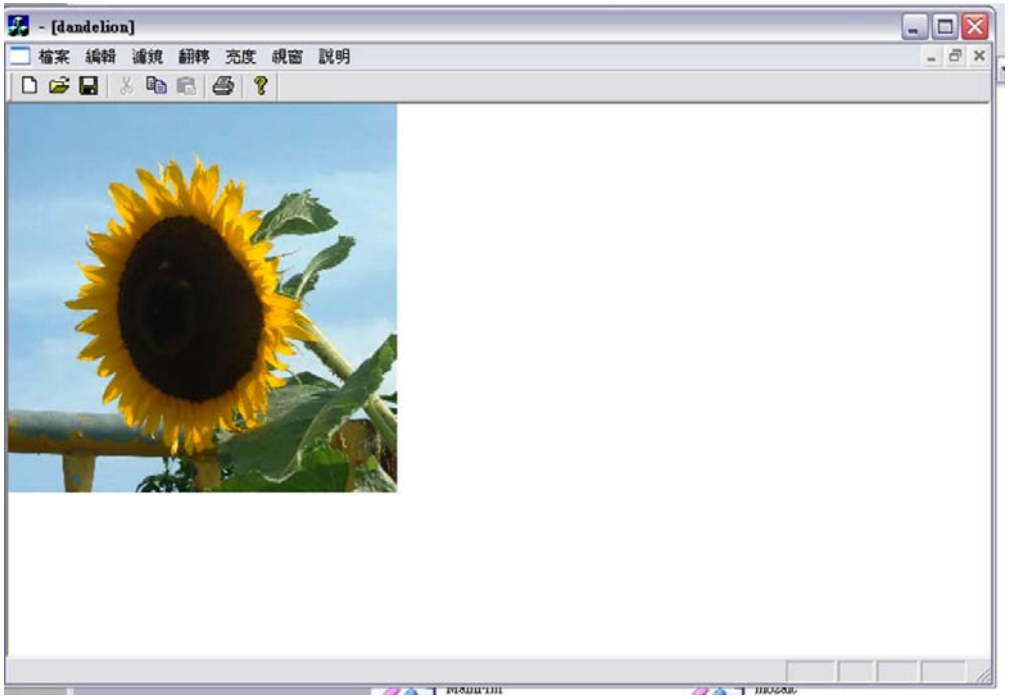
ASSEMBLY FINAL PROJECT

B94902105 資工三 陣一慧光

用C++跟ASSEMBLY作簡單的繪圖軟體



摘要、規格

- 摘要
  設計一個繪圖軟體
  沒有辦法全部都用assembly 所以用C++作顯示圖片、save跟load
  compiler使用Visual C++
  用assembly作負片效果，調整亮度，黑白，對比，翻轉，四分割，馬賽克，銳利化，模糊

- 規格
  只能用Windows的DIB或BMP檔
  24bit full color
  所有的assembly都寫在cpp檔的"_asm{"跟"}"之間

什麼叫DIB？

DIB就是BMP

Device Independent Bitmap

DIB（與設備無關的點陣圖）是微軟視窗圖形子系統（GDI）內部使用的一種點陣圖格式，它是微軟視窗平臺上的一個簡單的圖形文件格式。

By Wikipedia

負片效果

負片效果

- 效果：轉正片跟負片

- Algorithm：如果原本的照片的輝度是x%的話
  轉過的亮度是(100-x)%
  RGB是0~255
  所以算出255-data就可以

  先用loop把row全部轉好，然後再用loop把每個column都去轉

調整亮度

調整亮度

- 效果：調整亮度

- Algorithm：假設原本的照片的亮度是100%
  想要亮一點的話亮度改成141% 改兩次就會變成
  141%*141%=198.81% 大約兩倍
  想要暗一點的話亮度改成71% 改兩次就會變成
  71%*71%=50.41% 大約1/2倍
  141%要先乘141再除100來實現
  71%要先乘100再除141來實現
  跟負片效果一樣先用loop把橫的row全部轉好，然後
  再用loop把每個column都去轉

調整亮度

71%爲什麼不先乘71再除100？

71%*71%=50.41%

100/141*100/141=10000/19881

=0.502992807202856999144912227755 14…

所以先乘100再除141的方式比較可以接近50%

黑白

黑白

- 效果：轉成黑白照片

- Algorithm：黑白照片是RGB的值都一樣

可是單純的去算RGB的平均的話他的輝度會不一樣

假設白色的輝度爲100，R是30%，G是59%，B是11%

(我朋友告訴我的)

Gray的輝度

= R的輝度*0.3 + G的輝度*0.59 + B的輝度*0.11

所以(R*30+G*59+B*11)/100

對比(CONTRAST)

對比(CONTRAST)

- 效果：一般的繪圖軟體可以調-100~100
  我只作出100 把照片轉成8色照片

R=0 G=0 B=0	R=255 G=0 B=0	R=0 G=255 B=0	R=0 G=0 B=255
R=255 G=255 B=0	R=255 G=0 B=255	R=0 G=255 B=255	R,G,B=255

- Algorithm：RGB各個改成0或255 就會變成8色
  如果各輝度超過變數level的話改成255，以下的話改成0

  這個project的變數level = 128

翻轉

翻轉

- 效果：水平翻轉or垂直翻轉or180度旋轉
- Algorithm：交換最左邊跟最右邊，最上面跟最下面的點，用loop反覆(寬度/2)次or(高度/2)次

  水平翻轉了之後垂直翻轉就會變成轉180度

四分割

四分割

- 效果：把圖片縮成1/2然後排四張
  四分割之後再做同樣的動作就會變成八分割、十六分割

- Algorithm：讀取第偶數個點之後準備新的畫布，寫近四個地方
  例如 (x,y)=(10,20)的點寫進新的畫布的(5,10)、(寬度/2+5,10)、(5,高度/2+10)跟(寬度/2+5,高度/2+1)四個地方

馬賽克

馬賽克

- 效果：馬賽克
- Algorithm：取得2*2的四個點的RGB各平均，然後塞到同樣的四個點裡面去

鋭利化

銳利化

- 效果：把模糊的圖片變成明顯一點

- Algorithm：把一個點的輝度乘五倍，然後從那個值減上下左右的點的輝度，結果丟到新的畫布上。
  如果周圍沒有四個點的話輝度乘三倍再去減周圍的幾個點。

模糊

模糊

- 效果：把圖片變模糊一點

- Algorithm：算出中間一個點跟週圍的點總共九個點的平均，放到新的畫布的一個點裡面
  如果周圍沒有點就只用存在的點做同樣的動作

謝謝