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。

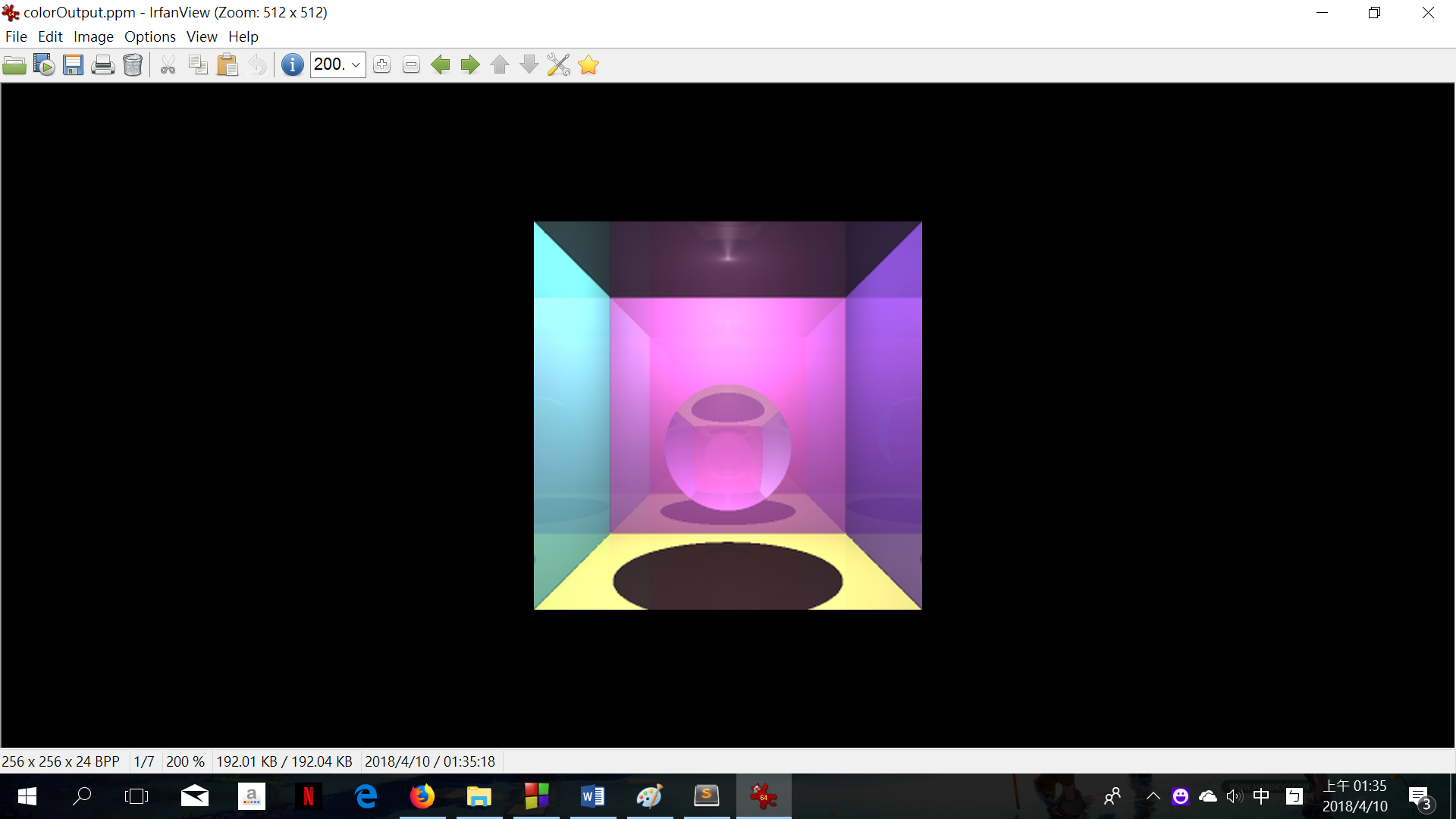
專案中有一個codeblocks的專案檔hw