是 image2 的 pixel，edx 是 image1 的 pixel。

looph: ecx-=2，因為縮小成原本的 1/2，所以只需要原本長度的一半次數。Push ecx，width 存入 ecx。

loopl: ecx-=2，因為縮小成原本的 1/2，所以只需要原本寬度的一半次數。先把 image1 目前的 pixel 存進 eax，再存入 image2 也就是 ebx。ebx 進到下一個 pixel，edx 進到下兩格的 pixel，因為要縮小，所以只存取原本的一半 pixel。

4. grow

```
1  .model flat, C
2
3  .code
4
5  grow proc
6      push ebp
7      mov ebp, esp
8      push ebx
9
10     mov ecx, [ebp+20] ;外圈height
11
12     mov ebx, [ebp+12] ;放大完的圖片存ebx
13     mov edx, [ebp+8] ;edx存圖片起始位址
14
15 looph:
16     dec ecx
17     push ecx
18     mov ecx, [ebp+16] ;內圈width
19     loop1:
20         dec ecx
21
22         mov eax, [edx] ;eax存原本圖片的pixel
23
24         ;原本一格複製成2*2四格
25         mov [ebx], eax ;左上
26         inc ebx
27         mov [ebx], eax ;右上
28         dec ebx
29
30         add ebx, [ebp+16] ;加兩次原本的width到下一行
31         add ebx, [ebp+16]
32
33         mov [ebx], eax ;左下
34         inc ebx
35         mov [ebx], eax ;右下
36         inc ebx
37
38         sub ebx, [ebp+16] ;減回去原本那行
39         sub ebx, [ebp+16]
40
41         inc edx ;edx++準備複製下一個pixel
42         cmp ecx, 0
43         jne loop1
44
45     pop ecx
46     add ebx, [ebp+16]
47     add ebx, [ebp+16]
48
49     cmp ecx, 0
50     jne looph
51
52     pop ebx
53     pop ebp
54     ret
55 grow endp
56
```

傳入 image1 的資料、image2 的資料、寬和高。

將資料位址存入 esp 再存進 ebp 裡，height 存入 ecx，ebx 是 image2 的 pixel，edx 是 image1 的 pixel。

looph：ecx—(height)，push ecx，ecx 存入 width。

loop1：ecx—(width)，將 eax 存入 image1 的 pixel，把 eax 在 image2 中複製成 2*2 的正方形，結束後移至下一個 pixel。

5. blur

```
4
5 ; STACK
6 ; | ebp
7 ; | return address
8 ; | image1 原圖位址
9 ; | width 寬度
10 ; | height 高度
11
12 blur proc
13     push ebp
14     mov ebp, esp
15     push esi
16     push ebx
17
18     mov ecx, [ebp+16] ; 存取圖片高度
19
20     mov esi, [ebp+8] ; 存取圖片位址
21
22 looph:
23     dec ecx
24     dec ecx
25     push ecx
26     mov ecx, [ebp+12] ; 存取圖片寬度
27     loop1:
28         dec ecx
29         dec ecx
30
31         xor eax, eax ; eax 清零
32         xor ebx, ebx ; ebx 清零
33         xor edx, edx ; edx 清零
34         mov al, [esi] ; al 存檔的pixel
35         inc esi ; 右移一pixel
36         mov bl, [esi] ; bl 存檔的右邊的pixel
37         add ax, bx ; ax = ax + bx
38         dec esi
39         add esi, [ebp+10] ; 下移一列 (i += width)
40         mov bl, [esi] ; bl 存檔的下面的pixel
41         inc esi ; 右移一pixel
42         mov dl, [esi] ; dl 存檔的右下的pixel
43         add bx, dx ; bx = bx + dx
44         add ax, bx ; ax = ax + bx (4個pixel相加)
45
46         shr eax, 2 ; eax /= 4
47
48         mov [esi], al ; 模糊(平均)後的pixel存回去
49         dec esi
50         mov [esi], al
51         sub esi, [ebp+12]
52         mov [esi], al
53         inc esi
54         mov [esi], al
55
56         inc esi ; 移至下一個未模糊的pixel
57
58         cmp ecx, 0
59         jna loop1
60 ; loop1結束
61
62     pop ecx
63     add esi, [ebp+12] ; 移到下一列繼續
64     cmp ecx, 0
65     jna looph
66 ; looph結束
67
68     pop ebx
69     pop esi
70     pop ebp
71     ret
72 blur endp
73
74 end
```

傳入 image 的資料、寬和高。

將資料位址存入 esp 再存進 ebp 裡，將 height 存入 ecx，image 目前的 pixel 存入 esi。

looph：ecx-2，因為我們的模糊是將自己和自己右邊、下面和右下的 pixel 進行平均，再存回那四格裡，所以要換至下一個未模糊的 pixel，也就是下兩格。

loop1：將上述提到的 4 格做平均，平均後的數值存進 al 裡，再放回那 4 格中，藉此達到模糊的目標。