

# Advanced Computer Graphics HW2 Report&操作說明

ntu\_r06922063 邱政凱

使用教學:

可以直接執行 hw2.exe。它會自動讀取同個路徑下名為 input.txt 的檔案當作輸入。若要自行指定檔名，可使用 cmd `./hw2 “檔名”`，以參數的形式把要使用的 input 檔名傳入。程式執行完會輸出一個名為 "colorOutput.ppm" 的 ppm 圖片檔，請使用 `infraview` 來檢視。

專案中有一個 codeblocks 的專案檔 hw2.cbp，可以使用此專案檔配合 code blocks 來開起來編譯。

hw2.cpp：這次作業的程式本體

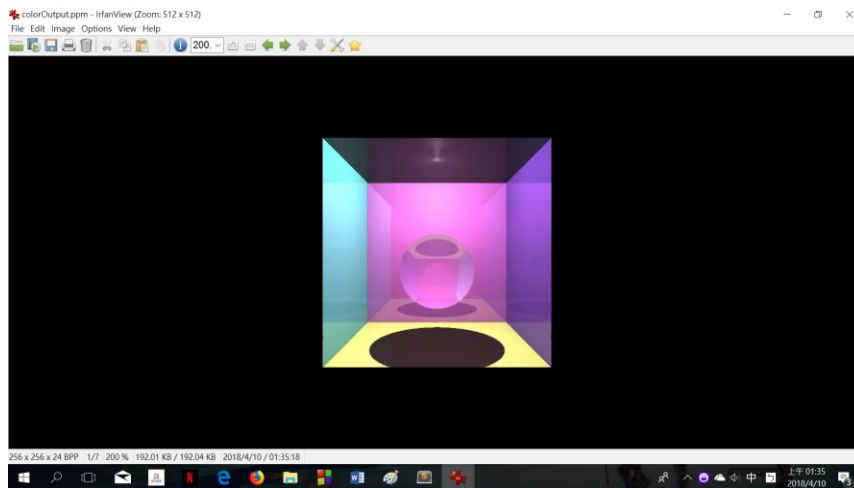
algebra3.cpp: 這次作業使用老師提供的 algebra3 函式庫

imageio.cpp：ppm 檔案相關的讀寫操作。

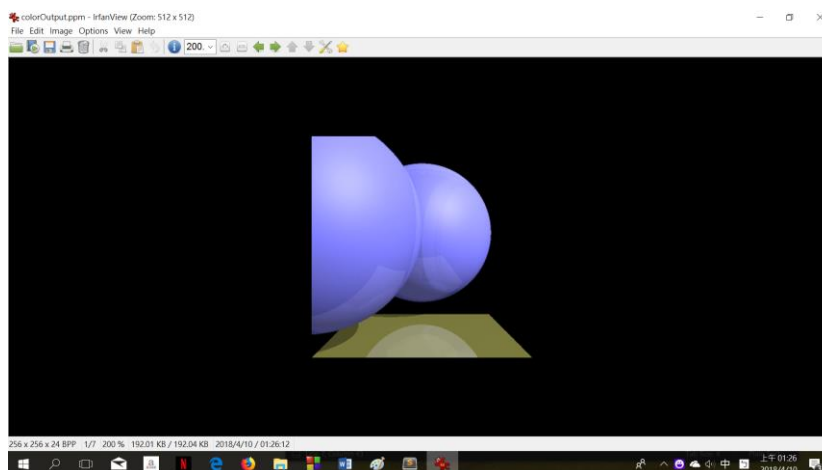
---

結果預覽: (分別使用我的 input.txt ~input6.txt，input6.txt 是老師給的 example input，其中除了 input1 和 5 的 culling 設成 true 以外，其他 culling 都設成 false(1 和 5 因為是 refraction，整個渲染在一個 box 中進行，因為想要從外面觀看，所以做了 culling)，並且都有使用 super sampling 進行 anti-aliasing)