2.cbp，可以使用此專案檔配合code blocks來開起來編譯。

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

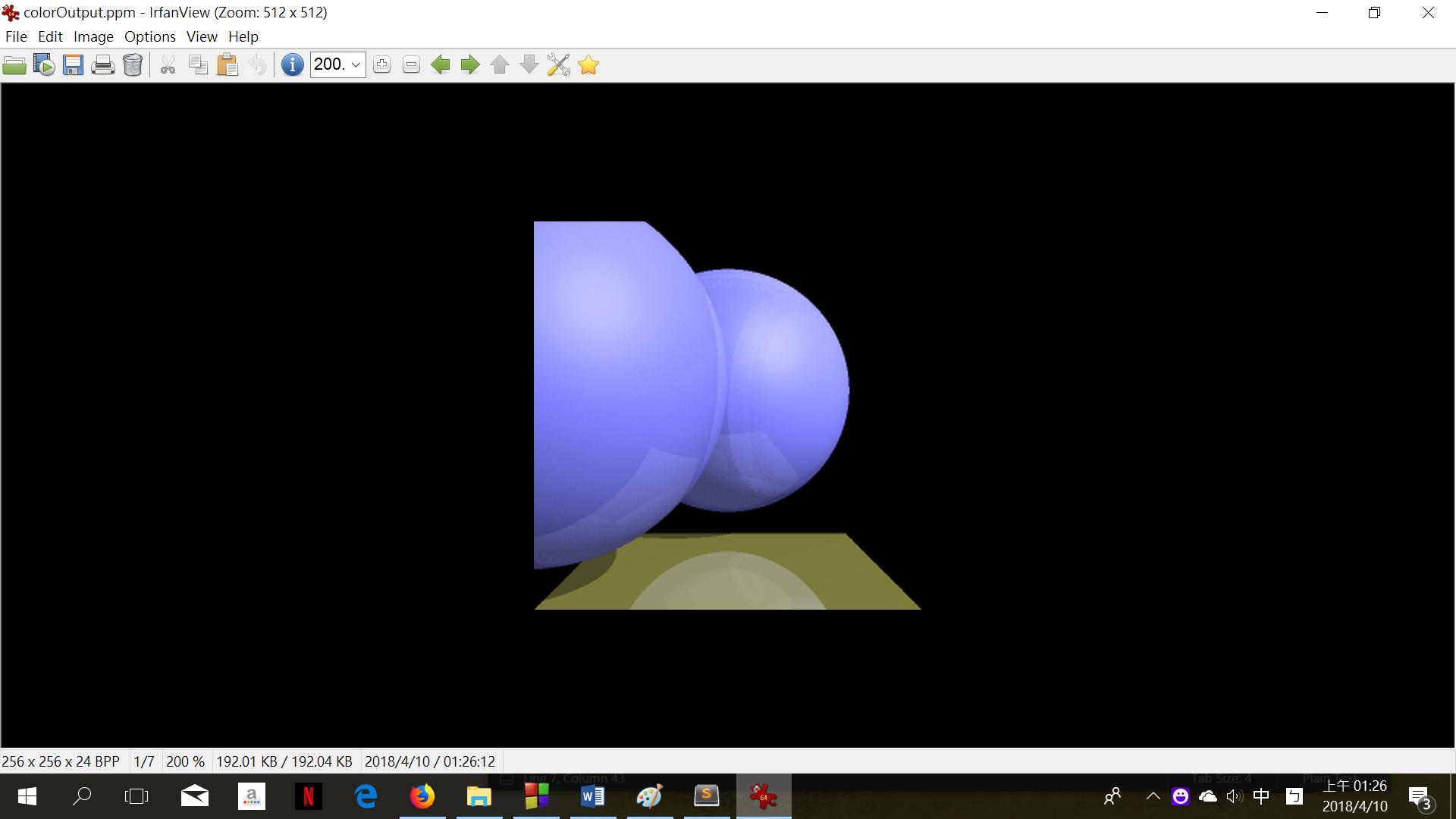
