

# Computer Graphics Assignment 4 Report

102062209 邱政凱

要啟動程式，請進入 CG\_HW4\_framework->CG\_HW4->CG\_HW4 啟動 CG\_HW4\_framework.exe 檔。操作方式承襲助教原本就寫在 framework 裡面的切換模型/Transformation 之外，藉由鍵盤上方的 1/2/3 來切換 Directional/Position/Spot Light、4 把燈關掉、5/6 增加/減少燈位置的 x 值、7/8 增加/減少燈位置的 y 值、9/0 增加/減少燈位置的 z 值。w/W 來切換 Wrap mode、M 切換 Magnify texture mapping 時的方式、m 切換 Minify texture mapping 時的方式、T 開啟/關閉 Texture Mapping、V/v 切換 Vertex Shader Lighting/Per-Pixel Lighting。

這次的作業要求我們承襲作業二的 Transformation 和作業三的 Lighting，再加入 Texture Mapping 的功能。(不過 Transformation 的部分助教給的 framework 已經寫的很好了所以可以不用管)。這次的實作幾乎都只需要用到各種 API 的呼叫，其實不需要用到什麼複雜的 CODE，只要清楚哪個 API 要在什麼時候 CALL 就好了，以及搞清楚每個 API 各自的功用是什麼，這一切都可以請教 google 大神，不過讓我搞混最久的就是 glActiveTexture()跟 glBindtexture()了，一開始一直把前者當後者用。Lighting 的部分也的確是把作業三的部分拿來再用就好了，只不過要注意的是 Fragment shader 的參數必須在 Vertex shader 中用 Varying 傳過去，而且必須設條件判斷，要不然兩個 shader 一起執行 Lighting 運算結果會有點詭異；我覺得 GLSL 寫起來最麻煩的部分應該就是 Debug 的部分吧，不能使用傳統 Printf Debug 法，要不然就是要用第三方工具或者是用不同的 gl\_FragColor 來當作 Printf 來代替使用(不過超難看懂的啊!!!)。不過也因為這次的作業同時處理了兩邊的 shader，總算稍微可以理解兩個 shader 不同的使用方式和功用。另外還有就是最後一個 cube 模型的部分，如果直接用 glm 的 parser 把參數抓進來會發現在要給 vnormals 賦值時會出現錯誤訊息，點進去 cube 的.obj file 看了一下才發現裡面沒有 normal 的資訊，必須用 glmFacetNormal 和 glVertexNormals 自己呼叫函式生成法向量才行。(這部分最賤了，其他模型都有在.OBJ 檔裡面都有附 normal 資訊只有這個沒有-.-)。