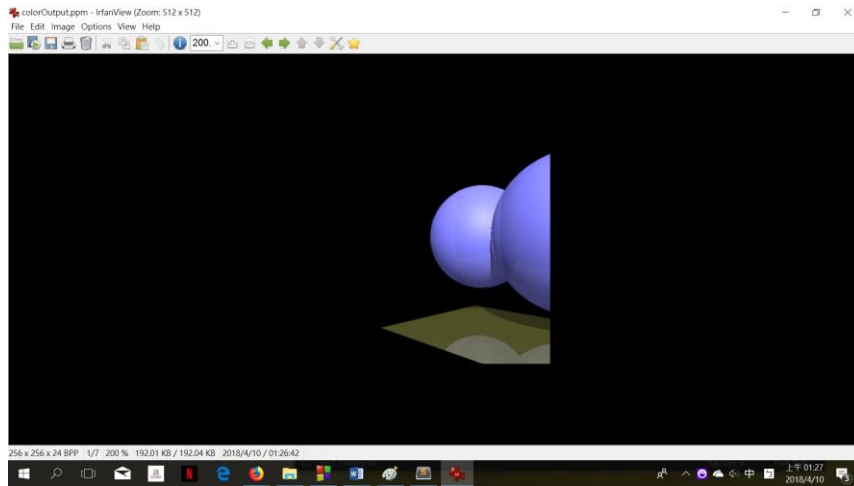
Input.txt:



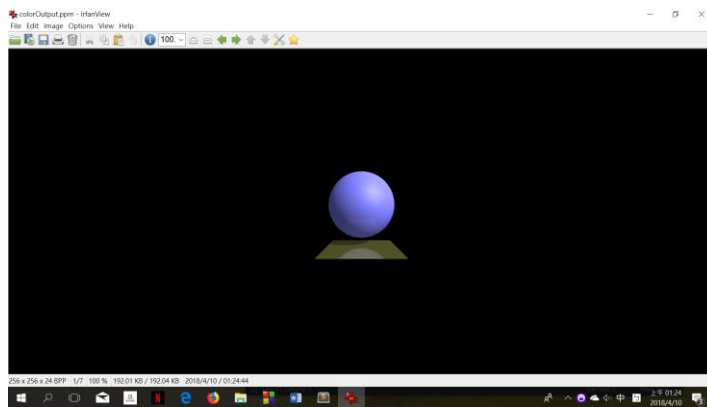
Input2.txt:



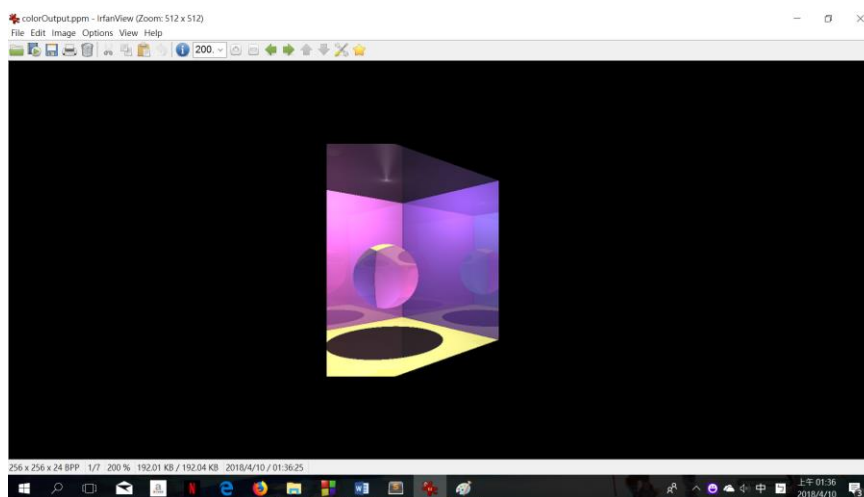
Input3.txt:



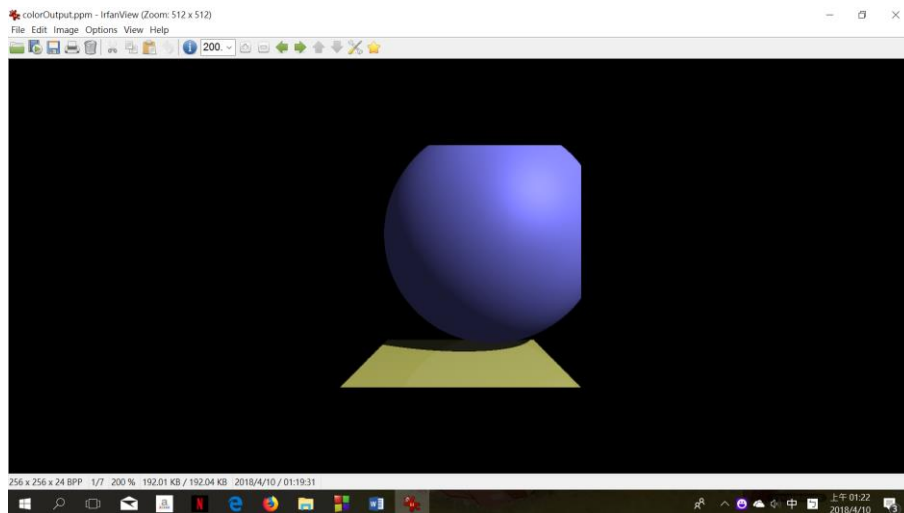
Input4.txt



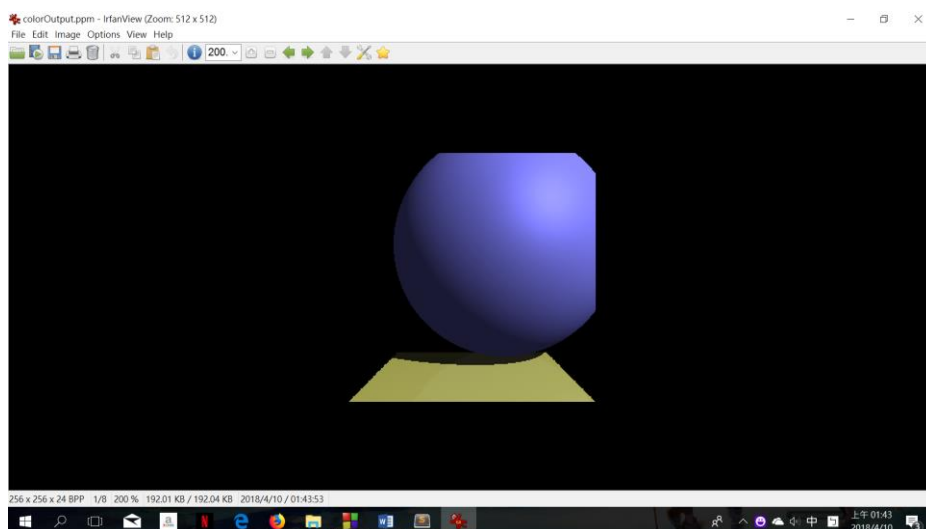
Input5.txt



Input6.txt



另外以下是沒有做 super sampling anti-aliasing 的結果，可以看到影子邊緣的鋸齒狀很嚴重。



參數說明:

Hw2.cpp 中有一些可以自行設定影響渲染結果的參數，全部都在檔案的最上方，他們的說明如下：

**backgroundColor**：若某個 pixel 的 ray 一開始從 camera 射出時就什麼都沒打到，把該 pixel 填上該顏色

**ambientIntensity** : Ambient light 的強度，因為 input file 中沒指定，故自行設置。

**shadowBias** : 判斷某點是否被某個物體遮蔽住時，為了避免往 light 方向設出去的 ray 馬上跟自己的 mesh 相交，因此用一個 shadowBias 乘上該點的 normal，把該點先往外凸出一點再進行設限的判斷。

**airReractivity**: 空氣(場景中不是 object 的地方)的折射率。

**SSAA\_RATIO** : 進行 Super sampling Anti aliasing 時，使用的 super resolution 倍率 (2 則會渲染 2x2 倍大的畫面並進行平均)

**Culling** : 畢竟 input 檔沒指名像是三角形的法向應該是往哪個方向，若 culling 開啟，則若某個 RAY 跟某個 MESH 的法向的內積 $>0$ ，則判斷此 mesh 應該被 cull 掉。若 culling 設成 false，則會把 normal 反轉跟 ray 做計算。

---

心得&報告:

這次作業基本上就是把基本的 ray tracer 完成，必須遞迴地反射和折射光線渲染更逼真的場景。

基本上演算法就是每次射出 ray 時，判斷場景裝所有物是否與其相交，取出離發射點距離最近打到的 mesh，並且根據該 mesh 的 material 參數計算那次"打到"對於整個 color 的 contribution，接著遞迴呼叫 tracing 函式根據該 mesh 的反射和折射率打出額外的兩條光

線，並且回傳他們一路累積的顏色能量，接著把”最初的 mesh 自己的顏色”+”反射的光一路累積的顏色”+”折射的光一路累積的顏色”相加就是該 pixel 最後的顏色。

另外一點需要注意的就是每次要判斷某個 mesh 是否會對目前的 ray 造成顏色的 contribution 時，需要判斷所有光源對於該點的貢獻，並且也要記得判斷如果該點跟某個光的中間被某個物體阻擋了，貢獻值就為 0(hard shadow，從我的 input1 和 5 可以看到 hard shadow 的效果)

最後一點比較特別的大概就是我還時做了 super sampling anti-aliasing。在一開始創建渲染的畫布時，就創造原本指定解析度 N 倍的解析度畫布，渲染完成後把每  $n*n$  個區域的值做平均當成最後指定大小的圖片對應 pixel 位置的值。