imageio.cpp : ppm檔案相關的讀寫操作。

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)

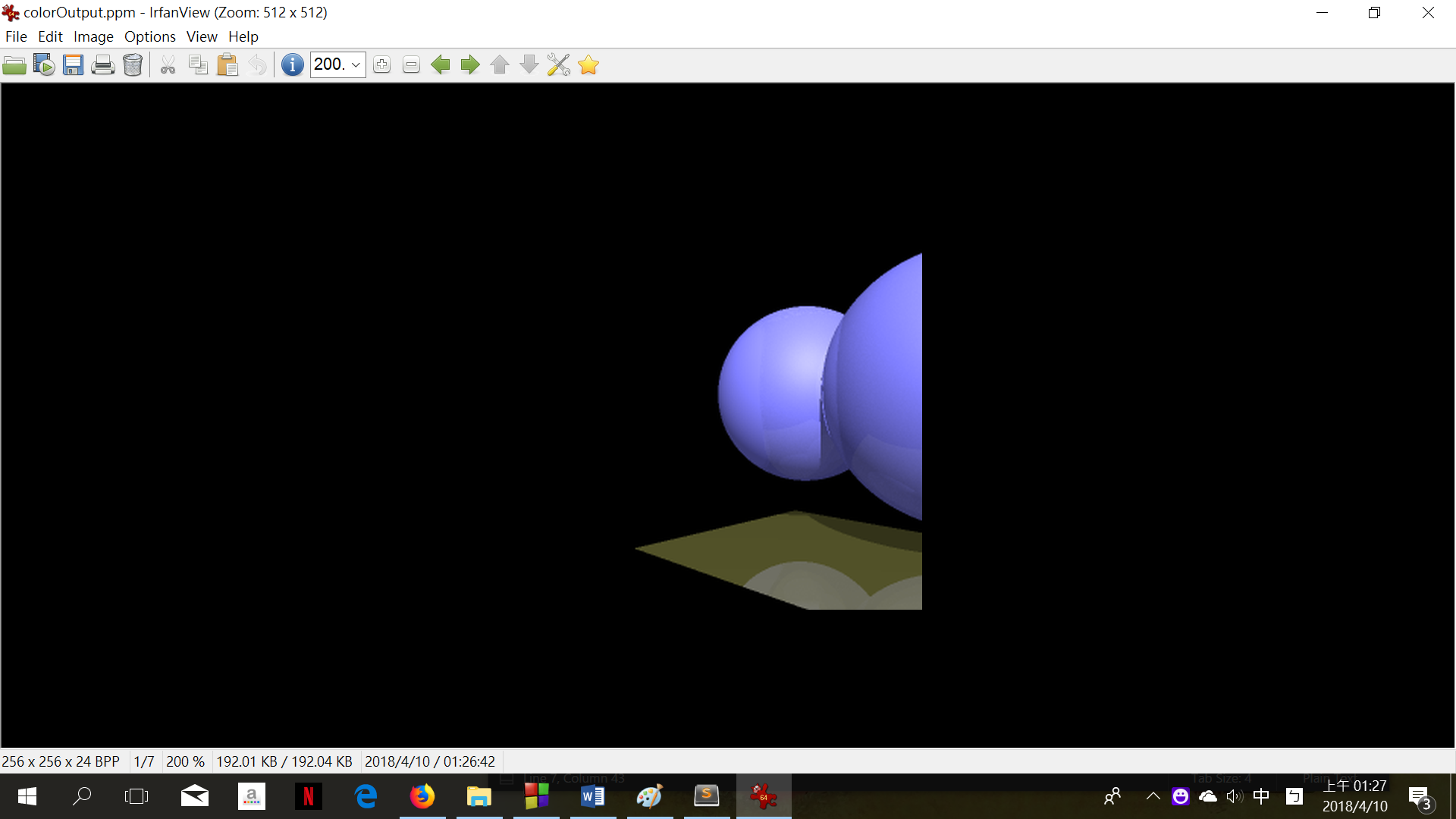
Input.txt:



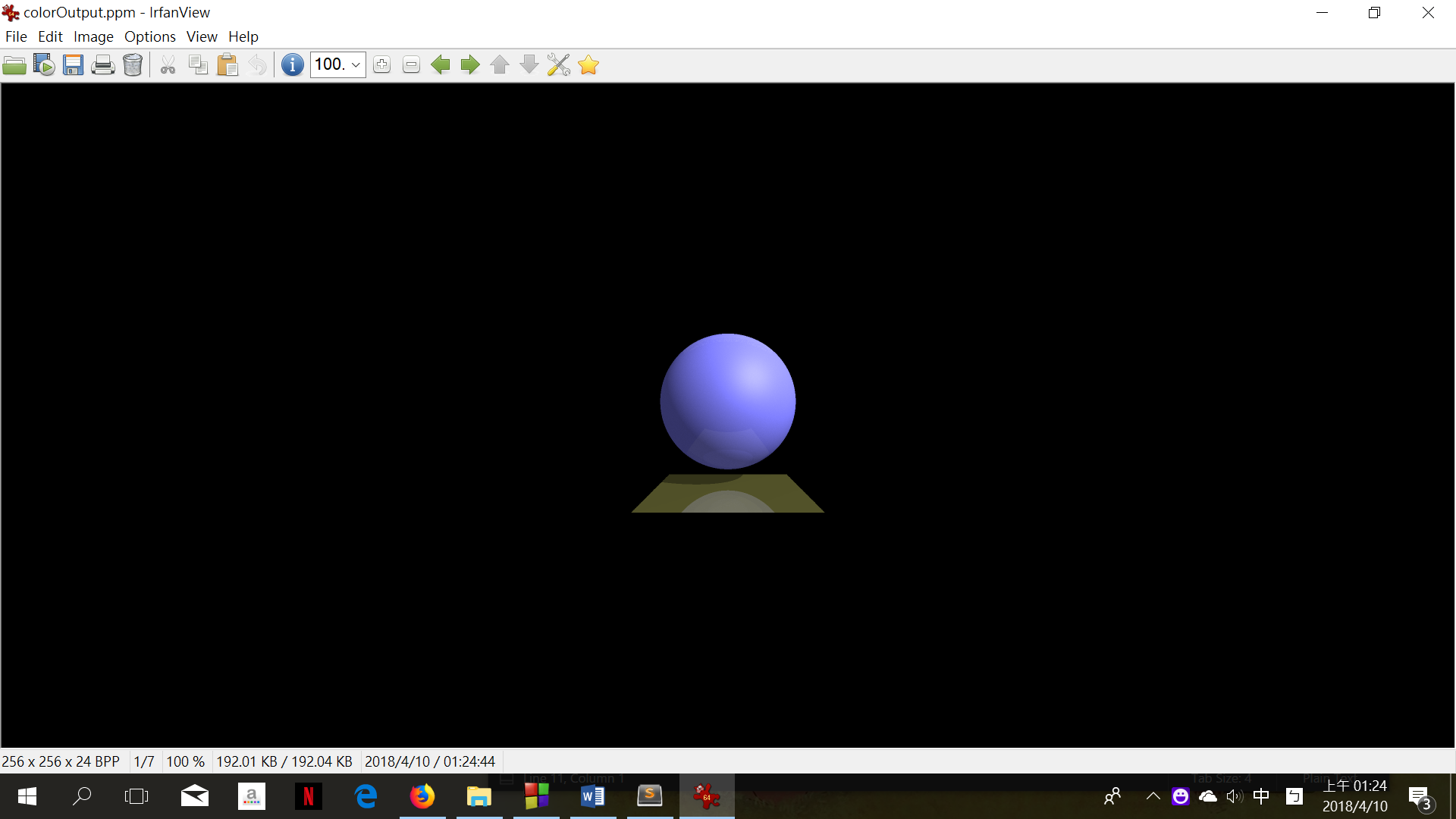
Input2.txt:



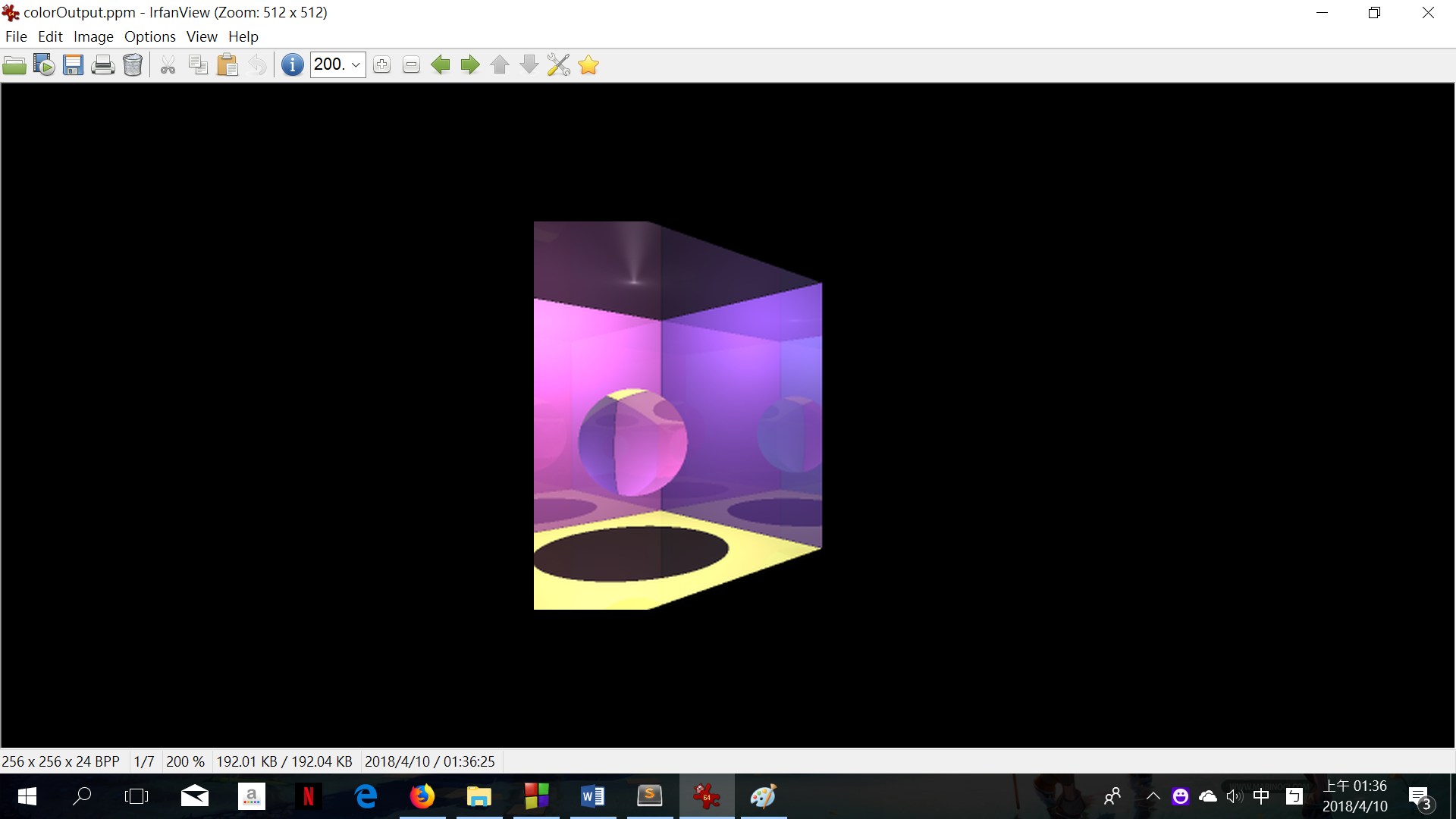
Input3.txt:



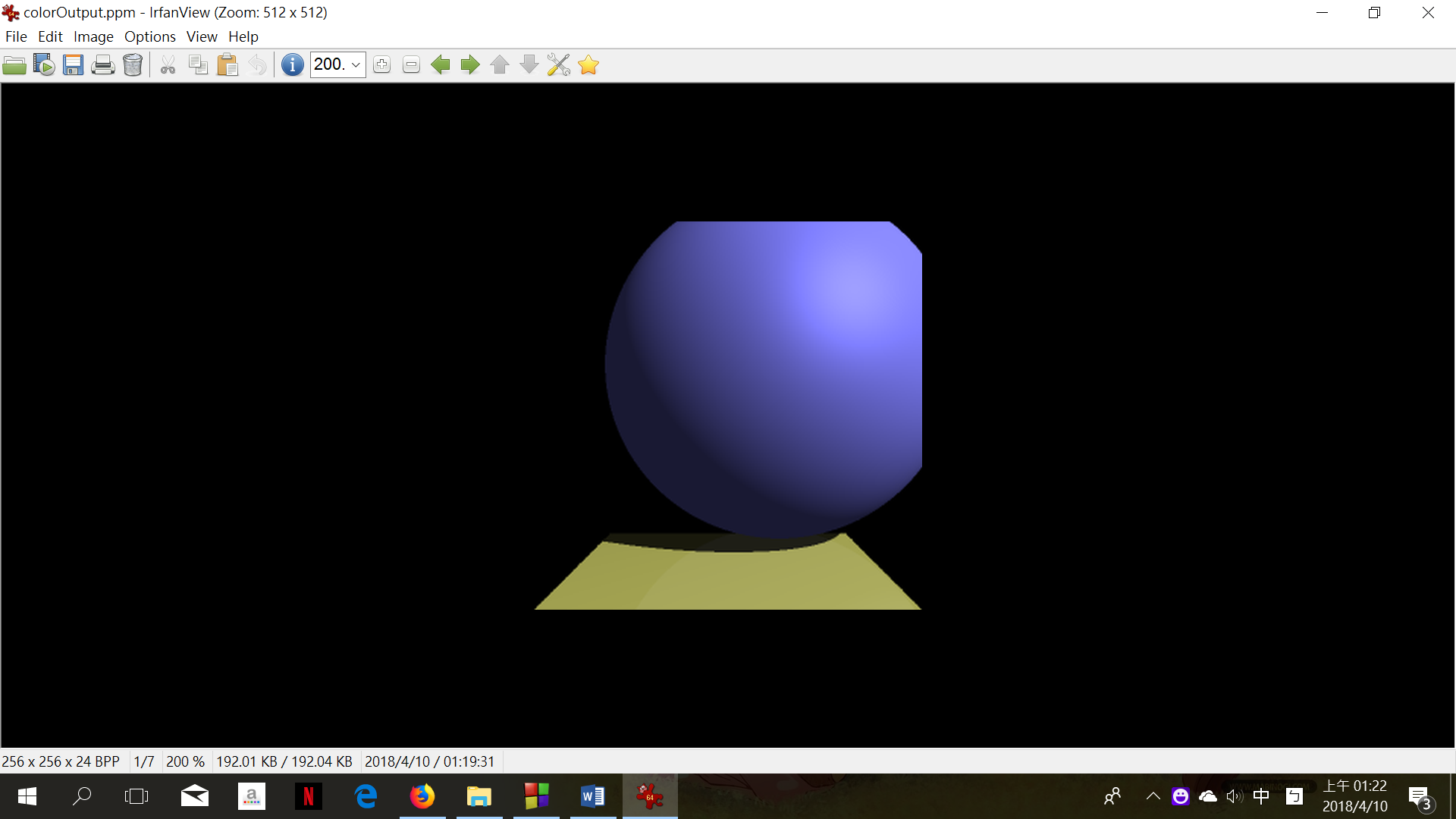
Input4.txt



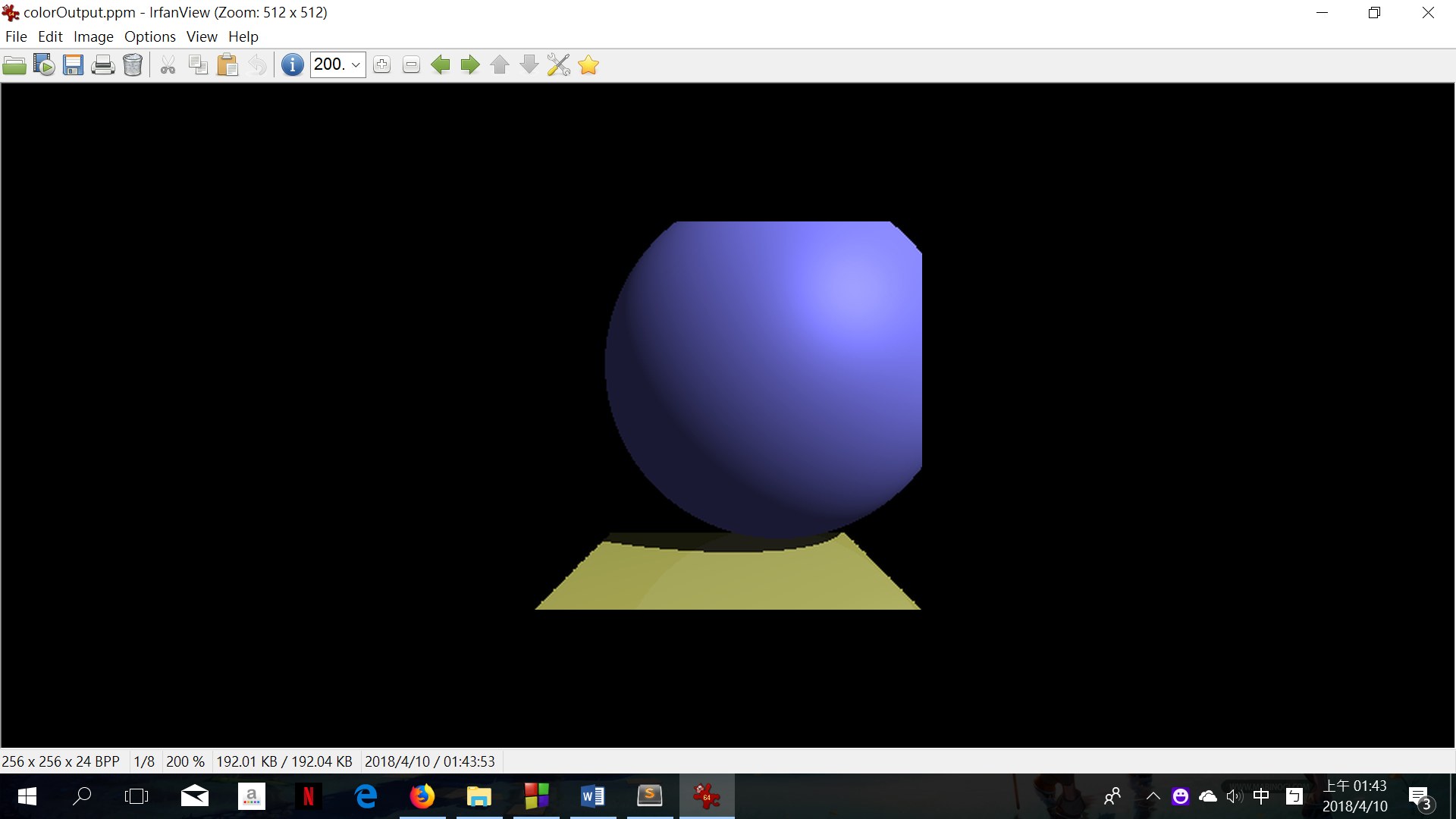
Input5.txt



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香交，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ㄤ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”香加就是該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。